Optimally-Personalized Hybrid Electric Vehicle Powertrain Control

DISSERTATION

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Xiangrui Zeng

Graduate Program in Mechanical Engineering

The Ohio State University

2016

Dissertation Committee:
Professor Junmin Wang, Advisor
Professor Ryan Harne
Professor Chia-Hsiang Menq
Professor Haijun Su
Abstract

One of the main goals of hybrid electric vehicle technology is to improve the energy efficiency. In industry and most of academic research, the powertrain control is designed and evaluated under standard driving cycles. However, the situations that a vehicle may encounter in the real world could be quite different from the standard cycles. Studies show that the human drivers have a great influence on the vehicle energy consumptions and emissions. The actual operating conditions that a vehicle faces are not only dependent on the roads and traffic, but also dependent on the drivers. A standard driving cycle can only represent the typical and averaged driving style under the typical driving scenarios, therefore the control strategies designed based on a standard driving cycle may not perform well for all different driving styles. This motivates the idea to design optimally-personalized hybrid electric vehicle control methods that can be adaptive to individual human driving styles and their driving routes. Human-subject experiments are conducted on a driving simulator to study the driving behaviors. A stochastic driver pedal model that can learn individual driver’s driving style is developed first. Then a theoretic investigation on worst-case relative cost optimal control problems, which is closely related to vehicle powertrain optimal control under real-world uncertain driving scenarios, is presented. A two-level control structure for plug-in hybrid electric vehicles is proposed, where the parameters in the lower-level controller can be on-line adjusted via optimization using historical driving data. The methods to optimize these parameters
are designed for fixed-route driving first, and then extended to multi-routes driving using the idea similar to the worst-case relative cost optimal control. The performances of the two proposed methods are shown through simulations using human driving data and stochastic driver model data respectively. The energy consumption results in both situations are close to the posteriori optimal result and outperform other existing methods, which show the effectiveness of applying optimally-personalized energy management strategy on hybrid electric vehicles. Finally, a route-based global energy-optimal speed planning method is also proposed. This off-line method provides a useful tool to evaluate the potential of other speed planning methods, for either eco-driving guidance applications or future automated vehicle controls. The contributions of this dissertation include 1) a novel stochastic driver pedal behavior model which can learn independent drivers’ driving styles is created, 2) a new worst-case relative cost optimal control method is proposed, 3) a real-time implementable stochastic optimal energy management strategy for hybrid electric vehicles running on fixed routes is designed using the statistics of history driving data, 4) the fix-route strategy is extended to the multi-route situation, and 5) an off-line global energy-optimal speed planning solution for road vehicles on a given route is presented.
Acknowledgments

I would like to express my sincere gratitude to my advisor Professor Junmin Wang who provides me continuous guidance, inspiration, advice, encouragement, and support during my Ph.D. study at The Ohio State University.

I also thank Professor Ryan Harne, Professor Chia-Hsiang Menq, and Professor Haijun Su for serving on my dissertation committee, and Professor Manoj Srinivasan for serving on my candidacy exam committee. Their valuable suggestions are very important to my dissertation research.

In addition, I thank my colleagues and friends at the Vehicle Systems and Control Laboratory, including Fengjun Yan, Yan Chen, Rongrong Wang, Umar Ibrahim, Xiaoyu Huang, Pingen Chen, Junfeng Zhao, Hui Zhang, Scott Schnelle, Guoguang Zhang, Zitian Yu, Yao Ma, SeHwan Kim, Yimin Chen and many others for their help.

Finally, I would like to thank my beloved parents, Fanxian Zeng and Weinong Xiao, for their love and supports.
Vita

July 2009 .......................................................... B.S., Vehicle Engineering, Tsinghua University

July 2012 .......................................................... M.S., Power Engineering and Engineering Thermophysics, Tsinghua University

August 2012 to present .................. Ph.D. Student, Department of Mechanical and Aerospace Engineering, The Ohio State University

2015-2016 ................................................. The Ohio State University Graduate School Presidential Fellowship

Publications

Journal Articles


Peer-Reviewed Conference Papers


Fields of Study

Major Field: Mechanical Engineering
Table of Contents

Abstract ........................................................................................................................................ ii

Acknowledgments ................................................................................................................ iv

Vita ................................................................................................................................................ v

List of Tables .......................................................................................................................... xv

List of Figures ........................................................................................................................ xvii

Chapter 1: Introduction ........................................................................................................... 1
  1.1. Background ......................................................................................................................... 1
  1.2. Literature Review .............................................................................................................. 4
      1.2.1. Literature Review of Driver Pedal Behavior Models ............................................ 4
      1.2.2. Literature Review of Hybrid Electric Vehicle Energy Management Strategies 9
  1.3. Driving Simulator Experiment Platform .......................................................................... 11
  1.4. Outline ............................................................................................................................... 13

Chapter 2: A Stochastic Driver Pedal Behavior Model .......................................................... 16
  2.1. The Driver Pedal Model Focused in this Research ........................................................ 17
  2.2. Driver Pedal Action Analysis ........................................................................................... 18
2.2.1. Driving Simulator Experiment ................................................................. 19
2.2.2. Intermittent Pedal Adjustments .............................................................. 20
2.2.3. Pedal Action Decomposition and Reconstruction ..................................... 21

2.3. A Brief Introduction of IOHMM .................................................................... 23

2.3.1. IOHMM Formulation .............................................................................. 23
2.3.2. Parameter Identification of IOHMM ........................................................ 24

2.4. Model Framework under IOHMM ............................................................... 25

2.4.1. Model Inputs and Outputs ...................................................................... 25
2.4.2. Model States ............................................................................................ 26
2.4.3. Output Probability Distribution .............................................................. 28

2.5. Model Details ............................................................................................... 30

2.5.1. State Transition Probability Functions .................................................... 30
2.5.2. Output Probability Distribution Functions ............................................. 33
2.5.3. Model Parameter Identification ............................................................... 37

2.6. Model Performance ..................................................................................... 37

2.6.1. One-Step Probability Prediction Test ...................................................... 38
2.6.2. Multi-Step Closed-Loop Prediction Test .................................................. 42

2.7. Summary ..................................................................................................... 48

Chapter 3: Worst-Case Relative Cost Optimal Control ..................................... 49

x
3.1. Optimal Control for Systems with Disturbances.............................................. 50
3.2. WCRC Problem Formulation............................................................................. 52
  3.2.1. System Dynamics ......................................................................................... 52
  3.2.2. Disturbance Sequences ................................................................................ 53
  3.2.3. Control Policy and Cost Function ................................................................. 54
  3.2.4. The Principle of Optimality and WCRC Problem ......................................... 55
3.3. Solution of WCRC Problem.............................................................................. 57
  3.3.1. A Key Theorem ............................................................................................ 57
  3.3.2. Solving Procedure ......................................................................................... 59
3.4. Extension to the Stochastic Case....................................................................... 60
  3.4.1. Probabilities of Disturbance Sequences ......................................................... 61
  3.4.2. Expected Cost Function ................................................................................. 62
  3.4.3. Stochastic WCRC Problem .......................................................................... 63
  3.4.4. Extension of Theorem 1 ............................................................................... 63
  3.4.5. Solving Procedure ......................................................................................... 65
3.5. Numerical Examples ......................................................................................... 65
  3.5.1. A Linear System with Quadratic Cost ............................................................ 65
  3.5.2. A Nonlinear Hybrid Electric Vehicle Problem .............................................. 67
3.6. Summary ............................................................................................................. 70
Chapter 4: Optimizing the Energy Management Strategy for Fixed-Route HEVs .......... 72

4.1. Fixed-Route Driving ................................................................................................. 73

4.2. Driving Data Analysis ............................................................................................. 75

4.2.1. Driving Simulator Experiment ............................................................................. 76

4.2.2. Characteristics of the Fixed-Route Driving Data .................................................. 77

4.3. Stochastic Optimal Control for Fixed-Route HEVs ................................................. 82

4.3.1. Two-Level Control Structure and Road Segmentation ....................................... 83

4.3.2. Basic HEV Model and Lower-Level Controller .................................................... 87

4.3.3. Segment-Based Stochastic Model ......................................................................... 89

4.3.4. SDP for Higher-Level Controller Design ............................................................ 94

4.3.5. Real-Time Implementation .................................................................................. 96

4.4. Simulation Studies .................................................................................................. 97

4.5. Summary ................................................................................................................ 103

Chapter 5: Optimizing the Energy Management Strategy for Multi-Route Personal Plug-in HEVs ........................................................................................................ 104

5.1. Multi-Route Personal HEVs .................................................................................... 104

5.1.1. Describing Different Routes ................................................................................ 105

5.1.2. Problem Formulation .......................................................................................... 106

5.2. Solution of the Proposed Problem ........................................................................... 108

xii
6.4.2. Parallel Computation ................................................................. 135
6.4.3. Additional Boolean State to Improve Comfortability ....................... 135
6.5. Simulation Examples ........................................................................ 136
  6.5.1. Route and Vehicle Information ..................................................... 136
  6.5.2. DP Parameters ........................................................................ 139
  6.5.3. Simulation Results .................................................................... 140
6.6. Summary ......................................................................................... 145

Chapter 7: Conclusions, Contributions, and Future work................................. 146
  7.1. Conclusions ................................................................................ 146
  7.2. Contributions ............................................................................ 148
  7.3. Future Work ............................................................................. 149

References .......................................................................................... 151
List of Tables

Table 1. Some physics-based driver models................................................................. 9
Table 2. Interested physical variables for different mode transition pairs.................. 31
Table 3. Equations for no-action probabilities in all modes ........................................ 35
Table 4. Equations for completely-release probabilities in all modes ......................... 35
Table 5. Equations for mean output values (except no-actions) in all modes .............. 36
Table 6. Standard deviations in all modes ................................................................. 36
Table 7. The state transition probability examples ..................................................... 41
Table 8. log V for each driver model applied to each driver’s driving data ................. 42
Table 9. Pedal position SEE (%) of models in different prediction horizons .............. 46
Table 10. Speed position SEE (m/s) of models in different prediction horizons ......... 46
Table 11. Average pedal position SEE (%) of different models in different prediction horizons, tested with Driver A’s data ............................................................. 48
Table 12. Cost comparison of three control policies .................................................... 70
Table 13. Test track parameters .................................................................................. 77
Table 14. Overview of test trips .................................................................................. 77
Table 15. Time and energy of 10 trips when the vehicle runs through an intersection ... 80
Table 16. Higher-level and lower-level system comparison ......................................... 85
Table 17. HEV powertrain specification ..................................................................... 97
Table 18. Total energy consumption results of the 24 trips under different strategies... 103
Table 19. Route information.......................................................................................... 114
Table 20. Total energy consumption results of the 200 trips under different strategies. 116
Table 21. Simulation vehicle parameters....................................................................... 138
Table 22. Cost and energy consumption results ............................................................. 144
List of Figures

Figure 1. Virtual reality driving simulator structure .............................................................. 12
Figure 2. The driving simulator platform [57] ........................................................................... 12
Figure 3. A human driver’s driving data collected on the driving simulator [59] ........... 21
Figure 4. Human driver pedal positions and pedal actions [59] ........................................... 22
Figure 5. The closed-loop driver-vehicle-road system [59] ...................................................... 26
Figure 6. Different modes mean different mappings from the input to the output [59] ... 27
Figure 7. The shape of a typical output probability distribution in a certain mode [59] .. 29
Figure 8. Transition factor function and transition probability evaluation function examples [59] .................................................................................................................. 32
Figure 9. Driving data from a human driver [59] ................................................................. 40
Figure 10. The output probability distribution result examples from one-step prediction test [59] ...................................................................................................................... 40
Figure 11. 10-second IOHMM-based driver pedal model prediction at t = 97s [59] ...... 44
Figure 12. 60-second IOHMM-based driver pedal model prediction at t = 56s [59] ...... 44
Figure 13. Performance comparison of different models in different predictions horizons [59] ........................................................................................................................................ 47
Figure 14. 192 admissible disturbance sequences and the nominal disturbance sequence [66] ........................................................................................................................................ 66
Figure 15. Absolute costs and relative costs of the different control methods [66] ........ 66
Figure 16. Speed, road grade, motor speed, and torque demand of the two routes [66] .. 68
Figure 17. The test track used in the driving simulator experiment [57].................... 76
Figure 18. The vehicle speed of 10 trips when running through an intersection [57]...... 78
Figure 19. The pedal positions of 10 trips when the vehicle runs through an intersection [57]........................................................................................................ 79
Figure 20. Time and energy of 24 trips when the vehicle runs through an intersection [57]........................................................................................................ 81
Figure 21. Overall time and global energy of 24 trips [57]................................. 81
Figure 22. Two-level structure of HEV control with fixed-route trip information [57]... 85
Figure 23. Road segmentation of the test map [57].......................................... 86
Figure 24. A HEV configuration [57].............................................................. 87
Figure 25. Boxplot examples of the SOC changes and fuel consumptions [57]........ 91
Figure 26. An example: approximated probability density distribution of $\Delta x_{soc}$ [57] .... 92
Figure 27. An example: approximated probability density distribution of $\Delta f_{fu}$ [57].... 93
Figure 28. SDP result: a 2-dimensional lookup table [57]................................. 96
Figure 29. The speed and torque demand profiles of one trip [57]....................... 99
Figure 30. Final SOC of the 24 trips under different strategies [57].................... 101
Figure 31. Battery SOCs of one trip under different strategies [57].................... 101
Figure 32. Equivalent fuel costs of the 24 trips under different strategies [57]......... 102
Figure 33. Road map that the simulation routes cover .................................... 113
Figure 34. Eight routes in the simulation......................................................... 113
Figure 35. ECMS equivalent energy cost under different factors ........................................ 115
Figure 36. An example of Some state constraints [95] .......................................................... 127
Figure 37. An example of the state skipping the stop sign [95] .......................................... 129
Figure 38. Another example of the state skipping the stop sign [95] .......................... 130
Figure 39. Relaxed state and input constraints for numerical DP implementation [95]. 131
Figure 40. A real-world route map for the simulation [95] .................................................. 137
Figure 41. Altitude and road grade of the route [95] ............................................................. 137
Figure 42. Driving efficiency map of one in-wheel motor [95] ............................................. 138
Figure 43. Regenerative braking efficiency map of one in-wheel motor [95] ................. 139
Figure 44. The optimal and nominal speed planning results of the fast trip [95] ............ 141
Figure 45. The optimal and nominal speed planning results of the slow trip [95] ....... 141
Figure 46. The optimal and nominal trip time-distance results and traffic light schedules [95] ........................................................................................................................................................................... 142
Figure 47. The optimal and nominal speed planning results in distance domain [95] ... 143
Figure 48. The interpolated optimal cost-to-go and the actual cost-to-go [95] .......... 145
Chapter 1: Introduction

1.1. Background

The two main goals of the vehicle powertrain control are to lower the energy consumption and tailpipe emissions. Various technologies on the powertrains such as exhaust gas recirculation (EGR), variable geometric turbocharger (VGT), variable valve actuation (VVA) and advanced combustion modes including homogenous charge compression ignition (HCCI), premixed controlled compression ignition (PCCI) and low temperature combustion (LTC) have been studied to achieve low fuel consumption and engine-out emissions. Plug-in hybrid electric vehicles (plug-in HEVs) can also significantly save energy and reduce emissions as a secondary power source is introduced to adjust the engine operating condition and make regenerative braking possible.

Most research and engineering applications of powertrain control are designed and evaluated based on the standard driving cycles [1][2][3]. A driving cycle, such as the Urban Dynamometer Driving Schedule (UDDS), Federal Test Procedure (FTP), and New York City Cycle (NYCC), is a sequence of data representing the vehicle speed versus time and it is considered as a typical or averaged operating cycle that a vehicle may encounter.

However, even though a driving cycle is carefully designed according to a vast amount of real-world driving data, the situations that a vehicle runs in the real world are
still different from that under a standard driving cycle. The control strategies optimized based on the cycles may not be the optimal ones for the real-world driving. Therefore it is necessary to study the powertrain system control methodology for the real-world circumstances.

In the real-world driving, there are a lot of uncertain factors that may affect the vehicle operations. With global positioning system (GPS), geographic information system (GIS), and other intelligent transportation systems (e.g. vehicle-to-vehicle communications), much information about the driving environment is available to the vehicle controller, for example, the road grade, the speed limit, and the traffic light information, which all could help to reduce the uncertainty and build an accurate model of the driving. In addition, the human driver, which is the most sophisticated and uncertain part of the driving system, plays a very important role and has not been studied thoroughly yet.

Each independent driver has his or her own driving style. Driving styles have a great influence on the vehicle energy consumption and emissions. Research in [4] compares the fuel consumption and emissions of different drivers. The experiment result shows that aggressive drivers will consume 12-40% more fuel than moderate drivers, depending on the road types and vehicle technologies. It is also shown that for gasoline engines equipped with three-way catalysts, due to the open-loop control in some transient conditions, CO emission can increase by 700%, HC can increase by 400%, and NOx can increase by 150% for the most aggressive drivers under certain road types. For Diesel engine vehicles the increase for CO and HC is below 50%, but still significant emission
rise can be observed. Energy consumption of different drivers in electric vehicles [5] and plug-in hybrid electric vehicles [6] has also been studied. In electric vehicles, the difference is even larger than that on the conventional vehicles, as the experiment shows that the regenerated energy can be highly influenced by drivers. In [5] the calculated regenerative energy capture efficiency is 93% for a moderate driver compared to only 15% for an aggressive driver, and there is a difference of 91% in total energy consumed between the best and worst driver on the high-speed circuit. These studies show that in similar environments, different driving styles may lead to significantly different energy consumption and emission performance, so the real-world driving cannot be represented by the averaged standard driving cycles. In fact, a standard driving cycle can only represent the typical and averaged driving style under the typical driving scenarios, therefore the control strategies designed based on a same standard driving cycle may not perform well for all different driving styles.

Besides the driving styles, each driver’s driving routes also have great impact on the energy consumption. For plug-in HEVs, it is important to use the electricity energy wisely such that the benefit of using multiple energy sources can be maximized. The desired strategy is closely related to the driving route, as for different lengths and types of route (e.g., local or highway, uphill or downhill) the optimal energy management strategy should be different. Therefore the driving routes should also be considered in the strategy personalization.

In reality, it is usually the case that one driver may operate one vehicle for a long time, maybe for several years. If the plug-in HEV powertrain control system is able to be
adaptive to the specific driver’s driving style and driving route, it may lead to a better energy consumption performance. This also motivates the research of this dissertation.

It is possible that in the future, autonomous vehicles might dominate the market when they become mature. However the personalized strategies are still necessary for two reasons. First, in the autonomous vehicle era human-driven vehicles will still exist on the road, as some people may just prefer to drive themselves. Second, even for autonomous vehicles, where there will be no more individual driving styles, the driving routes are still different for different customers. So the personalization focusing on the driving routes is still necessary.

1.2. Literature Review

1.2.1. Literature Review of Driver Pedal Behavior Models

Many studies have been conducted on the driver’s behaviors. In this study the research interest of driver’s behaviors will be limited to the driver’s pedal behavior under normal driving, which will have significant influence on the powertrain systems. The driver’s behaviors on steering, or in emergency like accident avoidance situations, are not considered.

In general there are two classes of driver models: data-driven models and physics-based models. In the data-driven models the driver’s behaviors are models by some universal signal models so they may not have explicit physical meanings. The physics-based models are usually designed to describe driving behaviors such that at least some of their parameters would have clear physical meanings.
1) Autoregressive exogenous (ARX) model

ARX model is a type of data-driven model. It uses the previous inputs and outputs (or states) to predict the future outputs, usually through a linear function \([7]\). ARX model can be applied to predict the driver power demand. A typical ARX model is as follows:

\[
y_k = f(r_k) + e_k,
\]

\[
r_k = [y_{k-1}, ..., y_{k-n}, u_{k-1}, ..., u_{k-n}]^T,
\]

where \(y_k\) is the output, \(e_k\) is the error term, \(u_k\) is the input, \(r_k\) is the regressive vector which contains the history of inputs and outputs. \(f(\cdot)\) is usually a linear function. It is usually difficult to describe the driver power demand signal by a single ARX model, because in different driving situations, e.g., car-following, accelerating, or turning, the driver outputs show different characteristics. Therefore, ARX model is usually used under the hybrid dynamical system (HDS) structure \([7][8][9][10][11]\). The discrete modes in the HDS can reflect different driving situations. ARX model is purely driven by data and it may not have a physical meaning. However, the HDS discrete modes may have implicit physical meanings corresponding to different driving situations. Many researches on ARX-based driver models focus on the discrete mode switching. The switching can be deterministic \([8][11]\) or stochastic (usually as a Hidden Markov Model) \([7][9][10]\). Under different HDS structures, ARX model identification methods are also different. All the papers mentioned above propose corresponding identification algorithms for the ARX-based models they present.

2) Gaussian mixture model (GMM)
GMM is also a data-driven model. In GMM the driver’s pedal position is assumed to have a joint probability distribution of some parameters which define the driving conditions [12], such as the car-following distance, velocity, pedal position and their first and second order derivatives. For example, let

\[ x_t = \{v_t, \Delta v_t, \Delta^2 v_t, f_t, \Delta f_t, \Delta^2 f_t, G_t, \Delta G_t, \Delta^2 G_t \}, \]

where \( v, f, G \) are the vehicle speed, car-following distance and pedal position respectively. \( \Delta(\cdot) \) operator means the derivatives. Define a set of augmented feature vector \( y_t = [x_t^T, G_{t+1}]^T \). Then, the joint density of the augmented vector \( y_t \) can be modeled by a GMM,

\[ P(y | \Phi) = \sum_{k=1}^{K} \alpha_k \phi_k(y), \]

where \( \alpha_k \) is the prior probability of the \( k \)-th mixture component, \( K \) is the total number of Gaussian components, and \( \phi_k(\cdot) \) is the unimodal Gaussian distribution. The predicted pedal \( \hat{G}_{t+1} \) can then be computed by the weighted predictions resulting from all mixture components [12][13]. GMM is usually used in the car-following situations. But it can be extended to all driving situations.

Both ARX model and GMM can be used to predict the future driver power or speed demand. In [14] the prediction performance of a GMM and a piecewise auto regressive exogenous (PWARX) model for car-following is evaluated. The prediction capabilities of both models as the prediction time increases from one to five seconds are illustrated. The
result shows that both GMM and PWARX models provide comparable performance. The PWARX model slightly outperforms the GMM model.

3) Markov chain model

Markov chains have been widely used to generate the driver’s torque demand, which is considered as a discrete-time stochastic dynamic process, in many hybrid electric vehicle control applications [15][16][17][18][19][20][21]. The motivation of modeling the driver’s torque demand as a Markov chain is to make the control strategy independent of a certain time-sequence driving cycle, but still maintain the characteristics of a certain driving cycle, so that the strategy (which is usually the result of an optimal control problem solved via stochastic dynamic programming) can be implemented in the real-world driving. A simple Markov chain describing the torque demand is as follows:

\[
P[T_{\text{dem}}(k+1) = T_{\text{dem}}^j | T_{\text{dem}}(k) = T_{\text{dem}}^i, \omega_{\text{wh}}(k) = \omega_{\text{wh}}^j] = p_{i,j},
\]

where \( T_{\text{dem}} \) is the torque demand, \( \omega_{\text{wh}} \) is the wheel speed, and \( P(\cdot | \cdot) \) stands for the conditional probability. This means that the torque demand transition probabilities are determined only by the current torque demand and wheel speed. The transition probabilities could be adapted online to reflect the actual driving pattern rather than a standard driving cycle [22]. Besides the HEV powertrain applications, the Markov chain can also be used for conventional powertrain control [21][22].

4) Machine learning model

Machine learning is used in some driving behavior modeling applications, including learning driver’s intention [23][24][25], identifying distraction [23], and predicting driver’s foot movement by vision-based foot gesture analysis [26]. It can also be used to
model human driver’s desired speed trajectory [27], but the method in [27] does not emphasize on individual drivers’ driving style differences and model performance is not compared for different drivers.

5) Physics-based car-following models

Physics-based models are usually employed for car-following or microscopic traffic flow research. Some models aim to describe the vehicle velocity or acceleration, but they can be considered as equivalent to the driver pedal modeling. At least some of the parameters in these models have physical meanings. The physics-based models are summarized in [28][29][30]. Some of the important models are listed in Table 1.

It should be noted that many of the physics-based models are developed in the 1950s or 1960s for the transportation research and they are not designed for the purposes of powertrain control which should maximally utilizes the GPS, GIS and other intelligent transportation system information. Most of these car-following models are deterministic, which means that they cannot describe the uncertain factors in driving. These speed models are usually not valid under other scenarios except car following.

6) Other driver models

Besides the abovementioned models, there are some other studies on data-driven driver pedal behavior models using neural network [39], semiotic method [40], and Bayesian model [41].
<table>
<thead>
<tr>
<th>Model</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gazis–Herman–Rothery (GHR) model [31]</td>
<td>[ a_n(t) = cv_n^m(t) \frac{\Delta v(t-T)}{\Delta x'(t-T)} ]</td>
</tr>
<tr>
<td>Linear follow-the-leader [32]</td>
<td>[ \dot{V}_r = (V_L - V_F) ]</td>
</tr>
<tr>
<td>Nonlinear follow-the-leader [33]</td>
<td>[ \Delta x(t-T) = a_{n-1}^2(t-T) + \beta v_{n-1}(t) + \beta v_n(t) + b_0 ]</td>
</tr>
<tr>
<td>Collision-avoidance (CA) model [34]</td>
<td>[ v_j(t) = V_f \left[ 1 - \exp{-\lambda V_j^{-1}[x_{j-1}(t-\Delta) - x_j(t-\Delta) - d_j]} \right] ]</td>
</tr>
<tr>
<td>Newell model [32]</td>
<td>[ v_a(t+\tau) = \min \left{ \frac{V_a(t) + 2.5a_n(1-v_a(t))}{V_a} \right} ]</td>
</tr>
<tr>
<td>Gipps model [35]</td>
<td>[ a_n(t) = C_1\Delta v(t-T) + C_2 { \Delta x(t-T) - D_n(t) } ]</td>
</tr>
<tr>
<td>Linear model [36]</td>
<td>[ D_n(t) = \alpha + \beta v(t-T) + \gamma \alpha_n(t-T) ]</td>
</tr>
<tr>
<td>Linear optimal control model [37]</td>
<td>[ u(t) = C_1(V_L - V_F) + C_2 {(S_L - S_F) - C_1 V_F} ]</td>
</tr>
<tr>
<td>Intelligent driver model [38]</td>
<td>[ s^2(v, \Delta v) = s_0(a) + s_1(a) \frac{v}{v_0^{(a)}} + T^a v + \frac{v \Delta v}{2a^2 b^{(a)}} ]</td>
</tr>
</tbody>
</table>

1.2.2. Literature Review of Hybrid Electric Vehicle Energy Management Strategies

Plug-in HEVs have been widely researched as they have the potential to improve vehicle fuel economy and reduce emissions. The internal combustion engines and electric
motors of the plug-in HEVs have quite different characteristics in terms of energy efficiency therefore the energy management strategies are crucial to the plug-in HEV performance. If the complete driving cycle information is known in advance, dynamic programming (DP) [1] and Pontryagin's minimum principle (PMP) [42][43] can be used to calculate the optimal energy management strategy. However knowing the cycle a priori is hardly possible in practice, so DP strategy usually cannot be implemented for real-world driving. Heuristic rules (usually developed based-on DP results) [1][44], fuzzy logic [45], and equivalent consumption minimization strategy [46][47] can be applied to the HEV energy management systems without any knowledge of the trip information.

However these strategies are usually developed or tuned based on a standard driving cycle and they cannot guarantee their performances over different driving scenarios. Though the complete driving information of the trip can hardly be known in advance, the near future powertrain operation may ususally be predicted in some way, and this information could also help improving HEV fuel economy and emission performance. Model predictive control (MPC) utilizes the prediction and optimizes the control strategy over the prediction horizon iteratively, thus it is also popular for HEV energy management systems [17][20][48][49][50][51][52][53][54][55][56]. Some studies show that the well-tuned ECMS and MPC strategies have very good performances under standard driving cycles as their energy consumption results are quite close to the optimal ones calculated by DP [46][48]. However, the tuning is dependent on the driving cycle and the comparison is only made under that specific driving cycle.
1.3. Driving Simulator Experiment Platform

In this research, the driving styles were studied on a virtual-reality (VR) driving simulator. The structure of the driving simulator system is shown in Figure 1. The VR simulation software platform includes a vehicle powertrain and dynamics model in Simulink and a VR software named Vizard. The steering wheel and pedal signals are sent to the Simulink model. The Simulink model communicates with Vizard VR software through User Datagram Protocol (UDP), receives the road geometric information from Vizard, computes the vehicle position and direction and sends the data back to Vizard. A vehicle-road-environment 3D model is imported to the Vizard VR software and a virtual reality driving scenario is programmed in Python language and runs in Vizard. Vizard VR engine will generates the output to the two displays according to the vehicle-road-environment 3D model and the vehicle position and direction data from Simulink. On this VR driving simulator, human-subject tests on driver’s behaviors under different driving circumstances could be conducted by applying different vehicle-road-environment 3D models in the Vizard. A photo of the driving simulator is shown in Figure 2.
Figure 1. Virtual reality driving simulator structure

Figure 2. The driving simulator platform [57]
The vehicle model used in the simulator is based on a mid-size passenger car with a 3.2L Diesel engine and 6-speed automatic transmission. The vehicle dynamic model is a 3-degree-of-freedom vehicle body model with four 4-degree-of-freedom tire models based on the Magic Formula tire model [58] with load transfer considered. A map-based engine model and a pedal-speed-map-based transmission shifting strategy are used in the simulator. The vehicle model and pedal data acquisition both update every 0.02 second. On this simulator, the traffic lights can be configured as vehicle-position-triggered so every driver can encounter the same lights despite different arrival time, if necessary.

1.4. Outline

The rest of this dissertation is organized as follows:

In Chapter 2, a stochastic driver pedal behavior model is developed. The proposed model provides a new way to describe driver’s pedal behavior after decomposing it into a sequence of actions. By considering the vehicle and road information as inputs and the pedal action as the output, an input-output hidden Markov model (IOHMM) is used to describe the pedal behavior. This model can reflect the individual driver’s driving style and performs well in prediction horizons from 1 second to 60 seconds.

Chapter 3 proposes a theoretic control method to solve the worst-case relative cost (WCRC) optimal control method for discrete-time nonlinear dynamic systems. WCRC optimal control problems share some similarities with the energy-optimal control problem for vehicle powertrains, as the one of the goals of vehicle powertrain control is to result in energy consumption close to the posteriori optimal cost. The problem targeted
in this chapter assumes that the disturbance sequence belongs to a known finite admissible set. Then the problem is extended to a stochastic case, where the probability range of each disturbance sequence is known. Two numerical examples are provided to illustrate the proposed method.

Chapter 4 focuses on developing a two-level stochastic approach to optimize the energy management strategy for HEVs running on fixed-route. The targeted fixed-route vehicles can be personal commuter cars, public transportation buses, and some utility vehicles. The historical data on the fixed route are utilized and a road-segment-based stochastic HEV energy consumption model is built. The higher-level energy optimization problem is solved by stochastic dynamic programming (SDP). The SDP computation uses the vehicle model and historical driving data on the fixed route and it can be conducted offline. The result of SDP is a 2-dimension lookup table of parameters for lower-level control strategy therefore this approach can be easily implemented in real-time in practice. Simulation results show that the proposed stochastic energy management strategy consumes only 1.8% more energy than the optimal result and outperforms other strategies.

Chapter 5 is an extension of Chapter 4, presenting an approach to optimize personal plug-in HEV energy management strategy using historical driving data and GPS information. The target vehicles do not need to run on a fixed route. Instead, just like most personal vehicles, they can run on multiple frequent routes and all the route data can be collected and utilized to develop an optimized strategy. The same road-segment-based HEV model is used. The control method is inspired by the WCRC optimal control theory.
proposed in Chapter 3, and the driver model in Chapter 2 is used as a tool to generate virtual driving data. Simulation results show that the proposed energy management strategy consumes only 2.5% more energy than the posteriori optimal result and outperforms other strategies.

Chapter 6 provides a globally optimal solution to an important problem: given a real-world route, what is the most energy-efficient way to drive a vehicle from the starting point to the destination within a certain period of time, under the ideal situation where there are no uncertainties in the trip. This solution is not a real-time method, but it is a very useful tool that can calculate the posteriori optimal speed trajectory and the optimal energy consumption. This method has a lot of potential for control method development and evaluation in the autonomous vehicle era.

Chapter 7 concludes the dissertation and gives a brief discussion on future work of this research.
Chapter 2: A Stochastic Driver Pedal Behavior Model

Human driver’s pedal behavior is difficult to model because it is the output of a very complicated virtual stochastic system. This chapter provides a new way to describe driver’s pedal behavior after decomposing it into a sequence of actions. By considering the vehicle and road information as inputs and the pedal action as the output, an input-output hidden Markov model (IOHMM) is used to describe the pedal behavior. The state transition and output distribution functions are designed, and the relation between the input and the key variables of the output distributions is analyzed and modeled using statistic methods. The model parameters can be identified for individual drivers using generalized estimation-maximization (GEM) method. One-step probability prediction test shows that the proposed model can capture and distinguish each individual driver’s driving style. The prediction capability of the proposed model is evaluated by comparing it with the human driver data collected on a driving simulator along with three other models. The results show that the proposed IOHMM-based driver pedal behavior model performs well in prediction horizons from 1 second to 60 seconds. This chapter is mainly based on the paper [59] which is submitted to a journal and under review.
2.1. The Driver Pedal Model Focused in this Research

Driver behaviors have great impacts on the energy consumption, emissions, and driving safety of road vehicles. In some vehicle powertrain supervisory control systems and intelligent driving assistance systems, driver models are incorporated to improve the performance via either driving behavior prediction or driver-in-the-loop control design [21][60][61][62]. This chapter focuses on the driver’s pedal behavior model that has the potential to be used in various vehicle powertrain model predictive control applications.

A comprehensive review of driver model is provided in Section 1.2.1. All of the aforementioned driver pedal prediction models use the sampled driver pedal absolute position as the output and study the characteristics of this output signal. These methods ignore the details in human driver’s sensing, decision making, and body moving processes, thus may not be able to sufficiently describe driver’s behaviors. In this chapter, the driver’s pedal behavior is considered as a sequence of actions. The probabilities of pedal actions are the outputs of the model, and all information that a driver receives is considered as the inputs to the model. This driver pedal behavior model is depicted under the IOHMM [63] framework. IOHMM can consider a large number of inputs, and the hidden state can represent the unknown driver’s driving mode. This IOHMM-based model can be nonlinear and its parameters can be identified off-line by techniques such as estimation-maximization (EM) or generalized estimation-maximization (GEM) method [63][64]. In this research, different human driver data are collected on a driving simulator. The pedal behavior model parameters are identified for each driver and the prediction performances are compared.
All experiment studies of this research are conducted on the same driving simulator with the same vehicle model. It should be noted that the driver’s pedal behavior may be dependent on the vehicle, the pedal to torque map, and even the physical pedal resistance feedback. This dependence is not studies in this research. The proposed model in this chapter is for generating/learning the drivers’ behavior for a given vehicle. For different vehicles the method developed in this chapter can still be applicable with the expectation that the model parameters may differ even for the same human driver due to the vehicle difference. This chapter will also focus on driver model sampled at about 1 second and has the potential to make prediction in horizons from several seconds to 1 minute. Therefore the proposed pedal behavior model can be employed for driving style identification and powertrain predictive control (e.g., to predict torque and speed in applications of model predictive control for HEVs [51]). Models sampled at much higher rate (e.g. 0.1s) can also be useful in some applications, but they may require more detailed inputs for better prediction performance (e.g., vision-based foot gesture analysis in [26]) and they will not be discussed in this chapter.

2.2. Driver Pedal Action Analysis

In this section, the driving simulator experiment used to conduct this driver pedal behavior research is introduced first. Based on the data collected on this platform, it is concluded that the driver’s adjustments to the pedal are intermittent, and the driver pedal action is defined and explained.
2.2.1. Driving Simulator Experiment

To better understand the human driver pedal behavior, human-subject experiments are conducted on the driving simulator introduced in Section 1.3. The road map model used is a mixture of highway and local roads with multiple stop signs, traffic lights, and speed limit changes. The human driver is required to follow the speed limits, stop signs, traffic lights, and other traffic regulations.

There is no other traffic on the road in this driving simulator so the car-following driving behavior is not tested. There are several reasons for not testing the car-following scenario. First, the proposed model in this research will be used to predict the future driver behavior and benefit the powertrain control. However in the car-following scenario such benefit cannot be obtained because the future driver behavior will be also dependent on the leading car whose future behavior is difficult to predict. Second, all the inputs used in the model can be obtained from current vehicle onboard sensors, geographic information system (GIS) and GPS. But car-following behavior model requires additional sensors or communication with the leading vehicle, which are not available at this moment for most cars. Third, car-following behavior has been extensively studied for the past decades, but relatively fewer models have been proposed for other scenarios. Therefore the model proposed in this work will focus on the data in other driving scenarios. Though the car-following driving data are not included in this research, the proposed IOHMM framework can be easily extended to the car-following scenario with additional inputs such as the distance to the leading vehicle, speed and acceleration of the leading vehicle, etc.
2.2.2. Intermittent Pedal Adjustments

When driving, the driver senses information from the vehicle and the road environment, makes decisions on whether he or she needs to adjust the pedal position, and takes actions by pushing or releasing the pedals. Usually, the sensing and decision-making processes are intermittent, which means that a driver may not be actively adjusting the pedal all the time, but only samples the vehicle and surrounding environment information, makes decisions when necessary, and conducts pedal position adjustments after a decision.

Figure 3 shows the vehicle speed and pedal position signals of a typical acceleration-cruise-stop (at a stop sign or a traffic light) process conducted by a human driver, sampled at 1 Hz. Positive pedal value means throttle pedal position and negative pedal value means braking pedal position. It can be seen that the driver almost does not make any adjustments on the pedal between the 1st second and the 4th second, between the 5th second and the 8th second, and only makes one minor adjustment between the 9th second and 16th second. This intermittent pedal adjustment phenomenon can be found in almost all data from different human drivers.
Figure 3. A human driver’s driving data collected on the driving simulator [59]

2.2.3. Pedal Action Decomposition and Reconstruction

The observation on the intermittent pedal adjustment motivates the idea of describing the pedal behavior by a sequence of actions. Each action is an adjustment of the pedal position. In general, an action can be depicted by two parameters: the magnitude of the action and the duration of the action. In our experiment, human driver data show that it can be assumed that for each driver, if two actions have the same magnitude, they have the same duration. With this assumption, a pedal action can be described by a scalar. When the sampling period is 1 second, almost all actions are completed within two sampling periods, and most of them are within one sampling period. In the rest of this chapter, for simplicity, it is assumed that the sampling period of pedal position is 1 second and every action takes only one sampling period. Figure 4 shows the decomposition of pedal positions into actions, and reconstruction of pedal position signals from the decomposed actions. It can be seen that in the 25-seconds
driving, only 9 actions are taken by the driver. Based on the 9 actions which are all assumed to be completed within 1 second, the reconstructed pedal signal is very close to the originally measured one, which means that the approximations on pedal action are acceptable.

![Figure 4. Human driver pedal positions and pedal actions [59]](image)

As the driver takes no action for most of the time, in order to describe the driver’s pedal behavior, it is necessary to model the probability distribution of the action, rather than only to model an averaged action or the most probable action (which is usually the trivial zero). This task can be completed under the IOHMM framework.
2.3. A Brief Introduction of IOHMM

IOHMM is an extension of the standard hidden Markov model without inputs. The basic formulation and GEM parameter identification for IOHMM will be introduced in this section.

2.3.1. IOHMM Formulation

The stochastic system considered in IOHMM is based on the following deterministic discrete-time discrete-state dynamical system,

\[ x_t = f(x_{t-1}, u_t) \]
\[ y_t = g(x_t, u_t), \]  
(6)

where \( u_t \in \mathbb{R}^m \) is the input vector at step \( t \), \( y_t \in \mathbb{R} \) is the output at step \( t \), and \( x_t \in X = \{1, 2, ..., N\} \) is the discrete state.

IOHMM considers the probabilistic version of the dynamics in (6). The current state probability distribution is determined by the previous state and current input, and the current output probability distribution is determined by the current state and input.

If \( \zeta_t \) is defined as the probability distribution vector of \( x_t \) over a set of discrete values, i.e., \( \zeta_t = [\xi_{1,t}, \xi_{2,t}, ..., \xi_{N,t}]^\top \in \mathbb{R}^N \) and \( \xi_{k,t} = P(x_t = k), k \in \{1, 2, ..., N\} \), the state transition can be written as,

\[ \zeta_t = \Phi(u_t)\zeta_{t-1}, \]  
(7)

where \( \Phi(u_t) \in \mathbb{R}^{N \times N} \) is a matrix.
If \( y_t \) also only takes \( M \) discrete values, the probability distribution vector of \( y_t \) over the set of finite discrete values can also be defined by a vector \( \eta_t \in \mathbb{R}^M \), then the output distribution can be written as

\[
\eta_t = \Pi(u_t)\zeta_t, \tag{8}
\]

where \( \Pi(u_t) \in \mathbb{R}^{M \times N} \) is a matrix.

### 2.3.2. Parameter Identification of IOHMM

Given an input-output sequence pair from \( t = 1 \) to \( t = T \), \( D = \{u_t^T, y_1^T\} \), where

\[
u_t^T = [u_1 \ u_2 \ldots u_T] \quad \text{and} \quad y_1^T = [y_1 \ y_2 \ldots y_T]
\]

(or given several such sequence pairs), as well as functions \( \Phi(\cdot), \Pi(\cdot) \) with unknown parameters, the identification problem can be solved via EM or GEM method. The aim of EM method is to maximize the log-likelihood function \( l(\Theta; D) = \log L(\Theta; D) \) where \( \Theta \) are the parameters and \( D \) are the observed input and output data. It can be achieved by a two-step method: estimation step and maximization step. EM was originally formalized in [64] and details about GEM for IOHMM can be found in [64]. These methods provide an iterative way to learn the parameters even if \( \Phi(\cdot) \) and \( \Pi(\cdot) \) are nonlinear. In this chapter, GEM is used to identify the model parameters.
2.4. Model Framework under IOHMM

This section explains how a driver pedal behavior model is built under the IOHMM framework. The inputs, output, and states of the IOHMM-based driver pedal behavior model are defined and the shape of output probability distribution is presented.

2.4.1. Model Inputs and Outputs

In the proposed model, the inputs to the IOHMM are the ones that the driver receives while driving, including the road and traffic information, the vehicle information, and current pedal position. In this research, the input signals include:

1) $v$ vehicle speed,
2) $v_l$ speed limit of the road,
3) $a$ vehicle acceleration,
4) $d$ distance to the stop sign or to the red light
5) $p$ current pedal position.

There are other potential inputs not included in this research, but could be added for a more comprehensive model:

6) distance to the leading vehicle,
7) leading vehicle speed,
8) leading vehicle acceleration,
9) distance to the next turn,
10) radius of the next turn,
11) road grade, etc.
The output of IOHMM is the probability distribution of pedal action. The pedal action is also discretized and the resolution is 1% for throttle pedal and brake pedal, therefore $M = 200$ and $\eta_t \in R^{200}$. In some situations, this probability distribution can be the output of the driver behavior model. But if a determined predicted pedal position is desired, an action translator with a random number generator (RNG) is needed to create a random variable with the specific probability distribution and output an actual pedal position. The closed-loop driver-vehicle-road system is shown in Figure 5. The red dash rectangle demonstrates the IOHMM-based driver pedal behavior model with inputs and outputs.

![Figure 5. The closed-loop driver-vehicle-road system [59]](image)

### 2.4.2. Model States

In IOHMM, the state $x_t$ determines the mapping between the input $u_t$ and the output $y_t$. The state transition is similar to the standard hidden Markov model, except that the state transition is not only dependent on the previous state $x_{t-1}$, but also dependent on the current input $u_t$. 

26
In this driver pedal behavior model, the state $x_t$ stands for the modes of driving behaviors, e.g., brake for a stop, free acceleration, or cruise, etc. In different modes, the driving behaviors have different patterns. For example, in the brake-for-a-stop mode, the driver pays a lot of attention on the distance to the position where he/she wants to stop, takes actions based on this distance feedback. In the cruise mode, the vehicle speed is the main feedback signal, and the driver takes actions to keep the speed close to a desired value. These modes determine the mapping between the input that a driver receives and the probabilities of the action that he or she takes.

The model structure of IOHMM is demonstrated in Figure 6. Different modes (states) mean different mappings from the input to the output probability distribution, and input can also impact mode transition probabilities. For each discrete driving mode $x_t$, the mapping from input $u_t$ to the output probability distribution vector $\eta_t$ forms one column of the matrix $\Pi(u_t)$. If the value of $x_t$ is unknown but the state probability distribution vector $\zeta_t$ is known, the output probability distribution $\eta_t$ can be determined by (8).

Figure 6. Different modes mean different mappings from the input to the output [59]
This driving mode is also hidden and the mode transition is stochastic, which can be explained by the following example. Assume that a driver is driving in acceleration mode, and the vehicle speed is close to the speed limit. For the same driver under the same input $u_i$, he or she may or may not transit into cruise mode. The driver may choose to continue focusing on acceleration, or to start focusing on speed feedback. This mode information cannot be observed directly, but can be somehow estimated from the observed output action.

Four modes are defined for the driver pedal behavior model:

1) acceleration mode,
2) cruise mode,
3) brake-for-a-stop mode,
4) stop mode.

Additional modes can be added for a more comprehensive model:

5) car-following mode,
6) brake-for-safe mode,
7) turning mode, etc.

2.4.3. Output Probability Distribution

Based on the understanding in human driving behavior and the observed driving data, the output probability distribution is in general not a simple normal distribution. Human driving data show that the probability of taking no action is usually high, and sometimes the driver tends to completely release the pedal. Therefore it is assumed that
the output probability distribution under a certain mode can be determined by the following four variables, which form a shape of distribution shown in Figure 7 (assuming that the current pedal position is 43% so the action that leads to zero pedal position is -43%):

1) $p_0$ the probability of taking no action,

2) $p_r$ the probability of the action that leads to zero pedal position,

3) $\mu$ the mean value of the action, if an action that does not lead to zero pedal position is taken,

4) $\sigma$ the standard deviation of the action, if an action that does not lead to zero pedal position is taken.

![Figure 7](image)

Figure 7. The shape of a typical output probability distribution in a certain mode [59]

With these four variables, the output under one known mode can be described by the following two equations,

$$\left[p_0, \sigma, \mu, p_r\right]^T = f_{out}(u_t, x_t),$$

$$\eta_t = f_{dia}(p_0, \sigma, \mu, p_r),$$
It should be noted that (9) (10) and Figure 7 only describe the situation that a determined mode is known. If only the probability of the mode is known, the output probability distribution would be the weighted sum of the output distributions under each mode, where the weight is the probability of the corresponding mode.

2.5. Model Details

Details of the proposed model including all the mode transitions and output mappings will be explained in the section.

2.5.1. State Transition Probability Functions

The state probability distribution defines the probability of state transitions. They are essentially some “soft thresholds” for mode transitions. For example, in a determined system, the transition condition from acceleration mode to cruise mode could be that the speed grows beyond a certain threshold. In this stochastic IOHMM system, the probability of such a transition increases as the speed grows. Usually this probability is very small when the vehicle speed is far below a certain soft threshold, and grows fast near this threshold, and is close to one when the speed is well above this threshold. Similar soft thresholds can be defined for each pair of mode transitions, and the probabilities of transition into every other mode and stay in the current mode form the complete state probability distribution.

Based on this idea, a transition factor, which functions as comparing the interested physical values with the threshold, and a corresponding transition probability evaluation
function, which renders this comparing result into a smooth probability change, are designed to depict the transition conditions. The transition factor is a number between 0 and 1, usually with some physical meanings. The transition probability evaluation function is a nonlinear mapping from [0, 1] to [0, 1] which translates the transition factor into the probability of transition.

The transition factor of the transition from mode $i$ to mode $j$ is defined as,

$$ f_{ij} = \begin{cases} 
0, & \text{if } \frac{K_{ij}(u_i) - (K_{thr,ij} - \alpha_{ij})}{2\alpha_{ij}} < 0 \\
\frac{K_{ij}(u_i) - (K_{thr,ij} - \alpha_{ij})}{2\alpha_{ij}}, & \text{if } 0 \leq \frac{K_{ij}(u_i) - (K_{thr,ij} - \alpha_{ij})}{2\alpha_{ij}} \leq 1, \\
1, & \text{if } \frac{K_{ij}(u_i) - (K_{thr,ij} - \alpha_{ij})}{2\alpha_{ij}} > 1
\end{cases} \quad (11) $$

where $K_{thr,ij}$ is the “soft threshold” to be identified and $\alpha_{ij}$ is a constant. $K_{ij}(u_i)$ is the interested physical variable for the transition. $K_{ij}(u_i)$ for different mode transitions are shown in Table 2.

<table>
<thead>
<tr>
<th>Mode from\Mode to</th>
<th>Acc.</th>
<th>Cruise</th>
<th>Brake</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acc.</td>
<td>--</td>
<td>$v - v_i$</td>
<td>$v^2 / d$</td>
</tr>
<tr>
<td>Cruise</td>
<td>$v - v_i$</td>
<td>--</td>
<td>$v^2 / d$</td>
</tr>
<tr>
<td>Brake</td>
<td>$v^2 / d$</td>
<td>$v - v_i$</td>
<td>--</td>
</tr>
</tbody>
</table>

The transition probability evaluation function is in the following form,
\[ T_{ij}(f_{ij}) = \frac{\arctan[\beta_{ij}(f_{ij} - 0.5)]}{2 \arctan(0.5 \beta_{ij})} + 0.5, \]  

(12)

where \( \beta_{ij} \) is a constant parameter to be identified. This function generates a smooth probability change as \( f_{ij} \) grows from 0 to 1, and the parameter \( \beta_{ij} \) can be used to adjust the shape of this function. Figure 8 shows the functions under different \( \alpha_{ij} \)’s and \( \beta_{ij} \)’s. It is assumed that \( K_y(u_i) = v \) and \( K_{thr,ij} = 15 \). The transition will occur when the vehicle speed is around 15m/s and the probability increases as the speed grows. When \( \beta_{ij} \) is large, the transition probability evaluation function curve is steep, which means that the transition will very likely occur just when \( f_{ij} \) hits the steep change. When \( \beta_{ij} \) is small, the curve is gentle, which means that the transition has larger probability to occur earlier or later. \( T_{ij} \) determines the element \( \phi_{ji} \) of \( \Phi(u_i) \). \( \alpha_{ij} \) and \( \beta_{ij} \) can show how “consistent” the driver is in terms of state transitions. In general, larger \( \alpha_{ij} \) and smaller \( \beta_{ij} \) mean that the driver is less consistent and the state transition is harder to predict.

![Figure 8. Transition factor function and transition probability evaluation function examples [59]](image-url)
2.5.2. Output Probability Distribution Functions

The most important part of output probability distribution function, which is \( f_{\text{out}}(u_i, x_i) \) in (9), is designed in three steps. The first step is to find the relationship between the probability of not taking action \( p_0 \) and inputs in each mode. The second step is to find the relationship between the probability of completely releasing the pedal \( p_r \) and inputs in each mode. The last step is to find the relationship between possible action value, which is a normal distributed random variable determined by its mean value \( \mu \) and standard deviation \( \sigma \), and inputs in each mode. All these steps are static model fitting problems therefore various statistics tools can be used to solve these problems and evaluate the performance of the models.

It should be noted that the goal of these analysis steps is to find the structure of \( f_{\text{out}}(u_i, x_i) \). After these steps, the parameters of the output probability distribution function will still be identified using GEM method. Multiple drivers’ data are analyzed in these steps and the best possible model that balances the complexity and performance is selected. It can also be assumed that the input-output data sequence used in this section all have known states, as we can manually select some data which have obvious driving mode meanings for the analysis, e.g., in the middle of a long cruise section, or at the beginning of a long acceleration section. Then the three-step analysis can be conducted for each mode separately.

The first two steps can be done via a logistic regression [65], which can be used to build a model for a dichotomous response variable with possible values 1 and 0. The logit function is defined as
\[ P(x) = \frac{e^{x^T g(x)}}{1 + e^{x^T g(x)}}, \]  
where \( g(x) \) is usually a linear function of the input vector \( x \).

In the first step, for a set of data with known state, the input of the logistic regression is all the input of the IOHMM \( u_t \), the output of the logistic regression is defined as

\[ y_{p_0,j} = \begin{cases} 
1, & y_i = 0 \\
0, & y_i \neq 0
\end{cases} \]  
where \( y_i \) is the actual human action. Logistic regression can be conducted via the Statistics and Machine Learning Toolbox in Matlab. Models with interaction and quadratic predictors are allowed in this regression model design. Some other terms with physical meanings, e.g., \( v^2/(2d) \) that is the expected constant deceleration to the stop, is also included and tested in the model. After considering multiple drivers’ data, evaluating the model performance through coefficients of multiple determination and adjusted coefficients of multiple determination [65], conducting necessary statistic tests, and balancing the computation load and performance, the final \( g_{p_0}(u) \) equations for \( p_0 \) is shown in Table 3. \( \Theta_{p_0} \)'s are row vectors with proper dimension whose elements are to be identified. By combing \( g_{p_0}(u) \) in Table 3 and (13), the equations for \( p_0 \) can be obtained.
Table 3. Equations for no-action probabilities in all modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acc.</td>
<td>$g_{p_0}(u) = \Theta^\text{acc}_{p_0} [v \ p\ a\ va\ vp\ v^2]^\top$</td>
</tr>
<tr>
<td>Cruise</td>
<td>$g_{p_0}(u) = \Theta^\text{cru}_{p_0} [(v - v_i) \ a\ p\ (v - v_i)p\ ap]^\top$</td>
</tr>
<tr>
<td>Brake</td>
<td>$g_{p_0}(u) = \Theta^\text{brk}_{p_0} [\frac{v^2}{d_m} \ a\ p\ \frac{v^2a}{d_m}\ ap\ \frac{a}{d_m} (\frac{v^2}{d_m})^2]^\top$ [where\ d_m = \max(d, 0.1)]</td>
</tr>
</tbody>
</table>

Similarly, $p_r$ can be determined by this method. Statistical analysis shows that it is only necessary to introduce $p_r$ in cruise mode, and the equations are shown in Table 4.

Table 4. Equations for completely-release probabilities in all modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acc.</td>
<td>$g_{p_r}(u) = 0$</td>
</tr>
<tr>
<td>Cruise</td>
<td>$g_{p_r}(u) = \Theta^\text{cru}_{p_r} [1 \ (v - v_i) \ a]^\top$</td>
</tr>
<tr>
<td>Brake</td>
<td>$g_{p_r}(u) = 0$</td>
</tr>
</tbody>
</table>

After removing the no-action point and completely-pedal-release point, the input-output sequence data in each mode can be analyzed using multiple regression. The final model equations for $\mu$ are shown in Table 5.
Table 5. Equations for mean output values (except no-actions) in all modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acc.</td>
<td>$\mu(u) = \Theta_{\mu}^{acc} [1 \ v \ p \ a \ va \ vp \ v^2]^T$</td>
</tr>
<tr>
<td>Cruise</td>
<td>$\mu(u) = \Theta_{\mu}^{cru} [1 \ (v - v_i) \ a \ p \ (v - v_i) p \ ap]^T$</td>
</tr>
<tr>
<td>Brake</td>
<td>$g_{\mu}(u) = \Theta_{\mu}^{brk} [1 \ \frac{v^2}{d_m} \ a \ p \ v \ \frac{v^2 a}{d_m} \ ap \ av \ \frac{(v^2)^2}{d_m}]^T$</td>
</tr>
</tbody>
</table>

where $d_m = \max(d, 0.1)$

The standard deviation is assumed to be a constant in each mode, as shown in Table 6.

Table 6. Standard deviations in all modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acc.</td>
<td>$\sigma = 6$</td>
</tr>
<tr>
<td>Cruise</td>
<td>$\sigma = 6$</td>
</tr>
<tr>
<td>Brake</td>
<td>$\sigma = 2$</td>
</tr>
</tbody>
</table>

All the $\Theta$’s are the parameters to be identified for each individual driver. After the identification and performance evaluation (the measure of model performance will be discussed later), interestingly the standard logistic regression (13) does not show as good performance as using the following alternative equation,

$$P(x) = \begin{cases} 0, & \text{if } 1 - e^{g(x)} < 0 \\ 1 - e^{g(x)}, & \text{if } 0 \leq 1 - e^{g(x)} \leq 1. \\ 1, & \text{if } 1 - e^{g(x)} > 1 \end{cases}$$

(15)

In this section, the logistic regression is only used to find the inputs and interaction of inputs that can affect the corresponding probabilities. Since using (15) in the model
and conducting parameter identification based on this equation can provide better performance, logistic regression equation (13) is replaced by (15) in the final model.

2.5.3. Model Parameter Identification

With the state transition described in (11)(12) and Table 2, output probability distribution function described in (9)(10)(15), Table 3, Table 4, Table 5 and Table 6 the IOHMM is completed and the corresponding dynamic and output functions in the form of (7) and (8) can be constructed.

The resultant model has 49 parameters. The parameter identification can be conducted using GEM. The Multistart function in Matlab Global Optimization Toolbox is used to find the global optimal in the maximization step in each iteration of GEM. The initial guess of the parameters has a great impact on the GEM computational time. Many other factors including the complexity of road environment and driving maneuvers as well as the driver behaviors’ consistency may also affect the convergence rate of GEM. In general, the parameter identification computation process takes about 2 to 10 hours (depending on the initial guess of parameters) on a personal desktop computer using 20 minutes driving data. After the identification, the model can be easily implemented online for real-time applications.

2.6. Model Performance

The performance of the proposed IOHMM-based driver pedal behavior model will be evaluated in this section. One-step probability prediction test can show that the
proposed model can capture different driver’s driving styles, and the multi-step closed-loop prediction test can demonstrate the prediction capability of the model.

2.6.1. One-Step Probability Prediction Test

According (7) and (8), given input $u_t$ and previous state $x_{t-1}$, or the probability of state $\zeta_{t-1}$, the proposed IOHMM model can calculate the probability of the action that the driver may take at the moment, thus predict the probability of the pedal position at the next sampling time. In this one-step probability prediction test, at step $t$, all the inputs including the road information, vehicle information and current pedal position are fed into the proposed model. An estimation of $\zeta_{t-1}$ is also made. The model can calculate an output probability distribution $\eta_t = [\eta_{1,t}, \eta_{2,t}, ..., \eta_{M,t}]^T$. If the index of the actual pedal action that the human driver took at this step is $i_a$, the probability of making a correct prediction would be $\eta_{i_a,t}$. Sometimes the driver may make a minor adjustment on the pedal position, which results in a small action very close to zero action. So in the evaluation, a prediction is considered “correct” if the actual human action is within $\pm 1\%$ of the predicted value. Then the adjusted correct probability is defined as,

$$\eta^*_{i_a,t} = \max(\eta_{i_{a-1},t}, \eta_{i_a,t}, \eta_{i_{a+1},t}).$$  

(16)

For a given time interval from $t = 1$ to $T = 1$, the product of all the probability of correct action prediction,

$$V = \prod_{t=1}^{T} \eta^*_{i_a,t},$$  

(17)
can be used to evaluate the performance of the model. The greater $V$ is, the better the model performs.

If the time interval is long, $V$ can be a very small number. So in this work,

$$
\log V = \log(\prod_{t=1}^{T} \eta_{t,i}^{*}) = \sum_{t=1}^{T} (\log \eta_{t,i}^{*}),
$$

(18)

will be used as the performance index.

In order to calculate $\eta_i$, $\zeta_{t-1}$ needs to be known. In the one-step prediction test, $\zeta_{t-1}$ is estimated using the following method: first, make a guess of a determined mode at step $t-4$, then $\zeta_{t-4}$ can be assumed to be known. Based on this guess and the proposed model, calculate $\zeta_{t-3}$, then $\zeta_{t-2}$, and finally $\zeta_{t-1}$ iteratively.

The data shown in Figure 9 are the driving data from a human driver on the driving simulator. Figure 10 shows the one-step probability prediction test result from 84th second to the 87th second with the aforementioned $\zeta_{t-1}$ estimation method. The x-axis value of the cross is the actual action the human driver takes. Table 7 shows state transition probability examples with the assumption that the current state is acceleration mode. The probability of transiting to cruise mode increases as the vehicle speed gets close to the road speed limit, and the probability of transiting to braking mode increases as the vehicle approaches a required stop.
Figure 9. Driving data from a human driver [59]

Figure 10. The output probability distribution result examples from one-step prediction test [59]
Table 7. The state transition probability examples

<table>
<thead>
<tr>
<th>Time(s)</th>
<th>Acc.</th>
<th>Cruise</th>
<th>Brake</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>102</td>
<td>0.9686</td>
<td>0.0314</td>
<td>0.000</td>
</tr>
<tr>
<td>103</td>
<td>0.9320</td>
<td>0.0680</td>
<td>0.000</td>
</tr>
<tr>
<td>104</td>
<td>0.9628</td>
<td>0.000</td>
<td>0.0372</td>
</tr>
<tr>
<td>105</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>

Four sets of driver model parameters are identified for four drivers. The data length used for identification varies from 5 to 15 minutes for each driver. Each data set contains 10 to 24 stops. All four driver models are applied to four comprehensive 100-second-long datasets from these four drivers. These 100-second-long datasets contain both local and highway road driving, with multiple stops. They are not included in the data used for model identification. $\log V$ can be calculated from the model output probability distributions and the actual driver actions. Table 8 shows the $\log V$ results. The bold numbers are the best results for each driver’s driving data. It can be seen that, for Drivers A, B and D, the driver model identified using the specific driver’s data performs better than the models from other drivers. For Driver C, the identified model is not the best, but it is still very close to the best result. It should also be noted that Driver C’s data have the largest $\log V$ absolute value, which means his or her driving behavior is more random and more difficult to predict than the other drivers. This result means that in general it is necessary to conduct parameter identification for each individual driver and the proposed model can capture the individual driving style with its identification method.
Table 8. \( \log V \) for each driver model applied to each driver’s driving data

<table>
<thead>
<tr>
<th>Model\Data</th>
<th>Driver A</th>
<th>Driver B</th>
<th>Driver C</th>
<th>Driver D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver A</td>
<td>-280.2</td>
<td>-341.5</td>
<td>-385.0</td>
<td>-339.3</td>
</tr>
<tr>
<td>Driver B</td>
<td>-293.7</td>
<td>-295.5</td>
<td>-388.6</td>
<td>-328.4</td>
</tr>
<tr>
<td>Driver C</td>
<td>-319.3</td>
<td>-375.3</td>
<td>-385.4</td>
<td>-368.0</td>
</tr>
<tr>
<td>Driver D</td>
<td>-302.5</td>
<td>-342.2</td>
<td>-399.0</td>
<td>-326.4</td>
</tr>
</tbody>
</table>

2.6.2. Multi-Step Closed-Loop Prediction Test

In the multi-step prediction, a vehicle-road model and a random number generator is necessary for the prediction. The proposed IOHMM-based model can only provide the output action probability distribution for one step. Then a random number is needed to translate the probability distribution into one determined action. This new pedal position is then fed into the vehicle-road model. The vehicle model is essentially the same as the model running in the driving simulator, except that the update frequency is lower at 10 Hz, and the road model is one-dimensional speed limits and stop position data. This vehicle-road model will update the input to the IOHMM model, such as vehicle speed, distance to the stop, etc. The IOHMM model still updates every one second. In this closed-loop way, the proposed model will be able to make multi-step prediction within one step. The input information to the model is all current input including road, vehicle and pedal information, and future road information such as position of the stop sign and speed limit. As a random number will be used for each prediction, the performance will be dependent on this number. Therefore, the model will be run for multiple times, and the best, worst, and average results of the proposed model will all be shown.

The following three other models are used for comparison.
1) A constant pedal position model. This model does not use any input information and assumes that the driver pedal position will be a constant for the near future. It can be used in some powertrain model predictive control applications. It is also a benchmark to test whether a model is making effective predictions.

2) An ARX model. In real-time the input and output of 3 previous steps are identified as an ARX model and the output of current step is predicted using this ARX model. This model considers each individual’s driving style and can be used in some extreme-short-term predictions (e.g., the sampling period is 0.01 second and the prediction horizon is 0.5 second in [60]).

3) A proportional-integral-derivative (PID) driver model. A reference speed is generated using all the road information with moderate acceleration, constant-speed cruise, and moderate brake. Then a PID controller is designed to track the reference speed. The input to the PID controller is the speed difference, and the output of the PID controller is the driver pedal position. This model can be used in some long-term prediction situations.

The input data used in this test are the first 100 second data shown in Figure 9. The data up to the 160th second are used for comparing the predicted pedal position and the actual pedal position. Figure 11 and Figure 12 are two examples of the model’s prediction output with a random number generator.
The standard error of estimate (SEE) is used to evaluate the models. SEE for pedal position is defined as

\[
S = \sqrt{\frac{\sum_{t=1}^{T} \sum_{h=1}^{H} (p_{act,t+h} - p_{prd,t,h})^2}{TH}},
\]  

(19)

Figure 11. 10-second IOHMM-based driver pedal model prediction at \( t = 97s \) [59]

Figure 12. 60-second IOHMM-based driver pedal model prediction at \( t = 56s \) [59]
where $p_{act,t+h}$ is the measured pedal position at step $t+h$, $p_{prd,t+h}$ is the predicted pedal position of step $t+h$, made at step $t$, $T$ is the length of the data used for the test, and $H$ is the prediction horizon. Similarly, SEE of speed can also be defined.

Table 9, Table 10, and Figure 13 display the performances of the proposed driver model and comparison models in different prediction horizons. The IOHMM-based model has been run for 50 times with different random numbers, and the minimum, average, and maximum SEE results of the 50 runs are shown. In the tables, The bold numbers are the best model results for each horizon (the average SEE is considered as the result of the IOHMM-based model). Similarly, in Figure 13, the average SEE is considered as the result of IOHMM-based mode. The two unmarked green solid lines close to the IOHMM result are the minimum and maximum SEE results of IOHMM. In one-step prediction, the constant pedal model performs the best, even slightly better than the proposed IOHMM-based model. This result is consistent with the statement in Section 2.2.2 that the driver’s adjustment to the pedal is intermittent and for most of the time the driver is not taking any action. But obviously the constant pedal model cannot perform well in long-term prediction. ARX models are not suitable for pedal position prediction in these time scales. It cannot provide good prediction results even if its parameters are identified in real-time. The PID model performs well only in the long-term prediction, as the road environment constraints will become the dominant factors for driver behavior and PID model is the only model among the three comparison models that can take into account the road information. The proposed IOHMM-based driver
model can make good predictions both in short-term and long-term. It has the lowest SEE values in all prediction horizons except the 1-second case.

Table 9. Pedal position SEE (%) of models in different prediction horizons

<table>
<thead>
<tr>
<th></th>
<th>Const.</th>
<th>ARX</th>
<th>PID</th>
<th>IOHMM-based model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min.</td>
<td>Aver.</td>
<td>Max.</td>
<td></td>
</tr>
<tr>
<td>1 step</td>
<td>23.3</td>
<td>44.6</td>
<td>99.4</td>
<td>21.1</td>
</tr>
<tr>
<td>5 steps</td>
<td>39.0</td>
<td>82.4</td>
<td>53.4</td>
<td>25.3</td>
</tr>
<tr>
<td>10 steps</td>
<td>46.3</td>
<td>83.5</td>
<td>48.6</td>
<td>26.5</td>
</tr>
<tr>
<td>15 steps</td>
<td>48.6</td>
<td>83.6</td>
<td>45.5</td>
<td>29.2</td>
</tr>
<tr>
<td>20 steps</td>
<td>48.2</td>
<td>82.1</td>
<td>43.3</td>
<td>31.9</td>
</tr>
<tr>
<td>30 steps</td>
<td>49.6</td>
<td>81.3</td>
<td>44.2</td>
<td>35.3</td>
</tr>
<tr>
<td>60 steps</td>
<td>53.0</td>
<td>78.7</td>
<td>49.4</td>
<td>44.5</td>
</tr>
</tbody>
</table>

Table 10. Speed position SEE (m/s) of models in different prediction horizons

<table>
<thead>
<tr>
<th></th>
<th>Const.</th>
<th>ARX</th>
<th>PID</th>
<th>IOHMM-based model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min.</td>
<td>Aver.</td>
<td>Max.</td>
<td></td>
</tr>
<tr>
<td>1 step</td>
<td><strong>0.498</strong></td>
<td>0.498</td>
<td>1.64</td>
<td>0.498</td>
</tr>
<tr>
<td>5 steps</td>
<td>2.21</td>
<td>4.82</td>
<td>2.60</td>
<td>1.52</td>
</tr>
<tr>
<td>10 steps</td>
<td>5.08</td>
<td>7.30</td>
<td>3.42</td>
<td>2.46</td>
</tr>
<tr>
<td>15 steps</td>
<td>7.22</td>
<td>10.4</td>
<td>4.11</td>
<td>2.89</td>
</tr>
<tr>
<td>20 steps</td>
<td>8.52</td>
<td>13.3</td>
<td>4.42</td>
<td>3.31</td>
</tr>
<tr>
<td>30 steps</td>
<td>10.4</td>
<td>17.7</td>
<td>5.11</td>
<td>4.13</td>
</tr>
<tr>
<td>60 steps</td>
<td>14.4</td>
<td>23.6</td>
<td>6.56</td>
<td>5.95</td>
</tr>
</tbody>
</table>
In the 60s long-term prediction cases, the SEE of pedal position can be as large as 47.0%. But when comparing with the constant model SEE benchmark, it still outperforms the benchmark 53.0% by a considerable margin, which means that making a prediction using the proposed model for such a long horizon is still better than making no efforts in prediction at all.

Multi-step closed-loop prediction test can also show that each driver’s driving style is captured by the model. Table 10 shows the average pedal SEE of applying different driver models to Driver A’s data. Each mode has been run 50 times with the random number generator. The bold numbers are the best model results for each horizon. It can be seen that in general the model with parameters specifically identified for Driver A performs the best for most prediction horizons. In the 10-step, 15-step and 20-step tests,
the Driver A’s model does not perform the best, but Driver A’s model SEE values are very close to the best ones in these three cases. These results indicate that the identification for individual drivers can still bring benefits in the multi-step pedal position predictions.

Table 11. Average pedal position SEE (%) of different models in different prediction horizons, tested with Driver A’s data

<table>
<thead>
<tr>
<th></th>
<th>Driver A</th>
<th>Driver B</th>
<th>Driver C</th>
<th>Driver D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 step</td>
<td>23.8</td>
<td>25.6</td>
<td>25.6</td>
<td>24.2</td>
</tr>
<tr>
<td>5 steps</td>
<td>26.7</td>
<td>27.4</td>
<td>28.0</td>
<td>26.8</td>
</tr>
<tr>
<td>10 steps</td>
<td>28.0</td>
<td>27.6</td>
<td>28.4</td>
<td>30.5</td>
</tr>
<tr>
<td>15 steps</td>
<td>31.5</td>
<td>31.4</td>
<td>31.1</td>
<td>35.3</td>
</tr>
<tr>
<td>20 steps</td>
<td>33.6</td>
<td>34.9</td>
<td>33.2</td>
<td>38.4</td>
</tr>
<tr>
<td>30 steps</td>
<td>37.6</td>
<td>40.5</td>
<td>38.1</td>
<td>41.9</td>
</tr>
<tr>
<td>60 steps</td>
<td>44.7</td>
<td>49.7</td>
<td>46.6</td>
<td>50.9</td>
</tr>
</tbody>
</table>

2.7. Summary

In this Chapter, an IOHMM-based driver pedal behavior model is proposed. This model can incorporate the road information, capture individual driver’s driving style, and predict the output action probability distribution. With a random number generator, the proposed model can make a determined prediction of future driver pedal behavior. By comparing with driving simulator human subject data, the model results show that it can reflect the individual driver’s driving style and performs well in prediction horizons from 1 second to 60 seconds.
Chapter 3: Worst-Case Relative Cost Optimal Control

Classic optimal control methods usually require complete information about the dynamic systems known a priori, which means that for a system with measurable disturbances, the optimal control and minimum cost cannot be calculated until the complete disturbance information is known. In some applications including hybrid electric vehicle energy management systems, the cost values may be highly dependent on the disturbances, if the power demand from driver is considered as generalized disturbances. In these systems, designing a controller which gives a minimum upper bound of the absolute cost may be of less interest than a controller bounding the relative cost, which is defined as the ratio between the actual cost and the posteriori calculated optimal cost. This chapter proposes a worst-case relative cost (WCRC) optimal control method for discrete-time nonlinear dynamic systems, assuming that the disturbance sequence belongs to a known finite admissible set. The control policy uses the current states, current accumulated cost, and all current and past disturbance values as feedback, and can achieve a minimum relative cost in the worst disturbance sequence case. Then the problem is extended to a stochastic case, where the probability range of each disturbance sequence is known, and the worst-case expected relative cost is minimized. Two numerical examples are provided to illustrate the proposed method. This chapter is mainly based on the paper [66] which is submitted to a journal and under review.
3.1. Optimal Control for Systems with Disturbances

The finite horizon optimal control problem for systems with disturbances can be solved via classic optimal control methods, only if the complete disturbance information during the horizon is known a priori. For a system with measurable but unpredictable disturbance, its optimal control policy over a finite-time process cannot be obtained until the process is completed. Some literatures focus on the optimal control of systems with uncertain future information. Robust model predictive control can be used on control problems with uncertain disturbances subject to constraints [67][68][69]. Its robustness usually focuses on stability, not optimality. Stochastic model predictive control can also be applied to systems with uncertainties and the cost function to be optimized can be either the expected cost, the worst-case cost, or the nominal cost [69][70]. Guaranteed-cost optimal control can obtain an upper bound of the cost by using the idea similar to common Lyapunov functions [71]. For linear systems, guaranteed cost optimal control methods are developed based on many variations of generalized Riccati equations [71][72][73][74][75][76]. Min-max (or minimax) optimal control [77][78][79][80] considers the worst-case scenario and gives the minimum upper bound of the cost. Dynamic game theory can also be applied to solve the min-max optimal control problems [81].

The performance index of all these abovementioned control methods is the absolute cost value, which is defined as the sum of the running cost at every step and a terminal cost. However, in systems with unknown future demands, for example, the hybrid electric
vehicle systems in which the future driver's torque demand is uncertain, or the building
temperature control systems in which the future weather condition is uncertain, the
energy cost is highly dependent on the uncertain demand itself. In these systems with
uncertain future disturbances, a controller that only concentrates on giving a minimum
upper bound of the absolute cost is not necessarily the suitable controller because even if
a guaranteed absolute cost upper bound is achieved, the performance may be very poor
when the actual power demand is relatively low. A better controller in these scenarios
should lead to a cost that is close to the posteriori optimal cost in all situations. This goal
cannot be achieved by only minimizing the upper bound of the absolute cost with respect
to the uncertain future information. The relative cost, which is defined as the ratio of the
actual cost and the posteriori optimal cost, would be a better performance index for these
applications. This concept is similar to that of the competitive analysis in computer
science, where algorithms are evaluated by comparing their results with the posteriori
optimal ones [82]. But competitive analysis only provides an evaluation standard to
analyze existing algorithms. It is not a tool to construct algorithms. The method proposed
in this chapter can construct an optimal control algorithm for a class of control problems,
which is a \textit{c-approximation algorithm} in the language of competitive analysis and the
competitive ratio \( c \) is minimized. The relative cost has also been widely used to assess the
performances of different control strategies in control applications [56][57][61], but
theoretical investigation has been rarely conducted.

This study will focus on designing a controller to achieve a minimum worst-case
relative cost when there are only a finite number of admissible disturbance sequences
over the horizon, i.e., the disturbance sequence is assumed to belong to a known finite set. First, it is assumed that we have no knowledge about the probability of each disturbance sequence. A worst-case relative cost optimal control policy is designed to minimize the worst-case relative cost over the horizon using all the available information up to the current step despite the uncertain future. It should be noted that because the relative cost is defined differently from the conventional accumulative cost, the Principle of Optimality which is the fundamental of the dynamic programming method cannot be directly applied to this problem. In this chapter, it is proved that a backward induction method can be used to generate the WCRC optimal control policy if the accumulated cost is included as an extra state. Then, the proposed WCRC optimal control method is extended to the stochastic case that the probability of each disturbance sequence can be estimated a priori, but instead of the exact value of each probability, only a rough range of each probability can be estimated (which is usually the case in practice: the accurate probability is very difficult to obtain, but statistics analysis may provide a range of the probability). The expected WCRC optimal control is designed, in which “worst-case” means that the event probability is the worst one within the known range.

3.2. WCRC Problem Formulation

3.2.1. System Dynamics

In this chapter, the system considered is the following discrete-time nonlinear system,

\[ x_{k+1} = f_k(x_k, u_k, w_k), \]  

(20)
where \( x_k \in \mathcal{X}, u_k \in \mathcal{U}, w_k \in \mathcal{W}, k \in \mathbb{N} \), and \( k \in \mathcal{K} \), with a given initial condition \( x_0 \).

### 3.2.2. Disturbance Sequences

The disturbances \( w_k \) from step 0 to step \( N-1 \) form a sequence \( s = \{w_0, w_1, ..., w_{N-1}\} \).

In the problem considered in this note, it is assumed that \( s \in \mathcal{S} \) where \( \mathcal{S} \) is a finite set which is known a priori. \( \mathcal{S} \) is called the finite admissible disturbance sequence set. An index number is assigned to each admissible disturbance sequence in the set \( \mathcal{S} \).

Therefore \( \mathcal{S} \equiv I_0 = \{1, 2, ..., M\} \) and \( M \) is a finite number. Each \( s_i \) is an admissible disturbance sequence and \( i \) is called the index of a disturbance sequence. \( I_0 \) is the initial admissible disturbance sequence index set.

The disturbance value \( w_k \) is assumed to be measured at every step \( k \). \( \mathcal{S} \) and the past disturbance sub-sequence can be used to extract some information about the future disturbance. Denote the \( j \)-th element of a specific disturbance sequence \( s_i \) as \( s_i(j) \). At step \( k \), define the admissible disturbance sequence index set \( I_k \) as

\[
I_k = \{ i \in I_0 | s_i \in \mathcal{S} | w_0, s_i(2) = w_1, ..., s_i(k+1) = w_k \} .
\]  

(21)

\( I_k \) is the set of all indices of the disturbance sequences that are still admissible at step \( k \).

For every process whose disturbance sequence belongs to the known finite admissible disturbance sequence set, \( I_k \) is always a “shrinking” set as \( k \) increases. The
last set $I_{N-1}$ contains only one element, whose index is denoted as $i_r$. This shrinking property can be described as

$$I_0 \supseteq I_1 \supseteq \ldots \supseteq I_{N-1} = \{i_r\}. \quad (22)$$

### 3.2.3. Control Policy and Cost Function

Define a control policy as $\pi = \{u_0(\cdot), u_1(\cdot), \ldots, u_{N-1}(\cdot)\}$, where $u_k(\cdot)$ is a time-varying feedback control law. Given the system dynamics in (20), the state trajectory is denoted as $x_k = x_k(x_0, \pi, s)$.

The accumulative absolute cost is defined as the sum of the running cost $g_k$ at every step and a terminal cost $g_N$,

$$J(x_0, \pi, s) = \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) + g_N(x_N). \quad (23)$$

When $s$ is known in advance, $J^*(x_0, s) = \min_{\pi} J(x_0, \pi, s)$ is called the optimal absolute cost and the corresponding control policy is called the absolute optimal control policy. In this problem, it is assumed that $J^*(x_0, s) > 0$.

In the context of absolute cost, define the cost-to-go function as the remaining cost from step $k$ to the end as

$$V_k(x_k, \pi, s) = \sum_{j=k}^{N-1} g_j(x_j, u_j, w_j) + g_N(x_N). \quad (24)$$

The relative cost $R$ is defined as the ratio between the accumulative absolute cost under a certain control policy $\pi$ and the optimal absolute cost,
\[ R(x_0, \pi, s) = \frac{J(x_0, \pi, s)}{J^*(x_0, s)}. \]  

(25)

Obviously, the minimum value of \( R \) is 1, and it can be achieved via optimal control method if \( s \) is known. But in the problem here, \( s \) is generally unknown when the controller starts to run. The objective is to design a good control policy \( \pi \) such that the relative cost \( R(x_0, \pi, s_i) \) is close to 1 for any \( i \in I \).

In the presence of uncertainties in disturbance sequence, the worst-case relative cost is defined as

\[ W(x_0, \pi, I) = \max_{i \in I} R(x_0, \pi, s_i) = \max_{i \in I} \frac{J(x_0, \pi, s_i)}{J^*(x_0, s_i)}. \]  

(26)

If a control policy \( \pi \) can minimize the worst-case relative cost, it can provide a guaranteed performance in terms of cost despite that the future disturbance sequence is uncertain.

3.2.4. The Principle of Optimality and WCRC Problem

The Principle of Optimality, which is “an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision” [83], cannot be directly applied to minimize \( W(x_0, \pi, I) \), because the cost function here is substantially different from the conventional accumulative cost. However, if the system is augmented with one more state, which is the accumulative absolute cost up to the current step, a new theorem can be proved to guarantee that the WCRC problem can be solved via a
backward induction. The augmented WCRC optimal control problem is formulated as follows.

**Problem 1**: For the augmented system \( x_{k+1} = f_k(x_k, u_k, w_k), \) \( z_{k+1} = g_k(x_k, u_k, w_k) + z_k \), with a given initial condition \( x_0, z_0 = 0 \), an indexed finite admissible disturbance sequence set \( S \) \( \in I_0 \), and an initial admissible disturbance sequence index set \( I_0 \), find a feedback control policy \( \pi = \{u_0(x_0, z_0, I_0), u_1(x_1, z_1, I_1), \ldots, u_{N-1}(x_{N-1}, z_{N-1}, I_{N-1})\} \), such that the worst-case relative cost \( W(x_0, \pi, I_0) = \max_{i \in I_0} J(x_0, \pi, s_i) \) is minimized.

The truncated augmented WCRC optimal control problem is formulated as follows.

**Problem 2**: Given \( x_0 \), the indexed finite admissible disturbance sequence set \( S \) \( \in I_0 \), the initial condition of the truncated problem \( x_k, z_k, I_k \), and \( k \), find a feedback control policy \( \pi_k = \{u_{k,0}(x_k, z_k, I_k), u_{k,1}(x_{k+1}, z_{k+1}, I_{k+1}), \ldots, u_{k, N-1}(x_{N-1}, z_{N-1}, I_{N-1})\} \), such that \( Y_k(x_k, z_k, \pi_k, I_k, x_0) = \max_{i \in I_k} \frac{z_k + V_k(x_k, \pi_k, s_i)}{J^*(x_0, s_i)} \) is minimized.

\( Y_k \) is essentially the worst-case relative cost on the interval from step 0 to the end, if the trajectory and the cost before step \( k \) is already determined, and the disturbance sequence belongs to the current admissible disturbance sequence set \( I_k \), while a control policy \( \pi_k \) is applied to the rest of the process.

Several things should be noted in this problem formulation. First, although \( w_k \) is not explicitly included in the feedback of \( u_k(\cdot) \), the knowledge of \( I_k \) implies that \( w_k = s_i(k+1), \forall i \in I_k \). So the current step disturbance value is considered in the
feedback of control policy. Second, in the truncated problem, $u_k(\cdot)$ is only defined for some “feasible” $I$. It is not defined for any $I$ such that it contains a disturbance sequence that is not included in $I_k$. Third, the WCRC optimal control policy may be nonunique and is not necessarily the best control policy when the disturbance sequence is not in the worst case. It only guarantees that when the disturbance sequence is not in the worst case, the relative cost is less than the worst-case relative cost, and a minimum common upper bound of relative cost is found for all cases. For the rest of this chapter, it is assumed that the WCRC optimal control policy exists and an approach of constructing such a policy will be presented.

3.3. Solution of WCRC Problem

3.3.1. A Key Theorem

A key theorem will be proposed to connect the solution of the truncated WCRC optimal control problem and the solution of the original WCRC optimal control problem.

**Theorem 1:** If $\pi^* = \{u_j^*(x_j, z_j, I_j), j \in \{0,1,...,N-1\}\}$ is a WCRC optimal control policy of the original WCRC optimal control problem, $\pi_k^* = \{u_{k,j}^*(x_j, z_j, I_j), j \in \{k,k+1,...,N-1\}\}$ is a WCRC optimal control policy of the corresponding truncated WCRC optimal control problem (which means $\pi_j^*$ minimizes $Y_k(x_k, z_k, x_0)$, and the initial condition of the truncated problem satisfies $x_k = x_k(x_0, \pi^*, s_j)$, $z_k = z_k(x_0, z_0, \pi^*, s_j)$, $\forall i \in I_k$. A new control policy $\tilde{\pi}$ is constructed by replacing $u_j^*(x_j, z_j, I_j)$ with $u_{k,j}^*(x_j, z_j, I_j)$ when possible, which means,
\[ \tilde{r}_{j}^{*}(x_{i}, z_{j}, I_{j}) = u_{k,j}^{*}(x_{j}, z_{j}, I_{j}), \text{ if } (x_{j}, z_{j}, I_{j}) \text{ is defined for } u_{k,j}^{*}(\cdot); \]

\[ \tilde{r}_{j}^{*}(x_{i}, z_{j}, I_{j}) = u_{j}^{*}(x_{j}, z_{j}, I_{j}), \text{ if } (x_{j}, z_{j}, I_{j}) \text{ is not defined for } u_{k,j}^{*}(\cdot), \text{ but defined for } u_{j}^{*}(\cdot). \]

Then \( \tilde{r} \) is also a WCRC optimal control policy of the original WCRC optimal control problem.

Theorem 1 essentially means that if a WCRC optimal control policy of the original WCRC optimal control problem is overlapped by a WCRC optimal control policy of the truncated problem, it remains an optimal control policy of the original WCRC optimal control problem.

Before proving the theorem, a lemma will be introduced.

**Lemma 1:** If \( i \not\in I_{k}, J(x_{0}, \tilde{r}_{j}^{*}, v_{0}, \pi^{*}, s_{j}). \)

This lemma implies that when overlapping the optimal control policy of the original WCRC optimal control problem with that of a truncated problem starting at step \( k \), only the cases whose disturbance sequence is still admissible at the step \( k \) will be affected.

**Proof of Lemma 1:** The “shrinking” property implies that \( u_{k,j}^{*}(x_{j}, z_{j}, I_{j}) \) is only defined for \( I_{j} \subset I_{k} \). If \( i \not\in I_{k}, u_{k,j}^{*}(x_{j}, z_{j}, I_{j}) \) is not defined for any \( I_{j} \) that contains \( i \), so \( \tilde{r}_{j}^{*}(x_{i}, z_{j}, I_{j}) = u_{j}^{*}(x_{j}, z_{j}, I_{j}) \) for all \( I_{j} \) as long as \( I_{j} \) contains \( i \). Therefore \( J(x_{0}, \tilde{r}_{j}^{*}, v_{0}, \pi^{*}, s_{j}). \)

**Proof of Theorem 1:**
\[
\max_{i \in I_0} \frac{J(x_0, \xi)}{J^*(x_0, s_i)}
= \max \{ \max_{i \in I_k} \frac{J(x_0, \xi)}{J^*(x_0, s_i)}, \underbrace{\max_{i \in I_{k+1}, \neq i_k} \frac{J(x_0, \pi^*_k, s_i)}{J^*(x_0, s_i)}}_{i_k}
\}
= \max \{ \max_{i \in I_k} \frac{J(x_0, \xi^*_k)}{J^*(x_0, s_i)}\}
\]

The second equality sign is due to Lemma 1. The third and fifth equality signs are due to the definition of \(z_k\) and \(V_k\) as well as the property of \(I_k\). The forth equality sign is due to the way that \(\xi^*_k\) is constructed. The less-than-or-equal-to sign is due to that \(\pi^*_k\) is the optimal control of the truncated WCRC problem.

As \(\pi^*_k\) is known as a WCRC optimal control that minimizes \(\max_{i \in I_0} \frac{J(x_0, \pi^*_k, s_i)}{J^*(x_0, s_i)}\), Equation (27) implies that the constructed \(\xi^*_k\) is also a WCRC optimal control policy.

### 3.3.2. Solving Procedure

Based on Theorem 1, the WCRC optimal control policy \(\pi^*_k\) can be constructed via a backward induction. The procedure is described as follows.
1) Calculate $J^*(x_0, s_i)$ for every $s_i \in \mathcal{S}$.

2) Set the initial condition $\pi^*_N$ and $V_N(x_N, \pi^*_N, s_i)$ for the backward induction. For the problem here, $\pi^*_N$ is null (no control is needed at step $N$), and $V_N(x_N, \pi^*_N, s_i) = g_N(x_N)$.

3) Conduct the backward induction. Suppose that $\pi^*_{k+1}$ is obtained for all possible $(x_j, z_j, I_j)$, $j \geq k + 1$, thus $V_{k+1}(x_{k+1}, \pi^*_{k+1}, s_i)$ is also known. The objective is to find a one-step policy $u_k(x_k, z_k, I_k)$. By Theorem 1, $\pi^*_k$ can be built by combining $u_k(x_k, z_k, I_k)$ and $\pi^*_{k+1}$. $u_k(x_k, z_k, I_k)$ can be found from the following equation,

$$ u_k(x_k, z_k, I_k) = \arg \min_{u_i \in \mathcal{U}_k} \max_{a_i \in \mathcal{A}_k} \frac{z_k + g_k(x_k, u_k, w_k) + V_{k+1}(x_{k+1}, \pi^*_{k+1}, s_i)}{J^*(x_0, s_i)} , \quad (28) $$

where $w_k = s_i(k+1)$ and $x_{k+1} = f_k(x_k, u_k, w_k)$.

After the backward induction is conducted from step $N-1$ to step 0, the complete WCRC optimal control policy $\pi^*$ is constructed.

3.4. Extension to the Stochastic Case

In the discussion above, all disturbance sequences in the set $\mathcal{S}$ are treated equally. However, in practice, there are some situations where we do have some a priori knowledge about the probability ranges of the occurrence of these sequences. The concept of deterministic worst-case relative cost optimal control can be extended to this case and the expected worst-case relative cost will be minimized.
3.4.1. Probabilities of Disturbance Sequences

For an indexed disturbance sequences set \( S = \{s_i | i \in \{1, 2, ..., M\} \} \), define a vector \( \mathbf{p} = [p_1, p_2, ..., p_M]^T \in \mathbb{R}^M \), where \( p_i \) is the probability that the disturbance sequence \( s_i \) occurs. It is assumed that at the initial step, the upper bound and lower bound of each probability are known. This implies that \( \mathbf{p} \) lies in a convex set \( P_0 \),

\[
P_0 = \{ \mathbf{p} | p_i \in [a_i, b_i], \mathbf{p}^T \mathbf{1} = 1 \},
\]

where \( \mathbf{1} \) is a vector with all entries being one. \( P_0 \) is called the admissible disturbance sequence probability set.

As the disturbance is measured at each step, the knowledge of the vector \( \mathbf{p} \) needs to be updated at every step. If at each step \( P_k \) is updated as the set of new probability, \( P_k \) will not have any “shrinking-like” properties as (22). Therefore in this method, a scaled probability is used, which means that the sum of the elements of \( \mathbf{p} \) may no longer be 1. As it will be shown that \( \mathbf{p} \) appears in both the denominator and the numerator of the definition of the expected relative cost, this will not affect the value of the final result.

At each step \( k \), the update law of \( P_k \) is as follows.

\[
P_k = \{ C_k \mathbf{p} \mid \mathbf{p} \in P_{k-1} \},
\]

where \( C_k \) is a diagonal matrix, \( c_{k,i} = 1 \) if \( i \in I_k \), and \( c_{k,i} = 0 \) if \( i \not\in I_k \).

This update law is essentially that \( p_i \) will remain its value if \( s_i \) is still admissible at step \( k \), and will become 0 if \( s_i \) becomes inadmissible.
By this update law, \( P_k \) is always convex. \( p \in P_k \) will be a scaled probability vector as \( p^T \mathbf{1} \leq 1 \). It should also be noted that \( C_kC_{k-1}...C_1 = C_k \), therefore \( P_k = C_kP_0 \).

### 3.4.2. Expected Cost Function

Given the indexed set \( S \) with a probability \( p \), the expected accumulative absolute cost is defined as,

\[
J_E(x_0, \pi, p) = E_{x_0, x_2, ..., x_M} \left\{ \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) + g_N(x_N) \right\} = p^T J(x_0, \pi),
\]

where \( J(x_0, \pi) = [J(x_0, \pi, s_1), J(x_0, \pi, s_2), ..., J(x_0, \pi, s_M)]^T \).

Similarly, define

\[
V_{E,k}(x_k, \pi, p) = p^T V_k(x_k, \pi),
\]

where \( V_k(x_k, \pi) = [V_k(x_k, \pi, s_1), V_k(x_k, \pi, s_2), ..., V_k(x_k, \pi, s_M)]^T \).

The expected optimal cost can be defined as,

\[
J^*_E(x_0, p) = E_{x_0, x_2, ..., x_M} \left\{ \min_{\pi} J(x_0, \pi, s_i) \right\} = p^T J^*(x_0),
\]

where \( J^*(x_0) = [J^*(x_0, s_1), J^*(x_0, s_2), ..., J^*(x_0, s_M)]^T \).

Then the expected WCRC is defined as,

\[
W_E(x_0, \pi, P) = \max_{p \in P} J^*_E(x_0, \pi, p) = \max_{p \in P} \frac{p^T J(x_0, \pi)}{p^T J^*(x_0)}.
\]

The “worst case” here means the worst probability \( p \) in the set \( P \). It is obvious that if \( P \) is scaled by a constant factor, the value of \( W_E(x_0, \pi, P) \) will not be affected.
3.4.3. Stochastic WCRC Problem

The stochastic WCRC optimal control problem formulation is the same as the deterministic case Problem 1, except that the WCRC $W(x_0, \pi, I_0)$ is replaced by the expected WCRC $W_E(x_0, \pi, P_0)$, and the feedback information of $I_k$ is replaced by $P_k$.

Similar definition for truncated WCRC optimal control problem can also be made by replacing $Y_k$ with $Y_{E,k} = \max_{p \in P} \frac{p^T 1 z_k + p^T V_k(x_0, \pi)}{p^T J(x_0)}$.

If in the initial admissible disturbance sequence probability set $P_0$, every $p_i$ is assumed to be either 0 or 1, i.e., $p_i \in \{0,1\}$, the stochastic problem is degraded to the deterministic WCRC problem discussed in Section 3.2.4.

3.4.4. Extension of Theorem 1

The extension of the key theorem to the stochastic case is very similar to the original one. The proof is slightly different.

**Proof of Theorem 1 Extension**: A few notations need to be defined first. Define

$$K_a(x_0, \pi, I_k) = C_k \left[ \begin{array}{c} J(x_0, \pi, s_1) \\ \vdots \\ J(x_0, \pi, s_M) \end{array} \right]^T,$$

and

$$K_a(x_0, \pi, I_k) = (I - C_k) \left[ \begin{array}{c} J(x_0, \pi, s_1) \\ \vdots \\ J(x_0, \pi, s_M) \end{array} \right]^T,$$

where $I$ is the identity matrix.
By Lemma 1, $K_n(x_0, \mathbf{\pi}^*, I_k)$.

Also it can be shown that,

$$
\max_{p \in \mathcal{P}_n} \sum_{i \in I_k} \frac{z_i + V_i(x_i, \mathbf{\pi}, s_i)}{J^*(x_i, s_i)} p_i
= \max_{p \in \mathcal{P}_n} \left[ \frac{z_k + V_k(x_k, \mathbf{\pi}, s_k)}{J^*(x_k, s_k)} \ldots \frac{z_k + V_k(x_k, \mathbf{\pi}, s_M)}{J^*(x_k, s_k)} \right] C_k p
= \max_{p \in \mathcal{P}_n} \sum_{i \in I_k} \frac{z_i + V_i(x_i, \mathbf{\pi}, s_i)}{J^*(x_i, s_i)} p_i.
$$

Then,

$$
\max_{i \in I_k} \frac{J(x_0, i)}{J^*(x_0, s_i)}
= \max \{ \max_{i \in I_k} \frac{J(x_0, i)}{J^*(x_0, s_i)} , \max_{i \in I_k, i \neq i_k} \frac{J(x_0, i)}{J^*(x_0, s_i)} \}
= \max \{ \max_{i \in I_k} \frac{J(x_0, i)}{J^*(x_0, s_i)} , \max_{i \in I_k, i \neq i_k} \frac{J(x_0, \mathbf{\pi}^*, s_i)}{J^*(x_0, s_i)} \}
= \max \{ \max_{i \in I_k} \frac{z_i + V_i(x_i, \mathbf{\pi}^*, s_i)}{J^*(x_i, s_i)} , \max_{i \in I_k, i \neq i_k} \frac{J(x_0, \mathbf{\pi}^*, s_i)}{J^*(x_0, s_i)} \}
\leq \max \{ \max_{i \in I_k} \frac{z_i + V_i(x_i, \mathbf{\pi}^*, s_i)}{J^*(x_i, s_i)} , \max_{i \in I_k, i \neq i_k} \frac{J(x_0, \mathbf{\pi}^*, s_i)}{J^*(x_0, s_i)} \}
= \max \{ \max_{i \in I_k} \frac{J(x_0, \mathbf{\pi}^*, s_i)}{J^*(x_0, s_i)} , \max_{i \in I_k, i \neq i_k} \frac{J(x_0, \mathbf{\pi}^*, s_i)}{J^*(x_0, s_i)} \}
= \max_{i \in I_0} \frac{J(x_0, \mathbf{\pi}^*, s_i)}{J^*(x_0, s_i)}. \quad (36)
$$

The second and the last equality signs are due to the definition of $K_n$ and $K_n$. The third equality sign is due to Lemma 1. The fifth and the seventh equality signs are due to
the relationship shown in (35). The rest of this proof is similar to that of the deterministic case.

Therefore $\tilde{\pi}$ is also a stochastic WCRC optimal control of the original problem.

### 3.4.5. Solving Procedure

The solving procedure is similar to the deterministic case. In the backward induction, the one-step policy is determined by the following equations,

$$u_k(x_k, z_k, I_k) = \arg \min_{u, z \in \mathcal{P}_k} \max_{p \in \mathcal{P}_k} \frac{p^T I_z k + p^T g_k(x_k, u_k, w_k) + p^T V_{k+1}(x_{k+1}, \pi^*_k)}{p^T J_k(x_0)}.$$  \hspace{1cm} (37)

### 3.5. Numerical Examples

#### 3.5.1. A Linear System with Quadratic Cost

Consider the system

$$x_{k+1} = 0.9 x_k + u_k - w_k,$$ \hspace{1cm} (38)

with a quadratic running cost

$$g_k(x_k, u_k, w_k) = x_k^2 + u_k^2,$$ \hspace{1cm} (39)

a terminal cost

$$g_N(x_N) = (1 - x_N)^2, N = 10,$$ \hspace{1cm} (40)

and an initial condition $x_0 = 0.8$. The constraints are $x \in [-2, 2]$ and $u \in [0.2, 1.6]$.

The admissible disturbance sequence set $\mathcal{S}$ contains 192 possible sequences, which are shown in Figure 14. A nominal disturbance sequence, which can be considered as an averaged disturbance, is also shown in Figure 14.
The posteriori absolute optimal costs of all the 192 cases are shown in the first subfigure of Figure 15 (sorted in descending order). It should be noted that the absolute optimal costs of the 192 cases vary from 15.462 to 11.035. So if the absolute-cost-based min-max methods are used, there is no guarantee on the performance of the low-optimal-cost cases.

Figure 15. Absolute costs and relative costs of the different control methods [66]
In the WCRC, the state and input spaces are discretized in order to conduct the backward induction. The proposed WCRC optimal control method is compared with the posteriori absolute optimal control and two other methods: min-max control which uses the information of the disturbance range at each step and minimizes the maximum absolute cost [84], and the nominal optimal control which just applies the optimal control of the model with a nominal disturbance to the system. The results in Figure 15 show that the WCRC optimal control can provide a cost much closer to the posteriori optimal cost in each case than the other two methods. The worst-case relative cost in WCRC optimal control is 1.0087, while the worst-case relative cost of min-max control is 1.1296, and that of nominal optimal control is 1.1571.

3.5.2. A Nonlinear Hybrid Electric Vehicle Problem

This example shows how the stochastic WCRC optimal control can be applied to a specific torque distribution control problem for parallel hybrid electric vehicles. The scenario of the optimal control problem is as follows. There is an intersection ahead and there are two possible routes after the intersection. The predicted vehicle speed and road grade of the two routes are shown in Figure 16. Before the vehicle reaches the intersection, the hybrid electric vehicle controller does not know which route the driver will choose, but it knows that the probability of choosing the first route is higher than the second one. The problem is how to distribute the torque demand between the engine and electric motor before and after the intersection to achieve a low relative cost.
Figure 16. Speed, road grade, motor speed, and torque demand of the two routes [66]

The model here is based on the vehicle model presented in [57]. The state is the vehicle battery state of charge (SOC),

\[ x_{k+1} = \frac{-2\pi T_{mot} n_{mot} \eta_m(T_{mot}, n_{mot}) \eta_b(T_{mot}, x_k) \Delta t}{60Q} + x_k, \quad (41) \]

where \( T_{mot} \) is the motor torque, \( n_{mot} \) is the motor speed, \( \eta_m \) is a variable related to the motor efficiency, \( \eta_b \) is a variable related to the battery efficiency. An approximation is used for the battery efficiency \( \eta_b \) as shown in (42), and the motor efficiency is a lookup table obtained from experimental data.

\[ \eta_b(T_{mot}, x_k) = \begin{cases} 
1/(0.4375x_k + 0.5063), & \text{if } T_{mot} \geq 0 \\
-0.4x_k + 0.94, & \text{if } T_{mot} < 0
\end{cases}. \quad (42) \]

The driver’s future torque demand \( T_{dem} \) and the future motor speed \( n_{mot} \) are considered as the uncertain disturbances. The system input is the engine torque \( T_{eng} \).
The engine and the motor are installed on the same shaft, with a clutch in between. The following three requirements need to be satisfied.

1) If \( T_{\text{dem}} \geq 0 \), \( T_{\text{eng}} + T_{\text{mot}} = T_{\text{dem}} \).

2) If \( T_{\text{dem}} < 0 \), \( T_{\text{eng}} = 0 \), and \( T_{\text{mot}} = \max\{T_{\text{dem}}, -T_{\text{reg,m}}\} \), where \( T_{\text{reg,m}} \) is the maximum absolute torque value for motor regenerative braking.

3) \( n_{\text{eng}} = \max\{n_{\text{idl}}, n_{\text{mot}}\} \), where \( n_{\text{idl}} \) is the engine idle speed.

The running cost is the fuel consumption, which is

\[
g_k(x_k, u_k, w_k) = l_{\text{fuel}}(T_{\text{eng}}, n_{\text{eng}}),
\]

where \( l_{\text{fuel}} \) is a 2-dimension map obtained from experiments.

The terminal cost contains two terms, one is the equivalent electricity energy consumption which is proportional to the battery SOC change, and the other one is a penalty term which tries to maintain the battery SOC around its target value.

\[
g_N(x_N) = \lambda(x_N - x_0) + \lambda_p (x_N - x_{\text{tar}})^2,
\]

where \( \lambda, \lambda_p \), and \( x_{\text{tar}} \) are constants.

There are constraints on the input to prevent the battery from being over-charged or over-depleted.

1) If \( x \geq x_{\text{upb}} \), then \( T_{\text{mot}} \geq 0 \), i.e. \( T_{\text{eng}} \leq \max\{T_{\text{dem}}, 0\} \).

2) If \( x \leq x_{\text{lob}} \), then \( T_{\text{mot}} \leq 0 \), i.e. \( T_{\text{eng}} \geq T_{\text{dem}} \).

The initial disturbance sequence probability set \( P_0 \) can be described as

\[
P_0 = \{p = [p_1 \quad p_2]^T \mid 0.5 \leq p_1 \leq 1, p_1 + p_2 = 1\}.
\]
The vehicle speed and road grade profiles are translated into motor speed and torque demand using a vehicle model. The step size is 1 second and the route decision is at the 23rd second. Before the 23rd second, the disturbances of the two cases are the same.

The posteriori absolute optimal control and WCRC optimal control results are obtained. A new control strategy which is called “optimal switch” is used for comparison purpose. As the posteriori absolute optimal control for the two cases can be calculated in advance but it is impossible to know which one should be used in advance, the optimal switch controller is a controller which happens to use the wrong absolute optimal control before step 23, then switches to the correct optimal control after step 23.

The cost results of the three controllers are shown in Table 12. WCRC optimal controller can guarantee an expected relative cost of 100.7%, while the optimal switch control may lead to an expected relative cost of 102.4%. In this problem, the worst case is 

\[ p = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \]

Table 12. Cost comparison of three control policies

<table>
<thead>
<tr>
<th></th>
<th>Absolute optimal</th>
<th>WCRC</th>
<th>Optimal switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1 cost</td>
<td>70.02</td>
<td>70.4</td>
<td>70.84</td>
</tr>
<tr>
<td>Case 2 cost</td>
<td>31.02</td>
<td>31.34</td>
<td>32.66</td>
</tr>
<tr>
<td>Worst-case expected cost</td>
<td>50.52</td>
<td>50.87</td>
<td>51.75</td>
</tr>
<tr>
<td>Worst-case expected relative cost</td>
<td>100%</td>
<td>100.7%</td>
<td>102.4%</td>
</tr>
</tbody>
</table>

3.6. Summary

In this Chapter, a worst-case relative cost optimal control for discrete-time nonlinear dynamic systems with a finite number of possible disturbance sequences over a finite
horizon is formulated and solved via backward induction. The method is also extended to the stochastic case where only the ranges of the disturbance sequence probabilities are known. Two examples are provided to illustrate the proposed methods.
Chapter 4: Optimizing the Energy Management Strategy for Fixed-Route HEVs

Many HEV energy management strategies are developed and evaluated under fixed driving cycles. However in the real-world driving, vehicles are very unlikely to be running under a fixed known cycle. Instead, a lot of vehicles run on fixed routes. Unfortunately, human driving data collected on a driving simulator shows that it is very difficult to select or create a determined typical driving cycle to represent the fixed-route driving due to the uncertainties in traffic light stops and driver behaviors. This chapter presents a two-level stochastic approach to optimize the energy management strategy for fixed-route HEVs. The historical data on the fixed route are utilized and a road-segment-based stochastic HEV energy consumption model is built. The higher-level energy optimization problem is solved by SDP. The SDP computation uses the vehicle model and historical driving data on the fixed route and it can be conducted offline. The result of SDP is a 2-dimension lookup table of parameters for lower-level control strategy therefore this approach can be easily real-time implemented in practice. The developed stochastic approach is compared with three strategies with data collected on the driving simulator: the optimal energy consumption by assuming all trip information is known in advance and solved by DP, a determined energy management approach using typical trip data of the fixed-route driving, and a simple strategy which does not require any route data. Simulation results show that the proposed stochastic energy management strategy
consumes 1.8% more energy than the optimal result after 24 trips on the fixed route and considerably outperforms the other two real-time HEV energy management strategies. This Chapter is mainly based on the published work [57].

4.1. Fixed-Route Driving

In the real-world driving, it is impossible that a vehicle strictly follows a fixed and known driving cycle. However, a lot of vehicles run on fixed routes. For example, the public transportation buses, some utility vehicles, and personal cars for commuters all run on fixed routes. The undiscovered information in the fixed-route driving can definitely help improving the control strategies of HEVs, because in general the more information we have for the entire trip, the better we can design the energy management strategies. Some research has been conducted on exploring the potential for fixed-route HEVs. A method to online predict the future speed profiles for vehicles operating in fixed-route service is presented in [85]. In [86] a clustering method to study distance-based driving pattern and energy management strategy for fixed-route driving are proposed. In [87] the energy consumption of an HEV is optimized using a pre-defined speed profile and an online adjustment scheme that compensates for the difference between the actual speed profile and the pre-defined profile is provided. In [88] an approach to select the most representative trip is presented and DP is applied to optimize the energy consumption of trip while the uncertainty in route length is considered. Then the battery SOC result from DP is used to design a real-time HEV control strategy for commuter vehicles. Most of the essential ideas of the aforementioned research are based on deriving a fixed driving cycle
as a typical trip of the fixed-route driving, and then designing an energy management strategy around the fixed cycle.

However, due to the different traffic conditions, traffic light changes, weather conditions, even driver’s emotions, the fixed-route vehicles cannot be considered as being driven under a fixed cycle. The actual speed and torque demands can vary substantially among each trip even on the same route by the same driver. Those attempts to describe or approximate the fixed-route driving as a fixed cycle cannot guarantee the optimal energy consumption result. More details about this phenomenon will be discussed later in Section 4.2. Some studies try to use stochastic approaches to describe the uncertain trip information and design an HEV energy management strategy based on stochastic models [15][17][48][54][55][56][89]. These previous studies use short-time-scale stochastic models, usually sampled at about 1 Hz (some varies from about 0.2 Hz to 5 Hz). These stochastic models cannot capture the long-term behavior if the total trip length is more than thousands of seconds. For example, a point-wise pedal Markov model sampled at 1 Hz can hardly provide enough information about the future state after thousands of steps. The methods described in the previous studies are not suitable for the long-term behavior description because of their structures and their target of predicting the point-wise speed and torque demands. Most of these models do not consider the road environment (e.g. local or highway, uphill or downhill) except [89]. But reference [89] only provides an example of using two different driving patterns (urban and extra-urban), which may be too simple especially for the fixed-route case where much more valuable
information is available. None of these models can take into account the uncertainties in the traffic light signals.

Unlike a point-wise model, a road-segment-based model [61][90] is more suitable for long-term prediction. This is because it extends the length in each prediction step thus fewer steps are required for longer prediction, and it can also distinguish some key road environment factors such as local road or highway, and uphill or downhill. In this chapter, a two-level control structure is used: in the lower level, a practical HEV energy management controller is implemented, but the key parameters in the controller will be determined in the higher level where a stochastic method is used based on the road-segment model. In this two-level structure, the higher level takes the advantage of long-term stochastic prediction and gives “strategic-level” command, while the lower-level control focuses on the local operational control.

In this chapter, it is assumed that the interested vehicle only runs on a fixed route, or the vehicle runs on a fixed route frequently and the controller is able to recognize it (for more information regarding route prediction and recognition, please refer to [91][92]). It is also assumed that the historical driving data of the fixed-route driving, including the speed and torque demand, are recorded and available.

4.2. Driving Data Analysis

In this section, the driving simulator experiment for fixed-route driving behavior study will be introduced first. Then the data collected in the driving simulator will be
analyzed to show that fixed-route driving cannot be represented by a fixed cycle, and the ideas on how the fixed-route information data can be used will be given.

4.2.1. Driving Simulator Experiment

Human driver driving experiments are conducted on the driving simulator introduced in Section 1.3. A 12-kilometer track, which is demonstrated in Figure 17, is designed as a combination of both local roads and highways. Road slopes are considered in the road model and vehicle model, which means there are some uphill and downhill in the trip. There are multiple stop signs and traffic lights on the local road. Some detailed track parameters are shown in Table 13. The driver is required to drive following the speed limits, stop signs, traffic light signals, and other traffic regulations. In the test, a test driver has driven on the track for 24 times, at the frequency of once or twice a day, to simulate a normal commuter behavior. An overview of the test trips is shown in Table 14.

![Figure 17. The test track used in the driving simulator experiment [57]](image)
Table 13. Test track parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Road length</td>
<td>8.18 km local, 3.55 km highway</td>
</tr>
<tr>
<td>Maximum road grade</td>
<td>9% uphill, -9% downhill</td>
</tr>
<tr>
<td>Number of turns</td>
<td>15 left, 18 right</td>
</tr>
<tr>
<td>Number of stop signs</td>
<td>7</td>
</tr>
<tr>
<td>Number of traffic lights</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 14. Overview of test trips

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average time</td>
<td>13 minutes 50 seconds</td>
</tr>
<tr>
<td>Average speed</td>
<td>15 km/h local, 96 km/h highway</td>
</tr>
<tr>
<td>Average number of stops (including at stop signs and red lights)</td>
<td>14.7</td>
</tr>
</tbody>
</table>

In this driving simulation, there is no other pertinent traffic on the road. The analysis of the test results in the following subsection will show that even if only the traffic lights and driving behavior variations are considered, the uncertainties in fixed-route driving cannot be ignored.

4.2.2. Characteristics of the Fixed-Route Driving Data

In the driving simulation, there is no traffic or weather change. Therefore the difference among different trips is only introduced by the traffic light schedule change and behavioral variation of the driver. The data analysis in this section will show that even though there are fewer uncertainties in the test than in the real world, the variations among different trips are still great and cannot be neglected.
Figure 18 and Figure 19 show the speed and pedal position of 10 trips when the vehicle runs through an intersection with traffic lights, in the time domain and the distance domain (only 10 of all the 24 trip data are shown in order to keep the figures uncluttered and readable). In 5 out of the 10 trips, the vehicle stops because of the red light, and in the other 5 trips, the vehicle does not stop and just runs through the intersection under green light. Furthermore, even within each group the speed and pedal position also vary a lot along either time or distance. It can be seen that it is impossible to define a single averaged or typical trip, which is in the form of a time sequence of speed and a time sequence of torque, to represent the vehicle behavior through this intersection.

Figure 18. The vehicle speed of 10 trips when running through an intersection [57]
Table 15 shows the time and energy of the 10 trips in this intersection area. As the HEV energy management strategies have a great impact on the fuel and electricity consumption, in this section, only the mechanical energy involved in the vehicle motion is considered so it is independent of the HEV control strategy. Propulsive energy is defined as the integral of positive power demand from the driver, and regenerative energy is the integral of regenerative braking power. They are both calculated from the recorded torque and speed data. The total energy is a roughly estimated value using the propulsive energy minus the regenerative energy times a constant conversion efficiency (which is assumed to be 0.7). The data show that even if the speed profile looks similar, the energy demands could be quite diverse. The energy consumption differences among trips on the
same route are really huge, e.g. the propulsive energy varies from 19 kJ to 374 kJ, and the regenerative energy varies from 0 to 285 kJ.

Table 15. Time and energy of 10 trips when the vehicle runs through an intersection

<table>
<thead>
<tr>
<th>Trip No.</th>
<th>Stop (Y/N)</th>
<th>Time (s)</th>
<th>Total Energy (kJ)</th>
<th>Propulsive Energy (kJ)</th>
<th>Regenerative Energy (kJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Y</td>
<td>52.1</td>
<td>65.6</td>
<td>221.2</td>
<td>222.2</td>
</tr>
<tr>
<td>2</td>
<td>Y</td>
<td>47.6</td>
<td>97.5</td>
<td>213.2</td>
<td>165.4</td>
</tr>
<tr>
<td>3</td>
<td>Y</td>
<td>50.7</td>
<td>77.1</td>
<td>219.4</td>
<td>203.3</td>
</tr>
<tr>
<td>4</td>
<td>Y</td>
<td>48.4</td>
<td>100.5</td>
<td>299.8</td>
<td>284.8</td>
</tr>
<tr>
<td>5</td>
<td>Y</td>
<td>49.9</td>
<td>193.2</td>
<td>374.4</td>
<td>258.9</td>
</tr>
<tr>
<td>6</td>
<td>N</td>
<td>35.6</td>
<td>81.6</td>
<td>140.0</td>
<td>83.4</td>
</tr>
<tr>
<td>7</td>
<td>N</td>
<td>34.7</td>
<td>3.9</td>
<td>19.1</td>
<td>21.7</td>
</tr>
<tr>
<td>8</td>
<td>N</td>
<td>33.5</td>
<td>122.5</td>
<td>122.5</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>N</td>
<td>29.5</td>
<td>176.5</td>
<td>246.3</td>
<td>99.8</td>
</tr>
<tr>
<td>10</td>
<td>N</td>
<td>34.0</td>
<td>61.3</td>
<td>157.2</td>
<td>137.0</td>
</tr>
</tbody>
</table>

Figure 20 is the histogram of time and energy consumptions of all 24 trips at the intersection. It shows that the local behavior of fixed-route driving varies trip by trip. Figure 21 shows the distribution of the overall time and global energy usage of the 24 trips. The differences are not as huge as the local ones, as some of the randomness is averaged over the long trip, but still significant, e.g. the median of regenerative energy is about 2400 kJ, but the maximum can be more than 2800 kJ, and the minimum value can be less than 2000 kJ. Therefore in terms of energy usage, it is also inappropriate to use one fixed cycle to approximate the fixed-route driving.
Figure 20. Time and energy of 24 trips when the vehicle runs through an intersection [57]

Figure 21. Overall time and global energy of 24 trips [57]
As the fixed-route driving cannot be considered as a determined cycle, the knowledge from the optimal strategy of any single trip cannot be used directly on other trips. It is also not easy to construct an “averaged trip”, especially when there are uncertainties on traffic lights so that we do not even know whether the vehicle will stop or keep running at a certain intersection. But these test data also imply that, we may be able to describe the fixed-route driving in a stochastic way. In fact not only the overall trip energy demand follows a certain probability distribution, in the viewpoint of statistic, a lot of properties for the fixed-route driving can be considered as random variables or random vectors whose probability distribution function can be approximated by calculating the frequency from historical data.

It should be noted that although the influence of traffic is not tested on the driving simulator, the ideas of describing the uncertainties in the driving as random variables and calculating the distribution from historical data are still valid when considering real-world driving with all real-world uncertainties. This motivates the concept of building a stochastic model using historical fixed-route driving data and optimizing the energy consumption based on this stochastic model.

4.3. Stochastic Optimal Control for Fixed-Route HEVs

Following the idea that some properties of the fixed-route driving can be considered as random variables with known probability distribution, the energy management of the HEV on fixed routes can be formulated as a stochastic optimal control problem. In this section, the problem formulation will be presented, and the solution will be given via
SDP. The result from SDP can be made into lookup tables so that this strategy can be implemented in real time.

4.3.1. Two-Level Control Structure and Road Segmentation

In [93] it is shown that the trip prediction is not very necessary for HEV energy management strategies unless the battery SOC may hit the boundaries of its range. In [50] it is claimed that the best strategy for a plug-in HEV is to deplete the battery right by the end of the trip. Therefore, for non-plug-in HEV, the key of energy management is to keep the SOC balanced during the trip, and for plug-in HEV, the key of the energy management is to deplete the battery right before reaching the charging station.

But the discussion in the previous section shows that, even for a known route, the energy usage of a trip is not a constant value but a random variable. Therefore the battery SOC balancing or depleting problem can be formulated in a stochastic way. Following this idea, a stochastic HEV energy consumption model will be built based on the fixed-route driving data, and the impact of the controller output on fuel and electricity consumptions will be included in the model.

The great difference brought by driver behavior uncertainties and traffic light changes suggest that it is not very appropriate to build a point-wise stochastic model, which tries to describe the speed or torque demand at every sampling time or sampling position. Therefore in this work, a road-segment-based stochastic description is constructed. The fuel and electricity consumption under a certain controller over a road segment are considered as random variables, and the parameters from the controller can
affect the probability distributions of these random variables. These parameters will be controlled (at a higher level) to achieve desired fuel and electricity consumption distributions in each road segment, thus to achieve minimum overall energy usage on the whole trip.

Due to the attribute of the random variables, the parameter optimization is conducted on the road segment level, which means it should be based on a well-designed HEV controller. This naturally leads to a two-level control structure for HEV control: a lower-level controller with tuning parameters, and a higher-level controller focusing on making use of the whole trip information and giving strategic-level commands to the lower-level controller.

The lower-level control is a simple effective and real-time strategy, which only focuses on current road segment and does not need to consider long-term influence on the powertrain, such as the total distance of the trip, or potential long uphill/downhill ahead. It can be an ECMS, rule-based, fuzzy or MPC controller. The higher-level control takes into account of the whole picture of the trip using all the available information of the fixed route, and calculates the optimal parameters for the lower-level controller by solving a stochastic optimization problem. While the lower-level controller can run at a high frequency (e.g. 100 Hz) to meet the requirement of HEV control, the higher-level controller only updates at the beginning of every road segment, outputting the parameters which the lower-level controller will use throughout the road segment. The structure of the HEV control system is shown in Figure 22, and a comparison between the higher-level and lower-level controller is shown in Table 16.
Figure 22. Two-level structure of HEV control with fixed-route trip information [57]

Table 16. Higher-level and lower-level system comparison

<table>
<thead>
<tr>
<th></th>
<th>Higher-level control</th>
<th>Lower-level control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target system</td>
<td>HEV powertrain with a lower-lever controller</td>
<td>HEV powertrain</td>
</tr>
<tr>
<td>System model</td>
<td>Road-segment-based model</td>
<td>Basic HEV model</td>
</tr>
<tr>
<td>Model time scale</td>
<td>Long (in this chapter, the average step is 590 meters long in distance)</td>
<td>Short (usually sampled at above 1 Hz frequency)</td>
</tr>
<tr>
<td>Model type</td>
<td>Stochastic</td>
<td>Determined</td>
</tr>
<tr>
<td>Model input</td>
<td>Lower-lever controller parameter</td>
<td>Engine torque, motor torque, etc.</td>
</tr>
<tr>
<td>Model feedback</td>
<td>SOC, current road segment</td>
<td>SOC, speed, etc.</td>
</tr>
<tr>
<td>Controller design method</td>
<td>Stochastic optimization over the whole trip</td>
<td>Any method (in this chapter, ECMS, which is instantaneous optimization, will be used as an example)</td>
</tr>
</tbody>
</table>
In order to capture the uncertainties of fixed-route driving, each road segment contains a road section which may lead to relatively large randomness in terms of the torque demand, e.g., a turn or an intersection. Therefore all the stop-or-go uncertainties caused by the traffic, red lights, and driving behavior uncertainties during the acceleration and brake can be represented by the chosen random variables associated with this road segment. There is usually less randomness in constant speed cruising, so each segment except the first and the last one, starts and ends at where the vehicle is supposed to be cruising. Figure 23 demonstrates the segmentation of the road map of the driving simulator test. This segmentation simplifies the fixed-route driving, avoids the difficulties in dealing with the random process of the whole trip, but still captures the key characteristics of the driving and leaves enough room for the controller to adjust the energy distribution while maintaining high overall efficiency.

Figure 23. Road segmentation of the test map [57]
4.3.2. Basic HEV Model and Lower-Level Controller

A basic HEV dynamic model is built first. The road-segment-based stochastic model will be developed based on this basic HEV dynamic model. The basic HEV model used in this work employs a map-based engine model. But it can be replaced by a more complicated transient model while the rest of this stochastic approach, including the real-time capability, is still valid. However, the benefit of using a map-based model is that the global optimization result using the same model could be easily obtained by DP and the theoretical performance of the proposed method can be evaluated. While this model is built for the HEV configuration in Figure 24, the method described in this section can be extended to other types of HEVs.

The HEV dynamics can be described by the following equations.

\[ \dot{SOC} = \frac{\tau_o \cdot OC \cdot \eta_m(n_m, T_m) \eta_m(n_m, T_m) \cdot 2\pi \eta_m T_m}{60}, \]  \hspace{1cm} (46)

\[ [T_m \ T_e]^T = f_{EMS}(T_{dr}, \nu, SOC, u), \]  \hspace{1cm} (47)

Figure 24. A HEV configuration [57]
\[ T_m + T_e = T_d, \]  

(48)

where \( SOC \) is the battery state of charge, \( \eta_b \) is the battery efficiency or the inverse of battery efficiency, depending on the sign of the motor speed and torque, \( n_m \) is the motor speed, \( T_m \) is the motor torque, \( \eta_m \) is the motor efficiency or the inverse of motor efficiency, depending on the sign of the motor speed and torque, \( T_e \) is the engine torque, \( f_{EMS} \) is the lower-level energy management strategy, \( T_{dr} \) is the torque demand from the driver, \( v \) is the vehicle speed, and \( u \) is a tuning parameter in the energy management strategy.

Equation (47) means that the torque distribution is determined by a parameterized control strategy, where \( u \) is the parameter to be determined. In general, \( f_{EMS} \), which is the lower-level controller, can be any HEV energy management with tuning parameters that can affect the energy distribution, e.g., ECMS, or rule-based strategies. Therefore the parameter \( u \) can be the electricity-to-fuel equivalence factor in ECMS, or the thresholds in rule-based strategies. As ECMS usually shows better performance than rule-based strategy, in this work, without much loss of generality, ECMS is used as the lower-level energy management strategy and \( u \) is assumed to be the electricity-to-fuel equivalence factor. ECMS can be described using the following equation [47]:

\[ T_m^*(\lambda) = \arg \min_{T_m} [\lambda * S(\lambda, n_m, \eta_m, e, T_e)], \]  

(49)
where \( \lambda \) is the electricity-to-fuel factor (which is also considered as the tuning parameter \( u \) for ECMS), and \( f_{f/u} \) is the engine instantaneous fuel consumption. \( T_m^* \) can be usually uniquely determined.

Equation (49) gives an ideal formula of ECMS. But in practice, some constraints need to be included in the strategy \( f_{EMS} \). For example, when the battery SOC exceeds its upper bound, charging the battery will not be allowed, and when the battery SOC drops below its lower bound, discharging the battery will not be permitted. It is assumed that the function \( f_{EMS}(\cdot) \) is a practical energy management strategy which is based on ECMS Equation (4) and has all these necessary constraints included.

4.3.3. Segment-Based Stochastic Model

In this subsection, the battery SOC change and fuel consumption over a road segment will be focused. The impact of the lower-level parameter, which is the electricity-to-fuel factor in the ECMS, will be discussed. Then a stochastic optimal control problem will be formulated by considering the battery SOC and fuel consumption over each road segment as random variables.

For the \( k \)-th segment of a certain trip, the battery SOC can be calculated as a function of the initial SOC \( x_{SOC}(k) \) of this road segment and the strategy parameter \( u \), if \( f_{SOC}(\cdot) \) and \( f_{f/u}(t) \) are known.

\[
\Delta x_{SOC}(k, x_{SOC}(k), u) = \int_{t_k}^{t_{k+1}} f_{SOC}(\cdot) \, dt,
\]

\[
\text{(50)}
\]
\[ \Delta f_{fu} [k, x_{SOC}(k), u] = \int_{t_k}^{t_{k+1}} f_{fu} dt, \tag{51} \]

where \( t_k \) is the start time of the \( k \)-th segment and \( t_{k+1} \) is the start time of the \((k+1)\)-th segment. \( \Delta x_{SOC}[k, x_{SOC}(k), u] \) and \( \Delta f_{fu}[k, x_{SOC}(k), u] \) stands for the battery SOC change and fuel consumption of Road Segment \( k \), with initial SOC of this segment \( x_{SOC}(k) \), and controller parameter \( u \).

As shown in Section 4.2.2, for any segment, the torque demand and speed profile vary trip by trip, so for the same road segment, even when the start SOC \( x_{SOC}(k) \) and strategy parameter \( u \) are the same, \( \Delta x_{SOC} \) and \( \Delta f_{fu} \) also vary trip by trip, because \( St \), and \( f_{fu}(t) \) are different trip by trip. Figure 25 is the boxplot of \( \Delta x_{SOC} \) and \( \Delta f_{fu} \) for Road Segment 10 under different strategy parameter \( u \) using the 24-trip data collected on the driving simulator and the basic HEV model. The maximum, minimum, median, quartiles and outliers of 24 trip data using each electricity-to-fuel factor over this road segment are shown. It can be seen that for a fixed \( u \), \( \Delta x_{SOC} \) and \( \Delta f_{fu} \) for different trips lie in a range, which means it can be considered as a random variable rather than a determined value. The trend of the impact of the strategy parameter \( u \) is still very obvious on the boxplot: the greater \( u \) is, the less electric energy is used and the more fuel is used. This means that if a stochastic model is built based on these data, the key impact of the parameter on the energy distribution can still be captured.
It is assumed that the battery SOC change $\Delta X_{SOC}$ and fuel consumption $\Delta F_{fu}$ in each segment are random variables. The probability distributions of the random variables are calculated using historical data and the basic HEV model. When a large number of trips are available, the probability distribution of the random variables $\Delta X_{SOC}[k,x_{SOC}(k),u]$ and $\Delta F_{fu}[k,x_{SOC}(k),u]$, which means the battery SOC change and fuel consumption of Road Segment $k$, with initial SOC $x_{SOC}(k)$, and lower-level controller parameter $u$, can be approximated by the relative frequencies from the existing data and the basic HEV model. It should be noted that when using historical data, the distribution of $\Delta X_{SOC}[k,x_{SOC}(k),u]$ and $\Delta F_{fu}[k,x_{SOC}(k),u]$ will automatically consider all the real-world uncertainties, including those introduced by the traffic, the red lights, and the driver behavior variation, because the observed historical torque and speed.
data are influenced by all the real-world factors during driving and it is technically impossible to remove the impact of any single factor.

In the computation, $x_{\text{SOC}}(k)$ and $u$ are also discretized, which means $m$ different $x_{\text{SOC}}(k)$ values and $n$ different $u$ values will be used in the distribution calculation, so $(m \times n)$ distributions can be obtained after the computation for each segment. Figure 26 and Figure 27 give some examples of the approximated probability density distributions of Road Segment 10, using the 24 trip data collected on the driving simulator. In this example, it is assumed that $m = 1$ and $n = 4$. The choice of $m$ depends on how the initial battery SOC value on a road segment will affect the SOC change over this road segment. If the SOC change is strongly dependent on the initial SOC value, which roughly means that the battery efficiency varies a lot over different SOC values, a larger $m$ is needed. More accurate approximation can be obtained by increasing the values of $m$ and $n$.

![Figure 26. An example: approximated probability density distribution of $\Delta X_{\text{soc}}$][57]
Figure 27. An example: approximated probability density distribution of $\Delta F_{fu}$ [57]

Based on these distributions, a road-segment based stochastic discrete model will be built. The state of this discrete model $x_{d,SOC}$ is the battery at the beginning of each segment. The discrete stochastic model dynamics is simply

$$x_{d,SOC}(k+1) = x_{d,SOC}(k) + \Delta X_{SOC}[k, x_{d,SOC}(k), u],$$  \hspace{1cm} (52)

where $k$ is the segment number. $k = 1, 2, 3, ..., N−1$, assuming that $(N−1)$ is the total number of road segments in this fixed route.

The running cost of the stochastic model is the expectation of the segment fuel usage.

$$l_z[k, x(k), u(k)] = \mathbb{E}\{\Delta F_{fu}[k, x_{d,SOC}(k), u(k)]\},$$  \hspace{1cm} (53)
where \( l_s \) is the running cost, \( E(\cdot) \) stands for the expectation of a random variable.

The terminal cost is the overall consumed electricity energy.

\[
J_f[x(N)] = \lambda_f [x_{d,SOC,0} - x_{d,SOC}(N)],
\]

where \( J_f \) is the terminal cost, \( \lambda_f \) is the price factor, \( x_{d,SOC,0} \) is the initial battery SOC. Unlike the electricity-to-fuel equivalence factor \( \lambda \) in ECMS, which can be adjusted according to the requirement of the strategy, \( \lambda_f \) is a constant value determined by the price of fuel and electricity. Though the price may also be changing, it is assumed as a constant for each trip.

The total cost is the sum of all running costs and terminal cost, which is the price of total energy consumed.

\[
J = J_f + \sum_{k=1}^{N-1} l_s(k).
\]

When the initial value \( x_{d,SOC}(1) = x_{d,SOC,0} \) is given, the total cost of the stochastic model is a function only dependent on the strategy parameters in each segment, \( u_1, u_2, \ldots, u_{N-1} \). So the stochastic optimal control problem is to select \( u_1, u_2, \ldots, u_{N-1} \) such that \( J \) is minimized.

### 4.3.4. SDP for Higher-Level Controller Design

Suppose that \( u \) can take values from a finite set \( U \), and \( x \) can take values from a finite set \( X \). The formulated problem (52)(53)(54)(55) can be solved using SDP. Assume that \( V_k(x) \) is the optimal cost at the k-th step, depending on the state \( x \), and
$u_k(x)$ is the optimal action at the $k$-th step if the state value is $x$. Then the problem can be solved by Backward Induction Algorithm [94],

$$V_k(x) = \min_{u \in U} [I_k(x,u) + \sum_{z \in X} P(z|x,u)V_{k+1}(z)], \quad (56)$$

$$u_k(x) = \arg \min_{u \in U} [I_k(x,u) + \sum_{z \in X} P(z|x,u)V_{k+1}(z)], \quad (57)$$

where $P(z|x,u)$ is short for the conditional probability $P[x(k+1) = z | x(k) = x, u(k) = u]$, which can be obtained from the probability distribution of $\Delta X_{SOC}[k, x_{SOC}(k), u]$. The algorithm will be conducted backwards from $k = N$ to $k = 1$. The initial value for $V_N$ is

$$V_N(x) = J_f(x), \forall x \in X. \quad (58)$$

The result of SDP will be a 2-dimensional table, whose inputs are segment number $k$ and battery SOC, and outputs are $u(k)$. The 2-dimensional table is shown in Figure 28. From the figure, it can be seen that in general for the same road segment the required strategy parameter $u$ will be smaller if the battery SOC is higher, which means the higher battery SOC is, the more electricity will be used in ECMS. There are a few exceptions of this rule because sometimes two different strategy parameters can give very close, or even the same SOC change and fuel consumption results (as can be seen in the boxplot in Figure 25), and such close results may lead to different optimal parameters at adjacent SOC grid points after the numerical optimization computation. So at these “unsmooth” positions, either using the table parameter value or a “filtered” value will give very close or even the same fuel consumption results, and it has no impact on the
control performance. However, under the same initial SOC for different road segment, it is not necessarily true that the strategy parameter \( u \) is greater if the remaining distance is shorter. For example, Road Segment 9 is mostly downhill with very low load, and the optimal strategy parameter \( u \) is always very small which means more electricity should be used.

Figure 28. SDP result: a 2-dimensional lookup table [57]

4.3.5. Real-Time Implementation

As shown above, with all historical data being available, the computationally-intensive SDP process can be conducted off-line. A more complicated HEV model considering transient process, or a more complicated lower-level controller with multiple heuristic rules or engine on/off control, can be easily embedded to basic model, thus the
SDP approach. In the real-time controller, at the beginning of each road segment, a table lookup is conducted to obtain the optimal strategy parameter $u$. The remaining tasks of the real-time HEV controller are not different from a standard lower-level controller. Every time a new trip is completed, the probability distribution can be updated with the latest trip information and the SDP can be ran again, then new tables would be available for the next trip. All these update can be done off-line. Therefore the entire strategy can be implemented without the requirement of very powerful onboard processors.

4.4. Simulation Studies

In this section the HEV powertrain specification and several comparative HEV energy management strategies will be introduced and the simulation results will be provided and discussed.

Table 17 gives the powertrain specification of the HEV model used in this section. The HEV powertrain simulations are conducted after the driving data are collected on the driving simulator. The speed and torque demand from the tests will be used as the constraint for the HEV simulation.

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engine</td>
<td>3.2L Diesel engine</td>
</tr>
<tr>
<td>Motor</td>
<td>42kW, max torque 200Nm</td>
</tr>
<tr>
<td>Battery</td>
<td>2.2kWh, SOC range [0.2,0.8]</td>
</tr>
<tr>
<td>Transmission</td>
<td>6 speed automatic</td>
</tr>
</tbody>
</table>
The driver torque demand and engine speed collected from the driving simulator test will be used in this simulation section. In practice, the probability distribution of the random variables used in the developed approach should be obtained from the historical data. However, due to the limited number of simulator test trips, it is assumed that the probability distributions calculated from the data of the 24 trips are representative and they are consistent with the historical data. The only differences when applying this method between the real-world driving data and simulator driving data are the probability distributions of the random variables \( \Delta X_{SOC}[k, x_{SOC}(k), u] \) and \( \Delta F_{fu}[k, x_{SOC}(k), u] \). These parameters actually vary among different routes and different drivers. There is no absolutely correct shape of these distributions, so even all uncertainties are considered in the validation part of the chapter, the distributions of a different trip could be completely different. Therefore with evident uncertainties shown in Section II, using the data collected on the driving simulator can provide a convincible validation of the proposed method.

As an example, the speed and torque demands of one trip are given in Figure 29. The numbers above the figure stand for road segment numbers. The SDP-map based energy management strategy will be applied to all the 24 trips. As the stochastic approach can only show its benefit when applied to a large amount of data, the overall energy consumption will be used to compare with other strategies.
Three strategies are developed to be compared with the SDP. The first one is a simple ECMS-based charge-deplete-charge-sustaining (CDCS) strategy with a fixed electricity-to-fuel factor. In this strategy, the HEV runs using the fixed electricity-to-fuel factor until the battery SOC reaches its lower bound, which is 0.2. After that, the battery is used only when the SOC is above the lower bound. No route information or previewed information is need for this CDCS strategy and it is considered as a baseline strategy. The second strategy is a road-segment-based deterministic dynamic programming (DDP) strategy. It can be considered as the deterministic version of the stochastic approach developed in Section III. Instead of using the probability distribution, the determined value from a “typical trip” is used. The typical trip is selected with the total propulsive energy consumption as 8319 kJ, which is close to the average value of all trips, as shown
in Figure 21. The same discrete model structure and cost function structure are used, except that the determined SOC change will replace the random variable in the model, and the determined fuel consumption will replace the expectation of the fuel consumption in the cost function. DP is used to solve this optimal control problem. This strategy is also a real-time strategy as once the typical trip is chosen, its result is a 2-dimensional table that is in the same form as the SDP result. The SOC at the beginning of each road segment is the input of the DDP result map as feedback so it is adaptive to the actual trip. The comparison between the DDP strategy and the proposed SDP will show advantage of considering uncertainties over using information from one determined fix-routed trip. The third strategy is not a real-time strategy, but a conventional point-wise DP (PDP) with full knowledge of the future information of current trip second by second. The result of this point-wise DP is considered as the optimal result.

Figure 30 shows the final battery SOC of the 24 trips under different strategies. PDP, SDP, and CDCS can all deplete the battery by the end of the trip, while DDP does not deplete all the electric energy stored in the battery for many of the trips. Though PDP, SDP, and CDCS give similar final SOC results, the ways they consume the electricity energy are actually quite different. As an example, Figure 31 shows the battery SOC for one trip under different strategies. Both PDP and SDP tend to deplete the battery right at the end of the trip. But CDCS uses the battery intensively at the beginning of the trip, and the battery SOC drops to its lower bound very soon.
Figure 30. Final SOC of the 24 trips under different strategies [57]

Figure 31. Battery SOCs of one trip under different strategies [57]

Figure 32 provides the equivalent fuel costs of the 24 trips, considering both electricity consumption and fuel usage. SDP result is close to PDP. DDP is generally not too bad but does not perform well on some of the trips. CDCS is the worst strategies among the four.
Energy consumption comparison results are given in Table 18. Total energy consumption results of the 24 trips under different strategies. The total electricity energy used by PDP, SDP, and CDCS are very close, as they all have depleted the battery at the end of the trip. DDP uses less electricity, but more fuel than PDP and SDP. The total energy cost is the combined cost for electricity and fuel, using the price factor \( \lambda_x \) in the cost function (49). SDP consumes 1.8% more energy than the optimal PDP, while the DDP uses 3.1% more, and CDCS is 8.8% more than PDP. This result proves that the proposed stochastic approach can give very good performance when the fixed-route driving information is fully explored and utilized.
Table 18. Total energy consumption results of the 24 trips under different strategies

<table>
<thead>
<tr>
<th></th>
<th>PDP</th>
<th>SDP</th>
<th>DDP</th>
<th>CDCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuel consumption (L)</td>
<td>11.39</td>
<td>11.64</td>
<td>11.91</td>
<td>12.59</td>
</tr>
<tr>
<td>Electricity consumption (kWh)</td>
<td>31.59</td>
<td>31.48</td>
<td>30.23</td>
<td>31.55</td>
</tr>
<tr>
<td>Total energy cost (L)</td>
<td>13.50</td>
<td>13.74</td>
<td>13.92</td>
<td>14.69</td>
</tr>
<tr>
<td>Normalized total energy cost</td>
<td>100.0%</td>
<td>101.8%</td>
<td>103.1%</td>
<td>108.8%</td>
</tr>
</tbody>
</table>

4.5. Summary

In this chapter, the fixed-route driving data collected on a driving simulator are analyzed and the result shows that it is necessary to use stochastic approach to describe the fixed-route driving behavior. A stochastic model based on road segmentation is built for the HEV powertrain system dynamics. By assuming that historical driving data on the fixed route are available for computing the distribution of battery SOC change and fuel consumption under different strategy parameters, the energy management optimization problem can be solved using SDP offline. The SDP result is a 2-dimensional lookup table and the proposed two-level controller can be easily implemented onboard. Simulations show that the energy consumption of the proposed SDP-based energy management strategy is 1.8% more than the optimal result, while the deterministic methods cannot perform well when applied to various trips on the same route.
Chapter 5: Optimizing the Energy Management Strategy for Multi-Route Personal Plug-in HEVs

Based on the fixed-route stochastic optimized control strategy presented in Chapter 4, a new optimized strategy for multi-route plug-in HEV will be proposed in this chapter. This situation of vehicle operation is close to most personal vehicles, for which one or two routes may be dominant while other frequent routes are also possible. The problem for multi-route plug-in HEV energy management strategy optimization is formulated. The future route uncertainty and road-segment-based random variables make this optimal control problem a little special. A theorem, which is inspired by the method in Chapter 3 is proposed and proved to show that it can still be solved via backward induction. Simulation studies are conducted using real-world routes and the driver model developed in Chapter 2. Results show that the proposed method can design an optimized energy management strategy which consumes less than 2.5% more energy than the posteriori optimal result, while the best constant ECMS strategy will use about 7.1% more energy than the optimal result.

5.1. Multi-Route Personal HEVs

In reality, almost no personal car runs only on a fixed route. However, for most personal cars, it may be possible to find several frequent routes from historical trip data.
This chapter will focus on optimizing the energy management strategy for plug-in HEVs running on multiple routes. All the possible routes are known a priori, and the controller has historical data of vehicle running on all these routes. The probability of running on a certain route is known. The controller does not know which route the driver will drive on if a future intersection may lead to different routes. But the current position of the vehicle can be used to narrow down the possible routes that the vehicle is going to run on. This information will be used to determine the best higher-level energy management strategy under the two-level control structure described in Section 4.3.1.

It is assumed that after each trip, the vehicle is fully-charged at home overnight, and not charged anywhere or anytime else. Therefore if the vehicle goes to several destination, e.g., work, grocery, kid’s school, it is considered as a new route, differently from the round-trip route to work or grocery alone. This chapter will focus on optimizing the energy consumption on these frequent routes. If the vehicle is taking a new route, the default energy management strategy will be used and this part of energy management will not be discussed here. The current position of the vehicle can be used to narrow down the possible routes that the vehicle is going to take.

5.1.1. Describing Different Routes

A route contains many road segments. However, slightly different from the definition in Chapter 4, the road segment defined in this chapter may start or end right at an intersection, such that different segment sequence can be used to construct different routes.
Denote a road segment as $s_j$, where $j$ is an index number. It should be noted that different direction of a road segment are considered as different segment, as the road grade and traffic light signal information can be different on a same road for two directions.

A route can be denoted as a sequence of road segments,

$$ r = \{s_{j_1}, s_{j_2}, \ldots, s_{j_m}\}. $$  \hfill (59)

All frequent routes can also be indexed, as $r_1, r_2, \ldots, r_n$. Each route may contain different total numbers of segments, which are denoted as $m_1, m_2, \ldots, m_n$. From historical data, the probability of running on each route is known as $p_1, p_2, \ldots, p_n$, and $\sum_{i=1}^{n} p_i \leq 1$. The index of the k-th element of a route is denoted as $r(k)$, for example, in the route described in (59), $r(1) = j_1$.

When the vehicle leaves home, it is unknown which route the driver is going to choose. The indices of all the possible routes form a set $I$,

$$ I = \{ \text{All } i, \text{ if route } r_i \text{ is possible} \}. $$

The initial knowledge of possible routes is $I_0$. Similarly to the disturbance sequence problem described in Section 3.2.2, at each step $k$, the information of $I$ will be updated as $I_k$, and $I_k$ is always shrinking as shown (22).

\subsection*{5.1.2. Problem Formulation}

As shown in Section 4.3.3, it is assumed that the battery SOC change $\Delta X_{SOC}$ and fuel consumption $\Delta F_{fu}$ in each segment are random variables, whose probability
distributions will be affected by the input \( u \). As the road segment may start and end at an intersection, \( \Delta x_{\text{SOC}}[k, x_{\text{SOC}}(k), u] \) and \( \Delta x_{\text{SOC}}[k + 1, x_{\text{SOC}}(k + 1), u] \) are not independent. But simulations results will show that even if this dependence is ignored, the optimized energy management strategy performs well.

The road-segment-based HEV model and energy cost functions are similar to the one shown in (52)(53)(54) in Chapter 4. However information about the road segment \( s_j \) (which is not just the “road segment number”, i.e., step number \( k \), in the fixed-route problem) will also be included in the system dynamics and running cost,

\[
x_{d,\text{SOC}}(k + 1) = x_{d,\text{SOC}}(k) + \Delta X_{\text{SOC}}[k, x_{d,\text{SOC}}(k), j, u],
\]

(60)

\[
l_s[k, x(k), j, u(k)] = \mathbb{E}\{\Delta F_{fu}[k, x_{d,\text{SOC}}(k), j, u(k)]\}.
\]

(61)

A terminal cost is defined for each route \( r_i \),

\[
J_f[x(N), i] = J_f[x(m_i + 1), i] = \lambda_f[x_{d,\text{SOC}}(m_i + 1) - x_{d,\text{SOC},0}].
\]

(62)

The initial condition for the battery SOC \( x_{d,\text{SOC}}(0) \) is assumed to be the same for every trip. Define the feedback control policy as \( \pi = \{u_0(\cdot), u_1(\cdot), \ldots\} \). For every known route \( r_i \), the target is to minimize the total energy cost,

\[
J_f(\pi, r_i) = J_f + \sum_{k=1}^{m_i} l_s(k),
\]

(63)

where \( j = r_i(k) \) is used to calculate the trajectory and cost in (60)(61) and (62).

With uncertain route information, target of the optimal control problem is to minimize expected the total energy cost, where the expectation energy cost value can be
considered as the average energy consumption in the long run under a certain control strategy.

\[ J(\pi, I_0) = \mathbb{E}[J_r(\pi, r)] = \sum_{i \in I_0} \left\{ p_i \{ J_f[x(m_i + 1), i] + \sum_{k=1}^{m_i} l_s(k) \} \right\}, \]

(64)

where \( j = r(k) \) when implementing (60)(61)(62) for the state trajectory and cost. In this problem formulation, \( \pi \) uses the state \( x_{d,SOE}(k) \) and route information \( I_k \) as feedback.

5.2. Solution of the Proposed Problem

The problem is slightly different from a conventional stochastic optimal control problem. The road segment (denoted as the segment index \( j \)) can be considered as a disturbance in (60), but this disturbance has several special properties: 1) it affects the stochastic system dynamics, 2) it affects the cost, 3) its sequence belongs to a known set, 4) this set is updated (shirking) at every step, and 5) the sequences in this set may be of different lengths. A theorem will be proved to show that backward induction can be used to solve this problem.

Before introducing the theorem, the cost-to-go function with its initial condition \( x(k) \) is defined as

\[ V[k, x(k), \pi_k, I_k] = \mathbb{E}_i \left\{ V[k, x(k), \pi_k, I_k] \right\} = \sum_{i \in I_k} \left\{ p_i \{ J_f[x(m_i + 1), i] + \sum_{q=k}^{m_i} l_s(q) \} \right\}. \]

(65)

The control policy \( \pi_k \) that minimizes \( V[k, x(k), \pi_k, I_k] \) (which only contains feedback input mapping after step \( k \)) is called the optimal remaining control policy.

Define
\[ W(k, \pi, I_k) = \sum_{i \in I_k} \left\{ p_i [\sum_{q=0}^{k-1} l_i(q)] \right\}. \] (66)

\( W(k, \pi, I_k) \) is the running cost from the beginning to step \( k \) under control policy \( \pi \) if the route (i.e., the road segment index sequence) belongs to \( I_k \). Every route in \( I_k \) is required to have the same road segment before step \( k \).

5.2.1. A Theorem

**Theorem 2**: Given the system and cost described in (60)(61)(62)(63)(64), if \( \pi^* \) is an optimal control policy that minimizes \( J(\pi, I_o) \), and \( \pi_k^* \) is an optimal control policy for problems starts at \( x(k) \) from step \( k \) and minimizes \( V[k, x(k), \pi_k, I_k] \), a new control policy \( \tilde{\pi} \) is constructed by replacing \( u^* \) in \( \pi^* \) with \( u_k^* \) in \( \pi_k^* \) when possible, which means, \( \tilde{\pi} : u_k^*(k, x, I) \) if \( (k, x, I) \) is defined for \( u_k^*(\cdot) \), and \( \tilde{\pi} : u_k(k, x, I) \) if \( (k, x, I) \) is not defined for \( u_k^*(\cdot) \) but defined for \( u_k(\cdot) \). Then \( \tilde{\pi} \) is also an optimal control policy that minimizes \( J(\pi, I_o) \).

Similarly to Lemma 1, it can be seen that if \( i \notin I_k \), \( J_i(\tilde{\pi}, \pi^*, r) \).

**Proof of Theorem 2**: 
\begin{align*}
J(\tilde{\pi}) &= \sum_{i \in I_k} \{ p_i[J_i(\tilde{\pi}), \tilde{\pi}] \} \\
&= \sum_{i \in I_k} \{ p_i[J_i(\tilde{\pi}), \tilde{\pi}] \} + \sum_{i \in I_k} \sum_{i \in I_0} \{ p_i[J_i(\pi^*, r_i)] \} \\
&= W(k, \tilde{\pi}) + \sum_{i \in I_k} \{ p_i[V_i[k, x(k), \pi^*, I_k]] \} + \sum_{i \in I_k} \sum_{i \in I_0} \{ p_i[J_i(\pi^*, r_i)] \} \\
&\leq W(k, \pi^*, I_k) + \sum_{i \in I_k} \{ p_i[V_i[k, x(k), \pi^*, I_k]] \} + \sum_{i \in I_k} \sum_{i \in I_0} \{ p_i[J_i(\pi^*, r_i)] \} \\
&= \sum_{i \in I_k} \{ p_i[J_i(\pi^*, r_i)] \} + \sum_{i \in I_k} \sum_{i \in I_0} \{ p_i[J_i(\pi^*, r_i)] \} \\
&= \sum_{i \in I_k} \{ p_i[J_i(\pi^*, r_i)] \} \\
&= J(\pi^*, I_0)
\end{align*}

The third equality sign is due to Lemma 1, the fourth equality sign is due to the definition of the remaining problem, the fifth and sixth equality sign is due to the construction of \(\tilde{\pi}\). The less-than-or-equal-to sign is due to that \(\pi_k^*\) is optimal.

Therefore \(\tilde{\pi}\) is also an optimal control policy that minimizes \(J(\pi, I_0)\).

### 5.2.2. Backward Induction

Based on Theorem 2, the optimal control strategy \(\pi^*\) can be constructed via backward induction. The solving procedure is as follows.

1) Set the initial condition for the backward induction.

\[ V(m_{\text{max}} + 1, x(m_{\text{max}} + 1), \pi^*_{m_{\text{max}} + 1}, I_{m_{\text{max}} + 1}) = J_f[x(m_{\text{max}} + 1), I_{\text{max}}] \], where \(m_{\text{max}}\) is the maximum
length of road segment sequences in the set $I_0$, and $i_{\text{max}}$ is the index of this longest sequence. $\pi^*_{\text{max}}$ is null (no control is needed at the terminal step).

2) Conduct backward induction. Suppose that $\pi^*_{k+1}$ and $V(k+1, \pi^*_{k+1}, I_{k+1})$ is obtained. The one-step input $u(k, x_k, I_k)$ can be found from the following equation,

$$u[k, x(k), I_k] = \arg \min_{u \in U(x_k)} \left\{ \sum_{i \in I_k} p_i \{ I_i[k, x(k), r_i(k), u(k)] + V[k+1, x(k+1), \pi_{k+1}, I_{k+1}] \} \right\}, \quad (68)$$

and

$$V[k, x(k), \pi_k, I_k] = \min_{u \in U(x_k)} \left\{ \sum_{i \in I_k} p_i \{ I_i[k, x(k), r_i(k), u(k)] + V[k+1, x(k+1), \pi_{k+1}, I_{k+1}] \} \right\}. \quad (69)$$

The backward induction is conducted until step 0, then the complete optimal control strategy $\pi^*$ is constructed.

5.3. Real-Time Implementation

The implementation of the proposed method here is similar to the one in Chapter 4.3.5. Off-line calculation using historical data will generate a map whose inputs are step number $k$, battery SOC at the beginning of a road segment, and future possible route information $I_k$. The output of this map is the optimized lower-level real-time controller parameters, e.g., the electricity-to-fuel factor in ECMS. The entire on-line strategy, including the map-lookup and lower-level control can be implemented without the requirement of very powerful onboard processors.
5.4. Simulation Studies

5.4.1. Vehicle Model

The vehicle model used in this study is the same as the model used in Section 4.4, except that the battery is 8.8 kWh.

5.4.2. Route Maps

Eight different routes are selected for the simulation. These routes cover both local and highway driving. The road map that the routes cover is shown in Figure 33. Figure 34 provides detailed look into these routes. The beginning section of all eight routes are the same, which means in this section the vehicle controller will have no information about which route the vehicle is going to choose. In the early sections, the controller has limited information about the route because many possible routes share the same early sections. Route 5 contains two destinations, to simulate the situation that the driver goes to grocery on the way from work to home. Therefore the controller cannot differentiate Route 5 and Route 3 (simulating going to work only) until very late of the trips.
Figure 33. Road map that the simulation routes cover

Figure 34. Eight routes in the simulation
Table 19 shows some route information. The length of eight routes varies from 4 km to 27.8 km. The approximate trip time and average speed only considers the period when the vehicle is moving, as it is assumed that the vehicle has start-stop system such that the time when the vehicle speed is zero has little impact on the energy consumption. The maximum number of stops shows all possible stops assuming that every traffic light the vehicle encounters is red. The actual number of stop is less than the maximum number.

<table>
<thead>
<tr>
<th>Route</th>
<th>Length (km)</th>
<th>Approximate trip time (min)</th>
<th>Maximum number of stops</th>
<th>Maximum speed (km/h)</th>
<th>Average speed (km/h)</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route 1</td>
<td>27.8</td>
<td>24-27</td>
<td>30</td>
<td>88.6</td>
<td>64.8</td>
<td>0.05</td>
</tr>
<tr>
<td>Route 2</td>
<td>10.7</td>
<td>12-14</td>
<td>19</td>
<td>72.4</td>
<td>49.8</td>
<td>0.1</td>
</tr>
<tr>
<td>Route 3</td>
<td>17.5</td>
<td>18-21</td>
<td>36</td>
<td>72.5</td>
<td>54.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Route 4</td>
<td>4.0</td>
<td>5-6</td>
<td>10</td>
<td>56.4</td>
<td>40.2</td>
<td>0.05</td>
</tr>
<tr>
<td>Route 5</td>
<td>21.1</td>
<td>22-28</td>
<td>42</td>
<td>72.5</td>
<td>53.8</td>
<td>0.1</td>
</tr>
<tr>
<td>Route 6</td>
<td>5.6</td>
<td>6-8</td>
<td>12</td>
<td>72.5</td>
<td>48.9</td>
<td>0.1</td>
</tr>
<tr>
<td>Route 7</td>
<td>14.8</td>
<td>15-17</td>
<td>22</td>
<td>72.5</td>
<td>55.5</td>
<td>0.25</td>
</tr>
<tr>
<td>Route 8</td>
<td>6.6</td>
<td>8-9</td>
<td>12</td>
<td>56.4</td>
<td>45.3</td>
<td>0.05</td>
</tr>
</tbody>
</table>

5.4.3. Trip Data

Both historical trip data and validation trip data are generated using the driver model presented in Chapter 2. First, 200 trips are simulated in the driver-vehicle-road closed-loop as historical data, in which the frequencies of each route is the probabilities shown in Table 19. These trips are used to obtain the data for the random variables $\Delta Y_{SOC}$ and $\Delta F_{fu}$ on each road segment, as shown in Figure 26 and Figure 27 in Section 4.3.3. Then, another 200 trips are simulated with the same frequencies of each route as method
validation data. The proposed method, and a constant factor ECMS, and posteriori DP will be applied to these 200 trips to compare the energy consumption results.

5.4.4. *Simulation Results*

ECMS is used as the lower-level controller. In the backward induction, both the state and input are discretized. The best constant-factor ECMS is obtained by searching different electricity-to-fuel factors, as shown in Figure 35.

![Figure 35. ECMS equivalent energy cost under different factors](image)

The energy consumption result is shown in Table 20. It can be seen that using the proposed method, the total energy cost is only 2.5% more than the posteriori optimal result, while the best constant-factor ECMS consumes 7.1% more energy than the optimal result.
Table 20. Total energy consumption results of the 200 trips under different strategies

<table>
<thead>
<tr>
<th></th>
<th>DP</th>
<th>Proposed method</th>
<th>Best ECMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuel consumption (L)</td>
<td>44.5</td>
<td>47.2</td>
<td>52.4</td>
</tr>
<tr>
<td>Electricity consumption (kWh)</td>
<td>838.6</td>
<td>825.6</td>
<td>801.0</td>
</tr>
<tr>
<td>Total equivalent energy cost (L)</td>
<td>85.3</td>
<td>87.4</td>
<td>91.4</td>
</tr>
<tr>
<td>Normalized total energy cost</td>
<td>100.0%</td>
<td>102.5%</td>
<td>107.1%</td>
</tr>
</tbody>
</table>

5.5. Summary

In this chapter, the fixed-route HEV energy management strategy using historical data proposed in Chapter 4 is extended to the multi-route cases. The unknown route information is shown as the unknown future road segments, which is considered as disturbance sequences. An optimal control problem whose target is to minimize the expectation of total energy cost is formulated. It is proved in this chapter that the backward induction can be used to construct the optimal higher-level energy management controller. Simulation using the driver model presented in Chapter 2 shows that under the proposed control strategy the plug-in HEV consumes only 2.5% more than the posteriori optimal result.
Chapter 6: Route-Based Global Energy-Optimal Speed Planning

This chapter provides a globally optimal solution to an important problem: given a real-world route, what is the most energy-efficient way to drive a vehicle from the starting point to the destination within a certain period of time, under the ideal situation where there are no uncertainties in the trip. Along the route, there may be multiple stop signs, traffic lights, turns and curved segments, roads with different grades and speed limits, and even leading vehicles with pre-known speed profiles. Most of such route information and features are actually constraints to the optimal vehicle speed control problem, but these constraints are described in two different domains. The most important concept in solving this problem is to convert the distance-domain route constraints to some time-domain state and input constraints that can be handled by optimization methods such as DP. Multiple techniques including cost-to-go function interpolation and parallel computing are used to reduce the computation of DP and make the problem solvable within a reasonable amount of time on a personal computer. Simulation of an electric vehicle running on an 8-km city route shows that the optimal speed trajectory can result in a significant energy saving (up to 19%) comparing to a nominal speed profile. This chapter is mainly based on the paper [95] which is submitted to a journal and under review.
6.1. Vehicle Speed Planning

Improving the energy efficiency of road vehicles is one of the most important tasks for automotive powertrain control. Researchers have found that on the same route, different speed profiles caused by different driving styles can result in significant energy consumption differences [96][97]. There are potentials in saving energy by adjusting the speed without too much sacrifice in comfortability, drivability, or travel time. This motivates the concept of eco-driving [98][99][100], where initially the drivers were suggested to follow some driving rules and patterns, e.g., avoiding fast accelerations and hard brakes, to achieve lower energy consumption. There are also approaches trying to design a reference speed in real-time and provide it as guidance to the driver [101][102].

In the context of automated vehicles and semi-automated vehicles (e.g. vehicles with adaptive cruise control [103]), the speed planning and control methods also need to consider the energy efficiency. All kinds of information related to the future trip, including the traffic condition, traffic light schedules, and road maps, can be utilized to design the best speed trajectory.

Many researchers have been working on figuring out an optimal or sub-optimal speed trajectory for vehicles running in various situations, both on-line and off-line. Road grade information is utilized in some literatures. The analytical solution to minimize the energy consumption for vehicles running on roads with variable grades is proposed in [104]. A look-ahead vehicle control considering road grade is developed in [105]. The optimal speed planning problem with a single traffic light is solved in several literatures. Analytical solution for vehicles running through a traffic light is proposed in [106]. DP is
used to solve a similar single traffic light problem in [107]. Optimal vehicle speed trajectory for a traffic light arterial with consideration of a queue is presented in [108]. Under the situation of multiple traffic lights, many existing approaches can find a sub-optimal solution, which first tries to avoid red lights, then optimizes the speed [109]. A predictive control utilizing multiple traffic light signals to improve fuel economy is designed in [110]. A multi-stage energy-optimal speed control for multiple traffic lights is proposed in [111]. Describing the speed planning problem in distance domain instead of time domain may bring some advantages, especially when dealing with stop signs, which is not easy to describe in time domain. Distance domain methods for energy management with road grade terrain preview are proposed in [112]. A similar approach considering traffic lights is proposed in [113], where it focuses on finding a speed trajectory that avoids red lights. When the vehicle encounters a red light, the speed profile will be adjusted iteratively until it avoids the red light, therefore no global optimality is guaranteed.

To the best of the authors’ knowledge, there is not yet a good way to find the global optimal speed planning solution on a route with multiple traffic lights, stop signs, and different road grades. The main difficulty is that the constraints of this optimal control problem are described in two different domains. The traffic light locations, stop sign locations and speed limits are described in the distance domain, while the traffic light schedules are described in the time domain. If the problem is formulated in time domain, as discussion in Section 6.3 will show that, the distance domain constraints are not in a standard form and they cannot be handled directly by DP or Pontryagin’s Minimum
Principle. If the problem is formulated in the distance domain, the time-domain traffic light schedule information cannot be taken into consideration.

In this chapter, the global optimal speed planning solution for vehicles on a given route is presented. The route may include multiple stop signs, traffic lights, turns and curved segments, roads with different grades and speed limits, and even leading vehicles with known speed profiles. As a major contribution of this work, the non-standard-form distance-domain constraints are converted to state and input constraints in the standard form in time domain. Then this comprehensive speed planning problem can be solved via DP. Multiple techniques are used to reduce the computational load of DP and the algorithm can run on a personal desktop computer within a reasonable amount of time.

Obviously, the method proposed here is not a real-time method. It requires a considerable computational resource and it does not consider any uncertain factors along the route. But it is a very useful tool that can calculate the posteriori optimal speed trajectory and optimal energy consumption. The result can be used to evaluate the performance of other speed planning methods, in either eco-driving guidance applications or automated / semi-automated vehicle controls, just like in hybrid electric vehicle energy management study, the posteriori DP results can be used to evaluate other real-time control strategies. This method can also be used to investigate the potential of energy saving for any given trips with proper route information. The concept of the constraint conversion is also important and may inspire some future real-time capable algorithms. It should also be noted that the advantage of this speed planning method cannot be shown in any driving cycle tests because it tries to change the speed profile itself.
6.2. Problem Formulation

In this section, the vehicle model will be introduced first. Then the optimal control problem of a vehicle running on a given route is formulated.

6.2.1. A General Vehicle Model

A general vehicle energy consumption model is built based on the following equations,

\[ \dot{v} = \frac{\tau / r - mg \sin(\theta) - C_d v^2 - \mu g \cos(\theta) \text{sign}(v)}{J / r_w^2 + m}, \]  \hspace{1cm} (70)

\[ P = \frac{T_d \eta(\nu, T_d)}{r_w}, \]  \hspace{1cm} (71)

\[ E = \int_0^{t_f} Pdt, \]  \hspace{1cm} (72)

where \( d \) is the distance, \( v \) is the vehicle speed, \( T_d \) is the total propel torque at the wheels, \( r_w \) is the radius of the wheel, \( m \) is the vehicle mass, \( g \) is the gravitational constant, \( \theta \) is the road grade, \( C_d \) is a constant coefficient for wind resistance, \( \mu \) is the rolling resistance coefficient, \( J \) is the rotational inertial for the wheel, transmission and powertrain, \( P \) is the power consumed by the energy source, \( \eta \) is the energy efficiency of the vehicle powertrain, \( E \) is the total energy consumption, and \( t_f \) is the final time.
In this model, it is assumed that the instantaneous energy efficiency $\eta$ is only dependent on the vehicle speed $v$ and torque $T_d$. This model can be used for an electric vehicle with one-gear transmission and slow battery SOC change so that the efficiency is only dependent on the vehicle speed and motor torque, or for an internal combustion engine vehicle whose gear number is only dependent on the vehicle speed and engine torque. Additional states, including the battery SOC for electrified vehicle, gear number for vehicles with transmissions, or planetary gear speed for vehicles with planetary gearset transmission, can be added to this model. The additional state may increase the computation time when solving the optimal control problem via DP, but the framework of the proposed method is still valid.

By assuming that the acceleration during each sampling period is a constant, the continuous model described in (70)(71)(72)(73) can be approximated by the following discrete model.

If $v(k) + a(k)\Delta t \geq 0$,

$$d(k + 1) = d(k) + v(k)\Delta t + \frac{1}{2}a(k)\Delta t^2,$$  \hspace{1cm} (74)

$$v(k + 1) = v(k) + a(k)\Delta t,$$  \hspace{1cm} (75)

$$P(k) = g_p[a(k), v(k), d(k)],$$  \hspace{1cm} (76)

$$E = \sum_{0}^{L} P(k),$$  \hspace{1cm} (77)
where \( k \) is the step, \( a \) is the acceleration, \( \Delta t \) is the sampling period, \( g_p(\cdot) \) is the function for power, and \( k_f \) is the final step. Regenerative braking (when available) is also considered in the function of \( g_p(\cdot) \).

As the vehicle speed considered in the problem is always non-negative, Equation (74) and (75) are only valid when \( v(k) + a(k)\Delta t \geq 0 \). When \( v(k) + a(k)\Delta t < 0 \), it is assumed that the vehicle speed remains zero for the rest of the sampling period after it stops. Therefore in this situation, Equation (74) and (75) are replaced by the following equations and \( g_p(\cdot) \) is also adjusted accordingly.

If \( v(k) + a(k)\Delta t < 0 \),

\[
d(k+1) = d(k) - \frac{v(k)^2}{2a(k)},
\]

(78)

\[
v(k+1) = 0.
\]

(79)

6.2.2. The Route-Based Optimal Control Problem

Let the state vector be \( x = [x_1 \ x_2]^T = [d \ v]^T \), and the input as \( u = a \), Equation \((74)(75)(76)(77)(78)(79)\) form an optimal control problem,

\[
\begin{align*}
x(k+1) &= f_k[x(k),u(k)], \\
J &= \sum_{o}^{k_f} g_k[x(k),u(k)],
\end{align*}
\]

(80)

(81)

subject to various constraints,

where \( f_k(\cdot) \) is the system dynamic function, and \( g_k(\cdot) \) is the running cost function.
In this formulation, the terminal step $k_f$ is given. For the vehicle energy minimization problem, the optimal solution for a fixed $k_f$ is actually the most energy-efficient trip for any terminal step $k \leq k_f$, as the vehicle is allowed to arrive at the destination early and stay there until the final step.

It should also be noted that for electric vehicles, $g_k[x(k),u(k)]$ can be designed to be dependent on $k$ to approximate the energy efficiency change as the battery SOC drops along the trip, without adding one more state to the system.

It is assumed that all route information and features, including road grades, speed limits, stop sign and traffic light positions, light schedules, and even leading vehicle position and speed etc., are known. These data can be obtained by assuming ideal situation where there is no uncertain traffic, or by collecting data from a real trip and conducting the calculation posteriori. This problem seems to be a classical optimal control problem, but the difficulty lies in describing and handling the constraints from two different domains, which will be discussed in the next section.

6.3. The Constraints of the Problem

This section analyzes the constraints of the proposed optimal control problems and describes them all in standard constraint forms.

Before formulating the constraints for this specific route-based problem, a review of standard DP constraints are given. In general, DP can handle various nonlinear constraints, in which most common ones are in the following two forms.
1) State constraint: given \( k \), \( x(k) \in X_k \). Initial condition \( x(0) = x_0 \) and final condition \( x(k_f) = x_{k_f} \) are both in this form. State boundaries can also be described in this form.

2) Input constraint: given \( k \), \( u(k) \in U_k(x_k) \). This constraint means that the range of input is dependent on the state.

Some of the route-based optimal control problem constraints are directly formulated in one of these two forms, while some others are not. A transformation is needed to convert those non-standard constraints into the standard forms.

6.3.1. Standard-Form Constraints

In our route-based optimal control problem, some of the constraints can be easily described in the standard forms.

1) The initial condition and the final condition, which are

\[
 x(0) = [0 \ 0]^T, \tag{82}
 \]

and

\[
 x(k_f) = [d_f \ 0]^T, \tag{83}
 \]

where \( d_f \) is the distance of the destination.

2) The acceleration constraint. In general, the range of the input, which is the range of acceleration, is dependent on the vehicle speed and road grade. This is essentially in the standard input constraint form, as the speed is part of the state, and the road grade is a known function of the distance, which is also part of the state. But in this energy
minimization problem, this constraint can be simplified. It is not necessary to consider the extreme conditions where the acceleration is pushed to its limit. It is assumed that the acceleration should always stay within the range where the driver and/or passengers feel comfortable, and the boundary of this range can be achieved during the trip. Therefore, this constraint can be described as,
\[ u(k) \in U_{co} = [a_{\text{min}}, a_{\text{max}}], \forall k. \] (84)

3) Speed limits. The speed limit of road can be described as
\[ v(d) \leq V_{sl}(d), \] (85)
where \( V_{sl}(d) \) is a known piecewise-constant function of the distance \( d \). This is essentially a subset of the state space, which means it is in the standard form,
\[ x(k) \in X_{sl}, \forall k. \] (86)

4) Speed lower bound. Similarly, a lower bound of speed can be defined as a constraint so that the vehicle will not stop or run unreasonably slowly in the middle of the road. This lower bound is related to the traffic light schedule, which means that the constraint set is time-varying,
\[ v(d,k) \geq V_{lo}(d,k). \] (87)
This constraint is still in the standard state constraint form,
\[ x(k) \in X_{lo}(k). \] (88)

5) The leading vehicle constraint. If there is a leading vehicle with a pre-known speed profile, described as a trajectory \( x_c(k) = [d_c(k), v_c(k)]^T, k = \{k_1, k_2, \ldots, k_n\} \) (the leading vehicle may appear only during part of the trip). Whether the vehicle is required
to keep a constant distance to the leading car, or a constant time-to-collision (TTC) to the leading vehicle, this can always be described as a state constraint, as either

\[d(v,k) \leq d_c(k) - d_0, \quad (89)\]

or

\[d(v,k) \leq d_c(k) - vt_{TTC}, \quad (90)\]

can be written in the standard state constraint from,

\[x(k) \in X_c[x_c(k)], k = \{k_{c1}, k_{c2}, ..., k_{cn}\}. \quad (91)\]

Figure 36 gives an example of the speed limit constraint, the speed lower bound constraint, and the lead vehicle constraint at a certain step shown as a subset of the state space. The colored region shows \(X_{sl} \cap X_{lo}(k) \cap X_c[x_c(k)]\). It shows that the speed limit constraint, speed lower bound constraint, and constant TTC constraint are all constraints in standard form, which can be illustrated as a subset of the state space. These constraints are relatively easy to describe and can be directly handled by DP.

Figure 36. An example of Some state constraints [95]
6.3.2. Non-Standard-Form Constraints

The stop sign constraint is in a different form, which can be described as,

$$\exists k, \text{ such that } x(k) = [d_{st}, 0]^T.$$  \hspace{2cm} (92)

Even if the exact stop position requirement can be relaxed, it is still essentially requiring the trajectory of the system go through a certain set, and the time of this going-through is free. It is not a constraint in the standard form which can be handled by DP. If the problem is discretized in the distance domain instead of time domain, this constraint would be a simple state constraint as $v(d_{st}) = 0$. But distance-domain discretization cannot be used if the traffic light schedules are to be considered. New approaches are needed to deal with this constraint. Actually, some other constraints including traffic lights and turns all require new ways to convert them into the standard forms so that the problem can be solved by DP.

The conversion of the stop sign constraint is the key concept of this approach, which will be introduce first.

1) Stop sign. Intuitively, a stop sign can also be described in the form of state constraint,

$$x(k) \in \left\{ \begin{array}{c} [d, v]^T \mid \text{if } \begin{array}{c} d = d_{st}, \nu = 0 \end{array} \end{array} \right\}, \forall k.$$ \hspace{2cm} (93)

But this cannot guarantee that the vehicle stops at the stop sign in the discretized model. It is possible that the state trajectory will just “skip” the stop sign position, as shown in Figure 37. The state may skip the stop sign if only (93) is considered. The colored region excluding the blue dash line in the middle (which is at the stop sign
position \( d_{st} \) is the feasible region considering (93) and the speed lower bound. The red circle is a possible state trajectory of the discrete model. The trajectory satisfies the constraints, but the vehicle does not stop.

![Diagram](image)

Figure 37. An example of the state skipping the stop sign [95]

In our problem, the distance \( d \) is always non-decreasing. If the vehicle is going to stop at the distance \( d_{st} \), due to the acceleration constraints (84), it has to start to decelerate before the stop sign. The speed at a position before the stop has to meet the following constraint,

\[
\text{if } d < d_{st}, \quad v^2 \leq 2 |a_{\min}| (d_{st} - d). \quad (94)
\]

Similarly the speed after the stop has to satisfy the following inequality,

\[
\text{if } d > d_{st}, \quad v^2 \leq 2 |a_{\max}| (d - d_{st}). \quad (95)
\]

Equation (94) and (95) are actually in the form of standard state constraints. However they are not sufficient to avoid the skip phenomenon, as shown in Figure 38. The colored region in this map is the feasible state at a stop sign considering (94), (95) and the speed lower bound. Inputs constraints are necessary because without them, some
potential inputs (e.g. \( u_1 \), \( u_2 \), \( u_3 \) and \( u_4 \)) may drive the next-step state directly to a feasible point after the stop sign without stopping.

![Figure 38. Another example of the state skipping the stop sign [95]](image)

Therefore only state constraints cannot guarantee that the vehicle stops at the stop sign. Additional constraints on inputs are needed to ensure that \( x(k+1) \) will not skip the stop. Therefore if \( x(k) \) is on the left half of feasible region excluding the exact stop point in Figure 38, \( f_k(x(k),u(k)) \) should also be on the left half of feasible region including the exact stop point. This is actually equivalent to an input constraint which can be converted to the following standard form,

\[
u(k) \in [a_{\min}, u_{st,\max}(d,v)], \forall k.
\] (96)

In the numerical DP implementation, these constraints are slightly relaxed to avoid the situation that the vehicle cannot stop exactly at the stop sign point because only a finite number of possible inputs are searched.

If \( d < d_{st} \), \( v^2 \leq 2 |a_{\min}| (d_{st} - d + \Delta d) \),

\[
\text{if } d > d_{st}, \ v^2 \leq 2 |a_{\max}| (d - d_{st} + \Delta d),
\] (97) (98)
where $\Delta d$ is small. The relaxed constraints are shown in Figure 39. The input constraint is also relaxed as if $x(k)$ is in Region 1, $f_u[x(k),u(k)]$ should be in Region 1 or Region 2. This is sufficient to guarantee that the state trajectory will always go through Region 2. Every state inside Region 2 is considered as a virtual stop. Therefore the vehicle stops or at least reach a very low speed near the stop sign, in which case the energy consumption and travel time is almost the same as a full stop, so the resultant optimal trajectory still makes sense.

![Figure 39. Relaxed state and input constraints for numerical DP implementation [95]](image)

Equation (94)(95) and the input constraint described above can be formulated in the standard form,

$$x(k) \in X_{st}, \forall k,$$  \hspace{1cm} (99)

$$u(k) \in U_{st}[x(k)], \forall k.$$  \hspace{1cm} (100)

It should also be noted as that it is not necessary to explicitly derive the analytic formulations for $X_{st}$ and $U_{st}$, because when implementing DP in the computer, all searches are numerical, therefore using (94), (95) and relaxed input rule in Figure 39 (in
numerical forms) for $x(k)$ and $f_k[x(k),u(k)]$ are sufficient to determine if a given state or input meets the constraints. However, in some cases explicit forms of state and input constraint sets may save computation time.

2) Traffic lights. When the traffic light is red, at the positions before the traffic light, the constraints are the same as a stop. There is no constraint at the positions after the traffic light. When the traffic light is green, it is considered as no constraint at all. The yellow light period can be considered as red for safety. This is just a time-varying version of the stop sign constraints,

$$x(k) \in X_{d\ell}(k),$$  \hspace{1cm} (101)

$$u(k) \in U_{d\ell}[x(k),k].$$  \hspace{1cm} (102)

3) Turns and curves. The vehicle is required to reduce its speed when going through curves or making turns. It is assumed that the maximum speed $v_{mu}$ that the vehicle can comfortably make a turn is known. These constraints are again similar to the stop signs, where the only difference is that the target speed at the turn is $v_{mu}$ instead of zero. If it is a long curve, the upper bound of the speed at every point along the curve is also known. Similarly to the stop signs, all these constraints can be described in the standard form,

$$x(k) \in X_{mu}, \forall k ,$$  \hspace{1cm} (103)

$$u(k) \in U_{mu}[x(k)], \forall k .$$  \hspace{1cm} (104)
6.4. DP Implementation

The route-based optimal control problem can be restated as:

*Given the system dynamic equation (80), minimize the cost function (81), subject to the constraints (82)(83)(84)(86)(88)(91)(99)(100)(101)(102)(103)(104).*

This problem is now solvable via DP. The state constraints can all be replaced by the input constraints \( u(k) \in U_x[x(k)] \) which can ensure that if \( x(k) \) meets the state constraints, \( x(k+1) \) also meets the state constraints. DP can be solved via backward induction,

\[
V_k[x(k)] = \min_{u(k) \in U_x[x(k)]} \{ g_k[x(k), u(k)] + V_{k+1} \{ f_k[x(k), u(k)] \} \},
\]

\[
u_k[x(k)] = \arg \min_{u(k) \in U_x[x(k)]} \{ g_k[x(k), u(k)] + V_{k+1} \{ f_k[x(k), u(k)] \} \},
\]

where \( V_k(\cdot) \) is called the cost-to-go function, which is the minimum possible accumulated cost from step \( k \) to the end, given the state \( x \) at step \( k \).

The computational requirement for solving this DP problem is high. Several techniques are used to reduce the computation time.

*6.4.1. Interpolation of Cost-to-Go Functions*

The core concept of implementing DP is calculating the cost-to-go function \( V_k(\cdot) \) through backward induction. When \( V_k(\cdot) \) has to be calculated numerically, the
computational load is directly related to how many $x(k)$ samples are chosen to calculate $V_k(\cdot)$. This is why DP is not suitable for high-dimensional problems as the required samples for $x(k)$ would be enormous if $x$ is a high-dimension vector.

Usually, the samples are chosen from a grid of $X_k$, which covers the feasible set of states at step $k$. For any $x(k)$ that is not on the grid, $V_k[x(k)]$ can be calculated via interpolation.

In this work, the state space is still gridded, but only a limited number of $x(k)$ samples are used to calculate $V_k(\cdot)$ at every step. A large number of the samples are intentionally put on the boundary of the feasible set to increase the accuracy near the boundaries. The rest of the samples are randomly chosen inside the feasible set.

The feasible set at each step, which is presented as the feasible points of the gridded state space, is calculated numerically backwards before implementing DP, i.e., if a valid input can be found that drives the state $x$ on the grid to a state $x^+$ which is inside the next-step feasible set, state $x$ is feasible. Calculating the feasible set before DP can ensure that the interpolation of $V_k(\cdot)$ will not be conducted at any infeasible state points. A distance constraint $d(k)\in[d_{k,\min},d_{k,\max}]$ that can be seen as a “moving window”, is also added to reduce the unnecessary computation of $V_k(\cdot)$ on practically less-interested trajectory points.


6.4.2. Parallel Computation

Parallel computation can greatly reduce the computation time for DP. For each sampling point $x(k)$ at step $k$, Equation (105) can be implemented independently. It should be noted that in our problem the calculating of $U_k[x(k)]$, or judging whether an input satisfies $u(k) \in U_k[x(k)]$ may also take significant amount of time, comparing to the implementation of finding the minimum in (105) numerically. Usually tens of thousands of sampling points are needed and each step may take several minutes totally. Therefore parallel computation is very effective for DP and the computation time is almost inverse proportional to the parallel thread number when this number is not too large.

6.4.3. Additional Boolean State to Improve Comfortability

In the problem described above, comfortability is not considered so that the resultant optimal torque operation may be unsmooth. Similar results are also reported in the hybrid electric vehicle torque-split problems in [114], where the solution is to introduce a penalty term in the cost function to penalize gear event and engine event that affect drivability and comfortability.

In this research, it is found that the torque oscillation is usually between zero and a certain value with relatively high energy-efficiency. Therefore it is not necessary to introduce the acceleration or torque as a full additional state. Instead, a Boolean state which indicates the level of acceleration or torque would be sufficient to remove the
oscillations. The extra Boolean state only doubles the computation needed for DP, which is significantly more computationally efficient than a full state.

The extra Boolean state is defined as follows,

\[
B(k) = \begin{cases} 
1, & \text{if } T_d > T_{th} \\
0, & \text{if } T_d \leq T_{th},
\end{cases}
\]

where \( T_{th} \) is a threshold.

The augmented state is defined as \( x_a = [d \ v \ B]^T \). The cost function is revised accordingly as,

\[
J_a = \sum_{k=0}^{k_f} \{g[k][x(k),u(k)] + \alpha I[B(k),x(k),u(k)]\},
\]

where \( \alpha \) is the penalty factor, and \( I \) is the indicator of torque increasing event. If the control input increases the torque from below \( T_{th} \) to above \( T_{th} \), \( I = 1 \). Otherwise, \( I = 0 \). The penalty factor is a constant no matter how long the trip is and it does not need to be tuned for different route.

6.5. Simulation Examples

In this section, the proposed method is applied to solve a route-based optimal speed control problem and simulation comparison results with nominal speed profiles are given.

6.5.1. Route and Vehicle Information

The route for this example problem is based on a real-world route in Columbus, Ohio, USA, shown in Figure 40.
Figure 40. A real-world route map for the simulation [95]

There are 9 traffic lights, 4 stop signs, and 3 turns along the route and the total length is about 8 km. The traffic light schedule cycles vary from 27 seconds to 60 seconds. It is assumed that the vehicle is running freely without leading vehicles. The road altitude and grade along the route are shown in Figure 41.

Figure 41. Altitude and road grade of the route [95]
The vehicle model is based on an electric vehicle with four in-wheel motors and no transmission. The vehicle parameters are given in Table 21 and motor efficiencies are shown in Figure 42 and Figure 43. The battery SOC change for this 8-km trip is less than 18% therefore the impact of battery SOC on powertrain efficiency can be neglected.

Table 21. Simulation vehicle parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle mass</td>
<td>2400 kg</td>
</tr>
<tr>
<td>Battery capacity</td>
<td>21.6 kWh</td>
</tr>
<tr>
<td>Tire effective radius</td>
<td>0.3 m</td>
</tr>
<tr>
<td>Max power of an in-wheel motor</td>
<td>60 kW</td>
</tr>
</tbody>
</table>

Figure 42. Driving efficiency map of one in-wheel motor [95]
Figure 43. Regenerative braking efficiency map of one in-wheel motor [95]

6.5.2. DP Parameters

The sampling period of the discrete model is 2 seconds. The grid spacing for route discretization is 1 meter, and the spacing for speed discretization varies from 0.19 m/s to 0.66 m/s, where it is denser in low speed, and sparser in high speed. At each step, no more than 61 valid inputs will be searched to determine the optimal input. The feasible state set is calculated numerically based on this grid from the final step to the first step, while \( d(k) \in [d_{k,\text{min}}, d_{k,\text{max}}] \) is limited to ±1 km of a nominal speed trajectory at step \( k \).

As stated in Section 6.4.1, not every point on the state grid is used to calculate \( V_k(\cdot) \). Two sample sets are chosen. The first sample set includes some the feasible points near the boundary of the feasible set. The maximum sample capacity is limited to 15000. The other sample set contains points inside the feasible set and the maximum sample capacity is also limited to 15000. Therefore no more than 30000 \( x(k) \) sample points are used to calculate \( V_k(\cdot) \) at each step. As there is an additional Boolean state, the computation required is doubled, i.e., \( V_k[x_a(k)] \) will be calculated for at the most 60000 \( x_a(k) \).
DP is implemented in Matlab on a desktop computer with i7-4790 CPU. Six threads are used in the parallel computing. The Matlab function scatteredInterpolant is used to conduct the interpolations. For this problem, computation time for feasible set calculation and DP implementation combined is approximately 10 hours.

6.5.3. Simulation Results

A fast nominal speed profile and a slow nominal speed profile are designed as the comparative trajectories. The vehicle is assumed to accelerate and decelerate at constant accelerations, and cruise at constant speeds. The fast profile has larger accelerations, harder brakes and cruises at the speed limits, while the slow profile has smaller accelerations, more gentle brakes, and cruises below the speed limits. In the nominal speed profiles, the vehicle does not utilize the predicted traffic light schedule information and stops when the light is red (it starts to brake if the light is yellow and the speed is not high enough to pass before it turns red, just like a human driver). The fast trip takes 680 seconds and the slow trip takes 838 seconds to finish the same route.

For both nominal speed trajectories, the corresponding optimal solution with the same trip time is obtained via DP. The time-domain trajectory of two nominal speed profiles and DP optimal speed trajectories are shown in Figure 44, Figure 45 and Figure 46. In Figure 46, Both the result of fast trips and slow trips are shown in this figure. The traffic light position and schedule is illustrated by the dash yellow lines. If the line is solid, it means that it is red light at that moment and the distance trajectory cannot go
through. The positions of stop signs are also marked in this figure by thin purple lines. The optimal speed trajectories avoid almost all red traffic lights.

Figure 44. The optimal and nominal speed planning results of the fast trip [95]

Figure 45. The optimal and nominal speed planning results of the slow trip [95]
Distance-domain speed results are shown in Figure 47. The fast trip results are shown in the upper figure, and the slow trip results are shown in the lower figure. The traffic light and stop sign symbols mark the position of the lights and signs, while the turn symbol marks the position and maximum allowed speed for the turns. All trajectories satisfy all the constraints. Unsurprisingly, the optimal speed trajectories avoid almost all red traffic lights and can satisfy all the constraints, including stops, speed limits, and lower speed boundaries.
In the slow optimal trip, the vehicle stops at the first stop sign for 30 seconds. This is not a normal operation for vehicles, but it is indeed the optimal solution (or at least one of the optimal solutions) when the trip duration is relatively long so the speed planning has a lot of flexibility. The traffic light schedules ahead are complicated and the waiting can provide a better timing for the vehicle. This operation can be avoided by shortening the DP trip time. In the fast trip, there are not such problems as the trip duration is short.

Table 22 shows the comparison of cost function (which is the sum of energy consumption and penalty terms) and energy consumption (which is shown as the battery SOC change). In the fast trip where there is less flexibility to adjust the speed, the optimal result shows about 14% advantage in energy saving. In the slow trip, the advantage is more than 19%.
Table 22. Cost and energy consumption results

<table>
<thead>
<tr>
<th></th>
<th>Cost</th>
<th>Improvement</th>
<th>Energy(kWh)</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast trips</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nominal</td>
<td>1.358×10^7</td>
<td>--</td>
<td>3.64</td>
<td>--</td>
</tr>
<tr>
<td>Optimal</td>
<td>1.155×10^7</td>
<td>15.0%</td>
<td>3.13</td>
<td>13.9%</td>
</tr>
<tr>
<td>Slow trips</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nominal</td>
<td>1.420×10^7</td>
<td>--</td>
<td>3.79</td>
<td>--</td>
</tr>
<tr>
<td>Optimal</td>
<td>1.130×10^7</td>
<td>20.4%</td>
<td>3.06</td>
<td>19.3%</td>
</tr>
</tbody>
</table>

The slow optimal trip uses less energy than the fast optimal trip, because as discussed in Chapter 6.2.2, the solution is always the most energy-efficient for any terminal step \( k \leq k_f \). Interestingly, the energy consumption of the slow nominal trip is more than the fast trip, which is mainly because there are more stop-restarts due to the red lights in the slow nominal trip.

If the interpolation of the cost-to-go function is accurate, the interpolated optimal cost-to-go value calculated from DP sample points should be close to the actual cost-to-go calculated from the resultant optimal trajectory. Figure 48 compares the two cost-to-go along the trip. For each trip, the two curves are very close and they can hardly be differentiated from this figure. This suggests that the interpolation can give a relatively accurate result and our solution after the interpolation approximation is close to the true optimal solution.
6.6. Summary

In this chapter, the optimal speed planning solution for a vehicle running on a given route with multiple stop signs, traffic lights, turns and curved segments, roads of different grades and speed limits is proposed. The distance-domain non-standard-form constraints are converted to time-domain standard constraints for appropriate handling in the optimization framework. The problem is solved via dynamic programming while multiple techniques are used to accelerate the computation. Simulation examples show that the optimal speed trajectory calculated using the proposed method may save up to 19% energy comparing to a nominal speed profile. This work may provide an evaluation tool for developing intelligent, energy-efficient vehicle speed control in real-world driving.
Chapter 7: Conclusions, Contributions, and Future work

7.1. Conclusions

This dissertation focuses on improving the energy efficiency of plug-in HEVs by using optimally-personalized powertrain control. The target is to achieve high energy-efficiency in real-world driving scenarios by integrating historical data of the individual drivers in the controller design. Human driver data are collected on a driving simulator and simulation studies are used to validate the proposed methods.

The driver’s pedal behavior is investigated and modeled in Chapter 2. Input-output hidden Markov model framework is used to describe the stochastic driving behaviors. The hidden state can represent the hidden driver’s mode, e.g., acceleration, cruise, etc. The output is the probability distribution of the driver’s pedal action, which is defined as the pedal position adjustment. The IOHMM-based model can be either directly used to predict the probability of driver action at the next step, or integrated with a random number generator and vehicle-road model to make multiple-step pedal position predictions. The parameter of the driver models can be identified using individual driver’s data off-line. Simulation results shows that the proposed model can capture each driver’s individual driving style and make better predictions in horizons up to 60 second than other models.
A worst-case relative cost optimal control method is proposed in Chapter 3. This is an investigation to a theoretic problem that is closely related to vehicle energy management optimization under different driving scenarios. WCRC optimal control focuses on constructing a control policy that can result in an actual cost close to the posteriori optimal cost by minimizing the relative cost, which is defined as the ratio between the actual cost and the optimal cost. In the problem described in this chapter, the disturbance sequence is assumed to belong to a known finite set. It is proved that backward induction can be used to build a WCRC optimal control policy. The stochastic case where the probability range of each disturbance sequence is also researched, and the solving procedure is provided. Two numerical examples are given to show the effectiveness of the methods.

Chapter 4 presents an optimized two-level control structure for HEVs running on fixed-routes. Historical data of the vehicle on this route are used to optimize the higher-level control strategy through a road-segment-based model. The battery SOC change and fuel consumption over each road segment are considered as random variables whose probability distributions can be affected by the lower-level controller parameters. Off-line stochastic dynamic programming is used to solve the problem and the result is a 2-D lookup table which will be used in real-time at the beginning of every road segment to obtain lower-level controller parameters. Simulation results show that the HEV energy consumption under the proposed method is only 1.8% more than the posteriori optimal result.
Chapter 5 extends the method of Chapter 4 to multi-route situations using the concept of finite disturbance sequence set similar to the one in Chapter 3. Based on a same road-segment-based HEV energy consumption model, the possible future route information is updated at the beginning of every road segment and the expectation of energy consumption is minimized. This method can also be real-time implemented after obtaining the lower-level controller parameter tables. Simulation studies are conducted using the driver model proposed in Chapter 2. Results show that the HEV energy consumption under the proposed method is better than any constant-factor ECMS and is only 2.5% more than the posteriori optimal cost.

Chapter 6 provides an off-line tool to compute the optimal speed given a route with stop, traffic light, speed limit, and road grade information. Distance-domain constraints are converted to time-domain constraints and the problem is solved via DP. Simulation examples show that the optimal speed trajectory calculated using the proposed method may save up to 19% energy comparing to a nominal speed profile.

7.2. Contributions

The contributions of this dissertation are explained as follows.

1) A novel stochastic driver pedal behavior model which can learn independent drivers’ driving styles is created.

2) A new worst-case relative cost optimal control method is proposed. This method can construct a feedback control policy that minimizes the relative cost for a class of discrete-time problem.
3) A real-time implementable stochastic optimal energy management strategy for hybrid electric vehicles running on fixed routes is designed using the statistics of history driving data.

4) The fix-route strategy is extended to the multi-route situation, where the location and traveling direction are also used to identify route information.

5) A global energy-optimal speed planning solution for road vehicles on a given route is presented. Along the route, there may be multiple stop signs, traffic lights, turns and curved segments, roads with different grades and speed limits, and even leading vehicles with pre-known speed profiles. The optimal speed trajectory can be calculated off-line using this method and be used to evaluate other speed planning strategies.

7.3. Future Work

Some potential topics of future work of this study are summarized as follows.

1) Exploring the potential of utilizing the driver model proposed in Chapter 2 for powertrain predictive control. Combined with a random number generator, the proposed model can generate deterministic future pedal positions and be integrated in many existing powertrain predictive control methods. However, the capability of making prediction on the action probability is not fully utilized in this way. New methods which can stochastically use the driver behavior prediction will need to be developed.

2) Extending the WCRC optimal control in Chapter 3 to a more general case. In this dissertation, the disturbance sequence of the WCRC optimal control problem is assumed
to belong to a finite set. If this requirement can be relaxed, the method will be much more practically useful.

3) Designing the complete automatic workflow to implement the multi-route optimally-personalized energy management strategy in Chapter 5. In this dissertation, the road segmentation and frequent route recognition algorithms are assumed to be completed either automatically or manually before applying the proposed method. In reality, they need to be integrated in the vehicle on-board controller or cloud server and implemented automatically. Machine learning approaches may be helpful for designing algorithms in these applications.
References


160


