Fuel Optimal Control Algorithms for Connected and Automated Plug-In Hybrid Vehicles

by

Di Chen

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in the University of Michigan
2021

Doctoral Committee:

Assistant Professor Youngki Kim, Co-Chair
Professor Anna Stefanopoulou, Co-Chair
Dr. Kenneth Butts, Toyota Motor North America R&D
Dr. Mike Huang, Sunrise Instruments LLC
Professor Ilya Kolmanovsky

Di Chen
dichencd@umich.edu
ORCID iD: 0000-0002-5202-566X

To my parents Shujie Chen and Junhong Wu, whose love and support are always present in my mind.

ACKNOWLEDGMENTS

First and foremost, I want to express my gratitude towards my Ph.D. advisor, Professor Anna Stefanopoulou, for allowing me to embark on the Ph.D. program. Thanks for believing in me and guiding me through graduate school. Second, I also want to thank Professor Youngki Kim, who has supported me academically and mentally since the first day of my Ph.D. journey. I also want to thank Professor Ilya Kolmanovsky, Dr. Kenneth Butts, and Dr. Mike Huang for joining my dissertation committee, taking the time and interest in evaluating my work, and providing valuable comments, guidance, and feedback.

The financial support for this research was provided by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0000837, also known as NEXTCAR. I want to thank Southwest Research Institute for the opportunity to be part of the exciting ARPA-E project.

I want to thank Dr. Mike Huang, who opened the door of numerical optimization to me, making me who I am today. My interest in numerical optimization began when Mike encouraged me to write my solver to replace fmincon and have since been loving do-everything-by-hand. Thank you, Mike, for all the trust and encouragement along the way.

I want to thank my colleagues who have helped and influenced me throughout this journey. Dr. Nan Li, Dr. Yuan Yao, Dr. Yan Chang, Dr. Niket Prakash, Dr. Bryan Maldonado, Dr. Shengqi Zhang, Saravanan Duraiarasan, and Mitchell Bieniek.

I want to thank my host, Dr. Claus Danielson, during my internship with Mitsubishi Electric Research Labs, who gave me the opportunity in the first place and then trusted me with the second one. I am also very grateful to work with Dr. Stefano Di Cairano during my second internship. Both of them are theoretically strong and push me to analyze and tackle a problem from a fundamental level. I am not working on the theoretical side, but both have shown extreme patience in pointing out the inaccuracies in my attempt to provide a more rigorous justification.

I also want to thank all my friends who supported me to get through the difficult time of Covid-19, Zitong Zhang, Changhao Li, Qijun Yao.

Last but not least, I want to thank my parents Shujie Chen and Junhong Wu, and my boyfriend, Kuan Liu, to whom I owe more than I can give back. Thanks for the love and support and for believing in my dreams.

TABLE OF CONTENTS

Chapter

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

LIST OF ALGORITHMS

ALGORITHM

LIST OF ABBREVIATIONS

PHEV plug-in hybrid electric vehicle

HEV hybrid electric vehicle

OCV open circuit voltage

MG2 motor

MG1 generator

NN neural network

SOC battery state-of-charge

MPC model predictive control

CDCS charge-depleting charge-sustaining

TPBVP two-point boundary value problem

DMIS Discrete Mixed-Integer Shooting

VD vehicle dynamics

PT powertrain

DP dynamic programming

DMP Discrete Maximum Principle

ADAS advanced driver assistance systems

LQ linear quadratic

PMP Pontryagin's Minimum Principle

ACC adaptive cruise control

CAV connected and automated vehicle

ICE internal combustion engine

SPaT the Signal Phase and Timing

EV electric vehicle

ECO-ACC ecological adaptive cruise controller

SOC state-of-charge

HV hybrid vehicle

SQP sequential quadratic programming

QP quadratic program

approximate-PI approximate policy iteration

EMPC econominc model predictive control

LV lead vehicle

TOP trajectory optimization problem

LTI linear time-invariant

LMI linear matrix inequality

RL reinforcement learning

TD temporal difference

CBF control barrier function

P-ECMS predictive equivalent minimization strategy

A-ECMS adaptive equivalent minimization strategy

DM Dual Motor Drive

ECMS equivalent minimization strategy

CACC cooperative adaptive cruise control

FE fuel economy

TPBVP two-point boundary value problem

OCP optimal control problem

KKT Karush-Kuhn-Tucker

MDP Markov Decision Process

FBRS regularized and smoothed Fischer-Burmeister method

NCP nonlinear complementarity problem

MIQP mixed-integer quadratic program

ABSTRACT

Improving the fuel economy of light-duty vehicles (LDV) is a compelling solution to stabilizing Greenhouse Gas (GHG) emissions and decreasing the reliance on fossil fuels. Over the years, there has been a considerable shift in the market of LDVs toward powertrain electrification, and plug-in hybrid electric vehicles (PHEVs) are considered to be the most cost-effective in avoiding GHG emissions. Meanwhile, the development of connected and automated vehicle (CAV) technologies permits energy-efficient driving with access to accurate trip information that integrates traffic and charging infrastructure. This thesis aims at developing optimization-based algorithms for controlling powertrain and vehicle longitudinal dynamics to fully exploit the potential for reducing fuel consumption of individual PHEVs by utilizing CAV technologies.

For a human-driven PHEV, a predictive equivalent minimization strategy (P-ECMS) is proposed to adjust the co-state based on the difference between the future battery state-of-charge (SOC) obtained from short-horizon prediction and a future reference SOC from SOC node planning. The SOC node planning which generates battery SOC reference waypoints, is performed using a simplified speed profile constructed from segmented traffic information, typically available from mobile mapping applications.

The PHEV powertrain, consisting of engine and electric motors, is mathematically modeled as a hybrid system as the state is defined by the values of the continuous variable, SOC, and discrete modes, hybrid vehicle (HV) and electric vehicle (EV) modes with the engine on/off. As a hybrid system, the optimal control of PHEVs necessitates a numerical approach to solving a mixed-integer optimization problem. It is of interest to have a unified numerical algorithm for solving such mixed-integer optimal control problems with many states and control inputs. Based on a Discrete Maximum Principle (DMP), a Discrete Mixed-Integer Shooting (DMIS) algorithm is proposed. The DMIS is demonstrated in successfully addressing the cranking fuel optimization in the energy management of a PHEV. It also serves as the foundation of the co-optimization problem considered in the remaining part of the thesis.

This thesis further investigates different control designs with an increased vehicle automation level combining vehicle dynamics and powertrain of PHEVs in within-a-lane traffic flow. This thesis starts with a sequential (or decentralized) optimization and then advances to direct fuel minimization by simultaneously optimizing the two subsystems in a centralized manner. When shifting toward online implementation, the unique challenge lies in the conflict between the long control horizon required for global optimality and the computational power limit. A receding horizon strategy is proposed to resolve the conflict between the horizon length and the computation complexity, with co-states approximating the future cost. In particular, the co-state is updated using a nominal trajectory and the temporal difference (TD) error based on the co-state dynamics.

The remaining work aims to develop a unified model predictive control (MPC) framework from the powertrain (PT) control of a human-driven to the combined vehicle dynamics (VD) and PT control of an automated PHEV. In the unified framework, the cost-to-go (the fuel consumption as the economic cost) is represented by the co-state associated with the battery SOC dynamics. In its application to automated PHEVs, a control barrier function (CBF) is augmented as an add-on block to modify the vehicle-level control input for guaranteed safety. This unified MPC framework allows for systematically evaluating the fuel economy and drivability performance of different levels and structures of optimization strategies.

CHAPTER 1

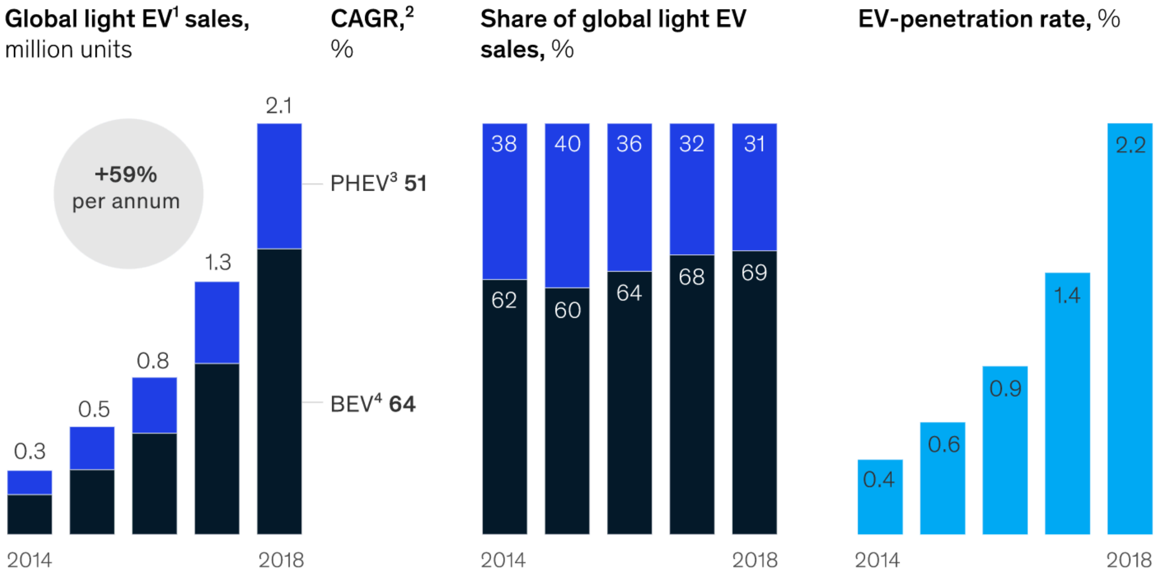
Introduction

1.1 Background

The transportation sector is responsible for 28% of the total U.S. Greenhouse Gas (GHG) emissions [6], 83% of which are emitted from vehicles, with the Light Duty Vehicle (LDV) segment being the largest contributor [7]. On the other hand, in 2018, the United States imported about 11% of the petroleum it consumed. Improving the fuel economy of LDVs is a compelling solution to stabilizing GHG emissions and decreasing the reliance on fossil fuels because transportation accounts for nearly three-fourths of total U.S. petroleum consumption. Over the years, there has been a considerable shift in the market of LDVs toward more efficient powertrains, especially powertrain electrification.

Hybrid Electric Vehicles (HEVs), Plug-in Hybrid Electric Vehicles (PHEVs), and Battery Electric Vehicles (BEVs) are examples of efforts in powertrain electrification for reducing GHG emissions. Over the years, the global EV industry continues to expand rapidly, as shown in Fig. 1.1. Electic vehicle sales grew to more than two million units globally in 2018: an increase of 63% on a year-on-year basis and at a rate slightly higher compared to previous years. The ratio of battery EVs (BEVs) to plug-in hybrid EVs (PHEVs) held relatively steady from 2017 [1]. HEVs and PHEVs typically use less fuel than conventional vehicles because they employ electric-drive technologies to boost efficiency. PHEVs and BEVs are both capable of being powered solely by electricity produced in the U.S. from natural gas, coal, nuclear energy, and renewable resources. PHEVs and BEVs can reduce fuel costs dramatically because of the high efficiency of electric-drive components.

Figure 1.2a shows that more considerable GHG reductions for LDVs are achieved with low-carbon fuels, and vehicle efficiency improvement [8]. Gasoline HEVs can reduce C2G GHG emissions to below 350g CO_2e/mi, as can other advanced vehicle technologies, such as PHEVs, BEVs, and FCEVs. However, today's BEVs have the disadvantage of a shorter range (per charge) than comparable conventional vehicles have (per tank of gas). The efficiency and driving range of BEVs vary substantially based on driving conditions. Also, there are environmental problems since BEVs are not the best in cold weather and on rural routes. Therefore, consumer acceptance of FCEVs and BEVs may be slowed by refueling inconvenience (scarcity of hydrogen stations, limited range, and long charging time with batteries). Given the wide availability of gasoline and diesel stations, the burden to find an alternative fuel source is an important barrier to market penetration [8]. Although the global BEVs market is amped up and on the rise, especially in China and Europe [9, 1, 10], the momentum is slowed down in the United States since the fuel price remains low [9].

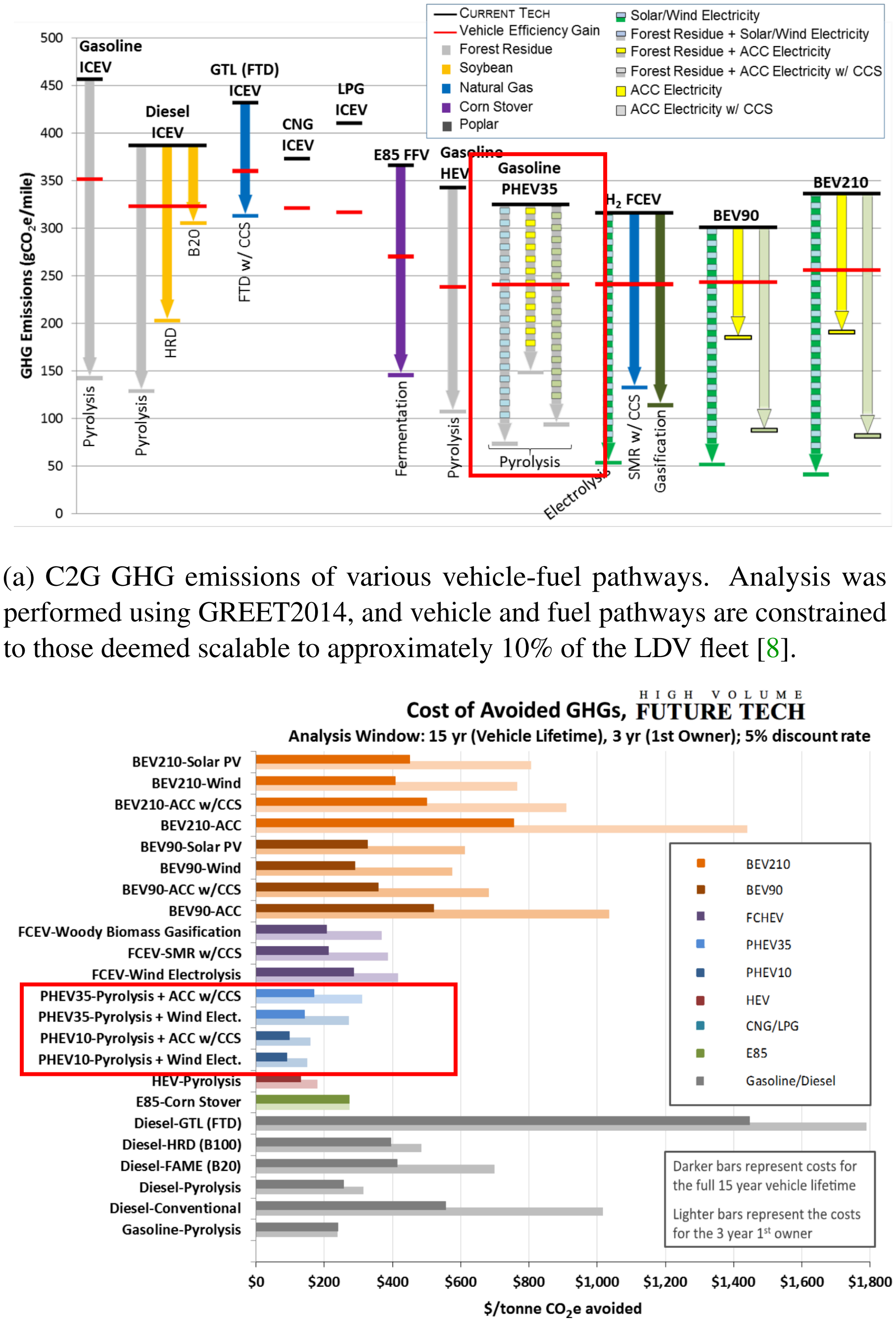


¹Electric vehicle. ²Compound annual growth rate. ³Plug-in hybrid electric vehicle. ⁴Battery electric vehicle.
Source: EV-Volumes.com; McKinsey analysis

Figure 1.1: The global market for electric vehicles has grown at about 60% per year, reaching 2.1 million in 2018 [1].

Among all the modern production vehicles, PHEVs are hybrids with high-capacity batteries that can be charged by plugging them into an electrical outlet or charging station. They can store enough electricity to reduce their petroleum use under typical driving conditions [11]. There are two main benefits with PHEVs among production vehicles that are well accepted by consumers, as highlighted in Fig. 1.2. On the one hand, among all the current technologies, PHEVs are shown to have the potential to emit less greenhouse gas, as shown in Fig. 1.2a. On the other hand, as seen from Fig. 1.2b, PHEVs are predicted to be the most cost-effective in avoiding GHG emissions. Combining the potential in reducing GHG emissions and the cost efficiency associated with avoided GHG emissions, PHEVs are still considered an important part of the conventional automotive industry, despite the drastic advancement in pure BEVs.

In the meantime, for PHEVs with two energy sources comes naturally the need for proper



(a) C2G GHG emissions of various vehicle-fuel pathways. Analysis was performed using GREET2014, and vehicle and fuel pathways are constrained to those deemed scalable to approximately 10% of the LDV fleet [8].

(b) Cost of avoiding GHG emissions by vehicle-fuel pathway for the future technology, high volume case (2013$), relative to the future technology gasoline ICEV [8].

Figure 1.2: Two main benefits with PHEVs compared to vehicles with different types of technologies.

coordination between fuel and electricity. Table. 1.1 illustrates some of the available PHEVs in the U.S. The average battery-only range is 22.6 miles. It means that for a typical daily commute without battery recharging in the middle, eventually, at some point in the day, fuel consumption would be inevitable [12]. For most of the production PHEVs, a charge-depleting charge-sustaining (CDCS) strategy is used, where the battery provides all the power needed (CD) until the battery charge is low before consuming fuel. The ICE then performs most of the propulsion in a Charge Sustaining (CS) mode for the remainder of the trip. The CS mode makes sure to keep a minimum level of battery charge and may involve occasional electric propulsion if enough energy is regenerated from braking [13]. This way, fuel will not be consumed when the trip is short. When the trip is long enough that it exceeds the pure EV range, knowing the fuel is only consumed because the battery has been fully depleted is more acceptable from a consumer point of view. The battery SOC trajectory with CDCS is conceptually illustrated in Fig. 1.3a. However, as identified in the literature, often, CDCS is not the most fuel-efficient energy management strategy. Charge blending (CB) strategies where the engine is used together with the battery throughout the entire trip, if designed properly, can lead to better fuel economy. The battery SOC trajectory with CB is conceptually illustrated in Fig. 1.3a. Figure 1.3b presents the offline optimization results under a scenario where the trip is long and exceeds the PHEVs' pure EV range. As can be seen from the first subplot, the initial and final SOC are the same in both cases. However, as seen from the second subplot, the total fuel consumed over the considered trip, even with optimized CS operation, still exceeds that with optimal charge blending (OCBD) strategy.

Although the potential of CB is realized through offline fuel minimization over a given trip, proper online CB strategies are still not mature for mass production. The following two main reasons are intertwined:

- Uncertainties: accurate information on the exact trip in advance may not be available. As indicated from the offline optimal CB strategy, CB strategies are only effective when considering the entire trip. Therefore, uncertainties in the future will inevitably degrade the performance of CB strategies.
- Inadequate battery depletion: A CDCS strategy uses battery first as needed. When the trip is long and exceeds the total battery range, the battery will be depleted to a low level before using any fuel. By comparison, it is nontrivial to prevent battery under-depletion for a CB strategy affected by future uncertainties. The battery under-depletion might lead to worse fuel economy than those with CDCS strategies.

The utilization of traffic information can benefit CB energy management strategies for electrified vehicles. Large-scale traffic monitoring systems would allow for the computation of a reference battery SOC trajectory with periodic updates based on new traffic information. The reference SOC can be used to solve the vehicle energy management in a receding horizon manner with

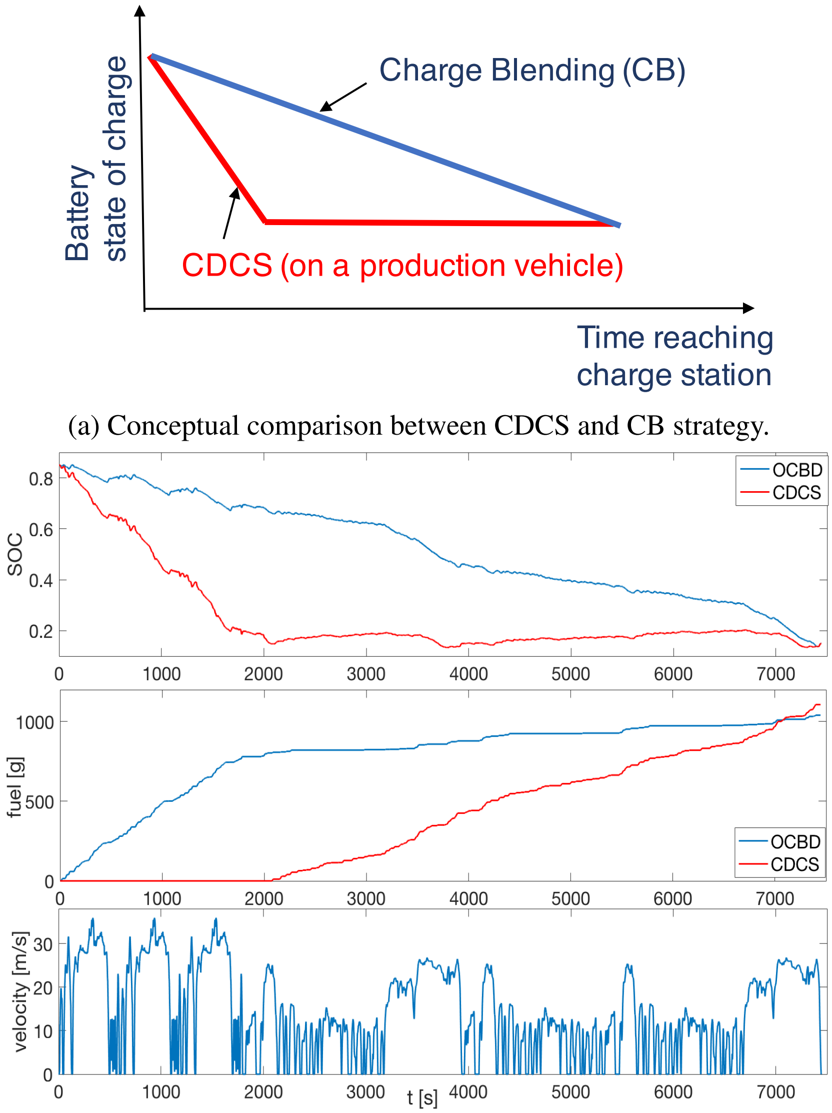
Table 1.1: Some examples of estimated EV mode range of available PHEVs

MAKE/MODEL	RANGE (MILE)
Chevrolet Volt	53
Honda Clarity PHEV	47
Chrysler Pacifica Hybrid	33
BMW 530e	30
Hyundai Sonata PHEV	27
Toyota Prius Prime	25
Ford Fusion Energi PHEV	21
Porsche Panamera 4 E-Hybrid	16
Mercedes C350e	11

increased confidence in forecasting future driving velocities. For instance, authors in [14] show that battery SOC planning level based on realtime traffic data with dynamic updates can achieve nearly 6% fuel economy gain compared to CDCS strategy.

Meanwhile, the automotive industry has witnessed a huge shift toward vehicle connectivity and automation, prompt to increase safety, driving comfort, and time-saving potential. With much easier access to information, increased processing power, and precision control, they also offer unprecedented opportunities for energy-efficient driving [15]. Specifically, connectivity to other vehicles and infrastructure allows better anticipation of upcoming events, such as hills, curves, slow traffic, state of traffic signals, and the movement of neighboring vehicles. Automation will enable vehicles to adjust their motion more precisely in anticipation of upcoming events and save energy [15, 16]. On the one hand, on-board or mobile navigation systems enabled by connectivity can receive the latest traffic information and road closings and adjust their route recommendations accordingly. Therefore, connectivity between cars and infrastructure can make much more information available to each vehicle. As a result, together with an increased level of automation, individual vehicles have the potential to drive with improved road safety and energy efficiency compared to a pure reactive human driver. Furthermore, in a connected vehicle world, automated vehicles can cooperate rather than compete with the exchange of intentions [15, 16]. Figure 1.4 schematically illustrates an example of the anticipative car following. Human drivers are often pure reactive when following other vehicles. By comparison, with additional information of the intent of preceding vehicles via V2V communication, a more anticipative car following can be achieved to maintain a safe following distance to the preceding vehicle. Enabled by automation, the inter-vehicular distance gap can be properly utilized to improve vehicle energy efficiency.

With the rapid development of technologies in driving automation and vehicle connectivity, it is possible to reduce the energy consumption of vehicles through energy-efficient driving



(a) Conceptual comparison between CDCS and CB strategy.

(b) Comparison between the CDCS with optimized CS operation and optimal charge blending (OCBD) on a stitched standard driving cycle. The trip is long enough to exceed the PHEV's pure EV range.

Figure 1.3: Comparison between CDCS and charge blending (CB).

[17]. Notably, for a PHEV with two energy sources (fuel and electricity), energy-efficient driving can expand further the fuel economy benefit beyond what can be achieved with the optimal power-split on an ordinary human driving. Efforts toward energy-efficient driving for (P)HEVs in the literature include [17, 18, 16, 19, 20, 21], to name a few. Especially in [16], comprehensive discussions are presented incorporating almost every aspect of CAVs for vehicles with different energy sources. Combining optimized control of vehicle dynamics and the optimized energy management strategy to maximize the overall system-level efficiency

becomes an active research topic and is under rapid development.

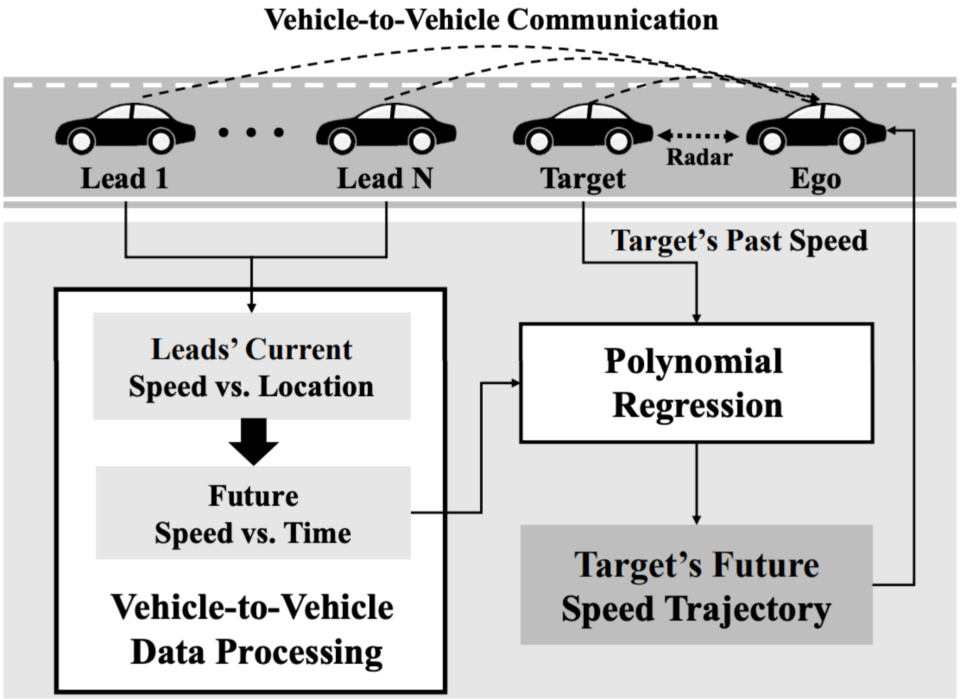

Vehicle-to-Vehicle Communication
Lead 1
Lead N
Target
Radar
Ego
Target's Past Speed
Leads' Current Speed vs. Location
Future Speed vs. Time
Vehicle-to-Vehicle Data Processing
Polynomial Regression
Target's Future Speed Trajectory

Figure 1.4: Schematic of an anticipative car following with speed forecasting method in [2].

1.2 Challenges in Plug-In Hybrid Vehicle Control

1.2.1 Online Fuel-Efficient Energy Management Strategy with Trip Preview

A trip is defined as the distance between recharge opportunities and is long enough to exceed its pure EV range in the sequel. Starting with a fully charged battery, ideally, the battery is expected to be fully depleted upon reaching the next charging opportunity. As discussed in the previous section, CDCS is not the optimal energy management strategy for considered trips in most cases, compared to the offline optimized charge blending (OCBD) strategy. However, this potential for further improvement in fuel optimality with OCBD can hardly be achieved when the charge blending is implemented online because the trip is not known in advance. The optimal energy management of PHEVs for minimizing fuel consumption is closely related to determining the optimal trajectory of battery SOC, referred to as *battery SOC planning* [22]. This optimal SOC trajectory can be used to solve PHEVs' energy management problem in a receding horizon manner online with forecasted driving profiles [14]. For instance, the authors in [14] showed that approximately 6% fuel economy gain could be achieved by dynamic battery SOC planning using realtime traffic data, compared to a CDCS strategy.

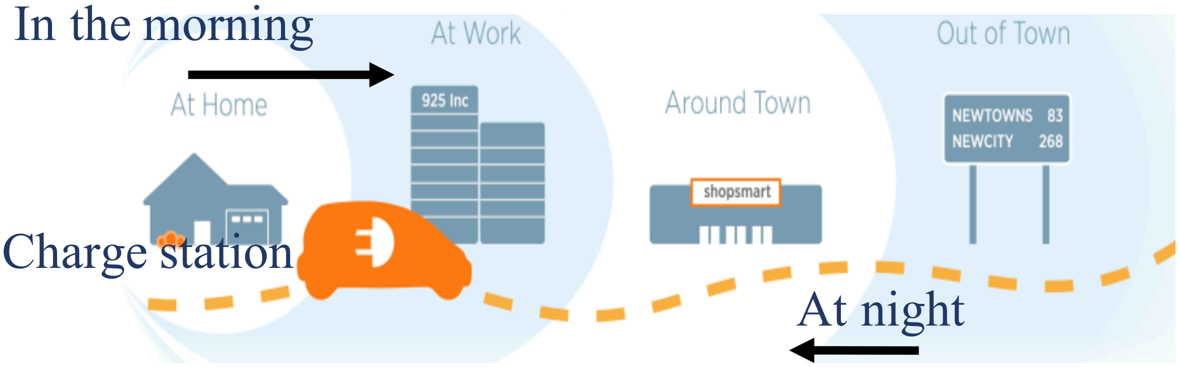


Figure 1.5: Schematic of a typical trip, adapted from https://www.chargepoint.com/.

From an optimal control perspective, effective SOC planning ideally requires comprehensive information about future driving conditions. Unfortunately, although large-scale traffic monitoring systems offer plenty of high-resolution details to facilitate the SOC trajectory planning over the entire trip, their accessibility is still limited. A more realistic scenario uses the realtime rough traffic information available from mobile mapping services for SOC planning. Although much less detailed on route segments, various traffic information is available from these mobile mapping services, including average speeds, segment distances, estimated travel time on each segment, and color codes indicating congestion levels. However, exploiting crude traffic information for effective SOC planning and properly using SOC reference to achieve a fuel-efficient CB strategy remains challenging.

1.2.2 Engine Cranking and Chattering Behavior

One of the fundamental challenges in the optimal control of all hybrid electric vehicles, including PHEVs, involves managing the engine on/off switching systematically. Minimizing the cranking fuel during starting (engine on) addresses fuel economy and drivability degradation. However, the powertrain operation determined from either the Hamiltonian minimization as in PMP or the equivalent minimization strategy (ECMS) could jump arbitrarily (chattering behavior) between HV and EV mode depending on the co-state or equivalence factor, which results in a busy engine on/off. There are relatively fewer research efforts in optimizing the transient engine behaviors than the abundant work where only quasi-static engine maps are considered. In [23], a model predictive control (MPC) torque-split strategy is proposed, where the transient engine characteristics are considered through an augmented weighted cost related to the engine mode switch. However, the engine mode within the prediction horizon is determined heuristically. In [24], a control algorithm is proposed to regulate the transitions between different operating points by using the battery to smooth the engine transients. However, the problem essentially becomes a regularization problem since a hybrid electric vehicle (HEV) is considered, where the SOC range is small. It is not applicable in the PHEV application because the desired SOC reference requires solving the fuel minimization problem first, where the busy engine cranking problem persists. In [25], an additional integer-valued engine on/off state is considered, with engine cranking cost augmented in the cost function. However, its numerical strategy is simplified by the constant co-state assumption, and the solution is obtained by sequentially applying DP and convex optimization. It is hardly applicable to the PHEV application, where the constant co-state assumption is not valid due to the large SOC span.

1.2.3 Combined Vehicle Dynamics and Power-Split Optimization with Connectivity and Automation

The combination of the VD with the PT control in a co-optimization framework for PHEVs with a long trip[1], in the presence of a lead vehicle, and a specified terminal battery SOC level is challenging. Most of the existing work seeks to design the combination of PT-level and VD-level control to be in a layered structure. The predominant roadblock in performing the co-optimization originates from the difficulty in its numerical implementation. Concretely, the complexity of the PHEV VD and PT co-optimization arises from the following four aspects.

1. *The Mixed-Integer nature*: The engine on/off decisions need to be made due to the need to coordinate two power sources (fuel and electricity), making the problem a mixed-integer nonlinear optimal control problem.

2. *Increased input and control dimensions*: Some of the existing work has used single shooting to solve the (P)HEV energy management numerically based on continuous-time PMP. However, its success is difficult to generalize because the battery SOC is the only state considered, and its dynamics are slow. When the vehicle level dynamics and powertrain level dynamics are considered simultaneously to achieve minimum fuel consumption, the total number of states and controls increases, making the PMP-based single shooting fragile [26].

3. *Unstable vehicle-level dynamics*: Not only does the number of states and controls increase, but the dynamics of the vehicle-following subsystem as a double-integrator

[1] Longer than the battery range.

are also unstable. As shown in [26], the unstable dynamics, combined with the state constraints, pose further challenges in applying single shooting.

4. *Long horizon problem*: Co-optimizing the velocity and its powertrain operation of a PHEV to achieve minimum fuel consumption has to be demonstrated in a long horizon problem with a distance longer than the battery range. A reasonable problem horizon for the considered PHEV is generally more prominent than one-hour[2]. Consequently, even with an approximated Newton-type method [27], the problem dimension would become prohibitive for executing the iterations.

1.2.4 Numerical Solution of Mixed-Integer Nonlinear Optimal Control Problems

The challenges in effective PHEV control revolve around solving mixed-integer nonlinear optimal control problems numerically from the previous discussions.

- *The optimization problems considered in this thesis are not compatible with any of the existing solvers for mixed-integer problems (MIP).*

  The fuel rate as the stage cost in the objective function of the direct minimum fuel consumption problem consists of complicated lookup tables as a function of the engine torque and speed. The system dynamics, when converted into equality constraints, are nonlinear and involve many static maps. The nonlinearity in the state cost and system dynamics does not fit the mixed-integer linear program (MILP) or mixed-integer quadratic program (MIQP) solvable by available mixed-integer commercial solvers, like CPLEX and Gurobi. Besides, these commercial solvers usually use branch and cut/bound algorithms, which grow exponentially with the problem horizon.

- *A direct and simultaneous method is not suitable for the optimization problems considered in this thesis.*

  Essentially, the fuel-efficient control of PHEVs is formulated as an optimal control or trajectory optimization problem, which needs to be solved numerically through suitable numerical methods. Widely used numerical methods include direct shooting (converting the trajectory optimization problem into a nonlinear program (NLP)), indirect (single) shooting, and multiple shooting [28]. A major difference between direct and indirect shooting occurs in the definition of the control functions. For indirect (single) shooting,

[2] For example, Prius Prime has an EPA-rated all-electric range of 25 miles. It would take around one hour to deplete the battery and enter the charge-sustaining mode in a city driving condition.

the control is defined at each point in time by the maximum principle[29], whereas the control (possibly parameterized) in direct shooting is determined directly. Both the direct and indirect shooting methods suffer from a common difficulty. The essential shortcoming of these methods is that small changes introduced early in the trajectory can propagate into very nonlinear changes at the end of the trajectory. The fundamental idea of multiple shooting is to break the trajectory into shorter pieces or segments. The multiple shooting concept can be incorporated into either a direct or indirect method. The schematic comparison between single and multiple shooting is illustrated in Fig. 1.6.

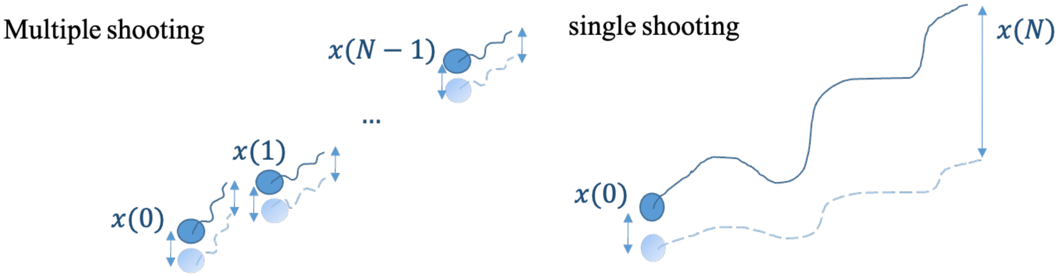


Figure 1.6: Illustration of single shooting and multiple shooting.

Based on my observation, it is less practical for PHEVs to adopt the direct and simultaneous approach. First of all, the fuel rate and efficiencies of motor and generator are static maps that are hard to be approximated accurately and efficiently by analytical expressions. As a result, the evaluation and approximation of the Hessian matrix required by multiple shooting would introduce significant numerical errors and be very computationally intensive. Second, with multiple shooting, the system dynamics are converted to equality constraints. Therefore in between multiple shooting iterations, the control sequence could be inconsistent with the state sequence, violating primal feasibility. It is especially notable in the PHEV application due to static maps. The numerical errors with the Newton-type method would make the system's dynamic equations very hard to satisfy with acceptable accuracy. Finally, a reasonable problem horizon for PHEVs is generally larger than one hour to necessitate the use of the engine and make the power-split optimization meaningful. Consequently, even with the approximated Newton-type method, the dimension of the resulting nonlinear program would become prohibitive for executing the iterations.

- *Fragile single shooting algorithm beyond single slow SOC state.*

With only power split optimization, the SOC (state) constraints can be assumed to be inactive, which facilitates the PMP-based analysis in continuous time. By comparison, in the presence of traffic, the vehicle following behavior necessitates the explicit consideration of pure state constraints (velocity and position). Notably, the position constraints are usually time-varying. The necessary condition for optimality with pure state constraints is less intuitive theoretically. Moreover, the jump condition associated with pure state constraints is hard for numerical algorithms. The SOC dynamics in the power split optimization are slow and stable. The benign dynamics enable the relatively easy numerical implementation of single shooting. The similarity between solving the two-point boundary value problem (TPBVP) resulting from PMP with single shooting and the existing ECMS indicates the potential for realtime implementation of single shooting in the PHEV application. However, as well be detailed later in this paper, the dynamics of the vehicle following system are fast and unstable. The unstable dynamics also pose a significant challenge to single shooting.

1.3 Organization and Contributions

This dissertation focuses on improving the total fuel economy of a PHEV over a trip by improving the current production energy management strategy and then the longitudinal velocity control and energy management strategy simultaneously in the traffic flow. The considered trip is long enough that the battery alone cannot provide the required total energy. Therefore, fully charged initially, the battery should be depleted to a desired low level at the end of the trip to reduce the fuel consumption by exploiting the electricity.

Chapters 2 and 3 focus on the powertrain level control of a human-driven PHEV. Chapter 2 considers the SOC planning with connectivity. Specifically, an effective method is proposed to determine the optimal sequence of SOC values at nodes between each segment of a route using segmented sparse traffic information. These SOC node values are used as boundary conditions for low-level vehicle energy management with more accurate traffic information available locally. Finally, it leads to the P-ECMS proposed in Chapter 3.

Combined vehicle-following and powertrain level control are considered starting from Chapter 4. Chapter 4 investigates a sequential, decentralized optimization approach, with velocity smoothing under the traffic constraints followed by the power-split optimization. Afterward, the centralized fuel minimization problem is considered. Finally, an iterative and hierarchical numerical strategy is proposed to resolve difficulties associated with the direct fuel minimization, combining the gradient projection with the single shooting.

This thesis focuses on solving the PHEV optimal control problems using single shooting-based numerical strategies. The co-state dynamics are typically unstable forward-in-time regardless of the stability of the state dynamics, thus making the continuous-time PMP-based single shooting fragile. By comparison, when the state dynamics are stable, the co-state dynamics become stable backward-in-time when formulated directly in discrete time with Karush-Kuhn-Tucker (KKT). When optimal control problems are formulated directly in discrete time, even when the state dynamics are initially unstable, they can be pre-stabilized, thus inducing stable co-state dynamics. As a result, formulating the optimal control problems directly in discrete time has good potential to avoid unstable state and co-state dynamics. Chapter 5 considers solving mixed-integer nonlinear optimal control problem (OCP) formulated directly in discrete-time. A modified Discrete Maximum Principle (DMP) is proposed to obtain first-order necessary conditions for optimality and a two-point boundary value problem (TPBVP) to be solved. A Discrete Mixed-Integer Shooting (DMIS) algorithm is then presented for solving the associated TPBVP numerically. The algorithm is first benchmarked against state-of-art solvers on several eco-driving problems. It is then demonstrated successfully in solving (1) the energy management of a PHEV to achieve minimum trip fuel consumption in consideration of fuel cranking and (2) the fuel consumption minimization of an autonomous PHEV by co-optimizing its velocity profile and powertrain operation.

Chapter 6 expands the discussions on the connection between the DMIS and approximate policy iteration (approximate-PI) in the reinforcement learning (RL) application, integrating the numerical optimization strategy with the online implementable framework. It then presents the design of a control framework to unify the PHEV fuel-efficient control from the PT-only control of a human-driven to the combined VD and PT control of an automated PHEV. In its application to automated PHEVs, a control barrier function (CBF) is augmented as an add-on block to modify the vehicle level control input to be executed whenever necessary. The unified control framework guarantees safe vehicle-following behavior when including longitudinal vehicle dynamics control.

It is critical to understand the fuel economy potential and the drivability with each optimization scheme with all the different optimization formulations. Chapter 7 seeks to systematically evaluate the fuel economy and drivability performance of different levels and structures of optimization strategies both offline and online. In terms of the levels of optimization strategies, both the PT level control for a human driver and the combined VD and PT level control for an autonomous driver are evaluated. In terms of the optimization structures for combined VD and PT level controls, both the centralized optimization structure, where the VD and PT are optimized together under one *single* objective function, and a decentralized (sequential) optimization structure, where the VD and PT are optimized separately with two different objective functions are of interest.

The main contributions of this thesis are summarized as follows:

- *Battery SOC node planning using connectivity*

  It is known from the offline simulation results that when the required trip energy is greater than the energy stored in a battery, an energy management strategy with SOC properly blended could lead to a better fuel economy compared to the CDCS strategy current in production. However, an effective online energy management strategy typically requires a reference SOC trajectory to guide the charge-blending to realize this potential benefit. Battery SOC planning is crucial for minimizing the fuel consumption of PHEVs. The optimal SOC trajectory depends on the traffic information or the speed profile of the entire trip. In chapter 2, crude traffic information about the velocity profile, similar to the type of information available in mobile navigation applications, is considered for SOC planning. Specifically, an effective method of determining the optimal sequence of SOC values at nodes between each segment of a route is proposed using segmented sparse traffic information. The SOC planning serves as the foundation of the online energy management detailed in chapter 3.

  - [30] D. Chen, Y. Kim and A. G. Stefanopoulou, ”State of Charge Node Planning with Segmented Traffic Information,” *2018 Annual American Control Conference (ACC)*, 2018, pp. 4969-4974, doi: 10.23919/ACC.2018.8431103.

- *Online predictive equivalent consumption minimization strategy (P-ECMS)*

  The reference SOC obtained with the SOC node planning algorithm is used as boundary conditions or a terminal condition for vehicle energy management with locally more accurate traffic information. The proposed P-ECMS adjusts its co-state or equivalence factor based on the difference between the future SOC obtained from short-horizon prediction and a future reference SOC. The benefits of the proposed P-ECMS are evaluated through vehicle simulations on a specified trip, compared against an A-ECMS with the same control parameters minus prediction. The P-ECMS is presented in chapter 3.

  - [31] D. Chen, Y. Kim and A. G. Stefanopoulou, ”Predictive Equivalent Consumption Minimization Strategy With Segmented Traffic Information,” in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14377-14390, Dec. 2020, doi: 10.1109/TVT.2020.3034552.

- *Eco-driving and power-split optimization in a sequential manner*

  With increase driving automation and connectivity, it is possible to control the vehicle’s longitudinal velocity to improve fuel efficiency further. The potential of fuel efficiency improvement of a PHEV is investigated by the sequential smoothing of a velocity profile given traffic constraints and the optimization of its charge depletion strategy. This study is presented in the first part of chapter 4 and investigates the additional potential in fuel economy compared to a pure powertrain level optimization.

  - [32] D. Chen, N. Prakash, A. Stefanopoulou, M. Huang, Y. Kim, and S. Hotz, "Sequential Optimization of Velocity and Charge Depletion in a Plug-in Hybrid Electric Vehicle.", *14th International Symposium on Advanced Vehicle Control, 2018*

- *A Comparative Study of H_∞ and Model Predictive Control Approach to Eco-Driving*

  Most of the existing work on eco-driving focuses on the design of model predictive control (MPC) controllers, an optimization-based control strategy. In the online MPC implementation, both the computation complexity and the prediction accuracy will affect the performance of the optimization-based controller. A simple but effective approach to designing an ecological adaptive cruise controller based on the H_∞ technique is investigated. The controller is designed to minimize the H_∞ norm of the transfer function from the internal states and the lead vehicle's velocity (disturbance) to the ego vehicle's acceleration to prompt driving smoothness. It implicitly handles position constraints via the minimization of the H_∞ norm of the transfer function from the lead vehicle's velocity as a disturbance to the time-headway tracking (performance). The proposed controller is beneficial as it is prediction and optimization-free, thus is lightweight for online implementation. The performance of the proposed controller is compared with an MPC-based car-following approach, precisely, acceleration minimization, in consideration of electrical and hybrid powertrain operations. This study is presented in Appendix B.

  - D. Chen, Y. Kim, "A Comparative Study of H_∞ and Model Predictive Control Approach to Eco-Driving.", *accepted to Modeling, Estimation and Control Conference (MECC 2021)*

- *An iterative and hierarchical approach to co-optimizing the velocity and powersplit of a PHEV to achieve direct fuel minimization*

  Via velocity smoothing in a sequential optimization approach, the power demand at the wheel is reduced. However, the engine is controlled to work in its high-efficiency region to minimize fuel consumption, typically high power. The engine would provide excessive power to charge the battery. This secondary energy conversion would end up damaging system-wise efficiency. Therefore, it is of interest to understand the optimality gap between the decentralized and a centralized approach, where the fuel minimization functions are the sole objective for the two subsystems. However, simultaneously co-optimizing the two subsystems brings up significant numerical difficulties. An iterative and hierarchical numerical strategy is proposed to resolve these difficulties. Single shooting is used to deal with the engine on/off decisions in the power-split optimization, and the gradient projection is used to deal with the unstable dynamics and the state constraints. The work in this study is presented in the last part of chapter 4.

  - [26] D. Chen, Y. Kim, M. Huang and A. Stefanopoulou, ”An Iterative and Hierarchical Approach to Co-optimizing the Velocity Profile and Power-split of Plug-in Hybrid Electric Vehicles,” *2020 American Control Conference (ACC)*, 2020, pp. 3059-3064, doi: 10.23919/ACC45564.2020.9147804.

- *Discrete-Time Mixed-Integer Shooting (DMIS) Algorithm*

  One of the PHEV optimal control challenges is the difficulties in obtaining the numerical solution of mixed-integer nonlinear optimal control problems. The work in this part considers solving mixed-integer nonlinear OCP rising from the control of PHEV powertrains. The OCPs are formulated in discrete-time, where a discrete-time maximum principle is proposed to obtain first-order necessary conditions for optimality and a TPBVP to be solved. Then a DMIS algorithm is presented for solving the associated TPBVP numerically. The DMIS is first benchmarked against a state-of-art FBRS-based quadratic program (QP) solver and a mixed-integer solver on two quadratic optimization problems with continuous and discrete control inputs. The DMIS algorithm is demonstrated successfully in addressing the energy management of a PHEV to achieve minimum trip fuel consumption. In particular, nonsmooth state and control transition costs are incorporated in the cost function to reduce frequent engine on/off behaviors. The DMIS algorithm is presented in the second half of chapter 5.

  - D. Chen, M. Huang, A. Stefanopoulou, “Discrete Mixed-Integer Shooting (DMIS): Algorithm and Application to PHEV Energy Management Accounting for Fuel Cranking”, *submitted to IEEE Transactions on Control Systems Technology, 2020*

- *Offline numerical solution of co-optimization with DMIS, and the analysis of the underlying physical reason for the fuel economy benefit with co-optimization.*

  The applicability of a DMIS algorithm is demonstrated in obtaining numerical solutions to the fuel minimization of an autonomous PHEV by co-optimizing its velocity profile and powertrain operation. The nonlinear minimum fuel consumption problem is formulated in discrete time, has both continuous and discrete decision variables, and is subject to nonlinear dynamics and time-varying state constraints. First, a TPBVP to be solved is obtained with a discrete-time maximum principle. Then a DMIS algorithm is applied to solve the associated TPBVP numerically. The underlying physical reason for the fuel economy benefit with co-optimization is also analyzed. The study is presented in chapter 5.

  - D. Chen, M. Huang, Y. Kim, A. Stefanopoulou, “Co-optimization of Velocity and Charge-Depletion for Plug-in Hybrid Electric Vehicles: Accounting for Acceleration and Jerk Constraints”, *submitted to the ASME Journal of Dynamic Systems, Measurement and Control*

- *Receding Horizon Co-Optimization Framework*

  The conflict between the long control horizon required for global optimality and computational power limits poses significant challenges to implementing the co-optimization online. A receding horizon strategy is presented with the co-states updated using a nominal trajectory and the temporal-difference to control the future cost, thus shortening the prediction horizon to 10-sec. The co-state update considers 1) the predicted SOC deviation from the nominal state trajectory, 2) the temporal difference (TD) deviation from the nominal co-state trajectory, and 3) the predicted SOC deviation from the desired terminal SOC. Besides, to resolve potential safety-critical constraint violations due to insufficient single shooting iterations, the control action obtained from the MPC is modified as needed in considering a maximal control invariant set. These studies are not presented in this thesis but are included in the following papers.

  - [33] Chen, D., Huang, M., Stefanopoulou, A., Kim, Y. (2021). “A Receding-Horizon Framework for Co-Optimizing the Velocity and Power-Split of Automated Plug-In Hybrid Electric Vehicles.” *ASME Letters in Dynamic Systems and Control*, 1(4), 041006.
  - D. Chen, Y. Kim, E. Hyeon, “Receding-Horizon Safe Co-Optimization of the Velocity and Power-split of Plug-in Hybrid Electric Vehicles with Imperfect Prediction.” presented at ACC21

- *Online Fuel Efficient Control: a DMIS-Based Unified Receding-Horizon Fuel-Efficient Control Framework*

  In the last part of Chapter 6, a unified receding horizon framework is presented, from the PT-only control of a human-driven PHEV to the combined VD and PT control of an automated PHEV. In the unified framework, the cost-to-go (the fuel consumption as the economic cost) is represented by the co-state associated with the SOC dynamics. This co-state is corrected internally using a single shooting method and externally based on the co-state TD-error. In its application to automated PHEVs, a CBF is augmented as an add-on block to modify the vehicle level control input to be executed whenever necessary. The unified control framework guarantees safe vehicle-following behavior when including longitudinal vehicle dynamics control. It allows for systematically evaluating the fuel economy and drivability performance of different levels and structures of optimization strategies.

- D. Chen, N. Li, M. Huang, Y. Kim, A. Stefanopoulou, “Online Fuel Efficient Control of Plug-In Hybrid Electric Vehicles: A Unified Reinforcement Learning Inspired Safe Model Predictive Control Framework”, *in preparation for an IEEE journal*

CHAPTER 2

State-of-Charge Node Planning Using Connectivity

The work presented in this chapter has been published in:
[30] D. Chen, Y. Kim and A. G. Stefanopoulou, "State of Charge Node Planning with Segmented Traffic Information," *2018 Annual American Control Conference (ACC)*, 2018, pp. 4969-4974, doi: 10.23919/ACC.2018.8431103.

2.1 Introduction

The utilization of traffic information can benefit energy management strategies for electrified vehicles. The accuracy of a large-scale traffic monitoring system is typically limited; a more realistic or nearly-achievable scenario would be the utilization of crude traffic information from mobile services, e.g., Google Maps, WAZE, and HERE WeGo. They provide traffic information on route segments, including highway systems, route distance, estimated time of arrival (ETA), and traffic condition levels with a color code which are much less detailed. Motivated by the immediate availability and popularity of such data, this chapter proposes to use this *less accurate* information in SOC planning and compare the fuel consumption of this scheme versus the one derived with the full velocity profile. The idea of SOC node planning is inspired by [34], where the authors try to optimally control HEV by decomposing the route into a series connection of segments with known properties. Expected fuel consumption is calculated over the route as a function of set-points for SOC in each segment and vehicle speed trajectories in each route segment. Then, DP is used to determine the sequence of SOC setpoints for each route segment.

As shown in Fig. 2.1, the traffic information about segments of a given route is provided. Based on the available information, a simplified velocity profile of each segment is constructed and connected (solid green line on the bottom plot). Then, an optimal SOC trajectory for this simplified velocity profile over the route is (approximately) computed. Only the SOC node values at the beginning and end of each segment are extracted (green diamonds on the top plot) to obtain a sequence of desired SOC node values $\text{SOC}_d(i)$, $i = 1, 2, ...n$. When the vehicle is driven on the first segment, the energy management with more accurate traffic information and disturbances may lead the actual SOC at the end of the first segment to a *slightly* different value (SOC_a, the blue star on the top plot). Assuming new traffic information is obtained at the end of the first segment for the remaining trip (black dashed line on the bottom plot), then new optimal SOC node values are recomputed from a newly simplified velocity profile for the remaining trip (black diamonds on the top plot).

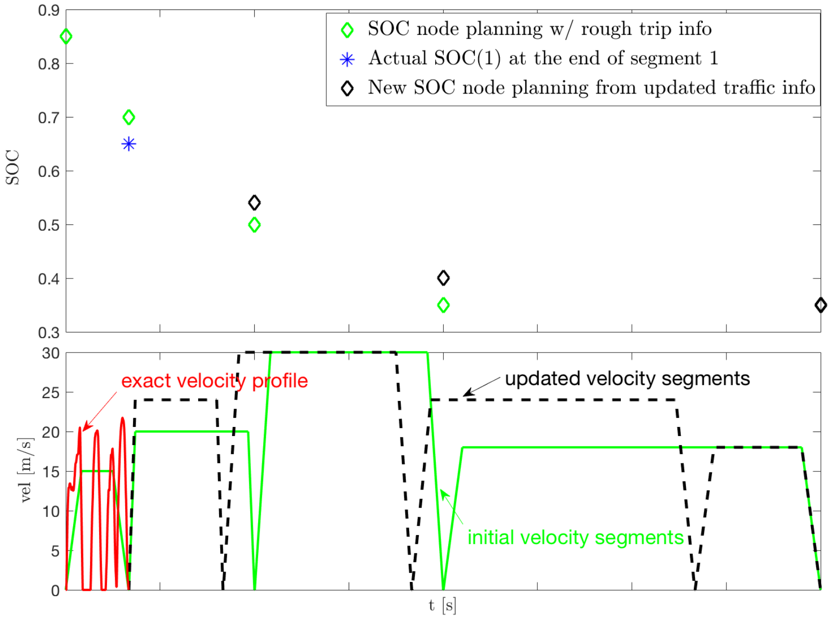


Figure 2.1: The proposed approach for SOC planning by approximately computing optimal SOC values at the end of velocity segments based on segmented trip information.

How these velocity segments are established and how the SOC nodes are obtained from the segmented velocity profile are detailed in the next section.

2.2 State-of-Charge Node Planning with Segmented Traffic Information

2.2.1 Traffic Simplification

One of the critical steps in SOC node planning is the simplification of the driving cycle from limited traffic information. This simplified velocity is constructed by connecting acceleration, cruising, and deceleration to match the segment distance, ETA, and traffic flow.

The authors in [34] assumed the availability of a predicted nominal vehicle speed trajectory for each route segment. The route segmentation criteria relate to substantial changes in either the average road grade or average vehicle speed but did not specify how the nominal trajectories are obtained. By comparison, the work presented in this section shows how a trapezoidal nominal vehicle speed trajectory could be constructed from statistical features.

Concretely, in this chapter, we assume we know: The segment trip length l; the estimated time spent on the segment T; the average acceleration/deceleration[1] a ($a > 0$ is the absolute value, and we assume the same magnitude for acceleration and deceleration); and finally, the entry and exit speed at this segment v_0 and v_f (can be considered as 0 if there is no available information).

There are 4 unknowns: acceleration time Δt_a, deceleration time Δt_d, cruising time Δt_c, and the cruising speed v_c. Then, velocity simplification can be converted into finding the above 4 unknowns subject to the following 4 equality constraints:

$$\begin{aligned} v_0 + a \cdot \Delta t_a - (v_f + a \cdot \Delta t_d) &= 0 \\ v_0 \cdot \Delta t_a + \frac{1}{2} a \cdot \Delta t_a^2 + v_f \cdot \Delta t_d + \frac{1}{2} a \cdot \Delta t_d^2 + v_c \cdot \Delta t_c &= l \\ v_c - (v_0 + a \cdot \Delta t_a) &= 0 \\ \Delta t_a + \Delta t_d + \Delta t_c &= T. \end{aligned} \tag{2.1}$$

Solving (2.1) gives us the 4 unknown values, which can be utilized to construct the piecewise linear velocity profile v (consist of trapezoidal velocity segments). Such a simplified, segmented velocity profile v will be used for SOC planning by considering a minimum fuel consumption problem with v as the external disturbance. This will be the topic of Section 2.2.3.

2.2.2 Examples and Discussions

Before introducing the minimum fuel consumption problem for SOC planning, the concept of traffic simplification through segmentation is first illustrated on some standard driving cycles.

Figure 2.2 shows two urban driving scenarios with different traffic levels and one highway driving scenario. They are constructed by combining various federal driving cycles such as UDDS, HWFET, NYCC, and LA92, for illustration purposes. Here the segments were manually selected. As is indicated from (2.1), apart from the segment trip length l and estimated time T

[1] It is shown in [35] that the traffic data on the road segments in urban conditions is consistent regardless of drivers. Consequently, the assumption that the typical acceleration value can be analyzed and extracted from historical data is reasonable.

on the segment, the acceleration a chosen to represent traffic also plays an important role in prediction. Given the same segment trip length, a larger acceleration and a longer estimated time on a segment usually indicate a more congested traffic condition because of more frequent start-stop behaviors. On the contrary, a smaller acceleration with a shorter estimated time on a segment indicates a smoother driving behavior and a less congested traffic level. The red curves in the third row of Fig. 2.2 show the actual segment trip energy, and the black curves are the corresponding predicted trip energy with the simplified velocity profile in the first row.

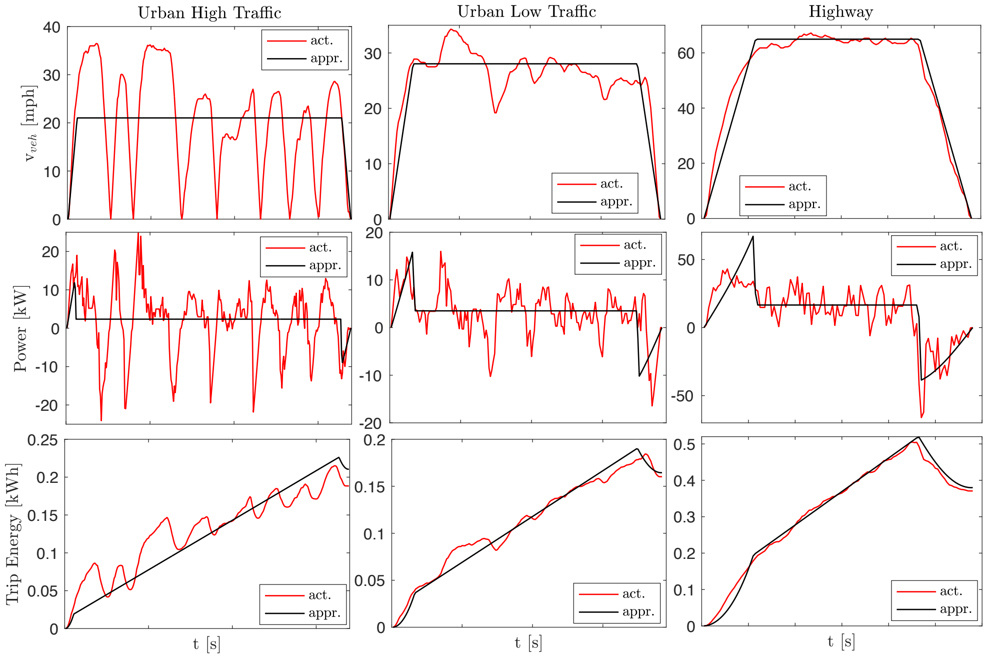


Figure 2.2: Examples of segmentation, from top to bottom: velocity, power demand and trip energy

The simplification of detailed velocity profile as stylized accelerating-cruising-decelerating creates a mismatch in the demanded power, as shown in the 2nd row of Fig. 2.2. This is especially true in high power demand, low congestion, and entry to high-speed trip segments. The mismatch can lead to SOC planning that is sub-optimal when compared to the optimal SOC trajectory with the actual detailed power demand the vehicle requires in reality. Some attempts have been made to study the traffic pattern in velocity and acceleration thresholding and distribution and establish criteria to classify different traffic conditions. However, those previous attempts lack a systematic process and rigorous justification. Therefore, future work will focus on the investigation of better classification of the traffic segments.

2.2.3 Approximation of the Minimum Fuel Consumption Problem

As discussed at the end of Section 2.2.1, the simplified and segmented velocity profile v will be used for SOC (node) planning. The SOC (node) planning is performed through solving an optimal control problem.

The optimal control problem for SOC planning is formulated in continuous-time to minimize the cost function J quantifying the total fuel consumption over a finite time horizon $[0, t_f]$, with t_f typically the total ETA:

$$\text{minimize } J(u, v, t) = \int_0^{t_f} \dot{m}_f(u)dt \tag{2.2a}$$

$$\text{subject to } \dot{x} = f(x, u, v) \quad \text{corresponding to (A.14)} \tag{2.2b}$$

$$x(0) = x_0, \quad x(t_f) = x_f \quad \text{initial and final SOC values} \tag{2.2c}$$

$$c(u, x, t) \leq 0 \quad \text{corresponding to (A.26).} \tag{2.2d}$$

In (2.2), the state $x =$ SOC, the control inputs $u = [ne\, te]^T$ include the engine speed and torque, and v as the speed of the considered PHEV is considered as known external disturbance[2]. The static fuel rate $\dot{m}_f(u)$, depending only on the engine operation point, is the considered metric in the stage cost.

Assumption 2-$\mathcal{A}$. The entire battery range is allowable for the considered optimal control problem (2.2), and the battery SOC lies always within the constraints, i.e., (A.26a) is inactive for any feasible input and known external disturbance trajectories $u[0, t_f], v[0, t_f]$, respectively. Meanwhile, constraints (A.26b)-(A.26h) do not depend on SOC.

Assumption 2-$\mathcal{A}$ indicates the absence of the pure state inequality constraints (as SOC is the only considered state in (4.3)). As a result, (2.2d) becomes $c(u, t) \leq 0$.

Remark 1. Note that the torque and speed limits of the motor MG2 and generator MG1 are all converted to the constraints on the engine, with the invariant constraints Ω on the engine torque and speed in HV mode in this section defined by a polytope of the form $[te, ne] \in \Omega := \{[te, ne] | H \begin{bmatrix} te \\ ne \end{bmatrix} \leq h\}$. The original admissible set of the engine is $[te, ne] \in \tilde{\Omega}(v) := \{[te, ne] | H \begin{bmatrix} te \\ ne \end{bmatrix} \leq \tilde{h}(v)\}$, where $h = \tilde{h}(v)$ except for only a single entry of $\tilde{h}$ depends on

[2] Although the purpose of this section is for SOC node planning with piese-wise linear, segmented velocity profiles, in the problem formulation (2.2), v can be any valid vehicle's velocity trajectory. As a result, (2.2) can refer to SOC planning problem with segmented traffic information or the offline power-split optimization problem on a particular trip v.

v. This section (also this thesis) considers the maximal Ω, such that $\Omega \subset \tilde{\Omega}(v), \forall v$. This is a reasonable simplification since the tightened constraint is an upper bound on engine speed that is rarely visited. As a result, (2.2d) becomes $c(u) \leq 0$, which is invariant.

The Hamiltonian for this problem has the form:

$$H(x, u, p) = \dot{m}_f(u) + p \cdot \dot{\text{SOC}}, \tag{2.3}$$

where p is the co-state associated with the SOC. Assumption 2-$\mathcal{A}$ simplifies the constraints to pure inequality control constraints, where no discontinuity in co-state p occurs [29]. The co-state p satisfies the adjoint equation

$$\dot{p} = -\frac{\partial H}{\partial x} = -\left(\frac{\partial H}{\partial V_{oc}}\frac{\partial V_{oc}}{\partial \text{SOC}} + \frac{\partial H}{\partial R_{batt}}\frac{\partial R_{batt}}{\partial \text{SOC}}\right). \tag{2.4}$$

As the considered powertrain has two different operating modes, namely, HV mode and EV mode[3], the Hamiltonian defined in (2.3) is split into the following:

$$H = \begin{cases} H_{HV} = \dot{m}_f + p\dot{\text{SOC}} & u = [ne\ te] \in \Omega \\ H_{EV} = p\dot{\text{SOC}} & u = [0, 0] \end{cases} \tag{2.5}$$

where H_{HV} and H_{EV} correspond to HV and EV modes, respectively. According to the PMP [29], at each time instant, the control is chosen as

$$u^* = \underset{u \in \Omega \cup [0,0]}{\arg\min}\{H\} = \arg\min\{H_{HV}, H_{EV}\}, \text{ where} \tag{2.6a}$$

$$\min\{H\} = \min\{H_{HV}, H_{EV}\}. \tag{2.6b}$$

Single shooting is used to solve the TPBVP in (2.2b), (2.2c), (2.4), (2.6a), where the initial guess of the co-state $p(0)$ is adjusted iteratively until the final SOC(t_f) is close to SOC$_f$ with desired accuracy.

2.3 Simulation Results and Discussions

This section presents and discusses simulation results to evaluate the performance of the proposed SOC node planning in a realistic scenario with *less accurate* traffic information[4]. For

[3] To be consistent throughout the thesis development, the Dual Motor Drive (DM) mode is omitted.

[4] The battery used in simulation in the comparison is half the size of the battery size used in other chapters of the thesis.

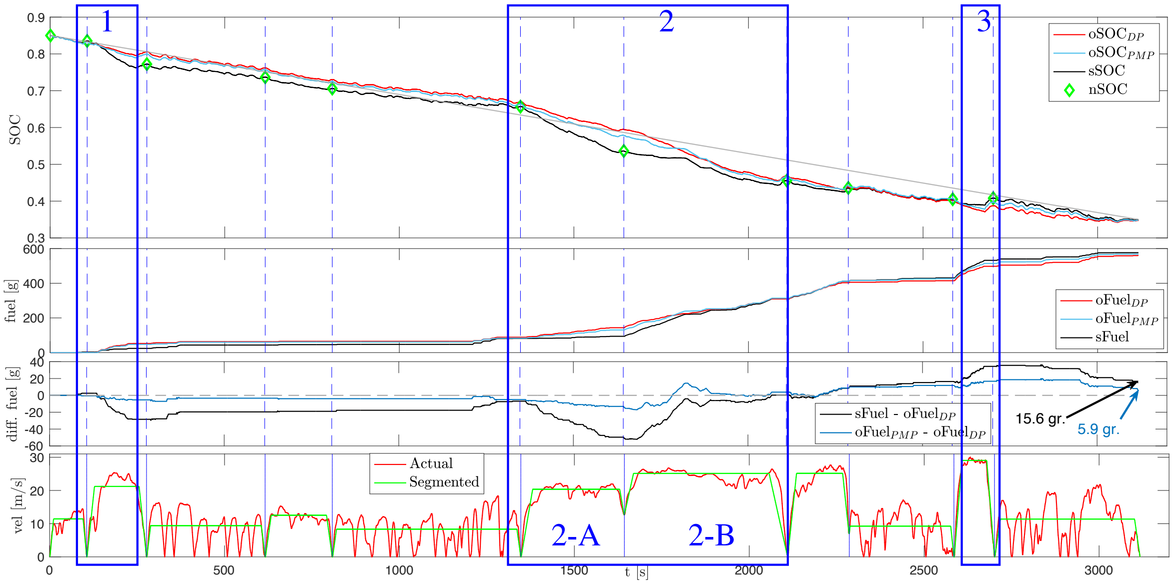


Figure 2.3: Performance of SOC node planning via segmented PMP in comparison with PMP and DP with a full knowledge of the trip: SOC, cumulative fuel, fuel difference, and vehicle velocity.

the SOC node planning to be meaningful, a trip is chosen such that the energy requirement over the trip exceeds its maximum available battery energy. SOC$_f$ is set to a low level assuming the availability of a recharging opportunity at the end of the trip.

The following procedures are established to simulate the real-world scenario and to evaluate the performance of the proposed method:

1. Before the trip, the segmented rough traffic information is provided by a traffic advisory or other mobile services. The segmented velocity profile is derived based on Section 2.2.1.

2. Afterwards, the TPBVP (2.2b), (2.2c), (2.4) based on PMP (abbreviated as PMP in the remainder of this section) is formulated with the simplified velocity profile and solved numerically with single shooting. The resulting SOC node sequence SOC$_d(i)$ at the beginning and end of each segment is extracted and stored.

3. Next, the sequence SOC$_d(i)$ is used as the boundary condition for each segment. Assuming that on each segment, exact knowledge of the velocity profile is available, a similar TPBVP for each short segment can be solved numerically.

4. Finally, DP and PMP are applied on the entire offline trip to evaluate 1) the offline performance of PMP with respect to DP as well as 2) the effectiveness of SOC$_d(i)$ as additional boundary conditions for short-horizon optimal control. The performance is

evaluated based on their total fuel efficiency as well as segment fuel efficiency. Note that the performance of the proposed method is evaluated against DP results instead of CDCS strategy because the commercial CDCS strategy typically contains complicated logic, not publicly available.

2.3.1 A Case Study: A Combined Federal Drive Cycle

The performance of the proposed SOC node planning is evaluated through a case study. A one-hour-long trip is designed by stitching US federal drive cycles, including HWFET, UDDS, LA92, and NYCC. Each trip contains different driving behaviors: low-speed urban traffic, medium-speed urban traffic, high-speed urban traffic, ramp driving, and highway driving. The first subplot in Fig. 2.3 shows the SOC trajectories with corresponding control algorithms:

- oSOC$_{DP}$: SOC trajectory computed by DP[1] with the entire trip information. This is the offline optimization on the detailed velocity profile for the particular trip in Fig. 2.3 and takes approximately 5 hours.

- oSOC$_{PMP}$: SOC trajectory computed by PMP with the entire trip information. It takes approximate 150s.

- nSOC: The desired SOC node sequence computed by PMP on the segmented simplified velocity profile. Only SOC node values in between segments are extracted, and it takes approximately 40s.

- sSOC: SOC trajectory computed by PMP with nSOCs as boundary conditions and exact velocity profiles on segments.

The second subplot in Fig. 2.3 shows cumulative fuel consumption from the three approaches, namely, oFuel$_{DP}$, oFuel$_{PMP}$ and sFuel which are 561.8, 567.7 and 576.4 grams, respectively. oFuel$_{PMP}$ is slightly higher than oFuel$_{DP}$ due to the use of a constant co-state instead of the co-state dynamics (2.4) in simulation[5]. It should be highlighted that the SOC node planning with the segmented velocity information leads to only 1% additional fuel consumption. To check the robustness of the proposed algorithm to changes in traffic patterns, two other scenarios with long highway driving duration in the beginning and at the end of the entire trip are also simulated and compared to their corresponding DP results. In both scenarios,

[1] A solution to the DP problem is numerically obtained by using *dpm* function[36]. The number of grid sizes of the states and inputs is sufficiently large so that the obtained solution can be considered as a *global optimum*.

[5] Note that as pointed out before, the battery considered in this section is relatively small. It is shown in [30] that the co-state can be approximated by a constant value with the considered battery.

the proposed algorithm with the segmented velocity increases the fuel consumption by 2.5% additional.

As can be seen from the first and third subplots in Fig. 2.3, three segments are highlighted as ①, ② and ③, where the SOC trajectory and fuel consumption from sPMP deviate considerably from the optimal solutions by PMP and DP. It can be observed that the mismatch happens primarily on high-speed operations. Specifically, the three apparent locations where mismatch happens can be further divided into two cases: (1) the mispredicted $\text{SOC}_d(i)$ is at the end of the high speed segment and causes an apparent fuel consumption difference when compared to offline DP solution – ① and ③ in Fig. 2.3; (2) mispredicted $\text{SOC}_d(i)$ happens in the middle of several high speed segments, and another high speed segment counteracts and helps recover segment fuel consumption difference when compared to DP – ② in Fig. 2.3.

2.4 Performance of Control Obtained Using Necessary Condition for Optimality

So far in this chapter, the control sequence is obtained by only considering the first-order necessary conditions for optimality under the form of PMP. Although PMP is well known to furnish necessary conditions for the optimality of the control of a dynamic system [37], these conditions are not, in general, sufficient for optimality [38]. In [39, 40], comparative analyses were done in terms of using DP and PMP to solve the resulting HEV minimum fuel consumption problem. It was observed that DP and PMP could achieve very similar results. To justify the utilization of only the first-order necessary conditions for the PHEV optimization problem in this chapter and throughout this thesis, the optimality of the resulting control sequence is worth investigating. The section seeks to justify the adoption of the PMP from two aspects

1. Since the power-split optimization problem of a human-driven PHEV (2.2) with an actual drive cycle involves only a single state (SOC) and two control inputs, it is possible to use DP to calculate the solution numerically. The DP results with reasonable discretization can be considered as *global optimal* solutions. By comparing the solution obtained with PMP[6] and DP, the optimality of PMP can be judged empirically [7].

[6] Here PMP refers to the single shooting-type of numerical strategy to solve the TPBVP obtained by considering the first-order necessary condition in the form of PMP.

[7] Note that the DP results are only available for the optimization of a human-driven PHEV. For the co-optimization problem considered in later chapters, the DP solution is not available due to the well-known "curse of dimensionality".

2. An additional justification is provided in this section from the aspect of sufficient condition for optimality.

2.4.1 Comparison to Dynamic Programming in Simulation

First, the performance of PMP and DP is compared on 4 trips. The trips are standard driving cycles combined with different orders and slightly scaled speed. The time-domain trajectories are compared in Fig. 2.4 and Fig. 2.5, and the results are also summarized in Table 2.1a and Table 2.1b. The battery used in simulation in the comparison is half the size of the battery size used in other chapters of the thesis to reduce the problem horizon and thus the total computation time required by DP.

The size of the discretization slightly influences the DP results. Nevertheless, the PMP and DP results are sufficiently close, as evident by the resulting total fuel consumption, indicating good performance (optimality) of the control sequence obtained by applying only necessary conditions for optimality.

Table 2.1: Comparison of the numerical solutions between DP and PMP.

(a) Simulation Results on 1.3HWFET+UDDS with different orders.

	1.3HWFET + UDDS	UDDS+ 1.3HWFET
SOC$_f$ (PMP)	0.3540	0.3593
SOC$_f$ (DP)	0.3474	0.3548
m_{fuel} [g] (PMP)	976.5832	977.9353
m_{fuel} [g] (DP)	977.4278	979.6739

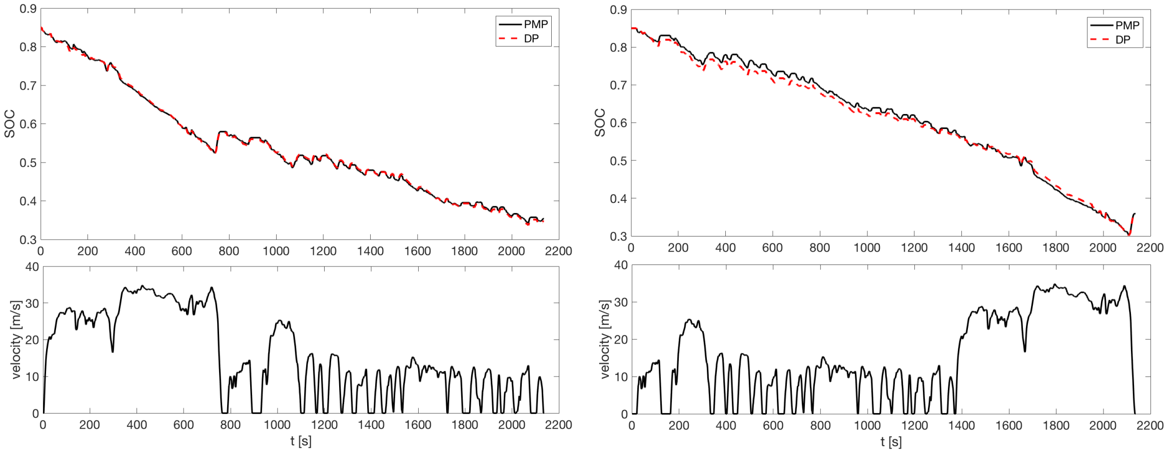
(b) Simulation Results on 2*1.3HWFET+UDDS with different orders.

	2*1.3HWFET + UDDS	UDDS+ 2*1.3HWFET
SOC$_f$ (PMP)	0.3554	0.3382
SOC$_f$ (DP)	0.3548	0.3474
m_{fuel} [g] (PMP)	1.8014e+03	1.7963e+03
m_{fuel} [g] (DP)	1.8052e+03	1.8030e+03

2.4.2 Sufficient Condition for Optimality

In the previous section, sufficient optimality of the control sequence (numerical solution) obtained by applying the necessary condition for optimality in the form of PMP is demonstrated through comparing with its corresponding DP solution. In this section, the empirical observation is strengthened by showing the solution satisfies the sufficient condition for optimality [38].

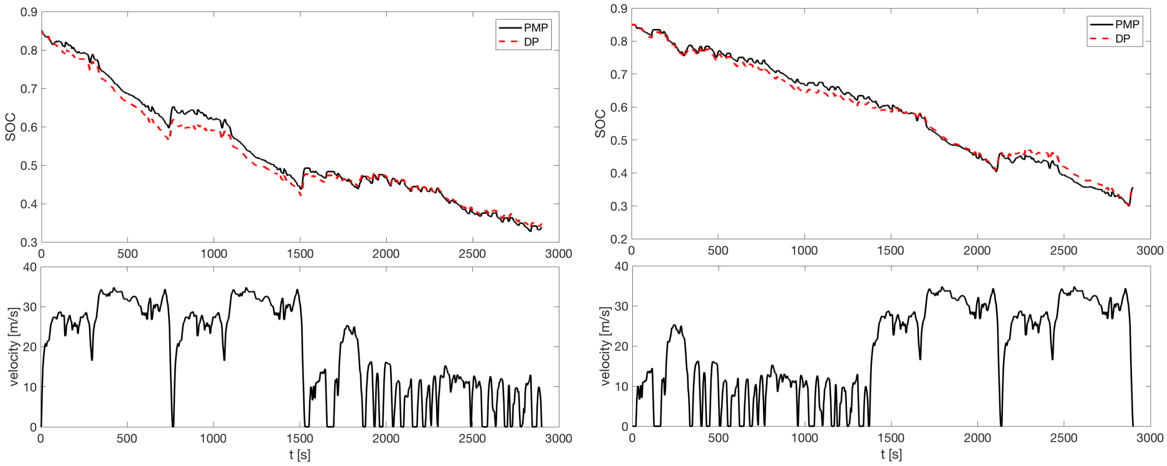
Under the assumption of the absence of SOC constraints and slightly tightened control constraints such that the admissible set for the control input becomes invariant, it is possible



(a) 1.3HWFET+UDDS

(b) UDDS+1.3HWFET

Figure 2.4: Comparison between PMP and DP (Table 2.1a).



(a) 2*1.3HWFET+UDDS

(b) UDDS+2*1.3HWFET

Figure 2.5: Comparison between PMP and DP (Table 2.1b).

to show the optimality of the resulting solution follow the line of development of sufficient condition for optimality as in [38][8].

Find a piecewise continuous[9] control vector $u \in \mathbb{R}^m$ and associated continuous and

[8] [38] discussed maximization. In the context of the minimization problem considered in this chapter, the concavity conditions are converted to convexity conditions correspondingly.

[9] The term "piecewise continuous" is used in the sense that at the points of discontinuity the one-sided limits always exist.

piecewise differentiable state vector $x \in \mathbb{R}^n$, defined on the fixed time interval $[t_0, t_f]$, that will

$$\text{minimize } \int_{t_0}^{t_f} l(x,u,t)dt, \tag{2.7a}$$

$$\text{subject to the vector differential equation } \dot{x} = f(x,u,t) \tag{2.7b}$$

$$\text{initial condition } x(t_0) = x_0 \tag{2.7c}$$

$$\text{terminal conditions } \begin{cases} x^i(t_f) = x_f^i & i = 1, \cdots, l \\ x^i(t_f) \geq x_f^i & i = l+1, \cdots, m(x_f^i, i = 1, \cdots, m \text{ fixed numbers}) \\ x^i(t_f) \text{ free} & i = m+1, \cdots, n \end{cases} \tag{2.7d}$$

$$\text{and control variable restriction } u(t) \in \mathcal{U}, \quad \mathcal{U} \text{ a given set in } \mathbb{R}^m. \tag{2.7e}$$

It is assumed that f is continuous differentiable w.r.t. x. A pair $(x(t), u(t))$ is called *admissible* if it satisfies (2.7b)-(2.7e). An admissible pair that minimizes the integral in (2.7a), and thus solves the given problem, is called an *optimal pair*. For convenience, the well known PMP (first-order necessary condition for optimality) is first stated.

Theorem 2.4.1. [38] In order that $(\bar{x}(t), \bar{u}(t))$ be an optimal pair for (2.7), it is necessary that there exist a constant p_0 and a continuous function $p(t) \in \mathbb{R}^n$, where for all $t \in [t_0, t_f]$, $(p_0, p(t)) \neq (0, 0)$, and such that

$$H(\bar{x}(t), u, p(t), t) \geq H(\bar{x}(t), \bar{u}(t), p(t), t). \quad \forall u \in \mathcal{U}, \tag{2.8a}$$

where the Hamiltonian function H is defined by

$$H(x, u, p, t) = p_0 \cdot l(x, u, t) + p^T f(x, u, t), \tag{2.8b}$$

except at the points of discontinuity of $\bar{u}(t)$,

$$\dot{p}(t) = -H_x(\bar{x}(t), \bar{u}(t), p(t), t). \tag{2.8c}$$

Furthermore,

$$p_0 = 1, \text{ or } p_0 = 1, \tag{2.8d}$$

and finally the following transversality conditions are satisfied

$$\begin{cases} p_i(t_f) \text{ no conditions} & i = 1, \cdots, l \\ p_i(t_f) \geq 0 \quad (= 0 \text{ if } \bar{x}^i(t_f) > x^i_f) & i = 1 + 1, \cdots, m \ . \\ p_i(t_f) = 0 & i = m + 1, \cdots, n \end{cases} \tag{2.8e}$$

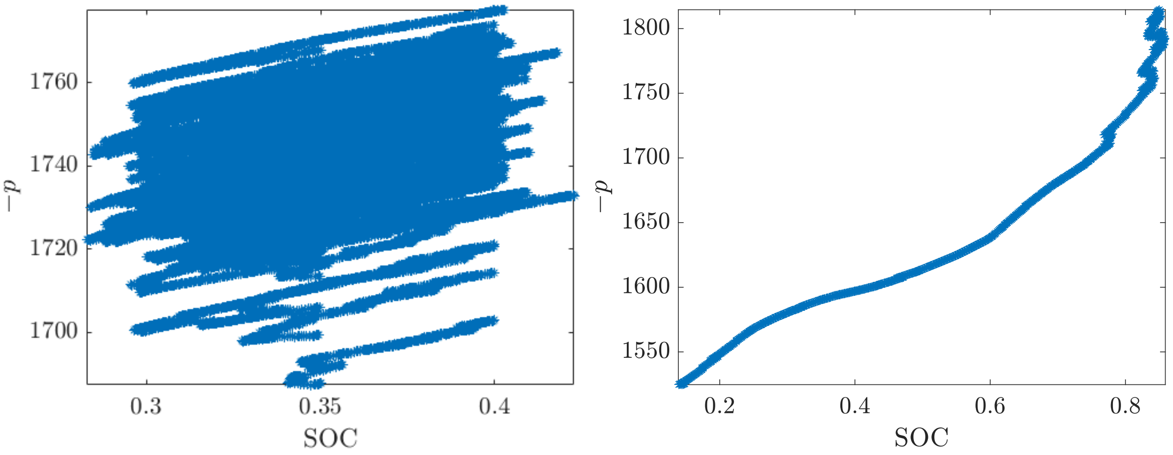
Since for the particular PHEV minimum fuel consumption problem (2.2) the admissible set for the control input(engine torque te and speed ne) consists of two disjoint sets $[te, ne] \in [0, 0] \cup \Omega := \{[te, ne] | H \begin{bmatrix} te \\ ne \end{bmatrix} \leq h\}$, a slightly generalized sufficient condition is considered. To this end, let us define for (2.7)

$$H^*(x, p, t) = \min_{u \in \mathcal{U}} H(x, u, p, t), \tag{2.9}$$

where $H(x, u, p, t)$ is as defined in (2.8a), and it is assumed that the minimum is attained.

Theorem 2.4.2. [38] Suppose $(\bar{x}(t), \bar{u}(t))$ is an admissible pair satisfying all the conditions with $p_0 = 1$. If $H^*(x, p(t), t)$ defined in (2.9) is convex in x, then $(\bar{x}(t), \bar{u}(t))$ is a solution to (2.7).

The proof of Theorem 2.4.2 is given in [38] and is therefore omitted here. For the particular PHEV minimum fuel consumption problem (2.2), $p_0 = 1$ is considered. It remains to show the convexity of the Hamiltonian (2.3) in terms of $x = \text{SOC}$. Note that the co-state $p = -\frac{\partial H^*(\text{SOC}, p, t)}{\partial \text{SOC}} < 0, \forall t$, SOC based on the simulation results. As a result, the convexity of the Hamiltonian can be seen from the SOC–$(-p)$ plot.



(a) The state (SOC) and co-state on many similar short Ann Arbor trips.

(b) The state (SOC) and co-state on a long trip consisting of stitched standard driving cycles.

Figure 2.6: Illustration of SOC–$(-p)$ plot on some trips.

As seen from Fig. 2.6, the slope of SOC–$(-p)$ plot (which is $\frac{\partial^2 H^*(\text{SOC},p,t)}{\partial \text{SOC}^2}$) is approximately positive across the entire SOC span. It indicates the convexity requirement is approximately satisfied for the considered problem. As a result, the sufficient condition for optimality is shown to be approximately satisfied, justifying again sufficient optimality of the control obtained by considering only the necessary condition for optimality.

2.5 Summary

This chapter presents a practical approach to scheduling battery SOC for PHEVs when sparse traffic information is available over a given route. Simulation results show that, compared to DP results with a full trip knowledge, the short horizon optimal control with SOC node values obtained from the proposed method consumes only 2.5% additional fuel. Furthermore, the SOC node values for the trip could be computed in 40 seconds, and they follow reasonably well the values from DP except for particular segments characterized as high speed/ power demand.

The control sequence obtained in this chapter is obtained by only considering the first-order *necessary* conditions for optimality under the form of PMP. To justify the utilization of only the first-order *necessary* conditions in this chapter, this chapter additionally seeks to demonstrate the optimality of the resulting control sequence from two aspects. First, it is shown in simulation that first-order necessary condition-based numerical solution achieves similar fuel consumption results as those obtained with DP. Afterward, additional justifications are provided in view of the sufficient condition for optimality. The confirmation of sufficient optimality lays out the foundation for a single shooting type of numerical strategy based on the first-order necessary condition developed throughout this thesis.

CHAPTER 3

Online Power-Split Optimization

The work presented in this chapter has been published in
[31] D. Chen, Y. Kim and A. G. Stefanopoulou, "Predictive Equivalent Consumption Minimization Strategy With Segmented Traffic Information," in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14377-14390, Dec. 2020, doi: 10.1109/TVT.2020.3034552.

3.1 Introduction

The achievement of optimal energy management of PHEV for minimizing fuel consumption is closely related to determining the optimal trajectory of SOC, referred to as *battery SOC planning* [22]. This optimal SOC trajectory can also be used to solve the energy management problem of PHEVs in a receding horizon manner online with forecasted driving profiles [14]. For instance, the authors in [14] showed that approximately 6% fuel economy gain could be achieved by dynamic battery SOC planning using real-time traffic data, compared to a CDCS strategy.

In Chapter 2, a SOC node planning scheme is proposed using segmented traffic information provided by mobile services like Google Maps. Encouraged by the observed potential fuel economy benefit, a P-ECMS is proposed in this chapter for online PHEV energy management as an innovative extension of the ECMS that has already been well developed. Figure 3.1 illustrates the overall framework. First, a simplified velocity trajectory is constructed for a given trip, from which the demanded torque/power trajectory can be calculated. The current battery SOC provided by the battery management system is used for initialization, and the terminal SOC target is a level desired at the end of the trip. With the simplified velocity and power demand trajectories as external inputs, the minimum fuel consumption problem can be defined. By solving this optimization problem, the resulting SOC trajectory can be obtained, with only the SOC node sequence $\text{SOC}_d(i)$ at the beginning and end of each segment being extracted. In online implementation, the proposed P-ECMS utilizes the planned SOC node

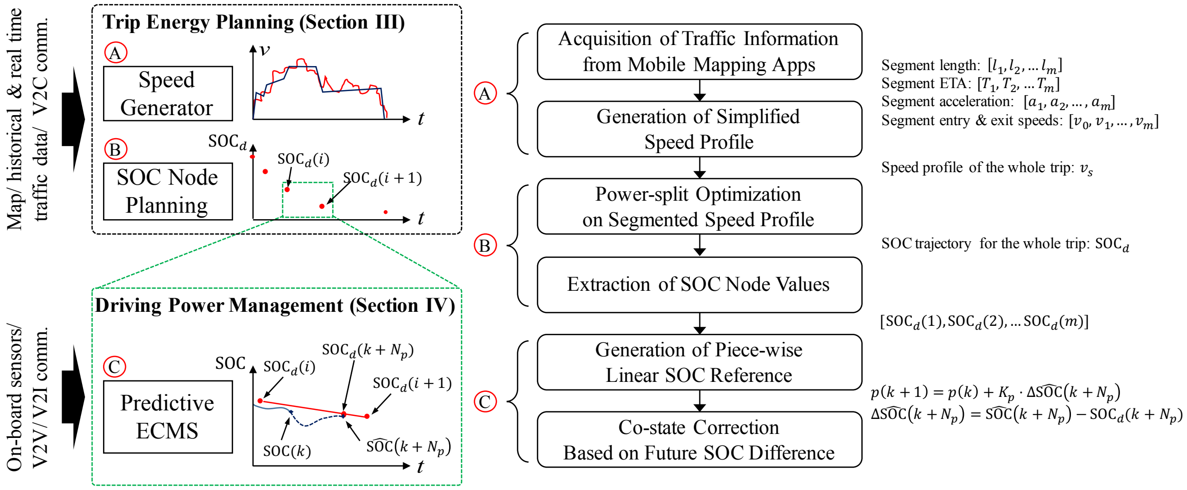

Map/ historical & real time traffic data/ V2C comm.
Trip Energy Planning (Section III)
A
Speed Generator
v
t
B
SOC Node Planning
SOC_d
$\mathrm{SOC}_d(i)$
$\mathrm{SOC}_d(i+1)$
t
On-board sensors/ V2V/ V2I comm.
Driving Power Management (Section IV)
C
Predictive ECMS
SOC
$\mathrm{SOC}_d(i)$
$\mathrm{SOC}_d(k+N_p)$
$\mathrm{SOC}_d(i+1)$
$\mathrm{SOC}(k)$
$\widehat{\mathrm{SOC}}(k+N_p)$
t
A
Acquisition of Traffic Information from Mobile Mapping Apps
Generation of Simplified Speed Profile
B
Power-split Optimization on Segmented Speed Profile
Extraction of SOC Node Values
C
Generation of Piece-wise Linear SOC Reference
Co-state Correction Based on Future SOC Difference
Segment length: $[l_1, l_2, \ldots l_m]$
Segment ETA: $[T_1, T_2, \ldots T_m]$
Segment acceleration: $[a_1, a_2, \ldots, a_m]$
Segment entry & exit speeds: $[v_0, v_1, \ldots, v_m]$
Speed profile of the whole trip: v_s
SOC trajectory for the whole trip: SOC_d
$[\mathrm{SOC}_d(1), \mathrm{SOC}_d(2), \ldots \mathrm{SOC}_d(m)]$
$p(k+1) = p(k) + K_p \cdot \Delta\widehat{\mathrm{SOC}}(k+N_p)$
$\Delta\widehat{\mathrm{SOC}}(k+N_p) = \widehat{\mathrm{SOC}}(k+N_p) - \mathrm{SOC}_d(k+N_p)$

Figure 3.1: A hierarchical approach to energy management via predictive ECMS with segmented traffic information.

values combined with a short-horizon velocity prediction to adjust the co-state.

The main contribution of this chapter is twofold: First, a P-ECMS is proposed and integrated into a framework that computes a reference SOC trajectory based on segmented traffic information. Second, simulations show the robustness[1] of the proposed P-ECMS to the initial co-state, prediction horizon length, the co-state update rate, and the accuracy of the predicted velocity. In particular, the performance of the proposed P-ECMS is compared with an existing A-ECMS algorithm with the same parameters except without prediction through simulations and, on average, is demonstrated to outperform the A-ECMS in terms of fuel economy and terminal SOC satisfaction.

3.2 Predictive Equivalent Consumption Minimization Strategy (P-ECMS)

This section briefly revisits the SOC node planning scheme with segmented traffic information proposed in Chapter 2. Afterward, a P-ECMS combined with the SOC node planning is proposed for online implementation.

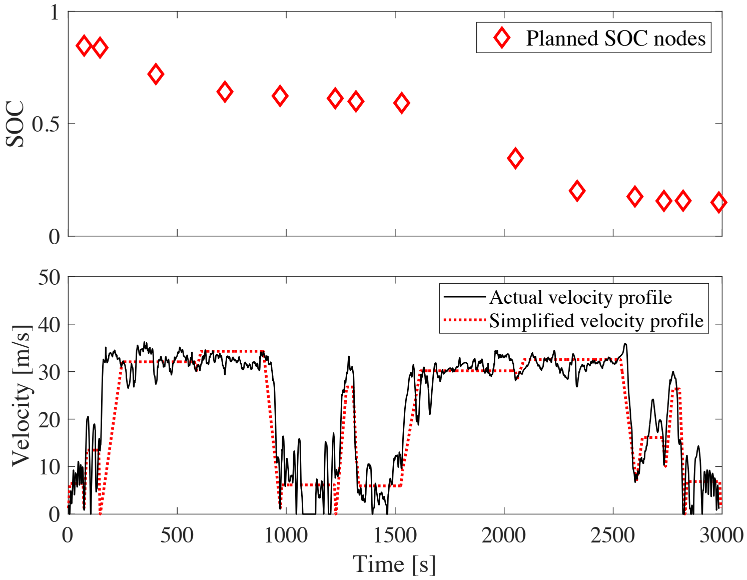


Figure 3.2: SOC planning by computing optimal SOC values at the ends of velocity segments, or nodes, based on segmented traffic information proposed in Chapter 2.

3.2.1 State-of-Charge Node Planning with Segmented Traffic Information

It is assumed that the rough traffic information available from mobile mapping apps (e.g., Google Maps) includes the trip segment length l, the estimated time of arrival (ETA) T for each segment, and color codes indicating the congestion level in each segment. Acceleration information is included in constructing the simplified velocity profile to indicate congestion levels. Then, the construction of a simplified velocity can be converted into solving a set of equations as in (2.1). The unknown values obtained by solving (2.1) will be utilized to construct the piece-wise linear velocity profile v_s (consist of trapezoidal velocity segments).

After constructing a simplified velocity profile, v_s with the segmented traffic information of the considered trip (illustrated as the red dashed curve in the bottom subplot of Fig. 3.2), the desired SOC trajectory SOC_d corresponding to this profile v_s can be obtained by solving the minimum fuel consumption problem over a finite time horizon $[0, t_f]$, with t_f typically the

[1] The term *robustness* refers to the fact that the total fuel consumption and terminal SOC are not heavily affected by the values of the control parameters.

total ETA of the entire trip:

$$\begin{aligned}
&\text{minimize } J = \int_0^{t_f} \dot{m}_f(ne, te)dt \\
&\text{subject to:} \\
&\qquad \dot{\text{SOC}}_d = f(\text{SOC}_d, ne, te, v_s, tp) \text{ detailed in (A.14)} \\
&\qquad \text{SOC}_d(0) = \text{SOC}_0, \quad \text{SOC}_d(t_f) = \text{SOC}_f \\
&\qquad (ne, te) \in \Omega \cap [0, 0],
\end{aligned} \tag{3.1}$$

where $\dot{m}_f$ is the engine fueling rate, which is a static map as a function of the engine torque te and speed ne. The driver's torque demand is represented by tp, and is defined as the function of the vehicle velocity v and acceleration a through a regression model that needs to be satisfied by the low-level powertrain components

$$tp = \alpha_0 + \alpha_1 \cdot a + \alpha_2 \cdot v_s + \alpha_3 \cdot v_s^2, \tag{3.2}$$

where $\alpha_i, i \in \{0, 1, 2, 3\}$ are the regression parameters. The engine speed ne and torque te can either be off (EV mode, where $(ne, te) = [0, 0]$) or subject to an invariant constraint Ω [2] in HV mode. In (3.1), the state $x = \text{SOC}$, the control inputs $u = [ne\ te]^T$ include the engine speed and torque, and v_s as the speed of the considered PHEV is considered as known external disturbance.

The continuous-time PMP is used to obtain the necessary conditions for optimality of (3.1) and the corresponding TPBVP. It is detailed in Chapter 2. The continuous-time TPBVP is then discretized to obtain the numerical solution. The Hamiltonian at $t = k\Delta t$ for this problem has the form:

$$H(k) = \dot{m}_f(ne(k), te(k)) + p(k) \cdot \dot{\text{SOC}}(k), \tag{3.3}$$

where $p(k)$ is the co-state at $t = k\Delta t$. For a given initial co-state $p(0)$, the discretized TPBVP can be converted to the following initial value problem (IVP): For $k = 1 : N_f$

$$u(k) = [ne(k), te(k)] = \underset{[0,0]\cup\Omega}{\arg\min}\, H(k) \tag{3.4a}$$

$$\text{SOC}_d(k+1) = \text{SOC}_d(k) + \Delta t \cdot f(\text{SOC}_d(k), u(k), \cdot) \tag{3.4b}$$

$$p(k+1) = p(k)\left(1 - \Delta t \frac{\partial f(\text{SOC}_d(k), u(k), \cdot)}{\partial \text{SOC}_d(k)}\right), \tag{3.4c}$$

[2] As discussed in Chapter 2, such an invariant set is obtained by assuming inactive SOC constraint and slightly tightening the speed-dependent constraint.

where $f(\text{SOC}_d(k), u(k), \cdot)$ abbreviates the battery SOC dynamics in (3.1).

To achieve the desired terminal SOC, SOC_f, single shooting is applied by iteratively adjusting the initial value of the co-state $p(0)$. The shooting iteration stops when the resulting terminal SOC value converges to the desired value (in this case, tolerance of 5e-4 is used). The SOC trajectory is obtained, and only the SOC node values at the beginning and end of each segment are collected to create a sequence of desired SOC node values $\text{SOC}_d(i)$, $i = 1, 2, ...m$ assuming there are m segments, as shown in the first subplot of Fig. 3.2. As will be detailed later, the desired SOC node sequence is used as the intermediate control waypoints in P-ECMS.

3.2.2 Predictive-ECMS with SOC Node Planning

In an A-ECMS, the co-state or equivalence factor[3] is adjusted based on the difference between current and reference SOC values with a proportional (P) or proportional-integral (PI) controller. It is, in practice, difficult to choose a control gain such that, at the end of the trip, the actual SOC of the system converges to the desired terminal SOC. This section proposes to resolve this challenge with a P-ECMS that updates the co-state (equivalence factor) with a detailed short-term prediction of the velocity profile based on the difference between the SOC at the end of the predicted short-horizon and its corresponding reference value.

Note that for HEVs, the battery acts mainly as an energy buffer. By comparison, the battery for a PHEV can be depleted from a high to a low SOC level throughout the trip. Consequently, to maximize the fuel economy benefit, it is crucial to guarantee that the actual SOC approaches the desired (low) level at the end of the trip.

3.2.2.1 Issues in A-ECMS with SOC Node Planning

The co-state in the A-ECMS is typically updated as

$$p(k+1) \ = p(k) + K_p\big(\text{SOC}(k) - \text{SOC}_d(k)\big), \tag{3.5}$$

where $\text{SOC}_d(k)$ is the reference SOC value at $t = k\Delta t$. $\text{SOC}_d(k)$ is obtained by the linear interpolation of the piecewise-linear SOC trajectory formed by connecting the selected SOC nodes. Note that a P-controller type A-ECMS with a fixed control gain K_p is considered to adjust the co-state. This control structure is similar to the P-ECMS, which allows us to understand the benefit of introducing predictions in ECMS more clearly.

When the co-state is updated based on (3.5), the performance of the A-ECMS for PHEV energy management strongly depends on the choice of the initial value of the co-state $p(0)$

[3] Note that as shown in [41], the equivalence factor is equivalent to the co-state under reasonable assumptions. In later work, the co-state is used interchangeably with the equivalence factor in ECMS.

and its update rate K_p. The drawbacks of the A-ECMS given in (3.5) are twofold based on observations and will be shown through simulations in Section 3.3.

1. The performance of the A-ECMS in terms of fuel economy and terminal SOC satisfaction varies with different $p(0)$ and K_p pairs. However, it is difficult to pick the "correct" tuning parameters for all trips encountered in practice. Observation has confirmed that with some combinations of $p(0)$ and K_p, the performance of A-ECMS degrades fuel economy significantly.

2. The feedback formulation (3.5) in the co-state cannot guarantee that the actual SOC at the end of the trip converges to the desired terminal SOC. If the terminal SOC is higher than expected, the fuel economy suffers due to the under-utilization of the battery.

The initial co-state $p(0)$ and its update rate K_p influence the convergence of the actual terminal SOC at the end of the trip. Moreover, they also influence how close the actual SOC tracks the reference SOC. The tracking closeness, in turn, influences the fuel economy. Figure 3.3 presents the trajectories of the battery SOC and cumulative fuel consumption results obtained from using four different strategies:

1. Offline Optimal: the optimal solution obtained by solving (3.1) with the actual speed profile (black-solid);

2. CDCS*: the modified CDCS strategy where the actual SOC at the end of the trip approaches the desired SOC value (red-solid) and the charge-sustaining part is optimized;

3. A-ECMS1: the A-ECMS whose $p(0)$ and K_p values are tuned so that the actual SOC at the end of the trip is close to the desired SOC value without close SOC reference tracking (green-solid);

4. A-ECMS2: the A-ECMS whose $p(0)$ and K_p values are tuned such that the actual SOC at the end of the trip is close to the desired SOC value and includes close SOC reference tracking (green-dashed).

The SOC reference trajectory used in A-ECMS is generated by the SOC planning discussed in the previous section. The simplified speed profile used in the exampled SOC planning and the actual speed profile are compared in Fig. 3.3. Note that to satisfy the terminal SOC constraint, K_p is adjusted through offline simulations [31]. As can be observed from the second subplot in Fig. 3.3 and Table 3.1, the fuel economy degrades if the resulting SOC trajectory tracks the reference SOC trajectory closely. Note that the actual final SOC at the end of the trip SOC(N_f) in A-ECMS2 is slightly higher than the desired value. That happens because, for this given $p(0)$, it is difficult to find a K_p that achieves the exact desired terminal SOC.

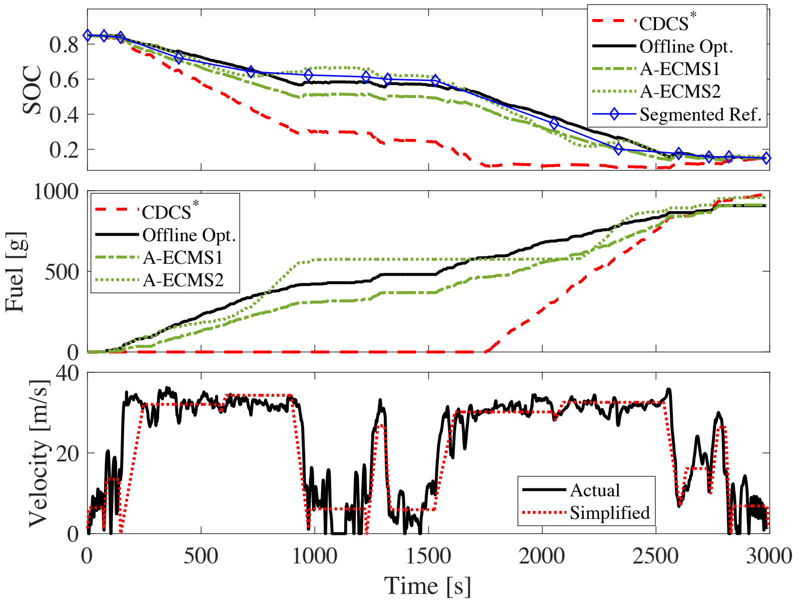


Figure 3.3: A-ECMS simulation results with different parameters over the considered drive cycle: (top) the battery SOC trajectories, (middle) the cumulative fuel consumption, and (bottom) the corresponding speed profile. Both of the A-ECMS use the SOC reference obtained with the segmented velocity profile.

Table 3.1: Final SOC and fuel consumption results from *offline* optimal solution, optimal CDCS, and A-ECMS with different parameters.

	$p(0)$	K_p	SOC(N_f)	Fuel [g]
A-ECMS1	-280	-0.08	0.15	911.1
A-ECMS2	-300	6.12	0.16	957.5
Offline Optimal Sol.			0.15	905.3
CDCS*			0.15	969.1

3.2.2.2 P-ECMS with SOC Node Planning

With SOC waypoints generated from the SOC node-planning in Section 3.2.1, the following P-ECMS is proposed for online implementation with detailed but not necessarily accurate short-term velocity prediction. The main difference between the A-ECMS and the P-ECMS is in utilizing SOC references constructed with rough traffic information. In A-ECMS, the co-state is updated with feedback from only the current SOC and with no feedback from the final SOC. By comparison, in P-ECMS, the co-state is updated with feedback in the short-horizon predicted future SOC. In details:

Step 0: Before driving on the road, construct a simplified segmented velocity profile (v_s) based on the segmented rough traffic information provided from mobile mapping apps (assume there are m segments)[4]. Then, given the initial battery SOC (SOC$_0$) and the desired SOC value at the end of the trip (SOC$_f$), solve (3.1) numerically with v_s. From the resulting SOC trajectory extract the desired SOC node sequence at the beginning and end of each segment (denoted as $\text{SOC}_d(0) = \text{SOC}_0, \text{SOC}_d(1), \ldots, \text{SOC}_d(m) = \text{SOC}_f$). Only these node values are stored in memory.

Step 1: Initialize the co-state value $p(0)$ from a predefined range $[p(0)^l, p(0)^u]$. As will be shown through simulations in Section 3.3, the proposed P-ECMS is usable to a wide range of $p(0)$.

If $k < N_f$, repeat:

Step 2: Given $p(k)$, from the instantaneous Hamiltonian minimization (3.3) the engine torque and speed command $te(k)$, $ne(k)$ can be determined. At the same time, simulate the powertrain dynamics forward to predict the SOC trajectory N_p-s ahead with a N_p-s predicted velocity trajectory. Within the prediction horizon, the engine torque and speed at each time instance is chosen by solving (3.3) with the co-state value fixed as $p(k)$. Denote the predicted SOC value at the end of prediction horizon as $\hat{\text{SOC}}(k + N_p)$.

Step 3: Correct $p(k + 1)$ as:

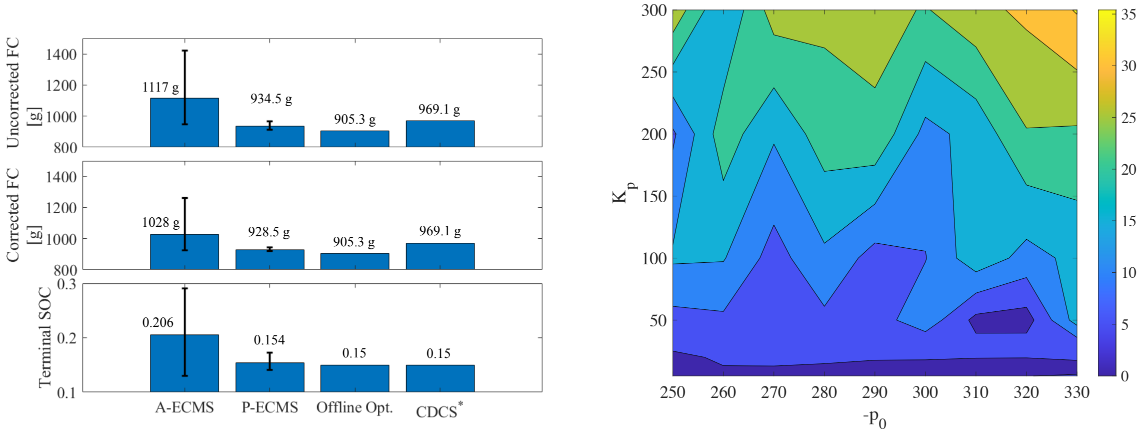
$$p(k+1) = p(k) + K_p\big(\hat{\text{SOC}}(k+N_p) - \text{SOC}_d(k+N_p)\big). \tag{3.6}$$

Here $\text{SOC}_d(k + N_p)$ is the linear-in-time interpolated value on the piecewise-linear SOC reference trajectory formulated by connecting the SOC node sequence. As will be shown through simulations in Section 3.3, the proposed P-ECMS is usable for a wide range of K_p.

3.3 Simulation Results and Discussions

In this section, the efficacy of the proposed P-ECMS is presented and discussed compared to alternative energy management strategies, including the A-ECMS, CDCS*, and an offline optimal solution with knowledge of the entire actual drive cycle. Especially featured is the performance comparison between the proposed P-ECMS (with prediction horizon $N_p = 30s$) and the A-ECMS with the same parameter setting but without prediction. Simulation results with different initial co-state $p(0)$ and co-state update rate K_p are shown in Fig. 3.4. Figure 3.4a demonstrates that stable and good sub-optimal fuel economy can be obtained by introducing prediction using the P-ECMS. By comparison, the performance of the A-ECMS has large variations (Fig. 3.4a) and degrades drastically compared to the P-ECMS beyond a narrow range

[4] Note that this work assumes we only have a short-horizon velocity prediction and do not have knowledge of a high-resolution velocity profile for the entire trip. Consequently, the simplified and stylized velocity profile is our best approximation of the entire trip.



(a) Summary of fuel consumption (FC) and actual terminal SOC comparison between A-ECMS and P-ECMS.

(b) Percentage difference in corrected fuel consumption between A-ECMS and P-ECMS. NOTE: a positive percentage number means that A-ECMS consumes more fuel than P-ECMS.

Figure 3.4: Performance comparison between P-ECMS with $N_p = 30$ seconds and A-ECMS with different $p(0)$ and K_p.

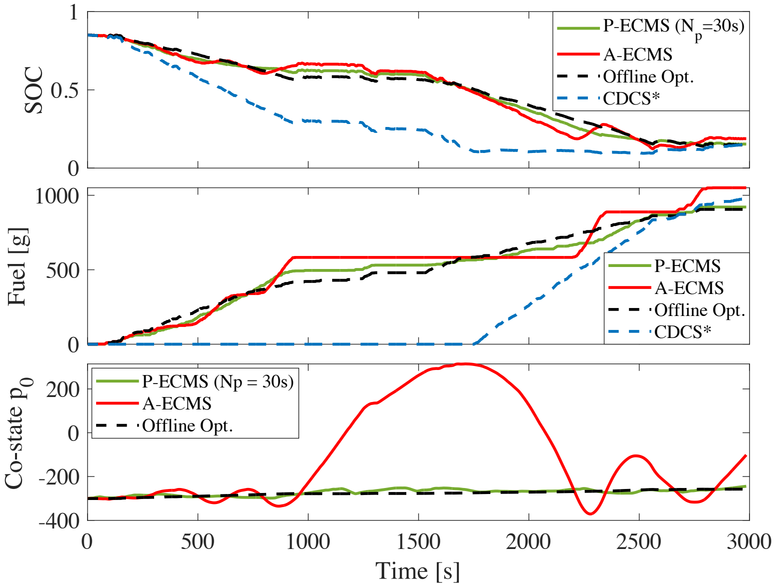


Figure 3.5: The time domain comparison between P-ECMS and A-ECMS with $p(0) = -300$ and $K_p = 30$

of parameters (Fig. 3.4b). Specifically, the P-ECMS can save 99.5 grams of fuel (corrected) on average, which corresponds to a 9.7% decrease in fuel consumption, compared to the

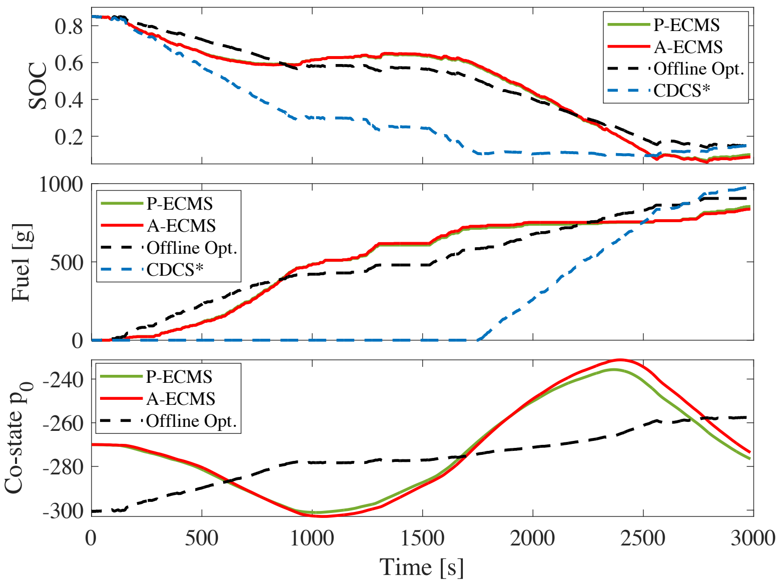


Figure 3.6: The time domain comparison between P-ECMS and A-ECMS with $p(0) = -280$ and $K_p = 1$

A-ECMS across different combinations of $p(0)$ and K_p[5]. It is also found that the averaged fuel consumption by the P-ECMS is 4.2% lower than that by CDCS*. Moreover, the standard deviation for fuel consumption with P-ECMS can be reduced by 96% from 84.1 to 3.4 grams compared to A-ECMS. When generated SOC nodes are used, the average performance with a linear-in-distance interpolation based on a predicted distance is similar to that with a linear-in-time interpolation. Complementary to Fig.4 (a), the average uncorrected fuel is 949.7 g. The average corrected fuel is 943.5 g, and the average terminal SOC is 0.154 when the generated SOC nodes are used applying a linear-in-distance interpolation with prediction distance.

Figures 3.5 and 3.6 present a comparison of the time-domain response among the P-ECMS, the A-ECMS with 2 different choices of $p(0)$ and K_p, the offline optimal and CDCS* results. In Fig. 3.5, the performance of the A-ECMS is significantly worse in terms of fuel economy compared to that of the P-ECMS, and worse still compared to CDCS*. Only in Fig. 3.6 is the performance of the A-ECMS similar to that of the P-ECMS.

Note that the parameters being considered do not influence the CDCS* and the optimal solutions. Consequently, they are marked as constant values in subsequent plots.

In the subsequent subsections, the robustness of the P-ECMS will be demonstrated in detail

[5] Note that this large performance improvement with P-ECMS upon A-ECMS does not conflict with Table 3.1. The result presented in Table 3.1 is obtained in an offline manner, where the K_p in (3.5) is adjusted iteratively on the entire trip such that the actual terminal with A-ECMS is close to the desired value[31]. By comparison, the result presented here is with an arbitrarily chosen K_p, where the actual terminal SOC with A-ECMS can deviate non-trivially from the desired value. As a result, the total fuel economy with A-ECMS degrades.

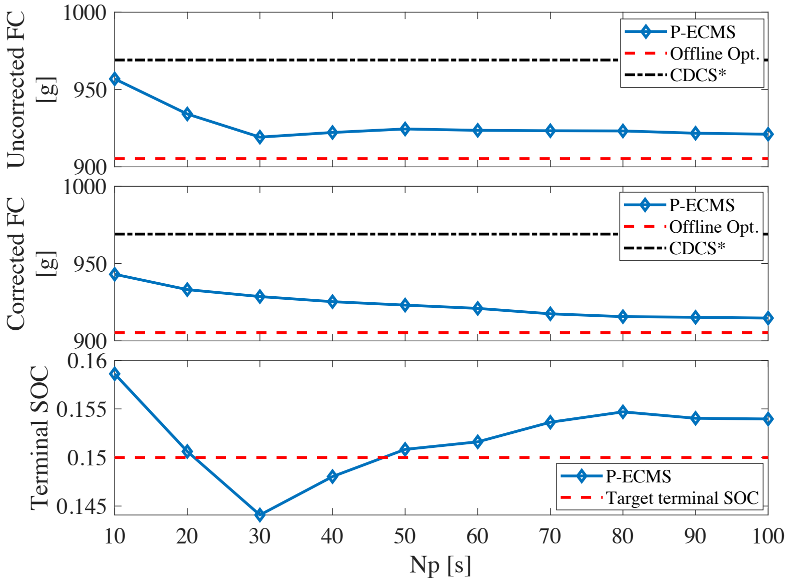


Figure 3.7: Influence of the prediction horizon N_p on the performance with $p(0) = -300$ and $K_p = 30$

from the following perspectives:

1. The predictive horizon N_p: P-ECMS is robust in prediction horizons. In general, a prediction horizon of 30 to 100 seconds demonstrates a good sub-optimal fuel economy compared to offline optimal results and considerable improvement upon CDCS based on our simulations. This is detailed in Section. 3.3.1.

2. The initial co-state $p(0)$: P-ECMS is robust even with inexact co-state initialization. In general, initial co-state $p(0) \in [-330, -250]$ [6] results in acceptable reduction in total fuel consumption. This is detailed in Section 3.3.2.

3. The co-state update rate K_p: P-ECMS is robust with various co-state update rates. In general, as long as K_p is not too small, we will have relatively good fuel economy. This is detailed in Section 3.3.3.

4. The accuracy of the velocity prediction: P-ECMS is robust with various levels of approximation of velocity trace. This is detailed in Section 3.3.4.

Besides, to show the effectiveness of the SOC node planning strategy discussed in Section 3.2.2, P-ECMS results obtained from three SOC planning approaches are presented. The results are discussed in Section 3.3.5 and highlight the importance of proper SOC planning.

[6] This range is obtained from offline simulations on various driving cycles.

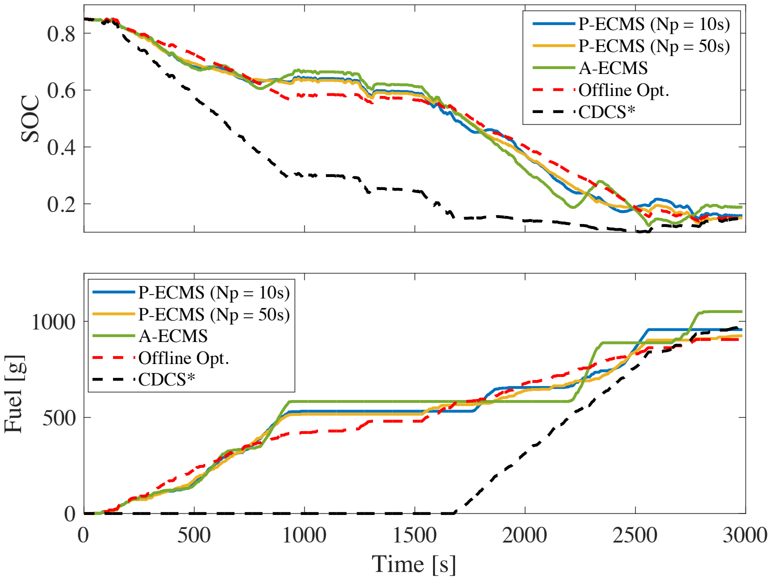


Figure 3.8: Time domain response of P-ECMS with different prediction horizons with $p(0)$ = -300 and K_p = 3

All the simulations have been performed on the driving cycle collected from real-world field testing, as shown in Fig. 3.2. Note that the SOC at the end of the trip is not guaranteed to be rigorously close to the desired SOC level in online implementation. For better comparison, we correct the total fuel consumption by considering the deviation of the terminal SOC from the desired terminal SOC and then present the fuel consumption results with (denoted as *corrected*) and without (denoted as *uncorrected*) SOC correction, which is computed by

$$m_{f,cor} = m_f - 1600(\text{SOC}_f - \text{SOC}(N_f)), \tag{3.7}$$

where m_f denotes the actual (uncorrected) total fuel consumption from simulation, and $m_{f,cor}$ denotes the corrected total fuel consumption based on terminal SOC difference. Since the power-split optimization of a PHEV is to maximize the utilization of the battery, any terminal SOC in the closed-loop simulation that is higher than the target value indicates energy left in the battery and thus represents an increase in total fuel consumption.

Baseline Definition: The purpose of the P-ECMS as a charge-blending optimization is to achieve better performance compared to CDCS, implemented on a production vehicle. Therefore, the CDCS is considered to be the baseline control strategy. However, in this work, CDCS* is used because 1) the terminal SOC can be guaranteed to approach the desired SOC through optimization, 2) the efficacy of the proposed P-ECMS can be shown even when compared to the "best" achievable CDCS, that is, the charge-sustaining part of the trip is solved separately from a fuel minimization problem (3.1) with an actual velocity profile for the charge-sustaining part.

3.3.1 Robustness with respect to the Prediction Horizon N_p

Figure 3.7 compares the results of the P-ECMS with different prediction horizons for the drive cycle in Fig. 3.2. The following parameters are used in this vehicle simulation: the co-state update rate $K_p = 30$ and the initial co-state $p(0) = -300$. Note that for purposes of comparison, the actual velocity trace (solid-black curve in Fig. 3.2) is applied for short-term prediction. In practice, it is still possible to obtain a reasonably accurate and detailed velocity trajectory for a short horizon, although the detailed velocity is not typically available for the entire trip. The work in [2] has shown that it is possible to predict a vehicle's velocity within a 10-second prediction horizon with DSRC information.

As can be seen from the first subplot in Fig. 3.7, the length of the prediction horizon does not significantly affect the uncorrected fuel consumption by the P-ECMS, especially when the prediction horizon N_p is longer than 30 seconds. It can also be observed that even with a short prediction horizon (e.g., $N_p = 10$ seconds), the closed-loop system's fuel consumption is still less than that of the CDCS* although the terminal SOC is much higher than the target terminal SOC value of 0.15. Meanwhile, as is shown in the second subplot in Fig. 3.7, as the prediction horizon increases, the corrected total fuel consumption decreases, which is not surprising. Moreover, as shown in the third subplot in Fig. 3.7, the actual terminal SOC is close to the target terminal SOC with the P-ECMS.

Figure 3.8 illustrates the corresponding time-domain response to Fig. 3.7. As seen in the first subplot, as the prediction horizon increases, the actual terminal SOC with the P-ECMS comes closer to the desired terminal SOC (SOC$_f = 0.15$), and the resulting total fuel consumption comes closer to the offline optimal fuel consumption (red-dashed) from the second subplot. By comparison, the A-ECMS (same parameter $p(0)$ and K_p without prediction) results in a much higher actual terminal SOC than the desired SOC, and total fuel consumption is even worse than CDCS*.

3.3.2 Robustness with respect to the Initial Co-State $p(0)$

Figures 3.9 and 3.10 show the influence of the initial co-state $p(0)$ on the performance of the P-ECMS and the A-ECMS with two different co-state update rate K_p values: in both cases, the prediction horizon is set to be $N_p = 30$-sec. Note that the co-state tuning rates in Fig. 3.9 and Fig. 3.10 are $K_p = 30$ and $K_p = 5$, respectively. The co-state is initialized with a value in the range from -330 to -250: $p(0) \in [-300, -250]$. As can be seen from these figures,

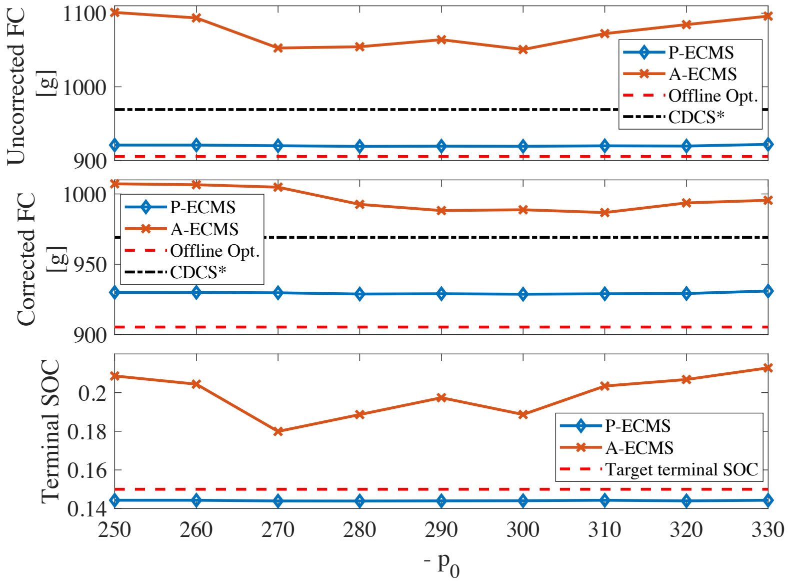


Figure 3.9: Influence of the initial co-state $p(0)$ on the performance of P-ECMS with $N_p = 30$ seconds and K_p = 30

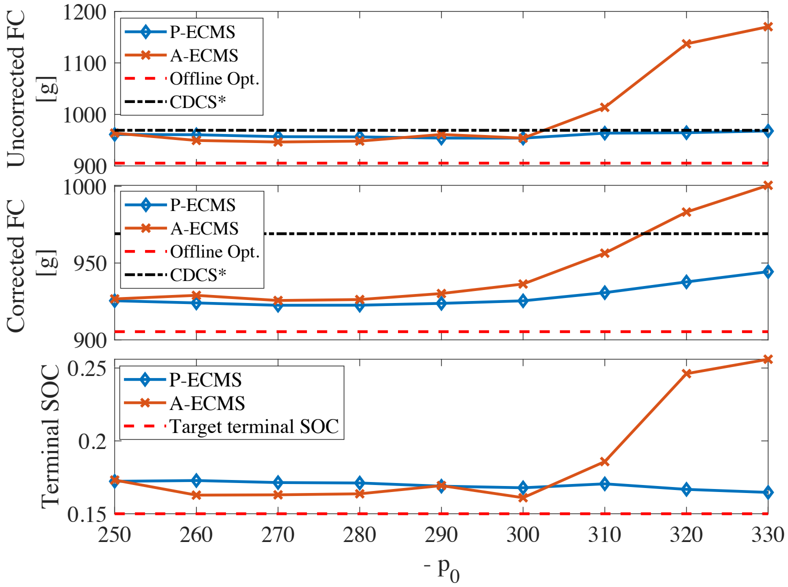


Figure 3.10: Influence of initial co-state on the performance of P-ECMS with $N_p = 30$ seconds and K_p = 5

with the P-ECMS, initializing the co-state within a certain range is sufficient to guarantee good sub-optimality. By comparison, the performance of the A-ECMS with the same parameters of K_p and $p(0)$ can vary significantly. As seen from Fig. 3.10, for the particular driving cycle considered, the A-ECMS works well with $K_p = 5$, meaning that the A-ECMS has the potential to achieve good fuel economy with the proper choice of K_p and $p(0)$. However, as shown in Fig. 3.9, the performance of the A-ECMS degrades drastically with $K_p = 30$. These simulation results clearly show the robustness of the P-ECMS with respect to the initial co-state $p(0)$.

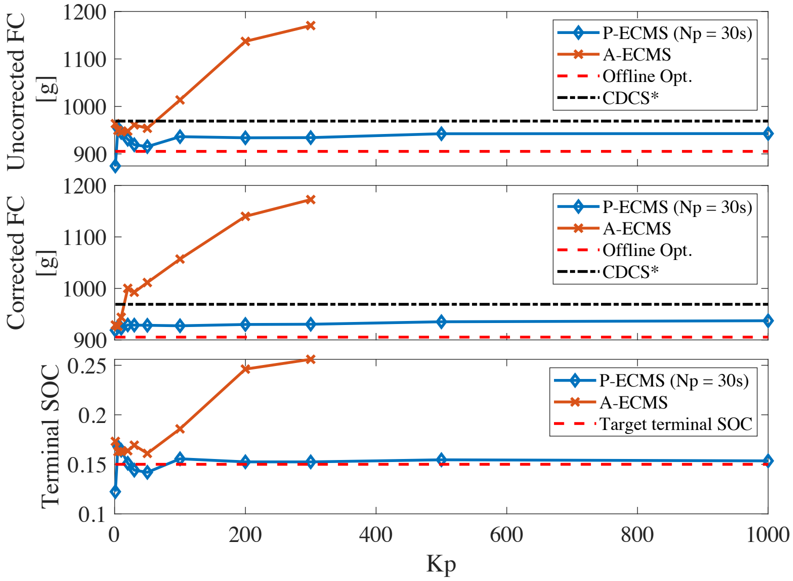


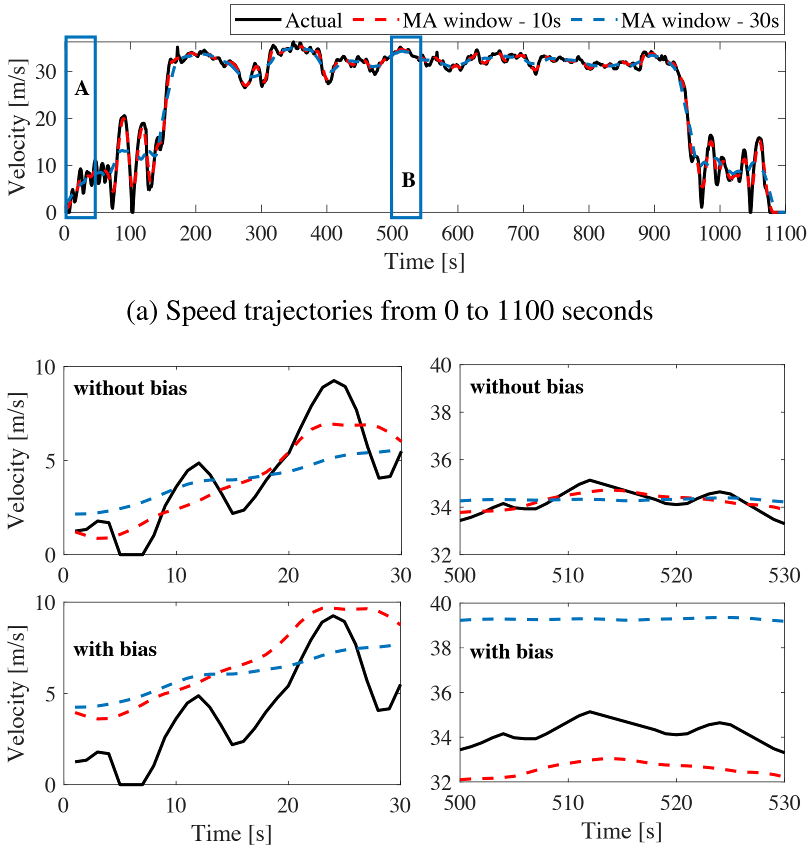
Figure 3.11: Influence of the co-state update rate K_p on the performance of P-ECMS with $N_p = 30$ seconds and $p(0)$ = -300

3.3.3 Robustness with respect to the Co-State Update Rates K_p

Figure 3.11 compares the performance of the P-ECMS and the A-ECMS with different co-state update rates K_p. The following simulation parameters are applied: prediction horizon $N_p = 30$ seconds and initial co-state $p(0) = -300$. It can be observed that good fuel economy and terminal SOC value can be guaranteed with the P-ECMS as long as K_p is not too small. Notably, even when K_p is as large as 1000, the P-ECMS can achieve better performance, compared to CDCS*. Note also that the A-ECMS can achieve good fuel economy only when K_p is relatively small. For this particular driving cycle, large co-state oscillations are observed in the A-ECMS when $K_p > 300$. If the absolute value of the co-state is very large, the controller will always engage the engine, and the battery will be charged so that battery SOC will be maintained at the upper bound. By comparison, if the absolute value of the co-state is considerably low, the controller will always employ the battery, making the operation CDCS. Meanwhile, it can also be observed that with the A-ECMS for a chosen prediction horizon and initial co-state, the terminal SOC and resulting total fuel consumption are sensitive to the change in K_p. As for the P-ECMS, the performance is found to be robust when $K_p > 30$.

However, that performance degrades slightly when K_p becomes very large, which is because a very large tuning rate will cause large oscillations in the resulting co-state trajectory. Even so, as can be observed in Fig. 3.11, the resulting performance with the P-ECMS is satisfactory within a large range of K_p.

3.3.4 Robustness with respect to Various Levels of Accuracy in Velocity Prediction



(a) Speed trajectories from 0 to 1100 seconds

(b) Predicted speed trajectories with $N_p = 30$ seconds: at 1 second denoted as part A (left) and at 500 second denoted as part B (right)

Figure 3.12: Comparison of speed trajectories: actual data and moving-averaged (MA) signals with two different data window.

In reality, prediction errors inevitably exist with any short-horizon velocity forecasting algorithm. Since velocity prediction itself is not the focus of this work, here, two types of inaccurate velocity profiles are employed to demonstrate the robustness of the P-ECMS in the presence of inaccurate velocity prediction.

1. Unbiased Prediction – the moving average of the actual velocity profile with different window sizes
2. Biased Prediction – the moving average of the actual velocity profile with different window sizes offset by a random number in (-5,5) m/s

The top subplot of Fig. 3.12 shows the moving average of the entire driving cycle, with window sizes being 10, 20, and 30-second, respectively. Assuming the prediction horizon $N_p = 30$ seconds, the middle and bottom subplots illustrate the possible predicted velocity profiles around the beginning and at 500s of the trip. Specifically, note that the middle subplots represent when the predicted velocity profiles are the moving average of the actual velocity profile with different window sizes with unbiased predictions. By comparison, the bottom subplots show the case with biased predictions.

Figure 3.13 summarizes the performance of the P-ECMS in the presence of both inaccurate/unbiased and inaccurate/biased velocity predictions. The following parameters are used in simulation: the initial co-state $p(0) = -300$, the co-state update rate $K_p = 30$ and the prediction horizon $N_p = 30$ seconds. As can be seen from Fig. 3.12, as the window size of the moving average velocity increases (which indicates an increased level of inaccuracy), the difference between the actual velocity and the predicted velocity is no longer negligible. Nonetheless, by comparing the blue bars in Fig. 3.13, it can be seen that the difference in the resulting fuel consumption is remarkably small, irrespective of the level of inaccuracy in velocity prediction. Also, by comparing the blue and yellow bars in Fig. 3.13, it can be seen that the P-ECMS is robust even when the velocity prediction is biased[7].

3.3.5 Discussion on the Importance of SOC Planning

The importance of the effective SOC planning strategy discussed in Section 3.2.1 is demonstrated by comparing the performance of the P-ECMS with the same $p(0)$, K_p, N_p, but different SOC references in this section. The parameters are chosen as: the initial co-state $p(0) = -300$, the co-state update rate $K_p = 30$ and the prediction horizon $N_p = 30$ seconds.

The results obtained with the P-ECMS using three different types of SOC references are compared and shown in Fig. 3.14, where

1. Linear-in-time SOC ref:

$$\text{SOC}_d(k) = \text{SOC}_0 + \frac{k}{N_f}(\text{SOC}_0 - \text{SOC}_f), \tag{3.8}$$

[7] It is observed P-ECMS with biased prediction performs slightly better than that with unbiased prediction. The reason is unclear, and how the prediction accuracy would affect the performance of P-ECMS is subject to further investigation.

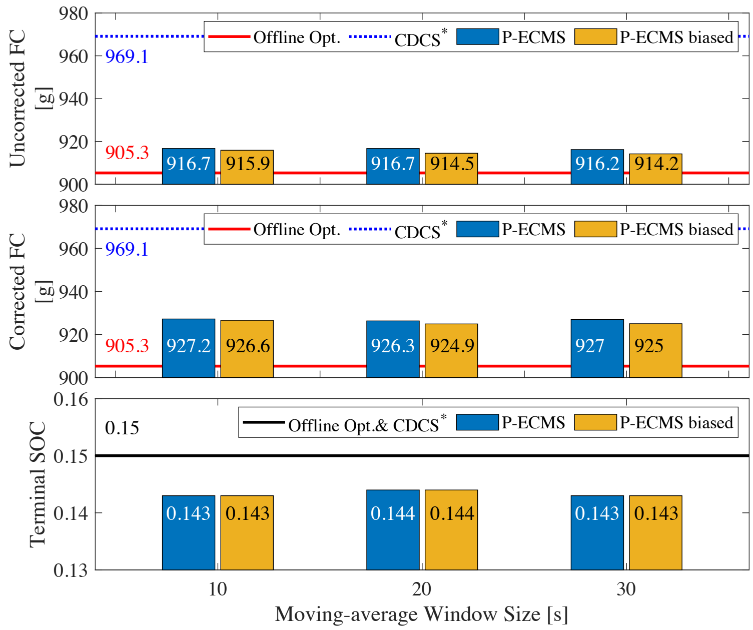


Figure 3.13: Influence of velocity prediction quality on the performance of P-ECMS with $N_p = 30$ seconds, $p(0)$ = -300 and K_p = 30

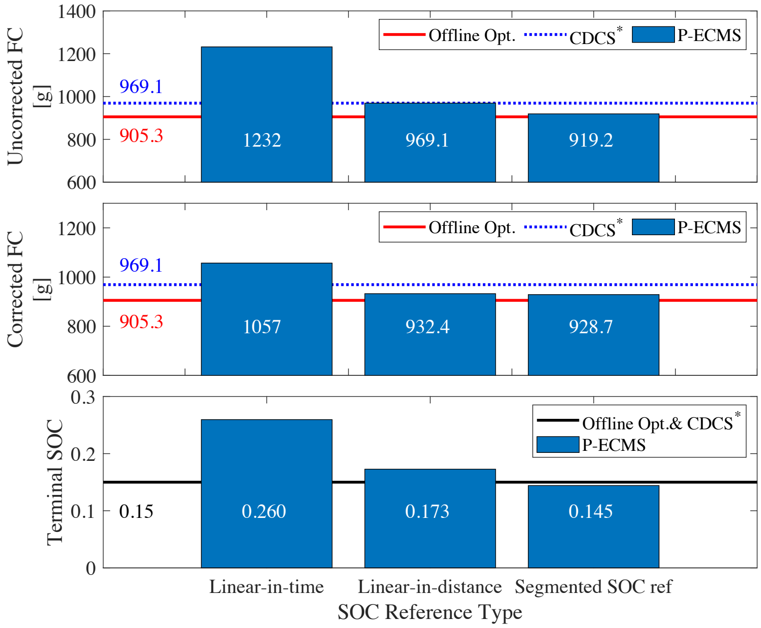


Figure 3.14: Influence of SOC reference on the performance of P-ECMS with $N_p = 30$ seconds, $p(0)$ = -300 and K_p = 30

where N_f represents the total trip time represented in discrete-time, and k represents the time $t = k\Delta t$.

2. Linear-in-distance SOC ref:

$$\mathrm{SOC}_d(k) = \mathrm{SOC}_0 + \frac{s(k)}{s_N}(\mathrm{SOC}_0 - \mathrm{SOC}_f), \tag{3.9}$$

   where s_N represents the total trip distance, and $s(k)$ represents the distance travelled (possibly predicted) up to time $t = k\Delta t$.

3. Segmented SOC ref: SOC planning-based reference with constructed simplified velocity profile as discussed in Section 3.2.1.

As can be seen from Fig. 3.14, among all the P-ECMS results, the fuel economy with the linear-in-time SOC reference is the worst. On the other hand, the fuel economy with the linear-in-distance SOC reference is better than that with the linear-in-time SOC reference. Even so, the P-ECMS achieves the best fuel economy with the SOC planning-based reference of the three. Also, as can be seen from the third subplot in Fig. 3.14, the actual terminal SOC values resulting from the P-ECMS with linear-in-time and linear-in-distance reference are much higher than the target terminal SOC. Moreover, the P-ECMS with SOC reference generated from our proposed method has led to better utilization of energy stored in the battery.

3.4 Summary

In this chapter, a P-ECMS is proposed combining the SOC node-planning strategy developed in Chapter 2 to minimize the fuel consumption of a PHEV. The proposed P-ECMS is a simple variant of the ECMS but adjusts the co-state based on differences in the predicted future SOC and the corresponding reference SOC. Introducing predictions such as those in the P-ECMS can reduce the sensitivity of fuel economy and terminal SOC performance to control parameters. The reduction in the sensitivity to the choices of the control parameters indicates the potential of the P-ECMS when encountering an unknown trip.

Simulation results demonstrate that the fuel economy performance of the P-controller type of A-ECMS depends heavily on the selected $p(0)$ and K_p pairs. Depending on the choice of $p(0)$ and K_p pairs, there is no guarantee that the actual terminal SOC will converge on the desired SOC in the A-ECMS. This eventually leads to fuel economy deterioration. By comparison, the proposed P-ECMS has shown that it can achieve better fuel economy robustly responding to the initial co-state, the co-state update rate, the prediction horizon, and compensating for inaccuracy in the velocity prediction.

The benefits of the P-ECMS can be summarized as follows:

1. Compared to the A-ECMS, the introduction of predictions from the P-ECMS can reduce the gap between the actual and desired terminal SOC on average.

2. Compared to the A-ECMS, the introduction of prediction from the P-ECMS demonstrates its robustness to the parameters in the co-state update formula.

3. Its resulting fuel consumption is robust to the prediction horizon as well as the velocity misprediction.

4. Compared to PMP, the P-ECMS does not solve the TPBVP as related to the minimum fuel consumption problem (3.1) with the actual velocity profile of the entire trip to get the correct initial co-state and co-state trajectory. It considers only the short-term future and gradually adjusts the co-state through feedback.

CHAPTER 4

Combined Velocity and Power-Split Optimization

The work presented in this chapter has been published in

1. [32] D. Chen, N. Prakash, A. Stefanopoulou, M. Huang, Y. Kim, and S. Hotz, "Sequential Optimization of Velocity and Charge Depletion in a Plug-in Hybrid Electric Vehicle.", *14th International Symposium on Advanced Vehicle Control, 2018*

2. [26] D. Chen, Y. Kim, M. Huang and A. Stefanopoulou, "An Iterative and Hierarchical Approach to Co-optimizing the Velocity Profile and Power-split of Plug-in Hybrid Electric Vehicles," *2020 American Control Conference (ACC)*, 2020, pp. 3059-3064, doi: 10.23919/ACC45564.2020.9147804.

4.1 Introduction

When shifting from the control of human-driven to automated PHEVs, recent developments in advanced driver assistance systems (ADAS) grants the freedom to manipulate the vehicle velocity within acceptable traffic constraints [42, 43] and provide the opportunity to reduce fuel consumption beyond what is possible with powertrain optimization alone. The opportunity also comes with the challenge to unify the velocity optimization and the management of multiple energy resources for PHEVs. At first glance, it may seem intuitive to split the above co-optimization problem into velocity optimization and charge depletion optimization sub-problems. The actual optimum, though, might not occur when they are solved separately and independently. One may pose the following question: is it justified to keep this sequential or decentralized structure for this problem without considering its co-optimization nature (communications between velocity optimization subsystem and power management subsystem)? The choice between a centralized controller and a sequential optimization define two limiting design extremes. The optimality of the sequential and decentralized structure for PHEV problem is not guaranteed since the two subsystems are closely related: velocity is the input to the power split optimization. The powertrain output torque influences the velocity through vehicle dynamics.

Over the years, several approaches have been proposed for the combined energy management and eco-driving (economical driving) of HEVs [44, 19, 21] with the concept of predictive cruise control. The authors in [45] showed that the optimal trajectory might be represented by using only two phases separated by a coasting phase under the assumption of limited traffic preview. Then, two-stage PMP[46] was applied to derive the formulation of an optimal controller analytically. In [44], a three-layer predictive control scheme with different update frequency and prediction horizons is proposed. The top layer plans the kinetic and electric energy in a convex optimization problem. The corresponding dual variables present as state references and fuel equivalents and are used adaptively in a real-time decision layer by solving an ECMS type of instantaneous optimization. In [21], the authors split the problem into long prediction horizon SOC planning and a short horizon combined velocity and SOC planning.

This chapter starts by investigating the potential of fuel efficiency improvement of a PHEV by the sequential smoothing of a velocity profile given traffic constraints and the optimization of its charge depletion strategy in an offline manner. First, the vehicle velocity is smoothed by minimizing the quadratic acceleration term subject to the traffic-acceptable following distance from a lead vehicle based on the lead vehicle's velocity preview. Afterward, the fuel consumed is minimized for the smoothed velocity profile by an optimal charge depletion policy that uses the knowledge of the entire trip profile to stretch the battery charge till the end of the trip instead of the CDCS strategy.

Afterward, this chapter investigates the additional fuel economy benefits with the direct fuel consumption minimization by co-optimizing the vehicle-following and the hybrid powertrain subsystem in a centralized manner upon sequentially optimizing the two subsystems (acceleration minimization followed by power-split optimization). However, challenges exist in obtaining the numerical solution of the co-optimization problem due to the following aspects: (i) a mixed-integer problem structure (engine on/off decision), (ii) the presence of pure state constraints whose dynamics do not explicitly depend on the control inputs (time-varying position constraints), and (iii) unstable dynamics when representing the vehicle-following dynamics by a double integrator. An iterative and hierarchical numerical strategy is proposed to resolve these difficulties, combining the gradient projection (direct method) with the single shooting (indirect method). Single shooting is used to deal with the engine on/off decisions in the power-split optimization, and the gradient projection is used to deal with the unstable dynamics and the state constraints.

4.2 Acceleration Minimization for Velocity Optimization

The fuel consumption minimization problem in-between re-charging events in this section follows a sequential optimization methodology. First, the velocity is smoothed, assuming perfect knowledge of the traffic constraints on the entire trip. Then, the battery charge depletion is optimized based on the smoothed velocity profile. Although this sequential optimization structure might be sub-optimal, it maintains the relative simplicity of two subsystems and makes the computation tractable. It is meaningful because:

1. It defines a vehicle-independent velocity optimization structure, and any vehicle with automatic longitudinal control can employ these optimal controls.

2. The use of a linear vehicle model in velocity optimization is computationally attractive for eventually solving an optimal control problem.

Moreover, it is also readily applicable to the existing energy management system (EMS). Since this section aims to evaluate the potential of our proposed sequential optimization method, perfect knowledge of the traffic constraints on the entire trip is assumed.

4.2.1 Velocity Smoothing based on Traffic Constraints

In this section, the velocity is smoothed through acceleration minimization. The vehicle velocity and position are described as a point-mass system:

$$\dot{z} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a \tag{4.1}$$

where $z = [s\ v]^T$, with s, v, the position and velocity of the vehicle, respectively; a is the acceleration of the vehicle. The acceleration limits are $a^{\min} \leq a(t) \leq a^{\max}$, and the speed limits on the various road segments specify the vehicle speed constraints as $v^{\min}(t) \leq v(t) \leq v^{\max}(t)$. Meanwhile, the position of the vehicle is constrained by the traffic condition and its lead vehicles. The vehicle should have an acceptable and safe traffic following behavior and therefore its position needs to satisfy $s^{\min}(t) \leq s(t) \leq s^{\max}(t)$. Here for simplicity, it is assumed the constraints on the acceleration, a, and the velocity of the target vehicle, v, are time-invariant.

Converting the continuous time vehicle following sub-system into discrete time system leads to the following linear time invariant system defined in discrete time:

$$z_{k+1} = Az_k + Ba_k, \tag{4.2}$$

where $z_k = [s_k, v_k]^T$ is the vehicle state, and a_k is the acceleration at time k, with A, B corresponding matrices [32]. The acceleration minimization is performed with respect to a sequence of control inputs $U = [a_0\ a_1\ ...\ a_{N-1}]^T$, N is the trip horizon and the problem is converted to a large dimensional constrained optimization problem. The objective function has the following form:

$$\min_{U} \quad J_1 = U^T U, \quad \text{s.t} \quad GU \leq W + Tz_0, \tag{4.3}$$

where $z_0 = [s_0\ v_0]^T$ is the initial position and velocity of the vehicle. The constraints on states are converted to control constraints. The derivation of matrices G, W and T can be seen in [32]. This is a standard quadratic minimization problem, and is solved by Matlab QP solver or other state-of-art QP solvers like [47].

The limits on the velocity are reasonable on almost all U.S.A. roads [48], [49]. The position of the target vehicle, however, depends on the position and velocity of its leader vehicle (s_L, v_L) and hence subjects to time-varying constraints defined as in [49]:

$$s^{\min} = s_L + v_L \frac{L}{10}, s^{\max} = s_L + \begin{cases} v_L d^{\max}, & \text{if } v_L < 9 \text{ m/s (20 MPH)}, \\ v_L d^{\min}, & \text{otherwise}, \end{cases} \tag{4.4}$$

where v_L and s_L are the velocity and position of the preceding vehicle, respectively; L is 4.5 m (1 car length), d_{max} is 3 m and d_{min} is 1.2 m.

The torque demand of the vehicle tp_k at time $t = k$ is defined as the function of the vehicle velocity v_k and acceleration a_k through a regression model that needs to be satisfied by the low-level powertrain components

$$tp_k = \alpha_0 + \alpha_1 \cdot a_k + \alpha_2 \cdot v_k + \alpha_3 \cdot v_k^2, \tag{4.5}$$

where $\alpha_i, i \in \{0, 1, 2, 3\}$ are the regression parameters.

4.2.2 Offline Controller Performance Evaluation

Various driving trips are created to evaluate the controller performance by stitching several federal test cycles, including the Urban Dynamometer Driving Schedule (UDDS), the Highway Fuel Economy Test (HWFET), the US06 Supplemental Federal Test Procedure. Here the results are reported for two specific permutations as presented in Table 4.1. Offline simulations have shown that different trips with different characteristics have different potentials for fuel economy improvement upon CDCS when doing charge blending optimization. The following two scenarios are selected to represent two extreme cases:

Table 4.1: Definition of two driving trips

Trip Name	Cycle sequence	Time
Worst CDCS	3×US06+UDDS+HWFET 2×UDDS+HWFET	2 hour 4 minutes
Best CDCS	3× HWFET+ 3× UDDS + 3× US06	2 hour 16 minutes

1. Worst CDCS Trip: the case where the result with CDCS strategy on a real velocity profile is expected to be far from optimal. The trip is arranged such that high power is frequently demanded at the beginning of the trip. With the optimal depletion strategy, the engine would be heavily engaged at first. Consequently, it reserves more battery capacity for later in the trip with much lower power demand. By contrast, the battery would be quickly depleted because of the initial high power demand. The engine has to engage during the rest of the trip to maintain the minimum SOC level.

2. Best CDCS Trip: the case where the result with CDCS strategy on a real velocity profile is expected to be near optimal. The trip is arranged such that relatively low power demand and high vehicle speed appear at the beginning of the trip. Therefore, it is more efficient for the system to use the battery first and engage the engine at the end of the trip, which has high power demand.

The total trip distance of Worst CDCS Trip is 66.87 miles, and the total trip distance of Best CDCS Trip is 77.13 miles. The total trip energy of the presented cycle exceeds the available battery energy of the simulated vehicle in both scenarios. Consequently, the charge depletion strategy will be important and have a significant influence on fuel consumption. The trip profiles and corresponding energy management strategies are summarized in Table 4.2.

Table 4.2: Velocity and energy management scenarios considered in vehicle simulation

Scenario	Velocity	Energy Management
RV-CDCS	Real	Charge depleting with EV mode followed by charge sustaining with HV mode
RV-OCBD	Real	Optimal charge blending
SV-CDCS	Smooth	Charge depleting with EV mode followed by charge sustaining with HV mode
SV-OCBD	Smooth	Optimal charge blending

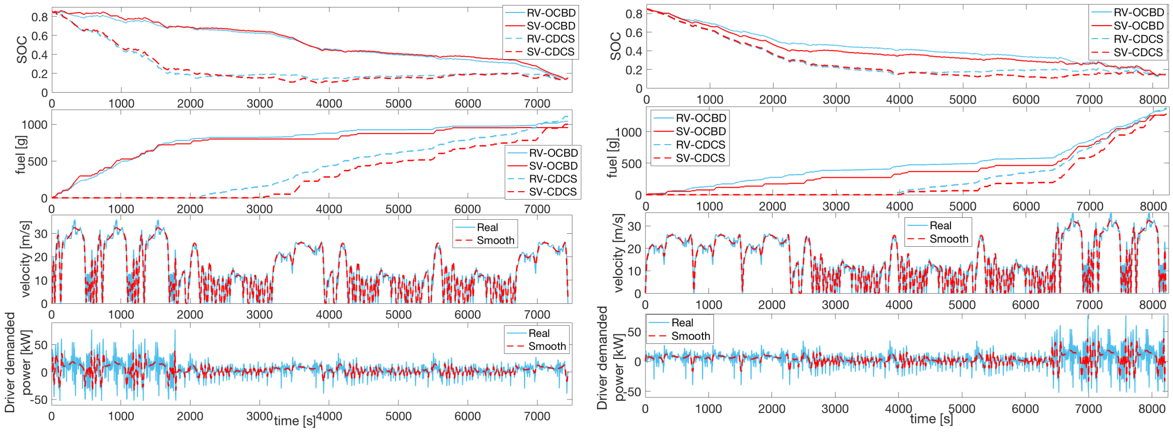
Note that the commercially applied CDCS strategy typically contains a complicated logic that is not publicly available. To allow comparisons, the baseline energy management strategy that approximates the CDCS strategy is defined as charge depletion followed by the optimized charge sustaining operation as discussed before.

Table 4.3: Fuel consumption, engine-on time, energy consumption results for different scenarios: Worst CDCS Trip. SOC(0)=0.85, SOC(t_f)=0.15.

Scenario	Fuel (kg)	% Fuel Reduct.	Engine-on Time (%)	Braking Energy @Wheel (kWh)	Tractive Energy @Wheel [kWh]	Pure EV miles
RV-CDCS	1.11	0	13.49	4.93	13.71	25.97
RV-OCBD	1.04	6.0	11.52	4.93	13.71	N/A
SV-CDCS	1.00	9.7	12.20	3.18	11.91	30.55
SV-OCBD	0.96	13.7	11.22	3.18	11.91	N/A

Table 4.4: Fuel consumption, engine-on time, energy consumption results for different scenarios: Best CDCS Trip. SOC(0)=0.85, SOC(t_f)=0.15.

Scenario	Fuel (kg)	% Fuel Reduct.	Engine-on Time (%)	Braking Energy @Wheel (kWh)	Tractive Energy @Wheel [kWh]	Pure EV miles
RV-CDCS	1.37	0	13.43	5.16	15.30	39.30
RV-OCBD	1.36	1.1	13.99	5.16	15.30	N/A
SV-CDCS	1.28	7.0	13.48	3.29	13.38	40.85
SV-OCBD	1.28	7.0	13.70	3.29	13.38	N/A



(a) Worst CDCS Trip

(b) Best CDCS Trip

Figure 4.1: Simulation results with different scenarios over two driving trips

4.2.3 Results and Discussion

The summary of the results are presented in Table 4.3 and 4.4. SOC trajectory and fuel consumption trajectory for these two scenarios are presented in Fig. 4.1a and 4.1b, respectively. The results show that, on Worst CDCS Trip, the sequential optimization of velocity and charge depletion (SV-OCBD) reduces the total fuel consumption by 13.74 % compared to the baseline depletion strategy on the Real Velocity profile (RV-CDCS). On Best CDCS Trip, the sequential optimization of velocity and charge depletion (SV-OCBD) reduces the total fuel consumption by 7.07 % compared to the baseline depletion strategy on the Real Velocity profile (RV-CDCS).

For the Worst CDCS Trip, the significant improvement of RV-OCBD from RV-CDCS is not unexpected. With CDCS, even though the engine is expected to operate at most efficient points after entering HV mode, it will be extensively used during the rest of the trip. This can be seen from Table 4.3, where compared to RV-CDCS strategy, RV-OCBD strategy reduces the total amount of time of engine operation in the moderate power region by slightly increasing the time for the engine to operate in the middle power region. As is shown in Table 4.3, the baseline strategy on real velocity has a longer duration of engine-on operation compared to the strategy that gives the optimal SOC depletion. However, for the Best CDCS Trip, the power demand at the beginning of the trip is moderate, and the high power demand appears at the end of the trip. The importance of " saving the battery for better use " decreases. Consequently, the RV-CDCS strategy that uses engine on the later part of the trip would be near-optimal compared to the RV-OCBD strategy.

The results from Best CDCS Trip conveys an important message: even when there is a small potential for charge depletion optimization (around 1 %), velocity smoothing provides a valuable opportunity to improve the fuel economy further. In addition, smooth velocity provides significant benefits even for a PHEV, where regenerative braking can be fully utilized.

A thorough comparison between different strategies on Worst CDCS Trip and Best CDCS Trip as shown in Table 4.3 and 4.4 gives us the following insight:

1. Smooth velocity profile significantly reduces average power demand. The lower power demand at the beginning of the trip enables longer mileage covered (with CDCS) before switching to hybrid mode as shown in Table 4.3 and 4.4.

2. Reducing tractive trip energy reduces the overall fuel consumption by a greater amount than what can be achieved with charge depletion optimization. Consequently, SV-CDCS has higher fuel reduction than RV-OCBD compared to RV-CDCS as shown in Table 4.3 and 4.4. Unfortunately, smooth velocity (SV) requires a high level of automation, namely vehicle following capability such as in cooperative adaptive cruise control [50].

3. The individual benefits of the velocity smoothing and charge depletion optimization are not additive. This happens because the negative energy (thus the potential for regenerative braking) is reduced by velocity acceleration minimization as shown in Table 4.3 and 4.4. In general, the SV diminishes the regenerative braking opportunity, but it creates the immense savings from the elimination of expensive and unnecessary acceleration that needs braking.

4. Despite the non-cumulative nature of individual benefits, the total reduction in fuel

consumption is still significant (from RV-CDCS to SV-OCBD).

4.3 An Iterative and Hierarchical Numerical Approach to Co-Optimization

In the previous section in this chapter, the fuel-economical driving of a PHEV is pursued by sequentially optimizing its velocity and charge depletion. The overall system is decomposed into a vehicle-following subsystem and a hybrid powertrain subsystem. Considering that people usually purchase a PHEV for its ability to reduce fuel consumption in the daily commute, it is meaningful to co-optimize the vehicle-following and powertrain dynamics under a *centralized* framework with fuel minimization as a single objective. However, challenges exist in the co-optimization because it results in a mixed-integer nonlinear optimal control problem with pure state constraints (up to second-order [29]), which is considerably hard to solve numerically.

In this section, the minimum fuel consumption problem is formulated for the entire trip as an OCP unifying the vehicle and hybrid powertrain dynamics. It then seeks to solve the OCP to obtain the velocity profile and the corresponding powertrain operation simultaneously, thus providing a benchmark for online strategies. Note that when the powertrain is included in the velocity planning, DP is the common approach to obtain the numerical solution [18, 17, 51, 52]. However, due to the discretization of additional control inputs and states, compromises are needed between the computational cost and the solution accuracy.

The main contributions of this section are twofold: (i) a hybrid solution scheme is proposed to address the difficulties in solving the mixed-integer OCP numerically with guaranteed pure state inequality and terminal equality constraints without discretizing the controls and states into grids, and (ii) considerable additional fuel economy benefit of the co-optimization is demonstrated upon the sequential optimization approach.

4.3.1 Co-Optimization Problem Formulation

The objective of co-optimizing the vehicle-following and hybrid powertrain subsystems is to minimize the total fuel consumption for the considered trip without constraints violation. In this section, first, the minimum fuel consumption (denote as co-optimization in the sequel) problem is formulated in continuous time. Then, a TPBVP is obtained and discretized based on the necessary condition for optimality. Afterward, issues related to the instantaneous minimization and the stability of the TPBVP are discussed.

4.3.1.1 Formulation of the Co-Optimization Problem

Remark 2. Note that similar to the discussion presented in Chapter 2, in the co-optimization problem considered in this section, the torque and speed limits of the motor MG2 and generator MG1 are all converted to the constraints on the engine, with the invariant constraints Ω on the engine torque and speed in HV mode in this section defined by a polytope of the form $[te, ne] \in \Omega := \{[te, ne] | H \begin{bmatrix} te \\ ne \end{bmatrix} \leq h\}$. The original admissible set of the engine is $[te, ne] \in \tilde{\Omega}(v) := \{[te, ne] | H \begin{bmatrix} te \\ ne \end{bmatrix} \leq \tilde{h}(v)\}$, where $h = \tilde{h}(v)$ except for only a single entry of $\tilde{h}$ depends on v. Different from the pure power-split optimization of a human-driven PHEV, here the constraints become a mixed state and control type since v becomes a state rather thanan external input. Neverthless, to simply the development, this section still considers the maximal Ω, such that $\Omega \subset \tilde{\Omega}(v), \forall v$. This is a reasonable simplification since the tightened constraint is an upper bound on engine speed that is rarely visited.

Define the system state $x = [s\ v\ \text{SOC}]^T$, consisting of the position, velocity and battery SOC, and the control input $u = [a\ ne\ te]^T$, including the acceleration, engine speed and torque. The minimum fuel consumption problem in continuous-time with given trip time t_f is formulated as[1]

$$\text{minimize} \quad J = \int_0^{t_f} \dot{m}_f(ne, te)dt, \tag{4.6a}$$

subject to the system dynamics ((4.1), (A.14)), the initial and desired terminal condition

$$x(0) = x_{int}, \ \ \text{SOC}(t_f) = \text{SOC}_f, \tag{4.6b}$$

and the state constraints on the ego vehicle's position and velocity

$$g(x) = \begin{bmatrix} s - s^{\max} \\ v - v^{\max} \\ -s + s^{\min} \\ -v + v^{\min} \end{bmatrix} \leq 0, \tag{4.6c}$$

where $v^{\min}$ and $v^{\max}$ are the minimum and maximum allowable velocity of the ego vehicle determined in consideration of the road type, location, and traffic flow. They are chosen to be constant values in this thesis. $s^{\min}$ and $s^{\max}$ are the minimum and maximum allowable distance

[1] The dependence on t is omitted for simplicity.

and are chosen to avoid cut-ins from adjacent lanes and collisions with the preceding vehicle, respectively.

$$s^{\min} = s_l(t-3) - 4 \tag{4.6d}$$

$$s^{\max} = s_l(t-1) - 1, \tag{4.6e}$$

where $s_l(t-3)$ and $s_l(t-1)$ refer to the lead vehicle's position 3s and 1s ago, respectively. g is continuous differentiable with respect to x. The system is also subject to the control constraints

$$\mathcal{U} := \{u = (a, te, ne) | a \in [a^{\min}, a^{\max}],\ [te, ne] \in [0, 0] \cup \Omega\}, \tag{4.6f}$$

where $a^{\min}$ and $a^{\max}$ are the minimum and maximum allowable acceleration due to the physical limits on the vehicle, respectively.

Note that since the considered PHEV has two operation modes: HV mode with engine-on and EV mode with engine-off, the engine on/off decision needs to be made. Consequently, the co-optimization problem (4.6) is a mixed-integer optimal control problem with time-varying pure state constraints (4.6c).

4.3.1.2 Necessary Condition for Optimality and TPBVP

From the necessary condition for optimality [29], the Hamiltonian for (4.6) is defined as:

$$H \ = \dot{m}_f + p^T f(x, u) = \dot{m}_f(ne, te) + \big(p_1 v + p_2 a + p_3 \dot{\text{SOC}}\big), \tag{4.7}$$

where $p = [p_1, p_2, p_3]^T$ is the co-state. Based on the discussions at the beginning of this section, the mixed state and control constraints are approximately considered as pure control constraints. The pure state constraints (4.6c) can be adjoined to the Hamiltonian (4.7) to form the following Lagrangian through a direct adjoining approach [53]

$$L \ = L(x, p, u, \mu) = H + \mu^T g(x) = \dot{m}_f(ne, te) + p_1 v + p_2 a + p_3 \dot{\text{SOC}} + \mu^T g(x), \tag{4.8}$$

where $\mu \in \mathbb{R}^4$ is the vector of the additional Lagrange multipliers associated with the pure state constraints (4.6c). With an informal formulation of the maximum principle that is used as a recipe while dealing with optimal control problems with state constraints in an applied setting [53], the optimal control u^* at time t is the instantaneous minimizer of the Hamiltonian (4.7)

$$u^* \ = \underset{u \in \mathcal{U}}{\arg\min} \big\{\dot{m}_f + p^T f(x, u)\big\} = \underset{u \in \mathcal{U}}{\arg\min} \big\{\dot{m}_f(ne, te) + p_1 v + p_2 a + p_3 \dot{\text{SOC}}\big\}. \tag{4.9}$$

The adjoint equation that needs to be satisfied by the co-state p is expressed as

$$\dot{p} \; = -\nabla_x L(x,p,u,\mu) = - \left[\left(\frac{\partial f(x,u)}{\partial x} \right)^T p + \left(\frac{\partial g(x)}{\partial x} \right)^T \mu \right], \tag{4.10}$$

since $\dot{m}_f$ is not an explicit function of x.

Discretizing the state and co-state dynamics with Forward Euler scheme and time step Δt, the following TPBVP can be obtained in discrete-time with unknown initial co-state p_0

$$x_0 = x_{int}, \;\; x_N = x_f \tag{4.11a}$$

$$u_k^* \; = \arg\min_{u_k \in \mathcal{U}} H_k \tag{4.11b}$$

$$x_{k+1} = x_k + f(x_k, u_k^*)\Delta t \tag{4.11c}$$

$$p_{k+1} = \left(I - \Delta t \frac{\partial f^T(x_k, u_k^*)}{\partial x_k} \right) p_k - \Delta t \frac{\partial g^T(x_k)}{\partial x_k} \mu_k \tag{4.11d}$$

$$g(x_k) \leq 0, \;\; \mu_k \geq 0, \;\; g(x_k)^T \mu_k = 0. \tag{4.11e}$$

4.3.1.3 Instantaneous Hamiltonian Minimization

To solve the TPBVP (4.11) in discrete-time numerically, first the control input at $t = k$ needs to be obtained through instantaneous Hamiltonian minimization (4.11b)

$$u_k^* \; = \arg\min_{u_k \in \mathcal{U}} \left\{ \dot{m}_{f,k} + p_k^T f(x_k, u_k) \right\} = \arg\min_{u_k \in \mathcal{U}} \left\{ H_{ps,k} + p_{2,k} a_k + p_{1,k} v_k \right\}, \tag{4.12}$$

where $H_{ps,k} \; = \; \dot{m}_{f,k}(ne_k, te_k) + p_{3,k}\dot{\text{SOC}}_k$ represents the Hamiltonian of the power-split optimization problem [54].

Note that the problem (4.12) with $u_k \; = \; [a_k, ne_k, te_k]^T$ has a hierarchical structure that can be exploited to decouple the instantaneous minimization problem: at time $t = k$ with the current state x_k and co-state p_k, for a given acceleration a_k, the engine speed ne_k and torque te_k can be chosen such that the fuel rate $\dot{m}_{f,k}$ is minimized in the same manner as in the power-split optimization. This means that, for each given a_k its corresponding optimal engine operation $ne_k^*(a_k, x_k, p_k), te_k^*(a_k, x_k, p_k)$ can be found. Consequently, the search for the control inputs as an instantaneous minimization problem in the original three-dimensional space can be decomposed in two lower dimensional sub-spaces: one-dimensional search on acceleration a_k, and two-dimensional search on engine torque te_k and engine speed ne_k with given acceleration a_k

$$a_k^* = \underset{a \in [a_k^{\min}, a_k^{\max}]}{\arg\min} \; H_{ps,k}^*(a, x_k, p_{3,k}) + p_{2,k}a + p_{1,k}v_k. \tag{4.13a}$$

Since $p_{1,k}v_k$ is determined at $t = k$, (4.13a) is equivalent to

$$a_k^* \; = \underset{a \in [a_k^{\min}, a_k^{\max}]}{\arg\min} \; H_{ps,k}^*(a, x_k, p_{3,k}) + p_{2,k}a. \tag{4.13b}$$

The typical optimization-based approach in the (P)HEV energy management is based on PMP and is solved numerically through single shooting on a related TPBVP rather than multiple shooting [55]. This is because engine on/off decisions need to be made for the (P)HEV energy management, resulting in a mixed-integer nonlinear optimization problem.

In the power-split optimization, the SOC is the only state considered and is modeled as a slow integrator. In the absence of active SOC constraints [54] and mixed state and control constraints, the resulting co-state is continuous. However, for the co-optimization problem, (i) additional states (velocity and position of the vehicle) with unstable, fast dynamics need to be incorporated into the TPBVP, while the combination of the fast unstable and slow dynamics complicates the indirect approach. (ii) The pure state constraints due to the preceding vehicle need to be explicitly considered, which yield discontinuities in the co-state trajectories. As is indicated in (4.11d) and (4.11e), if any of the state inequality constraints is active, the corresponding additional Lagrange multiplier would cause jumps in the co-state. Moreover, the dynamics of the position do not explicitly depend on the control input, making it hard to apply finite difference approximation related strategy as in [56] to deal with position constraints.

Meanwhile, for the discrete-time TPBVP (4.11), apart from the vehicle-following dynamics (a double-integrator) that are unstable forward-in-time as discussed above, in the meantime for the co-state dynamics (4.11d), the matrix in front of p_k is

$$A_p \doteq I - \Delta t \frac{\partial f^T(x_k, u_k^*)}{\partial x_k} = \begin{bmatrix} 1 & -\Delta t & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{\partial \dot{\text{SOC}}}{\partial v} & 1 - \frac{\partial \dot{\text{SOC}}}{\partial \text{SOC}} \end{bmatrix}. \tag{4.14}$$

In (4.14), $||A_p||_{\max} = \max_{ij} |a_{ij}| = \max\{\Delta t + 1, |\frac{\partial \dot{\text{SOC}}}{\partial \text{SOC}} - 1| + |\frac{\partial \dot{\text{SOC}}}{\partial v}|\}$. Since the sampling time $\Delta t > 0$, $||A_p||_{\max} \geq \Delta t + 1 > 1$. Based on the norm equivalence, $||A_p||_2 \geq ||A_p||_{\max} \geq 1$, $\forall \{\Delta t, \frac{\partial \dot{\text{SOC}}}{\partial \text{SOC}}, \frac{\partial \dot{\text{SOC}}}{\partial v}\}$. Consequently, the unforced response of the co-state in the absence of the additional Lagrange multipliers is

$$p_{k+1} = (A_{p,k}A_{p,k-1}...A_{p,1}A_{p,0})p_0 \tag{4.15a}$$

$$||p_{k+1}||_2 = ||(A_{p,k}...A_{p,0})p_0||_2 \leq ||A_{p,k}||_2...||A_{p,0}||_2||p_0||_2. \tag{4.15b}$$

Since $||A_{p,k}||_2 > 1, \forall k$ as discussed above, the unforced co-state dynamics without additional Lagrange multipliers are also unstable forward-in-time. The unstable state and co-state dynamics make it hard to implement single shooting numerically forward in time.

4.3.2 Numerical Strategy

In this section, an iterative and hierarchical approach combining the gradient projection and the single shooting is proposed to address the numerical difficulties related to the TPBVP (4.11) for the co-optimization discussed in the previous section.

4.3.2.1 Iterative and Hierarchical Solution Strategy

As discussed in the previous section, it is difficult to solve the TPBVP (4.11) with single shooting alone forward-in-time due to (i) the unstable state and co-state dynamics, (ii) the presence of the state constraints (position) whose dynamics do not explicitly depend on the control inputs, and (iii) the jumps in the co-state dynamics due to the pure state constraints. However, in the meantime, it is still desirable to keep the single shooting structure in choosing the engine torque te and speed ne in the powertrain level because of its potential in handling the engine on/off decisions. As pointed out in Section 4.3.1.3, the acceleration and engine operation points can be chosen in a hierarchical and decomposed way in the Hamiltonian minimization.

To address the issues mentioned above, a hybrid and hierarchical strategy is proposed that combines single shooting (indirect method) to optimize the power-split operation with the gradient projection (direct method) to optimize the acceleration *sequence* and handle the state constraints. As a result, the stability of the single shooting that is used in power-split optimization can be maintained.

The original co-optimization problem is reformulated as

$$\text{minimize } J^*(U_a) = \Big(\sum_{k=0}^{N-1} \dot{m}_{f,k}\Delta t\Big)^*_{U_a}, \ U_a = [a_0, a_1, ..., a_{N-1}]^T, \tag{4.16}$$

and its solution strategy is summarized in **Algorithm 1**. Steps 1, 2, 5-7, 9 in **Algorithm 1** correspond to the upper-level optimization on the acceleration sequence and steps 3 and 8

Algorithm 1 Iterative solution strategy

```
1: $m \leftarrow 0$
2: $U_a^m \leftarrow$ Any feasible acceleration sequence $U_a^0$
3: $J^m \leftarrow$ Solve the power-split optimization $J^*(U_a^m)$
4: while $m <$ MaxIter do
5:     $\nabla J^m \leftarrow$ Estimated gradient
6:     $\tilde{U}_a^{m+1} = U_a^m + \alpha_m \nabla J^m$
7:     $U_a^{m+1} \leftarrow \arg\min_{GU_a \leq W + Tz_0} ||U_a - \tilde{U}_a^{m+1}||^2$
8:     $J^{m+1} = J^*(U_a^{m+1})$
9:     $U_a^m \leftarrow U_a^{m+1}$, $J^m \leftarrow J^{m+1}$, $m \leftarrow m + 1$
10: end
```

correspond to the lower-level power-split optimization.

In (4.16) the entire acceleration sequence U_a is chosen as the upper-level decision variables. Since the vehicle-following subsystem is a linear time-invariant (LTI) system, given a sequence of acceleration U_a and the initial state $z_0 = [s_0, v_0]^T$, the speed and position sequences of the vehicle are determined through the state transition matrix, and the torque demand sequence is also fully determined. The position, velocity, and acceleration constraints can be converted to polytopic constraints on acceleration. This is followed by the lower-level power-split optimization, where the optimal fuel consumption under the given input sequences ($J^*(U_a)$) can be obtained. Through iterative interaction between the upper and lower levels, eventually, the optimal acceleration sequence U_a^*, the optimal powertrain operations and its corresponding fuel consumption $J^*(U_a^*)$ can be obtained.

To summarize, in the proposed numerical solution strategy, the upper-level generates the acceleration sequence and copes with the position, velocity, and acceleration constraints, whereas the power-split optimization and powertrain component constraints are considered in the lower-level for the given acceleration sequence.

4.3.2.2 Gradient Estimation and Projection

Since empirical static maps are heavily involved in the considered problem, significant computation effort will be spent on interpolation if the gradient ∇J^m is calculated (step 5 in **Algorithm 1**) by solving a new power-split optimization problem with each perturbation in the acceleration sequence. To relieve the computational burden, the gradient is not computed by evaluating J^* with each perturbation, but is estimated by (i) plugging in a perturbed acceleration sequence, (ii) using the nominal powertrain command obtained in the previous iteration, (iii) simulating the powertrain system over the resulting velocity trajectory, and (iv) estimating the difference in fuel consumption by SOC correction.

Specifically, step 3 in **Algorithm 1** solves the power-split optimization (fuel minimization) problem with the feasible acceleration sequence U_a^m at the m-th iteration, which results in a terminal SOC value SOC_N^m as well as the optimal engine operation sequence

$$\omega^*_{e,traj}(U_a^m) = [ne_0^*(a_0^m), ..., ne_{N-1}^*(a_{N-1}^m)], \quad T^*_{e,traj}(U_a^m) = [te_0^*(a_0^m), ..., te_{N-1}^*(a_{N-1}^m)].$$

The perturbed acceleration sequence $U_{a,i}^m$ at the i-th entry of the m-th iteration is obtained

$$U_{a,i}^m(j) = \begin{cases} U_a^m(j) + \Delta a = a_j^m + \Delta a, & j = i \\ U_a^m(j) = a_j^m, & \text{otherwise.} \end{cases}$$

The corresponding perturbed velocity sequence $V_i^m = [v_{1,i}^m, ..., v_{N,i}^m]$ can be obtained with the state transition matrix. The perturbed driver demanded torque trajectory can be obtained correspondingly. The overall system is simulated forward with $U_{a,i}^m$, the resulting V_i^m, the driver demanded torque, as well as the nominal engine operation sequence $\omega^*_{e,traj}(U_a^m)$ and $T^*_{e,traj}(U_a^m)$. Because of the perturbation in the acceleration and its corresponding changes in the velocity and torque demand sequence, the terminal SOC value $\text{SOC}_{N,i}^m$ is different from SOC_N^m. The difference $\Delta\text{SOC}_i^m = \text{SOC}_{N,i}^m - \text{SOC}_N^m$ is used to calculate ΔJ_i^m through $\Delta J_i^m = 100\alpha\Delta\text{SOC}_i^m$ (α: fuel consumption corresponding to 1% difference in SOC). Finally, the corresponding partial derivative can be estimated as $\frac{\partial J}{\partial a_i^m} \approx \frac{\Delta J_i^m}{\Delta a}$. Here one full horizon simulation per perturbation $U_{a,i}^m$ is required, and takes $\sim 6ms$ for a $\sim$ 1800s trip with 1s sampling time. However, the computation time can be further improved since for each perturbation $U_{a,i}^m$ the trajectories will differ only after i.

Since the inequality constraints on acceleration create a polytope specified by the matrix G, W, T and the initial condition z_0, the projection step 7 in **Algorithm 1** becomes a constrained QP problem and can be solved with IBM ILOG CPLEX Optimization Studio [57].

It should be pointed out that with many empirical static maps and a relatively large step size, it would be possible that the proposed iterative approach leads to jumps among local solutions very close in terms of objective function values. However, since the acceleration sequence resulting from each gradient projection step is feasible, the iterations can also be stopped when the value of the objective function does not decrease for several consecutive iterations. Therefore, in the actual implementation, a fixed number of iterations is used.

4.3.3 Results and Discussions

In this section, the results of the co-optimization with the proposed solution strategy over several driving cycles with different SOC spans are presented and discussed. Especially, three optimization approaches are compared:

1. Power-split optimization over the original driving cycle, denoted by *org.+ps.opt.* and shown as the black dashed curve. Note that the CDCS is not used as the baseline here because the initial power demand for the considered trip is high and cannot be met in pure EV mode.

2. Sequential optimization: power-split optimization over a speed profile optimized with acceleration minimization, denoted by *acc.min+ps.opt.* and shown as the solid blue curve.

3. Co-optimization, denoted by *co-opt.* and shown as the solid red curve.

In all the simulations, $\alpha_m = 0.01$ and MaxIter = 1000 as in **Algorithm 1** are used. A time step $\Delta t = 1$s is used in discretization. The position constraints of the ego vehicle at $t = k$ depend on its preceding vehicle's position (s_l, known for the entire trip), and are given by

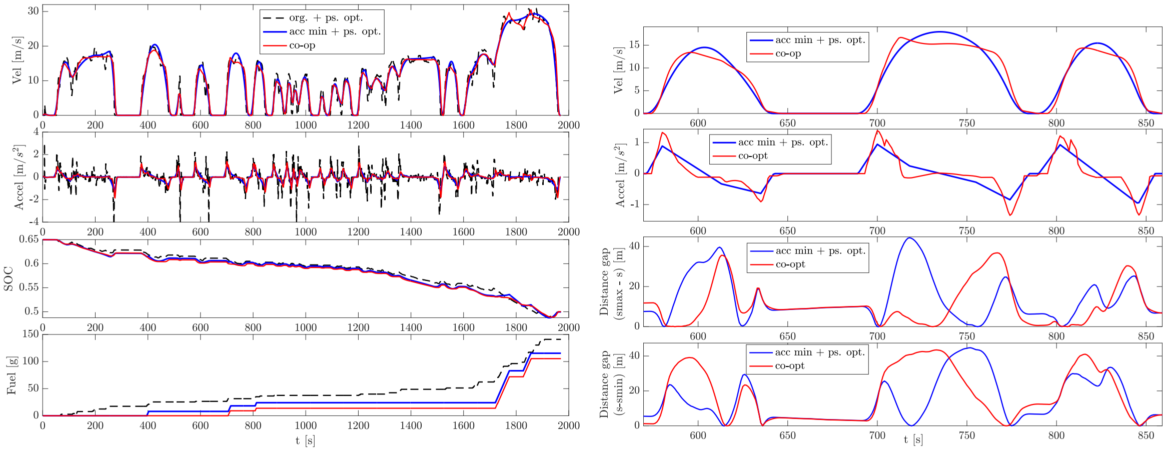
$$s_k^{\min} = s_{l,k-3} - 4, \;\; s_k^{\max} = s_{l,k-1} - 1. \tag{4.17}$$

Note that (i) additional fuel consumption for engine start-up is included in evaluating the total fuel consumption, (ii) the driving profile and the resulting position s_l of the lead vehicle used in simulations are obtained with the shifted original driving cycle, which defines the position constraints on the ego vehicle, (iii) the original driving cycles and the smoothed driving cycles through acceleration minimization satisfy the prescribed position constraints (4.17), and (iv) all the resulting trips travel the same time, distance and end with the same speed.

Table 4.5: Comparison of Fuel Consumption

Method	Final SOC	Fuel [g]	Improvement [%]
org. + ps. opt.	0.50	140.96	0
acc min. + ps. opt.	0.50	115.36	18.16 (0)
co-opt	0.50	105.40	25.23 (8.63)

As a case study, a trip of around 2000s is considered. The initial SOC is 0.65, and the desired terminal SOC is 0.50. In Fig. 4.2a, the first subplot shows the resulting velocity, the second subplot shows the acceleration, the third subplot shows the resulting SOC trajectory, and the fourth subplot shows the cumulative fuel consumption with three considered optimization approaches, respectively. The detailed fuel consumption and actual terminal SOC with each approach are summarized in Table. 4.5.



(a) Vehicle operation, battery SOC and fuel consumption.

(b) Zoomed in view of the vehicle operation comparison between the sequential and co-optimization results between 550-900s

Figure 4.2: Case study: initial SOC: 0.65, terminal SOC: 0.50

Table 4.6: Fuel economy benefit with co-optimization on various cycles upon other optimization approaches

Original cycle (also s_l)	SOC span	% Fuel reduction with co-optimization upon org. + ps. opt	% Fuel reduction with co-optimization upon acc min. + ps. opt
Partial UDDS (∼600s)	0.55→0.55	15.17	12.57
US06 (∼600s)	0.65→0.50	36.76	10.49
realistic cycle (∼2000s)	0.50→0.50	12.11	4.15

As is shown in Fig. 4.2a and Table. 4.5, the terminal SOC with all three optimization approaches are the same. Simulation results show that the co-optimization, admit with sub-optimality due to early stopping, can lead to a substantial decrease in fuel consumption by more than 25% compared to only the power split optimization on the original cycle (represents a human driver). Compared to sequential optimization, where the velocity profile and driver demanded power are smoothed out, it is still possible to achieve an additional 8% fuel consumption reduction with co-optimization.

A zoom-in comparison between the sequential and co-optimization is presented in Fig. 4.2b. From the second subplot, it can be observed that the operation of the vehicle with co-optimization can be characterized as *accel-glide-decel*. From the third and fourth subplots, the position constraints are satisfied in both cases.

4.4 Summary

This chapter first investigates the potential of a proposed sequential optimization of a velocity profile given traffic constraints and the corresponding charge depletion strategy of a PHEV. Afterward, an effective iterative and hierarchical scheme is presented to co-optimize the velocity and power-split operation of a PHEV. Specifically, fuel consumption is minimized directly in the presence of a preceding vehicle and under explicitly imposed velocity and position constraints. To address the challenges in obtaining a numerical solution to the co-optimization, a hybrid strategy is proposed that iterates over gradient projection and single shooting steps: (i) gradient projection is applied to obtain a feasible acceleration sequence and (ii) power-split of the PHEV is optimized over the velocity profile based on the acceleration sequence. Through a case study, the ability to solve co-optimization problems for relatively long drive cycles where the required energies exceed the available battery energies is demonstrated, critical for the PHEV applications. Furthermore, simulation results with various driving cycles show that an additional 4%-12% fuel consumption reduction can be achieved through co-optimization over a decentralized, sequential approach of acceleration smoothing followed by power-split optimization.

Up to this chapter, SOC is the only state considered in the powertrain dynamics. However, as will be detailed in the next chapter, instantaneous Hamiltonian minimization resulting from PMP could induce high-frequency switches in the control inputs, known as the *chattering behaviors*. Chattering behaviors are reflected in the (P)HEV power-split optimization problem as the busy engine on/offs. Addressing the chattering behaviors arising in the PMP-based single shooting motivates the development of a discrete-time mixed-integer shooting algorithm, which is the focus of the next chapter.

CHAPTER 5

Development of Discrete Mixed-Integer Shooting (DMIS) Algorithm

5.1 Computational Challenges in Engine Cranking and Co-Optimization

As discussed in Chapter 1 and the end of Chapter 4, systematically managing the engine on/off switching is one of the fundamental challenges in the optimal control of hybridized vehicles. For PHEVs, the energy management while avoiding frequent engine on/off switching through cranking fuel consideration is formulated as a mixed-integer type optimal control problem.

Solving mixed-integer optimal control problems numerically, however, remains challenging. Although commercial solvers like IBM-CPLEX [58] and Gurobi [59] have experienced rapid development in branch-and-cut/bound algorithms, they have rather limited applications to automotive control problems due to (i) the restricted cost functions and constraints forms and (ii) the exponential growth in the computation time as the problem horizon increases. Meanwhile, efforts have also been devoted to solving mixed-integer optimal control problems with direct and simultaneous methods [27, 60, 61], tailored to automotive applications. In [61], a mixed-integer nonlinear MPC of heavy-duty trucks with integer gear choices is tackled by first applying partial outer convexification and relaxation [27]. A nonlinear program with vanishing constraints is obtained by discretizing in time the modified problem. It is then solved with direct multiple shooting. Inequality constraints are enforced on each realization at each time instant to ensure feasibility with any rounding strategy.

Despite the successful application of the direct multiple shooting with a convex relaxation of the binary control variables to a real-world problem of controlling a heavy-duty truck, including predictive choice of gears [61, 60], it is less practical to adopt this direct and simultaneous approach for the optimal control of PHEVs. First of all, the fuel rate and motor efficiencies are tabular data (static maps) that are hard to be approximated accurately and efficiently by analytical expressions. The evaluation and approximation of the Hessian matrix in the corresponding NLP would introduce significant numerical errors and be very computationally intensive. Even though the BFGS formula [62] can be used to approximate Hessian based on gradient information, it is still computationally demanding due to the large underlying problem horizon.

Second, as observed when using single shooting to solve the TPBVP of the PHEV power-split optimization problem considered in previous chapters, several local minima already exist in the instantaneous Hamiltonian minimization problem. With a single shooting strategy, the control at each time instant is chosen to be the instantaneous Hamiltonian minimizer, which is of much lower dimension. Thus, it is possible to identify the potential local minima offline for efficient warm-start. By comparison, solving the NLP requires the decision of the entire control sequence at once. Properly warm-starting the control sequence as a whole, even for short-horizon problems to avoid local minima, becomes very difficult due to the combinatorial effect of the choice of an engine's initial operation point at every time instant. Moreover, the Hessian matrix of the NLP consists of elements of the instantiation of the Hamiltonians. As a result, it might be hard to escape the local minima determined by the warm-start strategy due to the local information provided by the Hessian matrix.

Third, the system dynamics are converted to equality constraints with multiple shooting. In between multiple shooting iterations, the control sequence could be inconsistent with the state sequence, violating primal feasibility. It is especially notable in our application due to static maps. The numerical errors with the Newton-type method would make the system's dynamic equations very hard to satisfy with acceptable accuracy. Finally, it is crucial to obtain the solution for the entire problem offline as the first step towards real-time online implementation with a limited problem horizon. As single percent fuel economy differences are significant at scale, it is valuable to understand the optimality gap of the "short-horizon" receding horizon control versus the full trajectory optimization result. A reasonable problem horizon for PHEVs is generally larger than one hour to necessitate the use of the engine and make the power-split optimization meaningful. Consequently, even with the approximated Newton-type method, the dimension of the resulting nonlinear program would become prohibitive for executing the iterations.

As observed from simulation results at the end of Chapter 2, the numerical solution obtained applying single shooting with the necessary condition for optimality is close to those obtained with DP. The sufficient optimality of the controls obtained this way and the complications when directly considering the NLP motivate the pursuit of an indirect single shooting-based numerical algorithm in this thesis.

5.2 A Discrete Maximum Principle (DMP)

As has been illustrated in Section 4.3.1.3, the co-state dynamics are unstable forward-in-time when the state dynamics are unstable, thus making the continuous-time PMP-based single shooting fragile. As shown later in this chapter, the co-state dynamics, when formulated in continuous time, are still unstable forward-in-time even though the state dynamics are stable. On the other hand, when the state dynamics are stable, the co-state dynamics become stable backward-in-time when formulated directly in discrete time with KKT conditions. When optimal control problems are formulated directly in discrete time, even when the state dynamics are initially unstable, they can be pre-stabilized, thus inducing stable co-state dynamics. As a result, formulating the optimal control problems directly in discrete time has good potential to avoid unstable state and co-state dynamics.

Motivated by the need to avoid unstable state and co-state dynamics in the TPBVP when formulated with continuous-time PMP (discussed in Section 5.5.1.2), and some practical control problems may not have trivial continuous-time re-formulations (for example, the problem discussed in Section 5.4), this chapter seeks to formulate the optimal control problems *directly in discrete-time*. A DMP[1] is then established where control can take mixed-integer values and functions involved are not necessarily differentiable, thus applicable to the PHEV optimal control problems considered in this thesis.

When the optimal control problems are formulated in discrete-time, KKT conditions are typically obtained as a necessary condition for a local optimum under suitable constraint qualification assumptions [63]. It requires differentiability of the objective function and constraints with respect to decision variables.

A maximum principle of the Pontryagin type (or DMP) for systems described by nonlinear difference equations can be obtained under certain smoothness and convexity requirement [64, 65] or relaxed directional convexity [66, 67] through separation of the reachable sets. A DMP can also be obtained from KKT conditions assuming certainty differentiability and convexity conditions on the state and control inputs [65, 68].

However, such requirements are typically strong in practice. With a DMP obtained from the separation of reachable sets [64, 65] (no particular assumptions on the control input), the convexity assumptions are always justified in the case of a system of nonlinear difference equations which approximates a system of nonlinear differential equations [64]. However, the twice-continuous differentiability assumption of the system dynamics and stage cost to the system state is still strong in practice in the presence of tabular data and max-type functions.

[1] In this thesis, the maximum principle and minimum principle are used interchangeably since the minimization problem can be converted to an equivalent maximization problem. However, they both refer to the minimization problem in this thesis.

The assumptions for the KKT conditions based DMP are also hard to be satisfied in practice. For example, in our application to the PHEV, the state cost (the fuel rate, a static map of engine torque and speed) and the SOC dynamics are not necessarily convex functions of control inputs. Moreover, with integer decision variables (the engine on/off decision), the above functions are not differentiable to the control inputs. Smoothing functions may be introduced; however, the choice is non-trivial and complicates design and calibration.

This section presents the efforts toward arriving at a DMP without assuming the differentiability on the control input u and state x. As a result, cases are also incorporated where specific control inputs can only take integer values, and non-differentiable max-type function can be included in the stage cost. Such a relaxation is critical to the PHEV applications since the engine on/off command is an integer-valued decision variable, and the engine cranking cost is formulated mathematically as a max-type function. A DMIS is then proposed to solve the TPBVP numerically. It should be pointed out that single shooting is rarely used when optimal control problems are directly formulated in discrete-time due to the asynchronous but co-dependent controls, states, and co-states. One of the few applications is when the system is linear, the stage cost is quadratic with an available closed-form solution for the control. A more welcoming approach is to convert the first-order necessary conditions to a generalized equation and solved them by Newton-type solvers [69].

The majority of the derivation presented in this section follows the line of strong Lagrange duality and saddle-point theorem as in [70, 71]. For a given nonlinear programming problem (primal problem), there is another nonlinear programming problem closely associated with it (Lagrangian dual problem). Under certain convexity assumptions and suitable constraint qualifications, the primal and dual problems have equal optimal objective values. As a by-product of one of the duality theorems, saddle point necessary optimality conditions without any differentiability assumptions can be established. The Lagrangian duality formulation has proved useful in discrete optimization where all or some of the variables are further restricted to be integers [63]. The strong Lagrange duality and saddle-point theorems for a constrained optimization problem considered in this section are based on two basic assumptions about the problem: (i) a convexity property in the image space of the constraints and (ii) a normality condition (constraint qualification) which rules out certain ill-behaved problems.

5.2.1 Duality and Saddle Point

First, the concept of duality and saddle point is revisited in this section for self-containment, according to [72]. Consider an optimization problem in the standard form

$$\begin{aligned} \text{minimize} \;\; & f_0(x) \\ \text{subject to} \;\; & f_i(x) \; \leq 0, \;\; i = 1, ..., m \\ & h_i(x) \; = 0, \;\; i = 1, ..., p, \end{aligned} \tag{5.1}$$

with variables $x \in \mathbb{R}^n$. Assume its domain $\mathcal{D} = \bigcap_{i=0}^{m} \textbf{dom}\ f_i \cap \bigcap_{i=1}^{p} \textbf{dom}\ h_i$ is nonempty, and denote the optimal value of (5.1) by p^*. There is no requirement of convexity on the problem.

The basic idea in Lagrangian duality is to take the constraints in (5.1) into account by augmenting the objective function with a weighted sum of the constraint functions. Define the *Lagrangian* $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \mapsto \mathbb{R}$ associated with the problem (5.1) as

$$L(x, \lambda, \upsilon) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \upsilon_i h_i(x), \tag{5.2}$$

with **dom** $L = \mathcal{D} \times \mathbb{R}^m \times \mathbb{R}^p$. Refer to λ_i as the *Lagrange multiplier* associated with the *i*th inequality constraint $f_i(x) \leq 0$; similarly refer to υ_i as the Lagrange multiplier associated with the *i*th equality constraint $h_i(x) = 0$. The vectors λ and υ are called the *dual variables* or *Lagrange multiplier vectors* associated with the problem (5.1).

Define the *Lagrange dual function* (or just *dual function*) $g : \mathbb{R}^m \times \mathbb{R}^p \mapsto \mathbb{R}$ as the minimum value of the Lagrangian over x: for $\lambda \in \mathbb{R}^m, \upsilon \in \mathbb{R}^p$,

$$g(\lambda, \upsilon) = \inf_{x \in \mathcal{D}} L(x, \lambda, \upsilon) = \inf_{x \in \mathcal{D}} (f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \upsilon_i h_i(x)). \tag{5.3}$$

When the Lagrangian is unbounded below in x, the dual function takes on the value $-\infty$. Since the dual function is the pointwise infimum of a family of affine functions of (λ, υ), it is concave, even when the problem (5.1) is not convex.

The dual function yields lower bounds on the optimal value p^* of the problem (5.1): For any $\lambda \geq 0$ and any υ

$$g(\lambda, \upsilon) \leq p^*. \tag{5.4}$$

Suppose $\tilde{x}$ is a feasible point for (5.1), i.e., $f_i(\tilde{x}) \leq 0$ and $h_i(\tilde{x}) = 0$, and $\lambda \geq 0$. Then

$$\sum_{i=1}^{m} \lambda_i f_i(\tilde{x}) + \sum_{i=1}^{p} \upsilon_i h_i(\tilde{x}) \leq 0, \tag{5.5}$$

and

$$L(\tilde{x}, \lambda, \upsilon) = f_0(\tilde{x}) + \sum_{i=1}^{m} \lambda_i f_i(\tilde{x}) + \sum_{i=1}^{p} \upsilon_i h_i(\tilde{x}) \leq f_0(\tilde{x}). \tag{5.6}$$

Hence

$$g(\lambda, \upsilon) = \inf_{x \in \mathcal{D}} L(x, \lambda, \upsilon) \leq L(\tilde{x}, \lambda, \upsilon) \leq f_0(\tilde{x}). \tag{5.7}$$

Since $g(\lambda, \upsilon) \leq f_0(\tilde{x})$ holds for every feasible point $\tilde{x}$, the inequality (5.4) follows. The dual function gives a nontrivial lower bound on p^* only when $\lambda \geq 0$ and $(\lambda, \upsilon) \in$ **dom** g, i.e., $g(\lambda, \upsilon) > -\infty$. A pair (λ, υ) with $\lambda \geq 0$ and $(\lambda, \upsilon) \in$ **dom** g is referred as *dual feasible*.

A pair $\tilde{w} \in W \subset \mathbb{R}^n$, $\tilde{z} \in Z \subset \mathbb{R}^m$ is referred as a saddle-point for f if

$$f(\tilde{w}, z) \leq f(\tilde{w}, \tilde{z}) \leq f(w, \tilde{z}) \tag{5.8}$$

for all $w \in W$ and $z \in Z$. In other words, $\tilde{w}$ minimizes $f(w, \tilde{z})$ (over $w \in W$) and $\tilde{z}$ maximizes $f(\tilde{w}, z)$ (over $z \in Z$):

$$f(\tilde{w}, \tilde{z}) = \inf_{w \in W} f(w, \tilde{z}), \;\; f(\tilde{w}, \tilde{z}) = \sup_{z \in Z} f(\tilde{w}, z). \tag{5.9}$$

5.2.2 Theorem of Alternative

Let $X \subset \mathbb{R}^n$, $Y \subset \mathbb{R}^m$ and $Z \subset \mathbb{R}^l$ be vector spaces with topological duals Y^* and Z^*, respectively. For a normed space V with dual space V^*, the value of a linear functional $w \in V^*$ at a point $v \in V$ is written as $\langle v, w \rangle$. Consider the optimization problem in the set X with $f : X \mapsto \mathbb{R}$, $g : X \mapsto Y$, and $h : X \mapsto Z$, find an element $x \in X$

$$\text{minimize } f(x) \tag{5.10a}$$

$$\text{s.t. } g(x) \leq 0 \tag{5.10b}$$

$$h(x) = 0 \tag{5.10c}$$

$$x \in X. \tag{5.10d}$$

Its Lagrangian function $L : X \times (Y^* \times Z^*) \mapsto \mathbb{R}$ is defined by

$$L(x; u, v) = f(x) + \langle g(x), u\rangle + \langle h(x), v\rangle, \tag{5.11}$$

for $x \in X$ and $(u, v) \in Y^* \times Z^*$. The dual functional $\gamma : Y^* \times Z^* \mapsto \mathbb{R}$ is defined by

$$\gamma(u, v) = \inf\{L(x; u, v), x \in X\}, \tag{5.12}$$

The Lagrangian dual optimization problem of the primal problem (5.10) is defined as

$$\max\{\gamma(u, v) : (u, v) \in P^* \times Z^*\}, \tag{5.13}$$

where $P^* := \{u | u \geq 0, u \in Y^*\}$.

Assumption 5-$\mathcal{A}$. (Convexity [70, 73, 74]) If $x_1 \in X, x_2 \in X$, and $\lambda \in [0, 1]$, then there is an element $x_3 \in X$ such that

$$f(x_3) \leq \lambda f(x_1) + (1 - \lambda)f(x_2) \tag{5.14}$$

$$g(x_3) \leq \lambda g(x_1) + (1 - \lambda)g(x_2) \tag{5.15}$$

$$h(x_3) = \lambda h(x_1) + (1 - \lambda)h(x_2). \tag{5.16}$$

Assumption 5-$\mathcal{B}$. (Existence of Interior Points [70]) Assume that:

1. the positive cone [75] P in the normed space Y has nonempty interior int P;

2. the image $h(X)$ under the mapping $h : X \mapsto Z$ in the normed space Z has nonempty interior int $h(X)$;

3. there is an open set $U \subset$ int $h(X)$ and a point $(M, y_M) \in \mathbb{R} \times Y$, such that for every $z \in U$, there is an $x \in X$ satisfying $f(x) \leq M$, $g(x) \leq y_M$, $h(x) = z$.

Define a multifunction $B : X \mapsto \mathbb{R} \times Y \times Z$ through

$$B(x) = \{(b, q, s) \in \mathbb{R} \times Y \times Z : f(x) \leq b, g(x) \leq q, h(x) = s\}, \tag{5.17}$$

then Assumption 5-$\mathcal{B}$ implies that the range of the multifunction B in $\mathbb{R} \times Y \times Z$ has nonempty interior.

Assumption 5-$\mathcal{C}$. (Constraint Qualification/ Normality Condition [70]) Assume that for every nonzero $(u, v) \in Y^* \times Z^*$ satisfying $u \geq 0, \exists x \in X$ such that $\langle g(x), u\rangle + \langle h(x), y\rangle < 0$.

Assumption 5-$\mathcal{C}$ guarantees the existence of a normal Lagrange multiplier by ruling out abnormal multipliers.

Theorem 5.2.1. (Eidelheit Separation Theorem [76]). Let K_l and K_2 be convex sets in X such that K_l has interior points and K_2 contains no interior point of K_1. Then there is a closed hyperplane H separating K_1 and K_2; i.e., there is an $x^* \in X^*$ such that $\sup_{x \in K_1}\langle x, x^*\rangle \le \inf_{x \in K_2}\langle x, x^*\rangle$. In other words, K_1 and K_2 lie in opposite half-spaces determined by H.

Definition 5.2.1. Inequality-Equality Systems.

System 1: There is an $x \in X$ such that $f(x) < 0, g(x) \le 0, h(x) = 0$.

System 2: There is a nonzero vector $(u_0, u, v) \in \mathbb{R} \times Y^* \times Z^*$ such that $u_0 \ge 0, u \ge 0$, and

$$f(x)u_0 + \langle g(x), u\rangle + \langle h(x), v\rangle \ge 0, \forall x \in X. \tag{5.18}$$

Proposition 5.2.1. (Theorem of Alternative [70]). If the convexity and interior point assumptions are satisfied by the functions f, g, h, then:

1. if System 1 has no solution x, it follows that System 2 has a solution (x_0, u, v);
2. if System 2 has a solution (u_0, u, v) and $u_0 > 0$, then system 1 has no solution.

Proof. Define the set

$$A = \{(b, q, s) \in \mathbb{R} \times Y \times Z : \exists x \in X, s.t.\ \ f(x) < b, g(x) \le q, h(x) = s\}. \tag{5.19}$$

Then System 1 has no solution iff $\theta \notin A$, where θ is the zero vector in $\mathbb{R} \times Y \times Z$. Assumption 5-$\mathcal{A}$ concerning convexity implies that A is convex, and Assumption 5-$\mathcal{B}$ implies the set A is not empty since

$$\text{int (cl } A) = \text{int } A \neq \emptyset. \tag{5.20}$$

It follows from Theorem 5.2.1 that ($K_1 := A$ and $K_2 = \{\theta\}$, and int $K_1 \cap K_2 = \emptyset$) there exists a nonzero vector $(\tilde{u}_0, \tilde{u}, \tilde{v}) \in \mathbb{R} \times Y^* \times Z^*$ such that

$$\sup_{(b,q,s) \in \text{cl } A} \le \inf\langle \theta, (\tilde{u}_0, \tilde{u}, \tilde{v})\rangle = 0. \tag{5.21}$$

Define $(u_0, u, v) := (-\tilde{u}_0, -\tilde{u}, -\tilde{v}) \in \mathbb{R} \times Y^* \times Z^*$, (5.21) is equivalent to the existence of a nonzero vector $(u_0, u, v) \in \mathbb{R} \times Y^* \times Z^*$ such that for every $(b, q, s) \in$ cl A

$$bu_0 + \langle q, u\rangle + \langle s, v\rangle \ge 0. \tag{5.22}$$

Let x be any point in X, and define

$$\bar{b} = f(x) + \epsilon + \xi, \;\; \bar{q} = g(x) + \eta, \;\; \bar{s} = h(x), \tag{5.23}$$

where $\epsilon > 0, \xi \geq 0, \eta \in P \subset Y$. Then for any $(\xi, \eta, 0), (\bar{b}, \bar{q}, \bar{s}) \in A$, it follows from (5.22) that $u_0 \geq 0$ and $u \geq 0$ since

$$\xi u_0 + \langle \eta, u \rangle \geq 0. \tag{5.24}$$

For every $x \in X$,

$$(f(x), g(x), h(x)) \in \text{cl } A, \tag{5.25}$$

and it follows from (5.22) that

$$f(x)u_0 + \langle g(x), u \rangle + \langle h(x), v \rangle \geq 0, \forall x \in X. \tag{5.26}$$

To prove the second claim, assume that $u_0 > 0, u \geq 0$, and $x \in X$ satisfies $g(x) \leq 0, h(x) = 0$. Then

$$\langle g(x), u \rangle \leq 0, \quad \langle h(x), v \rangle = 0, \forall v. \tag{5.27}$$

It follows from (5.18) that

$$f(x)u_0 \geq f(x)u_0 + \langle g(x), u \rangle + \langle h(x), v \rangle \geq 0, \tag{5.28}$$

and because $u_0 > 0$, it can be concluded that $f(x) \geq 0, \forall x \in X$ satisfying $g(x) \leq 0$ and $h(x) = 0$, which proves the second part. □

The existence of Lagrange multipliers follows from the theorem of the alternative by offsetting $f(x)$ by its optimal value.

Theorem 5.2.2. Assume that (i) $\bar{x} \in X$ is an optimal solution to the primal optimization problem (5.10), (ii) the convexity Assumption 5-$\mathcal{A}$ is satisfied, and (iii) the interior point Assumption 5-$\mathcal{B}$ is satisfied, or the spaces Y and Z are finite dimensional. Then there is a nonzero vector $(u_0, u, v) \in \mathbb{R} \times Y^* \times Z^*$, and a real-valued function F on X, defined by

$$F(x; u_0, u, v) = f(x)u_0 + \langle g(x), u \rangle + \langle h(x), v \rangle, x \in X, \tag{5.29}$$

and it satisfies

$$F(\bar{x}; u_0, u, v) \leq F(x; u_0, u, v), \forall x \in X \tag{5.30}$$

$$u_0 \geq 0, u \geq 0, \tag{5.31}$$

$$\langle g(\bar{x}), u \rangle = 0. \tag{5.32}$$

Proof. Let $c = f(\bar{x})$. Then there is no $x \in X$ such that

$$f(x) - c \leq 0, \quad g(x) \leq 0, \quad h(x) = 0. \tag{5.33}$$

It can be seen that (5.33) is the System 1 under Definition 5.2.1. Under the Assumptions 5-$\mathcal{A}$, 5-$\mathcal{B}$, per Theorem of Alternative there exists a nonzero vector (u_0, u, v) satisfying (5.31) and

$$(f(x) - c)u_0 + \langle g(x), u \rangle + \langle h(x), v \rangle \geq 0, \forall x \in X. \tag{5.34}$$

(5.32) is a direct result by applying (5.34) and $f(\bar{x}) = 0, g(\bar{x}) \leq 0, h(\bar{x}) = 0$ for $\bar{x} \in X$. (5.30) can be obtained from (5.34). Finally, note that for $u_0 \neq 0$, $F(x; u_0, u, v) \equiv u_0 L(x; u/u_0, v/u_0)$. □

Theorem 5.2.3. (Lagrange Multiplier Rule [70]). Assume that the conditions (i)-(iii) of Theorem 5.2.2 are satisfied and that the constraint qualification (Assumption 5-$\mathcal{C}$) is satisfied. Then there exists a normal Lagrange multiplier $(\tilde{u}, \tilde{v}) \in P^* \times Z^*$ for the constraints of the primal problem, such that

$$L(\tilde{x}, \tilde{u}, \tilde{v}) \leq L(x, \tilde{u}, \tilde{v}), \forall x \in X \tag{5.35}$$

$$\tilde{u} \geq 0 \tag{5.36}$$

$$\langle g(\tilde{x}), \tilde{u} \rangle = 0. \tag{5.37}$$

Proof. Theorem 5.2.2 implies the existence of a nonzero vector (u_0, u, v) and a function F satisfying (5.30)-(5.32). Assume $u_0 = 0$, then it follows that

$$\langle g(x), u \rangle + \langle h(x), v \rangle \geq 0, \forall x \in X, \tag{5.38}$$

where $(u, v) \neq 0, u \geq 0$. This violates Assumption 5-$\mathcal{C}$. Consequently, $u_0 > 0$, and F defines a Lagragian function

$$L(x; \bar{u}, \bar{v}) = (1/u_0)F(x; u_0, u, v), \tag{5.39}$$

where $\bar{u} = (1/u_0)u$ and $\bar{v} = (1/u_0)v$. □

The saddle-point theorem presented below is a direct application of Theorem 5.2.2 and 5.2.3.

Theorem 5.2.4. Assume that (i) $\bar{x} \in X$ is an optimal solution to the primal optimization problem, (ii) the convexity Assumption 5-$\mathcal{A}$, (iii) the interior point Assumption 5-$\mathcal{B}$ or the space Y and Z are finite-dimensional, and (iv) the constraint qualification Assumption 5-$\mathcal{C}$ is satisfied. Then, there exists a Lagrange multiplier $(\bar{u}, \bar{v}) \in P^* \times Z^*$ such that $(\bar{x}; \bar{u}; \bar{v})$ is a saddle point of the Lagrangian function of the problem on $X \times (P^* \times Z^*)$.

5.2.3 A DMP from Lagrange Duality with Application to (Mixed-Integer) Optimal Control Problem

Consider the general (mixed-integer) optimal control problem (*The primal problem* (**P**)) of minimizing[2]

$$J := J(X, U) = \phi(x_N) + \sum_{k=0}^{N-1} \tilde{l}(x_k, u_k), \tag{5.40a}$$

with the state $x_k \in \mathbb{R}^n$ and the control input $u_k \in \mathbb{R}^m$ at the time $t = k$ (certain control inputs could be restricted to only integer values as will be refined in the following). The state sequence is defined as $X = (x_0 \ldots x_N) \in \mathbb{R}^{n\cdot(N+1)}$, and the control sequence is defined as $U = (u_0 \ldots u_{N-1}) \in \mathbb{R}^{m\cdot N}$. In (5.40a) $\phi : \mathbb{R}^n \mapsto \mathbb{R}$ is the terminal cost, and $\tilde{l} : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}$ is the augmented stage cost. (5.40a) is subject to the initial condition

$$x_0 = q_0, \tag{5.40b}$$

the state dynamics

$$x_{k+1} = f(x_k, u_k), \tag{5.40c}$$

and the control input $u = (u^1, u^2)$ consists of discrete part u^1 and continuous part u^2, with constraints $u_k \in \mathcal{U}_k$ defined as

$$u_k^1 \in \mathbb{Z}^{m_1} \cap \Omega_k^1, \quad u_k^2 \times \Omega_k^2, \tag{5.40d}$$

[2] For simplicity, the case with terminal cost rather than terminal state equality constraint is presented here without loss of generality.

where $\Omega_k^1 \subset \mathbb{R}^{m_1}, \Omega_k^2 \subset \mathbb{R}^{m-m_1}$, and the (mixed state and) control constraints

$$(x_k, u_k) \in \mathcal{W}_k, \text{ expressed in the form of } g(x_k, u_k) \leq 0, \tag{5.40e}$$

where $g : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^p$. In particular, note that although pure state constraints are not presented explicitly in the (5.40), they are actually augmented to the stage cost $\tilde{l}(x, u)$ as (convex) penalties measuring the constraint violation. Following the discussion in [77],

$$\tilde{l}(x_k, u_k) = l(x_k, u_k) + \rho(d(x_k, \mathcal{X}_k)), \tag{5.40f}$$

where $d(x_k, \mathcal{X}_k)$ is a metric function describing the distance of the state x_k from the feasible region $\mathcal{X}_k$, and $\rho(\cdot)$ is a monotonically non-decreasing penalty function such that $\rho(0) = 0$, and grows quickly outside $\mathcal{X}_k$.

Note that for the primal problem (5.40), it is only required for $\phi, \tilde{l}$, and f to be locally Lipschitz continuous in x. No differentiability is assumed since non-smooth terms can appear in either the stage cost l, the terminal cost ϕ, or the dynamics f (for example, static maps are involved in f with some industrial applications).

Define $P = (p_0 \ldots p_N) \in \mathbb{R}^{n\cdot(N+1)}$ with $p_k \in \mathbb{R}^n$, and $\Upsilon = (\upsilon_0 \ldots \upsilon_{N-1}) \in \mathbb{R}^{p\cdot N}$ with $\upsilon_k \in \mathbb{R}^p$. The Lagrangian function $L : \mathbb{R}^{n\cdot(N+1)} \times \mathbb{R}^{m\cdot N} \times \mathbb{R}^{n(N+1)} \times \mathbb{R}^{p\cdot N} \mapsto \mathbb{R}$ for the primal problem (5.40) is defined as

$$\begin{aligned} L(X, U, P, \Upsilon) =& J + p_0^T(q_0 - x_0) + \sum_{k=0}^{N-1} p_{k+1}^T(f(x_k, u_k) - x_{k+1}) + \upsilon_k^T g(x_k, u_k) \\ =& \phi(x_N) + \sum_{k=0}^{N-1} \{\tilde{l}(x_k, u_k) + p_{k+1}^T f(x_k, u_k) + \upsilon_k^T g(x_k, u_k)\} - \sum_{k=0}^{N} p_k^T x_k + p_0^T q_0. \end{aligned} \tag{5.41}$$

The dual function $\Phi : \mathbb{R}^{n\cdot(N+1)} \times \mathbb{R}^{p\cdot N} \mapsto \mathbb{R} \cup \{-\infty\}$ is defined as

$$\Phi(P, \Upsilon) = \inf_{X,U} \{L(X, U, P, \Upsilon)\}. \tag{5.42a}$$

The Lagrange dual optimization problem is defined as

$$\text{maximize } \Phi(P, \Upsilon), \tag{5.42b}$$

$$\text{subject to } \Upsilon \geq 0. \tag{5.42c}$$

Any (X, U) satisfying the constraints (5.40c) – (5.40e), together with $x_0 = q_0$, and $d(x_k, \mathcal{X}_k)$ finite, $\forall k \in \{1, \ldots, N\}$ finite of the primal problem is called primal feasible, and any (P, Υ)

satisfying the constraint (5.42c) of the dual problem is dual feasible. In the sequel, denote the set of (X, U) satisfying primal feasibility as $\mathcal{X} \times \mathcal{U} \subset \mathbb{R}^{n\cdot(N+1)} \times \mathbb{R}^{m\cdot N}$.

Theorem 5.2.5. (Weak and strong duality [63, 71, 72]). If $(\hat{X}, \hat{U})$ is primal feasible, and $(\hat{P}, \hat{\Upsilon})$ is dual feasible, then $\Phi(\hat{P}, \hat{\Upsilon}) \leq J(\hat{X}, \hat{U})$. Further, if $\Phi(\hat{P}, \hat{\Upsilon}) = J(\hat{X}, \hat{U})$, then $(\hat{X}, \hat{U})$ and $(\hat{P}, \hat{\Upsilon})$ are optimal solutions to the primal and dual problems, respectively.

Theorem 5.2.5 states that any dual feasible vector provides a lower bound for the primal optimum, and any feasible vector an upper bound for the dual optimum [63, 71]. The strong duality condition $\Phi(\hat{P}, \hat{\Upsilon}) = J(\hat{X}, \hat{U})$ is sufficient for the primal and dual optimality.

Theorem 5.2.6. (Conditions equivalent to strong duality [63, 71]). Let $(\hat{X}, \hat{U})$ be primal feasible and $(\hat{P}, \hat{\Upsilon})$ dual feasible. Then the strong duality condition $\Phi(\hat{P}, \hat{\Upsilon}) = J(\hat{X}, \hat{U})$ is satisfied if and only if the following conditions are satisfied: $\inf_{X,U}\{L(X, U, \hat{P}, \hat{\Upsilon})\}$ is obtained at $(\hat{X}, \hat{U})$, and $\sum_{k=0}^{N-1} \hat{\upsilon}_k^T g(\hat{x}_k, \hat{u}_k) = 0$.

Theorem 5.2.6 for feasible primal and dual vectors $(\hat{X}, \hat{U})$ and $(\hat{P}, \hat{\Upsilon})$, minimization of the Lagrangian at $(\hat{X}, \hat{U})$ and the condition of the complementary slackness $\sum_{k=0}^{N-1} \hat{\upsilon}_k^T g(\hat{x}_k, \hat{u}_k) = 0$, are necessary and sufficient condition for the strong duality and therefore the primal and dual optimality of $(\hat{X}, \hat{U})$ and $(\hat{P}, \hat{\Upsilon})$. Consequently, the conditions under which a Lagrange multiplier $(\hat{P}, \hat{\Upsilon})$ together with a primal optimal solution $(\hat{X}, \hat{U})$ satisfy the strong duality condition is of great interest. As is shown in [71], the key ingredients are the generalized convexity and normality properties that will be summarized in the remaining section.

Condition 5.2.1. (Generalized convexity [71, 70]). For every (X_1, U_1), (X_2, U_2) and $\kappa \in [0, 1]$, there is an element (X_3, U_3), such that

$$J(X_3, U_3) \leq \kappa J(X_1, U_1) + (1-\kappa) J(X_2, U_2), \tag{5.43a}$$

$$F(X_3, U_3) = \kappa F(X_1, U_1) + (1-\kappa) F(X_2, U_2), \tag{5.43b}$$

where $F(X, U) = \big((q_0 - x_0),\ (f(x_0, u_0) - x_1),\ \dots, (f(x_{N-1}, u_{N-1}) - x_N)\big)$.

$$G(X_3, U_3) \leq \kappa G(X_1, U_1) + (1-\kappa) G(X_2, U_2), \tag{5.43c}$$

where $G(X, U) = \big(g(x_0, u_0),\ \dots, g(x_{N-1}, u_{N-1})\big)$.

Condition 5.2.2. (Sufficient condition for the convexity Condition 5.2.1[71]). For every $t \in \{0 \cdots\ N-1\}$, for every $x_1, x_2 \in \mathbb{R}^n$, for every $u_1, u_2 \in \mathcal{U}_k$, and for every $\kappa \in [0, 1]$, there is a

$u_3 \in \mathcal{U}_k$ such that:

$$\tilde{l}_k(x_3, u_3) \leq \kappa \tilde{l}_k(x_1, u_1) + (1-\kappa)\tilde{l}_k(x_2, u_2), \tag{5.44}$$

$$f(x_3, u_3) = \kappa f(x_1, u_1) + (1-\kappa) f(x_2, u_2), \tag{5.45}$$

$$g(x_3, u_3) \leq \kappa g(x_1, u_1) + (1-\kappa) g(x_2, u_2), \tag{5.46}$$

where $x_3 = \kappa x_1 + (1-\kappa)x_2$, and for $t = N$, the terminal cost ϕ is a convex function on $\mathbb{R}^n$.

As discussed in [71], a sufficient condition for (5.43b) in the form of (5.44) is related to the system controllability and can in general be satisfied. (5.46) in Condition 5.2.2 can be satisfied since the control constraints usually have an affine type.

Condition 5.2.3. (Normality). For every nonzero (P, Υ) there is (X, U) such that

$$p_0^T(q_0 - x_0) + \sum_{k=0}^{N-1} p_{k+1}^T(f(x_k, u_k) - x_{k+1}) + \upsilon_k^T g(x_k, u_k) < 0. \tag{5.47}$$

Theorem 5.2.7. (Necessary and sufficient conditions for optimality in the case of strong Lagrange duality [71]). Assume convexity (5.2.1) and normality (5.2.3). A primal feasible tuple (X^*, U^*) is optimal if and only if there exists a Lagrange multiplier vector (P^*, Υ^*) such that the following conditions are satisfied:

$$\min L(X, U, P^*, \Upsilon^*) \tag{5.48a}$$

is attained at the point (X^*, U^*),

$$\Upsilon^* \geq 0, \tag{5.48b}$$

and

$$\sum_{k=0}^{N-1} (\upsilon_k^*)^T g(x_k^*, u_k^*) = 0. \tag{5.48c}$$

The discrete maximum principle is a direct consequence of the theorem 5.2.7.

Theorem 5.2.8. (The discrete maximum principle of Pontryagin without differentiability [71]). Assume that convexity Condition 5.2.1 and normality Condition 5.2.3 are satisfied. If (X^*, U^*) is an optimal solution to the primal problem (5.40), then a Lagrange multiplier vector (P^*, Υ^*)

exists such that

$$\min\{L(X^*, U, P^*, \Upsilon^*)\} \tag{5.49a}$$

is obtained at U^*, and

$$\min\{L(X, U^*, P^*, \Upsilon^*)\} \tag{5.49b}$$

is obtained at X^*. (5.49b) is equivalent to the condition:

$$0 \in \frac{\partial L(X, U^*, P^*, \Upsilon^*)}{\partial X}|_{X^*}, \tag{5.49c}$$

where $\frac{\partial L(\cdot)}{\partial X}$ denotes the generalized gradient (subgradient here) of the Lagrangian, and the conditions of dual feasibility and complementary slackness are satisfied:

$$\Upsilon^* \geq 0, \quad \sum_{k=0}^{N-1} (\upsilon_k^*)^T g(x_k^*, u_k^*) = 0. \tag{5.49d}$$

Denote the Hamiltonian H_k

$$H_k := H_k(x_k, u_k, p_{k+1}, \upsilon_k) = \tilde{l}(x_k, u_k) + p_{k+1}^T f(x_k, u_k) + \upsilon_k^T g(x_k, u_k), \tag{5.50}$$

then in Theorem 5.2.8, (5.49a) is equivalent to

$$\begin{aligned} u_k^* &= \arg\min_{u \in \mathcal{U}_k} \{\tilde{l}(x_k^*, u) + (p_{k+1}^*)^T f(x_k^*, u) + (\upsilon_k^*)^T g(x_k, u)\} \\ &= \arg\min_{u \in \mathcal{U}_k} \{H_k(x_k^*, u, p_{k+1}^*, \upsilon_k^*)\}, \end{aligned} \tag{5.51}$$

where the case when some of the control inputs are restricted to take only integer values can be treated no differently in the instantaneous Hamiltonian minimization.

The condition (5.49c) is equivalent to the co-state dynamics

$$p_k^* \in \frac{\partial H_k(x_k, u_k^*, p_{k+1}^*, \upsilon_k^*)}{\partial x_k}|_{x_k^*}, \quad p_N^* \in \frac{\partial \phi(x_N)}{\partial x_N}|_{x_N^*} \tag{5.52}$$

where $\frac{\partial H_k(x_k, u_k, p_{k+1}, \upsilon_k)}{\partial x_k}$ and $\frac{\partial \phi(x_N)}{\partial x_N}$ denote the generalized gradient [78] of the Hamiltonian H and terminal cost ϕ at x, respectively.

Remark 3. Note that although in (5.52) the co-state dynamics involve generalized gradients, in the numerical implementation, finite difference method is used to evaluate an element from their generalized gradients [78]. Consequently, equality ($=$) replacing the set inclusion ($\in$) is used in (5.52) as well as in the sequel for simplicity. In addition, to avoid the additional Lagrange multiplier v_k associated with the mixed state and control constraints $g(x_k, u_k)$ in the single shooting-based numerical algorithms introduced in the following sections, $g(x_k, u_k)$ will be removed. Instead, the mixed state and control constraint violation will be augmented to the stage cost through a proper penalty when formulating the optimal control problems in the remaining thesis.

5.2.4 Generalized Convexity and Normality Conditions for the PHEV Application

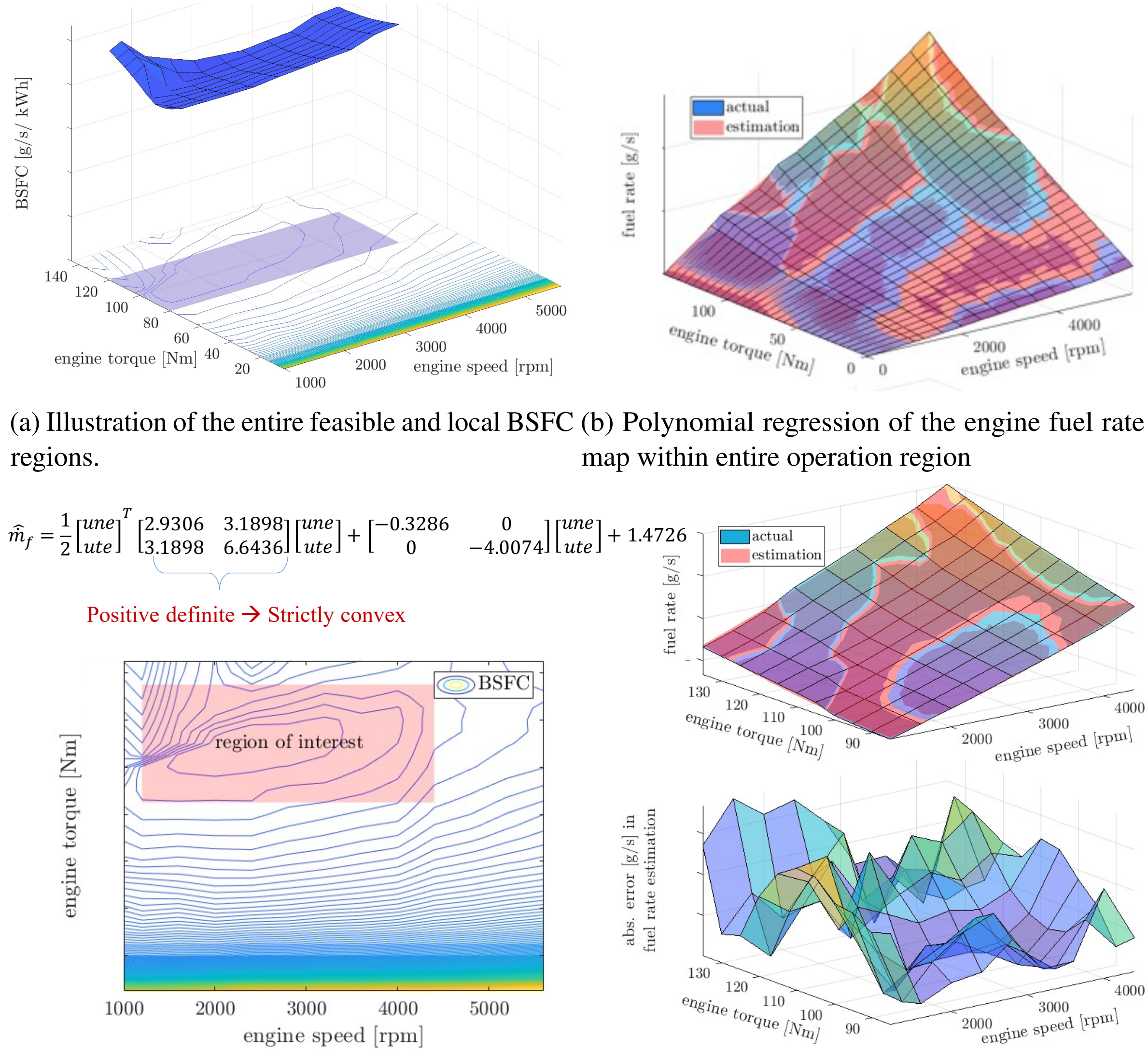
For the DMP to be applicable to the minimum fuel consumption problem of a PHEV considered in this thesis, it remains to check whether the generalized convexity Condition 5.2.1 and the normality Condition 5.2.3 are satisfied. As a sufficient condition for the convexity property 5.2.1, (5.44) is satisfied in the PHEV application since the penalty function (in the co-optimization) is convex in state and the fuel rate (stage cost) is locally a convex map with respect to the engine torque and speed, and it is always possible to choose $u_3 = 0$ (EV) mode to satisfy (5.44).

Linear regression is used to approximate the empirical fuel rate map with third-order polynomials, from which the Hessian can be calculated.

$$\dot{\hat{m}}_f = \sum_{i\in\{0,1,2,3\}, j\in\{0,1,2,3\}} \alpha_{i,j} \times une^i \times ute^j, \tag{5.53}$$

where une and ute are the normalized engine speed and torque. Figure 5.1a and Fig. 5.1b present the regression models of the entire fuel rate map. Unfortunately, the Hessian matrix of the fuel rate $H\dot{\hat{m}}_f$ has saddle point most of the times.

However, it is not necessary to consider the entire operation region. Under the minimum fuel consumption objective, the engine will always be commanded to operate within the lowest engine BSFC region, as marked as the shadowed region in Fig. 5.1a. Within the low BSFC region, the fuel rate map can be approximated with second-order polynomials, as shown in Fig. 5.1c. Admit that the engine has saddle point globally, but locally the region in consideration is convex.



(a) Illustration of the entire feasible and local BSFC regions.

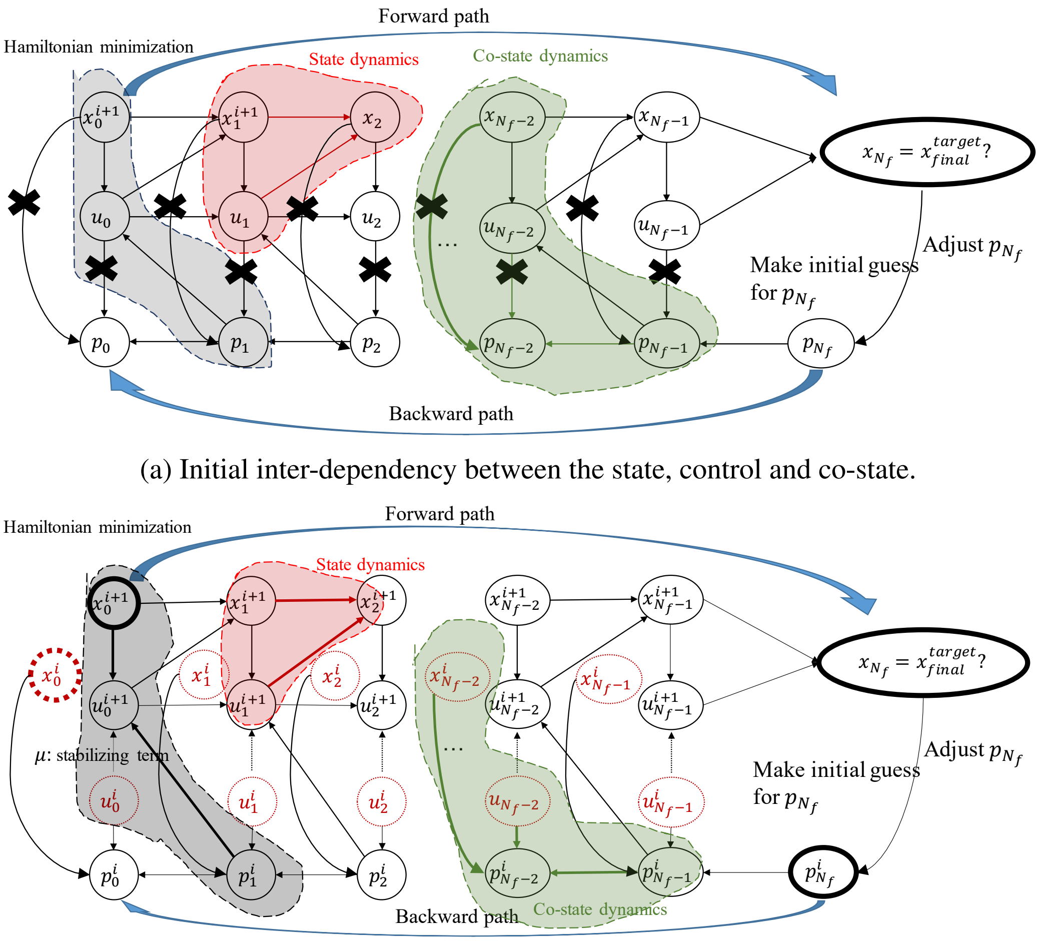
(b) Polynomial regression of the engine fuel rate map within entire operation region

(c) Polynomial regression of the engine fuel rate map within the low BSFC region.

Figure 5.1: Polynomial regression of the engine fuel rate map

5.3 Description of the DMIS Algorithm

In this section, a Discrete Mixed-Integer Shooting (DMIS) algorithm is presented as a modification of the single algorithm with co-state backward-in-time propagation [5]. The DMIS is summarized in Algorithm 2 and pictorially presented in Fig. 5.2. It serves as the numerical strategy to solve the TPBVP ((5.40b), (5.40c), (5.52) where the set inclusion will be replaced with equality in numerical evaluation, and the terminal co-state p_N^* could also be determined when the terminal cost function $\phi(x_N)$ is replaced by terminal equality and inequality constraints) obtained when applying DMP to the problem (**P**), but in consideration of additional terminal state equality constraints and integer-valued decision variables. Algorithm 2



(a) Initial inter-dependency between the state, control and co-state.

(b) Breaking the inter-dependency between the state, control and co-state.

Figure 5.2: Essential idea behind the DMIS.

is then benchmarked against a start-of-art multiple shooting algorithm and a mixed-integer solver on two optimization problems with continuous and discrete control inputs in Section 5.4. Finally, it is applied to solve the PHEV minimum fuel consumption problem with an additional engine cranking state (5.13, 5.14) numerically in Section 5.5.

The essential idea behind the DMIS algorithm is illustrated in Fig. 5.2. As seen from (5.50), the Hamiltonian at $t = k$ depends on the value of the co-state at $t = k + 1$, p_{k+1}. The dynamics of the co-state are defined backward-in-time and depend on the state x_k and control u_k at time $t = k$ as shown in (5.52). The issue is that the state and control trajectories are not known a priori. Since the system dynamics and cost function are nonlinear, no closed-form solution is available to break the loop. The authors proposed an algorithm [5] utilizing an initial guess of the control trajectories and a modified Hamiltonian by a penalty term enabling the successive iterations to be stabilized to resolve this issue. Note that in [5], only terminal state inequality constraints are considered. As a result, the terminal co-state is 0, and no shooting

Algorithm 2 Discrete Mixed-Integer Shooting (DMIS) (a modified version of [5])

```
Parameters: $\epsilon_u, \epsilon_J, \epsilon_x, d\mu, \gamma, i_{\max}$
Input: $u^0, x_0, x_f, p_N^0, \mu^0$
Output: $u^i, x^i, p^i, J^i$
$i \leftarrow 0$, $u^i \leftarrow$ any feasible initial control trajectory $u^0$
$x^i$, $J^i \leftarrow$ simulate the system forward with $u^i, x_0$
$\mu^i \leftarrow$ initial choice of $\mu^0$
$m \leftarrow 0, p_N^m \leftarrow$ initial guess of the terminal co-state $p_N^0$
while $i <$ inf do
    while $||x_N^i - x_f|| \geq \epsilon_x$ do
        $p^m \leftarrow$ co-state backward propagation with $x^i, u^i, p_N^m$
        $x^{i+1}, u^{i+1}, J \leftarrow$ simulate the system forward with $p^m$,
    where $u_k^{i+1} = \arg\min_u [H_k^{i+1} + \mu^i ||u - u_k^i||^2]$,
    and $H_k^{i+1} = l_k(x_k^{i+1}, u) + (p_{k+1}^m)^T x_{k+1}^{i+1}$
        $p_N^{m+1} = p_N^m + \Delta p_N^m(x_N^{i+1}, x_f)$
        $m \leftarrow m + 1, p_N^m \leftarrow p_N^{m+1}$
    end
    if $J > J^i - \epsilon_J$ and $||u^i - u^{i-1}|| > \epsilon_u$ then
        $\mu^i = \max(\mu^i + d\mu, \gamma\mu^i), m \leftarrow 0$
    else if $J > J^i - \epsilon_J$ then
        $\delta u^i = u^{i+1} - u^i$, $\mu^{i+1} = \max(0, \min(\mu^i - d\mu, \mu^i/\gamma))$
        $m \leftarrow 0, i \leftarrow i + 1$, $\mu^i \leftarrow \mu^{i+1}$, $J^i \leftarrow J$
    else
        $\delta u^i = u^{i+1} - u^i$
        $m \leftarrow 0, i \leftarrow i + 1$, $\mu^i \leftarrow \mu^{i+1}$, $J^i \leftarrow J$
    if $||\delta u^i|| \leq \epsilon_u$ and $i \geq i_{\max}$ then
        break
end
```

iterations are needed for the TPBVP. However, for the PHEV fuel-optimal control application considered in this thesis, terminal state equality constraints are considered (terminal SOC requirement). Besides, the original work [5] assumes twice continuously differentiability of the system dynamics and the state cost, which is relaxed in this thesis to incorporate cases with max-type terms (cranking cost in (5.11)).

In Algorithm 2, the iterations for the inner-loop and outer-loop are denoted as m and i, respectively. The inputs to Algorithm 2 are an initial feasible control trajectory u^0, the initial and terminal state x_0 and x_f, the initial guess of the terminal co-state p_N^0, and the initial choice of the penalty weight μ^0. The outputs from Algorithm 2 are the resulting control, state, co-state trajectories u^i, x^i, p^i, respectively, and the value of the cost function J^i when exiting from the current iteration i. Besides, parameters for this algorithm need to be selected in advance, including the thresholds of the change in the control, state and cost function, ϵ_u, ϵ_x and ϵ_J, respectively; scaling factors for the penalty weight $d\mu$ and γ, respectively; the maximum number of outer iterations $i_{\max}$. In **Step 8**, μ^i is the penalty weight, and $\mu^i||u - u_k^i||^2$ is a penalty term that enables the successive iterations to be stabilized: when μ^i is high, u will be different from u_k^i only when it dramatically decreases H [5].

Note that (i) the penalty weight μ is essential for stabilization and convergence of the DMIS iterations, as can be seen in [5]. (ii) In **Step 8**, the original Hamiltonian is recovered when $\mu^i||u - u_k^i||^2 = 0$.

5.3.1 Behavior of Successive Cost Function Values

In the original paper [5], it is shown that the values of the successive cost functions can always be decreased if the penalty weight μ takes sufficient high values. However, the results are based on continuous-time formulation followed by discretization. Here the results are showed to hold with the direct discrete-time formulation and supplement the proof to include the discrete-valued control.

The control input u here includes *continuous* and *discrete* control variables $u = [v, \alpha]$, and the system's dynamics in discrete-time can be re-written as

$$x_{k+1} = \sum_{q \in Q} \alpha_q f_{p,k}(x_k, v_k), \tag{5.54a}$$

where $\forall q \in Q$, $\alpha_q = 1$ if q is the *active* configuration and $\alpha_q = 0$ otherwise. More precisely,

$$\forall q \in Q, \;\; \alpha_q = \{0, 1\}, \;\; \text{and} \sum_{q \in Q} \alpha_q = 1. \tag{5.54b}$$

In (5.54a), $f_{p,k}(x_k, v_k)$ is locally Lipschitz-continuous in $x, v, \forall p$.

Assumption 5-$\mathcal{D}$. $H(x, u, p)$ is *strongly semismooth* in x.

The strongly semismoothness in Assumption 5-$\mathcal{D}$ is reasonable in the presence of either $\max$ type function or static maps. Denote $\delta x_k^i = (x_k^{i+1} - x_k^i)$, from [79]

$$H_k(x_k^{i+1}, u_k^i, p_{k+1}^i) - H_k(x_k^i, u_k^i, p_{k+1}^i) \leq (\frac{\partial H_k(x_k^i, u_k^i, p_{k+1}^i)}{\partial x_k})^T \delta x_k^i + \frac{L}{2}||\delta x_k^i||^2. \tag{5.55}$$

In between two successive shooting iterations,

$$J(u^{i+1}) - J(u^i) \;= \sum_{k=0}^{N-1} [L_k(x_k^{i+1}, u_k^{i+1}) - L_k(x_k^i, u_k^i)], \tag{5.56a}$$

which can be rewritten with the definition of the Hamiltonian

$$\begin{aligned} J(u^{i+1}) =& J(u^i) + \sum_{k=0}^{N-1} \big[H_k(x_k^{i+1}, u_k^{i+1}, p_{k+1}^i) - H_k(x_k^{i+1}, u_k^i, p_{k+1}^i) \\ & + H_k(x_k^{i+1}, u_k^i, p_{k+1}^i) - H_k(x_k^i, u_k^i, p_{k+1}^i) - (p_{k+1}^i)^T (f_k(x_k^{i+1}, u_k^{i+1}) - f_k(x_k^i, u_k^i))\big] \end{aligned} \tag{5.56b}$$

According to the approximate Hamiltonian minimization, denote $\delta u_k^i = u_k^{i+1} - u_k^i$, then

$$H_k(x_k^{i+1}, u_k^{i+1}, p_{k+1}^i) - H_k(x_k^{i+1}, u_k^i, p_{k+1}^i) \leq -\mu^i ||\delta u_k^i||^2. \tag{5.56c}$$

As a result,

$$\begin{aligned} J(u^{i+1}) \leq& J(u^i) + \sum_{k=0}^{N-1} [-\mu^i ||\delta u_k^i||^2 + \frac{L}{2}||\delta x_k^i||^2 + (\frac{\partial H_k(x_k^i, u_k^i, p_{k+1}^i)}{\partial x_k})^T \delta x_k^i \\ & - (p_{k+1}^i)^T (f_k(x_k^{i+1}, u_k^{i+1}) - f_k(x_k^i, u_k^i))] \\ =& J(u^i) + \sum_{k=0}^{N-1} [-\mu^i ||\delta u_k^i||^2 + \frac{L}{2}||\delta x_k^i||^2 + (p_k^i)^T \delta x_k^i - (p_{k+1}^i)^T \delta x_{k+1}^i] \\ =& J(u^i) + (p_0^i)^T \delta x_0^i - (p_N^i)^T \delta x_N^i + \sum_{k=0}^{N-1} [-\mu^i ||\delta u_k^i||^2 + \frac{L}{2}||\delta x_k^i||^2] \end{aligned} \tag{5.56d}$$

$$J(u^{i+1}) \leq J(u^i) + \sum_{k=0}^{N-1} \big[-\mu^i ||\delta u_k^i||^2 + \frac{L}{2}||\delta x_k^i||^2\big]. \tag{5.56e}$$

Since initial condition is given, $\delta x_0^i = 0, \forall i$. For the states with terminal inequality constraints $p_N^i = 0$, and for the states with terminal equality constraints, $\delta x_N^i = 0$. In either case, $(p_N^i)^T \delta x_N^i = 0$.

$$\begin{aligned} \delta x_{k+1}^i &= x_{k+1}^{i+1} - x_{k+1}^i = f_{p^{i+1},k}(x_k^{i+1}, v_k^{i+1}) - f_{p^i,k}(x_k^i, v_k^i) \\ &= \big[f_{p^{i+1},k}(x_k^{i+1}, v_k^{i+1}) - f_{p^{i+1},k}(x_k^i, v_k^i)\big] + \big[f_{p^{i+1},k}(x_k^i, v_k^i) - f_{p^i,k}(x_k^i, v_k^i)\big] \end{aligned} \tag{5.57}$$

Assumption 5-$\mathcal{E}$. $||f_{p^i}(x, v) - f_{p^j}(x, v)|| \leq \gamma_1 ||\alpha_k^{i+1} - \alpha_k^i||, \forall p^i, p^j \in Q$.

Mean value theorem for vector-valued functions [80] and (5-$\mathcal{E}$) yield

$$\begin{aligned} ||\delta x_{k+1}^i|| \leq& ||\frac{\partial f_{p^{i+1},k}}{\partial x_k}||||\delta x_k^i|| + ||\frac{\partial f_{p^{i+1},k}}{\partial v_k}||||\delta v_k^i|| + \gamma_1 ||\alpha_k^{i+1} - \alpha_k^i|| \\ \leq& M_1 ||\delta x_k^i|| + \gamma_2 ||\delta v_k^i|| + \gamma_1 ||\alpha_k^{i+1} - \alpha_k^i|| = M_1 ||\delta x_k^i|| + \begin{bmatrix} \gamma_2 & 0 \\ 0 & \gamma_1 \end{bmatrix} \begin{bmatrix} ||\delta v_k^i|| \\ ||\alpha_k^{i+1} - \alpha_k^i|| \end{bmatrix} \\ =& M_1 ||\delta x_k^i|| + \begin{bmatrix} \gamma_2 & 0 \\ 0 & \gamma_1 \end{bmatrix} ||u_k^{i+1} - u_k^i|| \leq M_1 ||\delta x_k^i|| + M_2 ||\delta u_k^i|| \end{aligned} \tag{5.58}$$

Consequently,

$$||\delta x_k^i|| \leq M_1||\delta x_{k-1}^i|| + M_2||\delta u_{k-1}^i|| \leq M_1(M_1||\delta x_{k-2}^i|| + M_2||\delta u_{k-2}^i||) + M_2||\delta u_{k-1}^i||$$
$$\leq \sum_{j=0}^{k-1} M_2 M_1^{k-j-1}||\delta u_j^i||, \tag{5.59}$$

since $\delta x_0^i = 0$. Using $M = \max_{j=0,1,...k-1}\{M_1^{k-j-1}\}$,

$$||\delta x_k^i||^2 \leq \sum_{j=0}^{k-1} 2M_2^2 M^2 ||\delta u_j^i||^2, \tag{5.60}$$

which means

$$\sum_{k=0}^{N-1} ||\delta x_k^i||^2 \leq \sum_{k=1}^{N-1}(\sum_{j=0}^{k-1} 2M_2^2 M^2 ||\delta u_j^i||^2) \leq \sum_{k=0}^{N-1} R||\delta u_k^i||^2 \tag{5.61}$$

$$J(u^{i+1}) - J(u^i) \leq \sum_{k=0}^{N-1}(R - \mu^i)||\delta u_k^i||^2 \tag{5.62}$$

As long as $||\delta u^i||_\infty \neq 0$, μ^i can always be chosen large enough to decrease $J(u^{i+1})$. However, in the actual implementation according to Algorithm 2, sometimes $||\delta u^i||_\infty = 0$ in some intermediate iterations for some μ^i, which means $u^{i+1} = u^i$ and thus $J(u^{i+1}) = J(u^i)$. Per **Step 14** μ^i would keep decreasing. After some iterations when μ^i becomes small and $||\delta u^i||_\infty \neq 0$, the value of μ^i may not to able to guarantee the decrease in decrease $J(u^{i+1})$ and μ^i will increase according to **Step 12**. As a result, in certain applications the shooting iterations in actual implementation would oscillated and trapped between **Step 12** and **Step 13** and μ^i is not able to decrease to 0. Addressing this issue will be the future work. Note that in part B, the penalty weight μ is fixed $\mu^i \equiv \mu > 0$ and determined experimentally to avoid oscillations. Admittedly, a nonzero μ could indicate potential sub-optimal solutions, the stabilization of DMIS iterations are more critical for online implementation than strict optimality, especially any real-time iteration (RTI) scheme is approximate and sub-optimal [81]. On the other hand, it is observed that a fixed, nonzero μ could still lead to close-to-optimal solutions on several numerical examples, since when the solution converges, $u^{i+1} = u^i$ indicates in **Step 8**, the stabilizing term $\mu^i||u - u_k^i||^2 = 0$, and the original Hamiltonian is recovered.

5.4 Benchmark Problems with Eco-Driving Applications

In this section, the performances of the DMIS are compared to other methods[3] on two benchmark problems with quadratic cost functions and linear state constraints: one with continuous-valued control inputs (QP) and the other one with integer-valued control inputs (mixed-integer quadratic program (MIQP)). The problem can be considered as the acceleration minimization of the vehicle-following dynamics.

1. For the continuous-valued control problem, DMIS is compared to the state-of-art QP solver using a FBRS-based multiple shooting (will be abbreviated as FBRS in the sequel) [47] ($\mathbf{P}_0$).

2. For the integer-valued control problem, DMIS is compared to IBM-CPLEX [57] ($\mathbf{P}_2$).

The first benchmark problem is chosen to be the acceleration minimization of a double-integrator ($\mathbf{P}_0$) with problem horizon N and time step Δt.

($\mathbf{P}_0$): Minimize

$$J = \sum_{k=0}^{N-1} \frac{1}{2} \cdot a_k^2 \cdot \Delta t, \tag{5.63a}$$

where a_k is the control input, denoting the vehicle acceleration at time $t = k$. (5.63a) is subject to the system dynamics

$$s_{k+1} = s_k + v_k \cdot \Delta t, \quad v_{k+1} = v_k + a_k \cdot \Delta t, \tag{5.63b}$$

where s_k and v_k are the position and velocity of the considered ego vehicle, respectively, and subject to state constraints

$$s_{l,k-[\frac{3}{\Delta t}]} - 4 = s_k^{\min} \leq s_k \leq s_k^{\max} = s_{l,k-[\frac{1}{\Delta t}]} - 1 \tag{5.63ca}$$

$$v_k^{\min} \leq v_k \leq v_k^{\max}, \tag{5.63cb}$$

where $s_{l,k-[\frac{3}{\Delta t}]}$, $s_{l,k-[\frac{1}{\Delta t}]}$ represents the lead vehicle's position 3s and 1s ago, respectively. The state constraints (5.63ca) is related to the presence of a lead vehicle. Therefore, its position is constrained by the lead vehicle, where $s_k^{\min}$ and $s_k^{\max}$ are the minimum and maximum allowable distance and are chosen to avoid cut-ins from adjacent lanes and collisions with the lead vehicle, respectively, and are functions of the position of the lead vehicle.

[3] Note that all the computations are done on a Mac OS X with an Intel® Core i5 2.7 GHz processor and 8GB RAM.

In addition, the control input a_k needs to satisfy

$$a_k^{\min} \leq a_k \leq a_k^{\max}. \tag{5.63d}$$

Note that the ego vehicle represented by a double-integrator system (5.63b) has unstable dynamics forward-in-time. It indicates a high sensitivity of the change in the terminal state to the perturbation in the terminal co-state and thus not suitable to apply single shooting. Although this is not an issue for multiple shooting, it is difficult to solve the associated TPBVP numerically with single shooting. Consequently, to make the system suitable for single shooting, the original double integrator system (5.63b) is pre-stabilized to formulate the equivalent system

$$s_{k+1} = s_k + v_k \cdot \Delta t, \quad v_{k+1} = K \cdot (s_{l,k} - s_k + R \cdot r_k), \tag{5.1}$$

where $K \in (0, 1)$ is a stabilizing feedback gain, R is the normalizing term, and r_k is the control input. As a result, the ego vehicle's velocity v_{k+1} is in a linear feedback form of the reference distance gap between the lead vehicle and itself, where this reference gap is optimized by changing the normalized input r_k. The acceleration a_k is a function of the control r_k, the state s_k, v_k and the position of the lead vehicle $s_{l,k}$ calculated by

$$a_k = \frac{v_{k+1} - v_k}{\Delta t} = \frac{K \cdot (s_{l,k} - s_k + R \cdot r_k) - v_k}{\Delta t}. \tag{5.2}$$

With the pre-stabilized reformulation (5.1), the control input becomes r_k, where a_k is a function of the state, the control input and the lead vehicle's positon $s_{l,k}$. It should be pointed out that v_{k+1} in (5.1) would only make sense with discrete-time formulation as there is no direct continuous-time counterpart.

Remark 4. Note that the pre-stabilizing controller in (5.1) is a variation of the adaptive cruise controller (ACC), but with an additional degree of freedom that seeks the optimal reference $\Delta d_k^* = s_{l,k} - s_k + R \cdot r_k$ by selecting r_k.

A smooth exterior penalty method is used to cope with the state constraints (5.63ca),(5.63cb). This, combined with the pre-stabilized reformulation (5.1), results in an equivalent acceleration minimization problem (**P**$_1$) which is then augmented with discrete controls.

(**P**$_1$): Minimize

$$\begin{aligned} J = \sum_{k=0}^{N-1} \frac{1}{2}\Delta t \cdot [(\frac{v_{k+1} - v_k}{\Delta t})^2 + (\max(s_k - s_k^{\max}, 0))^2 + (\max(s_k^{\min} - s_k, 0))^2 \\ + 10 \cdot (\max(-v_k, 0))^2] \end{aligned} \tag{5.3a}$$

Table 5.1: Performance comparison between DMIS and FBRS

	FBRS	DMIS
N (problem horizon)	1969	1969
Objective Function J	114.2773	114.0058
$s_N \;\; [m]$	2.1215e+04	2.1215e+04
$v_N \;\; [m/s]$	0.8383	0.8618
Total iterations	200	633983
Computation time $\;[s]$	17	297

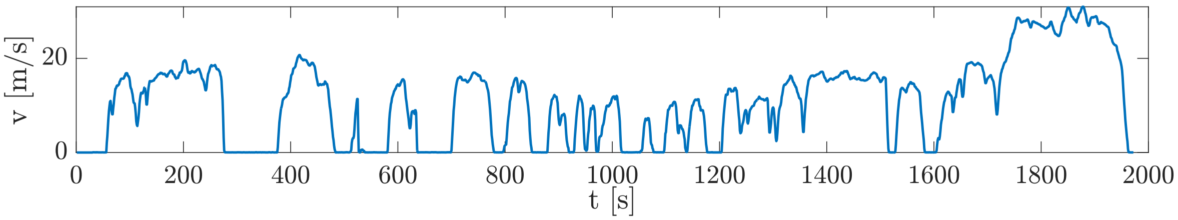


Figure 5.3: An illustration of the lead vehicle's trajectory, a drive cycle recorded in Ann Arbor, MI, USA [3].

subject to the dynamics (5.1) and the control constraints

$$r_k \in \begin{cases} [r_k^{\min}, r_k^{\max}], & \text{if } r_k \text{ takes } \textit{continuous} \text{ value} \\ \{-1, 0, 1\}, & \text{if } r_k \text{ takes } \textit{integer} \text{ value.} \end{cases} \tag{5.3b}$$

In (5.3a), the weights on the penalty terms are selected experimentally to balance the constraint satisfaction and the stability of the DMIS iterations.

5.4.1 Comparison against FBRS

The performance of the DMIS on ($\mathbf{P}_1$) is compared against multiple shooting on ($\mathbf{P}_0$) based on FBRS [47] when the control input r_k takes *continuous* value. Here $\Delta t = 1$, $R = 1$, $K = 0.1$, $r_k^{\max} = 2000$, $r_k^{\min} = -2000$.

The lead vehicle's velocity is shown in Fig. 5.3, where its position s_l is obtained by numerical integration. Figure 5.4 plots the resulting velocity, acceleration, and co-states trajectories obtained with multiple shooting and single shooting. It can be seen from Fig. 5.4 that the trajectories obtained with DMIS are very close to those with the multiple shooting. The results are also summarized in Table 5.1. As shown in Table 5.1, the objective function obtained with single shooting is very close to that with multiple shooting, where the small discrepancy in the cost function is due to the difference in the terminal state ($p_N = 0$ no terminal equality constrains are enforced). The computation time with single shooting is approximately 300

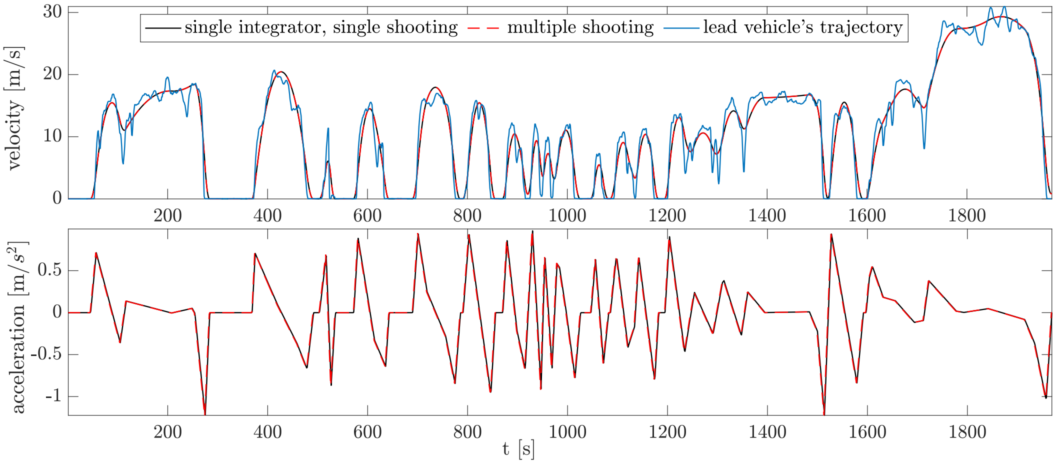


Figure 5.4: Result comparison between DMIS (single shooting) and FBRS (multiple shooting) with the lead vehicle's velocity shown in Fig. 5.3. Black solid curve: single integrator, single shooting, $K = 0.1$, $\mu^0 = 2000$, $\mu^f = 0.35$. Red dashed curve: multiple shooting. Blue solid curve: lead vehicle's trajectory.

sec (5 min) with more than 630k iterations, and the computation time with multiple shooting is 17s (200 iterations). The computation and number of iterations with single shooting is considerably longer compared to the multiple shooting. This is reasonable since single shooting only uses first-order information. By comparison, the multiple shooting exploits second-order information (Hessian) and thus provides faster convergence behavior, which cannot be extended to a mixed-integer domain.

5.4.2 Comparison against IBM-CPLEX

The performance of the DMIS on (**P**$_1$) is compared against the state-of-art IBM-CPLEX [57] mixed-interger quadratic program (MIQP) solver when the control input r_k takes *integer* value. Here $\Delta t = 1$, $R = 10$, $K = 0.3$. Note that since the $\max$ terms in (**P**$_1$) make the problem no longer MIQP for IBM-CPLEX, (**P**$_1$) is reformulated into the following form (**P**$_2$) by introducing additional relaxation variables to allow consistent comparison with the formulation (**P**$_1$) used in single shooting.

(**P**$_2$): Minimize

$$J = \sum_{k=0}^{N-1} \frac{1}{2}\Delta t \cdot [(\frac{v_{k+1} - v_k}{\Delta t})^2 + y_{1,k}^2 + y_{2,k}^2 + 10 \cdot y_{3,k}^2], \tag{5.4a}$$

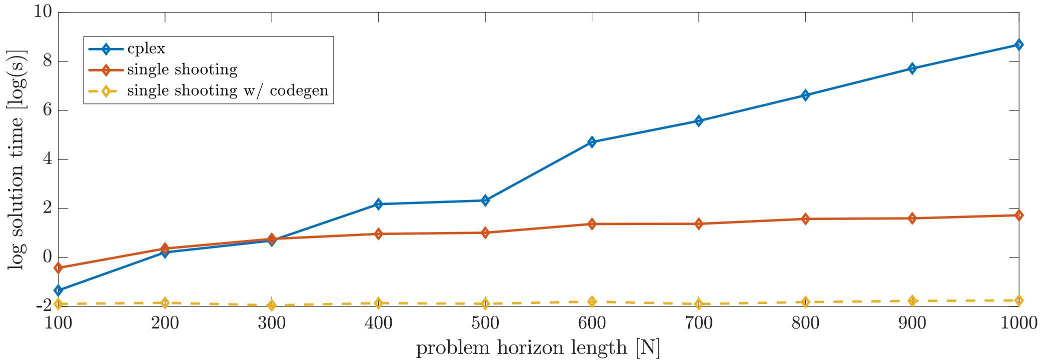


Figure 5.5: Semi-log plot of the computation time comparison as a function of the problem horizon. Blue solid: CPLEX. Red solid: Single shooting. Yellow dashed: Single shooting with MEX Function from Matlab C code generation.

subject to dynamics (5.1) and the constraints

$$s_k - s_k^{\max} - y_{1,k} \leq 0, \quad s_k^{\min} - s_k - y_{2,k} \leq 0, \quad -v_k - y_{3,k} \leq 0. \tag{5.4b}$$

The maximum difference in objective function values is 0.01% between the values obtained from single shooting and the integer-optimal objective function value with IBM-CPLEX, with different problem horizons. However, as shown in Table 5.2, the computation time with single shooting is significantly faster compared to that with IBM-CPLEX. It can be seen from the Semi-log plot of the computation time in Fig. 5.5) that the single shooting scales linearly to the problem horizon, by comparison, IBM-CPLEX with branch-and-cut based strategy scales exponentially to the problem horizon.

Table 5.2: Comparison against CPLEX: computation time [s]

N (problem horizon)	Cplex class	Single shooting	Single shooting + codegen
100	0.3	0.7	0.15
200	1.2	1.4	0.16
300	2.0	2.1	0.14
400	8.8	2.6	0.15
500	10.2	2.7	0.15
600	110.8	4.0	0.16
700	262.2	4.0	0.15
800	748.2	4.8	0.16
900	2223.0	4.9	0.17
1000	5880.8	5.6	0.17

5.5 Power-Split Optimization Accounting for Fuel Cranking

In this section, the power-split optimization with an additional cranking state is formulated directly in discrete-time, and its associated TPBVP is established with the modified DMIS. The resulting TPBVP is then solved numerically with DMIS Algorithm 2 presented in the previous section. The simulation results with DMIS are presented to solve the PHEV minimum fuel consumption problem offline with an additional engine cranking state. A significant reduction in cranking fuel consumption is observed in simulation compared to the PMP-based approach without the cranking state, demonstrating its capability to resolve the chattering behavior encountered by the continuous-time PMP-based solution, as discussed in Section 5.5.1.1. DMIS is demonstrated to tackle problems with multiple states, stabilize co-state dynamics that are initially unstable with a continuous-time formulation. Afterward, an MPC version of the DMIS is implemented on the dyno[4], demonstrating its real-time capability and fuel economy benefit compared to the stock CDCS strategy.

5.5.1 Limitation of the Continuous-Time Optimization

In terms of a suitable numerical strategy for the underlying minimum fuel consumption problem for the energy management and optimization of (P)HEVs, previous efforts [30, 3] focused on the feasibility study of the *single shooting* approach. For a trip with given (or predicted) velocity and thus torque demand profiles, the minimum fuel consumption problem is formulated in continuous time. PMP is then applied to get the dynamics of the co-states. The resulting continuous-time TPBVP is discretized, where single shooting is used to obtain its numerical solution. Good fuel economy compared to the baseline CDCS strategy is demonstrated in simulations.

Admittedly, from a numerical implementation perspective, PMP has demonstrated its strong applicability to the (P)HEV power-split optimization problem. In particular, the instantaneous Hamiltonian minimization avoids the explicit consideration of the integer decision variable representing the powertrain mode selection, as discussed in the sequel. However, instantaneous Hamiltonian minimization resulting from PMP could induce high-frequency switches in the control inputs, known as the *chattering behaviors*. Chattering behaviors are reflected in the (P)HEV power-split optimization problem as the busy engine on/offs. There are relatively fewer research efforts in optimizing the transient engine behaviors than the abundant work where only

[4] Acknowledgment: The dyno implementation and its results are work done by Dr. Mike Huang while he was with Toyota. I am also grateful to Dr. Ken Butts from Toyota for his efforts.

quasi-static engine maps are considered. In [23], an MPC torque-split strategy is proposed, where the transient engine characteristics are considered through an augmented weighted cost related to the engine mode switch. However, the engine mode within the prediction horizon is determined heuristically. In [24], a control algorithm is proposed to regulate the transitions between different operating points by using the battery to smoothen the engine transients. However, the problem essentially becomes a regularization problem for an HEV because of its small allowable SOC range. It is not applicable in the PHEV application because the desired SOC reference requires solving the fuel minimization problem first, where the busy engine cranking problem persists. In [25], an additional integer-valued engine on/off state is considered, with engine cranking cost augmented in the cost function. However, its numerical strategy is simplified by the constant co-state assumption, and the solution is obtained by sequentially applying dynamic programming (DP) and convex optimization. It is hardly applicable to the PHEV application, where the constant co-state assumption is not valid due to the large SOC span. Moreover, the real-time implementability of the proposed sequential optimization strategy is unclear.

As detailed in this section, the chattering behavior can only be addressed fundamentally by introducing an additional engine cranking state. However, a critical issue appears with the continuous-time formulation, i.e., the co-state corresponding to the cranking state becomes unstable even though the engine cranking has stable dynamics. As a result, such continuous-time, PMP-based shooting iterations are hard to stabilize in practice.

5.5.1.1 Chattering Behaviors in Hamiltonian Minimization

The Hamiltonian of the continuous-time minimum fuel consumption problem is

$$H = \dot{m}_f(u) + p \cdot \dot{\text{SOC}}(\text{SOC}, u, v, tp), \tag{5.5}$$

where as already stated in previous chapters, the state $x = \text{SOC}$, the control input $u = [te,\ ne]$, v is the velocity of the PHEV as the external disturbance and tp is the driver demanded torque. p is the co-state. Based on the necessary condition for optimality (PMP), the optimal control at time t is the instantaneous minimizer of the Hamiltonian (5.5)

$$u^* = \underset{u \in \mathcal{U}}{\arg\min}\{\dot{m}_f(u) + p \cdot \dot{\text{SOC}}(\text{SOC}, u, v, T_p)\}, \tag{5.6}$$

and the co-state dynamics are

$$\dot{p} \ = -\frac{\partial H}{\partial \text{SOC}} = -\frac{\partial \dot{\text{SOC}}(\text{SOC}, u, v, tp)}{\partial \text{SOC}} \cdot p, \tag{5.7}$$

since $\dot{m}_f$ does not explicitly depend on SOC.

The resulting discrete-time TPBVP with the forward-Euler scheme and a time step Δt

$$\text{SOC}_0 = \text{SOC}_{int}, \;\; \text{SOC}_N = \text{SOC}_f \tag{5.8a}$$

$$u_k^* = \arg\min_{u\in\mathcal{U}} \{\dot{m}_{f,k}(u) + p_k \cdot \dot{\text{SOC}}_k(\cdot)\}, \tag{5.8b}$$

$$\text{SOC}_{k+1} = (1 + \frac{\dot{\text{SOC}}_k(\cdot, u_k^*, \cdot)}{\text{SOC}_k} \cdot \Delta t) \cdot \text{SOC}_k, \tag{5.8c}$$

$$p_{k+1} = \big(1 - \frac{\partial \dot{\text{SOC}}_k(\cdot, u_k^*, \cdot)}{\partial \text{SOC}_k} \cdot \Delta t\big) \cdot p_k. \tag{5.8d}$$

Note that the PHEV powertrain is a switched hybrid system: it can either operate in HV mode, where the engine is on, or in EV mode where the engine is off. As discussed in [30], u_k^* in (5.8b) is obtained by comparing the HV ($u \in \Omega$) and EV mode Hamiltonian ($u = [0, 0]^T$)

$$u_k^* = \begin{cases} \arg\min_{u\in\Omega} H_{HV,k}^* & \text{if } \; H_{HV,k}^* < H_{EV,k} - \epsilon \\ [0, 0]^T & \text{otherwise} \end{cases}, \tag{5.9}$$

where

$$H_{HV,k}^* = \min_{u\in\Omega} \{\dot{m}_{f,k}(u) + p_k \cdot \dot{\text{SOC}}_k(\cdot, u, \cdot)\}, \tag{5.10a}$$

$$H_{EV,k} = \dot{m}_{f,k}(u) + p_k \cdot \dot{\text{SOC}}_k(\cdot, [0, 0]^T, \cdot), \tag{5.10b}$$

and ϵ is a threshold that could be chosen heuristically to defer the engine from being frequently cranked on. However, it can be seen that the powertrain operation determined from (5.9) could jump arbitrarily (chattering behavior) between HV and EV mode, resulting in busy engine on/off switches. Moreover, note that adding hysteresis between HV and EV mode switch by increasing ϵ does not fundamentally resolve the chattering behavior and is challenging to tune by our observations.

5.5.1.2 Limitation of the Continuous-Time Formulation

To properly address the chattering behaviors caused by the instantaneous Hamiltonian minimization (5.9), the original fuel minimization problem is augmented with the engine cranking dynamics and their associated cranking cost.

Minimize

$$J = \int_0^{t_f} \{\dot{m}_f(u) + m_c \cdot \max\big(\dot{e}(e, E), 0\big)\} dt, \tag{5.11a}$$

subject to the SOC dynamics and the normalized engine cranking dynamics modeled as [82]

$$\dot{e} \ = -\beta \cdot e + \beta \cdot E, \ \ \beta > 0, \tag{5.11b}$$

where $E \in \{0, 1\}$ is the engine on/off command. m_c is the constant cranking penalty, which corresponds, physically, to the cranking event energy cost (not a tuning parameter), and β is the inverse of the time constant for the engine cranking dynamics.

The necessary condition for optimality on (5.11) yields

$$\begin{aligned} H &= \dot{m}_f + m_c \cdot \max(\dot{e}, 0) + p_1 \cdot \dot{\text{SOC}} + p_2 \cdot \dot{e} && (5.12a) \\ \dot{p}_1 &= -\frac{\partial H}{\partial \text{SOC}} = -\frac{\partial \dot{\text{SOC}}}{\partial \text{SOC}} \cdot p_1 && (5.12b) \\ \dot{p}_2 &= -\frac{\partial H}{\partial e} = \beta \cdot p_2 + \beta \cdot m_c \cdot \mathbb{1}(\dot{e} > 0), && (5.12c) \end{aligned}$$

where $\mathbb{1}(\cdot)$ is the indicator function. It can be seen that although the engine cranking has stable dynamics (5.11b), the unforced dynamic state of the co-state p_2 (5.12c) is unstable.

5.5.2 Discrete-Time Formulation with Engine Cranking State

The discrete-time minimum fuel consumption problem with the SOC and engine cranking state is formulated as

Minimize

$$J = \sum_{k=0}^{N-1} \left\{ \Delta t \cdot \dot{m}_{f,k}(u_k) + m_c \cdot \max(\tilde{e}_{k+1} - \tilde{e}_k, 0) \right\}, \tag{5.13a}$$

subject to state dynamics with

$$\begin{aligned} \text{SOC}_{k+1} \ &= \text{SOC}_k + \Delta t \cdot f(\text{SOC}_k, u_k, v_k, tp_k) \\ \tilde{e}_{k+1} \ &= \tilde{\alpha} \cdot \tilde{e}_k + (1 - \tilde{\alpha}) \cdot E_k, \end{aligned} \tag{5.13b}$$

where E_k is the engine on/off command, and $\tilde{e}_k$ is the filtered engine on/off state at $t = k$. $\tilde{\alpha}$ in (5.13b) is related to β in (5.11b) by $\tilde{\alpha} = 1 - \Delta t \cdot \beta \in (0, 1)$. Meanwhile, the initial state and the desired terminal SOC are given

$$\text{SOC}_0 = \text{SOC}_{int}, \ \ \tilde{e}_0 = 0, \ \ \text{SOC}_N = \text{SOC}_f, \tag{5.13c}$$

and control constraints are defined as

$$u_k \; = [te_k, ne_k]^T \in \mathcal{U}, \;\text{ where } \mathcal{U} = \Omega \cup [0,0]^T, E_k \in \{0,1\}. \tag{5.13d}$$

According to the DMP presented in Section 5.2.3

$$H_k = \Delta t \cdot \dot{m}_{f,k}(u_k) + m_c \cdot \max(\tilde{e}_{k+1} - \tilde{e}_k, 0) + p_{1,k+1} \cdot \text{SOC}_{k+1} + p_{2,k+1} \cdot \tilde{e}_{k+1} \tag{5.14a}$$

$$[u_k, E_k] \in \underset{u\in\mathcal{U}, E\in\{0,1\}}{\arg\min} \big\{\Delta t \cdot \dot{m}_{f,k}(u_k) + m_c \cdot \max(\tilde{e}_{k+1} - \tilde{e}_k, 0) + p_{1,k+1} \cdot \text{SOC}_{k+1} \tag{5.14b}$$

$$+ p_{2,k+1} \cdot \tilde{e}_{k+1}\big\}$$

$$p_{1,k} = \frac{\partial H_k}{\partial \text{SOC}_k} = \frac{\partial \text{SOC}_{k+1}}{\partial \text{SOC}_k} \cdot p_{1,k+1} \tag{5.14c}$$

$$p_{2,k} = \frac{\partial H_k}{\partial \tilde{e}_k} = m_c \cdot (\tilde{\alpha} - 1) \cdot \mathbb{1}(\tilde{e}_{k+1} - \tilde{e}_k > 0) + \tilde{\alpha} \cdot p_{2,k+1}. \tag{5.14d}$$

It can be seen that the unforced dynamics of p_2 (5.14d) in discrete-time are stable backward-in-time since $\tilde{\alpha} \in (0,1)$.

Remark 5. Note that as discussed after Theorem 5.2.8, in the numerical implementation, finite difference is used to evaluate an element from their generalized gradients [78] in the co-state dynamics. Consequently, equality ($=$) replacing the set inclusion ($\in$) is used in (5.14c) and (5.14d) as well as in the sequel for simplicity.

5.5.3 Simulation Results

The parameters in the Algorithm 2 used for simulations in this section are as follows

$$\epsilon_u = 1, \;\; \epsilon_J = 0.001, \;\; d\mu = 0.01, \;\; \gamma = 1.01, \mu^0 = 0.4, \;\; \epsilon_x = \epsilon_{\text{SOC}} = 0.001.$$

To demonstrate the effectiveness of the DMIS in resolving the chattering behavior as discussed in Section 5.5.1.1 (in our application, the frequent engine on/off switches), the baseline is defined as the minimum fuel consumption problem formulation as in (5.11) without considering the engine cranking. The resolving discretized TPBVP (5.8) is solved numerically with PMP-based single shooting. Note that in the offline simulations, the velocity trajectories are the same for each simulation, with the driver demanded torque calculated by (4.5) (backward-looking model). As a result, the effect of driver variations is eliminated in the offline simulations.

Figure 5.6a compares the state and control trajectories obtained with DMIS on (5.14) against those with PMP-based single shooting on (5.8) with a realistic driving cycle. The first subplot presents the considered driving cycle. The second subplot shows the SOC trajectory. The third subplot shows the total fuel consumption. The fourth subplot shows the cranking fuel, and the fifth and sixth subplots show the engine on/off command resulting from the DMIS and PMP-based method, respectively. As can be seen from the fourth subplot, the cranking fuel with (5.14) is considerably less compared to that with (5.8). A detailed view of Fig. 5.6a between 50s and 300s is shown in Fig. 5.6b. As can be observed by comparing the fourth and fifth subplot of Fig. 5.6b, the engine switches on/off much less frequently with DMIS-based compared to the baseline PMP-based optimization without engine cranking optimization. The total fuel (cranking fuel included) is 527 grams with additional cranking fuel optimization (5.14), achieves 3$\%$ additional improvement in fuel economy compared to 544 grams without additional cranking fuel optimization (5.8).

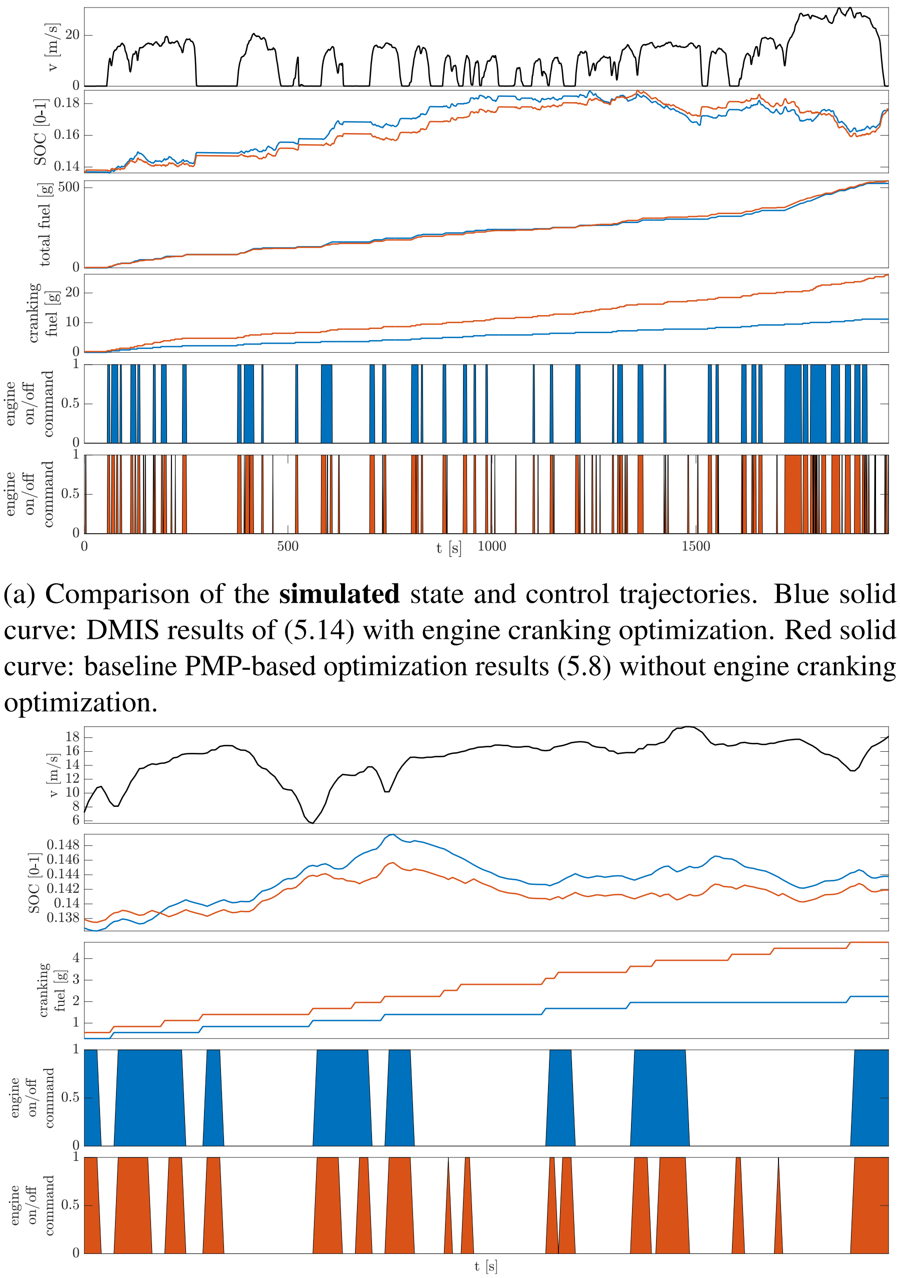
Besides, the penalty term for stabilizing shooting iterations eventually decreases to 0, as shown in Fig. 5.7. This indicates that in **Step 8** of Algorithm 2, u_k^{i+1} satisfies the discrete-time maximum principle.

5.6 Velocity and Power-Split Co-Optimization with DMIS

The proposed DMIS strategy not only addresses the chattering behaviors in the PHEV power-split optimization problem, it also enables systematically solving the PHEV velocity and power-split co-optimization. It demonstrates better scalability with respect to problem horizon compared to the hierarchical approach proposed in Chapter 4.

For co-optimizing the velocity profile of the PHEV and its powertrain dynamics to achieve minimum fuel consumption, it is reasonable to decouple the overall system into two subsystems: (i) the vehicle-following subsystem and (ii) the hybrid powertrain subsystem. These two subsystems are connected through the vehicle (longitudinal) velocity v_k, and the demanded torque tp_k (5.15c) to be satisfied by the powertrain components. In this section, the ego vehicle is assumed to drive on a single lane without grades for simplicity, and only its longitudinal dynamics are considered.

The vehicle level control focuses on controlling the longitudinal acceleration a_k of the vehicle in the presence of a lead vehicle. At the vehicle level, the states of consideration are the position s_k and the velocity v_k of the ego PHEV. The dynamics of the ego vehicle are defined as follows in the car-following context, which is a variant of the adaptive cruise control



(a) Comparison of the **simulated** state and control trajectories. Blue solid curve: DMIS results of (5.14) with engine cranking optimization. Red solid curve: baseline PMP-based optimization results (5.8) without engine cranking optimization.

(b) Detailed view of the state and control trajectories between 50-sec and 300-sec. Blue solid curve: DMIS results of (5.14) with engine cranking optimization. Red solid curve: baseline PMP-based optimization results (5.8) without engine cranking optimization.

Figure 5.6: Offline power-split optimization **simulation** results with a realistic driving cycle.

formulation, but with an additional input that controls the reference distance gap

$$\begin{aligned} s_{k+1} &= s_k + v_k \cdot \Delta t + \frac{1}{2} a_k \cdot \Delta t^2 \\ v_{k+1} &= K \cdot (s_{l,k} - s_k + R \cdot r_k), \end{aligned} \tag{5.15a}$$

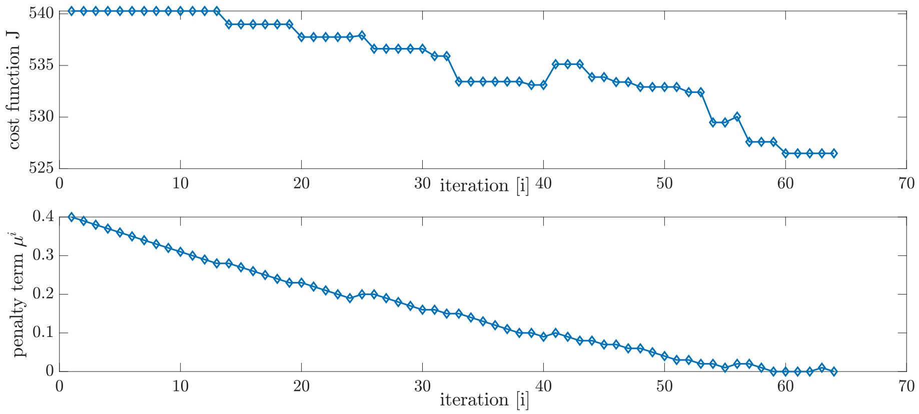


Figure 5.7: Convergence results of DMIS

with parameters similarly defined as in Section 5.4. Note that the exact integration is used in (5.15a) to enable a relatively large time step and avoid numerical inaccuracy and the influence on SOC dynamics. The states subject to the (5.63ca), (5.63cb).

The acceleration a_k becomes a function of the state and control input

$$a_k = \frac{v_{k+1} - v_k}{\Delta t} = \frac{K \cdot (s_{l,k} - s_k + R \cdot r_k) - v_k}{\Delta t} \tag{5.15b}$$

The torque demand of the vehicle tp_k at $t = k\Delta t$ is defined through a regression model that needs to be satisfied by the low-level powertrain components

$$tp_k = \alpha_0 + \alpha_1 \cdot a_k + \alpha_2 \cdot v_k + \alpha_3 \cdot v_k^2, \tag{5.15c}$$

where $\alpha_i, i \in \{0, 1, 2, 3\}$ are the regression parameters.

At the powertrain level, the hybrid powertrain subsystem consists of the engine, the motor (MG2), and the generator (MG1), which are connected through a power-split device. The engine torque te_k and speed ne_k in the powertrain subsystem are controlled to satisfy the required torque tp_k, which results in a specific fuel rate $\dot{m}_{f,k}$ and rate of change in battery state-of-charge (SOC).

The dynamics of the battery SOC and the engine cranking are given by (5.13b). Every time the engine is cranked on, an additional cranking fuel m_c per cranking event would be induced.

The fuel rate $\dot{m}_{f,k}$ at $t = k \cdot \Delta t$ is modeled as a static map as a function of the engine torque te_k and speed ne_k

$$\dot{m}_{f,k} = \dot{m}_{f,k}(ne_k, te_k). \tag{5.16}$$

5.6.1 Co-Optimization Problem Formulation

The problem of the PHEV velocity and power split co-optimization to achieve minimum fuel consumption (including cranking fuel) of a considered trip in the presence of a lead vehicle, with state constraints incorporated through smooth exterior penalties is formulated as

($\mathbf{P}_1$): Minimize

$$J = \sum_{k=0}^{N-1} l_k(x_k, u_k), \tag{5.17a}$$

where the stage cost l_k includes the fuel rate, the cranking fuel and the quadratic measurement of the constraint violation

$$\begin{aligned} l_k = {} & m_c \cdot \max(\tilde{e}_{k+1} - \tilde{e}_k, 0) + \Delta t \cdot \Big[\dot{m}_{f,k}(te_k, ne_k) + \frac{1}{2} \cdot \max\left(s_k - s_k^{\max}, 0\right)^2 \\ & + \frac{1}{2} \cdot \max\left(s_k^{\min} - s_k, 0\right)^2 + 2 \cdot \max\left(-v_k, 0\right)^2 + \max\left(a_k - a_k^{\max}, 0\right)^2 \\ & + \max\left(-a_k + a_k^{\min}, 0\right)^2\Big], \end{aligned} \tag{5.17b}$$

with $s_0, v_0,$ and SOC$_0$ given, and subject to the vehicle-following dynamics (5.15a), the battery SOC and the engine cranking dynamics (5.13b). Note that a quadratic exterior penalty method is used to penalize the constraint violation. Although other different constraint handling techniques [83], especially an ℓ_1-type, exact penalty method as suggested in [62] or a fourth-order penalty function to ensure the twice continuous differentiability of the Hamiltonian as in [84] can be used, it is observed in simulation that the second-order penalty function works best for stabilizing the co-state dynamics in DMIS iterations. Therefore, it is used in this chapter whenever state inequality constraints are considered in a DMIS-based numerical strategy. It will also be used in the remaining chapters without further explanation. The velocity v_k, position s_k and acceleration a_k are subject to the constraints defined the same as in (5.63cb), (5.63ca) and (5.63d). The control constraints are

$$[te_k, ne_k] \in \Omega_k \cup [0, 0]^T, \quad E_k \in \{0, 1\}. \tag{5.17c}$$

In addition, a desired SOC, SOC$_f$ needs to be satisfied at the end of the trip

$$\text{SOC}_N = \text{SOC}_f. \tag{5.17d}$$

To summarize, the co-optimization problem ($\mathbf{P}_1$) considers 4 states $x_k = [s_k, v_k, \text{SOC}_k, \tilde{e}_k]^T$ and 4 control inputs: $u_k = [r_k, E_k, te_k, ne_k]^T$, where the engine on/off command E_k takes only integer values. $w_k = s^l_k$ is the position of the lead vehicle (known external input). In this section, idealized/perfect forecasting of the lead vehicle's position is assumed. In the sequel, the control input u_k is grouped into 2 groups $u_k = [r_k, u^T_{2,k}]^T$, where $u_{2,k} = [E_k, ne_k, te_k]^T$ and the constraints are denoted as U_2.

5.6.2 Hierarchical Newton's Method for Hamiltonian Minimization

Previously, minimizing the Hamiltonian of the co-optimization problem is performed through discretization and greedy search for solution reliability. It is because several possible local minima have already been identified in the pure power-split optimization problem. With an increased dimension in the co-optimization problem, it is hard to visualize the shape of the Hamiltonian and thus hard to judge the reliability and performance of a gradient-based solver. Although discretization and greedy search are reliable ways of minimizing a nonsmooth objective function, thousands of discrete Hamiltonian values must be evaluated, making the total computation time still too high to be potentially implemented in real-time. In this section, the Hamiltonian minimization is performed hierarchically by exploiting the problem structure. As a result, it enables a gradient-based solver and improves the computational efficiency of the Hamiltonian minimization.

The Hamiltonian for (5.17) at time $t = k$ in DMIS [85] is

$$H_k = l_k(\cdot) + p^T_{k+1} x_{k+1}, \tag{5.18a}$$

and the control u_k is determined through instantaneous minimizater of (5.18a) as

$$u_k = \arg\min_{u \in U} H_k \tag{5.18b}$$

However, u_k obtained directly from (5.18b) requires a complicated 4-dimensional search in the feasible control space. To simplify the instantaneous Hamiltonian minimization, first, note that the engine on/off command E_k is a function of the engine speed

$$E_k(ne_k = 0) = 0, E_k(ne_k > 0) = 1. \tag{5.19}$$

Furthermore, for a given r_k, the acceleration a_k at time $t = k$ is determined through (5.2), and the driver demanded torque is calculated with (5.15c). Then the selection of $u_{2,k}$ is equivalent to the instantaneous consumption minimization in the power-split optimization problem [3]. More precisely, the original Hamiltonian minimization problem in 4-dimension (5.18b) can be decomposed into the following 2 low-dimensional optimization problems in a hierarchical manner

$$r_k = \underset{r_k \in [-1,1]}{\arg\min}\, H_k^*(r_k, x_k, p_{k+1}, w_k), \tag{5.20a}$$

with

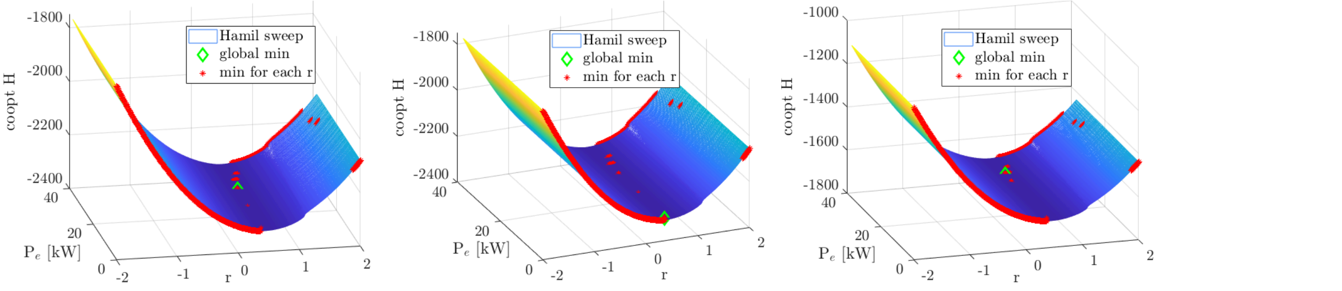
$$H_k^*(r_k, x_k, p_{k+1}, w_k) = H_k(r_k, u_{2,k}^*, x_k, p_{k+1}, w_k), \tag{5.20b}$$

where for a given r_k, the selection of $u_{2,k}$ is equivalent to the power-split optimization [3]

$$u_{2,k}^* = \underset{u_2 \in U_2}{\arg\min}\, H_k(r_k, u_2, x_k, p_{k+1}, w_k). \tag{5.20c}$$

Figure 5.8 illustrates a Hamiltonian sweep result. The engine torque and engine speed are lumped into a power term for visualization. The overall Hamiltonian sweep is presented in Fig. 5.8, and the minimum co-optimization Hamiltonian for each r_k (corresponding to the red stars in Fig. 5.8) is shown in Fig. 5.9. In Fig. 5.8, the red stars are to denote the minimum Hamiltonian corresponding to each given r_k. As can be seen from Fig. 5.9, the resulting minimum co-optimization Hamiltonian for each r_k presents a convex shape. The convexity of $H^*(r_k)$ motivates the use of a hierarchical Newton's method to minimize the co-optimization Hamiltonian. Note that the convexity is induced by the pre-stabilizing term $\mu \cdot (r - r^{i-1})^2$ as in Step. 8 in Algorithm 2.

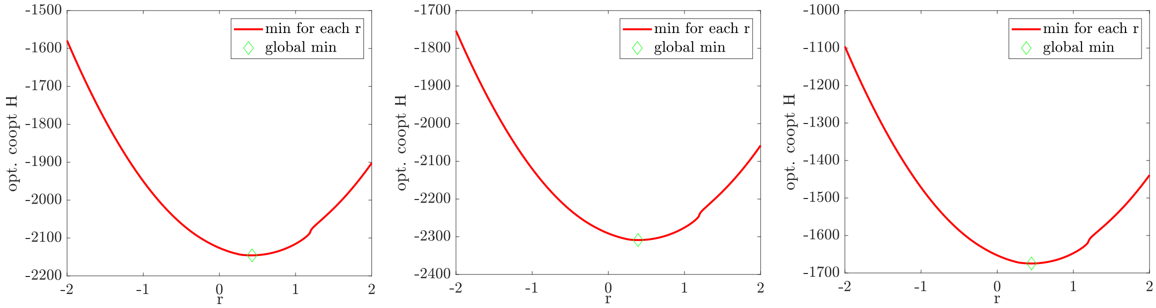
As can be seen from Fig. 5.9, the resulting minimum co-optimization Hamiltonian for each r_k presents a convex shape. The convexity of $H^*(r_k)$ motivates the use of a hierarchical Newton's method for the minimization of the co-optimization Hamiltonian. Note that the convexity is induced by the pre-stabilizing term $\mu \cdot (r - r^{i-1})^2$.



(a) Co-Opt Hamil, case 1. (b) Co-Opt Hamil, case 2. (c) Co-Opt Hamil, case 3.

Figure 5.8: Illustrations of co-optimization Hamiltonian with the considered PHEV powertrain.

However, issues exist in the motor limit violation. For a given r_k (thus a given acceleration a_k), it is not guaranteed the resulting motor and generator's torque limits are not violated.



(a) Co-Opt Hamil, case 1.

(b) Co-Opt Hamil, case 2.

(c) Co-Opt Hamil, case 3.

Figure 5.9: Minimum co-optimization Hamiltonian corresponds to each given r.

Therefore, motor and generator torque limits need to be properly included in the Hamiltonian minimization. To incorporate motor torque limits, the stage cost is augmented with

$$\tilde{l}_k(\cdot) = l_k(\cdot) + T \cdot \max(|T_{\text{mg2}}| - T_{\text{mg2}}^{\text{max}}, 0)^2 + T \cdot \max(|T_{\text{mg1}}| - T_{\text{mg1}}^{\text{max}}, 0)^2, \tag{5.21}$$

where T is the weight penalizing the torque constraint violation. Note that (5.21) is only used in the forward pass to choose the control input, not the backward pass to obtain the co-state dynamics.

Remark 6. Note that although the shape of the Hamiltonian is problem-specific, the idea of the hierarchical optimization still applies to the co-optimization of other hybrid powertrain and vehicle dynamics.

5.6.3 Underlying Reasons for Fuel Economy Benefit with Co-Optimization

In this section, a detailed analysis is provided to understand the underlying fundamental reasons for the fuel economy benefit with co-optimization. The simulation results of the co-optimization will be presented together with other different optimization strategies in Section 7, and thus omitted in this section to avoid duplication.

Compared to the existing sequential optimization approaches (thus decentralized), the direct fuel minimization of a PHEV by co-optimizing the velocity profile and powertrain operation (thus centralized) is different in that the powertrain dynamics is explicitly considered to determine vehicle acceleration in co-optimization. As compared to the fuel minimization of an internal combustion engine (ICE), or energy-efficient driving of a battery EV (single source vehicles), the PHEV (with two energy sources) co-optimization differs fundamentally in

1. Unlike the ICE, where the negative power demand at the wheel is wasted through heat,

the negative trip energy can be recovered by regenerative braking for a PHEV.

2. For a battery EV, the energy-efficient driving implies minimizing the total battery energy over a trip. By comparison, for a PHEV, the terminal SOC should be equal to the target value regardless of the choice of an optimization method. It means that the total battery energies spent are the same (or at least very close to each other).

For a PHEV, the engine and battery need to provide the power energy at the wheel; this means that $E_d \approx E_e + E_b$, with E_d, E_e and E_b representing the total demanded, the engine and battery energy, respectively. With the same initial and desired terminal SOCs, E_b-s are almost the same regardless of power management strategies as discussed above. This means that to minimize the total fuel consumption (or E_e), one should reduce the trip energy E_d (not just the positive part) while minimizing multiple energy conversion, i.e., using recuperated energy for vehicle propulsion.

For the vehicle level operation, from the demanded torque at wheel (5.15c), the power demand at wheel is given by

$$P_d = \frac{tp \cdot v}{R_t} = f(x) \cdot a + g(x). \tag{5.22}$$

where $f(x) = \frac{\alpha_1}{R_t} v$ and $g(x) = \frac{(\alpha_0 + \alpha_2 v + \alpha_3 v^2)}{R_t} v$ with $\alpha_{\{.\}}$ the regression parameters defined in (5.15c). It can be seen from (5.22) P_d is affine in the vehicle acceleration a[5]. Thus the Hamiltonian corresponds to the trip energy minimization problem is also affine in control a (when state constraints are not active). The affine relationship explains the reason why the resulting acceleration is of the bang-singular arc-bang type.

Ideally, the engine should operate at one of the local power-split Hamiltonian islands for the powertrain level operation due to the fuel minimization objective. However, if the driving cycle is smoothed through acceleration minimization, ignoring the powertrain dynamics, the resulting driver demanded power would decrease significantly. Because the best efficient engine operation is achieved at relatively high power, the engine would provide power exceeding the requested power, resulting in battery charge. Using the engine to charge the battery is inefficient at the system level due to the secondary energy conversion, albeit the engine's high efficiency. By comparison, co-optimization considers the vehicle dynamics and powertrain dynamics simultaneously and seeks to avoid the secondary energy conversion events as much as possible. However, as the passenger comfort requirement becomes more critical, it becomes harder to prevent the secondary energy conversion with co-optimization due to the need for the

[5] Although for numerical implementation, the vehicle following dynamics are pre-stabilized by the control input r, the actual command is the vehicle acceleration a.

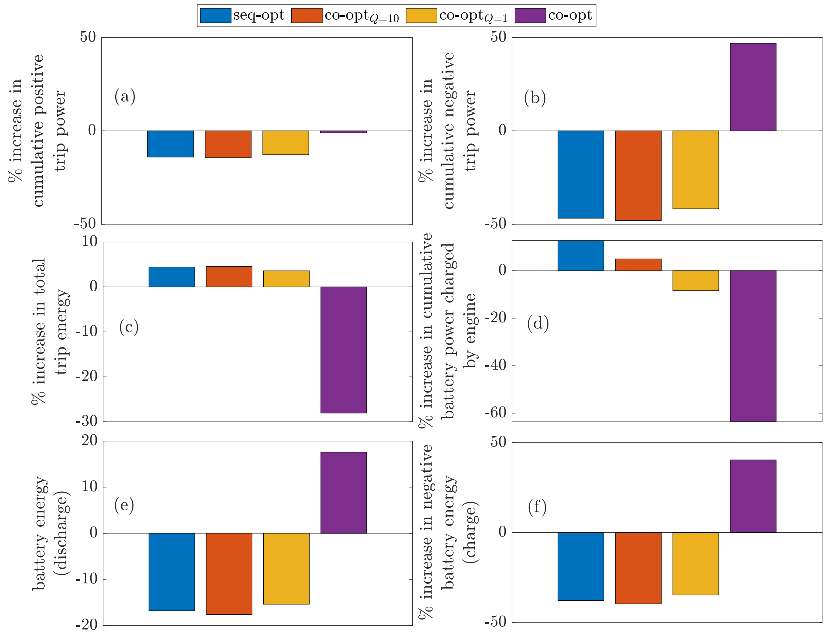


Figure 5.10: Trip energy analysis of the considered SOC depleting scenario.

engine to operate at a high-efficiency point. Consequently, the total fuel consumption increases with the smoothness of the vehicle's velocity.

A detailed trip energy analysis of the simulation results for the considered scenario is presented in Fig. 5.10. The percentage reduction of the energy metrics is computed against those from the baseline naturalistic human driving. It is noted that the trip energy refers to the cumulative driver demanded power over time. Figure 5.10(a) shows that the positive trip energy decreases as the velocity is smoothed out. Figure 5.10(b) shows that the smoother the velocity becomes, the less negative trip energy and thus regenerative braking potential could be extracted from the trip. As seen from Figs. 5.10(c) and (d), the co-optimization leads to less secondary energy conversion and improves system-wide efficiency. Compared to the baseline human-driven trace, the velocity profile through acceleration minimization induces a significant increase in the secondary energy conversion and reduces the system-wide efficiency.

As a bidirectional energy source, the battery can be charged and discharged. When charged, the charging power can either come from the negative driver demanded power (regenerative braking) or redundant engine power (secondary energy conversion). With the same initial and final battery SOC, the total battery energy spent would be sufficiently close to each other with different optimization strategies for the same trip. As a result, the more the battery is charged, the more positive battery energy would be available (discharge) to reduce fuel consumption. As can be seen from Fig. 5.10(e) and Fig. 5.10(f), the discharging battery energy increases, and the trip charging battery energy increases with co-optimization when the passenger comfort is not considered. However, as discussed above, the system-wide efficiency would reduce if the engine charges the battery. Consequently, the co-optimization seeks to shape the velocity profile within the required distance bound from the lead vehicle such that the potential in the negative trip energy could be maximized. This potential, however, cannot be realized when the driving trace is smoothed out.

5.7 Influence of Warm-Start on Convergence of DMIS

In simulations, it was observed that except for the case of the pure co-optimization case without penalty on driving discomfort, the resulting velocities are strongly affected by the warm-start vehicle-level control input trajectory $r^0 = (r_0 \, \cdots r_N)^0$ for a fixed number of single shooting iterations. In particular, in the presence of drivability requirement, DMIS exhibits faster convergence when warm-started from the vehicle-level control input obtained from the acceleration minimization result, as evident by Fig. 5.11. In both cases, the initial engine speed and torque sequences are the same. The observed slow convergence is also evident by the comparison between the DMIS and the multiple shooting in solving the acceleration minimization problem presented in Section 5.4.1. The number of (single shooting) iterations to achieve a similar solution quantity as with multiple shooting is considerably larger. It shows the underlying slow convergence issue when the vehicle-following dynamics (double integrator) are optimized with a velocity smoothing objective with DMIS.

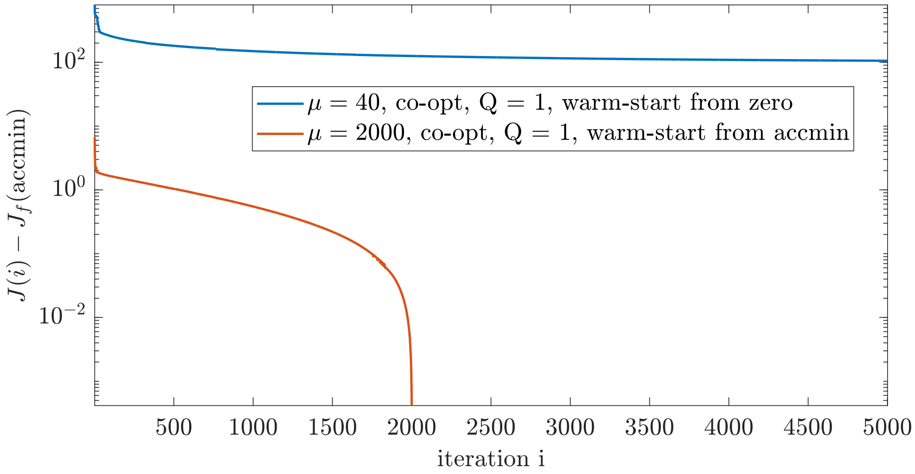
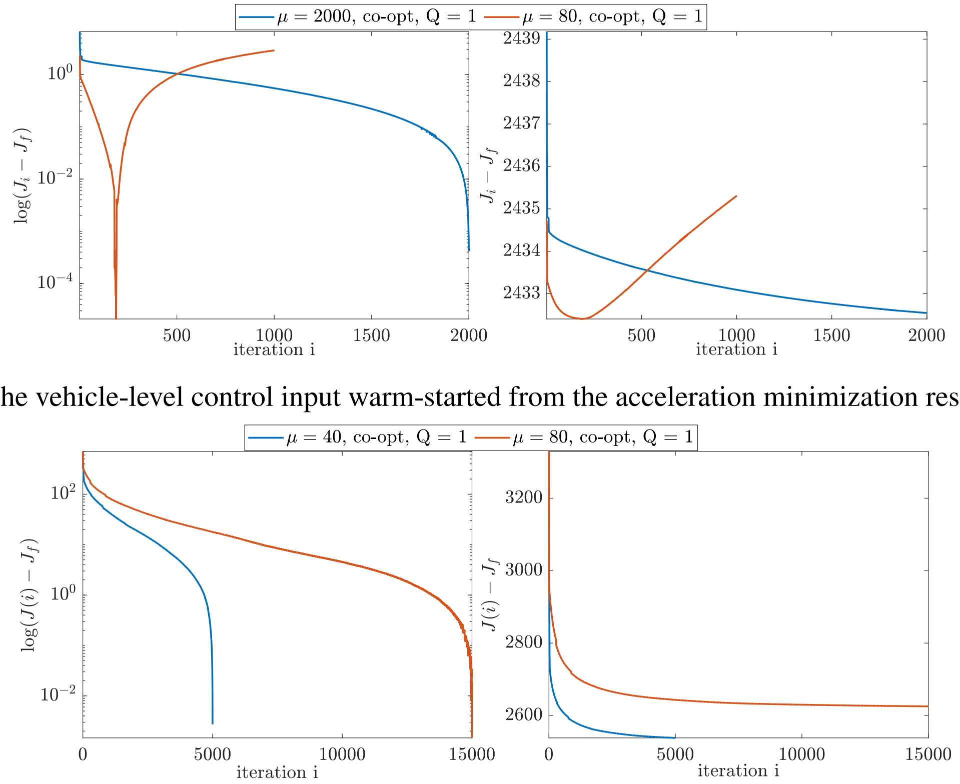


Figure 5.11: Comparison of behavior of the DMIS iterations under different warm-start strategies. The presented values are offset by the cost function value at the final DMIS iteration warm-started with acceleration minimization results.

In Section 5.3.1, it is shown that the values of the successive cost functions can always be decreased if the penalty weight μ takes sufficient high values. The importance of the penalty weight μ in controlling the behaviors of the DMIS iterations is illustrated in Fig. 5.12a. Note that the penalty weight is kept constant throughout the DMIS iterations in the co-optimization problems, rather than dynamically adaptive as in Algorithm 2, to guarantee a stabilized DMIS iterations due to the presence of vehicle-following dynamics. Such a constant μ is also used to obtain the numerical solution of (5.3a) to generate a comparable result to that with multiple shooting, as shown in Fig. 5.4.

The inequality relationship (5.62) indicates that the function values can always non-increase by choosing a sufficiently large penalty weight μ. As can be clearly observed from Fig. 5.12a, R in (5.62) for the considered co-optimization problem increases with DMIS iterations. As a result, the choice of the penalty weight $\mu = 80$ that guarantees a monotonically non-increasing cost function values is out-weighted by R after 200-iterations and fails to generate monotonic behavior as the red curve indicated in Fig. 5.12a. By comparison, when $\mu = 2000$, the cost function values are monotonically decreasing.



(a) The vehicle-level control input warm-started from the acceleration minimization results.

(b) The vehicle-level control input warm-started from zero.

Figure 5.12: Influence of the penalty weight μ on the values of the successive cost functions. J_i and J_f represent the cost function values at the i-th and final DMIS iterations, respectively.

For the considered co-optimization problem, Fig. 5.12a presents the behaviors of the DMIS iterations when the vehicle-level control sequence in DMIS is warm-started with that obtained from acceleration minimization. Fig. 5.12b presents the behaviors of the DMIS iterations when the vehicle-level control sequence in DMIS is warm-started with zero. By comparing the behaviors with the two initialization schemes, it can be seen that R when the vehicle-level control sequence is warm-started from zero is significantly smaller than that when the acceleration minimization-based warm-start sequence is used. As a result, a smaller penalty weight μ can be used and still generate monotonic behavior as shown in Fig. 5.12b. On the other hand, an increase in μ tends to slow down the convergence of the DMIS iterations, as can be observed in Fig. 5.12.

5.8 Summary

This section presented the Discrete Mixed-Integer Shooting (DMIS), a modified version of the single shooting algorithm in [5] for solving mixed-integer optimal control problems with terminal state equality constraints. The performance of the DMIS is benchmarked against state-of-art solvers on two benchmark problems with continuous and integer-valued control inputs. The computation time of the DMIS is shown to scale linearly as the problem horizon increases, compared to a typical exponential growth with branch and bound/cut algorithms. For our application to PHEV optimal control problems, first offline power-split optimization with the additional cranking state to address the chattering behaviors encountered in the PMP-based single shooting is solved with the proposed DMIS algorithm. A significant reduction in cranking fuel consumption is observed in simulation compared to the PMP-based single shooting without the cranking state.

Finally, the minimum fuel consumption problem for the PHEV by co-optimizing velocity and power-split operation is formulated and solved with the co-state backward-in-time propagation algorithm. Furthermore, the source of the fuel economy benefit with co-optimization is analyzed in detail.

CHAPTER 6

DMIS-Based Unified Receding-Horizon Fuel-Efficient Control Framework

6.1 Challenges in the Online Co-Optimization Implementation

Efforts toward online energy-efficient driving for (P)HEVs in the literature can be found in [16, 86, 87, 19, 21]. Most of the existing eco-driving approaches are based on a layered control framework that indirectly minimizes fuel consumption. Typically, each control layer is equipped with its objective function. However, although a layered control framework is advantageous for maintaining relative simplicity and modularity, there is no guarantee that the resulting system-wide efficiency is close to the maximum overall system-wide efficiency. As has been uncovered in Chapters 4 and 5, there is substantially additional potential in fuel economy when doing co-optimization compared to the layered optimization. The primary motivation of this chapter is to develop an online implementation strategy that co-optimizes the velocity and powertrain operations explicitly to achieve maximum system-wide efficiency while guaranteeing the safety and desired terminal SOC. Moreover, the proposed solution strategy is potentially implementable in real-time.

In general, for PHEVs the predominant roadblock to adopting a co-optimization control framework online originates from its numerical implementation. To be more concrete: to optimize the battery charge-depletion rate and the velocity for a PHEV, one needs to solve a trajectory optimization problem (TOP) for the entire trip with a specified SOC to be satisfied at the end of the trip. In the online implementation, the TOP is posed as an econominc model predictive control (EMPC) problem. To the authors' knowledge, the work of Huang et al., [3], is the first that succeeded in solving the TOP directly in real-time rather than tracking a reference explicitly. However, the work in [3] only considered the powertrain dynamics with a single battery SOC. For the co-optimization problem, due to additional controls and states, the real-time implementability is unclear when the EMPC aims to cover the entire trip. Besides, co-optimization relies on predicting a lead vehicle's driving trace to formulate its feasible search space. Such a prediction will become less accurate as the prediction horizon increases. As a result, the performance benefit with prediction could be nullified due to the discrepancy between the actual and predicted feasible search space.

Unlike relatively deterministic road information, forecasting human drivers' decisions accompanies high uncertainty in the optimization process. The controller needs to leverage the speed previews with different accuracy over short and long prediction horizons, the computational demands, and the performance degradation for the whole trip when considering only a limited prediction horizon. Among various efforts over the years, a multi-horizon model predictive controller has received particular attention [88, 89, 90] for integrated power and thermal management. In [88], the scheduling layer MPC with a long horizon generates reference torque and SOC trajectories used in the short horizon MPC to track. In [90] the fuel consumption over the remaining trip is approximated with the long-horizon layer. However, vehicle dynamics are not optimized; hence the prediction accuracy would not affect the safety (position) constraint, as is the case for our problem in the presence of traffic flow. Despite being updated at a low sampling rate, the long-horizon MPC layer still induces an extra computation burden. In light of the principle of optimality [91], the long-horizon MPC layer essentially serves to approximate the cost-to-go for the considered problem. When the route is driven repeatedly, the cost-to-go can be learned offline in a data-driven fashion through offline optimization.

This chapter expands the discussions on the connection between the DMIS and approximate policy iteration (approximate-PI) in the reinforcement learning (RL) application, integrating the numerical optimization strategy with the online implementable framework. Henceforward, the PT-only control of a human-driven to the combined VD and PT control of an automated PHEV are unified under the same DMIS-based framework. In the unified framework, the cost-to-go (the fuel consumption as the economic cost) is represented by the co-state associated with the SOC dynamics. This co-state is corrected both internally using a single shooting method and externally based on the co-state temporal difference (TD)-error. In its application to automated PHEVs, a control barrier function (CBF) is augmented as an add-on block to modify the vehicle level control input to be executed whenever necessary. The unified control framework guarantees safe vehicle-following behavior when including longitudinal vehicle dynamics control. The unified framework allows for systematically evaluating the fuel economy and drivability performance of different levels and structures of optimization strategies, detailed in Chapter 7.

6.2 Connection between DMIS and Approximate Policy Iteration (PI)

The DMIS algorithm presented in Chapter 5, based on a Discrete Maximum Principle (DMP), consists of a backward and a forward path, tightening itself closely to the policy evaluation and improvement steps in the approximate-PI. This section is devoted to establishing the connection between the approximate-PI and the DMIS. In light of the connection, the DMIS is not just the numerical strategy in solving the underlying optimization; it also inspires an online implementation strategy. A DMIS-based receding-horizon framework is then established, uniting the PT-only control of a human-driven and the combined VD and PT control of an automated PHEV. This will be the focus of the next section.

6.2.1 Forward Path in DMIS and Policy Improvement

In the forward path of DMIS, the control at time t in the $(i+1)$-th shooting iteration is chosen as an element minimizing the Hamiltonian H (dependency on external parameters w_t is omitted to make the equations concentrated), with the entire co-state sequence $\mathcal{P}^i = (p_0^i, \cdots, p_{N_f}^i)$ obtained from the i-th shooting iteration

$$u_t^{*,i+1} \in \arg\min_{u\in\mathcal{U}} H(x_t^{*,i+1}, u, p_{t+1}^i) = \arg\min_{u\in\mathcal{U}} \{l(x_t^{*,i+1}, u) + (p_{t+1}^i)^T f(x_t^{*,i+1}, u)\}, \quad (6.1a)$$

where

$$H(x_t^{*,i+1}, u_t^{*,i+1}, p_{t+1}^i) = \min_{u\in\mathcal{U}} H(x_t^{*,i+1}, u, p_{t+1}^i). \quad (6.1b)$$

The system dynamics are then propagated forward-in-time

$$x_0^{*,i+1} = x_0, \;\; x_{t+1}^{*,i+1} = f(x_t^{*,i+1}, u_t^{*,i+1}). \quad (6.1c)$$

The results of the forward pass in the $(i+1)$-th shooting iteration are the state and control sequences (policy)

$$\mathcal{X}^{*,i+1} = (x_0, x_1^{*,i+1}, \cdots, x_{N_f}^{*,i+1}), \quad \mathcal{U}^{*,i+1} = (u_0^{*,i+1}, \cdots, u_{N_f-1}^{*,i+1})$$

to be used in the backward pass in $(i+1)$-th shooting iteration. It can be seen that (6.1) is equivalent to the policy improvement step in approximate-PI when we consider $\mu^{i+1} =$

$(\mu_0^{i+1}, \cdots, \mu_{N_f-1}^{i+1})$ with

$$\mu_t^{i+1} = \arg\min_{u \in \mathcal{U}} l(x,u) + \hat{V}_{t+1}^{\mu^i}(f(x,u)), \tag{6.2a}$$

where the approximate cost-to-go is in the form of

$$\hat{V}_{t+1}^{\mu^i}(f(x,u)) = (p_{t+1}^i)^T f(x,u), \tag{6.2b}$$

where $p_{t+1}^i := p_{t+1}^{\mu^i}$ depends on the policy μ^i. In (6.2b), a potential constant offset term c on the right hand-side of (6.2b) can be omitted since it does not affect the choice of the control input.

6.2.2 Backward Path in DMIS and Policy Evaluation

With the control and state sequence $\mathcal{U}^{*,i+1}, \mathcal{X}^{*,i+1}$ obtained in the $(i+1)$-th shooting iteration, the $(i+1)$-th backward path proceeds as

$$\begin{aligned} p_{N_f}^{i+1} =& \begin{cases} 0 & \text{if the requirement is } x_{N_f} \in \mathcal{X}_f \\ p_{N_f}^i - [\frac{\partial c(\cdot)}{\partial p_{N_f}}|_{p_{N_f}^i}]^{-1} c(\cdot)^{i+1} & \text{if the requirement is } c(x_{N_f}, x_f) = 0 \end{cases} \\ p_t^{i+1} =& \frac{\partial H}{\partial x}|_{x_t^{*,i+1}, u_t^{*,i+1}, p_{t+1}^{i+1}} = \frac{\partial l(x,u)}{\partial x}|_{x_t^{*,i+1}, u_t^{*,i+1}} + \Big(\frac{\partial f(x,u)}{\partial x}|_{x_t^{*,i+1}, u_t^{*,i+1}}\Big)^T p_{t+1}^{i+1}, \end{aligned} \tag{6.3}$$

where $c(x_{N_f}, x_f)$ denotes the terminal state equality constraints. The terminal set $\mathcal{X}_f$ is defined by $g_f(x_f) \leq 0$. Note that if $g_f(x_f) = 0$ (terminal inequality constraints become active), the terminal co-state p_{N_f} is no longer zero, but is rather determined by $\frac{\partial g_f(x_f)}{\partial x_f}^T \nu_f$, with $\nu_f \geq 0, \nu_f^T g_f(x_f) = 0$ as in [65].

Assumption 6-$\mathcal{A}$. It is assumed that the terminal inequality constraints are inactive.

Note that p_{N_f}-s are initialized differently in (6.3), depending on whether a terminal inequality $(x_{N_f} \in \mathcal{X}_f)$ or a terminal equality $(c(x_{N_f}, x_f) = 0)$ constraint is present. These two types of initialization cover all possible conditions for the problem in consideration due to an absence of the terminal cost as in (6.8a) under Assumption 6-$\mathcal{A}$. The results of the backward path is the co-state sequence $\mathcal{P}^{i+1} = (p_0^{i+1}, \cdots, p_{N_f}^{i+1})$. Recall from (6.2) that the focus is on approximating the cost-to-go sequence $\mathcal{V} = (V_0, \cdots, V_{N_f})$ on the subspace $\hat{\mathcal{S}} := \{\mathcal{P}^T \mathcal{X} := (p_0^T x_0, \cdots, p_{N_f}^T x_{N_f})\} \subset \mathbb{R}^{N_f+1}$. Restricting the partial derivative of the Hamiltonian H with respect to the state x at time t to be the co-state p_t, the temporarily improved cost-to-go is projected back to the subspace $\hat{\mathcal{S}}$.

The forward (6.1) and backward (6.3) paths can be written compactly as

$$\mathcal{P}^{i+1} = \hat{\Pi} \circ \hat{T}_{\mu^i}\big(\mathcal{P}^i\big), \tag{6.4}$$

where $\hat{T}^i_\mu : \mathcal{P}^i \mapsto (\mathcal{X}^i, \mathcal{U}^i)$ corresponds to the forward path (6.1), where at i-th iteration, for a given sequence of co-state $\mathcal{P}^i$, the system is propagated forward-in-time with the control input μ^i_t chosen to minimize the instantaneous Hamiltonian. It corresponds to the *policy improvement step* in approximate-PI. $\hat{\Pi} : (\mathcal{X}^i, \mathcal{U}^i) \mapsto \mathcal{P}^{i+1}$ corresponds to the backward path, where the co-state dynamics are propagated backward-in-time as in (6.3) with the given state $\mathcal{X}^{*,i}$ and control $\mathcal{U}^{*,i}$ sequences obtained resulting from the previous policy μ^i. It corresponds to the *policy evaluation step* in approximate-PI. As a result, DMIS [85] seeks to find the sequence of co-states $\mathcal{P} = (p_0 \ \cdots \ p_{N_f})$ to be a fixed point under the projected Bellman operator $\hat{\Pi} \circ \hat{T}_\mu$ in the form of the co-state dynamics

$$\mathcal{P} = \hat{\Pi} \circ \hat{T}_\mu\big(\mathcal{P}\big), \tag{6.5}$$

to approximate the fixed-point of the optimal cost-to-go sequence in view of the principle of optimality

$$\hat{\mathcal{V}} = \mathcal{F}(\hat{\mathcal{V}}) \tag{6.6a}$$

with $\mathcal{V} : \mathcal{X} \mapsto \mathcal{V}(\mathcal{X}) = (V_0(x_0) \ V_1(x_1) \ \cdots \ V_{N_f}(x_{N_f})) \in \mathbb{R}^{N_f+1}$, where V_{N_f} is given, and $\mathcal{F} : \mathcal{V} \mapsto \mathcal{V}$ is defined with

$$V_t = \min_{u \in \mathcal{U}} \{ l(x_t, u) + V_{t+1}(f(x_t, u)) \}, \quad \forall t \in \{0, ..., N_f - 1\}. \tag{6.6b}$$

Comparing (6.5) with (6.6) it can be seen that (6.5) is an resemblance of the fixed point of the projected Bellman operator

$$\hat{\mathcal{V}}(\mathcal{X}) = \Pi \circ T\big(\hat{\mathcal{V}}(\mathcal{X})\big), \tag{6.7a}$$

where $\hat{\mathcal{V}}(\mathcal{X})$ is the projection of $\mathcal{V}(\mathcal{X})$ onto the approximation subspace $\mathcal{S} = (S_0 \ \cdots \ S_{N_f}) \subset \mathbb{R}^{N_f+1}$ defined by

$$S_t = \{ r_t^T \phi_t(x_t) | r_t \in \mathbb{R}^s \}, \tag{6.7b}$$

where $\phi_t(x_t) \in \mathbb{R}^s$ is the feature vector, consisting of polynomials in x of potentially infinite order s. r_t is the weight vector. As discussed in [92], allowing for different features $\phi_t(x_t)$ and different parameter vectors r_t for each stage t is necessary for nonstationary problems (e.g., if the state space changes over time[1]). The connection between $\mathcal{S}$ and $\hat{\mathcal{S}}$ can be established by reformulating the linear approximation structure with the co-state

$$\hat{V}_t(x_t) = r_t^T \phi_t(x_t) = p_t^T x_t + c. \tag{6.7c}$$

Note the last equality holds exactly since $\phi_t(x_t)$ consists of polynomial functions. As already discussed, the constant term c can be omitted.

The resulting stationary policy μ corresponds to the optimal control sequence $\mathcal{U}^* = (u_0^*, \cdots, u_{N_f-1}^*)$.

6.3 A Unified DMIS-Based Framework for Fuel-Efficient Control

6.3.1 Minimum Fuel Consumption Problems Formulated Offline

The following three types of problems are defined directly in discrete-time, representing PHEV fuel-efficient control with an increased level of automation and cooperation.

1. The power-split optimization problem (*ps-opt*): The considered PHEV is assumed to driven by a human, thus we can only control its powertrain operation.

2. The sequential optimization problem (*sequential-opt*): The powertrain and vehicle following dynamics are considered separately.

3. The co-optimization problem (*co-opt*): The powertrain and vehicle following dynamics are considered simultaneously.

In all three types of optimization problems to be considered, a minimum fuel consumption problem needs to be solved.

$$\text{minimize} \sum_{k=0}^{N_f-1} l(x_k, u_k, w_k), \tag{6.8a}$$

where w_k is the disturbances, which is the ego vehicle's velocity and torque demand for *ps-opt*, and the lead vehicle's velocity/position for *sequential-opt* and *co-opt*. (6.8a) is subject to

[1] This is exactly the case with the PHEV fuel-efficient control problems considered in this thesis. The total distance traveled by the PHEV is non-decreasing and the battery SOC is non-increasing in terms of the entire trip (also locally SOC can increase or decrease)

system dynamics (presented in previous sections), written in a compact form as

$$x_{k+1} = f(x_k, u_k), \tag{6.8b}$$

the control constraints

$$[te_k, ne_k] \in \Omega_k \cup [0, 0]^T, \;\; E_k \in \{0, 1\}, \tag{6.8c}$$

and the desired terminal SOC, SOC$_f$

$$\text{SOC}_{N_f} = \text{SOC}_f. \tag{6.8d}$$

The stage cost $l(\cdot)$ in (6.8a) is defined as

$$l(\cdot) = \underbrace{\dot{m}_{f,k} \cdot \Delta t}_{\text{static fuel}} + \underbrace{m_c \cdot \max(\tilde{e}_{k+1} - \tilde{e}_k, 0)}_{\text{cranking fuel}}, \tag{6.8e}$$

whereas in the co-optimization problem, additional penalty terms are augmented to (6.8e) to account for driving safety and comfort

$$\begin{aligned} l_{\text{coopt}}(\cdot) =& l(\cdot) + Q \cdot a_k^2 \\ &+ Q_s \cdot \underbrace{\max\left(s_k^l - s_k - d_0 - \tau_{max} v_k, 0\right)^2}_{\text{maximum time headway violation}} + Q_s \cdot \underbrace{\max\left(-s_k^l + s_k + d_0 + \tau_{min} v_k, 0\right)^2}_{\text{minimum time headway violation}} \\ &+ Q_v \cdot \underbrace{\max\left(-v_k, 0\right)^2}_{\text{forward driving violation}} \\ &+ Q_a \cdot \underbrace{\max\left(a_k - a_k^{\max}, 0\right)^2}_{\text{maximum acceleration violation}} + Q_a \cdot \underbrace{\max\left(-a_k + a_k^{\min}, 0\right)^2}_{\text{minimum acceleration violation}} \\ &+ Q_{\Delta a} \cdot \underbrace{\max\left(\Delta a_k - \Delta a_k^{\max}, 0\right)^2}_{\text{maximum jerk violation}} + Q_{\Delta a} \cdot \underbrace{\max\left(-\Delta a_k + \Delta a_k^{\min}, 0\right)^2}_{\text{minimum jerk (backward direction) violation}} \quad . \end{aligned} \tag{6.8f}$$

The first line in $l_{\text{coopt}}(\cdot)$ is the economic cost, including the fuel consumption and the cranking fuel, and potentially the additional acceleration minimization to prompt driving smoothness. The second and the third lines penalize the maximum and minimum time-headway violation. The fourth line penalizes the violation of forward driving. The fifth and sixth lines penalize the violations of the maximum and minimum acceleration constraints. The seventh and eighth lines penalize the violations of the maximum and minimum jerk constraints. In (6.8e), $\Delta a_k = a_k - a_{k-1}$ is the jerk. $Q, Q_s, Q_v, Q_a, Q_{\Delta a}$ are the tunable weights on the corresponding constraint violations. Note that all state constraints and state-and-control mixed constraints are handled by augmenting penalty terms to the state cost l. This way, only pure control constraints are considered as hard constraints. Such a formulation facilitates numerical computation by removing the need to consider jumps in the co-state. To strictly enforce these state and mixed constraints to further improve driving safety, a CBF-based strategy in Section 6.3.6.

Note that an additional optimization problem is considered in *sequential-opt*

$$\begin{aligned} &\text{minimize} \sum_{k=0}^{N_f-1} a_k^2 \Delta t, \\ &\text{subject to (4.2)} \\ &\text{and constraints summarized from line 2-6 in (6.8f).} \end{aligned} \tag{6.9}$$

6.3.2 From EMPC to Discounted EMPC Formulation

Now the focus is shifted from the offline trajectory optimization to its online implementation. The core fuel minimization problem to be solved in a receding horizon manner at time t is denoted as $\mathbf{P}_t^{N_f}(\text{SOC}_f)$[33],

$$\mathbf{P}_t^{N_f}(\text{SOC}_f)\colon \ \min \ J_{N_f|t} = \sum_{k=0}^{N_f-1} l_\star(x_{k|t}, u_{k|t}), \tag{6.10a}$$

with $\star \in \{\textit{ps-opt}, \textit{sequential-opt}, \textit{co-opt}\}$, and the corresponding dynamics, constraints formulated with predicted lead vehicle's trajectory and a desired terminal SOC requirement

$$\text{SOC}(N_f|t) = \text{SOC}_f. \tag{6.10b}$$

As discussed previously, there is no explicit reference trajectory to track in (6.10), necessitating the consideration of the entire trip due to (6.10b). For a PHEV, the entire length N_f is in general long (hours), making it computationally demanding to solve $\mathbf{P}_t^{N_f}(\text{SOC}_f)$ directly. Moreover, solving $\mathbf{P}_t^{N_f}(\text{SOC}_f)$ does not necessarily guarantee a fuel economy gain due to the large uncertainty into far-future.

Instead, the following discounted receding horizon problem is considered at time t, with a horizon length of N_p

$$\min \ J_{N_p|t}^{\rho} = \sum_{k=0}^{N_p-1} \rho^k l_\star(\cdot) + \rho^{N_p} \hat{V}_{N_p|t}(\text{SOC}_{N_p|t}), \tag{6.11}$$

where $\hat{V}_{N_p|t}(\text{SOC}_{N_p|t})$ is an approximation of the (N_p-step ahead) terminal cost at time t, and

$\rho \in (0, 1]$ is a factor to discount farther future rewards/costs, which corresponds less accurate predictions. Note that in *sequential-opt*, the velocity and acceleration trajectories are optimized before (6.11) through solving a N_p-step version of the acceleration minimization problem with the lead vehicle's trajectory predicted to formulate the position constraints. Therefore, (6.11) for the *sequential-opt* is essentially the same as the *ps-opt*, except with different velocity and acceleration trajectories.

In particular, this section considers the approximated terminal cost $\hat{V}_{N_p|t}(\text{SOC}_{N_p|t})$ in (6.11) to only depend on the predicted terminal SOC, $\text{SOC}_{N_p|t}$. The reasons are as follows. 1) There is a desired terminal SOC (6.10b) at the end of the trip. In view of the principle of optimality, it is desirable to drive the predicted SOC ($\text{SOC}_{N_p|t}$) to $\text{SOC}^*_{N_p+t}$ that is optimal to the actual remaining trip. 2) In view of the original EMPC problem (6.10), neither terminal cost nor terminal equality constraints are imposed on other states ($\tilde{e}_t$, or additionally, s_t, v_t), except for the persistent time-varying state constraints. Based on Assumption 6-$\mathcal{A}$, it is known in the classical optimal control theory that the terminal co-state is zero in the absence of both terminal cost and inequality constraints as indicated in the first line of (6.3).

6.3.3 Terminal Cost Approximation

This section examines approximating the terminal cost $\hat{V}_{N_p|t}(\text{SOC}_{N_p|t})$ in (6.11) with a particular form, and discuss how an initial terminal cost approximation $\hat{V}^0_{N_p|t}$ is improved while solving the (6.11) numerically using DMIS (6.2), (6.3) through iterative refinement.

First, when applying DMIS to solve the underlying optimization problem (6.8) *offline*, the terminal SOC is achieved by iteratively adjusting the SOC-associated terminal co-state $p_{N_f,\text{SOC}}$ until the simulated terminal SOC is in the vicinity of its desired value. When the DMIS iterations converge, this terminal co-state will converge to $p^*_{N_f,\text{SOC}} := p^*_{N_f,\text{SOC}}(\text{SOC}_f)$, as a function of the required terminal SOC, SOC_f. Afterward, the DMIS iterations become stationary, and (6.8) is equivalent to

$$\text{minimize } J_{N_f} = \phi^*(\text{SOC}_{N_f}) + \sum_{k=0}^{N_f-1} l(x_k, u_k, w_k), \tag{6.12}$$

where $\phi^*(\text{SOC}_{N_f}) = p^*_{N_f,\text{SOC}} \cdot \text{SOC}_{N_f}$ is the (optimal) terminal cost. without the terminal state constraint (6.8d).

Assumption 6-$\mathcal{B}$. The terminal SOC constraint, SOC_f, and the terminal co-state $p^*_{N_f,\text{SOC}}$ at the fixed-point of our iterations (6.5) have (at least locally) a one-to-one relationship.

Assumption 6-$\mathcal{B}$ is implicitly assumed in (6.3) when applying DMIS backward path, and is observed (at least locally) in simulations for our application. The role of the optimal Lagrange multiplier vector (i.e. the co-state) as a replacement of the primal terminal constraints in EMPC is also pointed out in a recent work [93]. As seen by comparing (6.8a), (6.8d) and (6.12), the information of the terminal state constraint is embedded in the terminal co-state. Compare (6.12) with (6.11), one can recognize that the co-state is functionally equivalent to a resemblance of the terminal cost, with the optimal cost-to-go defined as $V^*_{N_f} = p^*_{N_f,\text{SOC}} \cdot \text{SOC}_{N_f}$. Solving (6.12) results in the same optimal state, control (primal) and co-state (dual) sequence, $\mathcal{X}^*_{N_f} = (x_0, x^*_1, \cdots, x^*_{N_f})$, $\mathcal{U}^*_{N_f} = (u^*_0, \cdots, u^*_{N_f-1})$, $\mathcal{P}^*_{N_f} = (p^*_0, \cdots, p^*_{N_f})$ respectively, as solving (6.8a) and (6.8d).

When shifting to the online implementation, as discussed in the previous section, $\text{SOC}_{N_p|t}$ ideally should be driven to a state $\text{SOC}^*_{N_p+t}$ optimal to the remaining trip in view of the principle of optimality. Denote $p^*_{N_p|t,\text{SOC}}(\text{SOC})$ as the optimal terminal co-state when solving (6.11), except that the terminal cost is replaced with an equivalent terminal SOC constraint $\text{SOC}_{N_p|t} = \text{SOC}^*_{N_p+t}$. Then based on the discussions above, ideally, we pursue approximating the terminal cost as

$$\hat{V}^*_{N_p|t}(\text{SOC}_{N_p|t}) = p^*_{N_p|t,\text{SOC}}(\text{SOC}^*_{N_p+t}) \cdot \text{SOC}_{N_p|t}. \tag{6.13}$$

Note that (6.13) holds only when DMIS iterations converge. In the actual implementation, we initialize the approximation $\hat{V}^0_{N_p}(\text{SOC}_{N_p|t})$ in (6.11) with $p^0_{N_p|t,\text{SOC}} \cdot \text{SOC}_{N_p|t}$, and update $p^i_{N_p|t,\text{SOC}}$ iteratively through (6.1) to (6.3) to achieve (6.13). The iterative update in $p^i_{N_p|t,\text{SOC}}$ towards $p^*_{N_p|t,\text{SOC}}(\text{SOC}^*_{N_p+t})$ is similar to the parameter update step if a parameterized function $\hat{V}_\theta$ is to approximate the cost-to-go.

However, there are two potential issues. First, in the online implementation, $\text{SOC}^*_{N_p+t}$ is not known a priori because the future trajectory is uncertain as the future lead vehicle's trajectory is uncertain and, in turn, affects the feasible region defined by the position constraints. As a result, in reality, even if the DMIS iterations converge, one might only get $p^*_{N_p|t,\text{SOC}}(\text{SOC}^*_{N_p+t})$ at best, with $\text{SOC}^*_{N_p+t}$ being an estimation of the optimal $\text{SOC}^*_{N_p+t}$ provided $\text{SOC}^*_{N_p+t}$ being a feasible terminal constraint. Second, we are motivated to design algorithms for potential real-time implementation. Also, the estimation $\text{SOC}^*_{N_p+t}$ might even not be a feasible terminal constraint for DMIS to enforce convergence[2]. Therefore, only a fixed-number of DMIS iterations (maxiter) are performed per MPC step. As a result, one will eventually get

[2] To see this, one can consider a rather extreme case where the prediction horizon $N_p = 10$-sec. The current SOC(t) = 0.85, whereas $\text{SOC}^*_{N_p+t}$ = 0. It is physically impossible to completely deplete the battery within 10-sec under a normal driving condition. Therefore, the N_p-horizon optimization problem with terminal SOC constraint to be $\text{SOC}^*_{N_p+t}$ is infeasible. However, even under such a circumstance it is still possible to perform a maxiter number of DMIS iterations.

$V_{N_p|t}(\text{SOC}_{N_p|t}) = p^{\text{maxiter}}_{N_p|t,\text{SOC}}(\text{SOC}^*_{N_p+t}) \cdot \text{SOC}_{N_p|t}$ and the state, control and co-state sequence within the prediction horizon

$$\begin{aligned}
\mathcal{X}^{*,\text{maxiter}}_{N_p|t} &= (x_t, x^{*,\text{maxiter}}_{1|t}, \cdots, x^{*,\text{maxiter}}_{N_p|t}) \\
\mathcal{U}^{*,\text{maxiter}}_{N_p|t} &= (u^{*,\text{maxiter}}_{0|t}, \cdots, u^{*,\text{maxiter}}_{N_p-1|t}) \\
\mathcal{P}^{\text{maxiter}}_{N_p|t} &= (p^{\text{maxiter}}_{0|t}, \cdots, p^{\text{maxiter}}_{N_p|t}).
\end{aligned}$$

The control input u^{MPC}_t to be executed by the MPC controller at time t is define as $u^{\text{MPC}}_t := (u^*_{0|t})^{\text{maxiter}}$, where

$$(u^*_{0|t})^{\text{maxiter}} = \arg\min H^{\text{maxiter}}_{0|t} = \arg\min l(x_t, u, w_t) + (p^{\text{maxiter}}_{1|t})^T f(x_t, u). \tag{6.14}$$

It can be seen that the optimality of $(u^*_{0|t})^{\text{maxiter}}$ depends on the optimality of co-state $p^{\text{maxiter}}_{1|t}$. If $p^{\text{maxiter}}_{1|t} = p^*_{t+1}, \forall t$, with p^*_{t+1} extracted from $\mathcal{P}^*_{N_f}$, the resulting control action u^{MPC}_t executed by MPC is optimal. In the online implementation, however, it is not possible to guarantee $p^{\text{maxiter}}_{1|t} = p^*_{t+1}$. It is therefore the target of this section to improve the optimality of $p^{\text{maxiter}}_{1|t}$ with the MPC implementation.

As now becoming clear from (6.14), under the DMIS algorithm, the true utility of the MPC problem is to improve the optimality of the co-state through multi-step look-ahead. This well echos the advantage of multi-step look-ahead in improving the performance of RL [92, 94]. The strategy to improve the optimality of $p^{\text{maxiter}}_{1|t}$ *internally* within DMIS iterations, and *externally* with TD-error correction is the topic of next section.

6.3.4 Underlying Markov Decision Process and Time-Based Sampling

So far, we have considered the minimum fuel consumption problems on a particular trip, which is rather deterministic. It is also possible to model the real system we want to control as a Markov Decision Process (MDP) having the (possibly) stochastic state transition dynamics $\mathbb{P}[s_+|s, a]$, where s and a denote the states and controls, respectively. For the co-optimization problem[3], the uncertainty comes from the lead vehicle's remaining distance (to destination) and velocity. The state $s = [d\ v\ \tilde{e}\ \text{SOC}]^T$ includes the ego PHEV's 1) remaining distance to destination, 2) current velocity, 3) engine cranking state, and 4) battery SOC (i.e. the remaining electricity). The control a includes 1) the normalized reference for the vehicle speed control, 2) the engine cranking command, 3) the engine speed, and 4) the engine torque. The stage cost associated with the MDP is the fuel consumed between time steps, $l(s, a) = \Delta m_f$.

[3] Although this section mainly discusses the co-optimization problem, extensions to the power-split optimization and sequential optimization should be straightforward.

Remark 7. Note that the remaining distance to destination instead of the absolute distance travelled is considered in the state. This enables the convenient description of the terminal state for a considered trip as $s_f = [0 \; v \; \tilde{e} \; \text{SOC}_f]^T$.

The performance of a policy π at a state the average discounted reward accumulated by following the policy from the state

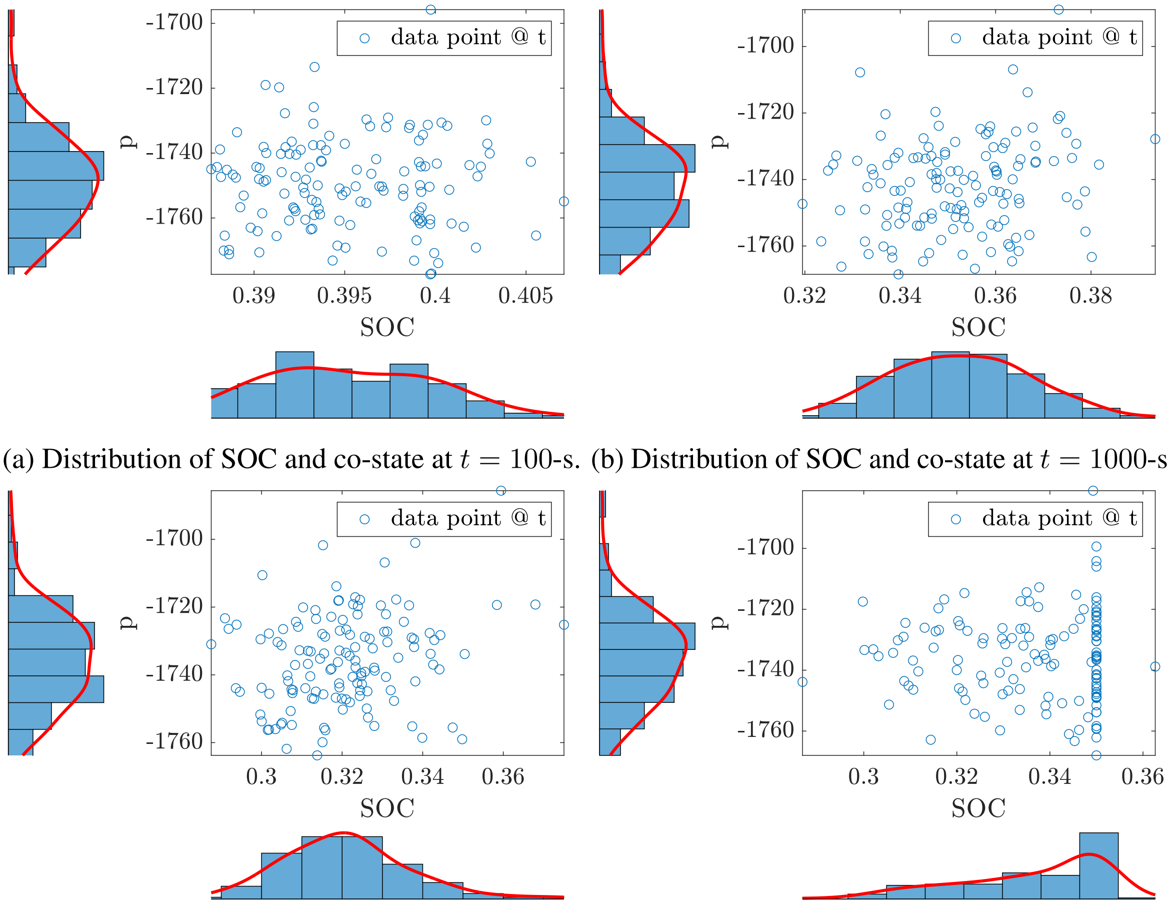
$$V^\pi(s) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \rho^t l(s_t, \pi(s_t)) | s_0 = s\Big], \tag{6.15}$$

which characterizes the (discounted) total fuel consumption over the remaining trip.

However, note that the entire continuous state space $s \in [0, d^{\max}] \times [0, v^{\max}] \times [0, 1] \times [0, 1]$ is too large to work on and it is hard to design an efficient value function approximator valid for the entire state space. For stationary problems with a long or infinite horizon as in (6.15), where the state space does not change with t, it is common to use the same features and parameters for all stages [92]. Following (6.7c), the value function is approximated by $\hat{V}(s) = r^T\phi(x) = p^T x$ (x is almost the same as s except that the remaining distance is considered in s), and based on our previous discussion, p is the co-state. As discussed in Section 6.2.2, the problem is non-stationary for a particular trip since the state space changes over time. It indicates that the focus should be trajectory-centric rather than individual state-centric.

In the online execution phase, the remaining distance to the destination is non-increasing, and the battery SOC is non-increasing in terms of the entire trip (also locally SOC can increase or decrease). As a result, in the online implementation, at time t, the experience between $[0, t-1]$ with associated state trajectory $s_{[0,t-1]}$ will be hard to be utilized since they will hardly ever be revisited in the remaining trip. As a result, it is desirable to get a good ”ground-truth” trajectory from the offline training phase. The above discussions motivate using repeated trips with sufficient variations on the same route for the offline training phase. It is observed that the state and co-state at each time instant roughly obey Gaussian distribution. As a result, the ground-truth value is sampled from an underlying Gaussian distribution at each time instant, assuming the mean value is the nominal trajectory. Note that for the considered PHEV fuel-efficient control problems, it might also be possible to get the nominal state and co-state trajectories in a distance domain. Meanwhile, there have been ongoing discussions on whether the ground-truth values should be taken as samples from an underlying distribution or just an interpolation on the nominal trajectories. The proper choice of the nominal trajectories and how the nominal trajectories should be used online are subject to further investigations.

Figure 6.1 and 6.2 illustrate the resulting distribution of the SOC and their associate co-state at different time steps on the considered repeated Ann Arbor trips with power-split optimization

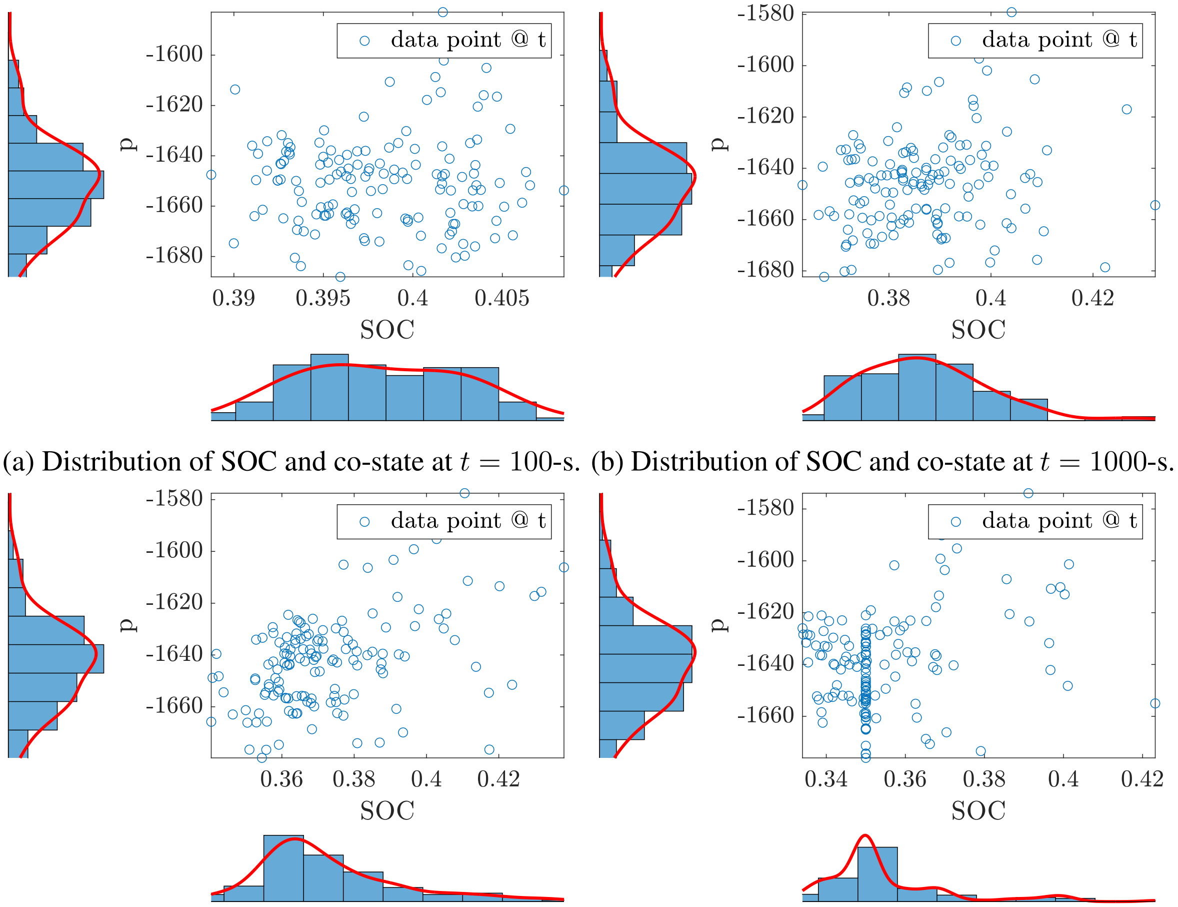


(a) Distribution of SOC and co-state at t = 100-s. (b) Distribution of SOC and co-state at t = 1000-s.

(c) Distribution of SOC and co-state at t = 1500-s. (d) Distribution of SOC and co-state at t = 1800-s.

Figure 6.1: Distribution of SOC and co-state of power-split optimization (*ps-opt*) on the repeated Ann Arbor trips.

and co-optimization, respectively. The total trip lengths vary from ∼1500s to ∼2300s. Detailed trip statistics are presented in Fig. 7.5 in Chapter 7. The initial SOC and terminal SOC values are $\text{SOC}_0 = 0.4, \text{SOC}_f = 0.35$, respectively, in all offline optimization results. As can be observed from Figure 6.1 and 6.2, especially in the middle of the considered trips, the distribution of state and co-state at each time instant roughly follows a normal distribution around a nominal trajectory. As will be seen from (6.16a) and (6.16b) below, in the online implementation, the ”ground-truth” state and co-state are sampled from normal distributions with the average values specified by the nominal state and co-state trajectories. As will be discussed further in Chapter 7, the nominal state and co-state trajectories are average trajectories of those obtained from the offline optimization. However, the distribution is less likely to be a normal distribution at the beginning and towards the end of the trip due to the initial and terminal SOC requirements. It indicates a possible more reasonable way to obtain nominal trajectory: sample from a normal distribution of the initial and terminal SOC, do offline optimization, and get average trajectories,



(a) Distribution of SOC and co-state at $t = 100$-s. (b) Distribution of SOC and co-state at $t = 1000$-s.

(c) Distribution of SOC and co-state at $t = 1500$-s. (d) Distribution of SOC and co-state at $t = 1800$-s.

Figure 6.2: Distribution of SOC and co-state of co-optimization (*co-opt*) on the repeated Ann Arbor trips.

instead of a fixed, same initial and terminal SOC. Also, right now, the weight of each trajectory is the same. It might be better to consider different weights on different trajectories to get nominal trajectories. These will be further investigated in future work.

6.3.5 Improving Terminal Cost in EMPC

At time t, consider (6.11) with an initial guess of the terminal cost $\hat{V}^0_{N_p|t}(\mathrm{SOC}_{N_p|t}) = p^0_{N_p|t} \cdot \mathrm{SOC}_{N_p|t}$. Warm-started with the shifted predicted state and control sequences $\mathcal{X}^0_{N_p|t} = (x^{*,\text{maxiter}}_{1|t-1}, \cdots, x^{*,\text{maxiter}}_{N_p|t-1}, x^{*,\text{maxiter}}_{N_p|t-1})$ and $\mathcal{U}^0_{N_p|t} = (u^{*,\text{maxiter}}_{1|t-1}, \cdots, u^{*,\text{maxiter}}_{N_p-1|t-1}, u^{*,\text{maxiter}}_{N_p-1|t-1})$, we have $\mathbf{P}^{N_p}_t(\mathrm{SOC}_{f,N_p|t})$:

$$p_{N_p+1|t,\mathrm{SOC}} \sim \mathcal{N}(\overline{p}_{t+N_p+1,\mathrm{SOC}},\ \sigma_1^2), \tag{6.16a}$$

$$\mathrm{SOC}_{f,N_p|t} \sim \mathcal{N}(\overline{\mathrm{SOC}}_{t+N_p},\ \sigma_2^2), \tag{6.16b}$$

where $\overline{\text{SOC}}$ and $\overline{p}_{\text{SOC}}$ are ground truth/nominal SOC and its associated co-state trajectories, and are obtained from offline simulations (*training data set*). How to obtain $\overline{\text{SOC}}$ and $\overline{p}_{\text{SOC}}$ will be detailed in Chapter 7.

Remark 8. Note that in (6.7c) the cost-to-go is approximated by the inner product of all states and their associated co-states. However, currently, the cost-to-go of the state N_p-step head is approximated by considering only the SOC and its associated co-state. The other states and co-states are ignored based on Assumption 6-$\mathcal{A}$. At time t, a multi-step look-ahead [92] through N_p-step MPC is used to improve the estimation of the cost-to-go $\hat{V}_{t+1}$ at $t+1$ starting from an initial estimation of $\hat{V}^0_{N_p}$ in the form of DMIS. Future work could consider adding a proper terminal cost (e.g., a Lyapunov function characterizing the requirement of allowable distance bands.)

$$\delta^{\text{TD}}_{N_p|t,\text{SOC}} = \underbrace{p^{\text{maxiter}}_{N_p|t-1,\text{SOC}}}_{@\text{ time } t-1} - \underbrace{\frac{\partial H_{N_p|t}(x_{N_p|t}, u_{N_p|t}, \hat{p}_{N_p+1|t,\text{SOC}})}{\partial \text{SOC}_{N_p|t}}}_{\text{ground truth correction}} \tag{6.16c}$$

$$p^0_{N_p|t,\text{SOC}} = p^{\text{maxiter}}_{N_p|t-1,\text{SOC}} + \gamma \delta^{\text{TD}}_{N_p|t,\text{SOC}}, \quad p^0_{\star,N_p|t} = 0 \tag{6.16d}$$

$$\mathcal{P}^0_{N_p|t} = \mathcal{P}^0_{N_p|t}(p^0_{N_p|t}, \mathcal{X}^0_{N_p|t}, \mathcal{U}^0_{N_p|t}) \tag{6.16e}$$

$$\mathcal{P}^{i+1}_{N_p|t} = \Pi T_\mu(\mathcal{P}^i_{N_p|t}, \text{SOC}_{f,N_p|t}), \quad i = 0, \cdots, \text{maxiter} \tag{6.16f}$$

$$u_t := u^*_{0|t} = \arg\min_{u \in \mathcal{U}} l(x_t, u) + \rho \cdot (p^{\text{maxiter}}_{1|t})^T f(x_t, u) \tag{6.16g}$$

$$T^{\mu_{\text{MPC}}} = \{u_0, \cdots, u_t, \cdots, u_{N_f-1}\}. \tag{6.16h}$$

In (6.16), $p_\star$ denotes the co-state associated with states other than SOC. As discussed in previous sections, the co-state corresponding to states other than SOC, the lack of terminal cost and constraints indicate a zero terminal co-state. The DMIS iteration $\mathcal{P}^{i+1}_{N_p|t} = \Pi \circ T_\mu(\mathcal{P}^i_{N_p|t})$ in (6.16f) is defined as:

$$p^{i+1}_{k|t} = \frac{\partial H_{k|t}}{\partial x_{k|t}}|_{x^{*,i}_{k|t}, (u^{*,i}_{k|t})^i, p^i_{k+1|t}} = \rho^k \frac{\partial l(\cdot)}{\partial x}|_{x^{*,i}_{k|t}, u^{*,i}_{k|t}} + \Big(\frac{\partial f(\cdot)}{\partial x}|_{x^{*,i}_{k|t}, u^{*,i}_{k|t}}\Big)^T p^i_{k+1|t}. \tag{6.17a}$$

$$p^{i+1}_{N_p|t,\text{SOC}} = p^i_{N_p|t,\text{SOC}} - [\frac{\partial c(\cdot)}{\partial p_{N_p|t,\text{SOC}}}|_{p^i_{N_p}}]^{-1} c(\text{SOC}_{f,N_p|t}, \cdot)^i$$

$$(u^*_{k|t})^{i+1} = \arg\min_{u \in \mathcal{U}} H(x^{i+1}_{k|t}, u, p^{i+1}_{k+1|t}) = \arg\min_{u \in \mathcal{U}} l(x^i_{k|t}, u) + (p^i_{k+1|t})^T f(x^i_{k|t}, u), \tag{6.17b}$$

which is functionally equivalent to approximate-PI. In (6.17a), $c(\text{SOC}_{f,N_p|t}, \cdot)$ denotes the equality constraint at the end of the prediction horizon $\text{SOC}_{N_p|t} = \text{SOC}_{f,N_p|t}$. (6.16a) and (6.16b) are to characterize the target value (similar to the target network in NN-based RL), where $p_{N_p+1|t,\text{SOC}}$ is used in the *external correction* step (6.16c), and $\text{SOC}_{N_p+1|t}$ is used in the *internal correction* step (6.17). Note that the inverse of the sensitivity of the terminal equality constraint $c(\cdot)$ to the terminal co-state p_{N_p}, $[\frac{\partial c(\cdot)}{\partial p_{N_p|t,\text{SOC}}}|_{p^i_{N_p}}]^{-1}$, is used in (6.17a) for the purpose of notation. Approximations can be made in the actual implementation to avoid unstable shooting iterations induced by large terminal co-state update size. As observed in simulations, in the absence of the nominal co-state trajectory $\bar{p}$, the initial guess terminal co-state $p^0_{N_p|t,\text{SOC}}$ will only be in a purely feedback form of the nominal SOC as in (6.17a). It then decreases to the P-ECMS presented in Chapter 3, where large oscillations will be induced in the resulting co-state trajectory. The external and internal corrections are therefore balancing the bias and variance in approximating the terminal cost.

On the one hand, it is observed that the state and co-state at each time instant roughly obey Gaussian distribution centered around the nominal trajectory, as illustrated in Figs. 6.1 and 6.2. As a result, in the online implementation, it is desirable to sample the ”ground-truth” state and co-state values from the assumed distribution rather than using the nominal trajectories directly due to the uncertainty of each individual trip. On the other hand, the importance of the noisy sample $\bar{p}_{\text{SOC}}$ is observed in simulations; although it is possible to warm-start the terminal co-state $p^0_{N_p|t}$ using the value $\bar{p}_{t+N_p}$ from the nominal co-state trajectory $\bar{p}$, due to the trip uncertainty, relying solely on $\bar{p}$ would induce a systematic bias to the terminal SOC at the end of the trip [33].

The complete algorithm is shown as Algorithm 3.

Algorithm 3 A unified DMIS-based framework for PHEV PT or combined VD + PT control

Inputs: prediction horizon N_p, nominal SOC $\overline{\text{SOC}}$ and its corresponding co-state trajectories $\bar{p}_{\text{SOC}}$, initial condition x_0, terminal SOC target SOC_f.

For *ps-opt* and *sequential-opt*: $x = (\text{SOC}\ \tilde{e}), u = (E\ ne\ te)$,

For *co-opt*: $x = (\text{SOC}, \tilde{e}\ s\ v), u = (E\ ne\ te\ r)$.

Warm-start the state and control sequence, $\mathcal{X}^0_{N_p|0}$ and $\mathcal{U}^0_{N_p|0}$ with either an pre-solved offline solution on a similar trip or obtained by propagating the prediction model forward-in-time with a base control sequence (zero control input).

```
1:  for $t = 1$ to $N_f$ do
2:      Predict lead vehicle's trajectories
            $\mathcal{V}^l_{N_p|t} := v^l_t, v^l_{1|t}, \cdots, v^l_{N_p|t}, \;\; \mathcal{S}^l_{N_p|t} := s^l_t, s^l_{1|t}, \cdots, s^l_{N_p|t}.$
3:      if sequential-opt or co-opt then
4:          Formulate the position constraints using $\mathcal{S}^l_{N_p|t}$ according to (6.21).
5:      if sequential-opt then
6:          Solve a $N_p$-step version of (6.9) with $\mathcal{S}^l_{N_p|t}$, get the acceleration and velocity sequence
            $\mathcal{V}^*_{N_p|t} := v^*_{1|t}, \cdots, v^*_{N_p|t}, \quad \mathcal{A}^*_{N_p|t} := a^*_{0|t}, \cdots, a^*_{N_p-1|t}.$
7:      for $i = 1$ to maxiter do
8:          if sequential-opt/ps-opt then
9:              $V^*_{N_p|t}/\mathcal{V}^l_{N_p|t}$ used as the predicted ego vehicle's trajectory in the power-split optimization.
        Policy Evaluation with $\text{SOC}_{f,N_p|t}$:
10:         for $k = 1 : N_p$ do
11:             $p^i_{k|t} = \frac{\partial H^i_{k|t}}{\partial x_{k|t}}$ as in (6.17a).
        Policy Improvement:
12:         for $k = 1 : N_p$ do
13:             $u^i_{k|t} = \arg\min\{H^i_{k|t} + \eta(u - u^{i-1}_{k+1|t-1})^2\}$ as in (6.17b). $\eta(\cdot)$ is a stabilizing term and is detailed in [85].
14:     Obtain $\mathcal{X}^{*,\text{maxiter}}_{N_p|t}, \mathcal{U}^{*,\text{maxiter}}_{N_p|t}$ and $\mathcal{P}^{*,\text{maxiter}}_{N_p|t}$.
15:     if has VD control then
16:         if sequential-opt then
17:             $a^*_{0|t}$ filtered by CBF (design is omitted but follow the procedure in Section 6.3.6) $\Longrightarrow \tilde{a}^*_{0|t}$.
18:         if co-opt then
19:             $r^*_{0|t}$ filtered by CBF (6.36)$\Longrightarrow \tilde{r}^*_{0|t}$.
20:     Execute $u^*_{0|t}$ (with $\tilde{a}^*_{0|t}$ in sequential-opt), simulate system forward.
        TD-error correction
21:     Noisy sample from $\bar{p}$ (6.16a), calculate the TD-error with (6.16c), and correct the initial guess of $p^0_{N_p|t+1,\text{SOC}}$ by (6.16d).
22:     Noisy sample from S$\bar{\text{O}}$C (6.16b), used in the policy evaluation (step 11).
23:     Shift the state and control sequence
            $\mathcal{X}^0_{N_p|t} = (x^{*,\text{maxiter}}_{1|t-1}, \cdots, x^{*,\text{maxiter}}_{N_p|t-1}, x^{*,\text{maxiter}}_{N_p|t-1})$
            $\mathcal{U}^0_{N_p|t} = (u^{*,\text{maxiter}}_{1|t-1}, \cdots, u^{*,\text{maxiter}}_{N_p-1|t-1}, u^{*,\text{maxiter}}_{N_p-1|t-1}).$
```

6.3.6 A Control Barrier Function (CBF) Approach to Enforce Safety

In the preliminary work on the online implementable *co-opt* controller [33], the constraint violation quantity is augmented to the stage cost (6.8f) with the exterior penalty method. However, it is observed that constraint satisfaction is closely related to the number of single shooting iterations. A small number of single shooting iterations are preferred to reduce computation burden, which yet could lead to constraint violation. Besides, the prediction of the lead vehicle's trajectory is used to formulate the position constraint in both the *co-opt* and *sequential-opt* problems. Due to the prediction inaccuracy, even though the underlying QP (6.9) with a prediction horizon N_p in *sequential-opt* can be solved rather reliably and fast enough to convergence, position constraint violation could still happen due to prediction inaccuracy.

Among the state and mixed state and control constraints considered, the position constraints in the vehicle-following control are time-varying and depend on the lead vehicle's position. The time-varying position constraints make the design of a safety-certified vehicle-following controller challenging. The work in [95] proposed finding an invariant tube out of time-varying constraints. A scalable maximal control invariant set was introduced to adjust the solution obtained from the EMPC to guarantee constraint satisfaction in the minimum and maximum distance gaps to the target lead vehicle despite imperfect prediction and insufficient single shooting iterations. However, it is not computationally efficient to carry a control-invariant set (even a polytope) online and perform a projection each time instant. Moreover, the acceleration (5.15b) in the reformulated vehicle-following dynamics is a function of the state and control input. The acceleration constraints are not included in calculating the maximal control invariant set because of the mixed state and control constraint nature. It is desirable to have a systematic way to deal with state constraints without significantly additional computational efforts. To this end, in this section a control barrier function [96] design is presented to enforce time-varying position constraints and allowable acceleration limits.

6.3.6.1 Barrier Functions for Discrete-Time Systems

The control barrier function (CBF) has proven its usefulness in safe-critical systems, especially automotive applications [96]. It provides inequality constraints in the control input that, when satisfied, imply forward invariance of the set [96]. This section follows the discrete-time version presented in [97].

I consider a *Safety Set* $\mathcal{C}$ and establish its forward invariance. The safety set and its boundary are assumed by characterized by a continuous function h as

$$\mathcal{C} :=\{x \in \mathcal{D} | h(x) \geq 0\} \tag{6.18}$$

$$\partial\mathcal{C} :=\{x \in \mathcal{D} | h(x) = 0\}. \tag{6.19}$$

Define the safety set, $\mathcal{C}$ as

$$\mathcal{C} = \{x \in \mathcal{D} | B(x) \geq 0\}, \tag{6.20}$$

where $B : \mathcal{D} \mapsto \mathbb{R}$ is called the discrete-time Exponential Control Barrier Function.

Definition 6.3.1. [97] A map $B : \mathcal{D} \mapsto \mathbb{R}$ is a discrete-time Exponential Control Barrier Function if

1. $B_0 \geq 0$, and
2. there exists a control input $u_k \in \mathbb{R}^m$ such that $\Delta B_k + \gamma B_k \geq 0, \forall k \in \mathbb{Z}^+, 0 < \gamma \leq 1$.

The set $\mathcal{C}$ is invariant along the trajectories of the discrete-time system if there exits a map $B : \mathcal{S} \mapsto \mathbb{R}$ such that:

1. $B_0 \geq 0$, and
2. $B_{k+1} - B_k + \gamma B_k \geq 0, \forall k \in \mathbb{Z}^+, 0 < \gamma \leq 1$

Proof. $B_t \geq (1-\gamma)^t B_0 \geq 0, \forall t \in \mathbb{Z}^+, 0 < \gamma \leq 1$, since $B_t \geq (1-\gamma)B_{t-1} \geq (1-\gamma)^2 B_{t-2} \geq ... \geq (1-\gamma)^t B_0$. □

6.3.6.2 Time-headway Constraints as Hard Constraints

For the vehicle following subsystem, the constraints are defined as the

$$s^l_k - s_k - d_0^{\min} - \tau_{\min} v_k \geq 0 \tag{6.21a}$$

$$-s^l_k + s_k + d_0^{\max} + \tau_{\max} v_k \geq 0, \tag{6.21b}$$

where $\tau_{\min}, \tau_{\max}$ are the minimum and maximum time-headway, respectively. $d_0^{\min} \leq d_0^{\max}$ to guarantee a small margin during standstill. The time-headway constraints (6.21) can be written compactly as

$$h(x_k, k) := Hx_k + F_k \geq 0, \tag{6.22}$$

where $H = \begin{bmatrix} -1 & -\tau_{\min} \\ 1 & \tau_{\max} \end{bmatrix}$, and $F_k = \begin{bmatrix} s^l_k - d^{\min}_0 \\ -s^l_k + d^{\max}_0 \end{bmatrix}$.

The dynamics of the vehicle-following subsystem (6.24) and (6.23)

$$\begin{aligned} s_{k+1} &= s_k + \frac{\Delta t}{2}(v_k + v_{k+1}) && (6.23) \\ v_{k+1} &= K(s^l_k - s_k - d_0 + R\, r_k), && (6.24) \end{aligned}$$

written compactly are

$$x_{k+1} = Ax_k + B_r r_k + B_s s^l_k + c_0, \tag{6.25}$$

where $A = \begin{bmatrix} 1 - \frac{\Delta t K}{2} & \frac{\Delta t}{2} \\ -K & 0 \end{bmatrix}$, $B_r = \begin{bmatrix} \frac{\Delta t K R}{2} \\ KR \end{bmatrix}$, $B_s = \begin{bmatrix} \frac{\Delta t K}{2} \\ K \end{bmatrix}$, and $c_0 = \begin{bmatrix} -\frac{\Delta t K}{2} \\ -K \end{bmatrix} d_0$.

Define $B_k = h(x_k, k)$, the CBF condition now becomes:

$$\begin{aligned} B_{k+1} + (\gamma - 1)B_k =& H(Ax_k + B_r r_k + B_s s^l_k + c_0) + F_{k+1} + (\gamma - 1)(Hx_k + F_k) && (6.26) \\ =& H[A + (\gamma - 1)I]x_k + HB_s s^l_k + Hc_0 + F_{k+1} + (\gamma - 1)F_k + HB_r r_k \geq 0 \end{aligned}$$

Note that one issue arises that in (6.26), F_{k+1} depends on the lead vehicle's position at $t = k+1$, s^l_{k+1}, which is unknown at $t = k$. To resolve this issue, we consider the following tightened constraints

$$\begin{aligned} s^l_{k+1} =& s^l_k + v^l_k \Delta t + \frac{\Delta t^2}{2} a^l_k \leq \overline{s}^l_{k+1} \text{ where } \overline{s}^l_{k+1} = s^l_k + v^l_k \Delta t + \frac{\Delta t^2}{2} a^l_{\max} && (6.27) \\ s^l_{k+1} =& s^l_k + v^l_k \Delta t + \frac{\Delta t^2}{2} a^l_k \geq \underline{s}^l_{k+1} \text{ where } \underline{s}^l_{k+1} = s^l_k + v^l_k \Delta t + \frac{\Delta t^2}{2} a^l_{\min}, && (6.28) \end{aligned}$$

where $a^l_{\max}$ and $a^l_{\min}$ are the lead vehicle's maximum and minimum allowable acceleration and can be estimated from data or a conservative estimation. As a result, the following constraints are to be satisfied

$$\begin{aligned} B_{k+1} + (\gamma - 1)B_k =& H[A + (\gamma - 1)I]x_k + HB_s s^l_k + Hc_0 && (6.29) \\ & + (\gamma - 1)F_k + HB_r r_k + F^{\min}_{k+1} \geq 0, \end{aligned}$$

where $F^{\min}_{k+1} = \begin{bmatrix} \underline{s}^l_{k+1} - d^{\min}_0 \\ -\overline{s}^l_{k+1} + d^{\max}_0 \end{bmatrix}$.

In addition, the acceleration constraints

$$a^{\min} \le a_k = \frac{\frac{1}{\tau}(s_k^l - s_k - d_0 + Rr_k) - v_k}{\Delta t} \le a^{\max}, \tag{6.30}$$

where we consider $|a^{\min}| = |a^{\max}|$. Since it may be the case that the time-headway constraints (6.21) will conflict with the acceleration constraints (6.30). We could either require an acceleration-based barrier function as in [96] allowing the time-headway constraints and acceleration constraints to be simultaneously satisfied, or we need to show that (6.26) and (6.30) can always be satisfied simultaneously.

Assume $x_k \in h(x_k, k)$, then the time-headway constraints (6.21) are satisfied all the time.

$$\begin{aligned}
\text{(6.21a)} \implies a_k\Delta t &= \frac{1}{\tau}(s_k^l - s_k - d_0 + Rr_k) - v_k \qquad &\text{(6.31a)}\\
&\ge \frac{\tau_{\min} - \tau}{\tau}v_k + \frac{1}{\tau}(d_0^{\min} - d_0) + \frac{R}{\tau}r_k\\
\text{(6.21b)} \implies a_k\Delta t &= \frac{1}{\tau}(s_k^l - s_k - d_0 + Rr_k) - v_k \qquad &\text{(6.31b)}\\
&\le \frac{\tau_{\max} - \tau}{\tau}v_k + \frac{1}{\tau}(d_0^{\max} - d_0) + \frac{R}{\tau}r_k.
\end{aligned}$$

To have $x_k \in h(x_k, k)$ also satisfying the acceleration constraints (6.30), we want

$$\begin{aligned}
\text{RHS of (6.31a)}: \ &\frac{\tau_{\min} - \tau}{\tau}v_k + \frac{1}{\tau}(d_0^{\min} - d_0) + \frac{R}{\tau}r_k \ge a^{\min}\Delta t \qquad &\text{(6.32a)}\\
\implies r_k \ge &\frac{1}{R}[\tau a^{\min}\Delta t + (\tau - \tau_{\min})v_k + (d_0 - d_0^{\min})]\\
\text{RHS of (6.31b)}: \ &\frac{\tau_{\max} - \tau}{\tau}v_k + \frac{1}{\tau}(d_0^{\max} - d_0) + \frac{R}{\tau}r_k \le a^{\max}\Delta t \qquad &\text{(6.32b)}\\
r_t \le &\frac{1}{R}[\tau a^{\max}\Delta t + (\tau - \tau_{\max})v_k + (d_0 - d_0^{\max})]
\end{aligned}$$

To guarantee the existence of a non-empty set (6.32),

$$\begin{aligned}
&\tau a^{\min}\Delta t + (\tau - \tau_{\min})v_k + (d_0 - d_0^{\min}) \le \tau a^{\max}\Delta t + (\tau - \tau_{\max})v_k + (d_0 - d_0^{\max}) \implies\\
&(\tau_{\max} - \tau_{\min})v_k + (d_0^{\max} - d_0^{\min}) \le \tau(a^{\max} - a^{\min})\Delta t. \qquad \text{(6.33)}
\end{aligned}$$

(6.33) can only be guaranteed when v_k satisfies $v_k \le v^a := \frac{\tau(a^{\max} - a^{\min})\Delta t - (d_0^{\max} - d_0^{\min})}{(\tau_{\max} - \tau_{\min})}$. When v_k is large, the set needs to be modified. When v_k increases, (6.33) can still be guaranteed by increasing $\tau_{\min}$ (thus reducing $(\tau_{\max} - \tau_{\min})$). Increasing $\tau_{\min}$ is also physically intuitive since when the speed v_k is high, it is reasonable to increase the time-headway between the ego and its lead vehicle to maintain safety.

Based on the above discussions, the modified time-headway constraints $h^a(x_k, k)$ that is acceleration admissible is

$$h^a(x_k, k) := H_k x_k + F_k \geq 0, \tag{6.34}$$

where $H_k = \begin{bmatrix} -1 & -\tau^a_{\min}(v_k) \\ 1 & \tau_{\max} \end{bmatrix}$, and $F_k = \begin{bmatrix} s^l_k - d^{\min}_0 \\ -s^l_k + d^{\max}_0 \end{bmatrix}$. In (6.34),

$$\tau^a_{\min}(v_k) = \begin{cases} \tau_{\min} & v_k \leq v^a \\ \tau_{\max} - \frac{\tau(a^{\max} - a^{\min})\Delta t - (d^{\max}_0 - d^{\min}_0)}{(v_k + a^{\max}\Delta t)} & v_k > v^a \end{cases}.$$

The speed-dependent minimum time-headway $\tau^a_{\min}(v_k)$ satisfies $\tau^a_{\min}(v_k) \in [\tau_{\min}, \tau_{\max}), \forall v_k$. To see this, first notice that when $v_k \leq v^a$, $\tau^a_{\min}(v_k) = \tau_{\min}$. When $v_k > v^a$,

$$\begin{aligned} v_k > \frac{\tau(a^{\max} - a^{\min})\Delta t - (d^{\max}_0 - d^{\min}_0)}{(\tau_{\max} - \tau_{\min})} \implies \\ (\tau_{\max} - \tau_{\min})v_k - \tau(a^{\max} - a^{\min})\Delta t + (d^{\max}_0 - d^{\min}_0) \geq 0 \\ (\tau_{\max} - \tau_{\min})(v_k + a^{\max}\Delta t) - \tau(a^{\max} - a^{\min})\Delta t + (d^{\max}_0 - d^{\min}_0) > (\tau_{\max} - \tau_{\min})v_k \\ - \tau(a^{\max} - a^{\min})\Delta t + (d^{\max}_0 - d^{\min}_0) \geq 0 \implies \tau^a_{\min}(v_k) - \tau_{\min} > 0, \quad \forall v_k > v^a. \end{aligned} \tag{6.35a}$$

Meanwhile, $\tau, a^{\max}, a^{\min}, d^{\max}_0, d^{\min}_0$ are chosen such that $\tau(a^{\max} - a^{\min})\Delta t - (d^{\max}_0 - d^{\min}_0) > 0$ to guarantee $\tau^a_{\min}(v_k) < \tau_{\max}$.

The constraints (6.29) are modified as

$$\begin{aligned} B_{k+1} + (\gamma - 1)B_k =& H_k[A + (\gamma - 1)I]x_k + H_k B_s s^l_k \\ &+ H_k c_0 + (\gamma - 1)F_k + H_k B_r r_k + F^{\min}_{k+1} \geq 0. \end{aligned} \tag{6.36}$$

Suppose (6.29) is recursive feasible, then the recursive feasibility of (6.36) can be guaranteed because (6.26) and (6.36) can be rearranged into

$$H(B_r r_k + h_k) + (\gamma - 1)F_k + F^{\min}_{k+1} \geq 0 \tag{6.37}$$

$$H_k(B_r r_k + h_k) + (\gamma - 1)F_k + F^{\min}_{k+1} \geq 0, \tag{6.38}$$

respectively. The first term in (6.37) and (6.38) are linear combinations of columns of H and H_k, respectively. Since the columns of H_k remains linearly independent and the relative position is preserved as illustrated in Fig. 6.3.

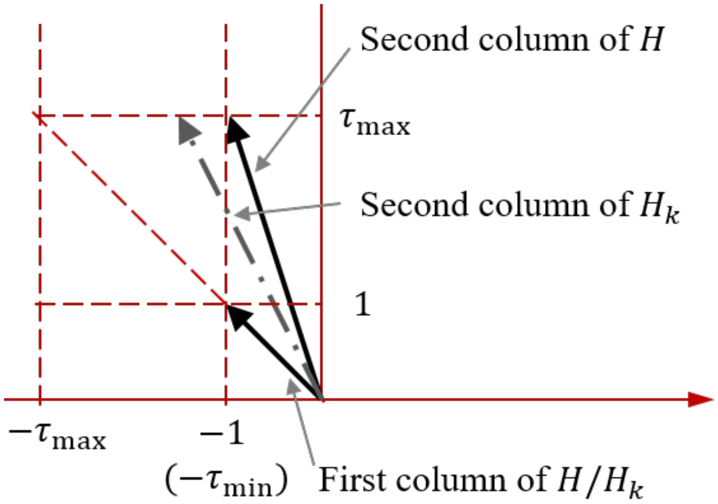


Figure 6.3: Visualization of the column vectors of H/H_k.

6.4 Summary

This chapter expands the discussions on the connection between the DMIS and approximate policy iteration (approximate-PI) in the reinforcement learning (RL) application, integrating the numerical optimization strategy with the online implementable framework. Henceforward, the PT-only control of a human-driven to the combined VD and PT control of an automated PHEV are unified under the same DMIS-based framework. In the unified framework, the cost-to-go (the fuel consumption as the economic cost) is represented by the co-state associated with the SOC dynamics. This co-state is corrected both internally using a single shooting method and externally based on the co-state TD-error. In its application to automated PHEVs, a CBF is augmented as an add-on block to modify the vehicle level control input to be executed whenever necessary. The unified control framework guarantees safe vehicle-following behavior when including longitudinal vehicle dynamics control. The unified framework allows for systematically evaluating the fuel economy and drivability performance of different levels and structures of optimization strategies, detailed in Chapter 7.

CHAPTER 7

Performance Analysis of Different Optimization Schemes in Fuel Economy and Driving Statistics

7.1 Introduction

So far, different optimization schemes have been presented, starting from the power-split optimization (PT level optimization), then shifting toward combined VD and PT optimizations. To achieve a combined VD and PT optimization, first the sequential optimization strategies is investigated, where the VD level optimization is performed first through acceleration minimization then followed by the PT optimization. The remaining efforts are devoted to the centralized VD and PT co-optimization. It is critical to understand the fuel economy potential and the drivability with each optimization scheme with all the different optimization schemes. This chapter seeks to systematically evaluate the fuel economy and drivability performance of different levels and structures of optimization strategies.

In terms of the levels of optimization strategies, both the PT level control for a human driver and the combined VD and PT level control for an autonomous driver are evaluated. In terms of the optimization structures for combined VD and PT level controls, both the centralized optimization structure, where the VD and PT are optimized together under one *single* objective function, and a decentralized (sequential) optimization structure, where the VD and PT are optimized separately with two different objective functions are of interest.

Specifically, this chapter will focus on

1. **Offline Illustrative Comparison (Section. 7.2):** One illustrative cycle comparison of offline optimization results with different optimization schemes in time domain.

2. **Offline Statistical Comparison (Section. 7.3):** Statistical comparison of offline optimization results with different optimization schemes on many similar Ann Arbor trips.

3. **Online Statistical Comparison (Section. 7.4):** Online Statistical comparison of online MPC results with different optimization schemes on many similar Ann Arbor trips.

7.2 Analysis of Offline Optimization Results in Time-Domain

This section compares the performance of different optimization strategies in an offline manner. The following optimization schemes are considered

1. **opt-cdcs** : The ego vehicle's velocity profile is identical to the lead vehicle's trajectory. The vehicle operates in an EV mode (charge depletion) first, followed by an optimal charge-sustaining mode, i.e., CDCS.

2. **ps-opt**: The ego vehicle's velocity profile is identical to the lead vehicle's trajectory. The vehicle operates in an optimal power-split mode. This strategy represents the optimal energy-management of a human-driven PHEV.

3. **seq-opt**: The ego vehicle's velocity profile is optimized by solving an acceleration minimization problem [32] that provides a smooth velocity. Then, power-split optimization is performed with the resulting velocity profile. This strategy represents the sequential optimization approach to eco-driving in the existing literature.

Moreover, this chapter discusses the tradeoff between passenger comfort and fuel consumption benefit. Specifically, the original co-optimization problem ($\mathbf{P}_1$ as in (5.17)) and its two additional variants are considered:

1. **co-opt**: The ego vehicle's velocity trajectory and power-split are optimized by solving the original co-optimization problem ($\mathbf{P}_1$) with acceleration limits ($a^{\max}$ and $a^{\min}$) in (5.17b).

2. **co-opt**$_{\Delta a}$: A variant of the co-optimization with jerk limits ($\Delta a^{\max}$ and $\Delta a^{\min}$) is additionally included. The augmented stage cost $\tilde{l}_k$ replaces l_k in (5.17b)

$$\tilde{l}_k = l_k + \max(\Delta a_k - \Delta a^{\max}, 0)^2 + \max(\Delta a^{\min} - \Delta a_k, 0)^2, \tag{7.1}$$

   where $\Delta a_k = a_k - a_{k-1}$, with $a_{-1} = 0$.

3. **co-opt**$_Q$: A variant of the co-optimization where vehicle acceleration is penalized to induce smoothness. The augmented stage cost $\bar{l}_k$ replaces l_k in (5.17b)

$$\bar{l}_k = l_k + Q \cdot a_k^2. \tag{7.2}$$

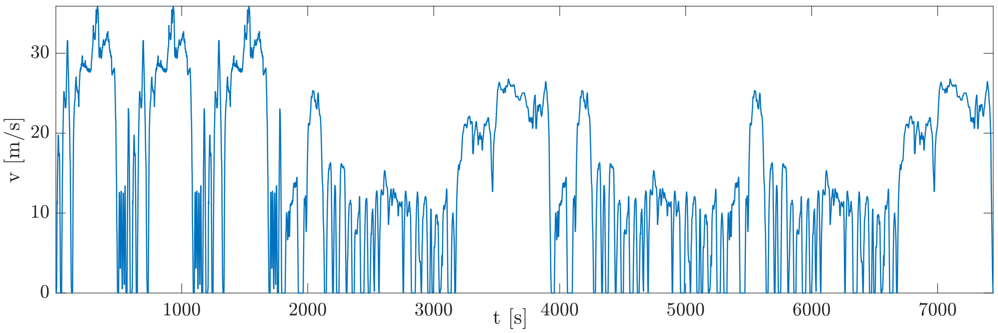


Figure 7.1: The lead/ego vehicle's trajectory in the considered CDCS-worst scenario.

For PHEVs, the trip considered in this thesis is usually defined to be between recharge opportunities, where the large battery is depleted from fully charged to a low level[1]. As discussed in previous sections, a 2-hr trip is considered consisting of stitched standard driving cycles (3UDDS + 3US06 + 3HWFET). The standard driving cycles are combined in a particular order to generate the most fuel consumption benefit with power-split optimization compared to CDCS as observed from offline simulation results. For the considered scenario, the trip is long enough to exceed the vehicle's pure EV range. The time-domain response of the considered scenario is shown in Fig. 7.2. The initial battery SOC is SOC$_0 = 0.85$, and the desired terminal SOC is SOC$_f = 0.15$. The first subplot presents the battery SOC trajectories. The second subplot presents the cumulative fuel consumption trajectories (including cranking fuel). The third subplot presents cumulative cranking fuel consumption trajectories. The fourth subplot presents the resulting velocity trajectories. The exact values are summarized in Table. 7.1. Note that in all the simulations, the actual terminal SOC is adequately close to the desired SOC terminal SOC SOC$_f$. To make Fig. 7.2 readable, only selective simulations are presented. The full evaluation of percentage fuel consumption reduction with different optimization schemes is presented in Fig. 7.3.

7.2.1 Trade-Off between Fuel Economy and Passenger Comfort

A thorough evaluation of fuel consumption reduction from the baseline CDCS strategy on the human driving with different optimization schemes is presented in Fig. 7.3. As clearly observed in Fig. 7.3, a nearly 30% reduction in total fuel consumption can be achieved with the co-optimization in the absence of additional passenger comfort constraints. By comparison, sequential optimization can only reach half of the fuel consumption reduction potential. All the remaining fuel consumption results with the co-optimization with passenger comfort constraints

[1] The Ann Arbor trips to be studied are relatively short and can be driven in a pure EV mode if starting with a fully-charged battery. Therefore, to make optimizations meaningful, a small SOC span is considered to force the engine to be used.

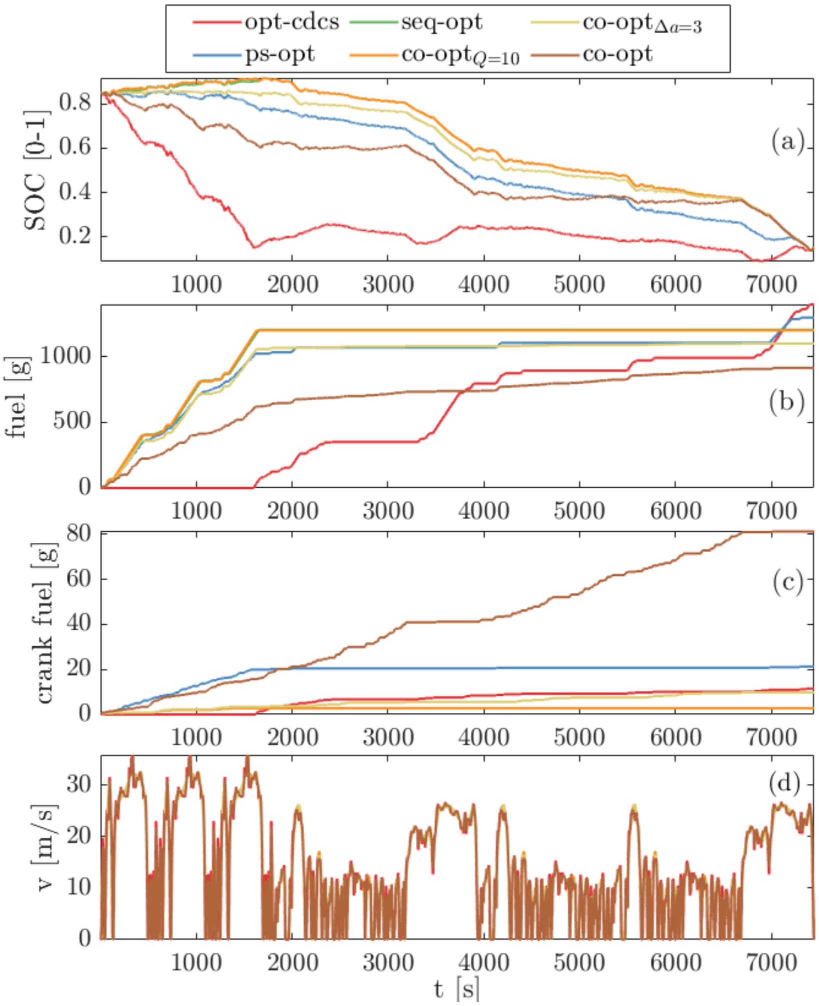


Figure 7.2: Time-domain response comparisons for selective simulation results of the considered CDCS-worst cycle.

fall between the sequential optimization and the full co-optimization. The trade-off between the fuel economy and passenger comfort is evident by the transition from sequential optimization to full co-optimization with neither acceleration penalty nor jerk limits. At the same time, it can be seen that the penalty weights on the acceleration affect the fuel economy dramatically. By comparison, the effects of different jerk limits on the fuel economy are more incremental. This observation is instructive to choose a cost function as the jerk level better characterizes the trade-off between the passenger comfort and the fuel economy. On the other hand, the quadratic acceleration penalty significantly limits the margin for acceleration variations and might not be the best indication of passenger comfort levels.

The time-domain responses of some selective control strategies indicating different levels of passenger comfort requirements are given in Fig. 7.2: it is be found that the terminal SOC constraint is well satisfied in all simulations as seen from Fig. 7.2(a). Figures 7.2(b) and

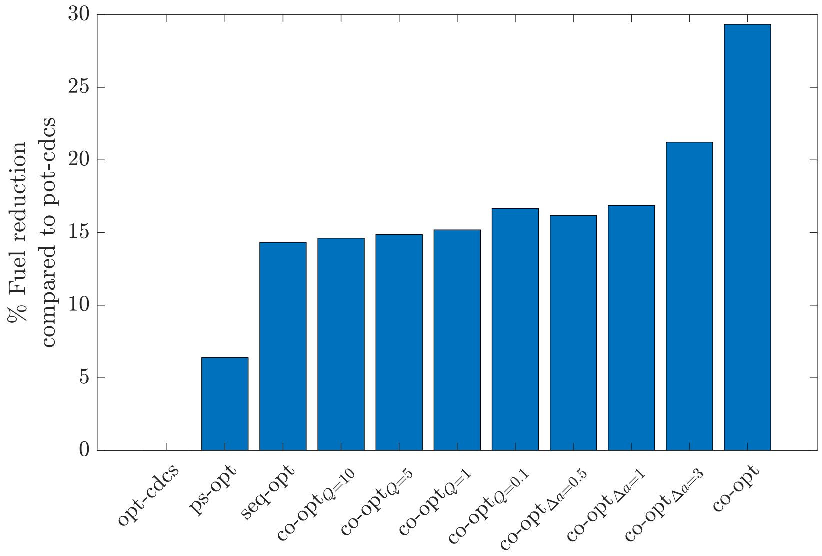


Figure 7.3: The full comparison of Fuel economy benefits.

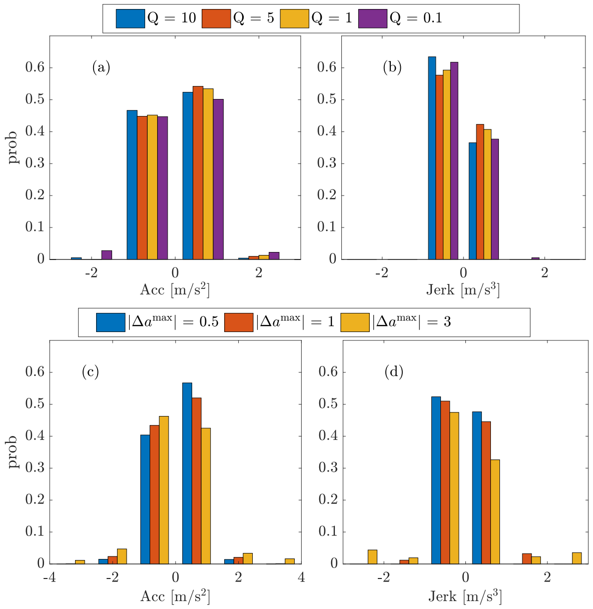
(c) present the cumulative fuel consumption trajectories and the cumulative cranking fuel consumption trajectories, respectively. Figure 7.2(d) shows the resulting velocity trajectories. The fuel consumption values are summarized in the second and third columns in Table. 7.1.

The first observation is the difference in the charge-blending properties indicated by SOC trajectories (see Fig. 7.2(a)). As the passenger comfort requirement tightened, the battery tends to be more heavily charged at the beginning of the trip to grant the vehicle an opportunity to run nearly exclusively in EV mode in the latter part of the trip. The reason is related to a smoothed power demand needed by the passenger comfort and the high-efficiency, high power engine operation points required by the fuel minimization objective. Another observation is that the considerable potential in fuel economy achieved by co-optimization (neither acceleration penalty nor jerk limit) is realized through frequent switching between the EV and HV modes. In particular, the co-optimization without additional passenger comfort constraints consumes significantly more cranking fuel than the other methods. As the passenger comfort requirement becomes more stringent in the co-optimization, the resulting velocity profiles become smoother, which leads to a reduction in the total cranking fuel. Lastly, as observed in Fig. 7.3, the fuel economy with co-optimization with an additional acceleration penalty where $Q = 10$ degrades to a level similar to the sequential optimization (acceleration minimization followed by power-split optimization). This observation is also evident by the similarity of the time-domain responses between the two simulations, as shown in Fig. 7.2.

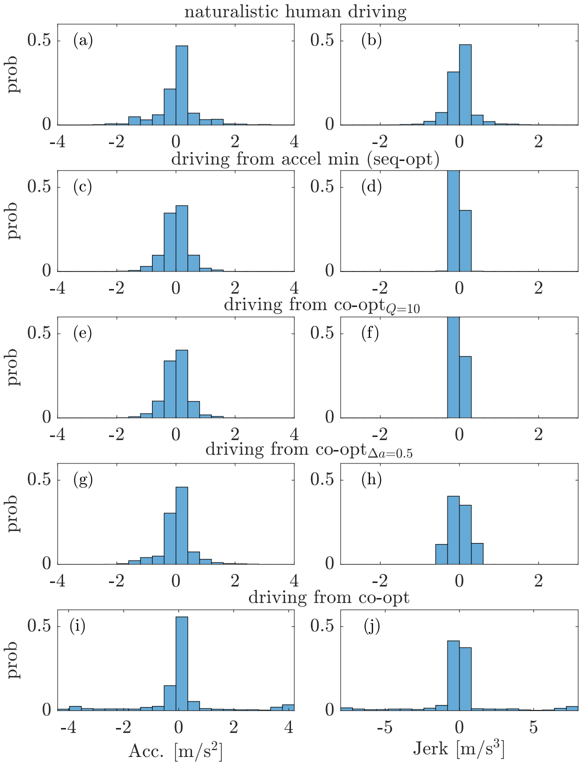
The acceleration and jerk statistics are compared across different optimization schemes in Fig. 7.4b (the distribution of acceleration in the first column and the distribution of jerk in the second column): the naturalistic human driving, that is, the original driving cycle in

Table 7.1: Fuel consumption and passenger comfort comparisons among selective simulations. opt-cdcs: the original human driving in Fig. 7.1, seq-opt: the velocity trace obtained by the acceleration minimization over the whole trip, co-opt: the velocity trace resulting from the co-optimization, with different passenger comfort considerations.

Test name	total fuel w/ crank fuel [g] (% reduction)	crank fuel [g]	max abs. accel [m/s^2]	ave abs. accel [m/s^2]	max abs. jerk [m/s^3]	ave abs. jerk [m/s^3]
opt-cdcs	1411 (0)	12	3.8	0.4	3.4	0.19
ps-opt	1321 (6)	21	-	-	-	-
seq-opt	1209 (14)	3	2.0	0.3	0.4	0.04
co-opt$_{Q=10}$	1205 (15)	3	1.7	0.3	0.4	0.04
co-opt$_{Q=0.1}$	1176 (17)	9	2.9	0.3	1.4	0.08
co-opt$_{\Delta a=1}$	1173 (17)	3	3.3	0.3	1.1	0.22
co-opt$_{\Delta a=3}$	1112 (21)	10	4.7	0.4	3.9	0.56
co-opt	997 (29)	82	4.0	0.7	8.0	0.95



(a) Passenger comfort analysis for different Q values penalizing squared acceleration term and different jerk limit: acceleration and jerk comparison.



(b) Passenger comfort analysis corresponding to Table 7.1: acceleration and jerk comparison.

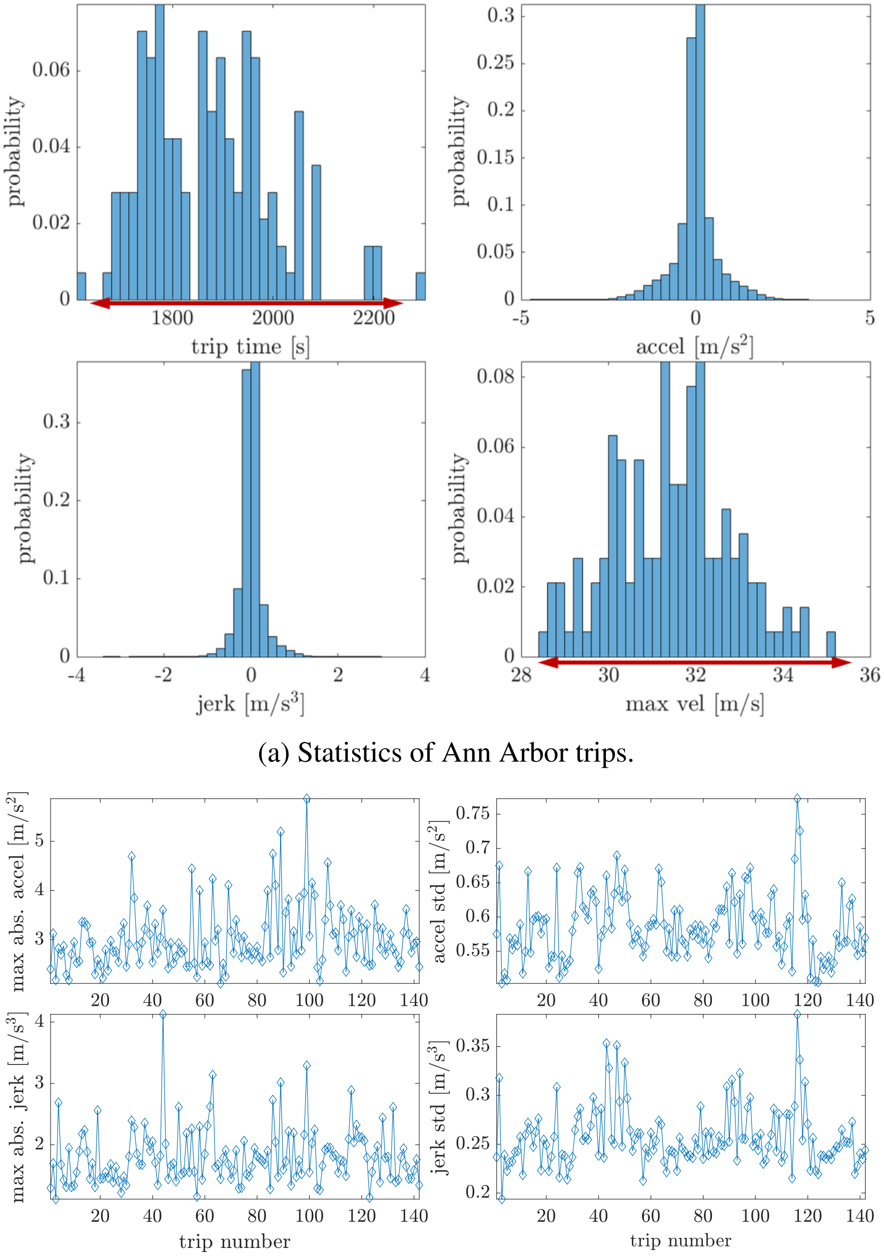
Fig. 7.1 (see Figs. 7.4b(a) and (b)); the driving resulting from the sequential optimization (see Figs. 7.4b(c) and (d)); the driving resulting from the co-optimization with acceleration penalty $Q = 10$ (see Figs. 7.4b(e) and (f)); the driving resulting from the co-optimization with jerk limit $|\Delta a^{\max}| = 0.5$ (see Figs. 7.4b(g) and (h)); and the driving resulting from the co-optimization without additional passenger comfort requirement (see Figs. 7.4b(i) and (j)). The maximum absolute and mean values of acceleration and jerk are summarized in the last four columns in Table. 7.1. The complete passenger comfort analysis for different Q values and jerk limits are shown in Fig. 7.4a.

As seen from Table. 7.1, the driving statistics of the sequential optimization are very close to those of the co-optimization with Q = 10 penalizing acceleration-quadratic term. It reflects the similarity in the vehicle dynamics level between the sequential optimization and the co-optimization with stringent passenger comfort requirements beyond the powertrain-level similarity (fuel consumption), as discussed above. Figure 7.4b(i) indicates that the vehicle-level operation (acceleration) with the full co-optimization is of bang-bang type and the resulting maximum jerk shown in Fig. 7.4b(j) is significantly high and hardly acceptable for a human driver/passenger from a passenger comfort perspective. However, this aggressive driving could still be valuable for more cost-effective and eco-friendly unmanned vehicles (carrying goods – not humans). Remarkably, the co-optimization, whose solution is enabled by DMIS, provides a unified way of investigating the fundamental trade-off between the fuel economy and the passenger comfort requirement. One additional observation is that the driving statistics with different Q values are very similar. This similarity implies the narrow fuel economy benefit margin with varying Q values shown in Fig. 7.3 and explains why jerk instead of acceleration penalty is a better indication of passenger comfort level.

7.3 Statistical Analysis of Offline Implementations

The fuel economy benefits and the drivability analysis are performed on a single trip in the previous section. Although it is important to understand their time-domain responses and the potential where optimized operations can provide large performance improvements, it is also critical to confirm the generality of the fuel economy trends and drivability. To this end, a set of similar trips recorded in Ann Arbor are chosen to statistically compare the performance of different optimization schemes. The trips statistics are summarized in Fig. 7.5. The top left-most subplot in Fig. 7.5a presents the trip times distribution of the considered 142 trips. It can be seen that there can be a more than 10-min difference in trip time. As shown in the bottom two subplots in Fig. 7.5a, there is enough variability in velocity both within a single trip and across different trips. The acceleration and jerk statistics of each trip are summarized in

Fig. 7.5b. The variability across the considered trips is also verified in terms of the maximum absolute acceleration and jerk and standard deviation of acceleration and jerk.

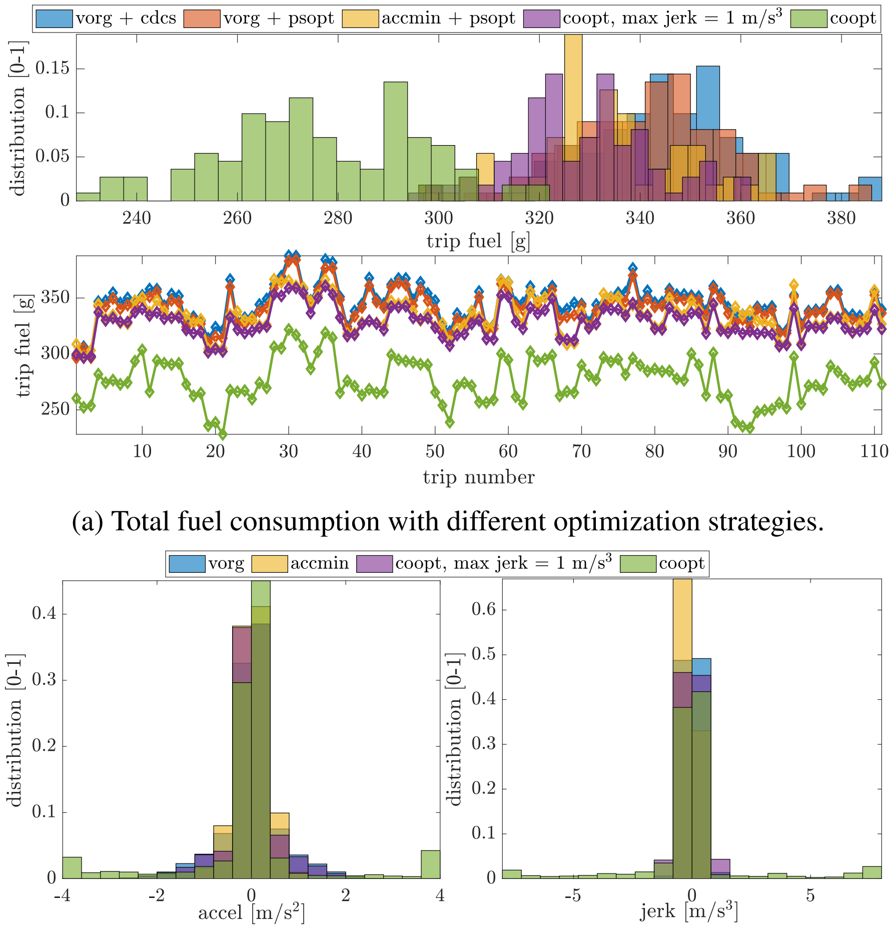


(a) Statistics of Ann Arbor trips.

(b) Acceleration and jerk statistics.

Figure 7.5: Illustration of the variability of the considered Ann Arbor driving data set.

For all the optimizations on all the trips considered, the initial SOC is 0.4 and the desired terminal SOC is 0.35. Since the optimizations are performed offline, it is possible for the actual terminal SOC to be sufficiently close to the desired value. The offline optimization results are compared in Fig. 7.6, where Fig. 7.6a presents the total fuel consumption distribution and



(a) Total fuel consumption with different optimization strategies.

(b) Acceleration and jerk comparison with different optimization strategies.

Figure 7.6: Comparison of offline implementations of different optimization strategies on Ann Arbor trips.

cycle-to-cycle comparison, and Fig. 7.6b shows that resulting acceleration and jerk distributions.

As clearly observed in Fig. 7.6a, the fuel economy with co-optimization in the absence of drivability requirement significantly and systematically outperforms any other optimization strategies. On the other hand, co-optimization where the maximum jerk is limited to be within ± 1 m/s^3 still slightly and systematically outperforms the sequential optimization strategy, where the velocity traces are maximally smoothed. It should also be noted that since the original naturalistic human-driven traces are already relatively smooth, as evident by the moderate acceleration and jerk showed later in Figures 7.5a and 7.5b, the fuel economy potential by doing the sequential optimization is relatively small compared to only PT level optimization performed on human drivers, even if the sequential optimization is a combined VD and PT level optimization. Besides, for a human driver, the potential fuel economy gain through the power-split optimization is relatively small compared to an optimized CDCS.

7.4 Statistical Analysis of Online Implementations of Different Optimization Schemes Under a Unified Framework

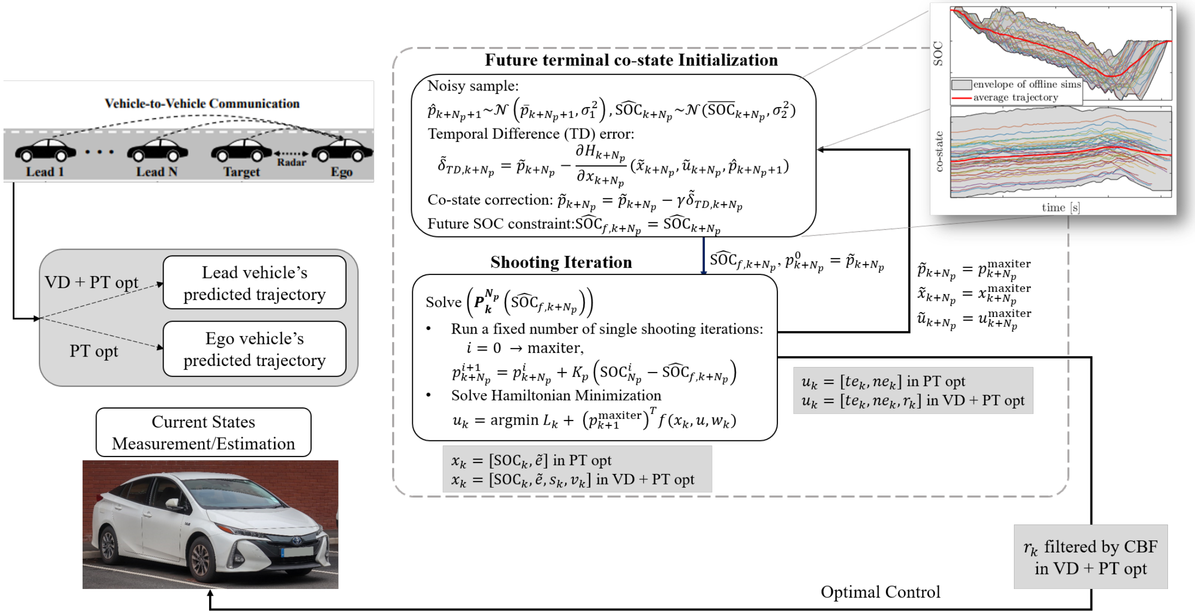

Vehicle-to-Vehicle Communication
Lead 1
Lead N
Target
Radar
Ego
VD + PT opt
PT opt
Lead vehicle's predicted trajectory
Ego vehicle's predicted trajectory
Current States Measurement/Estimation
Future terminal co-state Initialization
Noisy sample:
$\hat{p}_{k+N_p+1} \sim \mathcal{N}\left(\bar{p}_{k+N_p+1}, \sigma_1^2\right), \widehat{\mathrm{SOC}}_{k+N_p} \sim \mathcal{N}(\overline{\mathrm{SOC}}_{k+N_p}, \sigma_2^2)$
Temporal Difference (TD) error:
$\tilde{\delta}_{TD,k+N_p} = \tilde{p}_{k+N_p} - \frac{\partial H_{k+N_p}}{\partial x_{k+N_p}}(\tilde{x}_{k+N_p}, \tilde{u}_{k+N_p}, \hat{p}_{k+N_p+1})$
Co-state correction: $\tilde{p}_{k+N_p} = \tilde{p}_{k+N_p} - \gamma \tilde{\delta}_{TD,k+N_p}$
Future SOC constraint: $\widehat{\mathrm{SOC}}_{f,k+N_p} = \widehat{\mathrm{SOC}}_{k+N_p}$
Shooting Iteration
$\widehat{\mathrm{SOC}}_{f,k+N_p}, p^0_{k+N_p} = \tilde{p}_{k+N_p}$
Solve $\left(\boldsymbol{P}_{\boldsymbol{k}}^{N_p}\left(\widehat{\mathrm{SOC}}_{f,k+N_p}\right)\right)$
• Run a fixed number of single shooting iterations: $i = 0 \rightarrow$ maxiter,
$p^{i+1}_{k+N_p} = p^i_{k+N_p} + K_p\left(\mathrm{SOC}^i_{N_p} - \widehat{\mathrm{SOC}}_{f,k+N_p}\right)$
• Solve Hamiltonian Minimization
$u_k = \mathrm{argmin}\, L_k + \left(p^{\mathrm{maxiter}}_{k+1}\right)^T f(x_k, u, w_k)$
$x_k = [\mathrm{SOC}_k, \tilde{e}]$ in PT opt
$x_k = [\mathrm{SOC}_k, \tilde{e}, s_k, v_k]$ in VD + PT opt
$\tilde{p}_{k+N_p} = p^{\mathrm{maxiter}}_{k+N_p}$
$\tilde{x}_{k+N_p} = x^{\mathrm{maxiter}}_{k+N_p}$
$\tilde{u}_{k+N_p} = u^{\mathrm{maxiter}}_{k+N_p}$
$u_k = [te_k, ne_k]$ in PT opt
$u_k = [te_k, ne_k, r_k]$ in VD + PT opt
SOC
co-state
envelope of offline sims
average trajectory
time [s]
r_k filtered by CBF in VD + PT opt
Optimal Control

Figure 7.7: The unified DMIS-based framework for the PHEV PT-only and different strategies for the combined safe-augmented VD + PT controls

A unified DMIS-based framework from the PT-only control of a human-driven to the combined VD and PT control of an automated PHEV was presented in Chapter 6. In the unified framework, the cost-to-go (the fuel consumption as the economic cost) is represented by the co-state associated with the SOC dynamics, whereas this co-state is corrected both internally using a single shooting method and externally based on the co-state TD-error. In its application to automated PHEVs, a CBF is augmented as an add-on block to modify the vehicle level control input to be executed whenever necessary.

The proposed unified control framework allows for systematically evaluating the fuel economy and drivability performance of different levels and structures of optimization strategies. This section aims to present a solid demonstration of the efficacy of the unified control framework, with evaluation on the same Ann Arbor trips set.

7.4.1 Co-State Nominal Trajectories

The trips #1-50 are used to obtain nominal trajectories (as training set) to be used online, and the trips #51-142 are used to test the MPC performance (as test set). Concretely, for a particular optimization strategy (e.g., co-optimization or power-split optimization), offline optimizations are performed on the first 50 trips, where the velocity trajectory of each trip is either considered to be the lead vehicle's trajectory or the ego vehicle's trajectory. Then, the SOC and its associated co-state trajectories resulting from optimizations performed on the 50 trips as the training set are averaged to get the nominal trajectories corresponding to each optimization scheme. The concept and results of obtaining the nominal trajectories are demonstrated in Fig. 7.8[2]. The colored lines are trajectories resulting from particular trips, and the shadowed grey area denotes the envelope of the results on the training set. It can be seen that there is enough variability in the training data set. The thick red lines are the averaged SOC and its associated co-state trajectories and will be used as nominal trajectories in the online MPC implementation. Note that in the absence of the offline data, it can also be obtained online, through rollout algorithms and MPC [92]. An example of the online method based on MPC with aggregation is presented in the optimized power-split control of a human-driven PHEV [3].

In all the MPC implementations, 10 total shooting iterations are performed per MPC step. A prediction horizon of 10-sec is used in all the MPC implementations. The discount factor as discussed in Section 6.2 is selected to be $\rho = 0.99$.

7.4.2 Comparison of Online Optimization Results

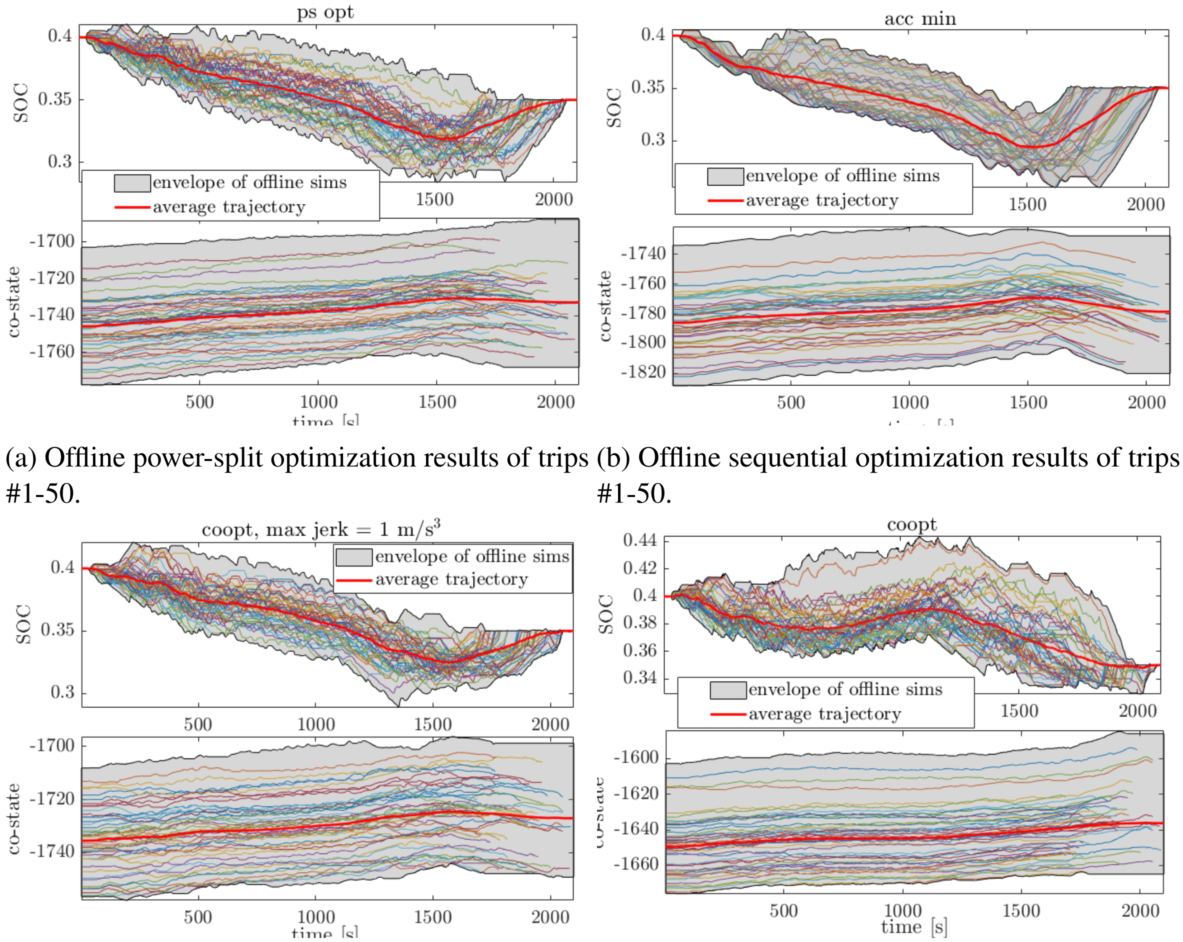
In this section, the fuel consumption results on the unseen Ann Arbor trips are compared with different MPC implementations. The following MPC strategies are considered

1. **psopt MPC**: A PT level controller, where the considered PHEV is assumed to be driven by a human. Only its power-split operation is optimized. The predicted lead vehicle's trajectory is used as the PHEV's trajectory for predictive energy management. Its nominal SOC and associated co-state trajectories are shown in Fig. 7.8a.

2. **accmin + psopt MPC**: A sequential VD + PT level controller. The vehicle-following dynamics are first optimized by minimizing the acceleration. The first step acceleration is modified whenever necessary by the CBF to ensure safe distance gap. Its nominal SOC and associated co-state trajectories are shown in Fig. 7.12b.

3. **coopt, max jerk = 1 m/s^3**: A centralized VD + PT level controller. An additional drivability requirement is added to limit the maximum jerk to be within ±1 m/s^3. The first step acceleration is modified whenever necessary by the CBF to ensure safe distance gap. Its nominal SOC and associated co-state trajectories are shown in Fig. 7.8c.

[2] Note that the optimal SOC trajectories do not present linear-in-time features. It is because while the total allowable SOC span is small, and the SOC is allowed to be freely placed (no SOC inequality constraint).



(a) Offline power-split optimization results of trips #1-50.

(b) Offline sequential optimization results of trips #1-50.

(c) Offline co-optimization results of trips #1-50.

(d) Offline co-optimization results of trips #1-50.

Figure 7.8: Offline simulation results on 50 similar trips and their corresponding nominal trajectories as an average of the offline simulation results.

4. **coopt**: A centralized VD + PT level controller without drivability requirement. The first step acceleration is modified whenever necessary by the CBF to ensure safe distance gap. Its nominal SOC and associated co-state trajectories are shown in Fig. 7.8d.

The comparison of the fuel consumption of the undriven Ann Arbor trips with different optimization schemes is shown in Fig. 7.9. It can be seen that the co-optimization without any drivability requirement outperforms all the other optimization schemes in terms of the fuel economy, even in their online implementations, despite all the prediction errors and incomplete numerical solutions to the optimization problems. The sequential optimization results lie in the middle and are similar to the co-optimization results with additional drivability requirements. All the MPC implementations with vehicle-level optimization outperform the energy management only optimization.

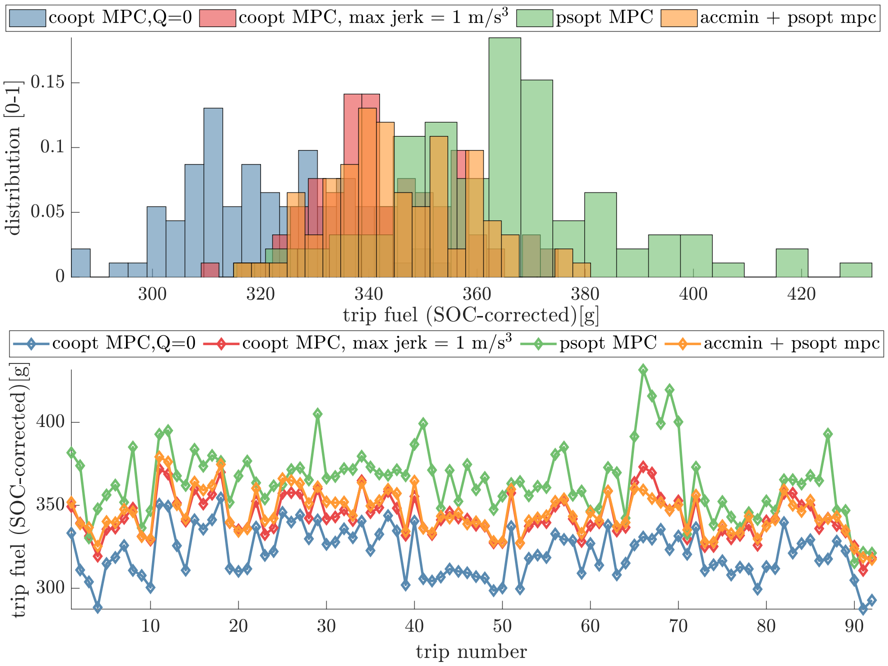


Figure 7.9: Comparison of total fuel consumption.

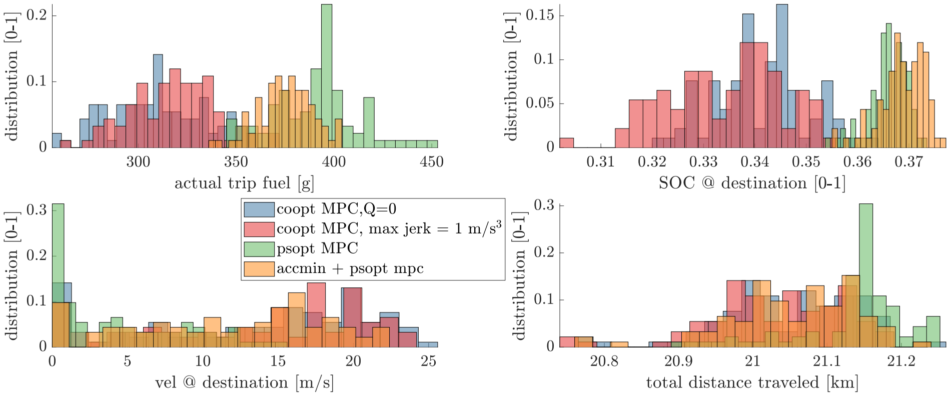
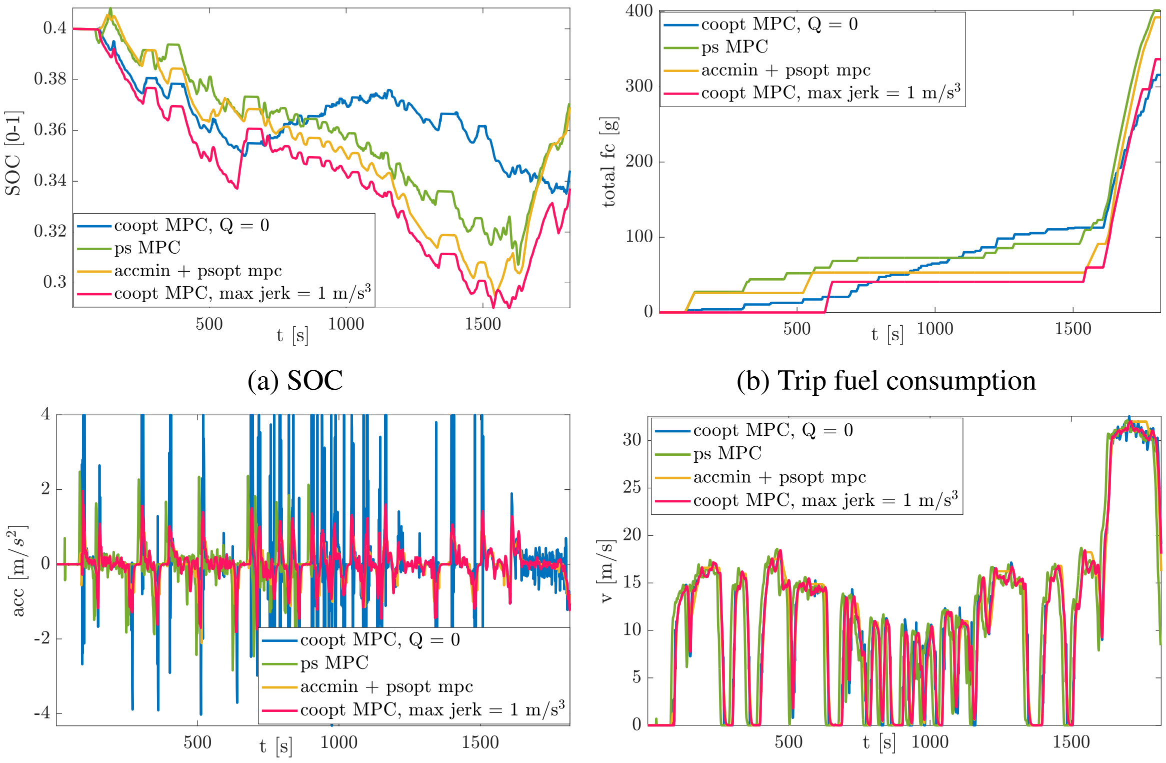


Figure 7.10: Comparison of actual total fuel consumption and final states, including final SOC, velocity and total distance traveled.

7.4.2.1 Time-Domain Responses on Several Ann Arbor Trips

In this section, the time domain responses of the resulting receding horizon implementation of different optimization strategies are presented. The result of a trip with short travel time is presented in Fig. 7.11, and the result of a trip with long travel time is presented in Fig. 7.12.

As observed in Figs. 7.11 and 7.12, in both cases, the terminal SOC with power-split only optimization and sequential optimization based control lead to non-trivial higher-than-desired value. It indicates the design of the training process should be improved as discussed in Chapter 6, or a better co-state update strategy is needed towards the end of the trip to avoid



(a) SOC

(b) Trip fuel consumption

(c) Acceleration

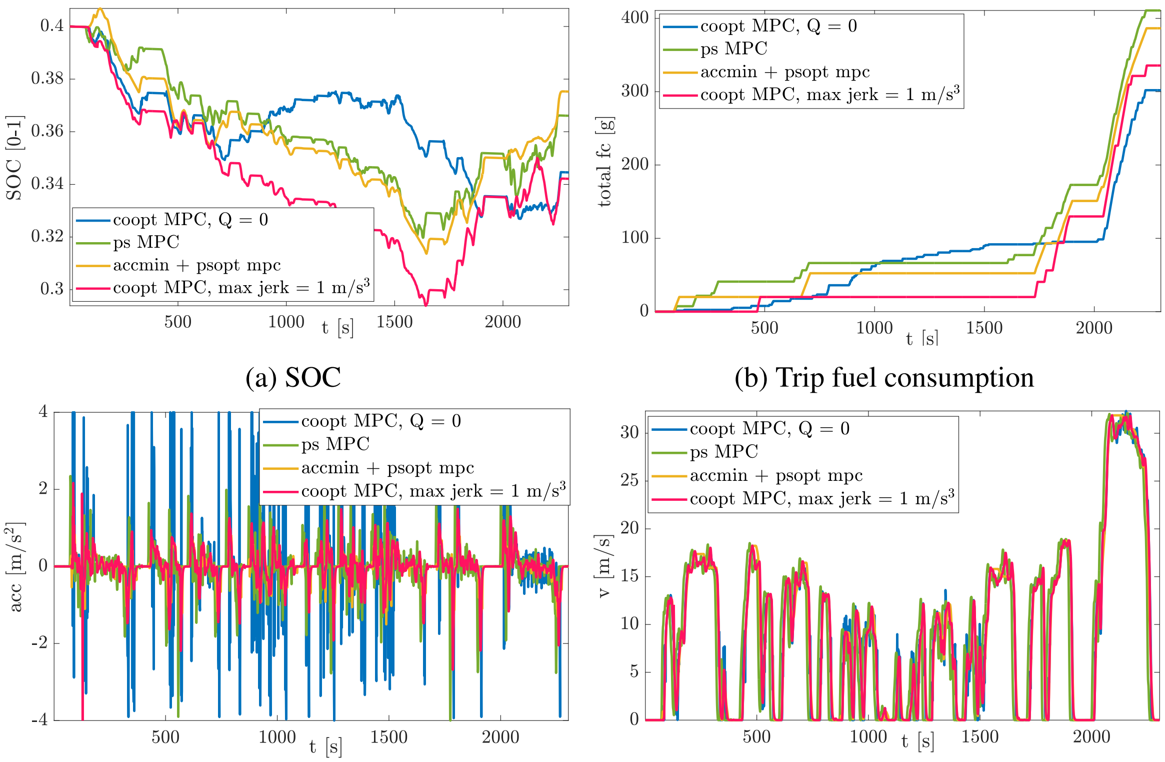
(d) Velocity

Figure 7.11: Time-domain response when the lead vehicle's trajectory is trip #77.

unnecessary battery charging.

7.4.3 Effectiveness of CBF

The effectiveness of CBF is demonstrated in Fig.7.13. As can be seen from the top left and the bottom left subplots, the minimum time-headway constraints are violated in both the acc min + psopt MPC implementation and the coopt MPC implementation. The constraint violations in accmin + psopt are due to the inaccuracies in predicting the lead vehicle's trajectories. In addition to the prediction inaccuracy, the incomplete single shooting iterations for computation efficiency will exaggerate the constraint violation. The constraint violation is added directly to the stage cost with the exterior penalty method. The solution quality, dominated by the number of single shooting iterations, will directly affect the constraint satisfaction. The minimum time-headway constraints are considered to be safe-critical for the vehicle the following control. Significant violation of minimum time-headway constraints could lead to rear-end collisions. The pure DMIS-based MPC implementations, without additional CBF-based safety filter may not be acceptable. By introducing CBF, the vehicle-level control is filtered such that the time-headway constraints are always satisfied.



(a) SOC

(b) Trip fuel consumption

(c) Acceleration

(d) Velocity

Figure 7.12: Time-domain response when the lead vehicle's trajectory is trip #58.

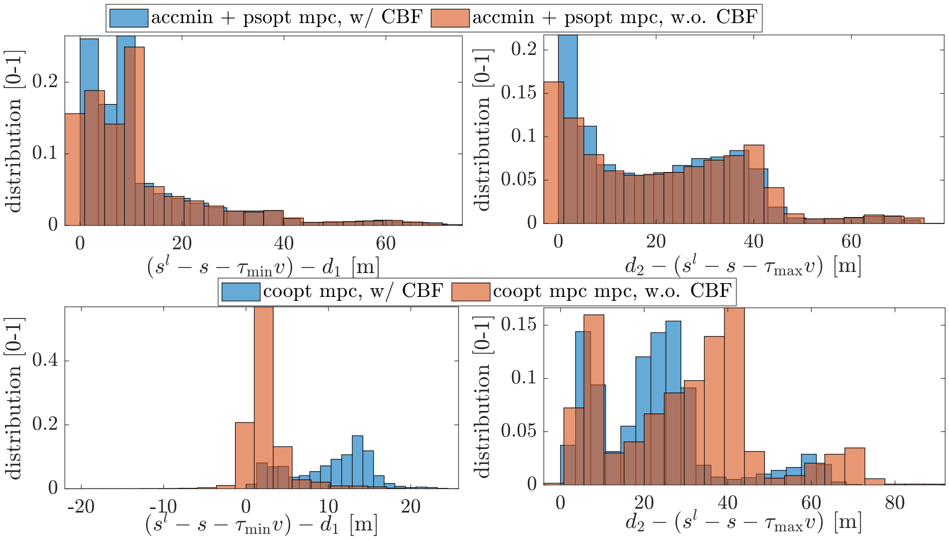


Figure 7.13: Comparison of MPC results with and without CBF.

7.5 Summary

This chapter seeks to provide a systematic way of benchmarking the fuel economy performance of the centralized velocity and powertrain co-optimization of a PHEV against the CDCS strategy typically implemented on a human-driven PHEV, and a layered decentralized optimization structure of an autonomous PHEV where the velocity is first smoothed via acceleration minimization, followed by the power-split optimization [32]. An important observation is that the fuel economy advantage obtained with the co-optimization decreases to a similar level as in the sequential optimization as the drivability requirement on the co-optimization increases, providing valuable instructions for selecting control framework under different scenarios emphasizing fuel economy or passenger comfort.

Afterward, the chapter presents a unified DMIS-based framework from the powertrain (PT)-only control of a human-driven to the combined vehicle dynamics (VD) and powertrain (PT) control of an automated plug-in hybrid electric vehicle (PHEV). In the unified framework, the co-state represents the cost-to-go, whereas the co-state is corrected both internally as in single shooting and externally based on the co-stat TD-error. In the application of automated PHEVs, a control barrier function (CBF) design is presented as an add-on block to modify the vehicle level control input to be executed whenever necessary. The unified control framework guarantees safe vehicle-following behavior when including longitudinal vehicle dynamics control. It is shown to generate consistent fuel economy benefits on a large number of unseen trips.

CHAPTER 8

Conclusion and Future Work

8.1 Results and Conclusions

In 2016, the department of energy (DOE) launched a project on the Next-Generation Energy Technologies for Connected and Automated On-Road Vehicles (NEXTCAR). The project aims at an additional 20% reduction in energy consumption of future connected and automated vehicles enabled by technologies that use connectivity and automation to co-optimize vehicle dynamic controls and powertrain operation. Vehicle dynamics and powertrain control technologies, implemented on a single-vehicle basis, across a cohort of cooperating vehicles, or across the entire vehicle fleet, could significantly improve individual vehicle and, ultimately, fleet energy efficiency [98]. To date, the funded teams have generated abundant work: on CAV overview [99, 100], traffic intersection management/crossing [101, 102], eco-routing and eco-driving/eco-ACC [103, 104, 105], and combined vehicle dynamics and powertrain control of CAVs [106, 107], just to name a few.

In the year 2021 (by the time of thesis submission), there have been more projects funded by DOE in the field of traffic control, utilization of data with connectivity, artificial intelligence for CAVs, characterization and validation of the behaviors of CAV controls, and metrics for assessing the impacts of energy-efficient mobility systems [108]. From individual technologies to technology integration to tools for evaluation and assessment, new technologies are actively created that provide economic benefits through enhanced mobility.

With their immense potential for increasing the country's energy security, economic vitality, and quality of life, plug-in electric vehicles (PEVs) – including PHEVs and all-EVs – will play a vital role in the country's transportation future [109]. Meanwhile, recent rapid advancements in driver assistance technologies and the deployment of vehicles with increased levels of connectivity and automation have created multiple opportunities to improve the efficiency of future vehicles in new ways. As part of the NEXTCAR project, this thesis aims to utilize CAV technologies for better powertrain control and combined vehicle dynamics and powertrain control design through optimization algorithms to improve individual PHEVs' energy efficiency significantly. In particular, this thesis aims to walk from "what is possible" to " what is practical".

This thesis starts by constructing a strategy to improve the powertrain control of a human-driven PHEV utilizing connectivity. First, battery SOC reference waypoints are obtained by a simplified speed profile constructed from segmented traffic information, typically available from mobile mapping applications. Then, a P-ECMS is proposed to adjust its co-state based on the difference between the future SOC obtained from short-horizon prediction and a future reference SOC from the node planning.

Pushing forward the fuel-efficient control of a PHEV beyond a ECMS-type of instantaneous optimization for its powertrain operation necessitates considering a trajectory optimization (i.e., optimal control) problem. As a hybrid system, the optimal control of PHEVs requires the numerical solution of optimization problems of mixed-integer type. Moreover, with an increased number of states and control inputs resulting from the engine cranking and combined vehicle and power level control, it is of interest to have a unified numerical algorithm for solving mixed-integer optimal control problems with many states and control inputs. Based on a Discrete Maximum Principle (DMP), a Discrete Mixed-Integer Shooting (DMIS) algorithm is proposed. The DMIS is demonstrated in successfully addressing the cranking fuel optimization in the energy management of a PHEV. It also serves as the foundation of the co-optimization problem considered in the remaining part of the thesis.

With the numerical tool developed, different control designs are presented in this thesis with the increased vehicle automation level combining vehicle dynamics and powertrain of PHEVs in within-a-lane traffic flow. This thesis starts with a sequential, decentralized optimization and then advances to direct fuel minimization by simultaneously optimizing the two subsystems in a centralized manner. When shifting toward online implementation, the striking challenge lies in the conflict between the long control horizon required for global optimality and the computational power limit. In this thesis, a receding horizon strategy is proposed to resolve the conflict between the horizon length and the computation complexity with co-states approximating the future cost, thus shortening the prediction horizon. In particular, the co-state is updated using a nominal trajectory and the TD error based on the co-state dynamics.

The remaining work aims at developing a unified DMIS-based receding horizon framework from the PT-only control of a human-driven to the combined VD and PT control of an automated PHEV. In the unified framework, the cost-to-go (the fuel consumption as the economic cost) is represented by the co-state associated with the SOC dynamics. In its application to automated PHEVs, a CBF is augmented as an add-on block to modify the vehicle level control input to be executed whenever necessary. The unified control framework guarantees safe vehicle-following behavior when including longitudinal vehicle dynamics control. Furthermore, it allows for systematically evaluating the fuel economy and drivability performance of different levels and structures of optimization strategies.

8.2 Future Work and Open Challenges

In this thesis, the fuel economy of the receding-horizon co-optimization of the vehicle longitudinal and powertrain dynamics of a PHEV was studied and validated through simulations. The next step would be validation of the results on a vehicle and test whether the proposed control algorithms developed are easily implementable on a vehicle ECU. In particular, the real-time implementability and reliability of the DMIS algorithm applying to the co-optimization problem are crucial to investigate. The sequential eco-driving and energy management framework has already been implemented. Therefore, it is of great interest to evaluate the sequential and co-optimization framework in fuel economy and passenger acceptance.

Improving the offline training and sample strategy for more effective “ground-truth” state and co-state values should also be considered in future work. As discussed in Section 6, currently, the nominal state and co-state trajectories are average trajectories of those obtained from the offline optimization. However, the distribution is less likely to be normally distributed at the beginning and towards the end of the trip due to the initial and terminal SOC requirements. It indicates a possible more reasonable way to obtain nominal trajectory: sample from a normal distribution of the initial and terminal SOC, do offline optimization, and get average trajectories, instead of a fixed, same initial and terminal SOC. Also, right now, the weight of each trajectory is the same. It might be better to consider different weights on different trajectories to get nominal trajectories.

Although the work presented in this thesis is application-driven, it would be interesting to provide more formal justification from a theoretical point of view, as preliminary investigated in Section 6.3.4, where the real system is modeled as a Markov Decision Process (MDP).

Finally, it would be interesting to extend and evaluate the DMIS-based MPC framework to other applications.

APPENDIX A

Control-Oriented Hybrid Powertrain Model and Static Map Regressions

A.1 PHEV Powertrain Control-Oriented Model

In this thesis, the 2017 Prius Prime PHEV is considered as the target powertrain architecture. The inputs to the model are the engine torque te and the engine speed ne, and the outputs of the main concern from the model are the fuel rate $\dot{m}_f$ and the change in SOC $\dot{\text{SOC}}$. The overall schematic view of the hybrid powertrain is shown in Fig. A.1. The minimum fuel consumption problems in Chapter 2, 3 and 4, when formulated in continuous-time, do not include the additional fuel consumption associated with engine cranking. Therefore, in those chapters, the SOC is considered as the only state. The minimum fuel consumption problems considered in all the other chapters are formulated directly in discrete time and account for the additional fuel consumption due to engine cranking. In those chapters, two states are considered in the powertrain dynamics, including the SOC and the normalized engine cranking $\tilde{e}$. The SOC dynamics are the focus of this section when developing the control-oriented model.

This chapter first revisits the control-oriented model for the SOC dynamics for the control design, simulation, and analysis throughout this thesis with great details. First, each critical component is modeled separately. Afterward, several simplifications are made, which leads to a control-oriented model used throughout this thesis. Static maps are heavily involved in the control-oriented model (A.14), including battery OCV, internal resistance, MG1 and MG2 power losses, and fuel rate. To make the simulation results presented in this thesis reproducible, regression models are established for powertrain components of the considered PHEV in this chapter. Notably, efforts are devoted to striking a balance between the model accuracy, compliance with the company data privacy, and the numerical difficulty resulting from smoothed regression models. It will be detailed in the second half of this chapter. Two main techniques are adopted in establishing regression models: the polynomial and the NN regression. Polynomial regressions are used to approximate the battery OCV and internal resistance, and the NNs are used to approximate the power loss and the fuel rate maps.

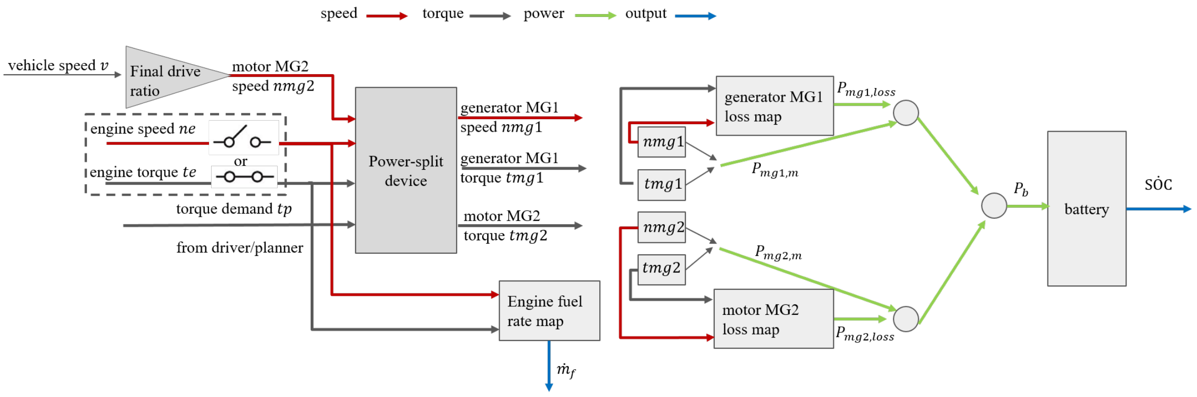


Figure A.1: A schematic control oriented model for Prius Prime PHEV

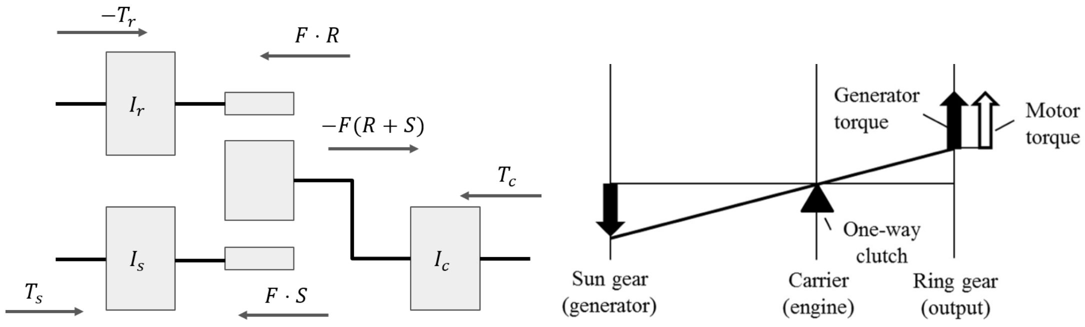
A.1.1 Modeling of the Power Split Device and One-way Clutch

The Prius new PHV transaxle for a compact vehicle installed in the new Prius Prime has a four-axle structure. It consists of a torsional damper with a torque limiter, one-way clutch, input shaft, planetary gear, generator, motor, reduction device, and differential device[4, 110].

The rotational speeds of the ring gear ω_r, sun gear ω_s, and the carrier gear ω_c satisfy the following relationship at all times:

$$\omega_s \cdot S + \omega_r \cdot R = \omega_c \cdot (R + S), \tag{A.1}$$

where S and R are the radii of the sun gear and the ring gear, respectively.



(a) Force analysis on a planetary gear set (adapted based on [111])

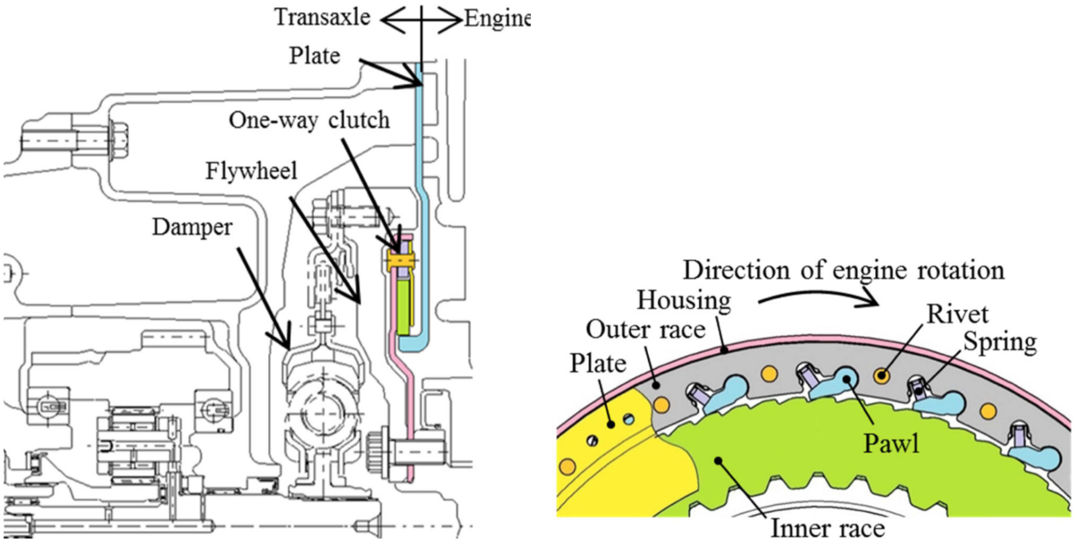
(b) Dual Motor Drive System [4]

Figure A.2: Free body diagram of the mechanical path.

The forces on the planetary gear set is shown in Fig. A.2a. The dynamics of the gear nodes are obtained as:

$$I_r \cdot \dot{\omega}_r = F \cdot R - T_r, \quad I_c \cdot \dot{\omega}_c = T_c - F \cdot (R + S), \quad I_s \cdot \dot{\omega}_s = F \cdot S - T_s, \tag{A.2}$$

where I_r, I_c, and I_s are the inertia of the ring, carrier and sun gear, respectively. T_r, T_c are the ring gear and carrier gear torque, respectively.



(a) Cross-section of one-way clutch installation configuration.

(b) Internal structure of one-way clutch.

Figure A.3: Illustration of the one-way clutch in the Prius Prime powertrain [4].

One noticeable new feature of 2017 Prius Prime is the addition of a one-way clutch (OWC) [110, 4] which prevents the engine from rotating in the reverse direction. Moreover, this new feature allows for extended driving in EV mode as the motor and generator can be used simultaneously for vehicle propulsion when the engine stops. This particular mode is referred to as DM. Fig. A.3a shows the cross-section of one-way clutch installation configuration. Fig. A.3b shows its internal structure. The housing (pink), outer race (gray), pawls (blue) springs, and plate (yellow) are integrated and rotate together. The housing is connected to the flywheel (engine) and rotates together with the engine. The inner race (green) engages with the plate (blue) between engine and transaxle through spline engagement and is fixed to the transaxle. There is friction between the plate (yellow) and the inner race (green). Thus under normal HV mode, the engine generates torque and rotates in the forward direction. The torque will flow from the engine (flywheel) to the damper (conventional clutch) and the input shaft and eventually to the carrier gear. However, suppose the engine stops ($ne = 0, te = 0$). The reverse direction-driven torque is from carrier gear (reverse direction torque from the generator), which will drive the engine to rotate in the reverse direction. In that case, the pawls and inner race will engage. Because the inner race is fixed to the transaxle, it will prevent the engine from rotating in the reverse direction. Moreover, the engagement between the inner race and fixed transaxle under this scenario generates an additional clutch torque T_{cl}, which counteracts the carrier gear force T_c, and therefore allows the reaction force F to be transmitted to the ring gear axis as shown in Fig. A.2b. Consequently, the use of the one-way clutch adds a DM to the original EV mode.

For the MG1 (generator) at the sun gear side, the governing equation is:

$$I_{mg1} \cdot \dot{n}mg1 = T_s + tmg1, \tag{A.3}$$

where $nmg1$, $nmg1$, and I_{mg1} are the MG1 torque, speed and inertia respectively.

Combined with sun gear dynamics and $nmg1 = \omega_s$, we get:

$$(I_{mg1} + I_s) \cdot \dot{n}mg1 = tmg1 + F \cdot S. \tag{A.4}$$

For the engine at the carrier gear side, the governing equation is:

$$I_e \cdot \dot{n}e = te + T_{cl} - T_c, \tag{A.5}$$

where te, ne and I_e are the engine torque, speed and inertia respectively. T_{cl} is the clutch torque. For different modes, either T_e or T_{cl} will be 0.

Combined with sun gear dynamics and $ne = \omega_c$, we get:

$$(I_e + I_c) \cdot \dot{n}e = te + T_{cl} - F \cdot (R + S). \tag{A.6}$$

For the motor and the motor reduction gear at the ring gear side, the governing equation is:

$$\Big(\frac{M \cdot R_{tire}^2}{(K_1 \cdot K_2)^2} + \frac{I_{mg2}}{(K_2)^2}\Big)\dot{\omega}_r = T_r + \frac{tmg2}{K_2} - \frac{T_f + (F_{aero} + F_{roll} - F_{grad})R_{tire}}{K_1 \cdot K_2} \tag{A.7}$$

where K_1 is the counter gear ratio, K_2 is the motor reduction gear ratio, M is the vehicle mass, R_{tire} is the tire radius, T_f is the additional friction brake ($T_f \geq 0$), F_{roll} is the rolling resistance, F_{aero} is the aerodynamic force, F_{grad} is the grade force, $tmg2$ is the MG2 torque, I_{mg2} is the MG2 inertia. The rolling resistance F_{roll} is calculated as:

$$F_{roll} = C_f \cdot M \cdot g \cdot \cos\theta, \tag{A.8}$$

where C_f is the rolling resistance coefficient, g is the gravity acceleration, θ is the road grade which is assumed to be positive when vehicle is driven down a hill.

The aerodynamic force F_{aero} is calculated as:

$$F_{aero} = \frac{1}{2} \cdot \rho \cdot A_f \cdot C_d \cdot v^2, \tag{A.9}$$

where ρ is the air density, A_f is the frontal area of the vehicle, C_d is the drag coefficient, v is the vehicle velocity. An additional kinematic relationship between vehicle speed v and ring gear speed ω_r is obtained as:

$$\omega_r = \frac{v}{R_{tire}} \cdot K_1 \cdot K_2. \tag{A.10}$$

The grade force F_{grad} is obtained as:

$$F_{grad} = M \cdot g \cdot \sin\theta. \tag{A.11}$$

On the output shaft, the torque transmitted by the ring gear the motor are combined:

$$T_{out} = (T_{MG2} + K_2 \cdot T_r) \cdot K_1 \tag{A.12}$$

The driver demanded torque T_{driver} follows:

$$T_{driver} = T_{out} - T_f \qquad (T_f \geq 0) \tag{A.13}$$

A.1.2 Model Simplification and Final Control-oriented Model

For the purpose of reducing calculation burden and SOC planning, it is justified to reduce the state system to contain only SOC by assuming in $\dot{\omega}_r = \dot{\omega}_c = \dot{\omega}_s = 0$ [111] in (A.2). The battery SOC is the main state in the charge depletion optimization of a PHEV. The battery SOC dynamics depends on battery power P_{batt}:

$$\dot{\text{SOC}} = -\frac{V_{oc} - \sqrt{V_{oc}^2 - 4P_{batt}R_{batt}}}{2C_{batt}R_{batt}}, \tag{A.14}$$

with

$$P_{batt} = P_{MG2,e} + P_{MG1,e}, \tag{A.15}$$

where $V_{oc} = V_{oc}(\text{SOC})$ and $R_{batt} = R_{batt}(\text{SOC})$ are the battery open-circuit voltage and internal resistance, respectively, and are functions of SOC; P_{batt} is the battery power; $P_{MG2,e}$

and $P_{MG1,e}$ are the motor and generator electrical power, respectively. The electrical power from the electrical machines are represented by:

$$\begin{aligned} P_{MG1,e} &= tmg1 \cdot nmg1 + P_{MG1,loss}, \\ P_{MG2,e} &= tmg2 \cdot nmg2 + P_{MG2,loss}, \end{aligned} \tag{A.16}$$

where $P_{MG1,loss}$ and $P_{MG2,loss}$ are motor and generator power losses, respectively, which are provided by the manufacturer. Given the vehicle speed v, the motor (MG2) speed $nmg2$ is determined by:

$$nmg2 = \frac{v}{R_{tire}} K_1. \tag{A.17}$$

The driver torque demand T_{driver} satisfies:

$$T_{driver} = T_{out} - T_f, \qquad (T_f \geq 0), \tag{A.18}$$

where T_f is additional friction brake; T_{out} is the powertrain output torque and is defined in (A.24). The driver torque demand T_{driver} is calculated from the vehicle longitudinal dynamics with v and $\dot{v}$ as inputs:

$$M_{eff}\dot{v} = \frac{T_{driver}}{R_{tire}} - \frac{1}{2}\rho A_f C_d v^2 - C_f M g \cos\theta + M g \sin\theta. \tag{A.19}$$

The speed of motor, generator, and engine are coupled via the power split device through the following kinematic relationship:

$$nmg1 \cdot S + nmg2 \cdot K_2 \cdot R = ne \cdot (R + S), \tag{A.20}$$

The torque coupling among motor $tmg2$, generator $tmg1$, engine te and clutch T_{cl} are related through the following relationships:

In Hybrid Vehicle (HV) & Electric Vehicle (EV) mode

$$tmg1 + F \cdot S = 0, \quad te - F \cdot (R + S) = 0 \tag{A.21}$$

In Dual Motor Drive (DM) mode

$$tmg1 + F \cdot S = 0, \quad T_{cl} - F \cdot (R + S) = 0. \tag{A.22}$$

The engine torque te and generator torque $tmg1$ are combined to the ring gear torque T_r

through reaction force F:

$$T_r = F \cdot R. \tag{A.23}$$

On the output shaft, the torque transmitted by the ring gear T_r and the torque from the motor $tmg2$ are combined as

$$T_{out} = (tmg2 + K_2 \cdot T_r) \cdot K_1. \tag{A.24}$$

A static map ϕ, that was experimentally determined by the manufacturer of the considered vehicle, is used to calculate the fueling rate $\dot{m}_f$ based on engine torque te and engine speed, ne given by:

$$\dot{m}_f = \phi(te, ne). \tag{A.25}$$

Meanwhile, the following physical constraints, usually time-varying, must be enforced at each time instant:

$$\text{SOC}^{\min}(t) \leq \text{SOC}(t) \leq \text{SOC}^{\max}(t) \tag{A.26a}$$

$$P_{batt}^{\min}(t) \leq P_{batt}(t) \leq P_{batt}^{\max}(t) \tag{A.26b}$$

$$ne^{\min}(t) \leq ne(t) \leq ne^{\max}(t) \tag{A.26c}$$

$$te^{\min}(t) \leq te(t) \leq te^{\max}(t) \tag{A.26d}$$

$$nmg1^{\min}(t) \leq nmg1(t) \leq nmg1^{\max}(t) \tag{A.26e}$$

$$tmg1^{\min}(t) \leq tmg1(t) \leq tmg1^{\max}(t) \tag{A.26f}$$

$$nmg2^{\min}(t) \leq nmg2(t) \leq nmg2^{\max}(t) \tag{A.26g}$$

$$tmg2^{\min}(t) \leq tmg2(t) \leq tmg2^{\max}(t) \tag{A.26h}$$

In this thesis, The following simplifications are made: 1) the battery SOC limits (A.26a) are assumed to be inactive since the SOC range, and the battery capacity is considerably large in a PHEV; therefore, these limits are not considered in the optimization, 2) the constraints on the motor and the generator in HV mode (A.26e)–(A.26h) are converted to the constraint on the engine torque and speed, together with (A.26c)–(A.26d), denoted as $\Omega(v)$, 3) the battery power limits (A.26b) and the constraints on the motor and the generator in EV mode are assumed to be inactive for a human-driven PHEV. This is because the power demand from a human is typically not very aggressive. For an automated PHEV, the constraints in EV mode are handled by torque saturation.

Remark 9. Note that although each individual term of the friction force is modeled in this section, in most of the thesis, a regression model is used to calculate the torque demand directly.

Remark 10. Despite the presence of an additional Dual Motor Drive mode, in this thesis, the considered PHEV is still assumed to operate under either HV or EV mode.

A.2 Regression Models and Validations

As discussed at the beginning of this chapter, to make the simulation results presented in this thesis reproducible, regression models for powertrain components of the considered PHEV need to be established. Two main techniques are adopted in establishing regression models: the polynomial and the NN regression. Polynomial regressions are used to approximate the battery OCV and internal resistance, and the NNs are used to approximate the power loss and the fuel rate maps.

A.2.1 Regression Models for Battery

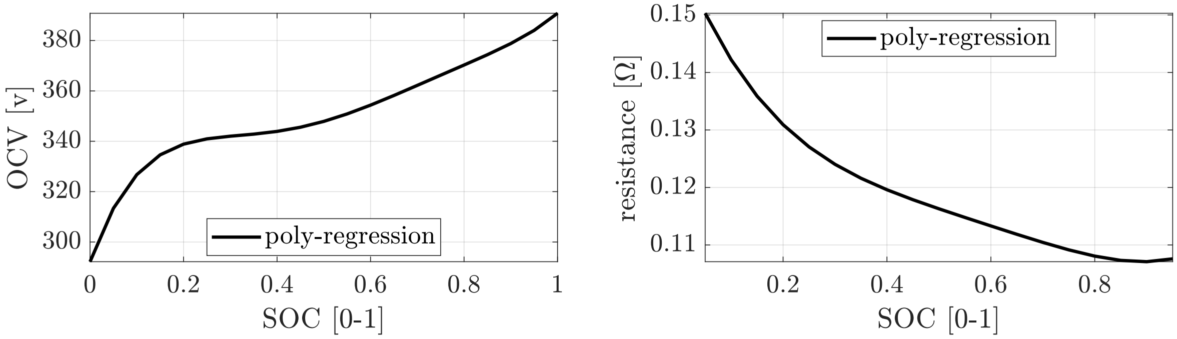


Figure A.4: Polynomial regression models for battery OCV and internal resistance.

A fifth-order polynomial is used to approximate the battery OCV.

$$\widehat{\mathrm{OCV}} = \alpha_0^{\mathrm{SOC}} + \alpha_1^{\mathrm{SOC}} \cdot \mathrm{SOC} + \alpha_2^{\mathrm{SOC}} \cdot \mathrm{SOC}^2 + \alpha_3^{\mathrm{SOC}} \cdot \mathrm{SOC}^3 + \alpha_4^{\mathrm{SOC}} \cdot \mathrm{SOC}^4 + \alpha_5^{\mathrm{SOC}} \cdot \mathrm{SOC}^5, \tag{A.27}$$

Define the feature matrices $X^{\mathrm{SOC}} = \begin{pmatrix} 1 & \mathrm{SOC}_{(1)} & \mathrm{SOC}_{(1)}^2 & \dots & \mathrm{SOC}_{(1)}^5 \\ 1 & \mathrm{SOC}_{(2)} & \mathrm{SOC}_{(2)}^2 & \dots & \mathrm{SOC}_{(2)}^5 \\ & & \dots & & \\ 1 & \mathrm{SOC}_{(n)} & \mathrm{SOC}_{(n)}^2 & \dots & \mathrm{SOC}_{(n)}^5 \end{pmatrix}$, and the actual output matrix $Y^{\text{SOC}} = \begin{pmatrix} \text{OCV}_{(1)} \\ \vdots \\ \text{OCV}_{(n)} \end{pmatrix}$, where $[\text{SOC}_{(1)}, \cdots, \text{SOC}_{(n)}], [\text{OCV}_{(1)}, \cdots, \text{OCV}_{(n)}], n \in \{1, \cdots, n\}$ denotes the sampled n date points from the actual SOC-OCV map. The parameter set $\alpha^{\text{SOC}} = (\alpha_0^{\text{SOC}}, \cdots, \alpha_5^{\text{SOC}})$ is obtained by $\alpha^{\text{SOC}} = X^{\text{SOC}} \backslash Y^{\text{SOC}}$.

A fourth-order polynomial is used to approximate the battery internal resistance.

$$\hat{r} = \alpha_0^r + \alpha_1^r \cdot \text{SOC} + \alpha_2^r \cdot \text{SOC}^2 + \alpha_3^r \cdot \text{SOC}^3 + \alpha_4^r \cdot \text{SOC}^4, \tag{A.28}$$

where $\hat{r}$ denotes the approximated battery internal resistance. Define the feature matrices $X^r = \begin{pmatrix} 1 & r_{(1)} & r_{(1)}^2 & \cdots & r_{(1)}^4 \\ 1 & r_{(2)} & r_{(2)}^2 & \cdots & r_{(2)}^4 \\ & \cdots & & & \\ 1 & r_{(n)} & r_{(n)}^2 & \cdots & r_{(n)}^4 \end{pmatrix}$, and the actual output matrix $Y^r = \begin{pmatrix} r_{(1)} \\ \vdots \\ r_{(n)} \end{pmatrix}$, where $[\text{SOC}_{(1)}, \cdots, \text{SOC}_{(n)}], [r_{(1)}, \cdots, r_{(n)}], n \in \{1, \cdots, n\}$ denotes the sampled n date points from the actual SOC-resistance map. The parameter set $\alpha^r = (\alpha_0^r, \cdots, \alpha_5^r)$ is obtained by $\alpha^r = X^r \backslash Y^r$.

The resulting parameters for the OCV regression are $\alpha_0^{\text{SOC}} = 0.7502$, $\alpha_1^{\text{SOC}} = 1.3355$, $\alpha_2^{\text{SOC}} = -5.4498$, $\alpha_3^{\text{SOC}} = 10.5698$, $\alpha_4^{\text{SOC}} = -9.2707$, and $\alpha_5^{\text{SOC}} = 3.0684$. The resulting parameters for the internal resistance regression are $\alpha_0^r = 0.1604$, $\alpha_1^r = -0.2266$, $\alpha_2^r = 0.4976$, $\alpha_3^r = -0.5616$, and $\alpha_4^r = 0.2392$. The resulting curves are shown in Fig. A.4.

A.2.2 Regression Models for Electric Motors and the Fuel Rate Map

As a linear regression technique, polynomials do not have enough capability to approximating maps due to their underlying smoothness. As a result, neural network models are used to capture some of the non-smoothness embedded in the motor efficiency maps.

A two-layer NN with 8 neurons per layer is used to get the regression models for MG1 and MG2. The resulting regression models are shown in Fig. A.5. To capture some of the non-smoothness embedded in the fuel rate map, a two-layer NN with 8 neurons per layer is used to get the regression models for the fuel rate map. The resulting regression models are shown in Fig. A.6.

A.2.3 Open-loop Simulations

The accuracy of the regression models affects the simulation results in two ways:

1. The accuracy of the open-loop simulations with a given control input sequence,

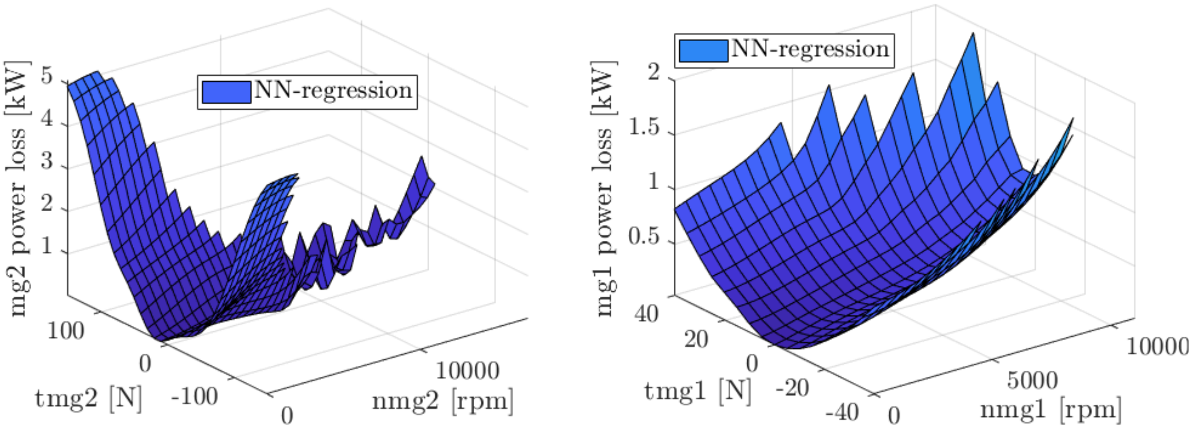


Figure A.5: NN regression models for MG2 and MG1 power loss maps.

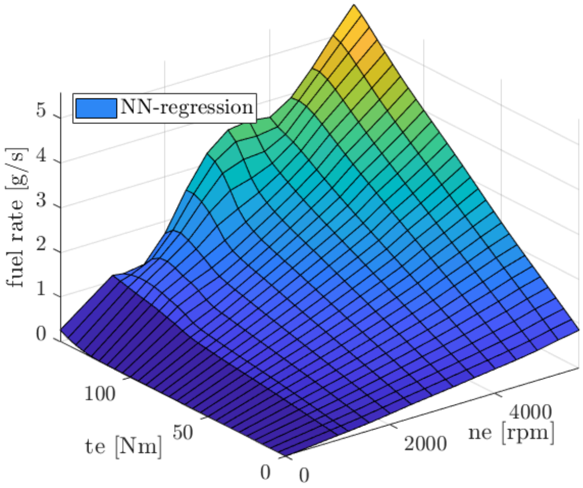


Figure A.6: NN regression model for the fuel rate map.

2. and the results of the Hamiltonian minimization in single shooting.

To test the performance of the regression models, the accuracy of the regression model is first compared in the open-loop simulations against the actual model with a given control input. The considered trip is long with wide SOC spans, which is suitable for testing the model accuracy over the entire range of SOC. The resulting time-domain responses of the SOC and fuel consumption with the two models are compared in Fig. A.7. The initial SOC-s are 0.85 in both cases, and the actual final SOC-s are 0.166 and 0.150 (10.7% difference) for the regression and the actual models, respectively. The total fuel consumption is 1285 and 1243 grams (3.4% difference) for the regression and the actual model. It can be seen from the full battery range open-loop simulation that the difference in SOC is rather large due to the cumulative error. The difference in the total fuel consumption is small, indicating the relatively good approximation accuracy of the neural network-based fuel rate map. Admit the existence of model inaccuracy, the overall trend is consistent, with higher terminal actual SOC and fuel consumption.

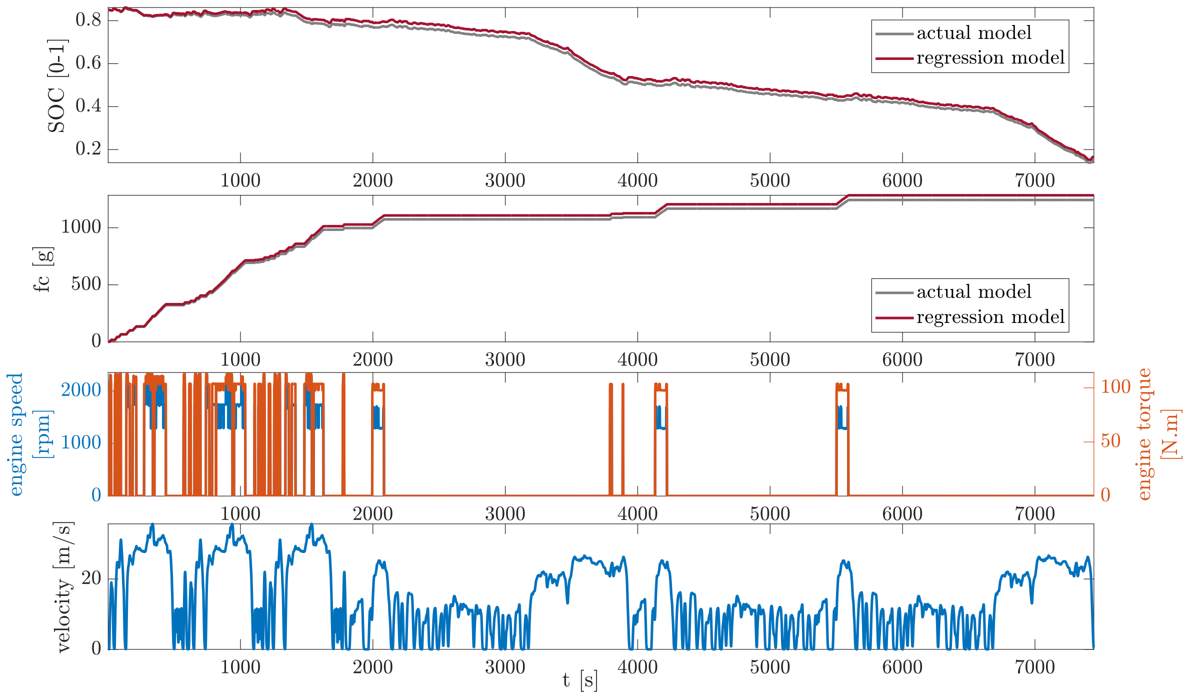


Figure A.7: Comparison of open-loop simulations with the actual model and the regression models.

A.3 Simulation Results with the Regression Models

Although it is crucial to demonstrate the accuracy of the regression models in open-loop simulations, the accuracy of the exact numbers is not the most critical quantity. The regression models are eventually used as the control-oriented model to select the control action, and different models will inevitably lead to potential discrepancies in optimized control actions. As pointed out in the previous section, the model discrepancies would affect the shape of the Hamiltonian and, in turn, the optimal control action. As a result, it is essential to investigate whether the overall trend, i.e., the trade-off between the fuel economy and drivability, the fuel economy benefit of the co-optimization upon the sequential optimization, will hold with the regression models. Besides, it is also important to observe whether the regression models (smoother than the actual maps) could lead to numerical difficulties in the instantaneous Hamiltonian minimization and the single shooting iteration.

A.3.1 Numerical Difficulties with the Regression Model

Inevitably, the regression models are smoother compared to the actual model. However, the smoothness will significantly impact the locations of the local minima of the Hamiltonian, which as a result, will affect the speed of convergence in the single shooting iterations. These observations will be detailed in the remaining section.

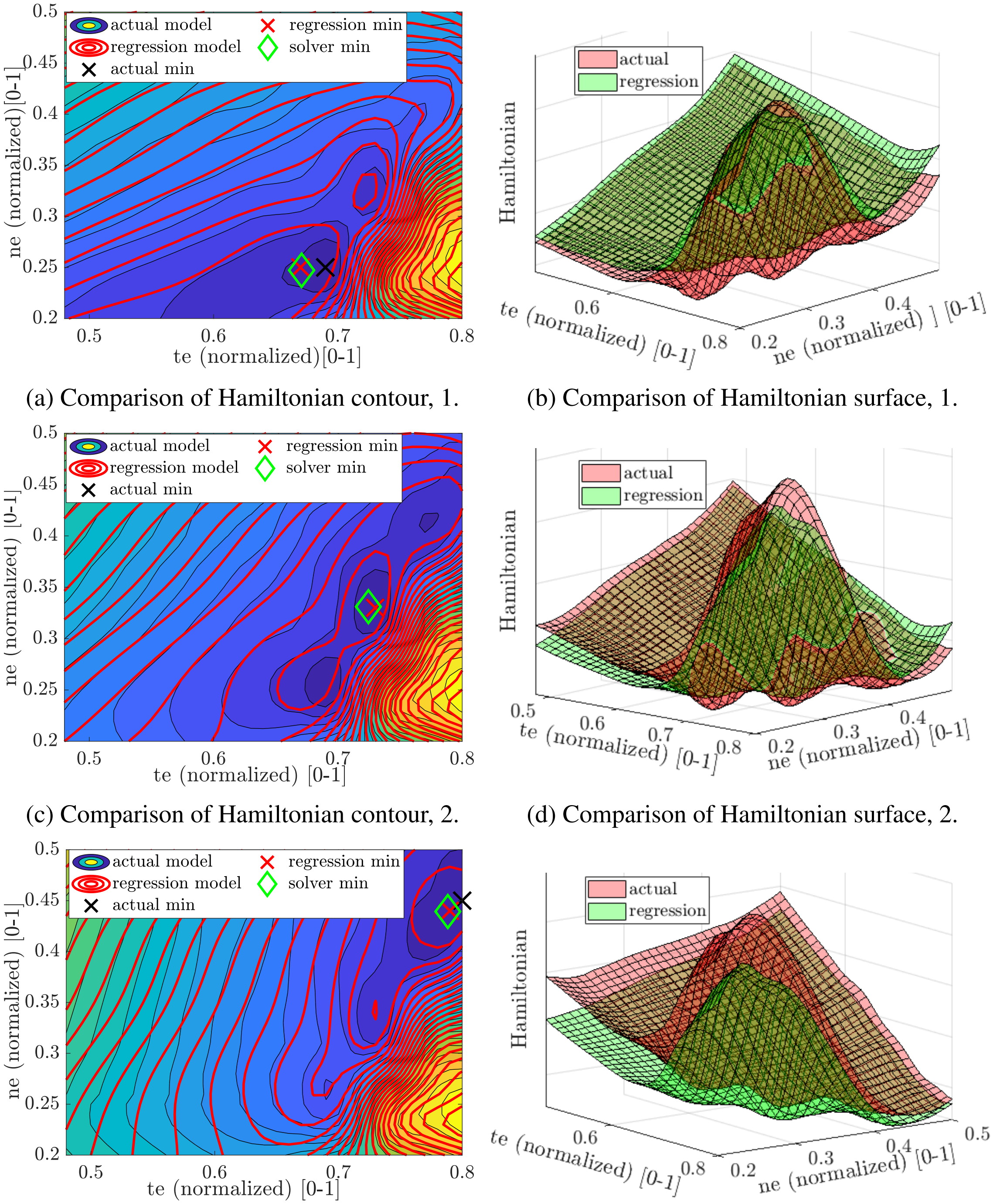
In a system open-loop evaluation level, it is enough to consider only the total discrepancy of the SOC and fuel consumption at the end of a trip to be evaluated. However, since the essential purpose of the control-oriented model is for numerical optimization, the regression model's inaccuracies would accumulate and end up significantly changing the shape of the Hamiltonian, which might lead to numerical difficulties and unexpected optimization results. For an arbitrary combination of regression models for battery, engine, and electric motors, it is not guaranteed that the power-split Hamiltonian shape would be close to that of the actual model. The power-split Hamiltonian resulting from the actual model is unique in that it has several relatively stationary local minima (deep valleys). However, based on my observation, the Hamiltonian obtained with regression models often has very shallow valleys, making it hard to warm-start a local Newton-type solver. Worse still, for some combinations of regression models, the local minima are non-stationary, making it hard to do effective warm-starts.

To illustrate the numerical difficulties in the instantaneous Hamiltonian minimization with regression models, Some illustrative examples of Hamiltonian sweep for a given set of state and co-state are presented, shown in Fig. A.8. The regression models used are obtained after different NN set-ups for electric motors and fuel rate approximations. A particular combination producing the closest local engine torque and speed minimizers to those with the actual maps is selected. As clearly seen from Fig. figs. A.8b, A.8d and A.8f, the local Hamiltonian valleys with regression models are much shallower compared to those with the actual model. As a result, the step size of the Newton step has to be adequately reduced.

The convergence of single shooting iterations with regression models is much slower than those with the actual model. The step size when choosing the vehicle level control needs to be much smaller. As a result, the number of single shooting iterations increases significantly for the control input r_k to be convergent.

A.3.2 One Cycle Comparison of Different Optimization Strategies with Regression Models

The time-domain response of the considered scenario is shown in Fig. A.9a. The initial battery SOC is $\mathrm{SOC}_0 = 0.85$, and the desired terminal SOC is $\mathrm{SOC}_f = 0.15$. The first subplot presents the battery SOC trajectories. The second subplot presents the cumulative



(a) Comparison of Hamiltonian contour, 1.

(b) Comparison of Hamiltonian surface, 1.

(c) Comparison of Hamiltonian contour, 2.

(d) Comparison of Hamiltonian surface, 2.

(e) Comparison of Hamiltonian surface, 3.

(f) Comparison of Hamiltonian surface, 3.

Figure A.8: Selective Hamiltonian comparisons.

fuel consumption trajectories (including cranking fuel). The third subplot presents cumulative cranking fuel consumption trajectories. The fourth subplot presents the resulting velocity trajectories. Note that in all the simulations, the actual terminal SOC is adequately close to the desired SOC terminal SOC SOC_f. To make Fig. A.9a readable, only selective simulations are presented. The full evaluation of percentage fuel consumption reduction with different optimization schemes is presented in Fig. A.10.

As clearly observed in Fig. A.9a, a 26% reduction in total fuel consumption can be achieved with co-optimization in the absence of additional drivability constraints. By comparison, the acceleration minimization followed by power-split optimization reaches slightly more than half of the fuel consumption reduction potential. All the remaining fuel consumption results with co-optimization with drivability constraints fall between the sequential optimization and the full co-optimization. Notably, the fuel economy benefits with co-optimization with an additional acceleration penalty where $Q = 5, 10$ degrades to a level similar to the sequential optimization (acceleration minimization followed by power-split optimization). It is also evident by the similarity of the time-domain responses between the two simulations. The fuel consumption results are consistent with those obtained with the actual model.

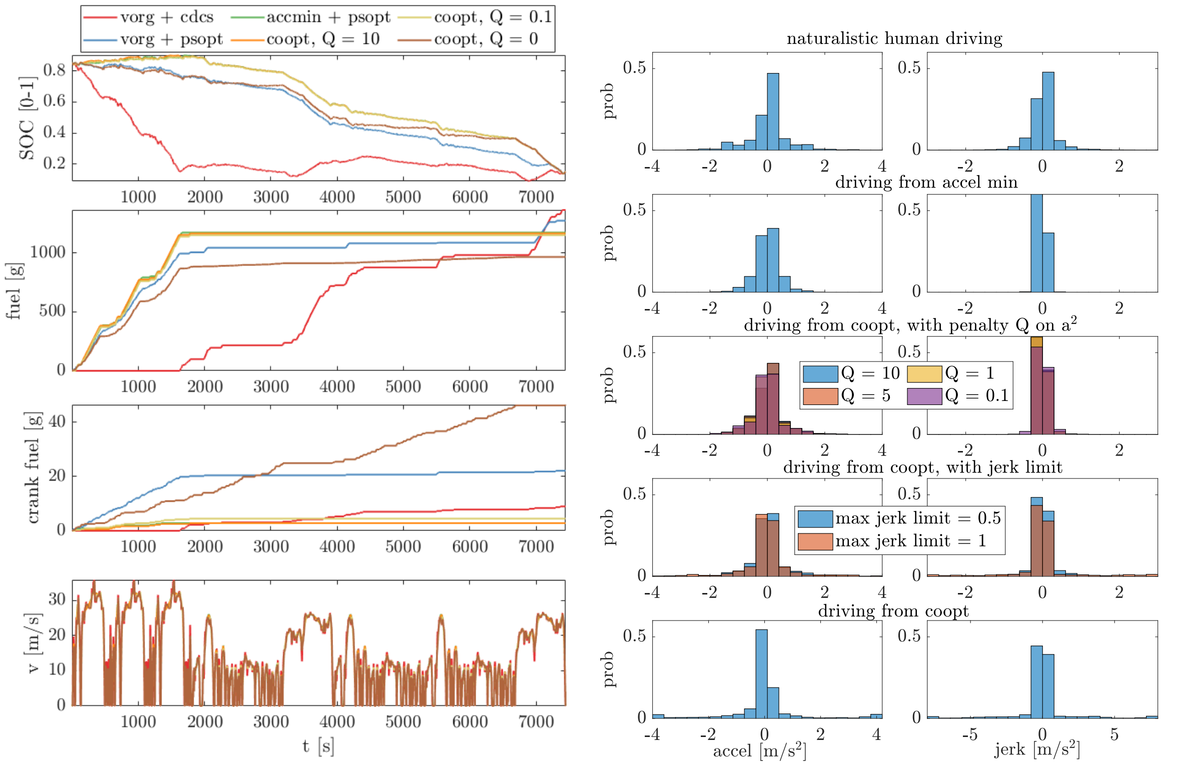
Similarly, it is also observed that the considerable potential in fuel economy with pure co-optimization is realized through frequent switches between the EV and HV modes. In particular, the co-optimization without additional drivability constraints consumes significantly more cranking fuel than any other method, as shown in the third subplot in Fig. A.10. As the requirement on the drivability becomes more stringent in co-optimization, the resulting velocity trace becomes smoother, which leads to a reduction in the total cranking fuel.

A.4 Summary

This chapter first revisits the control-oriented model for the SOC dynamics used for the control design, simulation, and analysis throughout this thesis by providing modeling details. First, each critical component is modeled separately. Afterward, several simplifications are made, leading to a control-oriented model used throughout this thesis.

Afterward, to make all the simulation results in this thesis reproducible, efforts are made in obtaining regression models to replace the actual models. First, polynomial regression is used to approximate the battery open circuit voltage (OCV) and internal resistance. Neural network models are used to approximate power losses of generator (MG1) and motor (MG2), and the fuel rate. The open-loop response with the regression models is compared with the actual model for a given control input trajectory. The regression models are then used to perform closed-loop simulations. The total fuel economy and drivability trends with the regression models are consistent with those with the actual models.

In trying to approximate static maps by smooth regression models, it is found that the smoothness induced by the regression models could cause difficulties in both the instantaneous



(a) Comparison of open-loop simulations with the actual model and the regression models.

(b) Comparison of open-loop simulations with the actual model and the regression models.

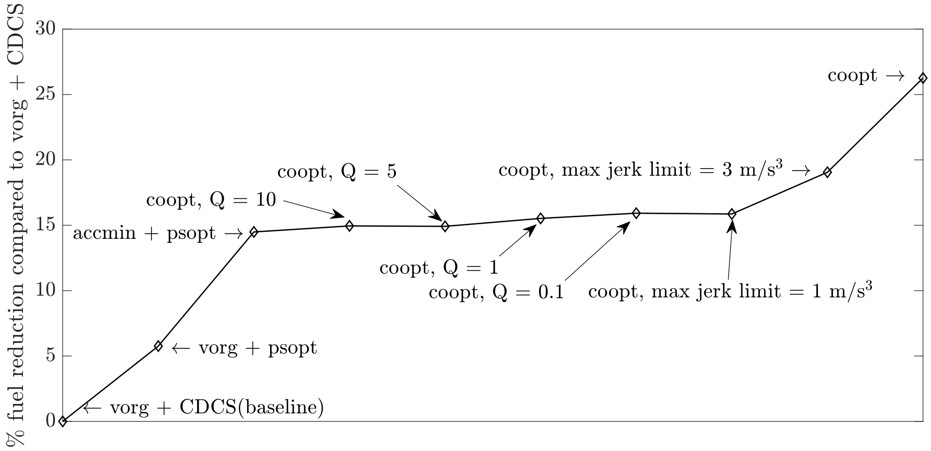


Figure A.10: Comparison of open-loop simulations with the actual model and the regression models.

Hamiltonian minimization and the convergence of the single shooting iterations. Therefore, it is of interest to investigate and quantify the contribution of the map smoothness on the total fuel economy and numerical difficulty as future work.

APPENDIX B

An H_∞ Approach for Velocity Optimization

Most of the existing work on eco-driving focuses on the design of MPC controllers for longitudinal vehicle dynamics (also the case when the sequential optimization discussed in Chapter 4 is converted into a receding horizon implementation), which is an optimization-based control strategy [112, 113, 114]. Essentially, the optimization based control of vehicle longitudinal dynamics typically requires

1. an optimization problem to be solved. In the acceleration minimization setting presented in the previous section, it requires to solve a QP,
2. the prediction of the lead vehicle's trajectory within the considered horizon, and
3. the discretization of the continuous-time dynamics, which necessitates balancing between the computational complexity and accuracy of the prediction model.

In the online (MPC) implementation, both the computation complexity and the prediction accuracy will affect the performance of the optimization-based controller. Inevitably, the lead vehicle's prediction as a disturbance can affect energy efficiency and constraint satisfaction.

Cooperative adaptive cruise control (CACC) as an extension of the adaptive cruise control (ACC) functionality, has shown its potential to significantly enhance road safety, improve highway utility, and increase traffic efficiency [115]. As CACC system is employed to ensure short inter-vehicle distances, disturbance amplifications in the upstream direction may occur. For this reason, disturbance attenuation, also known as string stability, is an essential issue for vehicle platoon control as well as the system stability [116]. It is well-known that robust control methodologies can effectively deal with modeling uncertainties and external disturbances [117]. As has been widely recognized, the linear consensus control and distributed robust control techniques enable insightful theoretical analysis and provide guarantees of string stability [99]. The requirements of string stability, robustness, and tracking performance are systematically measured by the H_∞ norm [115]. Although the scope of interest of this thesis is on the control of a *single* vehicle, the effectiveness of the H_∞ synthesis technique demonstrated in the vehicle platooning control motivates the development of purely linear feedback control for the eco-driving of a single vehicle in this section.

Most of the existing platooning control work employs H_∞ synthesis for robust and string stability [118, 116, 119, 120]. In this chapter, a H_∞ controller is developed for eco-driving in the presence of a lead vehicle with acceleration as the input. The controller is designed to balance the allowable following distance gaps and passenger comfort requirements. Different from the CACC design, the objective is not to track the desired time-headway accurately. Instead, a desired time-headway tracking objective is incorporated to replace the position constraints explicitly considered in the MPC controller. The lead vehicle's speed is viewed as a disturbance, with the controller designed to minimize its influence on the time-headway tracking, thus constraint violation. Meanwhile, the acceleration is considered as an indication of passenger comfort. The controller is designed to minimize the impacts from both the lead vehicle's speed as disturbance and the internal dynamics on acceleration to prompt driving smoothness. Although several works on (adaptive) cruise control [121, 122, 123, 124] adopted a H_∞ approach, their focuses are still on stability and tracking either a desired time-headway or a desired speed. The uniqueness of the work presented in this chapter is that it shifts the explicit position constraint requirement into an implicit consideration via the time-headway tracking objective. Besides, the quadratic acceleration cost in MPC problem is reformulated in the H_∞ design. It is considered as a design objective for the states and disturbance to impact the acceleration minimally.

B.1 Design of Different Velocity Following Controllers

B.1.1 An H_∞ Control Design

A linear state-feedback controller will be designed to track a desired time headway τ_s and smooth the velocity trajectory simultaneously. The dynamics of the vehicle-following subsystem are reformulated as

$$\dot{x} = Ax + B_u u + B_w w \tag{B.1a}$$

$$z_w = C_{x,w}(\tau_s)x + D_{u,w}u \tag{B.1b}$$

$$z_u = C_{x,u}x + D_{u,u}u \tag{B.1c}$$

where $x = [\Delta d \ v]^T = [s^l - s - \Delta d_0 \ v]^T$ are the distance gap between the lead and ego vehicle, the velocity of the ego vehicle, respectively. Δd_0 denotes the initial distance gap between the lead vehicle and the ego vehicle. s and s^l are the ego and lead vehicle's position, respectively. $u = a$ is the acceleration of the ego vehicle, and the disturbance $w = v^l$ is the lead vehicle's velocity. Two sets of outputs are considered to include disturbance-to-output and state-to-output properties, respectively. $z_w = [\Delta d - \tau_s v \ a]^T$ with $\tau_s \in [\tau_{\min}, \tau_{\max}]$ includes the desired time-headway tracking and the acceleration, and $z_u = a$ includes the acceleration. $A = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}$, $B_u = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $B_w = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $C_{x,w}(\tau_s) = \begin{bmatrix} \rho_1 & -\rho_1 \tau_s \\ 0 & 0 \end{bmatrix}$, and $D_{u,w} = \begin{bmatrix} 0 \\ \rho_2 \end{bmatrix}$, where ρ_1 and ρ_2 are tunable weights on time headway tracking or acceleration. $C_{x,u} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$, and $D_{u,u} = \begin{bmatrix} 0 \\ \rho_3 \end{bmatrix}$, where ρ_3 is a tunable weight on acceleration. ρ_1–ρ_3 are first chosen to be the inverse of the maximum distance gap and acceleration, and are then tuned such that no violation of (B.7e) and (B.7d) is observed on several test trips.

Remark 11. Note that although here the control-oriented model is assumed to be accurate, the model inaccuracy could be included in the H_∞ control design if, for example, the uncertainty terms can be represented by polytopic linear differential inclusions [117].

A linear state-feedback gain K is considered so as to minimize the impact of the exogenous disturbance w (the lead vehicle's velocity) on the output z_w including the time-headway tracking and the acceleration. A state-feedback gain K is pursued such that the L_2 gain

$$\sup_{||w||_2=1} ||z||_2 = \sup_{||w||_2 \neq 0} \frac{||z||_2}{||w||_2} \tag{B.2}$$

of the closed-loop system is less than a specified number γ. The L_2 gain for LTI systems is equal to the H_∞ norm of the corresponding transfer matrix. There exists a state-feedback gain K such that the L_2 gain of an LTI system is less than γ_1, if there exist K and $Q > 0$ such that,

$$\begin{bmatrix} (A + B_u K)Q + Q(A + B_u K)^T + B_w B_w^T & * \\ (C_{x,w}(\tau_s) + D_{u,w} K)Q & -\gamma_1^2 I \end{bmatrix} \leq 0 \tag{B.3}$$

In addition, the resulting acceleration a is also affected by the state x. For the driving smoothness objective, an additional linear matrix inequality (LMI) condition is considered

$$\begin{bmatrix} (A + B_u K)Q + Q(A + B_u K)^T & * \\ (C_{x,u} + D_{u,u} K)Q & -\gamma_2^2 I \end{bmatrix} \leq 0, \tag{B.4}$$

such that the output energy $\int_0^\infty z_u^T z_u \leq \gamma_2$.

With Δd_0 being the initial distance gap between the lead and the ego vehicle, the initial condition $x(0)$ can be assumed to be zero. Consequently, the acceleration constraints $||u|| \leq a_{\max}$ can be enforced through the following LMI

$$\begin{bmatrix} Q & Y^T \\ Y & a_{\max}^2 I \end{bmatrix} \geq 0 \tag{B.5}$$

The underlying optimization to be solved is

$$\begin{aligned} &\text{minimize } \gamma_1 + \gamma_2 \\ &\text{subject to } (\text{B.3}), (\text{B.4}), (\text{B.5}), \quad \forall \tau_s \in co\{\tau_{\min}, \tau_{\max}\}. \end{aligned} \tag{B.6}$$

B.1.2 Online Acceleration Minimization Formulation

The MPC problem to be solved at each time instant t, with a prediction horizon N_p is formulated as

$$\text{minimize} \sum_{k=0}^{N_p-1} a_{k|t}^2 \Delta t \tag{B.7a}$$

$$\text{subject to } x_{k+1|t} = A_d x_{k|t} + B_d a_{k|t}, \tag{B.7b}$$

the acceleration constraints

$$-a_{\max} \leq a_{k|t} \leq a_{\max} \tag{B.7c}$$

the minimum time-headway ($\tau_{\min}$) constraints

$$\tau_{\min} v_{k|t} + \Delta d_{\min} \leq s_{k|t}^l - s_{k|t} - \Delta d_0, \tag{B.7d}$$

where and the maximum time-headway ($\tau_{\max}$) constraints

$$\tau_{\min} v_{k|t} + \Delta d_{\max} \geq s_{k|t}^l - s_{k|t} - \Delta d_0. \tag{B.7e}$$

$\Delta d_{\min} \leq \Delta d_{\max}$ to guarantee a small margin during standstill. The vehicle dynamics (B.7b) in prediction is formulated in discrete-time, with $A_d = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, B_d = \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix}$. The velocity smoothness is the only cost as in (B.7a), whereas the time-headway requirements (B.7d),(B.7e) are enforced explicitly as constraints. The time-headway tracking requirement considered with H_∞ formulation is eliminated with the MPC formulation, allowing the controller to better exploit the allowable distance band to improve the driving smoothness.

Remark 12. Note that there is no terminal cost in the above MPC formulation (B.7a), as pointed out in [112], a MPC formulation without terminal penalty or constraints risks to be myopic in many situations. As investigated in [49], for a conventional internal combustion engine vehicle, a 20-sec prediction horizon is needed with perfect prediction in the absence of a terminal cost in the MPC formulation to achieve a fuel economy comparable to the optimal offline result. However, a sensational terminal cost is not intuitive in the absence of a quadratic term on the state. Although a terminal cost design was presented in [112] penalizing the terminal velocity and position, its design is rather heuristic. The optimality gap between the proposed terminal cost and the optimal form of the terminal cost is unclear. An advantage of the proposed H_∞ control design is that it provides the optimal cost-to-go matrix $P = Q^{-1}$ as a by-product when solving the LMI. It is therefore convenient to convert the H_∞ control design into an MPC formulation with the optimal cost-to-go $x^T Px$.

B.1.3 Discussion on the Design Difference

One immediate advantage with the H_∞ controller is that it does not require any information from vehicle-to-vehicle communication but from real-time onboard sensors, which is relatively standard on modern vehicles. However, the H_∞ control design necessities compromising the driving smoothness to the time-headway tracking, as position constraints (B.7d) and (B.7e) are only handled implicitly through a properly tuned weight ρ_1 on the time-headway tracking. Besides, particular realizations of the lead vehicle's trajectories are not accounted for in the H_∞ control design. Instead, the controller is designed to minimize the influence of the lead vehicle's velocity on the performance metric under *all* frequencies. Since the controller is designed to be only *reactive*, there will be inevitable degradation in driving smoothness compared to the pure acceleration minimization.

By comparison, the MPC design, with only the acceleration minimization objective, removes the conservativeness introduced by time-headway tracking. The advantage of MPC in handling state constraints allows the controller to maximize the utilization of the distance band to achieve extremely smooth driving. Moreover, with prediction, the controller acts *proactive*, thus eliminating the performance degradation with the H_∞ controller. However, several issues arise in the MPC controller, making it not necessarily a better approach than the H_∞ controller under all circumstances.

First, the MPC controller needs an optimization problem (B.7) to be solved online at each time instant due to the time-varying constraints (B.7e) and (B.7d). Albeit fast and reliable QP solvers [125], any cost increase due to the use of high-performance processors refrains the application of MPC in automotive systems except in very few cases. Second, as investigated in [49] the performance of the MPC controller is impacted by the prediction horizon. Admit that the performance improves with a larger prediction horizon; the computation cost might no longer be negligible. Moreover, (B.7) requires prediction of the lead vehicle's trajectory. A reasonable prediction accuracy necessitates vehicle-to-vehicle communication and is not available with only ADAS. With insufficient CAV penetration, it might not be possible to predict or hard to get a high-quality prediction. Prediction inaccuracy would make increasing N_p in (B.7) less beneficial. Moreover, as prediction is used in formulating the position constraints, prediction inaccuracy could lead to a constraint violation, even if (B.7) is solved reliably. The rear-end collision will be a severe consequence. Additional design is needed to cope with the constraint violation caused by prediction inaccuracy. The performance degradation caused by a short prediction length, prediction inaccuracy, and the extra efforts on constraint violation might offset the benefits with the MPC controller.

Since the ultimate goal of eco-driving is to improve energy or fuel efficiency, the comparison of different eco-driving controllers only becomes instructive when particular vehicle powertrains are included. The performance of H_∞ and MPC evaluated together with specific powertrains is the focus of the next section.

B.2 Simulation Results and Discussions

In this section, first, the performance of the H_∞ is compared with the MPC controller in terms of trip energy of the resulting trajectories. Afterward, different types of powertrains are included to evaluate the energy/fuel consumption. Specifically, vehicles with two types of powertrains are considered

1. pure EVs, which are single energy source vehicles,
2. and (P)HEVs, where both the engine and electric motors are used and thus have two energy sources. The initial and terminal SOC are required to be the same in all simulations.

The torque demand to be satisfied by the powertrain is calculated with (4.5) from the resulting velocity and acceleration trajectories. Three prediction methods are considered in the MPC implementation

1. constant velocity prediction: The lead vehicle's velocity is assumed to be its current value v_t^l and held constant within the prediction horizon.
2. constant acceleration prediction: The lead vehicle's acceleration is assumed to be its current value a_t^l and held constant within the prediction horizon. Since a constant non-

zero acceleration could lead to a negative or very large positive velocity, the resulting speed is saturated to zero or a maximum allowable value.

3. perfect velocity prediction: The lead vehicle's velocity within the prediction horizon is assumed to be its exact velocity trace in the next N_p-step. This corresponds to the idealized case with V2V communication technique.

The same Ann Arbor trip set is considered to generalize the performance comparison. The lead vehicle is assumed to drive exactly these trips, whereas the ego vehicle's driving traces are obtained either with H_∞ or MPC controller. The horizon length is chosen to be 10-sec to match those enabled by the existing velocity prediction algorithms [126, 2].

B.2.1 Trip Energy Comparison

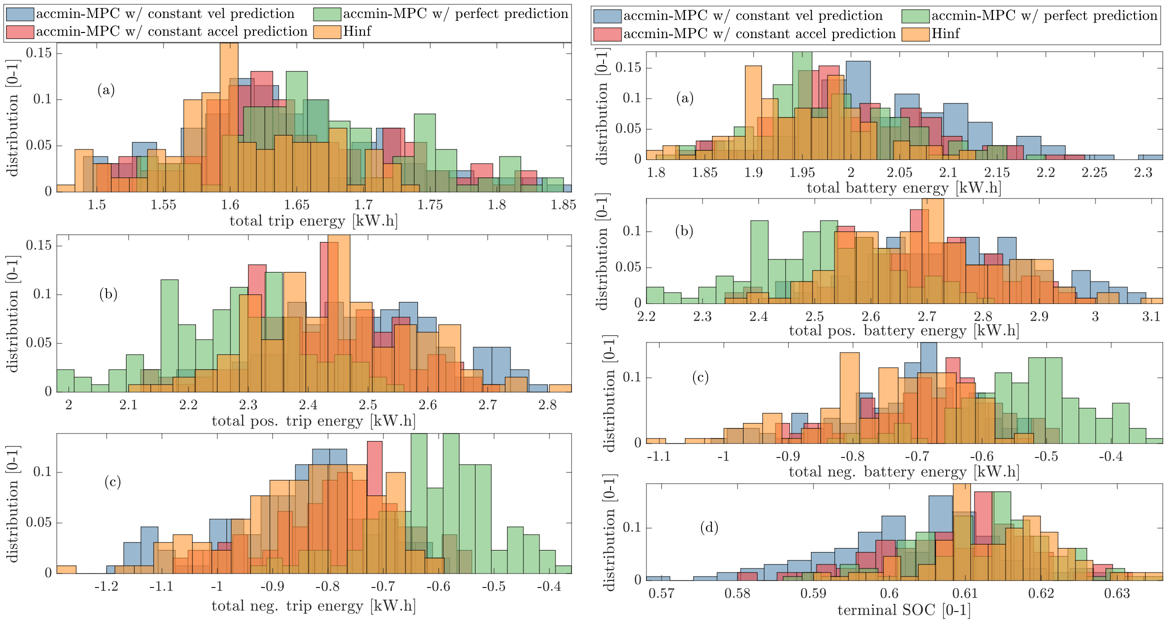
Figure B.1a quantifies the trip energy distributions resulting from the H_∞ and MPC controller prior to including any specific powertrain. The average total trip energies are 1.64, 1.65, 1.67, and 1.61 kWh, the average positive trip energies are 2.50, 2.43, 2.28, and 2.45 kWh , the average negative trip energies are -0.86, -0.88, -0.61, and -0.84 kWh for accmin-MPC with constant velocity, constant acceleration and perfect prediction, and H_∞ controller, respectively.

From Fig. B.1a(b) and (c), it can be seen that the accmin-MPC, under ideal prediction, significantly reduces both positive and negative trip energy as expected, due to its purely velocity smoothing objective. By comparison, both the positive and negative trip energy increases with the H_∞ controller, due to its time-headway tracking requirement. An important observation is that smoothing the velocity by acceleration minimization is not equivalent to the minimization of the total trip energy. Although the accmin-MPC with perfect prediction smooths the driving trace most significantly, its total trip energy is not statistically smaller compared the results with H_∞ controller, as evident from the Fig. B.1a(a). How the difference in the trip energy could affect the total energy or fuel efficiency is powertrain-dependent, and is the focus of the next section.

B.2.2 Energy/Fuel Consumption with Pure Electric and Hybrid Powetrains

Since the purpose of the eco-driving controller design via velocity smoothing is to improve the energy or fuel efficiency, in this section, two specific electrified powertrains[1] are considered

[1] The considered powertrain is the same as used in this thesis, which can either run in a pure EV mode with only electric motors or a HV mode, with both engine and electric motors.



(a) Statistical comparison of trip energies among different control strategies: (a) total trip energy, (b) positive trip energy, and (c) negative trip energy.

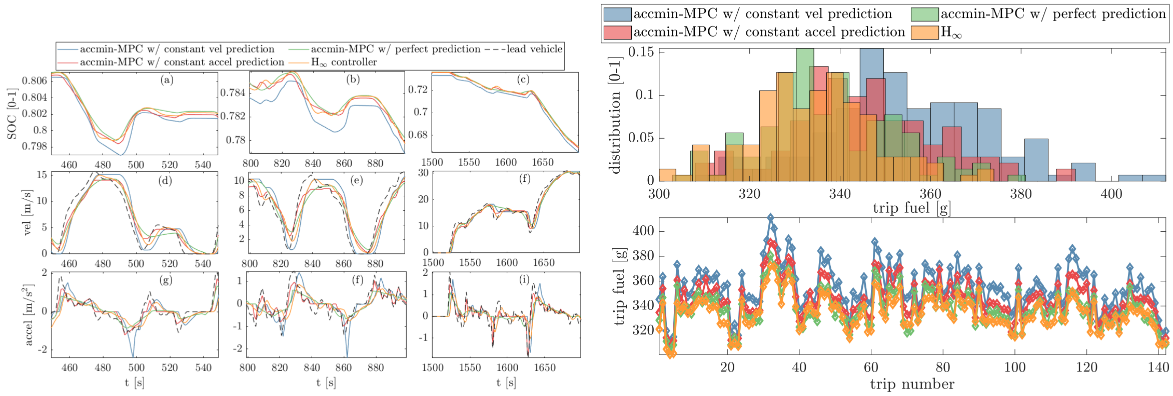
(b) Statistical comparison of resulting battery energies among different control strategies: (a) total battery energy, (b) total battery discharge energy, (c) total battery charge energy, and (d) the actual terminal SOC at the end of the trip.

Figure B.1: Comparison of resulting vehicle and battery energy for several trips.

to evaluate the driving traces obtained from the two different eco-driving controllers. To better compare the results, the energy/fuel efficiency is evaluated in an offline manner. The powertrain-level controller is not running together with the eco-driving controller but rather in an ad hoc manner, assuming the entire eco-driving trace is known.

First, the powertrain if forced to operate in a pure EV mode, effectively making the vehicle an EV. The total electrical energy is thus the performance metric. The initial SOC in all EV simulations are set to 0.85. The average total battery energies are 2.04, 2.00, 1.98, and 1.95 kWh, the average positive battery energies (discharge) are 2.76, 2.68, 2.51, and 2.70 kWh, the average negative battery energies (charge) are -0.72, -0.68, -0.54, and -0.75 kWh, and the average terminal SOC are 0.603, 0.609, 0.612, and 0.614 for accmin-MPC with constant velocity, constant acceleration and perfect prediction, and H_∞ controller, respectively. A higher terminal SOC is an indication of a lower total battery usage. Through slightly compromising passenger comfort, the total battery usage reduces with the H_∞ controller compared to the ideal case accmin-MPC under the considered prediction horizon.

An illustration of the time-domain responses over different periods with different controllers on a particular trip is shown in Fig. B.2a. As observed from the acceleration and velocity comparisons, the accmin-MPC with perfect prediction achieves the smoothest driving as expected. Under ideal forecast, it maximizes the allowable distance gap to generate an extremely smooth trajectory. The H_∞ controller achieves the second smoothest driving, as the position constraints are handled implicitly with the time-headway tracking objective, thus compromising the driving smoothness objective. Except for the constant velocity prediction case, all the results are shown to attenuate the lead vehicle's acceleration (smoother driving), as seen from Fig. B.2a(g)–(i). The accmin-MPC with constant velocity prediction is the most aggressive because the controller could force the ego vehicle to converge to the lead vehicle's current speed within the prediction horizon.



(a) Performance comparison among different control strategies over different periods: (a)–(c) battery SOC, (d)–(f) velocity, and (g)–(i) acceleration

(b) Statistical comparison of the resulting fuel consumption among different control strategies: (a) total fuel consumption distribution, and (b) trip-by-trip fuel consumption.

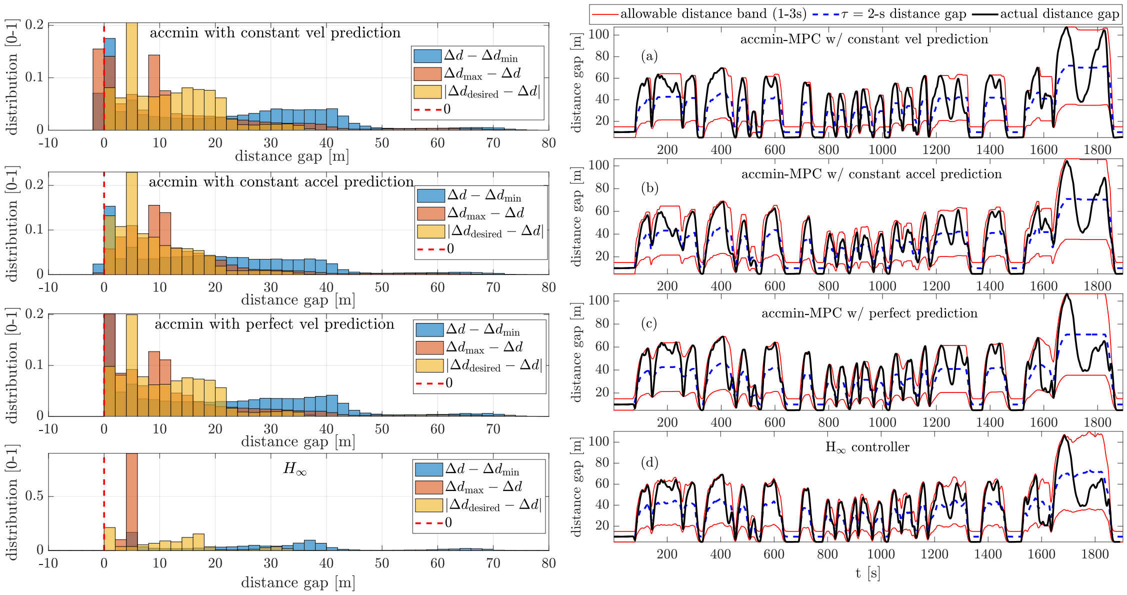
Figure B.2: Comparison of time-domain responses and fuel consumption with a hybrid powertrain.

Next, the fuel efficiency of a PHEV/HEV with power-split optimization is compared, representing the best achievable fuel efficiency. The power-split optimization is performed with the method described in [3, 31]. The initial SOC is 0.4, and the terminal SOC is 0.35.

The average total fuel consumption are 355, 345, 337 g for accmin-MPC with constant velocity, constant acceleration, perfect prediction, and 334 g for H_∞ controller, respectively. It can be seen that the best achievable performance in fuel economy for a (P)HEV with a H_∞ eco-driving controller is as good as what can be achieved with an accmin-MPC controller with a 10-sec perfect prediction.

B.2.3 Distance Gap Constraints

Figure B.3a compares the vehicle-following behaviors resulting from the H_∞ and MPC controller. Δd is the actual distance gap between the ego and its immediate preceding vehicle.



(a) Statistical comparison of distance gaps resulting different control strategies.

(b) Comparison of distance gap from the lead vehicles:(a) MPC w/ constant velocity prediction, (b) MPC w/ constant acceleration prediction, (c) MPC with perfect prediction, and (d) the proposed H_∞ controller

Figure B.3: Comparison of resulting distance gaps.

Δd_{min} denotes the minimum allowable distance gap (1-sec time-headway) for avoiding rear-end collisions. Δd_{max} is the maximum allowable distance gap (3-sec time-headway) for avoiding cut-ins from adjacent lanes. $\Delta d_{\text{desired}}$ represents the desired distance gap (2-sec time-headway). The remaining margins to the minimum and maximum allowable distance gaps and the absolute gaps to the desired time-headway are presented.

No constraint violation is observed when the accmin-MPC is applied with a perfectly predicted lead vehicle's trajectory. However, in the presence of prediction inaccuracy, accmin-MPC could lead to violations of either maximum or minimum distance gap requirement, as evident by Fig. B.3a(a) and (b). By comparison, the inherent time-headway tracking requirement guarantees the constraint satisfaction on all the considered trips.

An illustration of the distance gap behaviors in time-domain with different controllers on a particular trip is shown in Fig. B.3b. Constraint violations are observed in the MPC controller with constant velocity prediction, as seen from Fig. B.3b(a). An important observation is that the accmin-MPC, with perfect prediction, maximizes the utilization of the allowable distance band, as shown in Fig. B.3b(c). The margins often are enlarged when the MPC is performed with constant acceleration prediction, as seen from Fig. B.3b(b). By comparison, there are always small distance margins left with the H_∞ controller, as shown in Fig. B.3b(d).

BIBLIOGRAPHY

[1] McKinsey&Company, "Expanding electric-vehicle adoption despite early growing pains," Accessed November 2, 2020, https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/expanding-electric-vehicle-adoption-despite-early-growing-pains.

[2] Hyeon, E., Kim, Y., Prakash, N., and Stefanopoulou, A. G., "Short-term Speed Forecasting Using Vehicle Wireless Communications," *2019 American Control Conference (ACC)*, IEEE, 2019, pp. 736–741.

[3] Huang, M., Zhang, S., and Shibaike, Y., "Real-time Long Horizon Model Predictive Control of a Plug-in Hybrid Vehicle Power-Split Utilizing Trip Preview," Tech. rep., SAE Technical Paper, 2019.

[4] Ichikawa, S., Takeuchi, H., Fukuda, S., Kinomura, S., Tomita, Y., Suzuki, Y., and Hirasawa, T., "Development of new plug-in hybrid system for compact-class vehicle," *SAE International Journal of Alternative Powertrains*, Vol. 6, No. 1, 2017, pp. 95–102.

[5] Alamir, M. and Attia, S.-A., "On solving optimal control problems for switched hybrid nonlinear systems by strong variations algorithms," *Proceedings of 6th IFAC symposium on nonlinear control systems*, 2004, pp. 558–563.

[6] United States Environmental Protection Agency (EPA), "Total U.S. greenhouse gas emissions by economic sector in 2015," Accessed September 11, 2020, https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions.

[7] United States Environmental Protection Agency (EPA), "Federal Vehicle Standards," Accessed October 31, 2020, https://www.c2es.org/content/regulating-transportation-sector-carbon-emissions/.

[8] Elgowainy, A., Han, J., Ward, J., Joseck, F., Gohlke, D., Lindauer, A., Ramsden, T., Biddy, M., Alexander, M., Barnhart, S., et al., "Cradle-to-grave lifecycle analysis of US light duty vehicle-fuel pathways: a greenhouse gas emissions and economic assessment of current (2015) and future (2025-2030) technologies," Tech. rep., Argonne National Lab.(ANL), Argonne, IL (United States), 2016.

[9] McKinsey&Company, "The global electric-vehicle market is amped up and on the rise," Accessed November 2, 2020, https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-global-electric-vehicle-market-is-amped-up-and-on-the-rise.

[10] McKinsey&Company, "McKinsey Electric Vehicle Index: Europe cushions a global plunge in EV sales," Accessed November 2, 2020, `https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/mckinsey-electric-vehicle-index-europe-cushions-a-global-plunge-in-ev-sales`.

[11] United States Environmental Protection Agency (EPA), "Plug-in Hybrid Vehicles," Accessed September 11, 2020, `https://www.fueleconomy.gov/feg/phevtech.shtml`.

[12] McGuckin, N. and Fucci, A., "Summary of travel trends: 2017 national household travel survey," *Washington, DC: Federal Highway Administration*, 2018.

[13] Fridén, H. and Sahlin, H., "Energy management strategies for plug-in hybrid electric vehicles," 2012.

[14] Sun, C. et al., "Dynamic Traffic Feedback Data Enabled Energy Management in Plug-in Hybrid Electric Vehicles," *IEEE Transactions on Control Systems Technology*, Vol. 23, No. 3, May 2015, pp. 1075–1086.

[15] Vahidi, A. and Sciarretta, A., "Energy saving potentials of connected and automated vehicles," *Transportation Research Part C: Emerging Technologies*, Vol. 95, 2018, pp. 822–843.

[16] Sciarretta, A. and Vahidi, A., "Energy-Efficient Speed Profiles (Eco-Driving)," *Energy-Efficient Driving of Road Vehicles*, Springer, 2020, pp. 131–178.

[17] Bae, S., Choi, Y., Kim, Y., Guanetti, J., Borrelli, F., and Moura, S., "Real-time Ecological Velocity Planning for Plug-in Hybrid Vehicles with Partial Communication to Traffic Lights," *arXiv e-prints*, Mar 2019, pp. arXiv:1903.08784.

[18] Ngo, D. V., Hofman, T., Steinbuch, M., and Serrarens, A. F., "An optimal control-based algorithm for hybrid electric vehicle using preview route information," *Proceedings of the 2010 American Control Conference*, IEEE, 2010, pp. 5818–5823.

[19] Sciarretta, A., De Nunzio, G., and Ojeda, L. L., "Optimal ecodriving control: Energy-efficient driving of road vehicles as an optimal control problem," *IEEE Control Systems Magazine*, Vol. 35, No. 5, 2015, pp. 71–90.

[20] Padilla, G., Weiland, S., and Donkers, M., "A Global Optimal Solution to the Eco-Driving Problem," *IEEE control systems letters*, Vol. 2, No. 4, 2018, pp. 599–604.

[21] Heppeler, G., Sonntag, M., Wohlhaupter, U., and Sawodny, O., "Predictive planning of optimal velocity and state of charge trajectories for hybrid electric vehicles," *Control Engineering Practice*, Vol. 61, 2017, pp. 229–243.

[22] Martinez, C. M., Hu, X., Cao, D., Velenis, E., Gao, B., and Wellers, M., "Energy Management in Plug-in Hybrid Electric Vehicles: Recent Progress and a Connected Vehicles Perspective," *IEEE Transactions on Vehicular Technology*, Vol. 66, No. 6, June 2017, pp. 4534–4549.

[23] Yan, F., Wang, J., and Huang, K., "Hybrid electric vehicle model predictive control torque-split strategy incorporating engine transient characteristics," *IEEE transactions on vehicular technology*, Vol. 61, No. 6, 2012, pp. 2458–2467.

[24] Di Cairano, S., Liang, W., Kolmanovsky, I. V., Kuang, M. L., and Phillips, A. M., "Power smoothing energy management and its application to a series hybrid powertrain," *IEEE Transactions on control systems technology*, Vol. 21, No. 6, 2012, pp. 2091–2103.

[25] Nüesch, T., Elbert, P., Flankl, M., Onder, C., and Guzzella, L., "Convex optimization for the energy management of hybrid electric vehicles considering engine start and gearshift costs," *Energies*, Vol. 7, No. 2, 2014, pp. 834–856.

[26] Chen, D., Kim, Y., Huang, M., and Stefanopoulou, A., "An Iterative and Hierarchical Approach to Co-optimizing the Velocity Profile and Power-split of Plug-in Hybrid Electric Vehicles," *2020 American Control Conference (ACC)*, IEEE, 2020, pp. 3059–3064.

[27] Sager, S., *Numerical methods for mixed-integer optimal control problems*, Der Andere Verlag Tönning, 2005.

[28] Betts, J. T., "Survey of numerical methods for trajectory optimization," *Journal of guidance, control, and dynamics*, Vol. 21, No. 2, 1998, pp. 193–207.

[29] Bryson, A. E., *Applied optimal control: optimization, estimation and control*, CRC Press, 1975.

[30] Chen, D., Kim, Y., and Stefanopoulou, A. G., "State of Charge Node Planning with Segmented Traffic Information," *2018 Annual American Control Conference (ACC)*, IEEE, 2018, pp. 4969–4974.

[31] Chen, D., Kim, Y., and Stefanopoulou, A., "Predictive Equivalent Consumption Minimization Strategy with Segmented Traffic Information," *IEEE Transactions on Vehicular Technology*, 2020.

[32] Chen, D., Prakash, N., Stefanopoulou, A. G., Huang, M., Kim, Y., and Hotz, S. R., "Sequential Optimization of Velocity and Charge Depletion in a Plug-in Hybrid Electric Vehicle," *14th International Symposium on Advanced Vehicle Control*, 2018.

[33] Chen, D., Huang, M., Stefanopoulou, A. G., and Kim, Y., "A Receding-Horizon Framework for Co-Optimizing the Velocity and Power-Split of Automated Plug-In Hybrid Electric Vehicles," *ASME Letters in Dynamic Systems and Control*, 2021, pp. 1–7.

[34] Katsargyri, G. E. et al., "Optimally controlling Hybrid Electric Vehicles using path forecasting," *2009 American Control Conference*, June 2009, pp. 4613–4617.

[35] Yoon, J. et al., "Surface street traffic estimation," *Proceedings of the 5th international conference on Mobile systems, applications and services*, ACM, 2007, pp. 220–232.

[36] Sundstrom, O. and Guzzella, L., "A generic dynamic programming Matlab function," *2009 IEEE Control Applications, (CCA) Intelligent Control, (ISIC)*, 2009, pp. 1625–1630.

[37] Mangasarian, O. L., "Sufficient conditions for the optimal control of nonlinear systems," *SIAM Journal on control*, Vol. 4, No. 1, 1966, pp. 139–152.

[38] Seierstad, A. and Sydsaeter, K., "Sufficient conditions in optimal control theory," *International Economic Review*, 1977, pp. 367–391.

[39] Serrao, L., Onori, S., and Rizzoni, G., "A comparative analysis of energy management strategies for hybrid electric vehicles," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 133, No. 3, 2011, pp. 031012.

[40] Onori, S., Serrao, L., and Rizzoni, G., *Hybrid electric vehicles: energy management strategies*, Vol. 13, Springer, 2016.

[41] Serrao, L., Onori, S., and Rizzoni, G., "ECMS as a realization of Pontryagin's minimum principle for HEV control," *2009 American control conference*, IEEE, 2009, pp. 3964–3969.

[42] Hu, J., Shao, Y., Sun, Z., Wang, M., Bared, J., and Huang, P., "Integrated optimal eco-driving on rolling terrain for hybrid electric vehicle with vehicle-infrastructure communication," *Transportation Research Part C: Emerging Technologies*, Vol. 68, 2016, pp. 228–244.

[43] Zulkefli, M. A. M., Zheng, J., Sun, Z., and Liu, H. X., "Hybrid powertrain optimization with trajectory prediction based on inter-vehicle-communication and vehicle-infrastructure-integration," *Transportation Research Part C: Emerging Technologies*, Vol. 45, 2014, pp. 41–63.

[44] Johannesson, L., Murgovski, N., Jonasson, E., Hellgren, J., and Egardt, B., "Predictive energy management of hybrid long-haul trucks," *Control Engineering Practice*, Vol. 41, 2015, pp. 83–97.

[45] Kim, T. S., Manzie, C., and Sharma, R., "Two-stage optimal control of a parallel hybrid vehicle with traffic preview," *IFAC Proceedings Volumes*, Vol. 44, No. 1, 2011, pp. 2115–2120.

[46] Tomiyama, K., "Two-stage optimal control problems and optimality conditions," *Journal of Economic Dynamics and Control*, Vol. 9, No. 3, 1985, pp. 317–337.

[47] Liao-McPherson, D., Huang, M., and Kolmanovsky, I., "A Regularized and Smoothed Fischer-Burmeister Method for Quadratic Programming with Applications to Model Predictive Control," *IEEE Transactions on Automatic Control*, 2018.

[48] NHTSA, "Summary of State Speed Laws, 11 ed," 2010.

[49] Prakash, N., Cimini, G., Stefanopoulou, A. G., and Brusstar, M. J., "Assessing Fuel Economy From Automated Driving: Influence of Preview and Velocity Constraints," *ASME 2016 Dynamic Systems and Control Conference*, American Society of Mechanical Engineers, 2016, pp. V002T19A001–V002T19A001.

[50] Van Arem, B., Van Driel, C. J., and Visser, R., "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 7, No. 4, 2006, pp. 429–436.

[51] Zhu, J., Ngo, C., and Sciarretta, A., "Real-time optimal eco-driving for hybrid-electric vehicles," *IFAC-PapersOnLine*, Vol. 52, No. 5, 2019, pp. 562–567.

[52] Ozatay, E., Onori, S., Wollaeger, J., Ozguner, U., Rizzoni, G., Filev, D., Michelini, J., and Di Cairano, S., "Cloud-based velocity profile optimization for everyday driving: A dynamic-programming-based solution," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 15, No. 6, 2014, pp. 2491–2505.

[53] Hartl, R. F., Sethi, S. P., and Vickson, R. G., "A survey of the maximum principles for optimal control problems with state constraints," *SIAM review*, Vol. 37, No. 2, 1995, pp. 181–218.

[54] Chen, D., Kim, Y., and Stefanopoulou, A. G., "State of Charge Node Planning with Segmented Traffic Information," *2018 Annual American Control Conference (ACC)*, June 2018, pp. 4969–4974.

[55] Bock, H. G. and Plitt, K.-J., "A multiple shooting algorithm for direct solution of optimal control problems," *IFAC Proceedings Volumes*, Vol. 17, No. 2, 1984, pp. 1603–1608.

[56] Nikoobin, A. and Moradi, M., "Indirect solution of optimal control problems with state variable inequality constraints: finite difference approximation," *Robotica*, Vol. 35, No. 1, 2017, pp. 50–72.

[57] "IBM ILOG CPLEX Optimization Studio," `www.cplex.com`.

[58] Bliek1ú, C., Bonami, P., and Lodi, A., "Solving mixed-integer quadratic programming problems with IBM-CPLEX: a progress report," *Proceedings of the twenty-sixth RAMP symposium*, 2014, pp. 16–17.

[59] Gurobi Optimization, L., "Gurobi Optimizer Reference Manual," 2020.

[60] Kirches, C., *Fast numerical methods for mixed-integer nonlinear model-predictive control*, Springer, 2011.

[61] Kirches, C., Bock, H. G., Schlöder, J. P., and Sager, S., "Mixed-integer NMPC for predictive cruise control of heavy-duty trucks," *2013 European Control Conference (ECC)*, IEEE, 2013, pp. 4118–4123.

[62] Nocedal, J. and Wright, S., *Numerical optimization*, Springer Science & Business Media, 2006.

[63] Bazaraa, M. S., Sherali, H. D., and Shetty, C. M., *Nonlinear programming: theory and algorithms*, John Wiley & Sons, 2013.

[64] Halkin, H., "A maximum principle of the Pontryagin type for systems described by nonlinear difference equations," *SIAM Journal on control*, Vol. 4, No. 1, 1966, pp. 90–111.

[65] Goodwin, G., Seron, M. M., and De Doná, J. A., *Constrained control and estimation: an optimisation approach*, Springer Science & Business Media, 2006.

[66] Holtzman, J., "Convexity and the maximum principle for discrete systems," *IEEE Transactions on Automatic Control*, Vol. 11, No. 1, 1966, pp. 30–35.

[67] Holtzman, J., "On the maximum priciple for nonlinear discrete-time systems," *IEEE Transactions on Automatic Control*, Vol. 11, No. 2, 1966, pp. 273–274.

[68] Pearson, J. and Sridhar, R., "A discrete optimal control problem," *IEEE Transactions on automatic control*, Vol. 11, No. 2, 1966, pp. 171–174.

[69] Huang, M., *Low Complexity Model Predictive Control of a Diesel Engine Airpath.*, Ph.D. thesis, 2016.

[70] Tamminen, E., "Sufficient conditions for the existence of multipliers and Lagrangian duality in abstract optimization problems," *Journal of optimization theory and applications*, Vol. 82, No. 1, 1994, pp. 93–104.

[71] Tamminen, E. V., "Strong Lagrange duality and the maximum principle for nonlinear discrete time optimal control problems." *ESAIM: Control, Optimisation & Calculus of Variations*, Vol. 25, 2019.

[72] Boyd, S., Boyd, S. P., and Vandenberghe, L., *Convex optimization*, Cambridge university press, 2004.

[73] Sion, M. et al., "On general minimax theorems." *Pacific Journal of mathematics*, Vol. 8, No. 1, 1958, pp. 171–176.

[74] Fan, K., "Minimax theorems," *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 39, No. 1, 1953, pp. 42.

[75] Encyclopedia of Mathematics, "Positive cone," Accessed June 10, 2021, `https://encyclopediaofmath.org/wiki/Positive_cone`.

[76] Luenberger, D. G., *Optimization by vector space methods*, John Wiley & Sons, 1997.

[77] Smith, A. E. and Coit, D. W., "Penalty functions," *Handbook of evolutionary computation*, Vol. 97, No. 1, 1997, pp. C5.

[78] Clarke, F. H., *Optimization and nonsmooth analysis*, SIAM, 1990.

[79] Izmailov, A. F. and Solodov, M. V., *Newton-type methods for optimization and variational problems*, Springer, 2014.

[80] Rudin, W. et al., *Principles of mathematical analysis*, Vol. 3, McGraw-hill New York, 1964.

[81] Diehl, M., Bock, H. G., and Schlöder, J. P., "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM Journal on control and optimization*, Vol. 43, No. 5, 2005, pp. 1714–1736.

[82] Heywood, J. B., "Combustion engine fundamentals," *1^a Edição. Estados Unidos*, 1988.

[83] Huang, M., Nakada, H., Butts, K., and Kolmanovsky, I., "Nonlinear model predictive control of a diesel engine air path: A comparison of constraint handling and computational strategies," *IFAC-PapersOnLine*, Vol. 48, No. 23, 2015, pp. 372–379.

[84] Butts, K., Dontchev, A., Huang, M., and Kolmanovsky, I., "A perturbed chord (Newton-Kantorovich) method for constrained nonlinear model predictive control," *IFAC-PapersOnLine*, Vol. 49, No. 18, 2016, pp. 253–258.

[85] Chen, D., Huang, M., and Stefanopoulou, A. G., "Discrete Mixed-Integer Shooting (DMIS): Algorithm and Application to PHEV Energy Management Accounting for Fuel Cranking," submitted to IEEE Transactions on Control System Technology.

[86] Bae, S., Choi, Y., Kim, Y., Guanetti, J., Borrelli, F., and Moura, S., "Real-time Ecological Velocity Planning for Plug-in Hybrid Vehicles with Partial Communication to Traffic Lights," *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 1279–1285.

[87] Maamria, D., Gillet, K., Colin, G., Chamaillard, Y., and Nouillant, C., "Computation of eco-driving cycles for Hybrid Electric Vehicles: Comparative analysis," *Control Engineering Practice*, Vol. 71, 2018, pp. 44–52.

[88] Amini, M. R., Kolmanovsky, I., and Sun, J., "Hierarchical MPC for Robust Eco-Cooling of Connected and Automated Vehicles and Its Application to Electric Vehicle Battery Thermal Management," *IEEE Transactions on Control Systems Technology*, 2020, pp. 1–13.

[89] Hu, Q., Amini, M. R., Feng, Y., Yang, Z., Wang, H., Kolmanovsky, I., Sun, J., Wiese, A., Qiu, Z., and Buckland, J., "Engine and Aftertreatment Co-Optimization of Connected HEVs via Multi-Range Vehicle Speed Planning and Prediction," Tech. rep., SAE Technical Paper, 2020.

[90] Hu, Q., Amini, M. R., Wang, H., Kolmanovsky, I., and Sun, J., "Integrated Power and Thermal Management of Connected HEVs via Multi-Horizon MPC*," *2020 American Control Conference (ACC)*, 2020, pp. 3053–3058.

[91] Bellman, R. E. and Dreyfus, S. E., *Applied dynamic programming*, Princeton university press, 2015.

[92] Bertsekas, D. P., *Reinforcement learning and optimal control*, Athena Scientific Belmont, MA, 2019.

[93] Faulwasser, T. and Zanon, M., "Primal or Dual Terminal Constraints in Economic MPC? – Comparison and Insights," 2020.

[94] Lowrey, K., Rajeswaran, A., Kakade, S., Todorov, E., and Mordatch, I., "Plan online, learn offline: Efficient learning and exploration via model-based control," *arXiv preprint arXiv:1811.01848*, 2018.

[95] Chen, D., Kim, Y., and Hyeon, E., "Receding-Horizon Safe Co-Optimization of the Velocity and Power-split of Plug-in Hybrid Electric Vehicles with Imperfect Prediction," Accepted, 2021 Annual American Control Conference (ACC).

[96] Ames, A. D., Xu, X., Grizzle, J. W., and Tabuada, P., "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, Vol. 62, No. 8, 2016, pp. 3861–3876.

[97] Agrawal, A. and Sreenath, K., "Discrete Control Barrier Functions for Safety-Critical Control of Discrete Systems with Application to Bipedal Robot Navigation." *Robotics: Science and Systems*, 2017.

[98] Advanced Research Projects Agency – Energy, "Next-Generation Energy Technologies for Connected and Automated On-Road Vehicles," Accessed June 19, 2021, `https://arpa-e.energy.gov/technologies/programs/nextcar`.

[99] Guanetti, J., Kim, Y., and Borrelli, F., "Control of connected and automated vehicles: State of the art and future challenges," *Annual reviews in control*, Vol. 45, 2018, pp. 18–40.

[100] Zhang, Y. and Cassandras, C. G., "An impact study of integrating connected automated vehicles with conventional traffic," *Annual Reviews in Control*, Vol. 48, 2019, pp. 347–356.

[101] Meng, X. and Cassandras, C. G., "Optimal control of autonomous vehicles for non-stop signalized intersection crossing," *2018 IEEE Conference on Decision and Control (CDC)*, IEEE, 2018, pp. 6988–6993.

[102] Zhang, Y. and Cassandras, C. G., "Decentralized optimal control of connected automated vehicles at signal-free intersections including comfort-constrained turns and safety guarantees," *Automatica*, Vol. 109, 2019, pp. 108563.

[103] Malikopoulos, A. A., Hong, S., Park, B. B., Lee, J., and Ryu, S., "Optimal control for speed harmonization of automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 20, No. 7, 2018, pp. 2405–2417.

[104] Houshmand, A. and Cassandras, C. G., "Eco-routing of plug-in hybrid electric vehicles in transportation networks," *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 1508–1513.

[105] Zhao, J., Hu, Y., Muldoon, S., and Chang, C.-F., ““InfoRich” Eco-Driving Control Strategy for Connected and Automated Vehicles,” *2019 American Control Conference (ACC)*, IEEE, 2019, pp. 4621–4627.

[106] Amini, M. R., Sun, J., and Kolmanovsky, I., “Two-layer model predictive battery thermal and energy management optimization for connected and automated electric vehicles,” *2018 IEEE Conference on Decision and Control (CDC)*, IEEE, 2018, pp. 6976–6981.

[107] Zhao, L., Mahbub, A. I., and Malikopoulos, A. A., “Optimal vehicle dynamics and powertrain control for connected and automated vehicles,” *2019 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 2019, pp. 33–38.

[108] Office of Energy Efficiency and Renewable Energy, “Annual Merit Review: Agenda,” Accessed June 19, 2021, `https://www.energy.gov/eere/vehicles/annual-merit-review-agenda`.

[109] Office of Energy Efficiency and Renewable Energy, “Batteries, Charging, and Electric Vehicles,” Accessed June 19, 2021, `https://www.energy.gov/eere/vehicles/batteries-charging-and-electric-vehicles`.

[110] Suzuki, Y., Nishimine, A., Baba, S., Miyasaka, K., Tsuchida, M., Endo, H., Yamamura, N., and Miyazaki, T., “Development of new plug-in hybrid transaxle for compact-class vehicles,” Tech. rep., SAE Technical Paper, 2017.

[111] Liu, J. and Peng, H., “Modeling and control of a power-split hybrid vehicle,” *IEEE transactions on control systems technology*, Vol. 16, No. 6, 2008, pp. 1242–1251.

[112] Bertoni, L., Guanetti, J., Basso, M., Masoero, M., Cetinkunt, S., and Borrelli, F., “An adaptive cruise control for connected energy-saving electric vehicles,” *IFAC-PapersOnLine*, Vol. 50, No. 1, 2017, pp. 2359–2364.

[113] Turri, V., Besselink, B., and Johansson, K. H., “Cooperative Look-Ahead Control for Fuel-Efficient and Safe Heavy-Duty Vehicle Platooning,” *IEEE Transactions on Control Systems Technology*, Vol. 25, No. 1, 2017, pp. 12–28.

[114] Maamria, D., Gillet, K., Colin, G., Chamaillard, Y., and Nouillant, C., “Optimal predictive eco-driving cycles for conventional, electric, and hybrid electric cars,” *IEEE Transactions on Vehicular Technology*, Vol. 68, No. 7, 2019, pp. 6320–6330.

[115] Gao, F., Li, S. E., Zheng, Y., and Kum, D., “Robust control of heterogeneous vehicular platoon with uncertain dynamics and communication delay,” *IET Intelligent Transport Systems*, Vol. 10, No. 7, 2016, pp. 503–513.

[116] Kayacan, E., “Multiobjective H_∞ Control for String Stability of Cooperative Adaptive Cruise Control Systems,” *IEEE Transactions on Intelligent Vehicles*, Vol. 2, No. 1, 2017, pp. 52–61.

[117] Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V., *Linear matrix inequalities in system and control theory*, SIAM, 1994.

[118] Naus, G. J. L., Vugts, R. P. A., Ploeg, J., van de Molengraft, M. J. G., and Steinbuch, M., “String-Stable CACC Design and Experimental Validation: A Frequency-Domain Approach,” *IEEE Transactions on Vehicular Technology*, Vol. 59, No. 9, 2010, pp. 4268–4279.

[119] Zhou, Y., Ahn, S., Wang, M., and Hoogendoorn, S., “Stabilizing mixed vehicular platoons with connected automated vehicles: An H-infinity approach,” *Transportation Research Procedia*, Vol. 38, 2019, pp. 441–461.

[120] Maschuw, J. P., Keßler, G. C., and Abel, D., “LMI-based control of vehicle platoons for robust longitudinal guidance,” *IFAC Proceedings Volumes*, Vol. 41, No. 2, 2008, pp. 12111–12116.

[121] Németh, B., Gáspár, P., Orjuela, R., and Basset, M., “Robust H design of an automotive cruise control system,” IFAC, 2015.

[122] Junaid, K., Shuning, W., Usman, K., and Wencheng, T., “Intelligent longitudinal cruise control by quadratic minimization and robust synthesis,” *IEEE International Conference on Vehicular Electronics and Safety, 2005.*, IEEE, 2005, pp. 182–187.

[123] Liang, C.-Y. and Peng, H., “Optimal adaptive cruise control with guaranteed string stability,” *Vehicle system dynamics*, Vol. 32, No. 4-5, 1999, pp. 313–330.

[124] Li, S. E., Gao, F., Li, K., Wang, L.-Y., You, K., and Cao, D., “Robust longitudinal control of multi-vehicle systems—A distributed H-infinity method,” *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19, No. 9, 2017, pp. 2779–2788.

[125] Liao-McPherson, D., Huang, M., and Kolmanovsky, I., “A Regularized and Smoothed Fischer–Burmeister Method for Quadratic Programming With Applications to Model Predictive Control,” *IEEE Transactions on Automatic Control*, Vol. 64, No. 7, 2019, pp. 2937–2944.

[126] Liu, K., Asher, Z., Gong, X., Huang, M., and Kolmanovsky, I., “Vehicle velocity prediction and energy management strategy part 1: Deterministic and stochastic vehicle velocity prediction using machine learning,” Tech. rep., SAE Technical Paper, 2019.