

# **PREDICTIVE CONTROL OF A HYBRID POWERTRAIN**

By

Jie Yang

A DISSERTATION

Submitted to  
Michigan State University  
in partial fulfillment of the requirements  
for the degree of

Mechanical Engineering - Doctor of Philosophy

2015

# **ABSTRACT**

## **PREDICTIVE CONTROL OF A HYBRID POWERTRAIN**

By

Jie Yang

Powertrain supervisory control strategy plays an important role in the overall performance of hybrid electric vehicles (HEVs), especially for fuel economy improvement. The supervisory control includes power distribution, driver demand fulfillment, battery boundary management, fuel economy optimization, meeting emissions, etc. Developing an optimal control strategy is quite a challenge due to the high degrees of freedom introduced by multiple power sources in the hybrid powertrain. This dissertation focuses on driving torque prediction, battery boundary management, and fuel economy optimization.

For a hybrid powertrain, when the desired torque (driver torque demand) is over battery operational limits, the internal combustion (IC) engine needs to be turned on to deliver additional power (torque) to the powertrain. But the slow response of the IC engine, compared with electric motors (EMs), prevents it from providing power (torque) immediately. As a result, before the engine power is ready, the battery has to be over-discharged to provide the desired powertrain power (torque). This dissertation presents an adaptive recursive prediction algorithm to predict the future desired torque based on past and current vehicle pedal positions. The recursive nature of the prediction algorithm reduces the computational load significantly and makes it feasible for real-time implementation. Two weighting coefficients are introduced to make it possible to rely more on the data newly sampled and avoid numerical singularity calculations. This improves the prediction accuracy greatly, and also the prediction algorithm is able to adapt to different driver behaviors and driving conditions.

Based on the online-predicted desired torque and its error variance, a stochastic predictive boundary management strategy is proposed in this dissertation. The smallest upper bound of future desired torque for a given confidence level is obtained based on the predicted desired torque and prediction error variance and it is used to determine if the engine needs to be proactively turned on. That is, the engine can be ready to provide power for the “future” when the actual power (torque) demand exceeds the battery output limits. Correspondingly, the battery over-discharging duration can be reduced greatly, leading to extended battery life and improved HEV performance.

To optimize powertrain fuel economy, a model predictive control (MPC) strategy is developed based on the linear quadratic tracking (LQT) approach. The finite horizon LQT control is based on the discrete-time system model obtained by linearizing the nonlinear HEV and only the first step of the solution is applied for current control. This process is repeated for each control step. The effectiveness of the supervisory control strategy is studied and validated in simulations under typical driving cycles based on a forward power split HEV model. The developed MPC-LQT control scheme tracks the predicted desired torque trajectory over the prediction horizon, minimizes the powertrain fuel consumption, maintains the battery state of charge at the desired level, and operates the battery within its designed boundary.

Copyright by  
JIE YANG  
2015

## ACKNOWLEDGMENTS

Though this dissertation is an individual work, it would not be possible to reach the heights and explore the depths without the help, support, guidance, and encouragement of so many people.

Foremost, I would like to express the deepest appreciation to my advisor, Prof. Guoming (George) Zhu for his persistent support and advising for my study and research. His comprehensive control knowledge and extensive application experience continually and convincingly conveyed a spirit of exploration in regard to my research and scholarship, which inspired me all the time in my PhD program. His systematic guidance and enthusiastic encouragement not only helped me to develop my personal skills but also improved my way of thinking, which will benefit the rest of my career.

My sincere thanks also goes to my committee members, Dr. Harold J. Schock, Dr. Elias G. Strangas, and Dr. Jongeun Choi. Their insight discussions and constructive comments over the years were of great help in providing valuable feedback for my research.

I would like also to extend my thanks to the graduate students, research assistants, and engineers in our lab: Xuefei Chen, Ping Mi, Taotao Jin, Shupeng Zhang, Tao Zeng, Yingxu Wang, Ruitao Song, Yifan Men, Chao Cheng, Ravi Vedula, Kevin Moran and Tom Stuecken. They were always willing to help and provide their best suggestions. Without them, it would be a lonely lab.

Last but not the least, I would especially like to thank my parents, parents-in-law, sisters, my wife Ni Luo, and my daughter Irene Luo Yang who have been a great source of encouragement with their best wishes throughout my life.

# TABLE OF CONTENTS

LIST OF TABLES .....	viii
LIST OF FIGURES .....	ix
KEY TO ABBREVIATIONS.....	xii
<b>CHAPTER 1: INTRODUCTION .....</b>	<b>1</b>
1.1 Motivation .....	1
1.2 Research overview.....	2
1.2.1 Desired torque prediction.....	3
1.2.2 Battery boundary management .....	4
1.2.3 Model predictive control .....	5
1.3 Dissertation contributions.....	6
1.4 Dissertation outline.....	7
<b>CHAPTER 2: ADAPTIVE RECURSIVE PREDICTION OF THE DESIRED TORQUE .....</b>	<b>8</b>
2.1 Introduction .....	8
2.2 Desired torque prediction algorithms .....	12
2.2.1 Step-by-step prediction algorithm.....	12
2.2.2 Fixed-gain prediction algorithm.....	15
2.2.3 Adaptive recursive prediction algorithm.....	17
2.3 Hybrid electric vehicle model.....	24
2.3.1 Driver model .....	25
2.3.2 Hybrid powertrain model .....	25
2.3.2.1 IC Engine model .....	26
2.3.2.2 Electric motor/generator model .....	26
2.3.2.3 Planetary gear model.....	26
2.3.2.4 Battery model.....	27
2.3.3 Supervisory controller.....	28
2.4 Simulation study and validation .....	29
2.4.1 Weighting factor effect .....	29
2.4.2 Prediction algorithm validation and comparison .....	32
2.4.3 Influence of regression order .....	35
2.4.4 Prediction robustness .....	36
2.4.5 Prediction error and computational load.....	37
2.5 Conclusions .....	38
<b>CHAPTER 3: STOCHASTIC PREDICTIVE BATTERY BOUNDARY MANAGEMENT ....</b>	<b>40</b>
3.1 Introduction .....	40
3.2 Hybrid electric vehicle subsystem models and its baseline control strategy.....	44
3.2.1 Hybrid powertrain model .....	44
3.2.1.1 IC engine model.....	44
3.2.1.2 Battery model.....	44

3.2.2	Baseline power-follower control strategy .....	47
3.2.2.1	IC engine control logic.....	47
3.2.2.2	Power distribution.....	48
3.3	Prediction error stochastic analysis under different driving cycles.....	49
3.4	Battery boundary management strategy .....	55
3.4.1	Engine control logic.....	57
3.5	Simulation study and validation .....	59
3.5.1	Battery performance at low temperatures .....	60
3.5.2	PFCS simulation.....	61
3.5.3	PBM simulation .....	63
3.5.4	SPBM simulation .....	66
3.6	Conclusions .....	69
CHAPTER 4:	MODEL PREDICTIVE CONTROL BASED ON LINEAR QUADRATIC TRACKING.....	71
4.1	Introduction .....	71
4.2	Hybrid electric vehicle system model .....	74
4.2.1	Power transmission and vehicle dynamics.....	75
4.2.2	Hybrid battery dynamics .....	77
4.3	Model predictive control .....	77
4.3.1	Control-oriented system model .....	78
4.3.2	Linear quadratic tracking .....	82
4.4	Implementation feasibility study .....	85
4.4.1	Steady-state discrete-time Riccati solution .....	86
4.4.2	Iterative ARE solution.....	87
4.4.3	Computational load .....	87
4.5	Simulation study and validation .....	88
4.5.1	Power-follower and MPC-LQT control strategies comparison .....	89
4.5.2	Cost function weight selection .....	91
4.5.3	Fuel economy improvement study .....	93
4.6	Conclusions .....	94
CHAPTER 5:	CONCLUSIONS .....	96
5.1	Conclusions .....	96
5.2	Recommendation for future works .....	98
APPENDIX.....		100
APPENDIX A:	Key parameters of the HEV.....	100
APPENDIX B:	HEV MATLAB/Simulink models .....	101
APPENDIX C:	System functions.....	110
BIBLIOGRAPHY.....		119

## LIST OF TABLES

Table 2. 1 Driver model control parameters .....	25
Table 2. 2 Technical and simulation parameters of the HEV .....	28
Table 2. 3 Influence of prediction steps (FTP cycle).....	30
Table 2. 4 Prediction accuracy ( $n_1 = 8, n_2 = 6$ ) .....	38
Table 3. 1 Technical parameters of the HEV .....	46
Table 3. 2 Power distribution.....	49
Table 3. 3 Prediction error distribution in typical driving cycles .....	52
Table 3. 4 Relationship between confidence level and $\gamma$ .....	54
Table 3. 5 Key simulation parameters.....	59
Table 3. 6 Over-discharged energy reducing with PBM .....	65
Table 3. 7 Over-discharged energy reducing with SPBM .....	69
Table 4. 1 2-norm error and Time complexity ( $n=6, N=50$ ).....	88
Table 4. 2 Key simulation parameters .....	88
Table 4. 3 Optimal weightings line searching ( $Q=1$ ).....	91
Table 4. 4 Optimal weightings area seeking for best fuel consumption ( $Q=1$ ) .....	92
Table 4. 5 Total equivalent fuel consumptions .....	93
Table A. 1. Key parameters of the HEV .....	100

## LIST OF FIGURES

Figure 1. 1 Supervisory control scheme .....	2
Figure 2. 1 Step-by-step prediction diagram.....	12
Figure 2. 2 Flow chart of step-by-step prediction algorithm .....	15
Figure 2. 3 Multi-step prediction diagram .....	16
Figure 2. 4 Data diagram .....	16
Figure 2. 5 Flow chart of adaptive recursive prediction algorithm.....	23
Figure 2. 6 Architecture of the HEV model.....	24
Figure 2. 7 Driver model.....	25
Figure 2. 8 $\ P(k)\ _F$ trace with $\alpha = \beta = 1$ under FTP driving cycle .....	29
Figure 2. 9 $\ P(k)\ _F$ and adaptive $\alpha$ changing with $\beta = 1$ .....	31
Figure 2. 10 Influence of $\beta$ .....	32
Figure 2. 11 50-step step-by-step prediction.....	32
Figure 2. 12 50-step fixed-gain prediction.....	33
Figure 2. 13 Adaptive recursive prediction.....	34
Figure 2. 14 Driving cycle validation .....	35
Figure 2. 15 Prediction errors with different vector orders .....	36
Figure 2. 16 Comparison of accumulated torque error .....	37
Figure 3. 1 PFCS implementation.....	47
Figure 3. 2 Engine control diagram of PFCS.....	48
Figure 3. 3 Prediction error distribution .....	50
Figure 3. 4 Prediction error distribution under other four typical driving cycles .....	51

Figure 3. 5 Probability density function .....	53
Figure 3. 6 Confidence level profile .....	54
Figure 3. 7 SPBM implementation .....	55
Figure 3. 8 SPBM engine control logic diagram .....	58
Figure 3. 9 Battery temperature, maximum power and internal resistance trace .....	60
Figure 3. 10 Battery performance at different temperatures .....	61
Figure 3. 11 Battery power output under US06 cycle at 0°C .....	62
Figure 3. 12 Engine and EMB torque outputs .....	63
Figure 3. 13 Reduced battery over-discharging duration.....	64
Figure 3. 14 Over-discharged power reducing .....	65
Figure 3. 15 Over-discharging duration with confidence level .....	66
Figure 3. 16 Reduced over-discharged energy with number of engine-start.....	67
Figure 3. 17 Fuel consumption under different confidence levels.....	68
Figure 4. 1 Free-body diagram of power transmission and vehicle dynamics .....	75
Figure 4. 2 Prediction buffer .....	83
Figure 4. 3 Flow chart of MPC-LQT algorithm .....	85
Figure 4. 4 $\ P_{j,0} - \bar{P}_j\ _2$ and $\ P_{j,0} - \hat{P}_j\ _2$ traces under US06 cycle .....	86
Figure 4. 5 Power-follower and MPC-LQT performance .....	89
Figure 4. 6 Engine and motor operation maps under US06 driving cycle.....	90
Figure 4. 7 Optimal weightings seeking .....	92
Figure 4. 8 Total fuel consumption improvement comparison.....	94
Figure B.1 HEV model .....	101
Figure B.2 HEV driver model.....	101

Figure B.3 HEV supervisory controller model .....	102
Figure B.4 HEV engine on/off control model .....	102
Figure B.5 HEV supervisory power distribution model .....	103
Figure B.6 HEV engine model.....	103
Figure B. 7 HEV EMA model .....	104
Figure B. 8 HEV EMB model.....	104
Figure B. 9 HEV battery model .....	105
Figure B. 10 HEV battery VOC and Rint model .....	105
Figure B. 11 HEV battery limited power model.....	106
Figure B. 12 HEV battery output current model.....	107
Figure B. 13 HEV battery SOC model .....	107
Figure B. 14 HEV battery air flow/temperature model .....	108
Figure B. 15 HEV vehicle dynamics model .....	108
Figure B. 16 HEV braking regeneration model .....	109
Figure B. 17 HEV planetary gear model .....	109

## KEY TO ABBREVIATIONS

AR	Auto regressive
ARB02	The Californian Air Resources Board Number 2
ARE	Algebraic Riccati equation
ARMA	Auto regressive moving average
ARX	Auto regressive with extra input
ATE	Accumulated torque errors
CL	Confidence level
DOD	Depth of discharge
DP	Dynamic programming
DRE	Difference Riccati equation
EMA	Electric motor A
EMB	Electric motor B
EMs	Electric motors
FTP	Federal Test Procedure
HEVs	Hybrid electric vehicles
ICE	Internal combustion engine
IM240	Inspection and Maintenance 240s cycle
LQT	Linear quadratic tracking
LS	Least Squares
MPC	Model predictive control
NYCC	New York City Cycle

PBM	Predictive boundary management
PDF	Probability density function
PFCS	Power-follower control strategy
PID	Proportional integral derivative
RMS	Root mean square
SOC	State of charge
SPBM	Stochastic predictive boundary management
US06	Supplemental Federal Test Procedure (SFTP)

# CHAPTER 1: INTRODUCTION

## 1.1 Motivation

Compared to conventional vehicles powered by the internal combustion (IC) engine, hybrid electric vehicles (HEVs) are able to improve fuel economy with reduced emissions due to the following reasons. Firstly, part of the vehicle kinetic energy can be recovered through braking-regeneration; secondly, the IC engine can be turned off under stop or low-load conditions; and last but not the least, the HEV is able to operate the IC engine at its most efficient conditions with the help of electric motor and battery [1]. Two of the main challenges for HEVs are the limited useful life of the battery and the development of a real-time optimal control strategy used to improve the fuel economy without complete knowledge of future driving conditions.

At low temperatures, the HEV battery is more likely to be over-discharged than at normal temperatures due to the limited battery capacity, especially under transient operational conditions. However, frequent or persistent over-discharging could lead to permanent damage and reduced life of the battery, and hence affect the overall performance of the HEV. As part of the hybrid powertrain, the IC engine can be used to deliver additional power (torque) to the drivetrain when the battery capacity is limited and the battery over-discharging would occur. However, the slow response of the engine, compared with electric motor, prevents it from delivering torque immediately. Note that it could take half to one second to start the engine and make its power available. Before the engine power is available, the battery has to be over-discharged to maintain the powertrain power (torque) output, which could lead to reduced battery life and degraded HEV

performance. This often happens at extremely low temperatures. However, if the desired powertrain power (torque) can be predicted, the IC engine can get ready before the actual power exceeds the battery capability. In this case, the battery can be operated within its designed boundary or with reduced over-discharging duration leading to extended battery life, which is called battery boundary management in this dissertation.

The HEV supervisory control strategy is aimed to improve vehicle fuel economy, and it is a challenging task to optimize the fuel consumption in real-time without the complete mission knowledge. In the case that the complete knowledge is available, a global optimization approach, such as dynamic programming (DP), can be deployed. But this is not the case for practical applications since the mission information is generally not available. However, when the desired torque command can be predicted in real-time, this information can be used for optimal equivalent fuel consumption control over the prediction horizon, where model predictive control (MPC) and linear quadratic tracking (LQT) are used.

**1.2 Research overview**

Supervisory control strategy development has many challenges. This dissertation focuses on battery boundary management and powertrain fuel economy optimization. Figure 1. 1 shows the supervisory control scheme based upon the predicted desired power (torque) command.

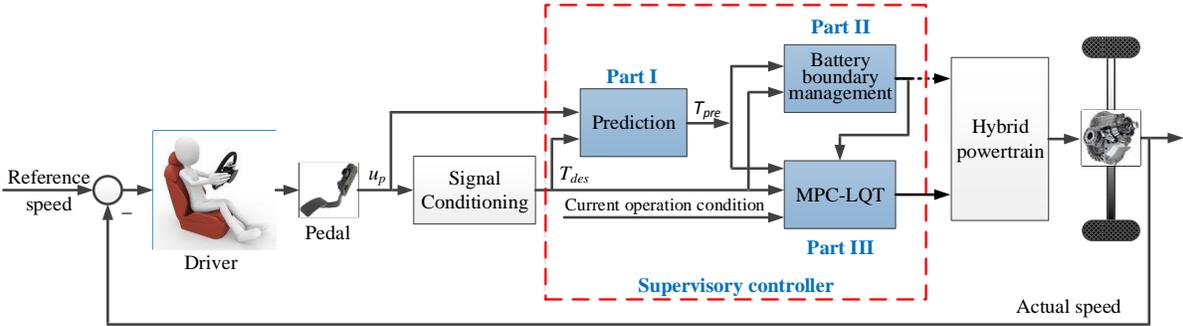


Figure 1. 1 Supervisory control scheme

As shown in Figure 1. 1 ( $u_p$ —pedal position;  $T_{des}$ —desired torque; and  $T_{pre}$ —predicted desired torque), there are three main topics in this dissertation. Part I studies the desired torque prediction. Due to real-time application, the prediction algorithm shall have low computational load with proper prediction accuracy and robustness to different driving behaviors and patterns. Part II describes the battery boundary management. Based on the desired torque prediction and its error variance, the control strategy manages the on-off operation of the IC engine to make engine power ready before the actual driving power demand exceeds the battery designed operational limits. Part III presents developed supervisory control scheme based on MPC and LQT. With prediction information available online, the MPC supervisory controller follows the predicted desired torque trajectory, minimizes the total equivalent fuel consumption over the prediction horizon, and at the same time, maintains the battery state of charge (SOC) at the desired level and keeps the battery operated within its operational boundary.

### **1.2.1 Desired torque prediction**

Prediction is useful to provide additional information for decision making and is widely used in many fields, such as statistics, science, engineering, and etc. Motivated by the potential application to the desired torque prediction in the area of the HEV supervisory control. This dissertation proposes an adaptive recursive desired torque prediction algorithm and investigates its feasibility based on past and current desired torque and pedal position information using autoregressive with extra input (ARX) model, where both pedal position  $u_p$  and conditioned pedal position  $T_{des}$  (desired torque) are inputs; see Figure 1. 1. In order to obtain the regression gains, the Least-Squares approach is utilized to minimize the error between actual system outputs and predictions. For the purpose of reducing the influence of old data and depending more on the current one, two weighting coefficients are introduced. This makes it possible for the prediction

algorithm to forget old driving patterns and adapt to the new ones. Since the regression gains are updated online at each step, the prediction algorithm is able to adapt to different driver behaviors with low computational load for real-time implementation. The effectiveness of the proposed prediction algorithm is studied and validated in simulations under typical driving cycles and compared with two existing prediction algorithms: step-by-step and fixed gain prediction. Among the three algorithms, the proposed one is the best in terms of the guaranteed prediction accuracy, low computational load, and robustness to different driver behavior and driving patterns.

### **1.2.2 Battery boundary management**

Keeping the hybrid battery operated within its designed boundary is a challenging task especially at low battery temperatures, and most existing work either limits battery power output or uses external device to increase the battery temperature so as to expand the battery capacity. Inspired by the multiple power sources available in an HEV, this research proposes a stochastic predictive boundary management (SPBM) based on the predicted desired torque and its error variance. In the hybrid powertrain, the IC engine is modeled as a first order dynamics system with start delay; the electric motor (or generator) is modeled without dynamics; and the battery capacity, internal resistant, discharging/charging current are modeled as functions of temperature, and assume they will not change over the prediction horizon . The proposed SPBM is capable of proactively turning on the IC engine when the predicted desired power (torque) is greater than the battery capacity to reduce the battery over-discharging duration. With the predicted desired torque and its error variance calculated in real-time, the SPBM can be implemented for practical applications due to its low computation load. The effectiveness of the SPBM is studied and validated in simulations under typical driving cycles and compared to the PBM and baseline power-follower strategy (without predictive boundary management), the SPBM reduces the

battery over-discharging duration greatly among the three especially under aggressive driving cycles leading to improved battery life and HEV performance.

### **1.2.3 Model predictive control**

Another challenge for the HEV supervisory control strategy development is how to optimally distribute power to the multiple power sources. Based on the desired torque prediction, this research proposes a model predictive control strategy using linear quadratic tracking (LQT) to obtain the close-loop control law for optimal power distribution, where the LQT controller is designed to track the predicted desired torque trajectory and minimize the total equivalent fuel consumption over the prediction horizon. The total equivalent fuel consumption is defined as the sum of engine fuel and battery equivalent fuel consumed. The control-oriented nonlinear powertrain model is linearized and discretized under current operational condition for the control law calculation. Once the control law is obtained, only the first step of the horizon is used for current control. Since the performance index is defined as the combination of tracking error and total equivalent fuel consumption, by carefully selecting the weights, the LQT is able to track the predicted desired torque, minimize the engine fuel consumption, and keep the battery SOC at the desired level. In addition, due to the inheritance of engine-on control logic from the battery boundary management strategy, the LQT is also capable of keeping the battery operated within its operational boundary.

The effectiveness of the proposed MPC supervisory control strategy is studied and validated in simulations under typical driving cycles, and also compared with baseline power-follower control strategy. The MPC-LQT is able to operate the engine at high-efficiency operational region, leading to significantly improved fuel economy especially under highway or mixed city and highway driving cycles while maintaining the battery SOC at the target level and keeping the

battery operated within the boundary. The reduced computational load using an approximated iterative Riccati solution for LQT makes it feasible for real-time implementation with very small fuel economy penalty.

### **1.3 Dissertation contributions**

The following is a list of major contributions:

- A multi-step desired torque prediction algorithm, based on pedal position signal, is proposed to predict the future desired torque over a given time horizon. In the algorithm, two weighting coefficients are utilized to rely less on the past data and more on the current one. This makes it possible for the algorithm to adapt to current driving patterns. Also, since regression gains are updated online, the proposed algorithm is able to adapt to different driver behaviors and the low computational load makes it feasible for real-time applications.
- A stochastic predictive boundary management strategy is proposed to reduce the battery over-discharging duration to extend battery life. The proposed strategy proactively turns the engine on and makes engine power available to the powertrain based on the predicted desired torque and its prediction error variance. Most importantly, the adaptive prediction and its error variance can be calculated in real-time with very low computational load, which makes the strategy feasible for practical application.
- A model predictive supervisory control strategy is developed based on linear quadratic tracking. The controller tracks the predicted driver torque demand with optimized total equivalent fuel consumption over a given horizon. At the same time the battery is operated within its operational boundary and its state of charge is maintained around the desired level.

## **1.4 Dissertation outline**

The material presented in this dissertation is organized as follows. In chapter 2, an adaptive recursive prediction algorithm of the desired torque is proposed and compared with two existing prediction algorithms, step-by-step and fixed-gain prediction ones. The proposed algorithm is validated through simulations. In chapter 3, a stochastic predictive boundary management strategy is developed to operate the battery within its operational boundary and reduce over-discharging duration. The proposed strategy is compared with predictive boundary management and baseline power-follower scheme and shows significant reduction of battery over-discharging duration. In chapter 4, a model predictive supervisory control strategy is proposed based on the linear quadratic tracking that follows the predicted driver torque demand, minimizes the equivalent fuel consumption over a given horizon, and at the same time maintains the battery within its operational boundary and battery state of charge close to the target level. The effectiveness of the proposed strategy is validated under typical driving cycles and compared with power-follower one. Chapter 5 adds conclusions and future work.

# CHAPTER 2: ADAPTIVE RECURSIVE PREDICTION OF THE DESIRED TORQUE

## 2.1 Introduction

Hybrid electric vehicle (HEV) has sparked the interest of researchers and car manufacturers for decades [1] due to its capability of improving fuel economy and reducing emissions. The hybrid powertrain has the freedom to use electric machine (EM) or (and) the internal combustion (IC) engine to deliver required power to the vehicle wheels [3]. Among the hybrid powertrain subsystems, battery-EM subsystem has much faster dynamics than an IC engine, and is often required to provide additional torque (or power) to the vehicle drive-train during the transient operations, especially during the low speed vehicle operations [4]. This enables the HEV to outperform conventional vehicles. However, when an HEV is under cold start operations, the available battery power is significantly reduced due to the cold temperatures [5] [6], and because of the IC engine start delay, the battery has to provide power at the level beyond its normal operation limit, which would dramatically shorten the battery life [7]. If the desired torque, therefore, can be predicted, the IC engine will be started early when necessary, which reduces the battery over-discharging duration and keeps the battery operated within the designed operational boundary, and hence to protect the battery. This makes torque demand prediction important for extending battery life.

References [1], [8]-[11] develop predictive energy management control strategies to estimate

future torque (power) demand, but the prediction were vision-based and prediction accuracy cannot be guaranteed. Reference [4] presents a methodology to predict available torque in real-time for HEVs, and [12] proposes an algorithm for predictive torque control. Although they can avoid issuing commands to exceed battery capacity, however the protection is very limited and is not suitable for extending battery life due to single step prediction. In addition, there has been some research on driver torque prediction, for example [13]-[14], however most of driver torque predictions are for driving behavior modeling such as power steering torque prediction. There also were many driver models with even more sophistic frameworks, such as auto-regressive with extra input (ARX) based models: stochastic switched ARX (SS-ARX) [15], piecewise affine ARX (PWARX) [16], probability weighted ARX (PrARX) [17][18] and switched ARX (SARX) [19] (to name a few). Whereas, their focuses are mainly on driver behavior modeling, identification, and prediction. Gaussian mixture model [20]-[23] and Markov chain model [20], [24]-[25] were also used to predict powertrain torque. The challenge is to obtain a feasible prediction model based the training data set for reduced prediction error and to select a proper model order for reduced computational load. It also worth to mention that both Gaussian matrix and Markov chain prediction models are with the fixed prediction gains.

Although there are many prediction algorithms in literature, few research was focused on multi-step prediction of future powertrain desired torque based on the past and current acceleration pedal position signal. The prediction algorithm can be divided into linear and nonlinear ones, where the linear method including auto-regressive (AR) model, auto-regressive moving average (ARMA) model, aforementioned ARX model, and so on; the typical nonlinear prediction adopts artificial neural networks. Whereas the main challenge to develop a good multi-step prediction model based upon the past data is to provide accurate prediction using minimal available

information for reduced computational load. References [26], [27], [28], and [31] present a typical multi-step prediction method, the recursive-one-step (or step-by-step) prediction, which uses the same prediction model iteratively, and the previous predictions are used along with the sampled data as inputs to obtain the next prediction. By applying the one-step prediction iteratively, multi-step prediction can be achieved. For one-step prediction, this technique is comparatively feasible. However, for multi-step prediction, due to the fact that the predicted data is used for predicting the next data point, the prediction error could be accumulated and the error increases significantly as the number of prediction step goes up.

Also, references [26] to [31] provide a method for direct-multi-step prediction. Compared with the recursive-one-step method, all the input is the measured data, i.e., the predicted data is not used iteratively for the multi-step prediction, and future signal is predicted directly in one prediction. Some improvements are achieved due to the elimination of the cumulative error, and comparatively more accurate results could be obtained [32]. However, for more than one models are required for direct-multi-step prediction, the computational load is very high, which makes real-time prediction difficult.

Reference [29] describes a DirRec strategy using the combination of recursive-one-step and direct-multi-step methods for multi-step prediction. This approach not only provides the prediction of each step but also validates the information generated from the previous prediction steps. Although the cumulative prediction error decreases over the recursive-one-step prediction for the DirRec strategy, the conditional independence of the trained models is not necessarily effective [33]. As an extension to the above prediction strategies for multiple-input and single-output system, references [31] and [34] present the case for multi-input and multi-output (MIMO) systems, where the predicted value is a vector of future values. Nevertheless the conditional independence

assumption of the DirRec algorithm is removed, the key disadvantage of these prediction strategies is the reduced prediction flexibility. In summary, the existing prediction work has three major disadvantages as summarized in [35]. Firstly, low prediction accuracy due to the utilization of the predicted data for future prediction; secondly, the high cumbersome online computational load makes it impossible for online applications because of the required matrix calculations; last, the prediction algorithm tends to converge to a fixed prediction model, which makes it difficult to adapt to different driving patterns for HEV applications.

Compared with two existing prediction algorithms, step-by-step and fixed-gain prediction ones, this research proposes an adaptive recursive multi-step prediction algorithm for estimating the hybrid powertrain torque demand in real-time based upon the ARX model. The proposed algorithm uses both conditioned and raw acceleration pedal position signals for direct multi-step prediction instead of iterative step-by-step approach, and the cumulative error is reduced dramatically. Also two weighting coefficients are introduced. One is used as the forgetting factor for the past data and it is adaptively adjusted to control the over and under flow problems to improve prediction accuracy; and the other is the weighting factor for current data used to make regression model adapt to new driving patterns quickly. Simulation study was conducted to compare three prediction algorithms under FTP driving cycle, along with the other four typical driving cycles. The study is also conducted to minimize regression order with the desired accuracy to make the real-time implementation possible. At last, the robustness to different driver behaviors is investigated by altering the driver model parameters (“D” gain) in simulation study. The main contribution of this research is to develop a pedal position based multi-step desired torque prediction algorithm to predict the future desired torque. In the simulation validation, 50-step prediction was utilized. The second contribution is the utilizing of  $\alpha$  and  $\beta$  to weight the past and incoming data, which weighs

less on the past (old) data and weighs more on current data. This is very important, since driver change could totally affect the desired powertrain torque pattern. Weighting more on new data would be important to have an accurate prediction.

This chapter is organized as follows. Section 2.2 discusses two existing prediction algorithms, step-by-step and fixed-gain ones, and proposes an adaptive recursive prediction algorithm. A hybrid vehicle model is described for evaluating the proposed algorithm in section 2.3. Section 2.4 presents the simulation results with respect to the effectiveness of the weighting factors, prediction errors and computational load of the three algorithms, regression order selection, as well as the prediction algorithm robustness to different driving patterns. Section 2.5 adds the conclusions.

## 2.2 Desired torque prediction algorithms

In statistics and signal processing, the AR and ARX models are widely used to represent certain random processes. For instance, it is often used to describe certain time-varying random processes with the output variable depending linearly on its previous values and inputs. In this research, AR and ARX models are selected to predict future powertrain desired torque based upon current and past acceleration pedal position output (raw pedal position output) and desired torque demand output (conditioned pedal position output).

### 2.2.1 Step-by-step prediction algorithm

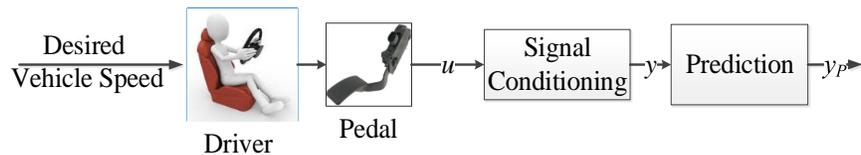


Figure 2. 1 Step-by-step prediction diagram

The step-by-step prediction algorithm makes one-step prediction based upon current and past output values. As is shown in Figure 2. 1, values of current and past desired torque demand outputs

$y$  (conditioned pedal position output) are utilized as inputs for the one-step prediction to estimate future desired torque  $y_p$ .

At time step  $k$ , AR based one-step prediction model can be expressed by

$$y_p(k+1) = \sum_{i=0}^{n-1} y(k-i)a_i(k) \quad (1.1)$$

where  $y_p(k+1)$  is the one-step ahead prediction based upon the current and past information  $y(k-i)$ ;  $n$  is the number of data points used for the one-step prediction at sample step  $k$ ;  $a_i(k)$  ( $i = 0, 1, \dots, n-1$ ) is the AR model weighting coefficients.

To establish the mapping between the predicted and the known outputs, equation (1.1) can be expressed as the following:

$$y_p(k+1) = \hat{\phi}_n(k) \cdot \hat{\Theta}_n(k) \quad (1.2)$$

where vector  $\hat{\phi}_n(k) = [y(k) \ y(k-1) \ \dots \ y(k-n+1)]$  represents the current and past data that can be obtained by sampling the conditioned pedal output  $y$ ; and  $\hat{\Theta}_n(k) = [a_0(k) \ a_1(k) \ \dots \ a_{n-1}(k)]^T$  denotes the AR model regression vector at sample step  $k$ .

Define the following vectors:

$$\hat{\Phi}_N(k) = \begin{bmatrix} \hat{\phi}_n(k-N+1) \\ \hat{\phi}_n(k-N+2) \\ \vdots \\ \hat{\phi}_n(k) \end{bmatrix} \quad (1.3)$$

$$Y_{p,N}(k) = \begin{bmatrix} y_p(k-N+1) \\ y_p(k-N+2) \\ \vdots \\ y_p(k) \end{bmatrix}, \quad Y_N(k) = \begin{bmatrix} y(k-N+1) \\ y(k-N+2) \\ \vdots \\ y(k) \end{bmatrix} \quad (1.4)$$

Equation (1.1) can be expressed as follows:

$$Y_{p,N}(k+1) = \hat{\Phi}_N(k) \hat{\Theta}_n(k) \quad (1.5)$$

To obtain regression vector  $\hat{\Theta}_n(k)$ , define the following cost function:

$$J_{one-step}(k) = \left\| Y_N(k) - \hat{\Phi}_N(k-1) \hat{\Theta}_n(k) \right\|^2, \quad N > 1 \quad (1.6)$$

The corresponding solution that minimizes the cost defined in equation (1.6) is the Least-Squares (LS) solution (see [36] and [37]) below, provided the indicated inverse exists.

$$\hat{\Theta}_n(k) = \left[ \hat{\Phi}_N^T(k-1) \hat{\Phi}_N(k-1) \right]^{-1} \hat{\Phi}_N^T(k-1) Y_N(k) \quad (1.7)$$

Note that equation (1.7) can be expressed in the following form to reduce computational load.

$$\hat{\Theta}_n(k) = \left[ \sum_{k=1}^N (\hat{\phi}_n^T(k-1) \hat{\phi}_n(k-1)) \right]^{-1} \left[ \sum_{k=1}^N \hat{\phi}_n^T(k-1) y(k) \right] \quad (1.8)$$

Since the regression vector defined in equation (1.8) can only provide one-step prediction, to have a multi-step prediction, the step-by-step prediction approach [38] is adopted, where the  $m$ -step prediction  $y_p(k+m)$  can be obtained using the following iterative formulae:

$$y_p(k+j+1) = \tilde{\Phi}_c(k, j) \tilde{\Theta}_n(k, j), \quad j=1, 2, \dots, m-1. \quad (1.9)$$

where

$$\tilde{\Phi}_c(k, j) = [y_p(k+j), \dots, y_p(k+1), y(k), \dots, y(k+j+1-n)] \quad (1.10)$$

and

$$\left\{ \begin{array}{l} \tilde{\Theta}_n(k, j+1) = \left[ I - \frac{\tilde{P}(k, j) \tilde{\Phi}_c^T(k, j) \tilde{\Phi}_c(k, j)}{\tilde{\Phi}_c(k, j) \tilde{P}(k, j) \tilde{\Phi}_c^T(k, j) + 1} \right] \left[ \tilde{\Theta}_n(k, j) + \tilde{P}(k, j) \tilde{\Phi}_c^T(k, j) y_p(k+j) \right] \\ \tilde{P}(k, j+1) = \tilde{P}(k, j) - \frac{\left[ \tilde{\Phi}_c(k, j) \tilde{P}(k, j) \right]^T \tilde{\Phi}_c(k, j) \tilde{P}(k, j)}{\tilde{\Phi}_c(k, j) \tilde{P}(k, j) \tilde{\Phi}_c^T(k, j) + 1} \\ \tilde{\Theta}_n(k, 1) = \hat{\Theta}_n(k), \quad \tilde{P}(k, 1) = \left[ \hat{\Phi}_N^T(k) \hat{\Phi}_N(k) \right]^{-1} \end{array} \right. \quad (1.11)$$

By applying the algorithm described in equations (1.9) to (1.11) repeatedly for  $j = 1, 2, \dots, m-$

1,  $m$ -step prediction can be achieved. The algorithm is also summarized in Figure 2. 2.

The disadvantages of the step-by-step prediction algorithm are two-fold: low prediction accuracy and high computational load due to the inverse calculation described in equation (1.8). The low prediction accuracy is mainly due to the fact that the predicted output was used repeatedly during the step-by-step prediction. This will be confirmed in the simulation section. High amount of computational load is due to the fact that a new regression vector needs to be calculated online for each prediction.

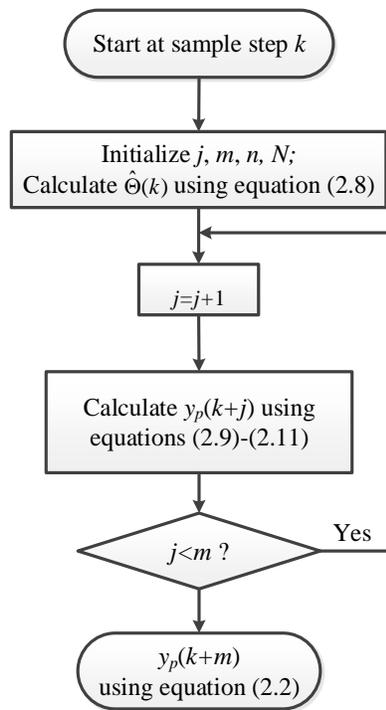


Figure 2. 2 Flow chart of step-by-step prediction algorithm

### 2.2.2 Fixed-gain prediction algorithm

For the fixed-gain prediction, the  $m$ -step prediction will be calculated directly without using the iterative process used in the step-by-step prediction. To improve the prediction accuracy, both conditioned pedal position output  $y$  and raw pedal position output  $u$  are used for prediction; see Figure 2. 3.

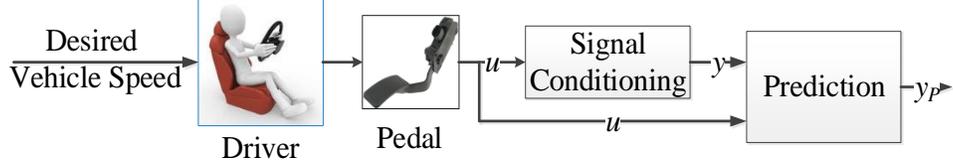


Figure 2.3 Multi-step prediction diagram

The ARX based fixed-gain multi-step prediction model can be expressed as

$$y_p(k+m) = \sum_{i=0}^{n-1} (y(k-i)a_i + u(k-i)b_i) \quad (1.12)$$

where  $y_p(k+m)$  is the multi-step prediction based upon the current and past information  $y(k-i)$  and  $u(k-i)$ ;  $a_i$  and  $b_i$  ( $i = 0, 1, \dots, n-1$ ) are the weighting coefficients of  $y$  and  $u$ , respectively.

Equation (1.12) can be expressed as follows:

$$y_p(k+m) = \varphi_{2n}(k) \cdot \bar{\Theta}_{2n} \quad (1.13)$$

where  $\bar{\Theta}_{2n} = [a_0, a_1, \dots, a_{n-1}, b_0, b_1, \dots, b_{n-1}]^T$  is the constant regression coefficient vector and the data vector  $\varphi_{2n}(k) = [y(k), \dots, y(k-n+1), u(k), \dots, u(k-n+1)]$  can be obtained by sampling the conditioned pedal position output  $y$  and raw pedal position output  $u$ , see Figure 2.4.

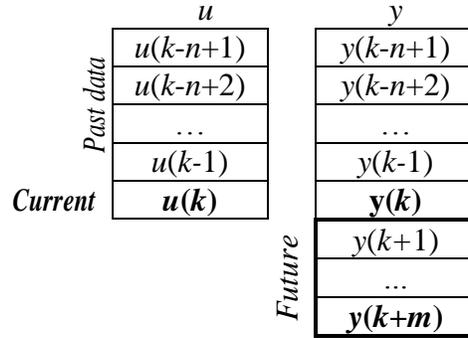


Figure 2.4 Data diagram

Note that the notations used in the fixed-gain prediction subsection are the same as those used in the step-by-step prediction subsection. Define the following vector in a similar way to the step-by-step algorithm.

$$\Phi_N(k) = \begin{bmatrix} \varphi_{2n}(k-N+1) \\ \varphi_{2n}(k-N+2) \\ \vdots \\ \varphi_{2n}(k) \end{bmatrix} \quad (1.14)$$

and we have

$$Y_{p,N}(k+m) = \Phi_N(k) \bar{\Theta}_{2n} \quad (1.15)$$

where  $Y_{p,N}(k+m)$  is defined in (1.4). Now define the following cost function:

$$J_{multi-step}(k) = \|Y_N(k) - \Phi_N(k-m) \bar{\Theta}_{2n}\|^2, \quad N > 1 \quad (1.16)$$

where  $Y_N(k)$  is defined in (1.4). The LS solution that minimizes the cost in (1.16) is provided below, assuming the indicated inverse exists.

$$\bar{\Theta}_{2n} = \left[ \Phi_N^T(k-m) \Phi_N(k-m) \right]^{-1} \Phi_N^T(k-m) Y_N(k) \quad (1.17)$$

The major difference between the step-by-step and fixed-gain multi-step prediction is that the ARX regression gain vector is updated in every step for the step-by-step algorithm, while the fixed-gain one is not. In the fixed-gain algorithm,  $\bar{\Theta}_{2n}$  is calculated based upon the pre-collected data and it will not be updated during the online prediction process. Since the driver behaviors could be quite different for different drivers under different driving cycles, the fixed-gain prediction may not be able to provide accurate prediction, which leads to the development of the adaptive recursive prediction algorithm.

### 2.2.3 Adaptive recursive prediction algorithm

As discussed in the above subsection, the fixed-gain algorithm may not provide satisfactory prediction since it cannot adapt to different driving pattern and behaviors. This subsection proposes the adaptive recursive prediction algorithm, where the ARX adaptive recursive multi-step

prediction model can be expressed as follows,

$$y_p(k+m) = \varphi_{2n}(k) \cdot \tilde{\Theta}_{2n}(k) \quad (1.18)$$

where  $y_p(k+m)$  and  $\varphi_{2n}(k)$  are defined in (1.12) and (1.13),

$\tilde{\Theta}_{2n}(k) = [a_0(k), a_1(k), \dots, a_{n-1}(k), b_0(k), b_1(k), \dots, b_{n-1}(k)]^T$  is the ARX model regression vector at sample step  $k$ , and define the following

$$Y_p(k) = \begin{bmatrix} y_p(0) \\ y_p(1) \\ \vdots \\ y_p(k) \end{bmatrix}, Y(k) = \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(k) \end{bmatrix}, \Phi(k) = \begin{bmatrix} \varphi_{2n}(0) \\ \varphi_{2n}(1) \\ \vdots \\ \varphi_{2n}(k) \end{bmatrix} \quad (1.19)$$

Define the multi-step prediction as follows

$$\begin{bmatrix} \alpha Y_p(k+m-1) \\ \beta y_p(k+m) \end{bmatrix} = \begin{bmatrix} \alpha \Phi(k-1) \\ \beta \varphi_{2n}(k) \end{bmatrix} \tilde{\Theta}_{2n}(k) \quad (1.20)$$

where  $\alpha \in (\xi, 1]$  ( $\xi > 0$ ) and  $\beta \geq 1$  are two weighting coefficients for the iterative regression prediction algorithm. Note that  $\alpha$  is a forgetting factor used for the data sampled in the past, and  $\beta$  is used to increase the weight for the current sample data.

To calculate the regression coefficient at each sample, define the following cost function at time step  $k$ .

$$J_{multi-step}(k) = \left\| \begin{bmatrix} \alpha Y(k-1) \\ \beta y(k) \end{bmatrix} - \begin{bmatrix} \alpha \Phi(k-m-1) \\ \beta \varphi_{2n}(k-m) \end{bmatrix} \tilde{\Theta}_{2n}(k) \right\|^2, N > 1 \quad (1.21)$$

The corresponding LS solution that minimizes the cost defined in (1.21) is given below, provided the indicated inverse exists.

$$\tilde{\Theta}_{2n}(k) = \left[ \alpha^2 \Phi^T(k-m-1)\Phi(k-m-1) + \beta^2 \varphi_{2n}^T(k-m)\varphi_{2n}(k-m) \right]^{-1} \cdot \left[ \alpha^2 \Phi^T(k-m-1)Y(k-1) + \beta^2 \varphi_{2n}^T(k-m)y(k) \right] \quad (1.22)$$

Applying the approach in [38] to avoid matrix inversion in (1.22),

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B[DA^{-1}B + C^{-1}]^{-1}DA^{-1} \quad (1.23)$$

Let  $A = \alpha^2 \Phi^T(k-m-1)\Phi(k-m-1)$ ,  $B = \beta^2 \varphi_{2n}^T(k-m)$ ,  $C = I$ , and  $D = \varphi_{2n}(k-m)$ , and

(1.23) becomes

$$\begin{aligned} \tilde{\Theta}_{2n}(k) = & \left( \frac{1}{\alpha^2} [\Phi^T(k-m-1)\Phi(k-m-1)]^{-1} - \frac{\beta^2}{\alpha^2} [\Phi^T(k-m-1)\Phi(k-m-1)]^{-1} \varphi_{2n}^T(k-m) \right. \\ & \cdot \left. \left[ \frac{\beta^2}{\alpha^2} \varphi_{2n}(k-m) [\Phi^T(k-m-1)\Phi(k-m-1)]^{-1} \varphi_{2n}^T(k-m) + 1 \right]^{-1} \right. \\ & \cdot \varphi_{2n}(k-m) \frac{1}{\alpha^2} [\Phi^T(k-m-1)\Phi(k-m-1)]^{-1} \left. \right) \\ & \cdot [\alpha^2 \Phi^T(k-m-1)Y(k-1) + \beta^2 \Phi^T(k-m-1)y(k)] \end{aligned} \quad (1.24)$$

Reorganize (1.24) leads to

$$\begin{aligned} \tilde{\Theta}_{2n}(k) = & [\Phi^T(k-m-1)\Phi(k-m-1)]^{-1} \Phi^T(k-m-1)Y(k-1) - \frac{\beta^2}{\alpha^2} [\Phi^T(k-m-1)\Phi(k-m-1)]^{-1} \varphi_{2n}^T(k-m) \\ & \cdot \left[ \frac{\beta^2}{\alpha^2} \varphi_{2n}(k-m) [\Phi^T(k-m-1)\Phi(k-m-1)]^{-1} \varphi_{2n}^T(k-m) + 1 \right]^{-1} \varphi_{2n}(k-m) [\Phi^T(k-m-1)\Phi(k-m-1)]^{-1} \\ & \cdot \Phi^T(k-m-1)Y(k-1) + \frac{\beta^2}{\alpha^2} [\Phi^T(k-m-1)\Phi(k-m-1)]^{-1} \varphi_{2n}^T(k-m)y(k) - \frac{\beta^2}{\alpha^2} \\ & \cdot [\Phi^T(k-m-1)\Phi(k-m-1)]^{-1} \varphi_{2n}^T(k-m) \left[ \frac{\beta^2}{\alpha^2} \varphi_{2n}(k-m) [\Phi^T(k-m-1)\Phi(k-m-1)]^{-1} \varphi_{2n}^T(k-m) + 1 \right]^{-1} \\ & \cdot \varphi_{2n}(k-m) \frac{\beta^2}{\alpha^2} [\Phi^T(k-m-1)\Phi(k-m-1)]^{-1} \varphi_{2n}^T(k-m)y(k) \end{aligned} \quad (1.25)$$

Since

$$\tilde{\Theta}_{2n}(k-1) = [\Phi^T(k-m-1)\Phi(k-m-1)]^{-1} \Phi^T(k-m-1)Y(k-1) \quad (1.26)$$

Noted that

$$\Phi(k-m-1) = \begin{bmatrix} \alpha \cdot \Phi(k-m-2) \\ \beta \cdot \varphi_{2n}(k-m-1) \end{bmatrix}, Y(k-1) = \begin{bmatrix} \alpha \cdot Y(k-2) \\ \beta \cdot y(k-1) \end{bmatrix} \quad (1.27)$$

(1.26) can also be expressed as the following

$$\begin{aligned} \tilde{\Theta}_{2n}(k-1) = & \left[ \alpha^2 \Phi^T(k-m-2)\Phi(k-m-2) + \beta^2 \varphi_{2n}^T(k-m-1)\varphi_{2n}(k-m-1) \right]^{-1} \\ & \cdot \left[ \alpha^2 \Phi^T(k-m-2)Y(k-2) + \beta^2 \varphi_{2n}^T(k-m-1)y(k-1) \right] \end{aligned} \quad (1.28)$$

Letting

$$P(k-1) = \left[ \Phi^T(k-m-1)\Phi(k-m-1) \right]^{-1} \quad (1.29)$$

Due to (1.27), equation (1.29) is also in the following form,

$$P(k-1) = \left[ \alpha^2 \Phi^T(k-m-2)\Phi(k-m-2) + \beta^2 \varphi_{2n}^T(k-m-1)\varphi_{2n}(k-m-1) \right]^{-1} \quad (1.30)$$

Based upon (1.26) and (1.29), (1.25) can be simplified as

$$\tilde{\Theta}_{2n}(k) = \left( I - P(k-1) \frac{\varphi_{2n}^T(k-m)\varphi_{2n}(k-m)}{\varphi_{2n}^T(k-m)P(k-1)\varphi_{2n}^T(k-m) + \alpha^2 / \beta^2} \right) \left( \tilde{\Theta}_{2n}(k-1) + \frac{\alpha^2}{\beta^2} P(k)\varphi_{2n}^T(k-m)y(k) \right) \quad (1.31)$$

Since

$$P(k) = \left[ \alpha^2 \Phi^T(k-m-1)\Phi(k-m-1) + \beta^2 \varphi_{2n}^T(k-m)\varphi_{2n}(k-m) \right]^{-1} \quad (1.32)$$

then we have

$$P(k) = \frac{1}{\alpha^2} P(k-1) \left( I - \frac{\varphi_{2n}^T(k-m)\varphi_{2n}(k-m)}{(\beta^2 / \alpha^2)\varphi_{2n}^T(k-m)P(k-1)\varphi_{2n}^T(k-m) + 1} \frac{\beta^2}{\alpha^2} P(k-1) \right) \quad (1.33)$$

Noted that  $P(k)$  can be expressed as the following to guarantee that it is symmetric during the numerical calculations:

$$P(k) = \frac{1}{\alpha^2} \left( P(k-1) - \frac{(\varphi_{2n}(k-m)P(k-1))^T \varphi_{2n}(k-m)P(k-1)}{\varphi_{2n}^T(k-m)P(k-1)\varphi_{2n}^T(k-m) + \alpha^2 / \beta^2} \right) \quad (1.34)$$

Therefore the adaptive recursive prediction algorithm can be expressed as

$$\begin{cases}
\tilde{\Theta}_{2n}(k) = \left( I - P(k-1) \frac{\varphi_{2n}^T(k-m)\varphi_{2n}(k-m)}{\varphi_{2n}(k-m)P(k-1)\varphi_{2n}^T(k-m) + \alpha^2 / \beta^2} \right) \left( \tilde{\Theta}_{2n}(k-1) + \frac{\alpha^2}{\beta^2} P(k-1)\varphi_{2n}^T(k-m)y(k) \right) \\
P(k) = \frac{1}{\alpha^2} \left( P(k-1) - \frac{(\varphi_{2n}(k-m)P(k-1))^T \varphi_{2n}(k-m)P(k-1)}{\varphi_{2n}(k-m)P(k-1)\varphi_{2n}^T(k-m) + \alpha^2 / \beta^2} \right) \\
\tilde{\Theta}_{2n}(1) = \bar{\Theta}_{2n}, P(1) = [\Phi_N^T(1-m)\Phi_N(1-m)]^{-1}
\end{cases}
\tag{1.35}$$

Note that the initial values of  $\tilde{\Theta}_{2n}(1)$  and  $P(1)$  can be obtained using the approach described in the fixed-gain prediction subsection with pre-collected data. Although the adaptive recursive algorithm calculates the regression vector online, due to the iterative nature, the computational load required to execute the algorithm in real-time is relatively low. This will be demonstrated in subsection 2.4.5.

From the definition in (1.35), it is clear that matrix  $P(k)$  ( $k=1, 2, \dots$ ) is a monotonically decreasing positive semi-definite and symmetric matrix when  $\alpha = \beta = 1$ . Since a monotonically decreasing sequence bounded below will converge to a limit above or equal to its lower bound. Once  $P(k)$  is converged,  $\tilde{\Theta}_{2n}(k)$  will become a constant vector, and the algorithm becomes a fixed-gain prediction algorithm. As a result, the multi-step prediction algorithm will not be able to adapt to different driver behaviors. However, in the practical application, we would like to let  $\tilde{\Theta}_{2n}(k)$  change at certain rate to reflect the change of driver behavior, so that the predicted torque will also be affected by the driver behaviors, which is the main motivation that parameters  $\alpha$  and  $\beta$  are added as weighting factors so that the weight of past and current data can be adjusted in the regression algorithm.

Based upon the definition of  $P(k-1)$  in appendix, it is necessary to solve the  $P(k)$  iteratively, since  $P(k)$  contains newly sampled information  $\varphi_{2n}(k-m)$ . Parameter  $\alpha \in (\xi, 1]$  is mainly used as

forgetting factor so that the effect of the old data could be diminished. From equation (1.35), it can be observed that when  $\alpha < 1$ , matrix  $P(k)$  might not be a monotonically decreasing, since  $\alpha^{-2}$  is greater than 1. It can also be observed that the past data is weighted by  $\alpha^k$ , and as  $k$  goes to infinity the effect of the past reduces so that the current data becomes important. In this case, the adaptive algorithm can adapt to different driving patterns and behaviors.

For vehicles under different driving cycles with different drivers, the driving behavior could be quite different. To adapt to different driver behaviors, it is desirable to eliminate the influence of extremely old data and incorporate current ones [39] due to the process changing. In this study,  $\alpha$  is chosen to be less than one to forget the past data, that is, to get the effect of the past data decay exponentially. The smaller the coefficient  $\alpha$ , the faster the past data is discarded. Parameter  $\beta$  is used to weight the current data. The bigger the coefficient  $\beta$ , the faster the regression model adapts to current driving behavior.

However, even with a fixed  $\alpha$ , the decay rate of the past data could change due to different driving patterns and behaviors. High decay rate could reduce prediction accuracy due to lack of enough past information. For instance, when the vehicle is stopped at the traffic light, no new information will be provided by the acceleration pedal during that period, and the past data is the only information available for the prediction. To solve this problem, it is proposed to maintain the size (norm) of matrix  $P(k)$  constant by adaptively adjusting parameter  $\alpha$ . The following adaptive scheme is used.

$$\bar{\alpha}(k) = \alpha(k-1) - \gamma \cdot e(k-1), \quad \alpha(k) = \text{sat}_{\alpha}(\bar{\alpha}(k)) \quad (1.36)$$

where  $\text{sat}_{\alpha}(\cdot)$  is a saturation function to keep  $\alpha \in (\xi, 1]$ ; and  $\gamma > 0$  is the adaptive gain. Error  $e(k)$  is defined below

$$\bar{e}(k) = \frac{1}{p} \sum_{i=0}^{p-1} (P_n - \|P(k-i)\|_F), \quad e(k) = \text{sat}_e(\bar{e}(k)), \quad p > 0 \quad (1.37)$$

where  $\text{sat}_e(\cdot)$  is a saturation function to limit error  $e(k)$  to be between  $-\text{err}_{\max}$  and  $\text{err}_{\max}$ , and  $\text{err}_{\max} > 0$ ;  $P_n$  is the given norm target for matrix  $P(k)$ ; and  $\|\cdot\|_F$  represents the Frobenius norm.

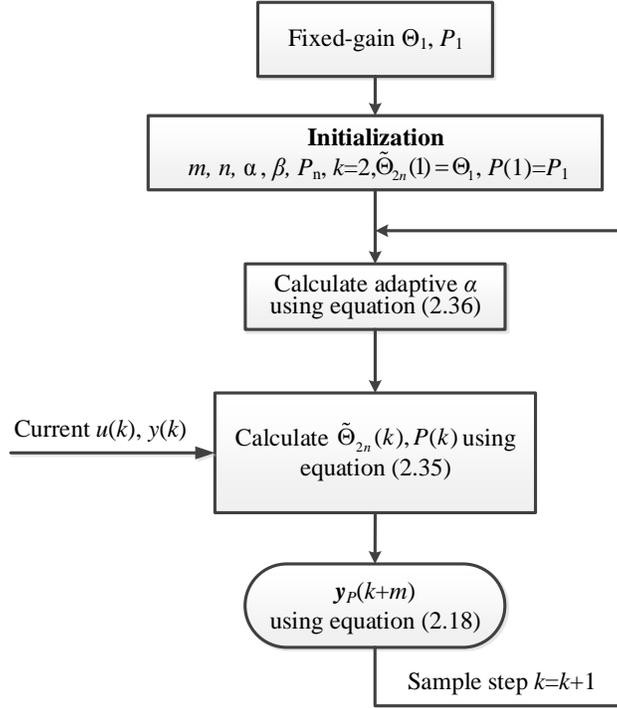


Figure 2. 5 Flow chart of adaptive recursive prediction algorithm

From (1.21), one may say that only one parameter (the ratio of  $\alpha$  and  $\beta$ ) is needed for the recursive regression algorithm since cost function (1.21) is equivalent to the cost function with  $\alpha = 1$  and free parameter  $\beta$ . However, for each iteration step, we are minimizing a different cost function, which makes parameter  $\alpha$  independent of  $\beta$ . This can be observed from  $P(k)$  in (1.35), where  $\alpha^{-2}$  is an independent parameter. The specific influences of parameters  $\alpha$  and  $\beta$  to the recursive regression prediction will be discussed in the simulation section.

Note that, in this research, the driving data was collected from a PID simulation driver model based on different driving cycles, which could be different from a real driver on the vehicles or

driving simulators. Therefore, the future work is to validate the proposed algorithm under actual driving conditions.

The adaptive recursive prediction algorithm, using fixed-gain regression parameters as initial regression parameter and updating the regression coefficients online, is shown in Figure 2. 5.

### 2.3 Hybrid electric vehicle model

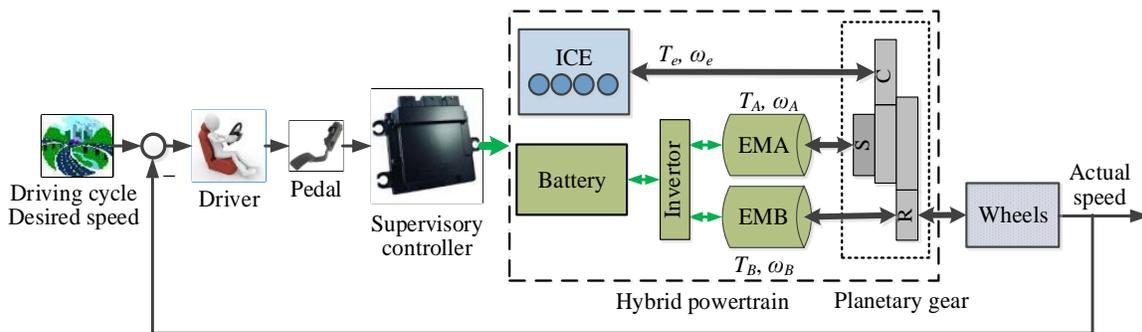


Figure 2. 6 Architecture of the HEV model

There are two different approaches for the HEV modeling: the backward and forward modeling. The backward model calculates the powertrain fuel consumption based upon the vehicle speed of a given driving cycle. Therefore, the driver model is not required since it is assumed that the given driving cycle speed is the vehicle speed. The benefit of the backward vehicle model is its simplicity and low computational load [40]. This type of model is often used for study vehicle fuel economy. The forward modeling approach tracks the driving cycle vehicle speed using a driver model, where the driver model tries to follow the given driving cycle speed. The advantage of the forward modeling approach is that it simulates the actual vehicle behavior assuming that the driver model is accurate. It provides practical vehicle simulations with higher computational load than that of backward model. The forward vehicle model is often used for steady transient vehicle performance study such as fuel economy and emissions.

The HEV model is constructed in the MATLAB/Simulink, which is shown in Figure 2. 6.

### 2.3.1 Driver model

The driver model consists of two PID controllers: one is for controlling the accelerator pedal, and the other is for brake pedal, see Figure 2. 7. Both controllers using the vehicle speed error as the input.

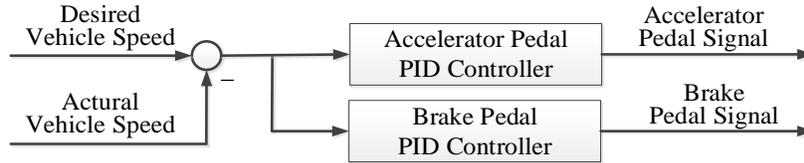


Figure 2. 7 Driver model

The driver model control parameters are shown in Table 2. 1. These parameters are tuned to minimize the vehicle speed tracking error. Note that the parameters can also be tuned to reflect different driver behaviors, and usually relatively high “D” gain represents aggressive driver behavior. Meanwhile, to let powertrain subsystem have the proper acceleration pedal information, a first order low pass filter is used to condition the raw pedal position output signal; see Figure 2. 1 and Figure 2. 3. The low pass filter parameter is included in Table 2. 1.

Table 2. 1 Driver model control parameters

Item	Parameter	Value
Accelerator Pedal PID Controller	Proportional gain, $K_p$	150
	Integral gain, $K_i$	10
	Derivative gain, $K_d$	0 (default)
Brake Pedal PID Controller	Proportional gain, $K_p$	200
	Integral gain, $K_i$	0.01
	Derivative gain, $K_d$	0
Low Pass Filter	Time constant, $\tau$	0.05

### 2.3.2 Hybrid powertrain model

The hybrid powertrain model includes an IC engine, two EMs and a planetary gear. EM can work as a generator or a motor. At low speeds, the vehicle can be powered by EMs and/or the IC engine; see Figure 2. 6.

### 2.3.2.1 IC Engine model

The output power of the IC engine is modeled as a function of fueling input  $\dot{m}_e$  below

$$\tau_e \dot{P}_e = -P_e + \eta_e(T_e, \omega_e) \cdot P_{in}(\dot{m}_e, H_{LHV}), P_e = T_e \omega_e \quad (1.38)$$

where  $P_e$  is the IC engine power output;  $\eta_e$  is the efficiency map of the IC engine as a function of engine output torque  $T_e$  and speed  $\omega_e$ ;  $\tau_e$  is the IC engine time constant;  $P_{in}$  is the input fuel energy as a function of fuel mass flow rate  $\dot{m}_e$  and the corresponding fuel low heating value  $H_{LHV}$ .

### 2.3.2.2 Electric motor/generator model

Since the response time of EM is much fast than the IC engine [41], the EM is modeled without dynamics as follows:

$$P_m = T_m \omega_m = \eta_m(T_m, \omega_m) P_{elec}(U, I_m) \quad (1.39)$$

where  $P_m$  is the motor power output and,  $\eta_m$  is efficiency map of motor as a function of motor output torque  $T_m$  and speed  $\omega_m$ ;  $P_{elec}$  is the electric power input, a function of electric bus voltage  $U$  and motor current  $I_m$ . When the motor is operated under the generator mode, it is modeled as

$$P_g = UI_g = \eta_g(T_m, \omega_m) P_m(T_m, \omega_m) \quad (1.40)$$

where  $P_g$  is the generated power;  $I_g$  is the generator current; and  $\eta_g$  is the efficiency map of generator.

### 2.3.2.3 Planetary gear model

The planetary gear is connected to all the mechanical power plants and acts as a transmission to split power between the engine and EMs. For the planetary gear set configuration, the following equation holds

$$\omega_R R + \omega_S S = \omega_C (R + S) \quad (1.41)$$

where  $\omega_R$ ,  $\omega_S$ , and  $\omega_C$  are the rotational speed of ring gear, sun gear, and carrier gear, respectively;

and  $R$  and  $S$  are the number of ring and sun gear teeth, respectively.

The rotational gear speeds and torques satisfy the following equations,

$$\begin{cases} \omega_A = \omega_S = \omega_C(1 + R/S) - \omega_R R/S, \\ \omega_B = \omega_R = v_{veh} f_d / r_w \\ \omega_e = \omega_C \end{cases} \quad (1.42)$$

$$\begin{cases} T_A = T_S = -T_C S / (R + S) \\ T_B = T_R = F_t r_w / f_d - T_C R / (R + S) \\ T_e = T_C \end{cases} \quad (1.43)$$

where  $v_{veh}$  is the vehicle linear speed;  $f_d$  is the final drive ratio;  $r_w$  is the radius of the wheel;  $T_R$ ,  $T_S$  and  $T_C$  are the ring gear, sun gear and carrier gear torques, respectively; and  $F_t$  is the traction force.

#### 2.3.2.4 Battery model

Battery state of charge (SOC) model is relatively complicated compared with other subsystem models in an HEV. The simple and effective Rint model [42] is used and, the estimation of SOC can be expressed as follows:

$$SOC = \frac{Ah_{max} - Ah_{used} \eta_{coulomb}}{Ah_{max}} \quad (1.44)$$

where  $\eta_{coulomb}$  is the coulomb efficiency of the battery,  $Ah_{max}$  is the maximum capacity of the battery, and  $Ah_{used}$  is the used capacity of the battery.

Noticed that SOC should be maintained within its predefined bounds to protect the battery for extended life.

Thermal model was also developed to monitor the effects of the average internal battery temperature to be used to estimate the battery capacity under driving condition, and the model is based upon as the following equation:

$$T_{bat} = \int_0^t \frac{Q_{gen} - Q_{case}}{m_{bat} \cdot c_p} d\tau \quad (1.45)$$

where  $Q_{gen}$  is the heat generated by the battery calculated from coulomb and internal resistance losses;  $Q_{case}$  is the combined heat of the conduction and convection from the battery to the air;  $m_{bat}$  is the mass of energy storage module;  $c_p$  is the average heat capacity of the module. Note that the battery temperature changes gradually under a driving cycle and the desired driving torque changes at a much fast rate. In the simulation study in this research the battery capacity within the prediction horizon is based upon the current battery temperature.

### 2.3.3 Supervisory controller

Table 2. 2 Technical and simulation parameters of the HEV

Item	Parameter	Value
Vehicle	Mass	2956 kg
Engine	Type and description	90-degree V-type, liquid-cooled
	Power output	345 hp (257.2 kW) @ 5300 rpm
	Torque	380 lb-ft. (515.2 N.m) @ 4200 rpm
	Maximum Engine Speed	5800 rpm
Generator	Type	Brushless permanent magnet
	Rated/Peak power	100/150kW
EM	Type	Brushless permanent magnet
	Rated/Peak power	200 /300 kW
	Maximum torque	1300 N.m
	Speed range	0-5000 r/min
Battery	Type	Lithium battery pack
	Rated voltage, Voltage range	360V, 300-420 V
	Rated capacity	90 Ah
Simulation	Default driving cycle	FTP
	Battery initial temperature	-30°C
	Prediction steps	50 (0.5s)
	Simulation solver	Fixed-step, 0.01s

Based upon the driving torque (desired torque) demand, the supervisory controller controls the IC engine and electric machine powers on the basis of the sub-system states such as SOC, engine and EM speed and load to meet the torque requirement. Noted that the prediction algorithm is part

of the supervisory controller located right after the driver model; see Figure 2. 6. It uses both the raw and conditioned acceleration pedal position (desired torque) signals as inputs to predict future desired torque. The detailed control strategy is not provided here.

The main technical parameters of the HEV and simulation variables in this research are listed in Table 2. 2.

## 2.4 Simulation study and validation

### 2.4.1 Weighting factor effect

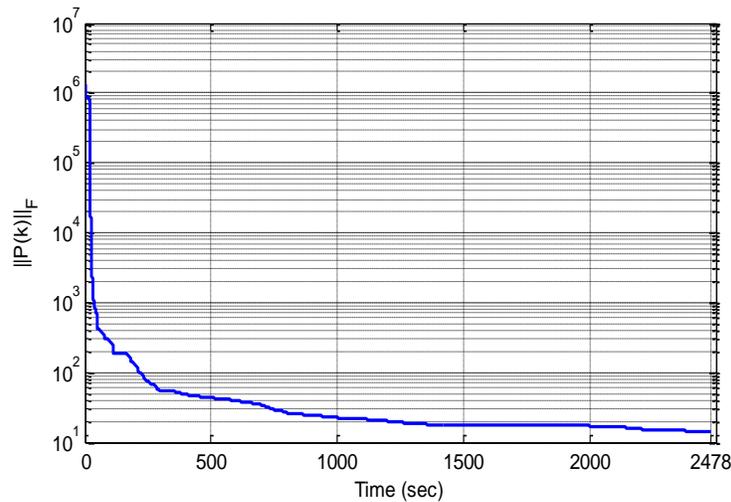


Figure 2. 8  $\|P(k)\|_F$  trace with  $\alpha = \beta = 1$  under FTP driving cycle

From equation (1.35) it can be observed that when  $\alpha = \beta = 1$ ,  $\|P(k)\|_F$  is a monotonically decreasing function of time. Figure 2. 8 shows that under the FTP driving cycle  $\|P(k)\|_F$  would gradually converge to zero as time increases, leading to a fixed-gain prediction as described in subsection 2.2.3. This could result in low prediction accuracy since the prediction algorithm cannot adaptive to the different driving patterns and behaviors.

However, letting  $\alpha$  to be a constant less than 1, for example  $\alpha = 0.999$  with  $\beta = 1$ , could lead to

unbounded  $\|P(k)\|_F$  and may result in increasing prediction errors. Therefore, a non-constant adaptive  $\alpha$  proposed in (1.36) is required.

The adaptive  $\alpha$  algorithm described in equations (1.36) and (1.37) has a few parameters to be selected. The following list describes the parameter selection process:

**Saturation bounds for  $\alpha$  :** The upper bound for  $\alpha$  is 1 since  $\alpha$  greater than 1 leads to over emphasizing the past data; and the lower bound of  $\alpha$  is equal to  $\xi < 1$  and was selected as 0.9995 for this application to make sure that the past data will be forgotten gradually. Note that the forgiving effect of the given  $\alpha$  over time is dependent on the sample period of the prediction algorithm. For the same forgiving effect, the required  $\alpha$  approaches one as the sample period reduces. Since the sample period is 0.01s, the selected  $\alpha$  is very close one at 0.9995. The selected bounds are shown in Table 4.

**Adaptive gain  $\gamma$  :** Parameter  $\gamma$  is related to the convergence rate of  $\|P(k)\|_F$ . Large  $\gamma$  leads to fast convergence but could also cause the adaptive process unstable so it is a tune coefficient  $\gamma$  is selected as  $3 \times 10^{-6}$ ; see Table 2. 3.

**Error bounds for  $e(k)$  :** Error bounds are used to limit the effect of large error to a fairly slow adaptive process. The absolute bound  $err_{\max}$  was selected to be 10% of  $P_n$  in this study.

Table 2. 3 Influence of prediction steps (FTP cycle)

Item	Value	Item	Value
$sat_\alpha(\alpha)$	[0.9995, 1]	$sat_e(e)$	[-40, 40]
$p$	10	$\gamma$	$3 \times 10^{-6}$
$P_n$	$4 \times 10^2$	$m$	50

**Parameters  $p$  and  $P_n$  :** These two parameters are tuned to have the minimal prediction error; see values in Table 2. 3.

The simulation is conducted under the FTP driving cycle with the parameters shown in Table

2. 3, and  $\|P(k)\|_F$  converges to its target value  $4 \times 10^2$  within about 50 seconds and remains at the same level afterwards; see in Figure 2. 9. The  $\alpha$  stays between its saturated range  $[0.9995, 1]$ . From Figure 2. 9 it can also be observed that adaptive coefficient  $\alpha$  equals to one when the vehicle speed is close to constant or zero. Note that in this case the acceleration pedal position will also remain close to constant, leading to almost no new information provided. In this case, coefficient  $\alpha$  is selected to one so that the past information will not be forgotten and can be used for prediction.

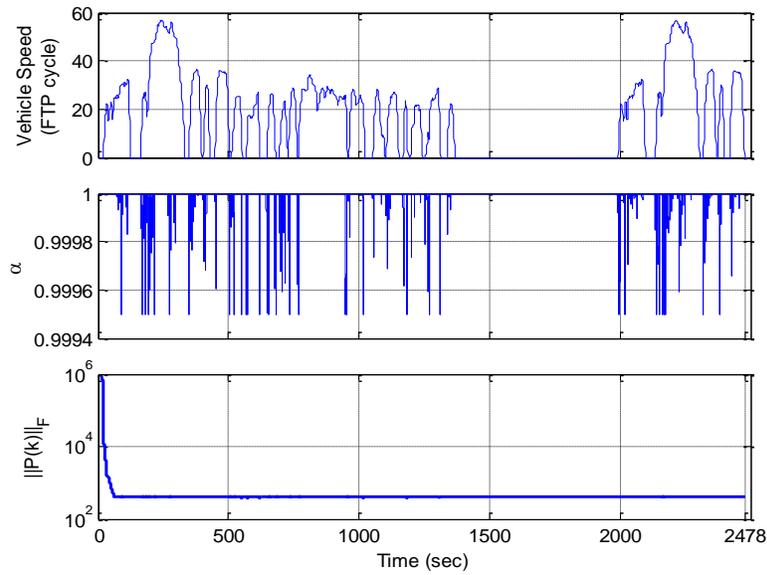


Figure 2. 9  $\|P(k)\|_F$  and adaptive  $\alpha$  changing with  $\beta = 1$

Parameter  $\beta$  is mainly used to influence prediction performance at the beginning, so that the initial prediction error can be reduced. This is due to the fact that as prediction continues, the influence of  $\beta$  would be significantly reduced. Figure 2. 10 shows prediction error variations as  $\beta$  changes with  $\alpha = 1$ . It can be observed that as  $\beta$  increases, the initial predictive error reduces significantly and the error remains almost unchanged after 80 seconds. Also, when  $\beta$  is greater than 20, the initial prediction error improves a little. Therefore,  $\beta$  was chosen to be 20.

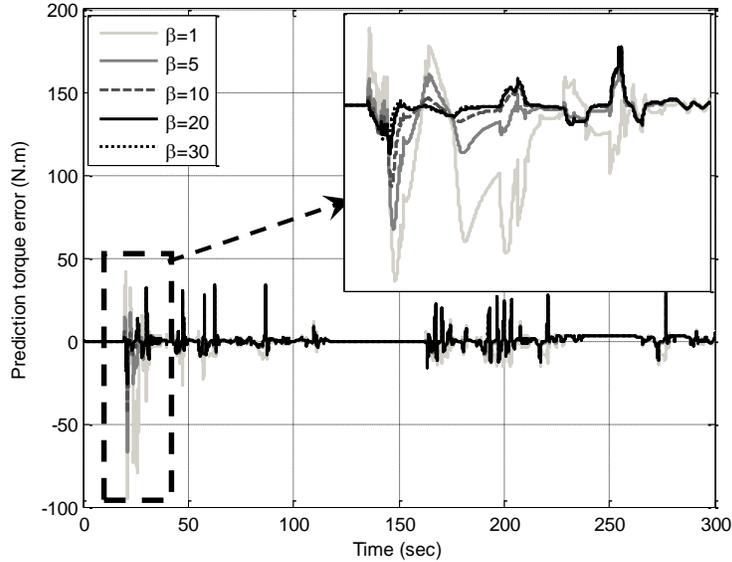


Figure 2.10 Influence of  $\beta$

### 2.4.2 Prediction algorithm validation and comparison

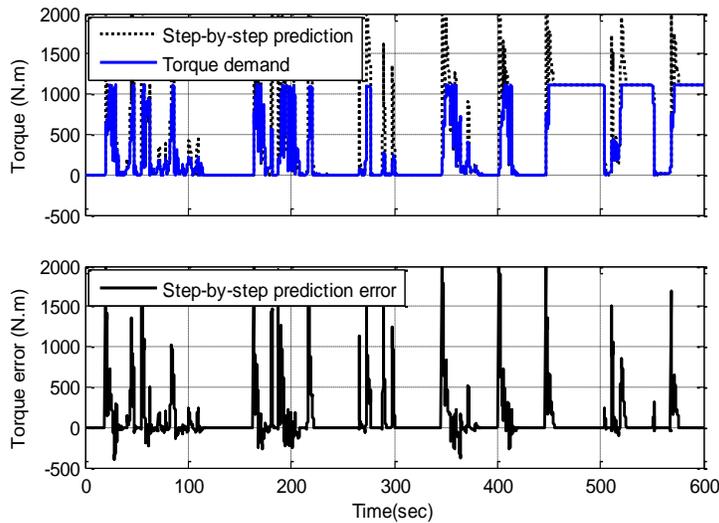


Figure 2.11 50-step step-by-step prediction

The step-by-step prediction algorithm described in Figure 2. 2 of subsection 2.2.1 realizes  $m$ -step prediction by iteratively applying equations (1.9) to (1.11) for  $m$  times. However, the prediction errors could be fairly high due to the error accumulation by repeatedly using equations (1.9) to (1.11). The other disadvantage is that the raw pedal position output is not used. Figure 2. 11 shows step-by-step prediction results for  $m=50$ , where the dotted line represents predictive

torque for the first 600 seconds of the FTP cycle, and the solid one is the torque demand (conditioned pedal position output). It is clear that the peak prediction error is as much as 200% of its torque demand, indicating that it cannot be used in practice.

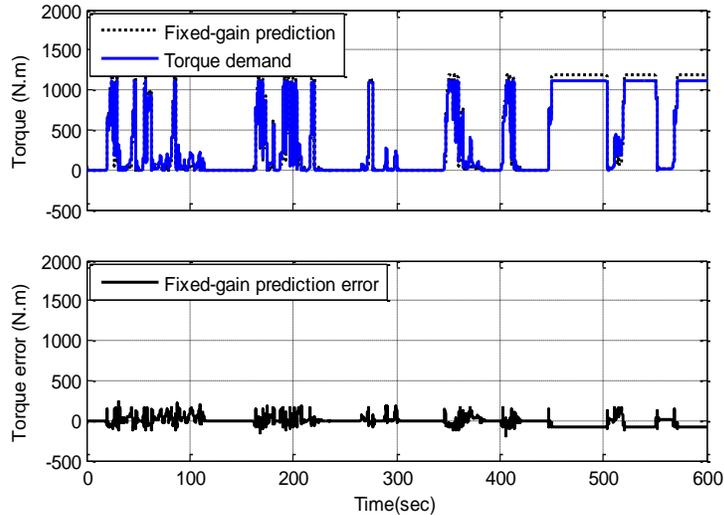


Figure 2. 12 50-step fixed-gain prediction

Fixed-gain prediction algorithm is capable of making multi-step prediction directly based upon both conditioned and raw pedal position outputs. However, the regression gain is fixed and is not updated online. Although the fixed-gain method gets rid of the cumulated error, the prediction error is still fairly high. Figure 2. 12 shows the 50-step fixed-gain prediction results, where the dotted and solid lines represent the predicted torque and actual torque demand, respectively. It can be seen that the peak torque error reduced notably from 200% to 22% over the first 600 seconds of the FTP driving cycle, but the prediction error is still too high to be applied in applications. Note that there is also a steady state prediction error between 400s and 600s (see Figure 2. 12) since the DC-gain is not equal to one for the fixed-gain prediction transfer function.

The adaptive recursive prediction algorithm is proposed to overcome drawbacks of both step-by-step and fixed-gain prediction algorithms with following features:

Regression gain is updated online to adapt to different driving patterns and behaviors.

Weighting factors ( $\alpha$  and  $\beta$ ) are introduced to scale the past and current data; and  $\alpha$  is adaptively adjusted also to overcome the numerical over and under flow issues; see the discussion in subsections 2.2.3 and 2.4.1.

Online computational load is significantly reduced due to the iterative nature, and it will be discussed in subsection 2.4.5 and therefore, it is suitable for real-time prediction.

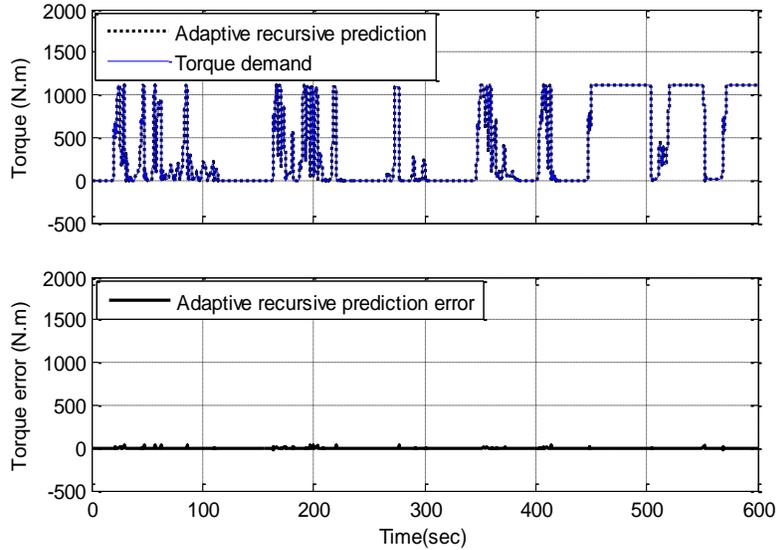


Figure 2. 13 Adaptive recursive prediction

Figure 2. 13 shows the simulation results under the first 600 seconds under FTP driving cycle. It indicates that the adaptive recursive prediction algorithm reduces the peak prediction error from 22% to approximately 4% comparing with the fixed-gain predictions. This makes the proposed algorithm the best among the three prediction algorithms. Note that the regression order is chosen as 6, see the next subsection for the order selection process.

Simulation results indicate that adaptive recursive algorithm provides a fairly good performance for the multi-step prediction under FTP driving cycle. However, the FTP driving cycle covers only a certain driving conditions. To check the prediction accuracy under other conditions, four other typical driving cycles, IM240, NYCC, US06 and ARB02, are used to compare with the FTP simulation results. Figure 2. 14 shows the simulation results under all five

driving cycles, and the peak prediction errors are all below 4% of their torque demand.

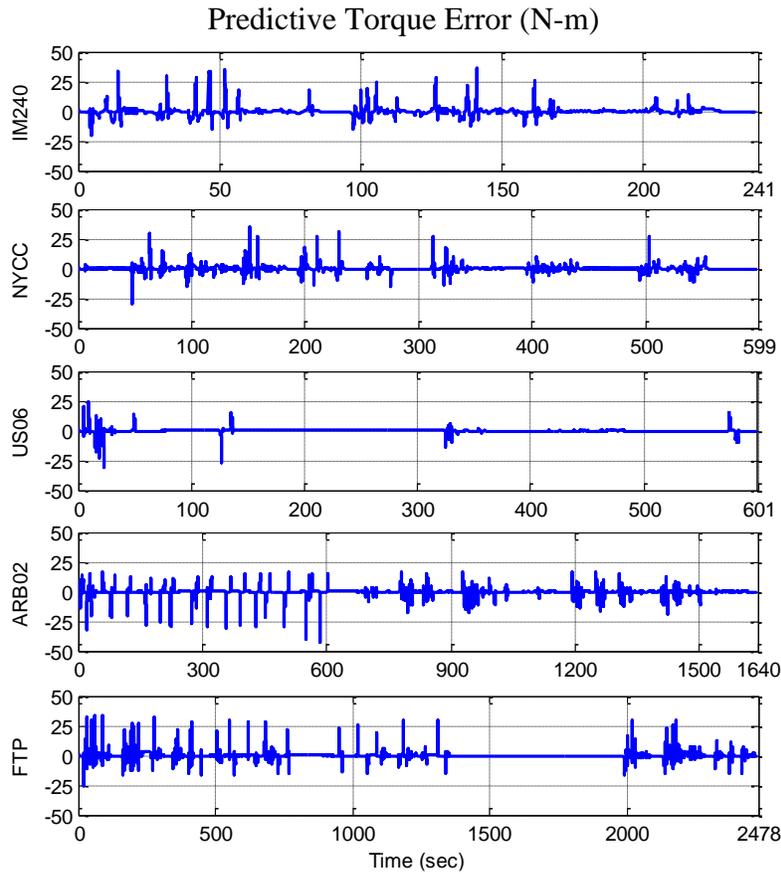


Figure 2.14 Driving cycle validation

### 2.4.3 Influence of regression order

As shown in (1.35), for the adaptive recursive algorithm,  $\phi_{2n}(k)$  contains past data information, and the regression order ( $2n$ ) depends on numbers of how many past  $u$  and  $y$  were used. Normally, the higher the regression order, the smaller the prediction error. However, too much past data could also lead to the poor prediction model as indicated below in the simulation study. On the other hand, high prediction model order would increase the computational load exponentially, making the online application impossible. The study conducted in this subsection is to find a proper regression order with satisfactory prediction error.

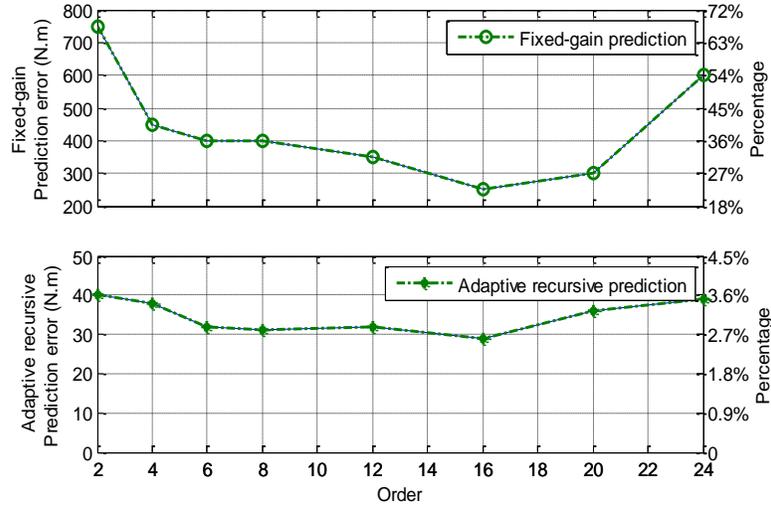


Figure 2. 15 Prediction errors with different vector orders

50-step prediction errors with different regression orders under FTP cycle are studied, and the simulation results are shown in Figure 2. 15. It can be observed that for the fixed-gain prediction, the maximum prediction errors range from 68% to 22%. The smallest error occurs at regression order 16. However, for the adaptive recursive prediction algorithm, the maximum predictive error is under 3.6% over the regression order range from 2 to 24. Although the smallest prediction error occurs at order 16, the prediction error remains almost constant between order 6 and 16. Considering the computational load, regression order 6 was chosen due to the following reasons:

- The computational load decreases dramatically from  $O(10 \times 16^2)$  to  $O(10 \times 6^2)$  (see subsection 2.4.5) with a less than 0.03% increment of prediction error;
- The prediction error will be increased significantly if regression order 4 is selected.

#### 2.4.4 Prediction robustness

Prediction robustness to different driver behaviors is an important prediction property. As stated in subsection 2.2.3, under different driving cycles with different drivers, driving behavior could be quite different. To have different driver behaviors, the driver model parameter (the derivative or “D” gain of the PID controller) is adjusted. Therefore, an aggressive or hasty driving

behavior can be modeled with a relative high “D” gain. As the driving behavior gets aggressive or hasty, the “D” gain increases. The influence of “D” gain changing on accumulated torque errors (ATE) is shown in Figure 2. 16, where the ATE is defined as

$$ATE \triangleq \sum abs(Torque\ error) / \sum abs(Torque) \quad (1.46)$$

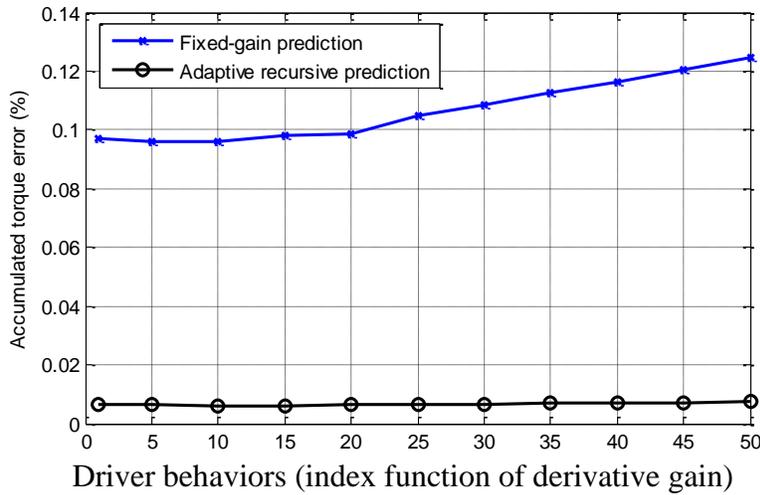


Figure 2. 16 Comparison of accumulated torque error

It can be observed from Figure 2. 16 that with the increment of derivative gain, the fixed-gain ATE goes up, while the adaptive recursive ATE remains almost the same, indicating the adaptive recursive prediction is robust to different driver behaviors, which is a very important property in practical applications. As stated before, in this research, the driver behavior data is only collected from a PID driver model based on different driving cycles, which may be quite different from the data collected from a real driver on a real vehicle or a simulator. Therefore, one of our future works is to validate the proposed algorithm under actual driving situations.

#### 2.4.5 Prediction error and computational load

Table 2. 4 shows the prediction errors of all three studied prediction algorithms in this research, where the step-by-step algorithm has the highest prediction errors; the fixed-gain algorithm is with the medium prediction errors among the three algorithms; and the adaptive recursive algorithm is

the most accurate one. It is also interest to see associated prediction algorithm computational load even though it seems unfair to compare the computational efficiency of a recursive algorithm with a matrix inversion-based algorithm. Note that the computational load of a prediction algorithm can be expressed as time complexity and formulated as

$$T(n) = O(f(n)) \quad (1.47)$$

where  $T(n)$  is the maximum amount of time used for any input of size  $n$ ;  $O(\cdot)$  is the time complexity notation;  $f(n)$  is the computational load growth rate [43], [44]. The approximate computational load growth rate are listed in Table 2. 4 for all three algorithms. From Table 2. 4, it can be see that the adaptive recursive algorithm has the time complexity of  $O(360)$ ; the step-by-step algorithm has the time complexity of  $O(1664)$ ; and the fixed-gain algorithm has the lowest time complexity of  $O(16)$  since its prediction model is not updated online. As a summary, the adaptive recursive algorithm has the smallest estimation error with relatively low time complexity due to the fact that there is no matrix inverse calculation with a relatively low regression order 6.

Table 2. 4 Prediction accuracy ( $n_1 = 8, n_2 = 6$ )

	<b>Step-by-step</b>	<b>Fixed-gain</b>	<b>Adaptive recursive</b>
<b>Prediction errors</b>	200%	22%	4%
$f(n)$	$2(n_1)^3 + 10(n_1)^2$	$2n_1$	$10(n_2)^2$
<b>Time Complexity</b>	$O(1664)$	$O(16)$	$O(360)$

## 2.5 Conclusions

Three desired torque prediction algorithms are studied in this research, step-by-step, fixed-gain, and the proposed adaptive recursive prediction algorithms. Through simulation study under FTP and other four typical driving cycles, the step-by-step prediction algorithm is not capable of real-time application due to the heavy computational load requirement and very high prediction error.

Although the fixed-gain prediction algorithm reduces the computational load significantly with much lower prediction error than that of the step-by-step one, the prediction error is still too high to be used in applications. The proposed adaptive recursive prediction algorithm updates the prediction gains online and its relatively low computational load makes it feasible for real-time applications. The simulation results also show that the adaptive recursive prediction algorithm reduces the prediction error significantly with a 4% maximum error. The introduced weighting factors for the past and current data are the key for improving prediction accuracy and avoiding the over and under flow issues during recursive calculations. In addition, due to the online updating of regression gains, the proposed algorithm is also robust to various driver behaviors. Furthermore, the proposed algorithm has the lowest regression order among the three, leading to significant low computational load especially compared with the step-by-step prediction algorithm.

## **CHAPTER 3: STOCHASTIC PREDICTIVE BATTERY BOUNDARY MANAGEMENT**

### **3.1 Introduction**

Due to the prominent advantages of increased fuel economy with reduced emissions, Hybrid Electric Vehicles (HEVs) outperform conventional vehicles powered only by the internal combustion (IC) engine. As stated in the previous work [45], when the driver requested power is greater than the battery output power limit and the engine is off, slow response of the IC engine, especially during the crank-start, leads to over-discharge the battery to meet the output power requirement until the engine is ready. Note that the battery performance decreases significantly over time especially at low temperature [46]-[47] due to the increment of its internal resistance and inherent decrement of its energy storage capacity [5]-[6]. Therefore, when the drivetrain torque is solely provided by the EM(s) under cold start operations with low battery temperature, the battery is more likely to be over-discharged than at the normal temperature. In this case, the actual battery power output will exceed its power output limit. Over-discharging is inevitable due to rapid transient powertrain responses required by real-world driving. However, frequent or persistent over-discharging will permanently damage the battery, leading to reduced battery life. If the total battery over-discharging duration is reduced, the battery useful life can be extended. Note that the battery over-charging is also a very interesting topic but it is not the subject of this chapter.

Many control schemes have been developed to prevent batteries from over-discharging as summarized in [48]. Low battery voltage sensing is a widely used method to prevent further power draw by reducing the EM power output. However, sudden reduction of powertrain power output not only reduces the driving performance but also puts the vehicle in danger of a collision. Low battery voltage sensing with power cutback is a similar scheme as the aforementioned one. It cutbacks the power by certain percentage when the battery reaches the given power threshold. This approach overcomes the sudden power cutback, but it is difficult to determine the percentage of the power cutback. State of Charge (SOC) estimation with power cutback scheme is proposed to limit battery discharging power, but selecting an SOC level to cutback power is a challenge. Reference [49] presents a road testing result of the ambient temperature influence to the HEV battery performance but it does not consider how to reduce or eliminate over-discharging. Reference [5] proposes two approaches to mitigate the impact of low temperature on battery performance: increasing the battery temperature with an external heater or combining the battery with a super-capacitor. An energy management strategy is presented in [50] to minimize the battery usage thereby to extend battery life, where the reduced battery usage is based on a small penalty on fuel consumption by limiting battery power output instead of preventing battery from over-discharging.

The literature review indicates that the prior research in this area is limited to either limiting the battery power output or using external device to increase battery temperature. The disadvantage of limiting battery power output is the degraded driving performance, and the disadvantage of using battery heating element is the increment of both system cost and vehicle weight. As a summary, the existing methods of preventing battery from over-discharging involve only in powertrain electric subsystems (e.g., EM and battery). However, as part of the hybrid powertrain,

the IC engine is able to provide additional power to the powertrain when the available electric power is limited; and once the engine is turned on, the engine can provide additional power to meeting the driver's power request and the battery over-discharging can be reduced or eliminated. However, the main issue is that the slow response characteristics of the IC engine, especially when the engine is fully stopped, can prevent it from providing the additional power on time. In this chapter, a stochastic predictive boundary management (SPBM) strategy is proposed based on three main signals, predicted torque demand [45], prediction error variance, and confidence level (see subsection 3.3), to proactively turn on the IC engine when the predicted power exceeds current battery capacity. In this way, the battery over-discharging duration can be significantly reduced. The prediction error variance is calculated in real-time and used to determine if the engine needs to be crank-started. A series-parallel forward HEV model [51] is developed in MATLAB/Simulink for developing and validating the proposed control strategy. In the hybrid powertrain model, the battery capacity and its maximum discharging/charging current are functions of temperature. That is, as the temperature decreases the battery maximum output power reduces, so does the maximum allowed discharging/charging current.

For comparison purpose, two control strategies are used in this chapter: the baseline power-follower control strategy (PFCS) [42], [52]-[54] (an extension from a series to parallel hybrid powertrain) and the predictive boundary management strategy (PBM) proposed in [55]. For the PFCS strategy, once the driving power demand is greater than the battery output power limit, the IC engine will be turned on to provide additional power to meeting the hybrid powertrain torque demand. However, it could take more than half second for the engine to be ready to make power available. While the engine is getting ready, the battery has to be over-discharged to meet the powertrain power output requirement, which can lead to the reduced battery life. However, based

on the predicted torque demand, the PBM strategy can proactively turn on the engine half second (or more) ahead before the actual power demand exceeds the battery output power limit. This reduces battery over-discharging duration effectively, leading to improved battery life and HEV performance.

The prediction error stochastic properties are analyzed under five typical driving cycles. It shows that the prediction error distribution can be modeled as an average of three different normal distribution functions, where the deviation of each normal distribution function is proportional to the prediction error variance. Most importantly the prediction error variance can be calculated online. Therefore, with the predicted torque and online calculated error variance, it is feasible to find a smallest upper bound of future desired power with a given confidence level. As a result, an SPBM strategy is proposed, where the smallest upper bound of future desired power is used to determine if the engine needs to be proactively turned on. Note that the smallest upper bound of future desired power level is a function of predicted torque, percentage of confidence level, and prediction error variance. Also, it is found that the percentage of confidence level can be used as a calibration to balance the battery over-discharging duration and fuel economy penalty. The effectiveness of SPBM strategy is validated in simulations under five typical driving cycles with different initial battery temperatures. Significant improvement is achieved over both baseline PFCS and PBM strategies.

The rest of the chapter is organized as follows. In section 3.2, a simple series-parallel hybrid vehicle model is presented along with the baseline PFCS. Section 3.3 analyzes the prediction error stochastic properties. The SPBM strategy is proposed in section 3.4; and simulation results are provided in section 3.5. Conclusions are added in section 3.6.

## 3.2 Hybrid electric vehicle subsystem models and its baseline control strategy

In this chapter, the same hybrid electric vehicle model described in Figure 2. 6 is used, also including the same driver model described in Figure 2. 7, the same EMB and EMA models described in (1.39) and (1.40), and planetary gear model described from (1.41) to (1.43).

### 3.2.1 Hybrid powertrain model

In this subsection, the models of the two main hybrid powertrain components are presented: IC engine and battery models.

#### 3.2.1.1 IC engine model

The IC engine is described as a first order dynamic subsystem similar to [45]. In the model, the engine start delay is included, and the output power ( $P_e$ ) is expressed as a function of engine input energy and engine efficiency map as follows,

$$\begin{cases} \tau_e \dot{P}_e = P_{ein} \cdot \eta_e(T_e, \omega_e) - P_e, P_e = T_e \omega_e & t \geq \tau_s \\ P_e = 0 & t < \tau_s \end{cases} \quad (2.1)$$

where  $P_e$  is the IC engine output power, a product of engine torque  $T_e$  and speed  $\omega_e$ ;  $P_{ein}$  is the engine input energy and  $\eta_e$  is the engine efficiency map;  $\tau_e$  is the IC engine time constant; and  $\tau_s$  is the engine crank-start delay time.

#### 3.2.1.2 Battery model

The SOC is one of the most important parameters in the battery management system, and in general it is defined as the ratio of the current battery capacity to the nominal fully-charged battery capacity [42], [52], [60]-[71]. However, at different temperatures the battery capacity could be different, and hence the SOC is modeled as a function of both discharging/charging current and battery temperature. Note that an accurate SOC model can be used to not only protect the battery from over-discharging but also improve fuel economy. In this chapter, the SOC is one of the

parameters used to determine when the IC engine shall be turned on or off. Therefore, considering the impact of temperature to battery capacity, the Ah Coulomb counting method is used for SOC calculation:

$$SOC(t) = SOC(t_0) - \int_{t_0}^t \frac{\eta \cdot I(\tau)}{C_m(T)} d\tau \quad (2.2)$$

where  $SOC(t_0)$  is the initial value of SOC;  $C_m(T)$  is the maximum battery capacity at temperature  $T$ , and normally the lower the temperature, the lower the battery capacity;  $\eta$  is the Coulomb efficiency defined as the ratio of discharged capacity to the capacity needed to be charged back to the initial state of the discharge [66], where  $\eta = 1$  for discharging and  $\eta < 1$  for charging;  $I(\tau)$  is the current at time  $\tau$  and it can be calculated using the following formulae

$$\begin{cases} I(\tau) = \frac{V_{OC} - \sqrt{V_{OC}^2 - 4r_{bat}(T)P_{br}}}{2r_{bat}(T)} \\ V_{OC} = f(SOC, T) \\ r_{bat}(T) = \begin{cases} r_{ch}(T) & \text{charge} \\ r_{dis}(T) & \text{discharge} \end{cases} \end{cases} \quad (2.3)$$

where  $P_{br}$  is the required battery output power;  $V_{OC}$  is the open circuit voltage that is a function of SOC and  $T$ ; and  $r_{bat}(T)$  is the battery internal resistance and different resistance values are used for charging and discharging due to the variant concentrations of available reactants for charging and discharging [67]. The battery current  $I(\tau)$  satisfies the following constrains:

$$\begin{cases} I(\tau) \leq I_{dis-max}(T) & I(\tau) > 0 \\ I(\tau) \geq I_{ch-max}(T) & I(\tau) < 0 \end{cases} \quad (2.4)$$

where  $I(\tau) > 0$  is for discharging, and  $I(\tau) < 0$  for charging;  $I_{dis-max}(T)$  is the maximum allowed battery discharging current (positive) at temperature  $T$ , and  $I_{ch-max}(T)$  is the maximum allowed battery charging current (negative) at temperature  $T$ . Note that  $I_{dis-max}(T)$  and  $I_{ch-max}(T)$  can be

calculated based on the constrain of depth of discharge (DOD), and be expressed as

$$\begin{cases} I_{\text{dis-max}}(T) = \frac{V_{OC} - V_p}{r_{\text{dis}}(T)} \\ I_{\text{ch-max}}(T) = \frac{V_p - V_{OC}}{r_{\text{ch}}(T)} \end{cases} \quad (2.5)$$

where  $V_p$  is the lower bound of DOD battery voltage that is a function of SOC and temperature  $T$ .

Notice that the SOC should be maintained within its predefined operational range to protect the battery and it can be seen from (2.2) that the SOC rate of variation is inversely proportional to  $C_m(T)$ . Therefore, under low temperature with reduced  $C_m(T)$ , the SOC drops more quickly than that under normal temperature. Also the DOD is only used to prevent the battery from extremely over-discharging.

Table 3. 1 Technical parameters of the HEV

Item	Parameter	Value
Vehicle	Mass	1750kg
Engine	Power output	43 kW (58hp) @ 4000 rpm
	Torque	102 N-m (75lbf-ft) @4000rpm
Generator	Type	Brushless permanent magnet
	Peak power	15 kW (20hp)
Motor	Type	Brushless permanent magnet
	Peak power	30 kW (40hp) @ 940~ 2000 rpm
	Maximum torque	305 N-m (225lbf-ft) @ 0~940 rpm
Battery	Type	Lithium battery pack
	Rated voltage	288 V
	Rated capacity	6.0 Ah

Thermal model is also developed to monitor the effects of average internal battery temperature for estimating the battery capacity using the following equation:

$$T_{\text{bat}} = \int_0^t \frac{Q_{\text{gen}} - Q_{\text{case}}}{m_{\text{bat}} \cdot c_p} dt \quad (2.6)$$

where  $Q_{\text{gen}}$  is the heat generated from Coulomb and internal resistance losses;  $Q_{\text{case}}$  is the combined

heat of the conduction and convection from the battery to the air;  $m_{bat}$  is the mass of the energy storage module;  $c_p$  is the average heat capacity of the module. Since the battery temperature changes gradually under a driving cycle and the desired driving torque changes at a much fast pace, in the simulation study, the battery capacity within the prediction horizon is based on the current battery temperature.

The main parameters of vehicle and powertrain system are listed in Table 3. 1.

### 3.2.2 Baseline power-follower control strategy

The baseline control strategy is an extension of the equivalent power-follower control strategy (PFCS) of a series hybrid powertrain [42] to the parallel one. The desired torque (power), battery power output limit, SOC, vehicle speed, and other parameters are used as primary control parameters to turn on or off the engine, see Figure 3. 1. When the engine is off, the battery-EM set is the only power source for the vehicle; and when the engine is on, engine along with the EM set provides the power for the vehicle operation and charges the battery to target SOC level. Note that the control strategy was not optimized for fuel economy since it is not the subject of this study.

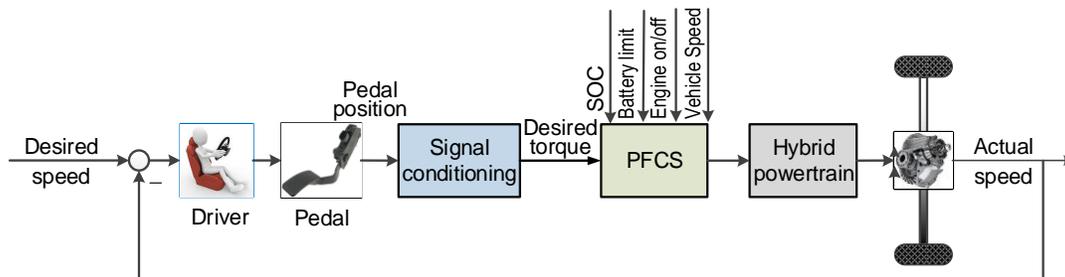


Figure 3. 1 PFCS implementation

#### 3.2.2.1 IC engine control logic

The engine control logic can be described below:

$$S_{eg}(k) = \begin{cases} 1 & \begin{cases} P_r \geq P_{b,lim} \text{ or} \\ SOC \leq SOC_L \text{ or} \\ SOC \geq SOC_U \ \& \ P_{b,lim} > P_r > P_{b,s} \ \& \ S_{eg}(k-1) = 1 \text{ or} \\ SOC_U > SOC > SOC_L \ \& \ P_r < P_{b,lim} \ \& \ S_{eg}(k-1) = 1 \end{cases} \\ 0 & \begin{cases} SOC \geq SOC_U \ \& \ P_r \leq P_{b,s} \ \text{or} \\ SOC > SOC_L \ \& \ P_r < P_{b,lim} \ \& \ S_{eg}(k-1) = 0 \end{cases} \end{cases} \quad (2.7)$$

where  $S_{eg}(k)=1$  means that engine is on;  $S_{eg}(k)=0$  indicates that engine is off;  $S_{eg}(k-1)$  represents the previous step engine state;  $SOC_L$  and  $SOC_U$  are the lower and upper bounds of SOC under normal battery operations, respectively;  $P_r$  is the driver desired power;  $P_{b,s}$  and  $P_{b,lim}$  are the predetermined battery output power thresholds at a given temperature to form hysteresis to avoid frequently turning the engine on or off, where  $P_{b,s} < P_{b,lim}$ .

The control logic is also illustrated in Figure 3. 2, where  $P_{e,max}$  is the maximum engine power output; the gridded portion is the hysteresis area. Note that both minimal engine on and off durations are also part of the engine on-off control strategy.

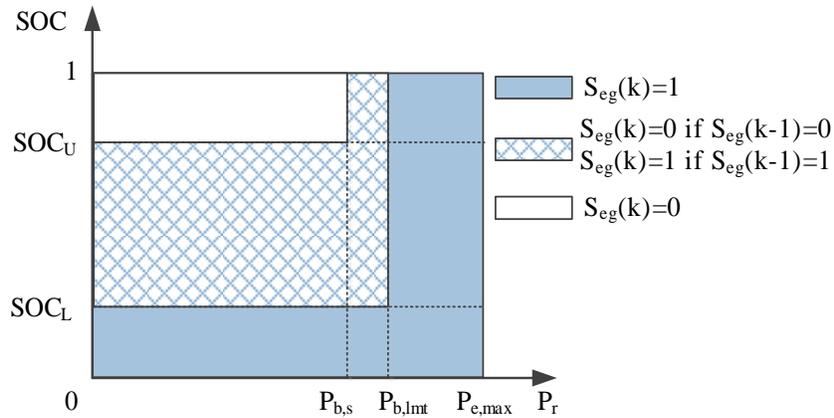


Figure 3. 2 Engine control diagram of PFCS

### 3.2.2.2 Power distribution

The corresponding parallel HEV PFCS power distribution is stated as the following,

$$\begin{cases} P_e = \begin{cases} 0, & S_{eg}(k) = 0 \\ (P_r - P_B) + P_A, & S_{eg}(k) = 1 \end{cases} \\ P_{bat} = \begin{cases} P_r, & S_{eg}(k) = 0 \\ P_B, & S_{eg}(k) = 1 \end{cases} \end{cases} \quad (2.8)$$

where  $P_{bat}$  is the battery power output;  $P_B$  is the EMB power output; and  $P_A$  is the generator power output satisfying the following equation,

$$P_A = P_{ch}(SOC_T - SOC) \quad (2.9)$$

where  $SOC_T$  is the target  $SOC$ ; and  $P_{ch}$  is the preselected battery charging power constant.

During the braking operation, the power charged to the battery is expressed as

$$P_c = \begin{cases} P_A + P_B, & S_{eg}(k) = 1 \\ P_B, & S_{eg}(k) = 0 \end{cases} \quad (2.10)$$

The power distribution can also be explained in Table 3. 2.

Engine state, $S_{eg}(k)$	<b>0</b>	<b>1</b>
Engine power output, $P_e$	0	$(P_r - P_B) + P_A$
Battery power output, $P_{bat}$	$P_r$	$P_B$
Charging power, $P_c$	$P_B$	$P_A + P_B$

Since allowing charging current to exceed the battery charging limit could damage the battery and reduce its useful life, regenerative power needs to be well controlled. The details will not be described here since it is not the subject of this chapter.

### 3.3 Prediction error stochastic analysis under different driving cycles

This subsection studies the characteristics of the prediction errors of the adaptive recursive algorithm. The prediction error can be defined as the difference between the actual and predicted desired torque as shown in the following,

$$\varepsilon(k) = y(k) - \hat{y}(k) \quad (2.11)$$

where  $y(k)$  is the actual desired torque and  $\hat{y}(k) = \tilde{\Theta}_{2n}(k-m) \cdot \varphi_{2n}(k-m)$  is the predicted one at time  $k$ . For a given number of data points  $N$ , the mean value  $\mu$  and variance  $\sigma$  of  $\varepsilon(k)$  can be calculated as follows.

$$\begin{cases} \mu = \frac{1}{N} \sum_{i=0}^{N-1} \varepsilon(i) \\ \sigma^2 = \frac{1}{N} \sum_{i=0}^{N-1} \varepsilon^2(i) \end{cases} \quad (2.12)$$

To study the prediction error distribution, simulations were conducted under the FTP driving cycle. The probability density function (PDF) of the prediction errors is plotted in Figure 3. 3, where  $\chi$  is the prediction error.

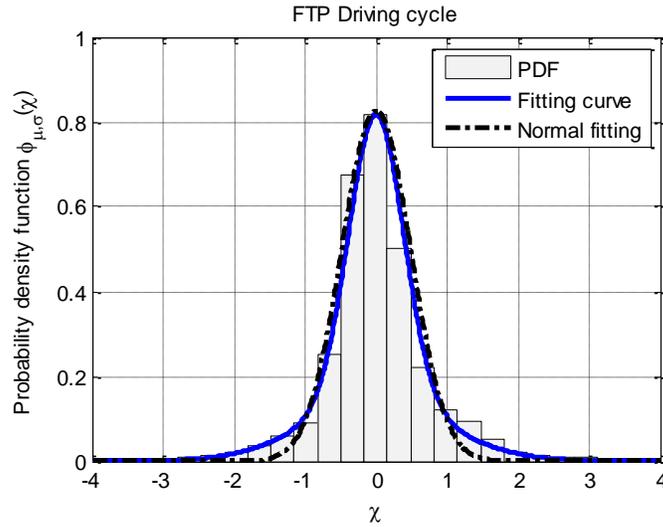


Figure 3. 3 Prediction error distribution

It shows that the mean of prediction errors is very close to zero (see Table 3. 3 for numerical values), and its distribution is also close to normal distribution; see the fitted normal distribution curve (dashed line) with mean ( $\mu = 0$ ) and variance ( $\sigma = 0.4017$ ). The distribution can also be fitted with improved accuracy by an average of three different normal distributions (see the solid line in

the figure) with the PDF expressed below,

$$\varphi(\mu, \sigma) = (\mathbb{N}(\mu, \lambda_1\sigma) + \mathbb{N}(\mu, \lambda_2\sigma) + \mathbb{N}(\mu, \lambda_3\sigma)) / 3 \quad (2.13)$$

where  $\mu$  and  $\sigma$  are the mean and variance of the prediction errors, respectively; and  $\mathbb{N}(\mu, \lambda_1\sigma)$ ,  $\mathbb{N}(\mu, \lambda_2\sigma)$  and  $\mathbb{N}(\mu, \lambda_3\sigma)$  are normal distributions with mean value  $\mu$  and variances  $\lambda_1\sigma$ ,  $\lambda_2\sigma$ , and  $\lambda_3\sigma$ , respectively. The coefficients associated with the three distributions are  $\lambda_1 = 0.86$ ,  $\lambda_2 = 0.92$  and  $\lambda_3 = 2.2$ . This composite normal distribution improves the fitting and reduces the distribution error over the fitted normal distribution.

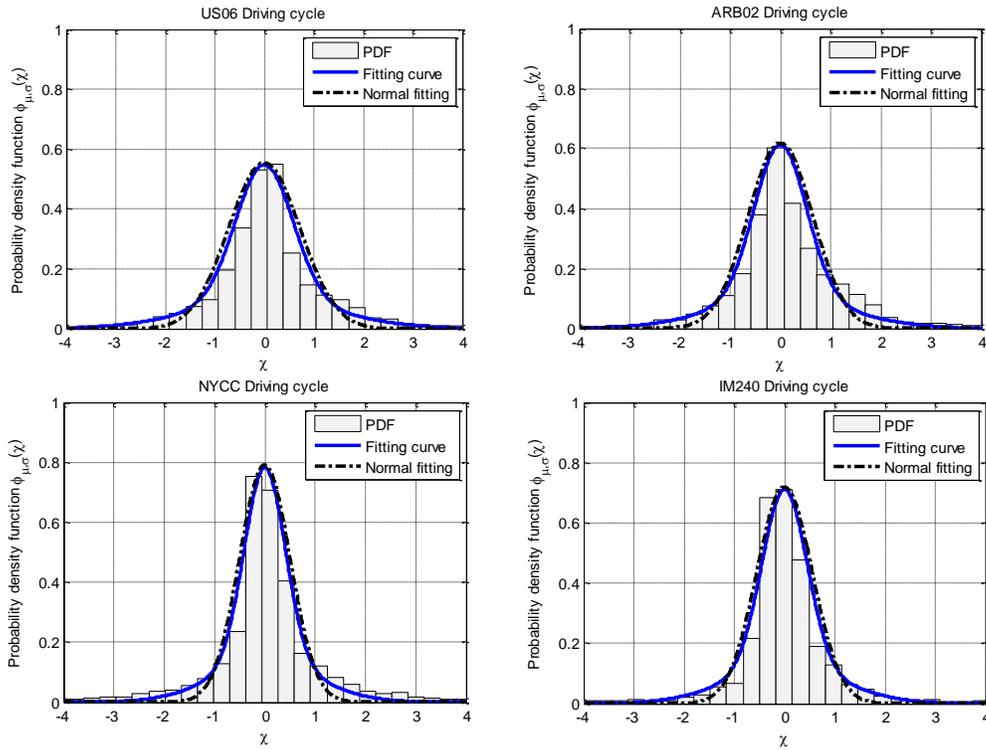


Figure 3. 4 Prediction error distribution under other four typical driving cycles

Simulations were also conducted under other four typical driving cycles, US06, ARB02, NYCC, and IM240. The prediction error PDFs are plotted in Figure 3. 4 and their means and variances are provided in Table 3. 3. It is clear that all the mean values of the prediction errors are extremely close to zero, which indicates that the predicted value represents the statistical mean

value of the actual signals. Simulation results also demonstrated that PDF fitting expression (2.13) can also be used for these four typical driving cycles with the same coefficients  $\lambda_1, \lambda_2$  and  $\lambda_3$ . This is very important since in this case only the error variance is needed to determine the PDF.

Table 3. 3 Prediction error distribution in typical driving cycles

<b>Distribution</b>	<b>US06</b>	<b>ARB02</b>	<b>NYCC</b>	<b>FTP</b>	<b>IM240</b>
$\mu$ (mean)	0.0003	0.0012	0.0009	0.0014	0.0018
$\sigma$ (variance)	0.9494	0.7546	0.5071	0.4017	0.4037

Since the adaptive recursive prediction is the Least-Squares (LS) based estimation, the LS estimation provides zero error mean and the above simulation results confirm that

$$\mu \triangleq \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=0}^{N-1} \varepsilon(i) = E_{\infty}[\varepsilon(k)] = 0 \quad (2.14)$$

where  $E_{\infty}(\cdot)$  is the expectation operator. However, it should be pointed out that as time  $N$  goes to infinity, the computational load to calculate both mean and variance increases. For the purpose of real-time implementation, the following equations were used

$$\begin{cases} \mu(k) = \frac{1}{N} \sum_{i=k-N+1}^k \varepsilon(i) \\ \sigma^2(k) = \frac{1}{N} \sum_{i=k-N+1}^k \varepsilon^2(i) \end{cases} \quad (2.15)$$

where  $N$  is the number of samples of the moving window for the mean and variance calculation. Noticed that  $N$  should be selected properly to make real-time calculation possible while meeting the accuracy requirement.

Assuming that data  $\varepsilon(i)$  ( $i = k - N, k - N + 1, \dots, k - 1$ ) is stored in the memory, the mean  $\mu(k)$  and variance  $\sigma(k)$  can be updated using current and 1-step recursive history data as follows:

$$\begin{cases} \mu(k) = \mu(k-1) - \frac{1}{N} \varepsilon(k-N) + \frac{1}{N} \varepsilon(k) \\ \sigma^2(k) = \sigma^2(k-1) - \frac{1}{N} \varepsilon^2(k-N) + \frac{1}{N} \varepsilon^2(k) \end{cases} \quad (2.16)$$

Therefore, for the fixed point calculation, selecting the  $N$  in the form of  $2^l$  ( $l > 0$ , an integer) has the advantage of reducing calculation load since the division can be completed by arithmetic shift of a binary number.

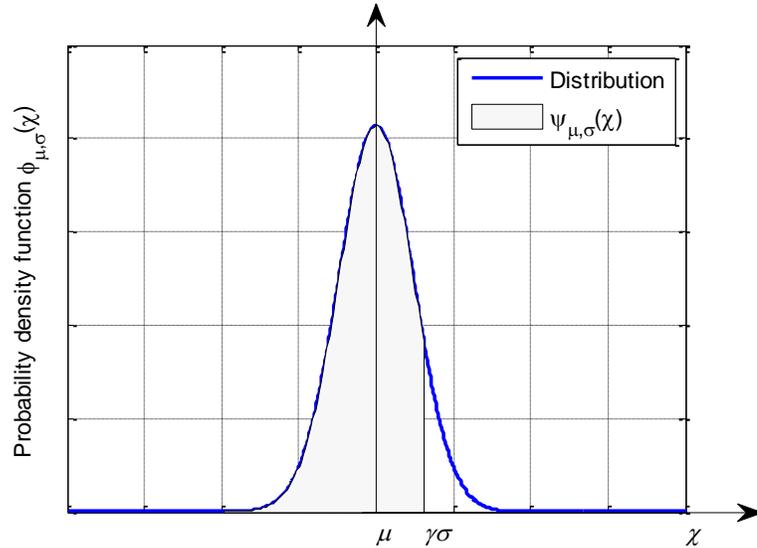


Figure 3. 5 Probability density function

Since the prediction error is zero mean with known PDF of the averaged three Gaussian distributions, there are potentials to further reduce the battery over-discharging based upon the PDF. As shown in Figure 3. 5, the bell curve is the prediction error PDF. Note that since the mean of the prediction error is zero, the predicted value is the mean value of the signal. If the predicted power (torque) value is directly used as the threshold to start the engine (e.g. the PBM case), there is a 50% chance that the engine will get started proactively to cover the power demand. This is because there is a 50% chance that the actual desired power (torque) is higher than the mean one (see Figure 3. 5), where the cumulative probability (or the percentage of confidence level) can be expressed as the integral of the known PDF.

$$\psi(\mu, \sigma) = \frac{1}{3\sqrt{2\pi}} \int_{-\infty}^{\mu+\gamma\sigma} \left( \frac{e^{-(\chi-\mu)^2/12(\lambda_1\sigma)^2}}{\lambda_1\sigma} + \frac{e^{-(\chi-\mu)^2/12(\lambda_2\sigma)^2}}{\lambda_2\sigma} + \frac{e^{-(\chi-\mu)^2/12(\lambda_3\sigma)^2}}{\lambda_3\sigma} \right) d\chi \quad (2.17)$$

where  $\gamma$  is the given variance multiplier.

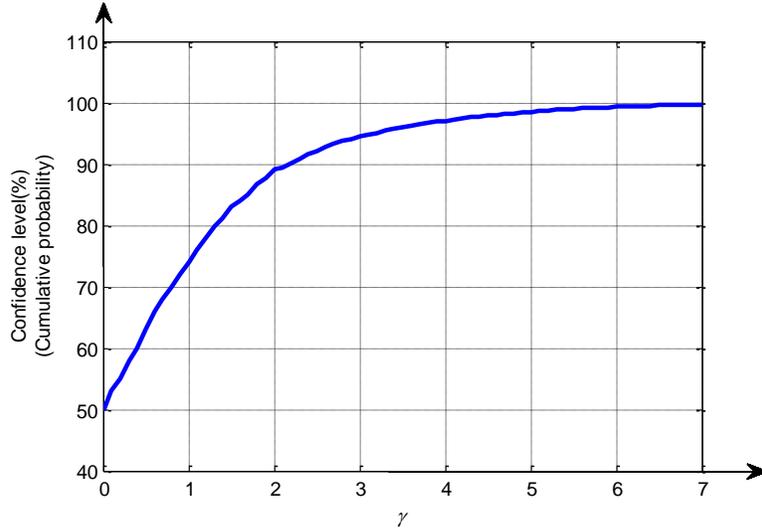


Figure 3. 6 Confidence level profile

Figure 3. 5 shows the PDF of prediction errors for the FTP driving cycle, where the shaded area represents the cumulative probability (or percentage of confidence level), and it is proportional to the multiplier ( $\gamma$ ) of prediction error variance. The bigger the  $\gamma$ , the larger the cumulative probability (shaded area). The relationship between  $\gamma$  and the cumulative probability is plotted in Figure 3. 6.

It is clear that as  $\gamma$  goes up, the percentage of cumulative probability is increased from 50% to 100% as listed in Table 3. 4.

$\gamma$	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5
<b>Confidence level</b>	50.0	63.1	74.4	82.8	88.6	92.2	94.5	95.9
	%	%	%	%	%	%	%	%
$\gamma$	4.0	4.5	5.0	5.5	6.0	6.5	7.0	
<b>Confidence level</b>	97.1	97.9	98.5	98.9	99.3	99.5	99.7	
	%	%	%	%	%	%	%	

Given a percentage of confidence level along with the predicted powertrain power, a smallest upper bound of future desired power level can be determined that will be used to proactively turn on the engine, and hence the battery over-discharged duration can be further reduced, which lead

to the proposed SPBM strategy. Note that the PBM strategy is a special case of SPBM with 50% of confidence level ( $\gamma = 0$ ).

### 3.4 Battery boundary management strategy

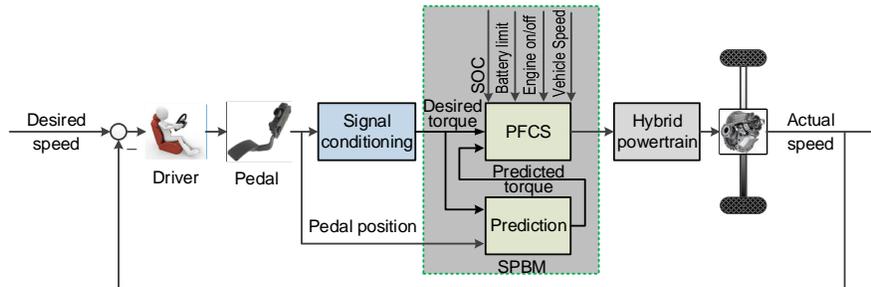


Figure 3. 7 SPBM implementation

One of the key strategies for a hybrid supervisory controller is to distribute power among engine, generator, and motor based upon driver demand and current powertrain status. This section discusses the proposed SPBM strategy. The main advantage of the aforementioned PFCS is its simplicity. Whereas, similar to other exist hybrid powertrain control systems, the battery is easily to be over-discharged under the PFCS, leading to reduced battery life and degraded overall HEV performance. This could happen especially under the transient operations at low ambient temperature. Since the PFCS uses desired power to determine if the IC engine needs to be turned on to ensure that enough power is provided to the powertrain. However, it could take more than half second for an IC engine to crank-start and make power available. During this period, the battery has to be over-discharged to provide desired power to the powertrain until the engine power is available. To reduce the overall battery over-discharged power, an SPBM strategy is proposed based on the predicted desired torque (power) and its prediction error variance to proactively turn the engine on and make its power available, when the predicted future power with certain confidence level exceeds current battery output power limit. As a result, the battery over-

discharging duration can be reduced significantly. For the control strategy implementation, both current desired torque demand and predicted one with prediction error variance are used as inputs to hybrid powertrain control system, see Figure 3. 7.

The core of the proposed control strategy consists of two portions. One is using the adaptive recursive prediction algorithm [45] to estimate future hybrid powertrain desired torque (power) based upon the current and past desired torque information, see Figure 3. 7. Both the raw and conditioned acceleration pedal positions (desired torques) are used as inputs for the desired torque prediction. The other is the prediction error variance calculation based on the current and past prediction error; see (2.16). Note that both desired torque prediction and its prediction error variance are calculated in real-time. For an appropriate pre-selected confidence level, the smallest upper bound of future desired power can be found in terms of  $\hat{y}_p(k) + \gamma\sigma$  based on the prediction error variance, where  $\hat{y}_p(k)$  is the predicted torque at time  $k$ . It will be used to determine whether the engine needs to be proactively turned on; see the discussion at the end of section 3.3.

As demonstrated in section III, the predicted value is the mean of the future signal, and the PBM strategy is a special case of SPBM one with 50% of confidence level (cumulated probability). The higher the confidence level, the larger the smallest upper bound of the future desired power, and consequently, the earlier the engine will get crank-started, which further reduces the battery over-discharging duration. On the other hand, large confidence level could also lead to unnecessary engine-start. Note that from Figure 3. 6 there is a one-to-one mapping between the percentage of confidence level and the variance multiplier  $\gamma$ . Therefore, for a given percentage of confidence level (CL), a smallest upper bound of the driving power,  $y_{CL}(k)$  can be found; see Figure 3. 5, where

$$y_{CL}(k) = \hat{y}_p(k) + \gamma\sigma \quad (2.18)$$

Note that the multiplier  $\gamma$  is a positive number. The CL needs to be selected properly since high CL could lead to unnecessary engine-start and the selection needs to balance the reduction of battery over-discharging duration and unnecessary engine-start. The details of how to select the confidence level will be addressed in subsection 3.5.4.

### 3.4.1 Engine control logic

As aforementioned, it could take more than half second for the engine to crank-start and make the power available to the powertrain. Based upon the raw and conditioned pedal positions, the prediction algorithm is able to forecast the future desired torque half second ahead. The proposed SPBM strategy uses the predicted desired torque and its prediction error variance with the given confidence level to proactively turn on the engine half second in advance to make the engine power available, which could reduce the battery over-discharging duration greatly and thereby to extend the useful life of battery.

Based on  $y_p$ , prediction error variance  $\sigma$ , and the given confidence level  $\gamma$ , the SPBM engine control logic, inherited from PFCS, is described as the following:

$$S_{eg}(k) = \begin{cases} 1 & \begin{cases} P_d \geq P_{b,lim} \text{ or} \\ SOC \leq SOC_L \text{ or} \\ SOC \geq SOC_U \ \& \ P_{b,lim} > P_r > P_{b,s} \ \& \ S_{eg}(k-1) = 1 \text{ or} \\ SOC_U > SOC > SOC_L \ \& \ P_r < P_{b,lim} \ \& \ S_{eg}(k-1) = 1 \end{cases} \\ 0 & \begin{cases} SOC \geq SOC_U \ \& \ P_r \leq P_{b,s} \text{ or} \\ SOC > SOC_L \ \& \ P_r < P_{b,lim} \ \& \ S_{eg}(k-1) = 0 \end{cases} \end{cases} \quad (2.19)$$

where  $SOC_L$ ,  $SOC_U$ ,  $S_{eg}(k-1)$ ,  $P_{b,lim}$  and  $P_{b,s}$  are defined in (2.7); and  $P_d$  is the predicted desired power which can be expressed as follows.

$$P_d = (y_p + \gamma\sigma) \frac{v_{veh} f_d}{r_w} \quad (2.20)$$

where the predicted desired torque  $y_p$  can be obtained using the algorithm mentioned in section 3.3; prediction error variance  $\sigma$  is calculated on-line;  $v_{veh}$  is the current vehicle speed;  $r_w$  is the tire radius; and  $f_d$  is the final drive ratio from planetary gear to wheel. The control logic can also be illustrated in Figure 3. 8, where the  $x$ -axis is  $P_r$  and  $P_d$ , which is different from PFCS.

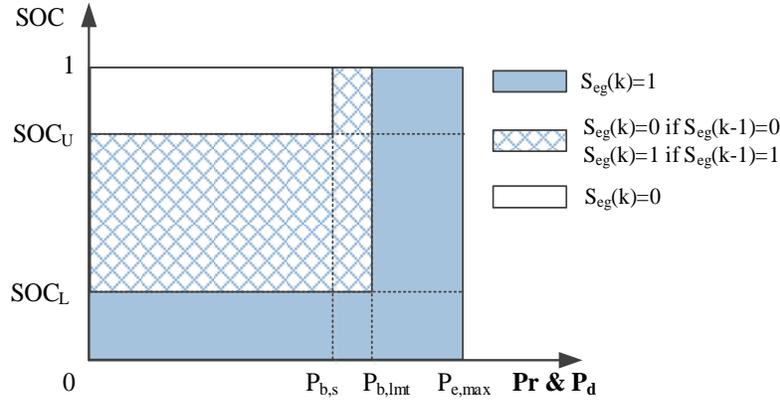


Figure 3. 8 SPBM engine control logic diagram

There are three engine operation regions in Figure 3. 8. Region 1 is engine-on region (the dark area), where predicted power is equal to or greater than battery power limit  $P_{b,lim}$  or SOC is equal to or below the predefined SOC lower bound. This region can be extended to the adjacent gridded hysteresis region, Region 2 (the gridded area). In this region the engine state remains unchanged to prevent the engine from being frequently turned on or off, where SOC is greater than its lower bound but below its upper bound and predicted power is smaller than the battery output power limit  $P_{b,lim}$  and greater than battery output power thresholds  $P_{b,s}$ . Region 3 (white area) is the engine-off region, where SOC is equal to or greater than predefined SOC upper bound and desired power is smaller than battery output power thresholds  $P_{b,s}$ . This region can be extended to the adjacent gridded hysteresis region as well.

Note that the desired power (torque) is used for engine off control in (2.19) because there is

almost no engine off delay, i.e., it can be shut down immediately. Also a minimum engine-on and off time strategies are also used to avoid engine being frequently turned on and off. But the minimum engine-off time needs to be carefully chosen since long minimum engine-off time [42] could prevent the engine to be turned on immediately as needed while the minimal-off time is still active. This could limit reducing the battery over-discharging duration effectively. In this research, the minimum engine-off time is only used to ensure that the engine is completely turned off before it is turned on.

For the proposed SPBM strategy power distribution, equations (2.8), (2.9) and (2.10) hold.

### 3.5 Simulation study and validation

To evaluate the performance of the baseline PFCS, PBM strategy, and the proposed SPBM strategy, the developed series-parallel HEV model was simulated under five typical driving cycles. They are US06, ARB02, NYCC, IM240 and FTP, where the US06 and ARB02 driving cycles are used to study the influence of aggressive driving behaviors with rapid speed variations after vehicle start-up; the NYCC cycle is used for studying rich stop-and-go city driving conditions; IM240 is used for highway driving test, and FTP is for the mixed city and highway driving studies. The key simulation parameters are listed in Table 3. 5.

Table 3. 5 Key simulation parameters

Item	Parameter	Value
Engine	Start response time, $\tau_s$	0.5 s
	Time constant, $\tau_e$	0.015~0.04s
	Minimum engine on time	15s
Battery	<i>SOC</i> upper bound, $SOC_U$	0.75
	<i>SOC</i> lower bound, $SOC_L$	0.45
	Engine start <i>SOC</i>	0.5
	<i>SOC</i> target, $SOC_T$	0.6
Prediction	Prediction horizon	0.5s
	Step size	0.01s

### 3.5.1 Battery performance at low temperatures

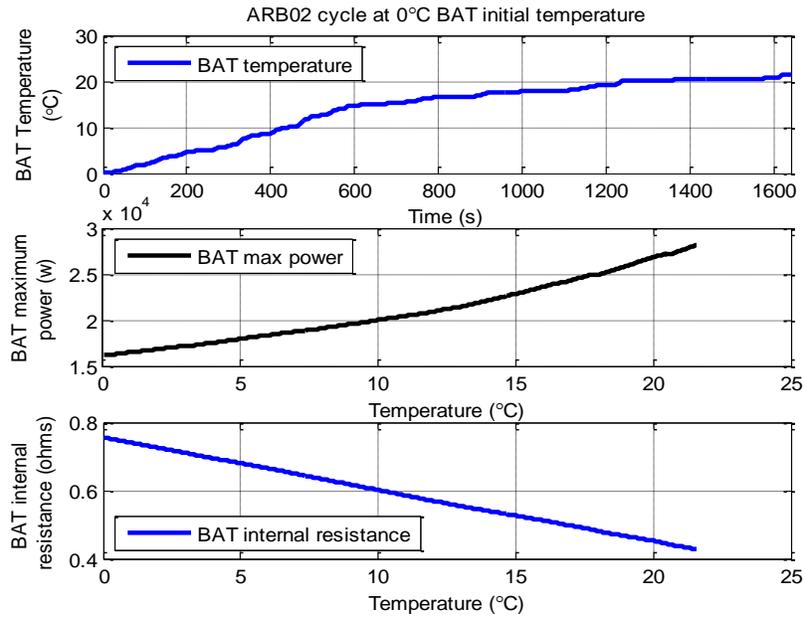


Figure 3. 9 Battery temperature, maximum power and internal resistance trace

Figure 3. 9 shows the simulation results of battery temperature, maximum power, and internal resistance variations under ARB02 driving cycle with 0°C initial battery temperature. The battery temperature gradually goes up to around 22°C at the end of ARB02 driving cycle, and with the increment of temperature, the battery maximum power is increased correspondingly while the battery internal resistance decreases.

Figure 3. 10 provides simulation results of the battery performance under ARB02 driving cycle at different initial battery temperatures, where positive battery power output indicates battery discharging and negative one means charging. The battery over-discharging/charging happens when the battery power output exceeds its limit. Since the battery maximum power is decreased as the internal resistance increases at low temperatures and so does the power output limit, the battery is easier to be over-discharged compared with at higher temperatures. For example at -20°C, the battery is more frequently over-discharged than at 20°C.

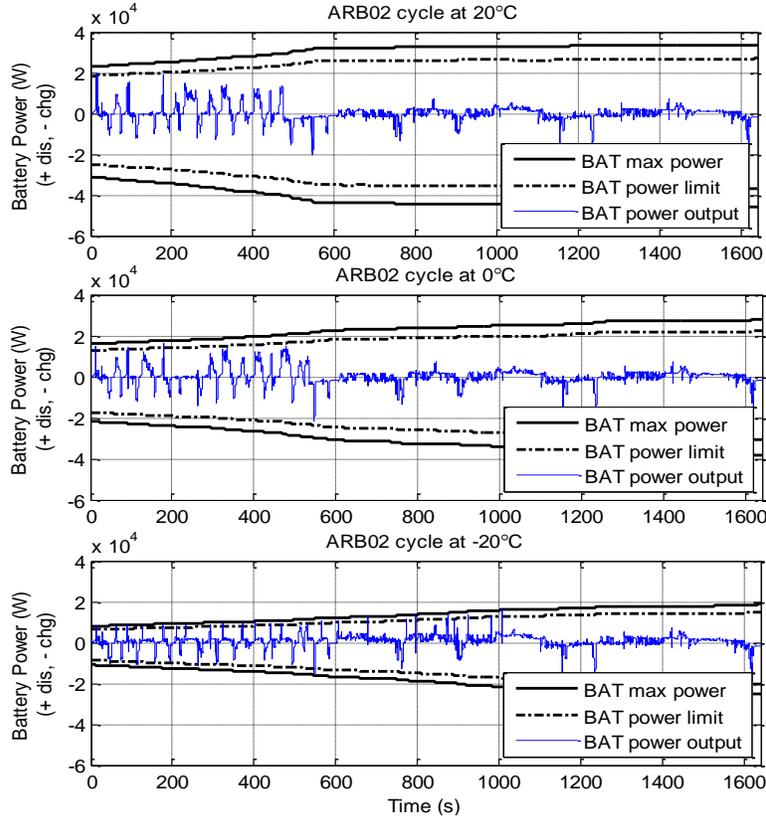


Figure 3. 10 Battery performance at different temperatures

### 3.5.2 PFCS simulation

Figure 3. 11 shows battery power output and engine on and off status under the US06 driving cycle at close to 0°C. When the vehicle starts up, the IC engine is off; and once the battery output power reaches its output power limit at around 90<sup>th</sup> second, the engine is turned on. Due to the minimal engine-on time requirement and SOC discharging/charging hysteresis rules (between 0.45 and 0.75), the engine is kept on until the engine off conditions are satisfied around the 520<sup>th</sup> seconds.

From equation (2.8), it is clear that when the engine is off ( $S_{eg} = 0$ ), the battery-EMB set provides all the driving power to the powertrain; and while the engine is on ( $S_{eg} = 1$ ), the engine provides majority of the driving power to the drivetrain except under the transient operations. In

Figure 3. 12, there is a certain EMB torque output when the engine is on at around the 90<sup>th</sup>, 130<sup>th</sup>, and 500<sup>th</sup> seconds, which is due to the relatively slow response of the engine that requires the EMB to provide additional torque to the drivetrain.

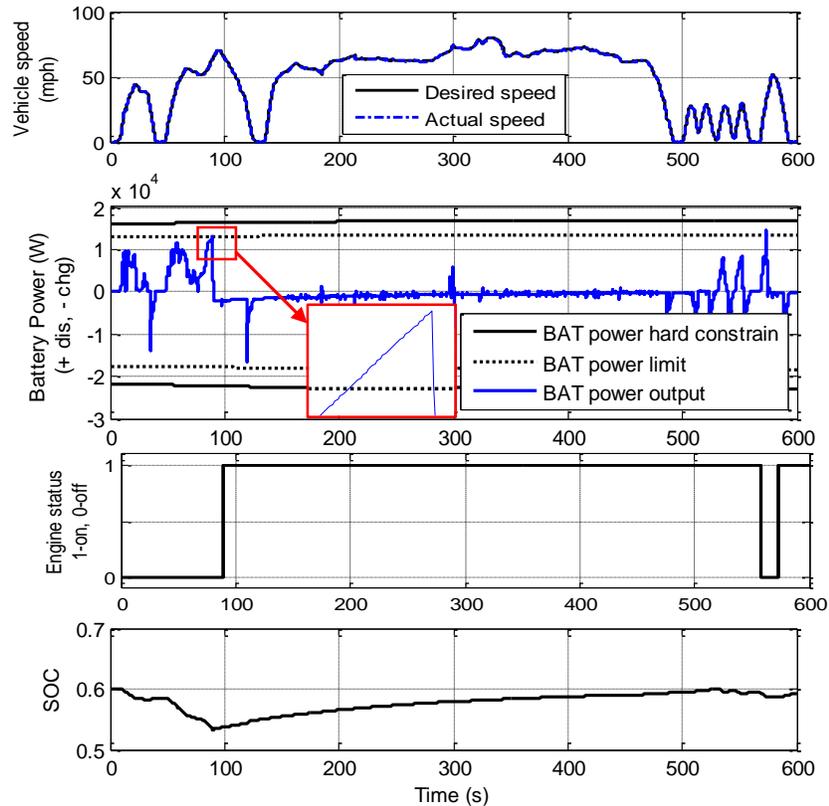


Figure 3. 11 Battery power output under US06 cycle at 0°C

When the vehicle is under deceleration, the EMB also provides regenerative torque (negative EMB torque in Figure 3. 12) for braking and generating electricity to charge the battery.

It can also be observed from Figure 3. 12 that there is a short time delay between the engine desired torque output and the actual one. In order to maintain the powertrain power output at the desired level, the battery has to provide power at a level that is beyond its normal operational limit, see the enlarged area in Figure 3. 11. In this case, the battery power output exceeds its output limits before the engine power becomes available. As stated previously, a short period of over-discharging is inevitable and might not damage the battery, but frequent or persistence over-

discharging, especially at low battery temperature (it can be observed in Figure 3. 9 at  $-30^{\circ}\text{C}$  case), could permanent damage the battery, leading to reduced battery life. If the total battery over-discharging power is reduced, then the battery can be protected and its useful life can be extended, which leads to the PBM strategy.

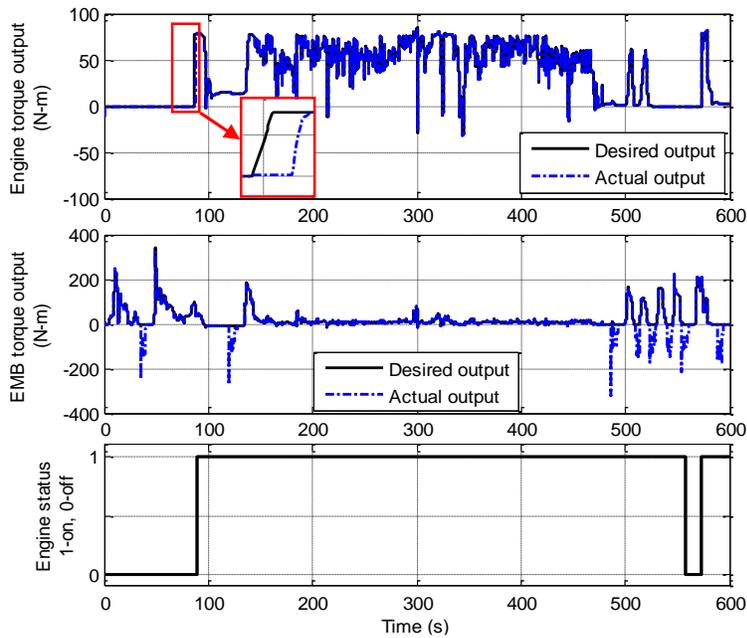


Figure 3. 12 Engine and EMB torque outputs

### 3.5.3 PBM simulation

Besides the input signals used for baseline control strategy, the predicted desired torque (power) is directly used as an additional input for the PBM strategy to turn engine on. In order to validate the PBM strategy and evaluate its performance, simulations were conducted under the same condition as that for PFCS. As shown in the enlarged area in the top graph of Figure 3. 13, the battery over-discharging duration is reduced to a certain degree, where the triangle area above the battery power limit line is reduced to a small dotted-line enclosed triangle area, detailed (enlarged) plot see Figure 3. 14.

Also, it can be seen from SOC profile in the bottom plot of Figure 3. 13 that once the engine

is turned on, the battery is being charged. In order to quantify how much battery energy is over-discharged, the following integration is used as a measure,

$$E(t) = \begin{cases} \int_0^t (P_b - P_{b,lim}) d\tau, & P_b > P_{b,lim} \\ 0, & P_b \leq P_{b,lim} \end{cases} \quad (2.21)$$

The total reduction of over-discharged energy is defined as

$$E_r(t) = \left( 1 - \frac{E_{PBM}(t)}{E_{PFCS}(t)} \right) \quad (2.22)$$

where  $E_{PBM}(t)$  and  $E_{PFCS}(t)$  represent total over-discharged energy integration under PFCS and PBM strategy, respectively.

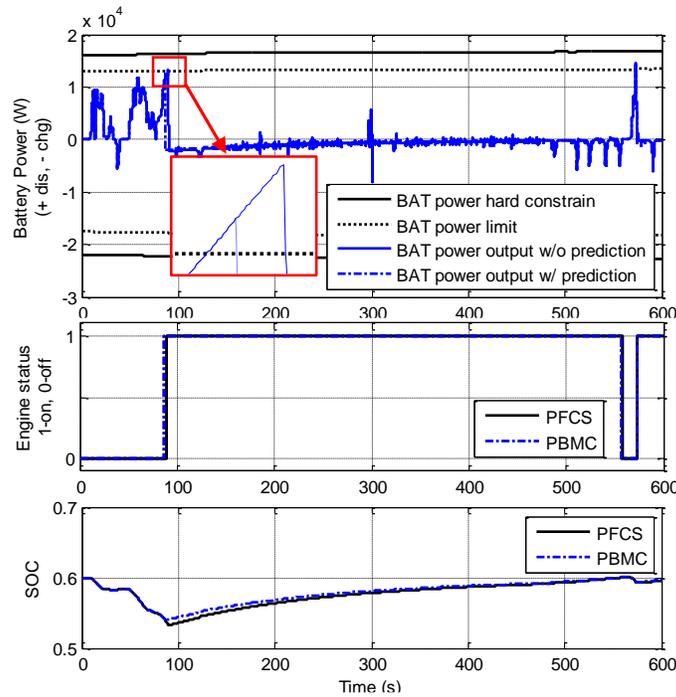


Figure 3. 13 Reduced battery over-discharging duration

The over-discharged energy reduction can be illustrated in Figure 3. 14, where the gridded area represents over-discharged energy reduction for the simulation case shown in Figure 3. 13. It is obvious that after the PBM strategy is applied, the total over-discharged energy is reduced to a small gray area. Specific simulation data is listed in Table 3. 6.

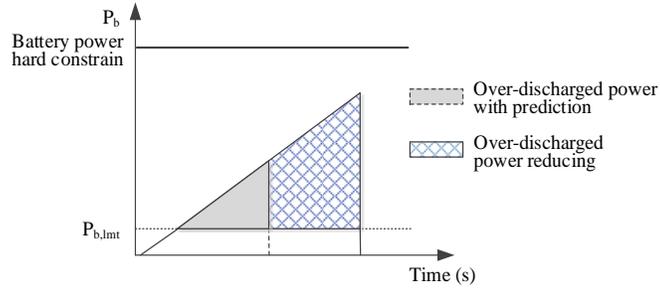


Figure 3. 14 Over-discharged power reducing

Simulation results under US06 driving cycle indicates that PBM strategy provides a fairly good improvement of reducing the battery over-discharging duration. To check the effectiveness of the PBM strategy, four other typical driving cycles, ARB02, NYCC, IM240, and FTP are selected for simulation validation, and the results are summarized in Table 3. 6.

Table 3. 6 Over-discharged energy reducing with PBM

Temperature	ARB02	US06	NYCC	FTP	IM240
20°C	68%	-	-	-	-
10°C	60%	-	-	-	-
0°C	62%	75%	-	-	-
-10°C	51%	85%	39%	42%	-
-20°C	57%	80%	44%	56%	38%
-30°C	65%	77%	48%	52%	26%
Average	68%		47%		32%

It can be observed from Table 3. 6 that the PBM strategy performs quite well under five typical driving cycles especially at low temperature, and is very effective when vehicle is operated under aggressive driving cycles with high vehicle speed variations, i.e. US06 and ARB02, and the reduction of the over-discharged energy is more than 65%. Also, under city or mixed city and highway driving cycle (e.g. FTP), the total over-discharged battery energy is reduced more than 45%. Even under IM240 highway driving cycle, the reduction is still more than 30%. Notice that, “-” in Table 3. 6 indicates no battery over-discharging occurred.

### 3.5.4 SPBM simulation

As mentioned previously, given a percentage of confidence level for the predicted powertrain power, a smallest upper bound of future desired power level can be found and to be used to proactively turn on the engine, and hence the battery over-discharged duration can be further reduced. Note that there exists a one-to-one mapping (Figure 3. 6) between the percentage of confidence level and the variance multiplier  $\gamma$ , and for a given confidence level the corresponding  $\gamma$  can be found. However, large  $\gamma$  (high confidence level) could lead to unnecessary engine-start, therefore, the percentage of confidence level needs to be selected properly to balance the reduction of battery over-discharge and unnecessary engine-start.

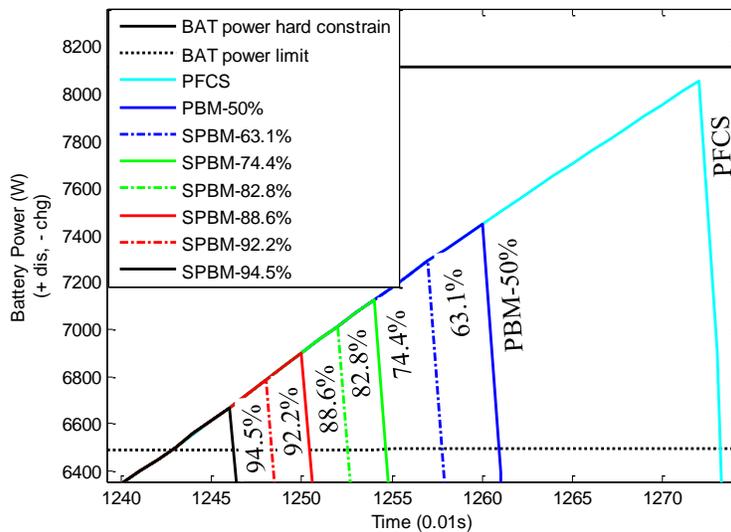


Figure 3. 15 Over-discharging duration with confidence level

Figure 3. 15 shows that under FTP driving cycle at temperature of  $-10^{\circ}\text{C}$ , the battery over-discharging duration reduction with different confidence level from 50% to 94.5%. Firstly, the battery over-discharging duration is greatly reduced using PBM over the PFCS strategy, which was presented in previous subsection. Secondly, the battery over-discharging duration is further reduced using the SPBM strategy under the confidence levels greater than 50% over the PBM strategy, and it also shows that the higher the confidence level, the shorter the battery over-

discharging duration. However, the reduction of the over-discharging energy (area) becomes smaller as the percentage of confidence level is close to 100%.

On the other hand, as the over-discharging duration is gradually reduced with the increment of confidence level, the number of engine-start increases significantly at the same time. Figure 3. 16 shows that under FTP driving cycle the integration of the battery over-discharged energy is reduced from around 360 to 55 ( $\text{kWh} \cdot 3.6 \times 10^{-6}$ ) as confidence level increases from 50% to 94.5% ( $\gamma$  increases from 0 to 3), whereas, the number of engine-start is increased from 6 to 14. Note that when confidence level is greater than 88.6% ( $\gamma > 2$ ), the battery over-discharged energy reduction is quite limited, for example, as confidence level increases from 88.6% to 94.5% ( $\gamma$  increases from 2 to 3), the integration of the over-discharging energy reduction is only 5 from 60 to 55 ( $\text{kWh} \cdot 3.6 \times 10^{-6}$ ), see Figure 3. 16. There is a trade-off relationship between over-discharged energy reduction and number of engine-start (increased fuel consumption). Considering the balance between reduced battery over-discharging duration and the number of unnecessary engine-start, the confidence level is chosen as 74.4% ( $\gamma = 1$ ).

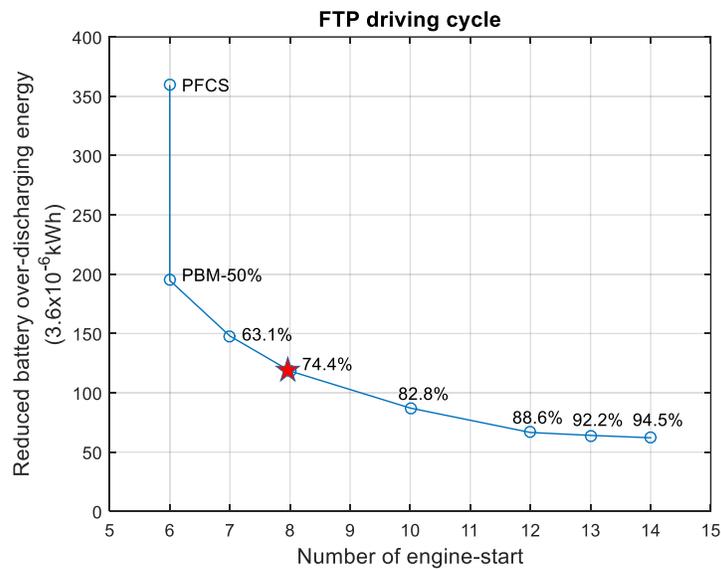


Figure 3. 16 Reduced over-discharged energy with number of engine-start

The goal of studying the trade-off between the reduced over-discharging duration and the increased number of engine-start is to balance the fuel economy and reduced over-discharging energy. It can be seen from Figure 3. 17 that as confidence level goes up from 50% to 94.5%, the fuel consumption increases from 0.355 gallon to 0.371 gallon under FTP driving cycle due to the increment of number of engine-starts, that is, the increment of confidence level can result in a fuel economy penalty. For instance, change of confidence level from 50% to 74% leads to 1.9% fuel economy penalty.

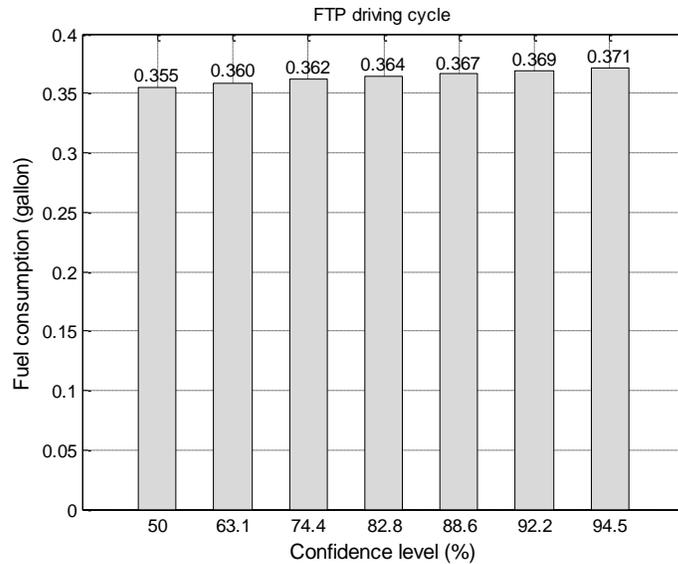


Figure 3. 17 Fuel consumption under different confidence levels

Simulations are also conducted under four other typical driving cycles, US06, ARB02, NYCC and IM240 to study the performance of the proposed SPBM strategy at 74.4% confidence level, and the simulation results are listed in Table 3. 7 using the following error definition similar to equation (2.22).

$$E_{r\_SPBM}(t) = \left( 1 - \frac{E_{SPBM}(t)}{E_{PFCS}(t)} \right) \quad (2.23)$$

Table 3. 7 Over-discharged energy reducing with SPBM

Temperature	ARB02	US06	NYCC	FTP	IM240
20°C	81%	-	-	-	-
10°C	78%	-	-	-	-
0°C	79%	89%	-	-	-
-10°C	70%	92%	62%	66%	-
-20°C	75%	90%	80%	62%	41%
-30°C	79%	87%	71%	62%	32%
Average	82%		67%		37%

It can be observed from Table 3. 7 that the proposed SPBM strategy exhibits better performance over the PBM strategy in Table 3. 6. Under US06 and ARB02 aggressive driving cycles, the reduction of the battery over-discharging energy is increased from 69% to 83%; under FTP and NYCC city or mixed city and highway driving cycles, the improvement is from 47% to 67%; and under IM240 highway driving cycle, the improvement is from 32% to 37%, respectively.

### 3.6 Conclusions

In order to extend the battery life, a hybrid powertrain stochastic predictive boundary management (SPBM) strategy is proposed in this chapter to reduce the battery over-discharging duration. Simulation studies are conducted under five typical driving cycles at different initial battery temperatures to compare the performance of proposed SPBM strategy with two other control strategies: the baseline power-follower control strategy (PFCS), and predictive boundary management (PBM) strategy. Simulation results show that the PFCS cannot effectively prevent the battery from over-discharging that will lead to the reduced battery life and degraded HEV performance; the PBM strategy reduces the battery over-discharging duration at certain degree, however, the proposed SPBM strategy is capable of reducing the battery over-discharging duration significantly over both PFCS and PBM strategy. The reduction of the average over-discharging energy is 82% under the aggressive US06 and ARB02 driving cycles, 67% under the NYCC and

FTP driving cycles, and 37% under the IM240 highway driving cycle. Most importantly, the adaptive prediction and its error variance can be calculated in real-time with very low computational load, which makes the proposed strategy feasible for practical applications.

## CHAPTER 4: MODEL PREDICTIVE CONTROL BASED ON LINEAR QUADRATIC TRACKING

### 4.1 Introduction

Compared with traditional vehicles powered by a single power source, the internal combustion (IC) engine, the hybrid electric vehicles (HEVs) propelled by both engine and electric motor(s) are able to improve fuel economy with reduced emissions by utilizing regenerative braking power, efficient power distribution and management [72]-[73]. Typical HEVs (series, parallel, or power-split configuration) include IC engine, battery, and motor-generator subsystems. Due to the high degrees of freedom introduced by multiple power sources, HEV powertrains require sophisticated supervisory control strategies to manage multiple power sources for the best fuel economy over traditional powertrains. Improvement in fuel economy and emissions of HEVs strongly depends on their supervisory control strategies.

Many HEV supervisory control strategies have been proposed and developed in the past. Equivalent fuel consumption control is one of the popular strategies studied in literature [42], [74]-[75], where the cost function is defined as the instant sum of actual fuel and weighted electric energy (equivalent fuel). However, such a cost function is heavily dependent on the priori knowledge of the driving conditions or scheduled driving cycles, and relies only on the information at current instant for one-step ahead control without considering future torque demand.

Reference [3] provides a model predictive control (MPC) strategy to optimally schedule the torque split ratio of a parallel hybrid vehicle. The equivalent fuel consumption cost function is minimized over a prediction horizon, and the prediction is based on the vehicle velocity estimation provided by telemetry. References [76] and [77] address an MPC scheme for a power-split HEV to manage the battery with or without the ultra-capacitor package, whereas the prediction is based on the assumption that torque demand decays exponentially. Literature [78] uses an MPC scheme to calculate the optimal control inputs for an ecological driving HEV to improve fuel economy, and the MPC scheme depends greatly on the traffic information of the intelligent transportation system (ITS). Reference [79] reports an MPC strategy for a torque-split HEV, where the engine start-stop transient characteristic is considered and the cost includes fuel consumption, equivalent battery energy, and engine key-on penalty. However, the control assumes that the vehicle future powertrain torque demands and driving cycle information are available. Reference [80] presents an energy management strategy for the vehicular electric power system of a series HEV to reduce the fuel consumption and emissions, where the optimization is based on dynamic programming (DP) and the control scheme is for an on-off electric generator, and [81] proposes a Markov chain based stochastic optimal control for plug-in HEV power management.

As a summary, most of the existing work has the following disadvantages. Firstly, the cost function is designed for the current instant. Secondly, the future powertrain power demand is predicted based on the assumption that the traffic/or driving cycle information is available in advance or based on the empirical exponential equation, where the driver behavior is ignored. It is well known that driver behavior is one of the important factors affecting the fuel economy. Finally, the optimal control strategy does not guarantee that the battery operates within its operational boundary and its state of charge (SOC) maintains at the desired level.

This chapter develops an MPC scheme based on linear quadratic tracking (LQT) control to distribute power of a power-split HEV to track the driver torque demand utilizing the early work [45], [82] of the desired-torque prediction. For each step, the LQT controller minimizes the cost function (a combination of the tracking criterion and equivalent fuel consumption) over the prediction horizon and at the same time keeps the battery at its desired SOC. The LQT control law is updated at each step based on the new prediction horizon and powertrain states. In the chapter, a forward control-oriented hybrid vehicle model is developed in Simulink for designing and validating the proposed MPC-LQT strategy, and the control-oriented system model is linearized and discretized at each step and used for calculating the LQT control law. Four main steps are included in the MPC-LQT control scheme at each sample point: 1) linearizing and discretizing the system model at current operational condition (this can be done offline); 2) calculating the optimal LQT solution over the prediction horizon; 3) using the first step of LQT control scheme as the current control output; and 4) moving the entire prediction horizon to the next operational step.

The effectiveness of the corresponding control strategy is studied in simulations under four typical driving cycles. Compared with baseline power-follower strategy, the proposed strategy improves the fuel economy significantly while maintaining the battery SOC at the desired level. 9.9% fuel economy improvement is achieved under the FTP cycle, 7.3% under IM240, 3.6% under ARB02, and 3.8% under US06.

Note that solving the LQT problem at each step requires to solve the difference Riccati equation (DRE) backwards that leads to very high computational load. Two open questions are if it is feasible to use an approximated Riccati solution to replace the exact DRE solution and what is the associated penalty to fuel economy. Two possible approximated solutions are investigated: algebraic Riccati equation (ARE) and iterative ARE solutions; see details in section 4.4. The ARE

and iterative ARE solutions have a potential of reducing the computational load by 8.3 and 50 times, respectively, comparing with the DRE solution. Simulation results show that the iterative ARE solution provides better approximation in terms of fuel economy over the ARE one. For instance, the fuel economy improvement is 9.5% for iterative solution and 9.3% for steady-state under the FTP cycle. This indicates that iterative (ARE) solution balances the computational load and fuel economy improvement for the MPC-LQT strategy and makes it possible for real-time implementation.

The main contribution of this chapter is a real-time feasible MPC-LQT HEV supervisory control strategy that tracks the predicted driver torque demand with optimized equivalent fuel consumption over a given horizon. At the same time, it guarantees that SOC remains at the target level and the battery operates within its operational boundary.

The rest of the chapter is organized as follows. In section 4.2, a simplified HEV system model is presented. Section 4.3 describes the model predictive control scheme and provides the optimal solution using LQT control. Section 4.4 investigates the feasibility of approximating the DRE solution for real-time implementation and simulation results are provided in section 4.5. The last section adds some conclusions.

## **4.2 Hybrid electric vehicle system model**

In this chapter a forward hybrid vehicle model, shown in Figure 2. 6, is developed and used for control strategy development and validation.

The hybrid vehicle model includes driver, hybrid powertrain, and vehicle dynamics subsystem models. For the driver model, it consists of two PID controllers using the vehicle speed error as input and generates acceleration and brake pedal position control signals. The resulting signals are conditioned and converted into the desired powertrain driving and braking torques for the

powertrain supervisory controller to generate control commands for hybrid powertrain subsystems.

For the hybrid powertrain subsystem model, it includes an IC engine, two electric motors (EMs), a battery, and a planetary gear set (see Figure 2. 6), where the IC engine is connected to the carrier gear (C), and the electric motors A and B (denoted by EMA and EMB, respectively) are connected to the sun gear (S) and ring gear (R) of the planetary gear assembly, respectively. In the HEV model, both EMs can be operated as a generator but only EMB serves as a traction motor.

#### 4.2.1 Power transmission and vehicle dynamics

Figure 4. 1 shows the free-body diagram of the powertrain and vehicle dynamic systems. The power generated by the IC engine is split into two paths: mechanical and electrical. The mechanical path includes the IC engine power transmitted from carrier gear directly to the ring gear that is connected to the vehicle main driven shaft. The electrical path converts the rest of the IC engine power to electricity by EMA for charging the battery. EMB, connected to the ring gear, drive the vehicle directly and to regenerate power during braking.

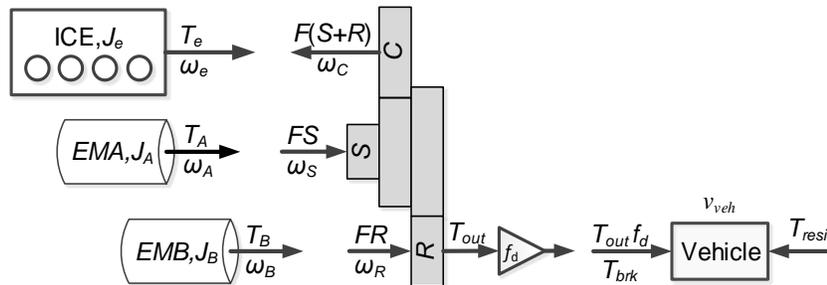


Figure 4. 1 Free-body diagram of power transmission and vehicle dynamics

In order to obtain the control-oriented dynamic model, four key assumptions are made. They are a) all the shaft connections in power transmission system are rigid and no mechanical power loss; b) the inertias of the IC engine and EMs are lumped with the inertias of carrier, sun, and ring

gears; c) the pinion gears and coupler inertias are ignored; and d) only longitudinal vehicle dynamics are considered.

The powertrain and vehicle dynamics are modeled as following:

$$J_e \dot{\omega}_e = T_e - F(S + R) \quad (3.1)$$

$$J_A \dot{\omega}_A = T_A + F \cdot S \quad (3.2)$$

$$J_B \dot{\omega}_B = T_B + F \cdot R - T_{out} \quad (3.3)$$

$$m_{veh} \dot{v}_{veh} = \frac{T_{out} f_d + T_{brk}}{r_w} - \frac{1}{2} \rho C_d A_f v_{veh}^2 - m_{veh} g (f_r \cos \theta + \sin \theta) \quad (3.4)$$

where  $J_e$ ,  $J_A$ , and  $J_B$  are inertias of the IC engine, EMA, and EMB, respectively;  $\omega_e$ ,  $\omega_A$ , and  $\omega_B$  are the corresponding speeds, and  $T_e$ ,  $T_A$ , and  $T_B$  are the torques.  $F$  represents the internal force applied to the planetary pinion gears;  $T_{out}$  is the torque output of the powertrain;  $m_{veh}$ ,  $v_{veh}$ ,  $r_w$ , and  $f_d$  are the vehicle mass, speed, tire radius, and final drive ratio, respectively;  $g$  is the acceleration of gravity;  $f_r$  is the rolling resistance coefficient;  $T_{brk}$  is the brake torque;  $\rho$ ,  $C_d$ , and  $A_f$  are the air density, vehicle drag coefficient and front area, respectively; and  $R$ ,  $S$ , and  $C$  are the numbers of teeth of the ring, sun and carrier gears, respectively. They obey the following constrains:

$$\omega_A S + \omega_B R = \omega_e (R + S) \quad (3.5)$$

and

$$v_{veh} = \omega_B r_w / f_d \quad (3.6)$$

By substituting equations (3.5) and (3.6) into (3.1) - (3.4) to eliminate the interaction force  $F$ , the system model can be simplified below:

$$\left[ J_e + J_A \left( \frac{R+S}{S} \right)^2 \right] \dot{\omega}_e - \left[ J_A \frac{R(R+S)}{S^2} \right] \dot{\omega}_B = T_e + \frac{R+S}{S} T_A \quad (3.7)$$

$$\begin{aligned} \left[ -J_A \frac{R(R+S)}{S^2} \right] \dot{\omega}_e + \left[ J_B + m_{veh} \left( \frac{r_w}{f_d} \right)^2 + J_A \left( \frac{R}{S} \right)^2 \right] \dot{\omega}_B \\ = T_B - \frac{R}{S} T_A + \frac{T_{brk}}{f_d} - \frac{1}{2} \rho C_d A_f \frac{\omega_B^2 r_w^3}{f_d^3} - m_{veh} g \frac{r_w}{f_d} (f_r \cos \theta + \sin \theta) \end{aligned} \quad (3.8)$$

#### 4.2.2 Hybrid battery dynamics

Battery SOC is a very important parameter that reflects the status of the battery energy reservation. Its dynamics [72], [82] is described by

$$\dot{SOC} = - \frac{V_{oc} - \sqrt{V_{oc}^2 - 4r_{bat} P_{bat}}}{2r_{bat} C_m(T)} \quad (3.9)$$

where  $V_{oc}$ ,  $r_{bat}$ , and  $C_m(T)$  are the battery open-circuit voltage, internal resistance, and capacity, respectively. Note that the battery capacity is a function of temperature  $T$ .  $P_{bat}$  is the battery power usage that can be expressed as,

$$P_{bat} = T_A \omega_A + T_B \omega_B \quad (3.10)$$

or described as the following for non-regeneration case,

$$P_{bat} = T_{des} \omega_B - T_e \omega_e \quad (3.11)$$

where  $T_{des}$  is the desired powertrain torque that is obtained from the acceleration and braking pedal position. Noting that positive battery power indicates discharging and negative represents charging.

#### 4.3 Model predictive control

In this section, a model predictive control (MPC) strategy is presented, where the linear quadratic tracking (LQT) approach is used to obtain the closed-loop control law for the system

model discretized at the current operational condition. The LQT control tracks the predicted desired-torque trajectory over the finite prediction horizon while the equivalent fuel consumption is optimized and the battery SOC is maintained at the desired level.

#### 4.3.1 Control-oriented system model

Since the hybrid vehicle system model described in (3.7) - (3.9) is nonlinear, the vehicle model needs to be linearized at each sample time under its current operational condition. Due to the fact that the brake torque is always an independent input which is from the driver directly, so there are three input torques for the system,  $T_e$ ,  $T_A$ , and  $T_B$ . The linearized powertrain model around an operational condition ( $\omega_{e0}$ ,  $\omega_{B0}$ ,  $SOC_0$ ,  $T_{e0}$ ,  $T_{A0}$ ,  $T_{B0}$ , and  $T_{out0}$ ) can be expressed below

$$\begin{bmatrix} \dot{\tilde{\omega}}_e \\ \dot{\tilde{\omega}}_B \end{bmatrix} = \begin{bmatrix} 0 & \frac{E_{12}\rho C_d A_f r_w^3 \omega_{B0}}{f_d^3 E_D} \\ 0 & -\frac{E_{11}\rho C_d A_f r_w^3 \omega_{B0}}{f_d^3 E_D} \end{bmatrix} \begin{bmatrix} \tilde{\omega}_e \\ \tilde{\omega}_B \end{bmatrix} + \begin{bmatrix} \frac{E_{22}}{E_D} & \frac{E_{22}(R+S)+E_{12}R}{SE_D} & \frac{-E_{12}}{E_D} \\ -\frac{E_{21}}{E_D} & -\frac{E_{21}(R+S)+E_{11}R}{SE_D} & \frac{E_{11}}{E_D} \end{bmatrix} \begin{bmatrix} \tilde{T}_e \\ \tilde{T}_A \\ \tilde{T}_B \end{bmatrix} \quad (3.12)$$

where

$$E_{11} = J_e + J_A \left( \frac{R+S}{S} \right)^2, \quad E_{12} = E_{21} = -J_A \frac{R(R+S)}{S^2}, \quad E_{22} = J_B + m_{veh} \left( \frac{r_w}{f_d} \right)^2 + J_A \left( \frac{R}{S} \right)^2, \quad \text{and}$$

$$E_D = E_{11}E_{22} - E_{12}E_{21}.$$

The associated linearized SOC model is

$$S\dot{OC} = \Gamma_{sA} \tilde{T}_A + \Gamma_{sB} \tilde{T}_B + \Gamma_{soe} \tilde{\omega}_e + \Gamma_{sob} \tilde{\omega}_B \quad (3.13)$$

where

$$\Gamma_{sA} = [\omega_{e0}(R+S) - \omega_{B0}R] / S / \Omega, \quad \Gamma_{sB} = \omega_{B0} / \Omega, \quad \Gamma_{soe} = T_{A0}(R+S) / S / \Omega,$$

$$\Gamma_{sob} = (T_{B0} - T_{A0}R/S) / \Omega, \quad \Omega = -C_m \sqrt{V_{OC}^2 - 4r_{bat} [T_{A0} [\omega_{e0}(R+S) - \omega_{B0}R] / S + T_{B0}\omega_{B0}]}$$

Due to the actuator dynamics (engine, EMA, and EMB), the following dynamic equations are augmented into the linearized system.

$$\begin{cases} \dot{T}_e = -\frac{1}{\tau_e}T_e + \frac{1}{\tau_e}T_{e\_des} \\ \dot{T}_A = -\frac{1}{\tau_A}T_A + \frac{1}{\tau_A}T_{A\_des} \\ \dot{T}_B = -\frac{1}{\tau_B}T_B + \frac{1}{\tau_B}T_{B\_des} \end{cases} \quad (3.14)$$

where  $\tau_e$ ,  $\tau_A$ , and  $\tau_B$  are the time constants of IC engine, EMA, and EMB, respectively;  $T_{e\_des}$ ,  $T_{A\_des}$ , and  $T_{B\_des}$  are the desired torque of IC engine, EMA, and EMB, respectively; and the relationships between system inputs and fuel  $\dot{m}_e$  (or equivalent fuel consumption  $\dot{m}_{eq}$ ) are,

$$\begin{cases} \dot{m}_e = \frac{T_{e\_des}\omega_e}{\eta_e H_{LHV}} \\ \dot{m}_{eq} = \frac{C_{bat}}{2r_{bat}} \left[ V_{oc}^2 - V_{oc} \sqrt{V_{oc}^2 - 4r_{bat} P_{bat}} \right] \end{cases} \quad (3.15)$$

where  $\dot{m}_e$  is the actual IC engine fuel mass flowrate;  $H_{LHV}$  is the corresponding low heating value [42] of fuel;  $\eta_e$  is the IC engine efficiency varying over the engine operational map;  $\dot{m}_{eq}$  is the equivalent fuel consumption (flowrate) of the battery pack [42]; and  $C_{bat}$  is the equivalent fuel economy coefficients for the battery defined by

$$C_{bat} = \begin{cases} \alpha_1 [1 - \beta_1 (SOC - SOC_T)] & \text{discharging} \\ \alpha_2 [1 + \beta_2 (SOC - SOC_T)] & \text{charging} \end{cases} \quad (3.16)$$

where  $\alpha_1$ ,  $\alpha_2$ ,  $\beta_1$ , and  $\beta_2$  are weighting coefficients to be selected to optimize the equivalent fuel economy and to encourage battery discharging (or charging) when SOC is greater (or less) than the desired target  $SOC_T$ ; see [42] for details.

Since the goal is to minimize the total equivalent fuel consumption and the error between the actual powertrain output torque and the predicted desired-torque trajectory, the following cost function is defined,

$$\mathbf{J}_c = \int_0^{N*t_d} \left\{ [T_{pre}(\tau) - y(\tau)]^2 + \left( r_e [\dot{m}_e(T_{e\_des}, \tau)]^2 + r_{eq} [\dot{m}_{eq}(T_{A\_des}, T_{B\_des}, \tau)]^2 \right) \right\} d\tau \quad (3.17)$$

where  $t_d$  is the step size;  $N$  is the number of prediction steps;  $r_e$  and  $r_{eq}$  are the weighting coefficients for engine and battery equivalent fuel consumption; and  $T_{pre}(\tau)$  is the desired-torque prediction output based on the past and current desired powertrain torques using the adaptive recursive prediction algorithm (see [45] for details).

Since the cost function (3.17) is fairly nonlinear due to (3.15) and also the control inputs need to be converted in terms of  $\dot{m}_e$  and  $\dot{m}_{eq}$ , let

$$\mathbf{u} = [\dot{m}_e \quad \dot{m}_{eq}]^T \quad (3.18)$$

Using (3.11) and (3.15) yields the following equations

$$\begin{cases} T_{e\_des} = \frac{\eta_e H_{LHV}}{\omega_e} \dot{m}_e \\ T_{des} = \frac{1}{C_{bat} \omega_B} \dot{m}_{eq} - \frac{r_{bat}}{V_{oc}^2 C_{bat}^2 \omega_B} \dot{m}_{eq}^2 + \frac{T_e \omega_e}{\omega_B} \end{cases} \quad (3.19)$$

The linearized model of (3.14) at the same operation condition ( $\omega_{e0}$ ,  $\omega_{B0}$ ,  $SOC_0$ ,  $T_{e0}$ ,  $T_{A0}$ ,  $T_{B0}$ ,  $T_{out0}$ ,  $\dot{m}_{e0}$  and  $\dot{m}_{eq0}$ ) is shown below.

$$\begin{cases} \dot{\tilde{T}}_e = -\frac{1}{\tau_e} \tilde{T}_e + \Gamma_{e\omega e} \tilde{\omega}_e + \Gamma_{em} \tilde{m}_e \\ \dot{\tilde{T}}_A = \Gamma_{ATe} \tilde{T}_e - \frac{1}{\tau_A} \tilde{T}_A \\ \dot{\tilde{T}}_B = \Gamma_{BT_e} \tilde{T}_e - \frac{1}{\tau_B} \tilde{T}_B + \Gamma_{B\omega e} \tilde{\omega}_e + \Gamma_{B\omega B} \tilde{\omega}_B + \Gamma_{Beq} \tilde{m}_{eq} \end{cases} \quad (3.20)$$

where

$$\Gamma_{e\omega e} = -\frac{\eta_e H_{LHV} \dot{m}_{e0}}{\tau_e \omega_{e0}^2}, \Gamma_{em} = \frac{\eta_e H_{LHV}}{\tau_e \omega_{e0}}, \Gamma_{ATe} = -\frac{S}{\tau_A (R+S)}, \Gamma_{BTe} = -\frac{1}{\tau_B} \left( \frac{R}{R+S} - \frac{\omega_{e0}}{\omega_{B0}} \right),$$

$$\Gamma_{Beq} = \frac{1}{\tau_B C_{bat} \omega_{B0}} \left( 1 - 2 \frac{r_{bat}}{V_{oc}^2 C_{bat}} \right), \Gamma_{B\omega B} = -\frac{1}{\tau_B \omega_{B0}^2} \left( \frac{\dot{m}_{eq0}}{C_{bat}} - \frac{r_{bat} \dot{m}_{eq0}^2}{V_{oc}^2 C_{bat}^2} + T_{e0} \omega_{e0} \right), \Gamma_{B\omega e} = \frac{T_{e0}}{\tau_B \omega_{B0}}.$$

Therefore, equation (3.20) together with (3.12) and (3.13), forms the linearized system expressed below,

$$\begin{cases} \dot{\tilde{x}} = A_c \tilde{x} + B_c \tilde{u} \\ \tilde{y} = C_c \tilde{x} \end{cases}, \quad \tilde{u} = \begin{bmatrix} \dot{\tilde{m}}_e \\ \dot{\tilde{m}}_{eq} \end{bmatrix}, \quad \tilde{y} = \tilde{T}_{out} \quad (3.21)$$

where

$$A_c = \begin{bmatrix} -\frac{1}{\tau_e} & 0 & 0 & \Gamma_{e\omega e} & 0 & 0 \\ \Gamma_{ATe} & -\frac{1}{\tau_A} & 0 & 0 & 0 & 0 \\ \Gamma_{BTe} & 0 & -\frac{1}{\tau_B} & \Gamma_{B\omega e} & \Gamma_{B\omega B} & 0 \\ \frac{E_{22}}{E_D} & \frac{E_{22}(R+S) + E_{12}R}{SE_D} & \frac{-E_{12}}{E_D} & 0 & \frac{E_{12}\rho C_d A_f r_w^3 \omega_{B0}}{f_d^3 E_D} & 0 \\ \frac{-E_{21}}{E_D} & \frac{-E_{21}(R+S) + E_{11}R}{SE_D} & \frac{E_{11}}{E_D} & 0 & \frac{-E_{11}\rho C_d A_f r_w^3 \omega_{B0}}{f_d^3 E_D} & 0 \\ 0 & \Gamma_{sA} & \Gamma_{sB} & \Gamma_{s\omega e} & \Gamma_{s\omega B} & 0 \end{bmatrix}, \quad B_c = \begin{bmatrix} \Gamma_{em} & 0 \\ 0 & 0 \\ 0 & \Gamma_{Beq} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$$C_c^T = \begin{bmatrix} 0 \\ -R/S \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \tilde{x} = \begin{bmatrix} \tilde{T}_e \\ \tilde{T}_A \\ \tilde{T}_B \\ \tilde{\omega}_e \\ \tilde{\omega}_B \\ S\hat{O}C \end{bmatrix}.$$

Note that  $\omega_e = \omega_{e0} + \tilde{\omega}_e$ ,  $\omega_B = \omega_{B0} + \tilde{\omega}_B$ ,  $SOC = SOC_0 + \tilde{SOC}$ ,  $T_e = T_{e0} + \tilde{T}_e$ ,  $T_A = T_{A0} + \tilde{T}_A$ ,  $T_B = T_{B0} + \tilde{T}_B$ ,  $T_{out} = T_{out0} + \tilde{T}_{out}$ ,  $T_{A\_des} = -T_e S / (R + S)$ , and  $T_{B\_des} = T_{des} - T_e R / (R + S)$ .

and the following constraints need to be satisfied to operate the powertrain within its physical limits,

$$0 \leq T_e \leq T_e^{\max}; 0 \leq \omega_e \leq \omega_e^{\max}; T_B^{\min} \leq T_B \leq T_B^{\max}; \omega_B^{\min} \leq \omega_B \leq \omega_B^{\max};$$

$$T_A^{\min} \leq T_A \leq T_A^{\max}; \omega_A^{\min} \leq \omega_A \leq \omega_A^{\max}; P_{bat}^{\min} \leq P_{bat} \leq P_{bat}^{\max}; SOC^{\min} \leq SOC \leq SOC^{\max}.$$

where the superscript “min” and “max” refer to the known lower and upper limit, respectively.

Also note that the battery power constraints,  $P_{bat}^{\min}$  and  $P_{bat}^{\max}$ , are time-varying as a function of temperature.

The linearized continuous-time model at current time index “ $j$ ” is then discretized with a sample period of 10 *ms* for the LQT control in discrete-time domain based on [36].

$$\begin{cases} \tilde{x}_{j,k+1} &= A_d(j)\tilde{x}_{j,k} + B_d(j)\tilde{u}_{j,k} \\ \tilde{y}_{j,k} &= C_d(j)\tilde{x}_{j,k} \end{cases} \quad (3.22)$$

where  $A_d(j) = e^{A_c \tau_s}$ ,  $B_d(j) = e^{A_c \tau_s} B_c \tau_s$ ,  $C_d(j) = C_c$ ,  $\tilde{u}_{j,k}^T = [\dot{\tilde{m}}_{j,e\_k} \quad \dot{\tilde{m}}_{j,eq\_k}]$ ,

$\tilde{x}_{j,k}^T = [\tilde{T}_{j,ek} \quad \tilde{T}_{j,Ak} \quad \tilde{T}_{j,Bk} \quad \tilde{\omega}_{j,ek} \quad \tilde{\omega}_{j,Bk} \quad \tilde{SOC}_{j,k}]$ , and  $\tilde{y}_{j,k} = \tilde{T}_{j,out\_k}$ . Note that  $A_d(j)$ ,  $B_d(j)$ , and

$C_d(j)$  need to be linearized at each sampling point “ $j$ ” under the current operational condition so that the LQT control can be utilized. Note the subscript “ $j$ ” represents the current discrete-time index and “ $k$ ” is prediction horizon index.

### 4.3.2 Linear quadratic tracking

In this subsection, a finite horizon LQT controller is designed based on the linearized discrete-time system model at current operational condition to track the predicted desired-torque trajectory.

Considering the system model obtained in (3.22), the discrete-time cost function of (3.17) for the current time index “ $j$ ” can be written as

$$\begin{aligned} \mathbf{J}(j) = & \frac{1}{2} \left[ \tilde{T}_{pre}(j, N) - \tilde{y}_{j, N} \right]^T \mathbf{Q} \left[ \tilde{T}_{pre}(j, N) - \tilde{y}_{j, N} \right] \\ & + \frac{1}{2} \sum_{k=0}^{N-1} \left\{ \left[ \tilde{T}_{pre}(j, k) - \tilde{y}_{j, k} \right]^T \mathbf{Q} \left[ \tilde{T}_{pre}(j, k) - \tilde{y}_{j, k} \right] + \tilde{u}_{j, k}^T \mathbf{R} \tilde{u}_{j, k} \right\} \end{aligned} \quad (3.23)$$

where  $\mathbf{Q}$  is a positive semi-definite symmetric matrix;  $\mathbf{R} = \text{diag}[r_e, r_{eq}]$  is a positive definite symmetric matrix;  $\tilde{T}_{pre}(j, k) = T_{pre}(j, k) - T_{pre}(j, k-1)$ , and when  $k=0, T_{pre}(j, 0) - T_{des}(j)$ ; The initial state is  $x_{j,0}$ ; the final state  $x_{j,N}$  is free; and  $N$  is fixed. This is a typical finite horizon LQT problem with the following performance output

$$e_{j, k} = \tilde{T}_{pre}(j, k) - \tilde{y}_{j, k} \quad (3.24)$$

For the prediction information, it is stored in a reversed buffer in which the newest predicted desired torque information stays at the end of the buffer while the oldest one is at the beginning, see Figure 4. 2.

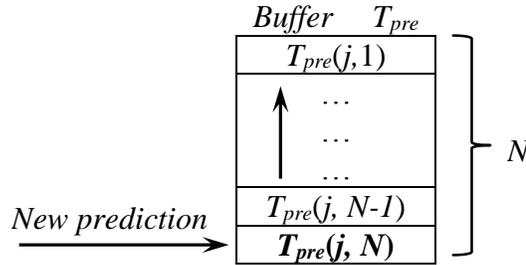


Figure 4. 2 Prediction buffer

In order to obtain the optimal tracking solution, define the following Hamiltonian function [84].

$$\begin{aligned} \mathbf{H}(x_{j, k}, \tilde{u}_{j, k}, \lambda_{j, k+1}) = & \frac{1}{2} \sum_{k=0}^{N-1} \left\{ \left[ \tilde{T}_{pre}(j, k) - \tilde{y}_{j, k} \right]^T \mathbf{Q} \left[ \tilde{T}_{pre}(j, k) - \tilde{y}_{j, k} \right] + \tilde{u}_{j, k}^T \mathbf{R} \tilde{u}_{j, k} \right\} \\ & + \lambda_{j, k+1} \left[ \mathbf{A}_d(j) \tilde{x}_{j, k} + \mathbf{B}_d(j) \tilde{u}_{j, k} \right] \end{aligned} \quad (3.25)$$

Based on [84] the optimal state  $\tilde{x}_{j,k}^*$ , co-state  $\lambda_{j,k}^*$ , and control  $\tilde{u}_{j,k}^*$  satisfy the following equations:

$$\frac{\partial \mathbf{H}}{\partial \lambda_{j,k+1}^*} = \tilde{x}_{j,k+1}^* = A_d(j)\tilde{x}_{j,k}^* + B_d(j)\tilde{u}_{j,k}^* \quad (3.26)$$

$$\frac{\partial \mathbf{H}}{\partial \tilde{x}_{j,k}^*} = \lambda_{j,k}^* = A_d^T(j)\lambda_{j,k+1}^* + C_d^T(j)\mathbf{Q}C_d(j)\tilde{x}_{j,k}^* - C_d^T(j)\mathbf{Q}\tilde{T}_{pre}(j,k) \quad (3.27)$$

$$\frac{\partial \mathbf{H}}{\partial \tilde{u}_{j,k}^*} = 0 = B_d^T(j)\lambda_{j,k+1}^* + \mathbf{R}\tilde{u}_{j,k}^* \Rightarrow \tilde{u}_{j,k}^* = -\mathbf{R}^{-1}B_d^T(j)\lambda_{j,k+1}^* \quad (3.28)$$

Therefore, the augmented system is

$$\begin{bmatrix} \tilde{x}_{j,k+1}^* \\ \lambda_{j,k}^* \end{bmatrix} = \begin{bmatrix} A_d(j) & -B_d(j)\mathbf{R}^{-1}B_d^T(j) \\ C_d^T(j)\mathbf{Q}C_d(j) & A_d^T(j) \end{bmatrix} \begin{bmatrix} \tilde{x}_{j,k}^* \\ \lambda_{j,k+1}^* \end{bmatrix} - \begin{bmatrix} 0 \\ C_d^T(j)\mathbf{Q} \end{bmatrix} \tilde{T}_{pre}(j,k) \quad (3.29)$$

and the terminal condition can be expressed as

$$\lambda_{j,N}^* = C_d^T(j)\mathbf{Q}C_d(j)\tilde{x}_{j,N}^* - C_d^T(j)\mathbf{Q}\tilde{T}_{pre}(j,N) \quad (3.30)$$

Let  $\lambda_{j,k}^* = \mathbf{P}_{j,k}\mathbf{x}_{j,k}^* - \mathbf{g}_{j,k}$  and eliminating  $\lambda_{j,k}^*$  and  $\lambda_{j,k+1}^*$  leads to

$$\mathbf{P}_{j,k} = A_d^T(j)\mathbf{P}_{j,k+1} \left[ \mathbf{I} + B_d(j)\mathbf{R}^{-1}B_d^T(j)\mathbf{P}_{j,k+1} \right]^{-1} A_d(j) + C_d^T(j)\mathbf{Q}C_d(j) \quad (3.31)$$

$$\mathbf{g}_{j,k} = A_d^T(j) \left\{ \mathbf{I} - \left[ \mathbf{P}_{j,k+1}^{-1} + B_d(j)\mathbf{R}^{-1}B_d^T(j) \right]^{-1} B_d(j)\mathbf{R}^{-1}B_d^T(j) \right\} \mathbf{g}_{j,k+1} + C_d^T(j)\mathbf{Q}\tilde{T}_{pre}(j,k+1) \quad (3.32)$$

with the following terminal conditions

$$\mathbf{P}_{j,N} = C_d^T(j)\mathbf{Q}C_d(j) \text{ and } \mathbf{g}_{j,N} = C_d^T(j)\mathbf{Q}\tilde{T}_{pre}(j,N) \quad (3.33)$$

Note that (3.31) is a matrix difference Riccati equation (DRE) that can be solved backwards using the terminal condition (3.33) and the vector difference equation (3.32) can also be solved backwards using the terminal condition (3.33). The resulting optimal control  $\tilde{u}_{j,k}^*$  can be obtained for  $k=0, 1, \dots, N-1$ ,

$$\tilde{\mathbf{u}}_{j,k}^* = -(\mathbf{R} + \mathbf{B}_d^T(j)\mathbf{P}_{j,k+1}\mathbf{B}_d(j))^{-1} \mathbf{B}_d^T(j)(\mathbf{P}_{j,k+1}\mathbf{A}_d(j)\tilde{\mathbf{x}}_{j,k} + \mathbf{g}_{j,k+1}) \quad (3.34)$$

Finally, at current time index “ $j$ ”, let  $\Delta\mathbf{u}_j^* = \tilde{\mathbf{u}}_{j,0}^*$ , the MPC-LQT optimal control is

$$\mathbf{u}_j^* = \mathbf{u}_{j-1}^* + \Delta\mathbf{u}_j^*; \quad \mathbf{u}_{-1}^* = 0 \text{ and } j = 0, 1, \dots \quad (3.35)$$

The MPC-LQT control algorithm is also shown in Figure 4. 3.

Note that the engine-on strategy [55], [82], based on the predicted desired torque, is used to keep the battery operated within its operational boundary.

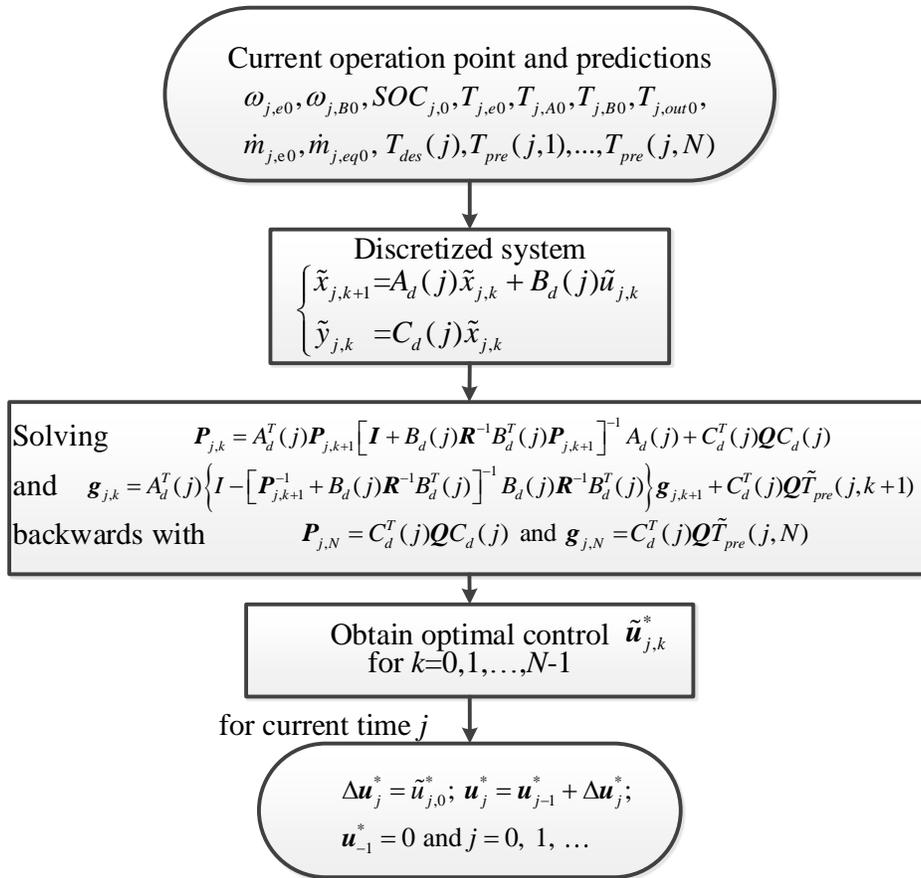


Figure 4. 3 Flow chart of MPC-LQT algorithm

#### 4.4 Implementation feasibility study

As shown in Figure 4. 3, to solve the matrix sequence  $\mathbf{P}_{j,k}$ , DRE (3.31) needs to be iterated for  $N$  (prediction horizon) times at each step, yielding very high computational load (or time

complexity). This makes it impossible for real-time implementation. This section intends to find an approximation of the DRE solution with low computational load.

#### 4.4.1 Steady-state discrete-time Riccati solution

It is well known that as  $N$  goes to infinity, the finite horizon Riccati equation becomes the infinite horizon one. That is, at time index “ $j$ ” the Riccati equation solution of (3.31) is a constant matrix  $\bar{P}_j$  satisfying the following algebraic Riccati equation (ARE)

$$\bar{P}_j = A_d^T(j)\bar{P}_j A_d(j) + C_d^T(j)QC_d(j) - A_d^T(j)\bar{P}_j B_d(j) [B_d^T(j)\bar{P}_j B_d(j) + R]^{-1} B_d^T(j)\bar{P}_j A_d(j) \quad (3.36)$$

Therefore, it is logical to use the ARE solution of infinity horizon problem to approximate the DRE solution.

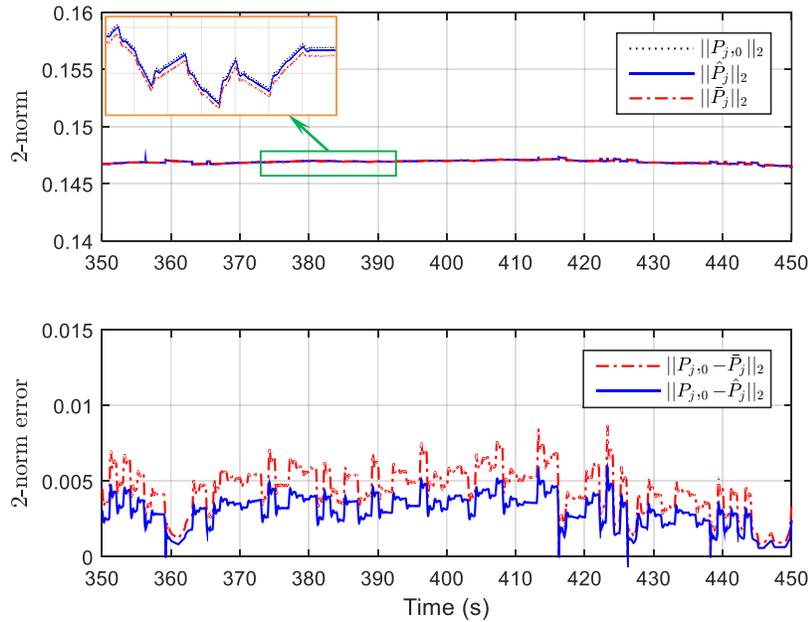


Figure 4. 4  $\|P_{j,0} - \bar{P}_j\|_2$  and  $\|P_{j,0} - \hat{P}_j\|_2$  traces under US06 cycle

Figure 4. 4 shows the 2-norm error trace (dash-dot line) between the first ( $k = 0$ ) DRE solution at current step  $P_{j,0}$  and ARE solution  $\bar{P}_j$  under US06 driving cycle, where the error is defined as

$$\bar{e}_j = \|\mathbf{P}_{j,0} - \bar{\mathbf{P}}_j\|_2 \quad (3.37)$$

It can be seen from the lower plot of Figure 4. 4 that the maximum relative error defined below is around 6.8%.

$$\bar{e}_{j,r} = \bar{e}_j \left( \|\mathbf{P}_{j,0}\|_2 \right)^{-1} \quad (3.38)$$

The effect of 6.8% relative error to the actual fuel economy will be studied in the next section.

#### 4.4.2 Iterative ARE solution

The second approximation is based on the observation that the powertrain operational condition changes gradually. As a result, the solution of DRE shall also vary slowly. Therefore, it is possible to approximate the DRE solution by an iterative solution on the following Riccati equation

$$\hat{\mathbf{P}}_j = A_d^T(j)\hat{\mathbf{P}}_{j-1}A_d(j) + C_d^T(j)\mathbf{Q}C_d(j) - A_d^T(j)\hat{\mathbf{P}}_{j-1}B_d(j) \left[ B_d^T(j)\hat{\mathbf{P}}_{j-1}B_d(j) + \mathbf{R} \right]^{-1} B_d^T(j)\hat{\mathbf{P}}_{j-1}A_d(j) \quad (3.39)$$

where  $\hat{\mathbf{P}}_0 = \bar{\mathbf{P}}_0$ . Using similar absolute and relative error definitions to (3.37) and (3.38) leads to,

$$\hat{e}_j = \|\mathbf{P}_{j,0} - \hat{\mathbf{P}}_j\|_2 \quad \text{and} \quad \hat{e}_{j,r} = \hat{e}_j \left( \|\mathbf{P}_{j,0}\|_2 \right)^{-1} \quad (3.40)$$

The associated 2-norm error trace is plotted in Figure 4. 4 (solid line) with around 4.8% maximum relative error that is less than ARE one.

#### 4.4.3 Computational load

Time complexity is often used to evaluate the computational load of an algorithm and can be expressed as

$$T(n) = O(f(n)) \quad (3.41)$$

where  $T(n)$  is the maximum amount of time used for any input size  $n$ ;  $O(\cdot)$  denotes the time complexity; and  $f(n)$  is the growth rate of computational load [45], [43]-[44]. Table 4. 1 lists the

2-norm error and associated time complexity of three Riccati solution approaches. It can be seen that the DRE solution has very high time complexity of  $O(55800)$  due to the matrix equation solved backwards iteratively; the ARE solution approach solves the Riccati equation directly and reduces the time complexity greatly down to  $O(6696)$  with a peak 2-norm error of 6.8%; and the iterative ARE solution has a time complexity of  $O(1116)$  with a maximum relative 2-norm error of 4.8%.

Table 4. 1 2-norm error and Time complexity ( $n=6, N=50$ )

	DRE solution	ARE solution	Iterative ARE solution
2-norm error (max)	-	6.8%	4.8%
Optimization	Optimal	approximated	approximated
Growth rate $f(n)$	$N \times [5(n^3) + n^2]$	$n \times [5(n^3) + n^2]$	$5(n^3) + n^2$
Time Complexity	$O(55800)$	$O(6696)$	$O(1116)$

Therefore, it is expected that the iterative ARE solution shall provide a good approximation to the exact DRE solution with significantly reduced time complexity.

#### 4.5 Simulation study and validation

Table 4. 2 Key simulation parameters

Item	Parameter	Value
Vehicle	Mass	1750kg
	Start delay	0.5 s
Engine	Time constant	0.15s
	Power output	41 kW
	Max torque	115 N-m
Battery	SOC upper bound	0.75
	SOC lower bound	0.45
	SOC target	0.60
Motor	Power output	30 kW
	Max torque	320 N-m

The developed forward HEV model (see vehicle parameters in Table 4. 2) was used to study the performance of the MPC-LQT scheme under four typical driving cycles, US06, ARB02, IM240, and FTP, where US06 and ARB02 are with aggressive driving behavior and fast speed variations; IM240 is for the highway, and FTP is with mixed city and highway driving.

#### 4.5.1 Power-follower and MPC-LQT control strategies comparison

Figure 4. 5 shows the simulation results of the power-follower and MPC-LQT control under US06 driving cycle, where the DRE solution is used to generate the LQT solution. The weight coefficients for this simulation are  $Q = 1$ ,  $r_e = 35$ , and  $r_{eq} = 300$ .

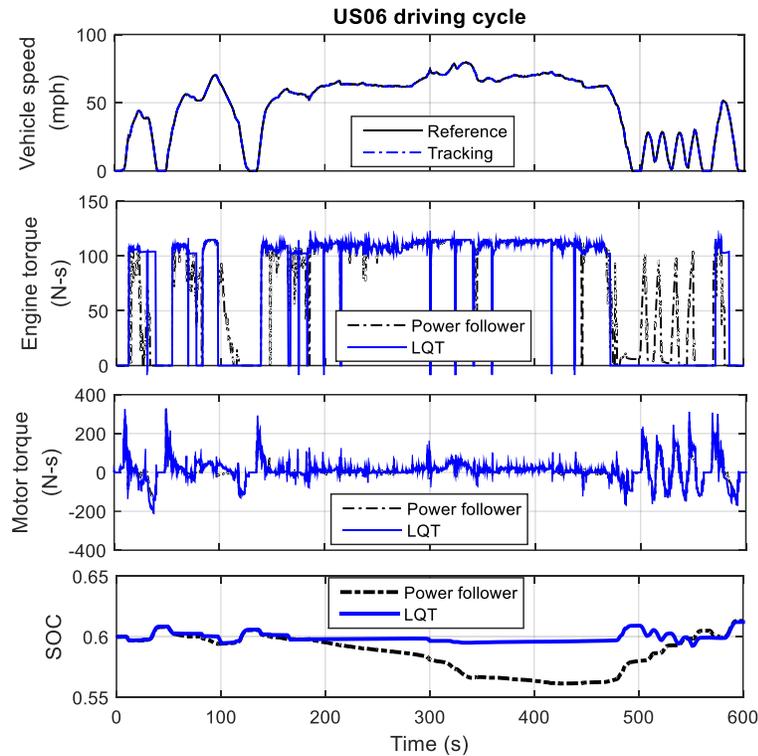


Figure 4. 5 Power-follower and MPC-LQT performance

- **Acceleration:** Near 10s, 50s, 130s and after 500s, motor assists engine to deliver the required power;
- **Engine-on operation:** Once the engine is on (see [82] for detailed engine-on strategy), the MPC-LQT splits the requested power optimally between engine and battery, and both the engine and motor torque outputs are optimized.
- **SOC:** The MPC-LQT maintains the SOC close to the target level, and the power-follower tends to deep discharge the battery.
- **Deceleration:** the motor is in regeneration mode when the vehicle decelerates, where the

mechanical energy is converted to electricity to charge the battery at around 25s, 100s, 480s, and after 500s.

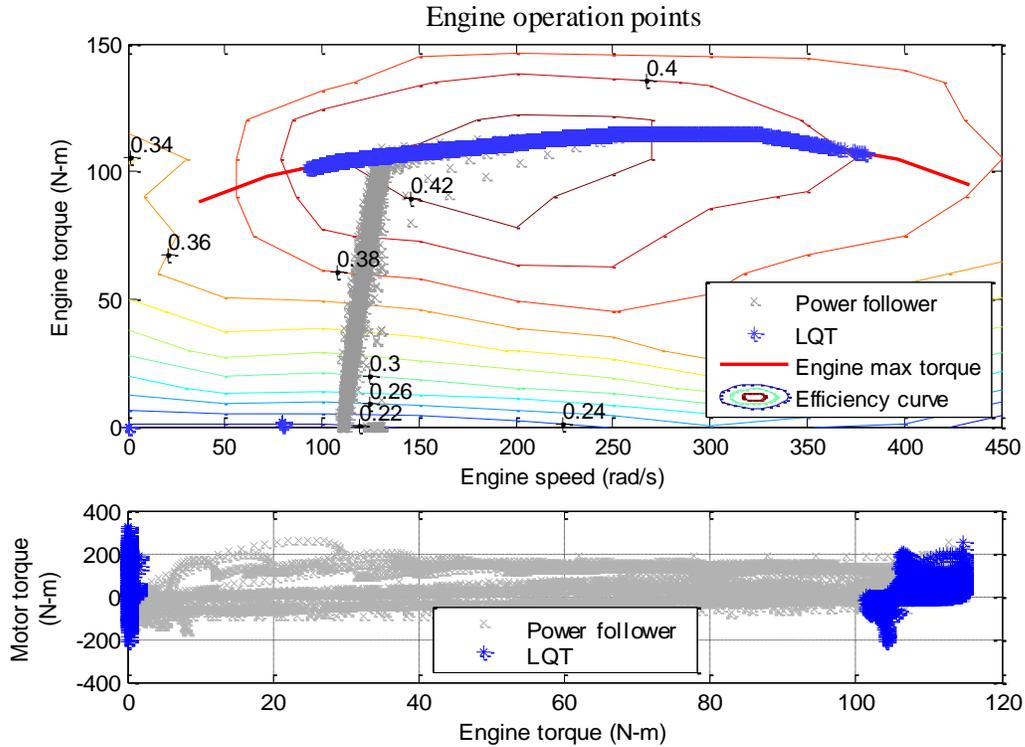


Figure 4. 6 Engine and motor operation maps under US06 driving cycle

As stated in [42], the fuel economy performance of the internal combustion engine mainly depends on its operation conditions. Figure 4. 6 shows engine and motor operation maps of both power-follower and MPC-LQT control strategies, where in the power follower control strategy the engine output power follows the driver's requested, and in contrast, the MPC-LQT scheme considers both engine efficiency and battery operational range. That is, the engine is operated in its most efficient region and at the same time the battery is protected from over discharging/charging and its SOC is maintained close to its target. For the battery protection, it is a part of MPC-LQT control strategy also based on desired torque prediction. That is, once the predicted powertrain power is greater than the battery capacity, the IC engine will be turned on

proactively to provide power to the drivetrain [55], [82]. As a result, the battery over-discharging duration is reduced and therefore the battery is protected.

#### 4.5.2 Cost function weight selection

Table 4. 3 Optimal weightings line searching ( $Q=1$ )

<b>Weightings</b>		<b>Tracking error</b>	<b>Fuel consumption</b>
$r_e$	$r_{eq}$	(N-m, RMS)	(gallon / %)
1	10	2.224	0.3178 / 3.2
5	50	2.736	0.3170 / 3.5
10	100	2.892	0.3166 / 3.6
20	200	2.966	0.3163 / 3.7
<b>30</b>	<b>300</b>	<b>3.012</b>	<b>0.3161 / 3.8</b>
<b>40</b>	<b>400</b>	<b>3.018</b>	<b>0.3161 / 3.8</b>
50	500	3.024	0.3162 / 3.7

Note: RMS means root mean square.

As indicated in the cost function (3.23),  $Q$  is a positive semi-definite symmetric matrix (coefficient in this case),  $R = \text{diag} [r_e, r_{eq}]$  is a positive definite symmetric matrix. Since  $Q$  and  $R$  are not independent,  $Q$  is set to 1 and  $R (r_e, r_{eq})$  is used to tune the cost function for the best fuel economy performance. That is,  $r_e$  and  $r_{eq}$  are utilized as tuning parameters for engine fuel economy over a given driving cycle. Note that increasing  $r_e$  leads to reduced engine fuel consumption relatively over the given horizon and the same for  $r_{eq}$  to battery. In this section, a two-step cost function weight selection search is conducted to find the best  $r_e$  and  $r_{eq}$  for the best overall fuel economy under a given driving cycle, where the overall fuel economy is defined as the quantity of the fuel used for the IC engine plus the equivalent fuel consumption due to the SOC difference at the start and end of the driving cycle. Firstly, a fixed ratio of 1 to 10 is used for  $r_e$  and  $r_{eq}$ ; see Table 4. 3 and Figure 4. 7. That is, the weighting coefficients are searched along a line where both  $r_e$  and  $r_{eq}$  are increased linearly. The searching points are listed in Table 4. 3 and shown in Figure

4. 7. The coefficients with the best overall fuel economy are highlighted with bold characters. Note that in this study the US06 driving cycle is used.

Table 4. 4 Optimal weightings area seeking for best fuel consumption ( $Q=1$ )

$r_e \backslash r_{eq}$	250	300	350	400	450
25	0.3162	0.3162	0.3162	0.3163	0.3163
30	0.3162	0.3161	0.3161	0.3162	0.3162
35	0.3161	<b>0.3160</b>	0.3161	0.3162	0.3162
40	0.3161	<b>0.3160</b>	<b>0.3160</b>	0.3161	0.3162
45	0.3161	0.3161	0.3161	0.3162	0.3162

Once the candidate coefficients with the fixed ratio for the best fuel economy are located, area search is conducted around the neighborhood of these coefficients, see Table 4. 4.

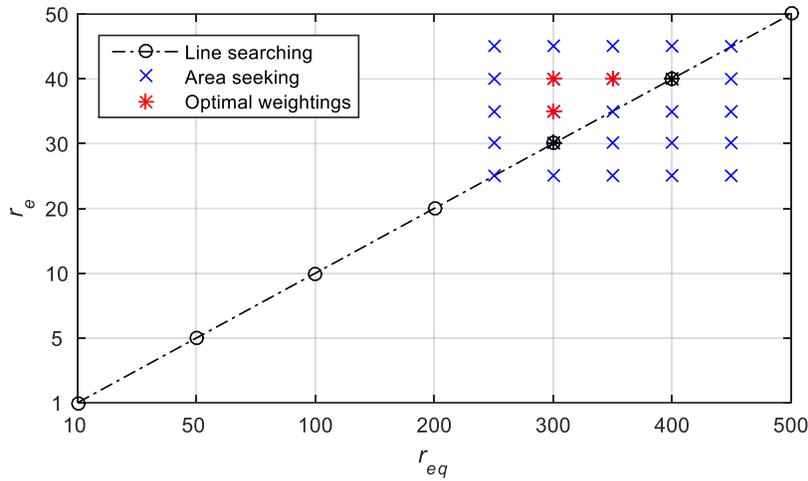


Figure 4. 7 Optimal weightings seeking

As is shown in Table 4. 3, with the increment of weighting coefficients ( $r_e$  and  $r_{eq}$ ) along the search line, the root mean square (RMS) of tracking error goes up slowly. This is expected since the relative weighting for tracking error decreases. The overall fuel consumption reaches its minimal when ( $r_e = 30, r_{eq} = 300$ ) and ( $r_e = 40, r_{eq} = 400$ ) marked with black '\*' in Figure 4. 7. Therefore, the area used for searching weighting coefficients with the best overall fuel economy is listed in Table 4. 4 and marked with blue 'x' in Figure 4. 7, where the optimal weighting

coefficients are positioned at three locations marked with red ‘\*’ and they are  $(r_e = 35, r_{eq} = 300)$ ,  $(r_e = 40, r_{eq} = 300)$ , and  $(r_e = 40, r_{eq} = 400)$ . The optimal weighting coefficients of  $r_e = 35$  and  $r_{eq} = 300$  are selected for the rest of study since it has the smallest tracking error with similar overall fuel economy.

### 4.5.3 Fuel economy improvement study

For this study the overall fuel economy is defined as the accumulated engine fuel consumption plus the equivalent fuel consumption due to the SOC difference at start and end of the driving cycle. Four typical previously mentioned driving cycles are utilized for studying the fuel economy improvement over the traditional power-follower one, and the total equivalent fuel consumptions are listed in Table 4. 5.

Table 4. 5 Total equivalent fuel consumptions

<b>Driving cycles</b> (distance, miles)	<b>Power follower</b> (gallon)	<b>Backwards/ Improvement</b> (gallon/%)	<b>Steady-state/ Improvement</b> (gallon/%)	<b>Iterative/ Improvement</b> (gallon/%)
FTP (11.04)	0.3127	0.2816/9.9	0.2835/9.3	0.2831/9.5
IM240 (1.96)	0.0409	0.0379/7.3	0.0380/7.1	0.0379/7.3
ARB02 (19.8)	0.7385	0.7122/3.6	0.7127/3.5	0.7126/3.5
US06 (8.01)	0.3286	0.3161/3.8	0.3168/3.6	0.3164/3.7

It can be seen from Table 4. 5 that with exact DRE solution the best fuel economy (9.9%) improvement is achieved under FTP mixed city and highway driving cycle. Under IM240, the improvement is 7.3%, and Under ARB02 and US06 are 3.6% and 3.8%, respectively. The reduced fuel economy for ARB02 and US06 is due to the aggressive driving behavior and fast vehicle speed variations. With ARE solution approximating the DRE one, the maximum overall fuel consumption improvement reduces to 9.3% (from 9.9%) for the FTP cycle, and the improvements are 7.1%, 3.5%, and 3.6% for IM240, ARB02 and US06 cycles, respectively. While the

corresponding improvements with iterative ARE solution approximation are 9.5%, 7.3%, 3.5% and 3.7%, respectively. Note that they are slightly better than ARE solution case and little worse than the exact DRE case. The total equivalent fuel economy (consumption) improvements are plotted in Figure 4. 8.

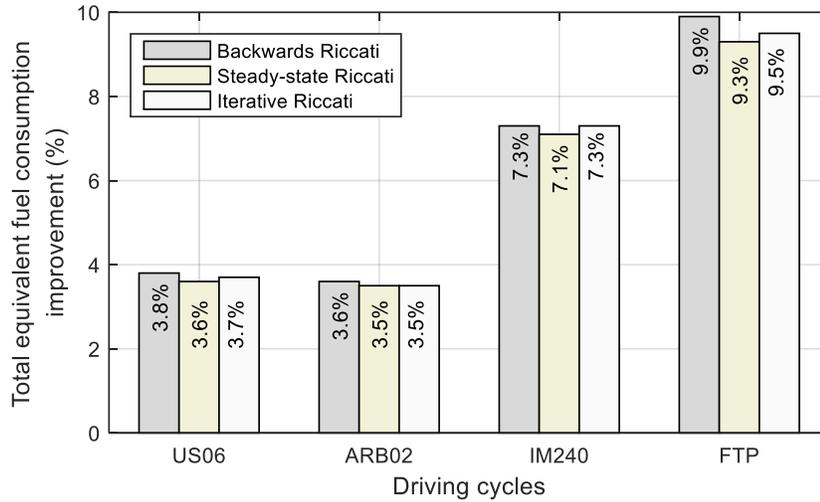


Figure 4. 8 Total fuel consumption improvement comparison

It can be observed from Figure 4. 8 that the optimal MPC-LQT control law is able to improve the overall fuel economy effectively. Using exact DRE solution achieves the best fuel economy among the three options but with extremely heavy computation load (see subsection 4.4.3 for details); Using the ARE solution approximation leads the least overall fuel economy improvement; and the iterative ARE solution approximation yields reasonable overall fuel economy improvement with computational load reduction of 50 times and it is also feasible for real-time implementation. Therefore, balancing the overall fuel economy improvement and computational load, the iterative ARE solution is chosen for MPC-LQT scheme in real-time applications.

#### 4.6 Conclusions

A model predictive control (MPC) scheme, based on linear quadratic tracking (LQT) optimal control, is presented in this chapter. The proposed MPC-LQT supervisory control strategy tracks

the predicted driver torque demand and minimizes the equivalent fuel consumption over a given horizon and at the same time the battery state of charge (SOC) is maintained close to the target level and the battery is operated within its operational boundary. The effectiveness of the proposed MPC-LQT control strategy is validated in simulations under four typical driving cycles. Compared with baseline power-follower control strategy, the fuel economy is greatly improved with the exact difference Riccati equation (DRE) solution. To be specific, 9.9% is achieved over the traditional power-follower scheme under FTP driving cycle, 7.3% under IM240, 3.6% under ARB02, and 3.8% under US06. Since the MPC-LQT scheme using the exact DRE solution has extremely high computational load and it is not feasible for real-time implementation, two approximated solutions to the DRE are studied: ARE and iterative ARE solutions. The iterative ARE reduces the computational load dramatically by 50 times with a peak relative error of 4.8% for the Riccati equation solution and with fairly small overall fuel economy penalty. Therefore, for real-time implementation, the iterative ARE solution approximation is chosen for MPC-LQT scheme.

## CHAPTER 5: CONCLUSIONS

### 5.1 Conclusions

This dissertation presents a model predictive supervisory control strategy using linear quadratic tracking control based on the developed adaptive recursive desired torque prediction algorithm and stochastic predictive battery boundary management strategy. The conclusions are summarized as follows.

The proposed adaptive recursive prediction algorithm shows significant improvement over two exist step-by-step and fixed-gain algorithms based on the simulation results. The step-by-step prediction algorithm is not capable of real-time application due to the extremely high computational load and large prediction error. Although the fixed gain prediction algorithm reduces the computational load greatly with relatively smaller prediction error over step-by-step one, its prediction error is not low enough for practical applications. The proposed adaptive recursive prediction algorithm updates its prediction gains online and reduces the prediction error significantly with only 4% peak error under FTP and other four typical driving cycles. The introduced two weighting coefficients for the past and current pedal position signals are the key to improve the prediction accuracy and to avoid numerical over and under flow during real-time calculations. In addition, the algorithm is robust to different driver behaviors due to the regression gains updated online. With the extremely low computational load this algorithm is feasible for real-time implementation.

The proposed stochastic predictive boundary management (SPBM) strategy greatly reduces the battery over-discharging duration over the baseline power-follower and non-stochastic predictive battery boundary management strategies, where the simulation studies was conducted under typical driving cycles with different initial battery temperatures. The baseline power-follower control strategy cannot effectively prevent the battery from over-discharging, leading to reduced battery life and degraded HEV performance; and the non-stochastic predictive boundary management strategy reduces the battery over-discharging duration to a certain degree, however the proposed SPBM strategy is able to reduce the battery over-discharging duration significantly especially under aggressive driving cycles. The average over-discharge energy reduction is 82% under US06 and ARB02 driving cycles, 67% under NYCC and FTP driving cycles, and 37% under IM240 driving cycle. Most importantly, since the desired torque prediction and its error variance can be calculated online with very low computational load, the proposed SPBM strategy can be implemented for practical application.

The developed model predictive supervisory control strategy using linear quadratic tracking (LQT) control greatly improves the HEV performance over baseline power-follower strategy, where the predicted desired torque is used for LQT and stochastic predictive battery boundary management was integrated into the model predictive control (MPC). Simulation results show that the MPC-LQT strategy tracks the predicted driver torque demand, minimizes the total equivalent fuel consumption, and at the same time maintains the battery SOC close to its desired level while keeping the battery operating within the designed boundary. The fuel economy is improved significantly with exact difference Riccati equation (DRE) solution. 9.9% is achieved under FTP driving cycle, 7.3% under IM240 cycle, 3.6% under ARB02 cycle and 3.8% under US06 driving cycle. Most importantly, after studied two approximation solutions to the difference Riccati

equation (DRE) used in the LQT control: algebraic Riccati equation (ARE) and iterative ARE solutions. It is found that compared to exact DRE the iterative ARE solution reduces the computational load dramatically by 50 times with only 4.8% peak relative error and however, the overall fuel economy penalty is very small. For real-time implementation, it is recommended to use the iterative ARE approximation solution for the MPC-LQT control scheme.

## **5.2 Recommendation for future works**

The research work in this dissertation is still in its early stage. There are several technical and practical issues deserving further investigations:

- 1) In this dissertation, within the finite prediction horizon, both the battery discharging/charging limits and current SOC are assumed to be constants. However, as stated in chapters 2 and 3, the battery capacity and internal resistance are functions of temperature. The battery discharging/charging limits and SOC shall be varied within the prediction horizon.
- 2) The maximum prediction horizon is confined to 0.5s. It might not be enough for various driving conditions. For instance, if the vehicle speed is 50 mph, 0.5s prediction horizon only lasts 11.2 m in distance.
- 3) Road gradient is a very important factor that affects the fuel economy, it should be considered in future works.
- 4) All the algorithms developed in this dissertation are for real-time implementation. However, they are only validated in the MATLAB/Simulink simulations, real-time implementation is recommend for future investigations.

## **APPENDIX**

## APPENDIX

### APPENDIX A: Key parameters of the HEV

Table A. 1. Key parameters of the HEV

Item	Parameter	Value	Item	Parameter	Value
air_density	$\rho$	1.23 kg/m <sup>3</sup>	vc_idle_spd		80 rad/s
cs_eng_on_soc		0.5	veh_FA	$A_f$	3.12 m <sup>2</sup>
cs_hi_soc		0.75	veh_cargo_mass	$m_c$	0
cs_lo_soc		0.45	veh_CD	$c_d$	0.37
cs_target_soc		0.60	veh_gravity	$g$	9.8 m/s <sup>2</sup>
ess_init_soc		0.6	wheel_mass	$m_w$	50 kg
cs_min_off_time		3s	wheel_radius	$r_w$	0.3 m
cs_min_on_time		10s			
cyc_grade		0			
ess_module_num		40			
ess_module_mass		0.9979 kg			
fd_ratio	$f_d$	3.2667			
fc_fuel_den	$\rho_f$	749 kg/m <sup>3</sup>			
tx_pg_r	$R$	86			
tx_pg_s	$S$	44			

## APPENDIX B: HEV MATLAB/Simulink models

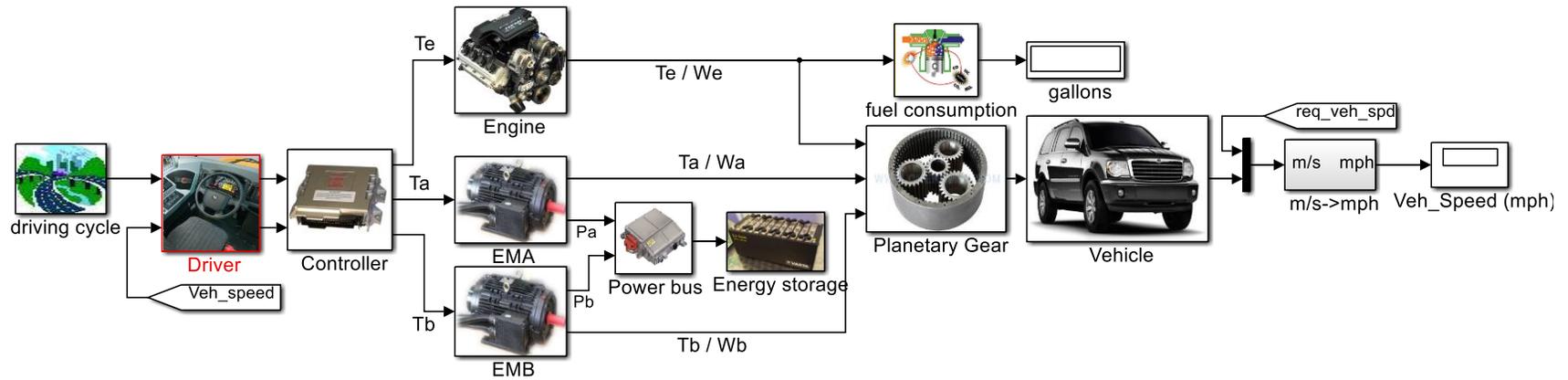


Figure B.1 HEV model

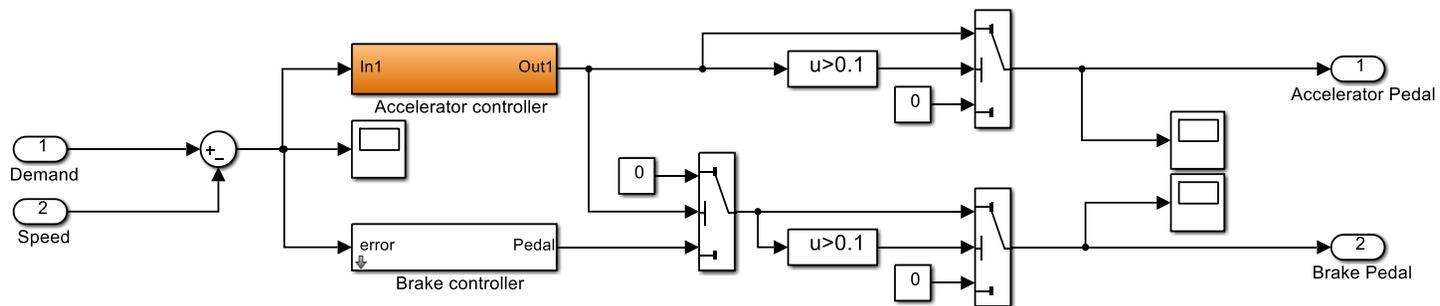


Figure B.2 HEV driver model

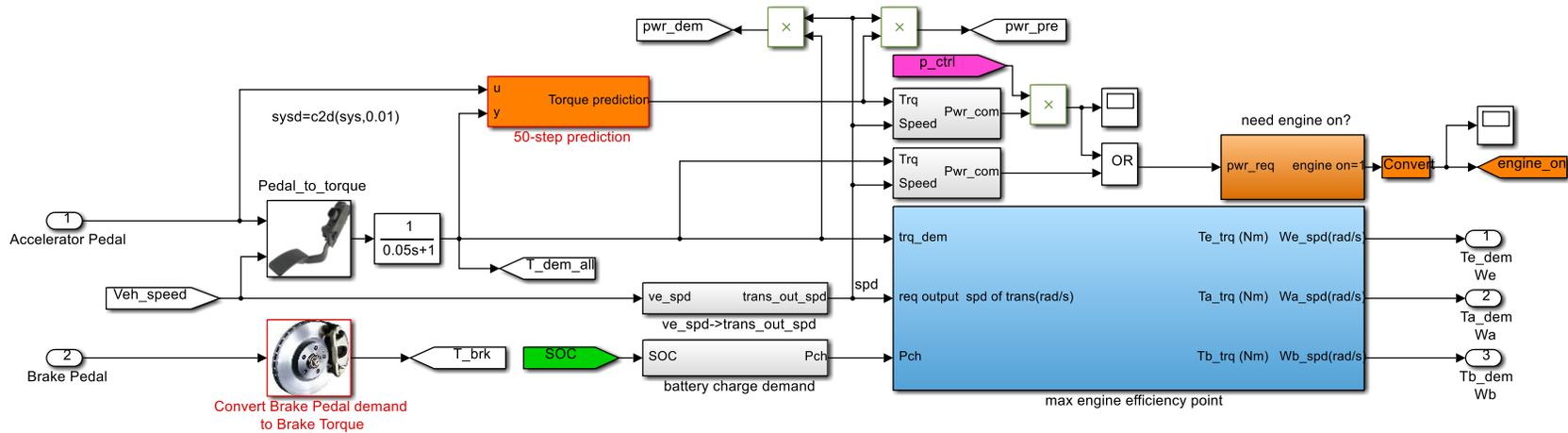


Figure B.3 HEV supervisory controller model

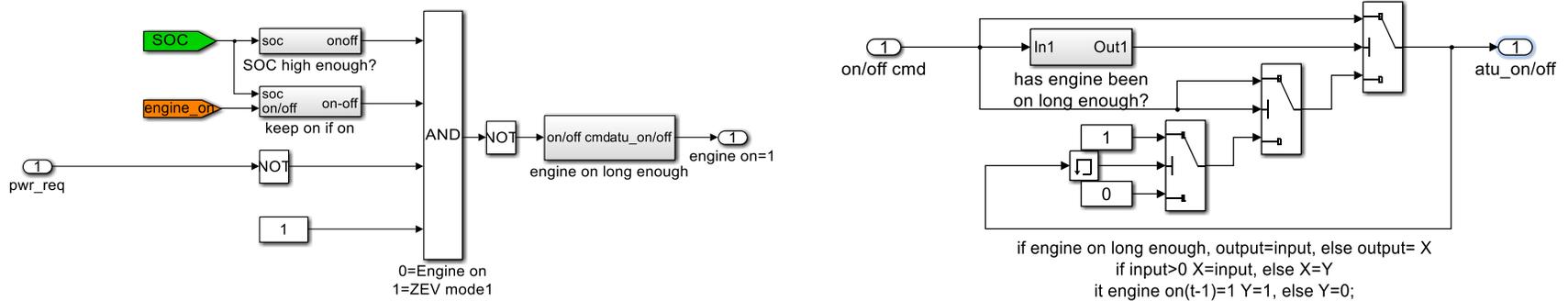


Figure B.4 HEV engine on/off control model

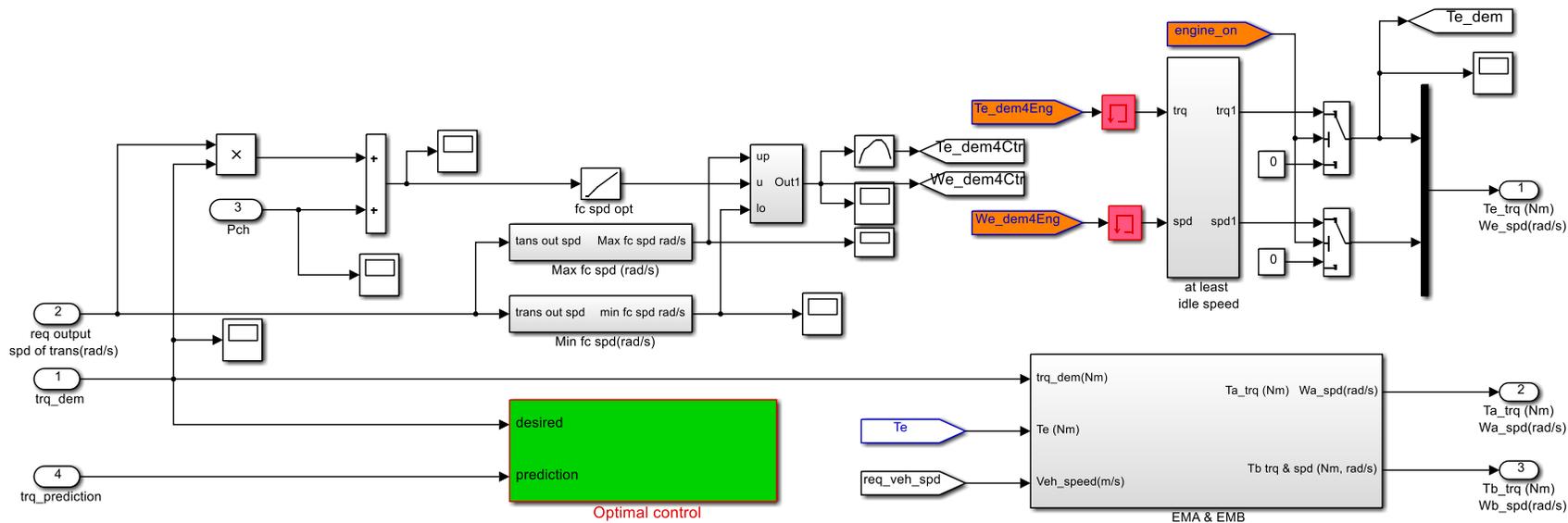


Figure B.5 HEV supervisory power distribution model

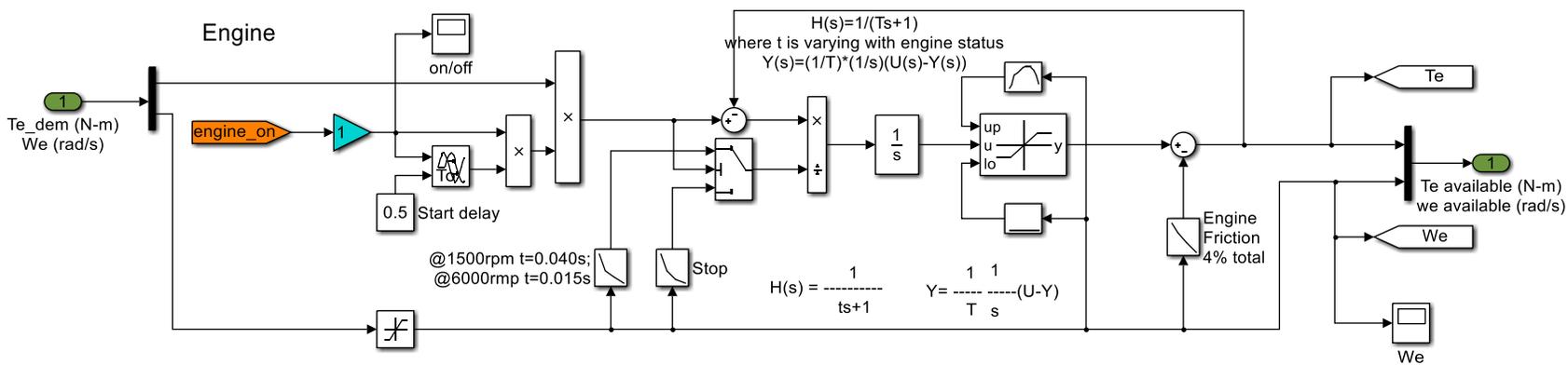


Figure B.6 HEV engine model

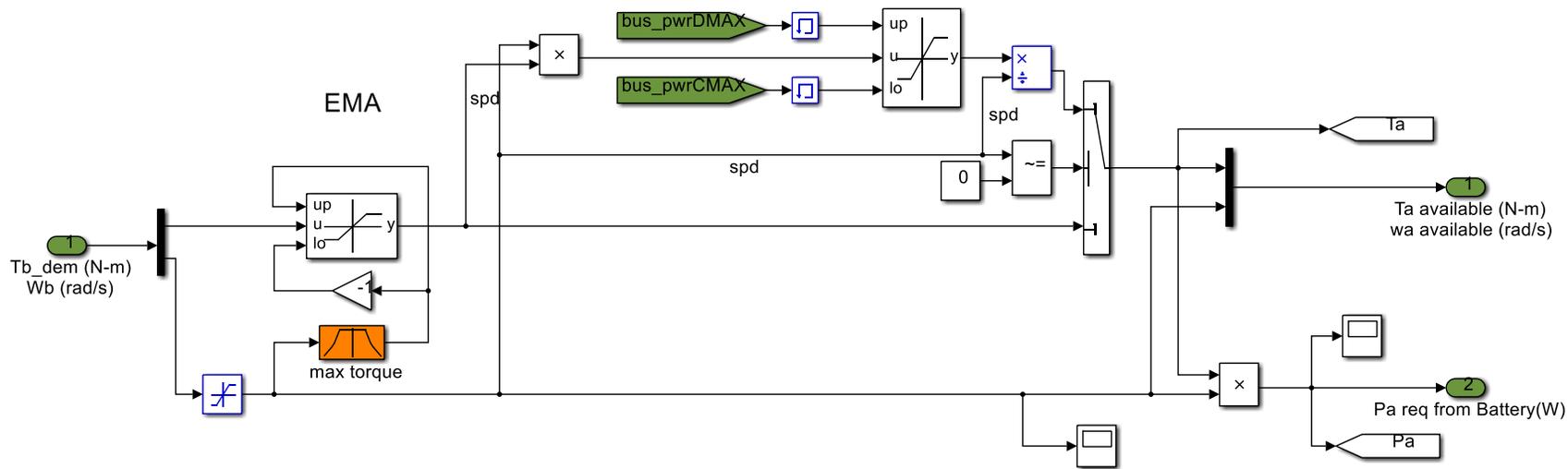


Figure B. 7 HEV EMA model

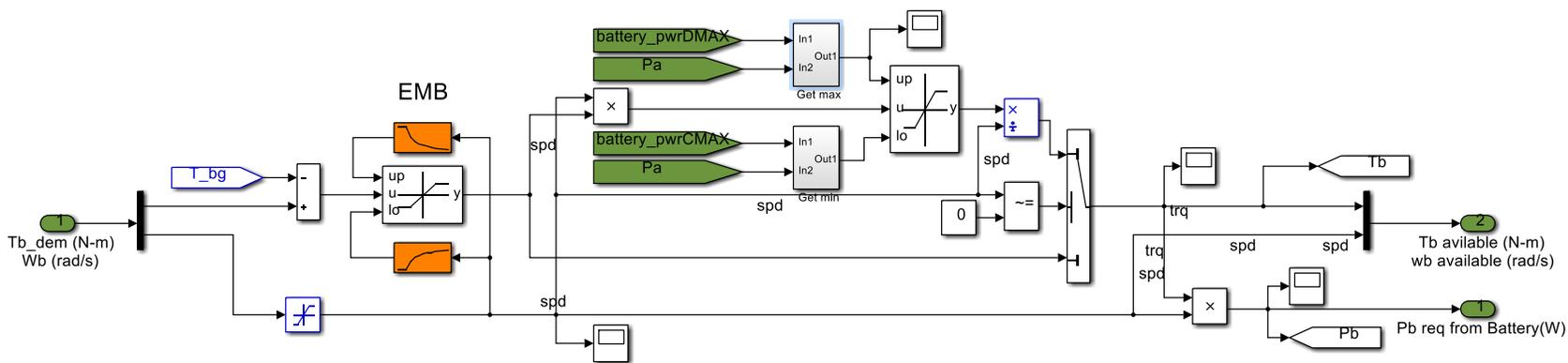


Figure B. 8 HEV EMB model

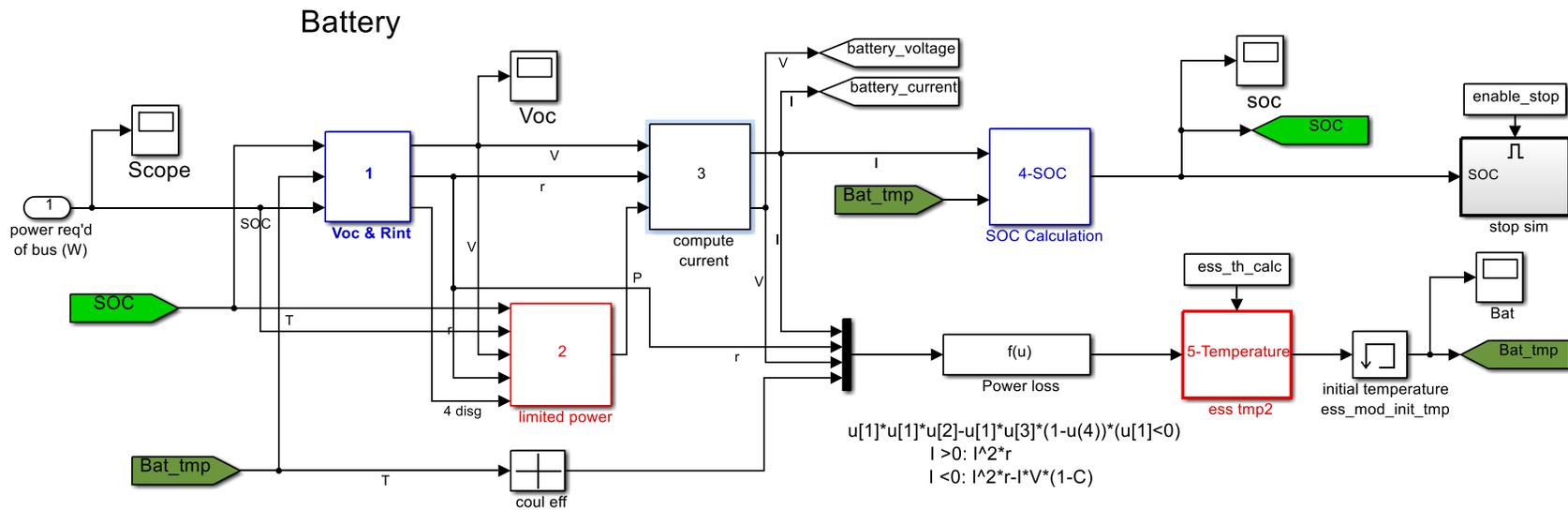
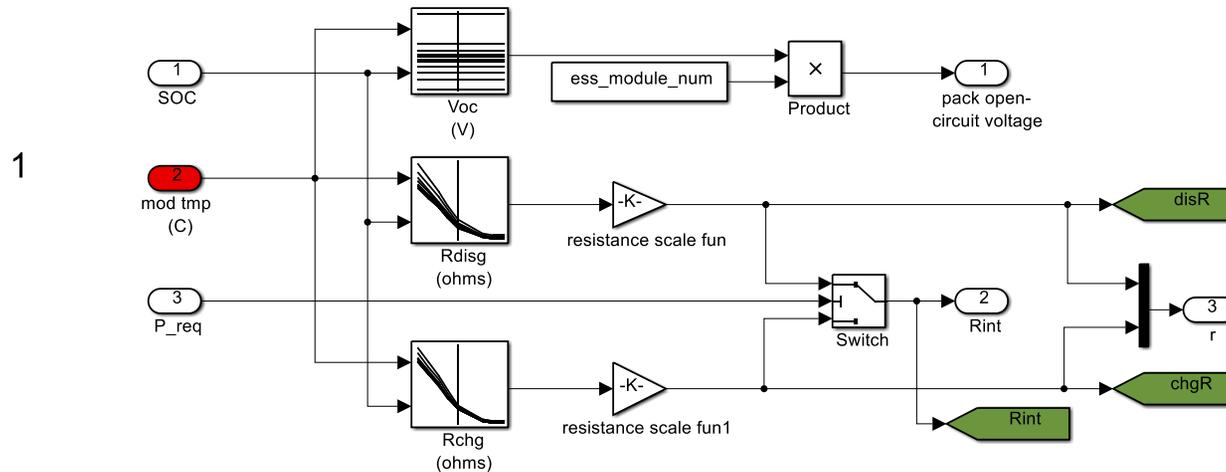


Figure B. 9 HEV battery model



1. Pack Open Circuit Voltage and Internal Resistance:  
 This block calculates Voc and Rint given the current SOC and required battery power

Figure B. 10 HEV battery VOC and Rint model

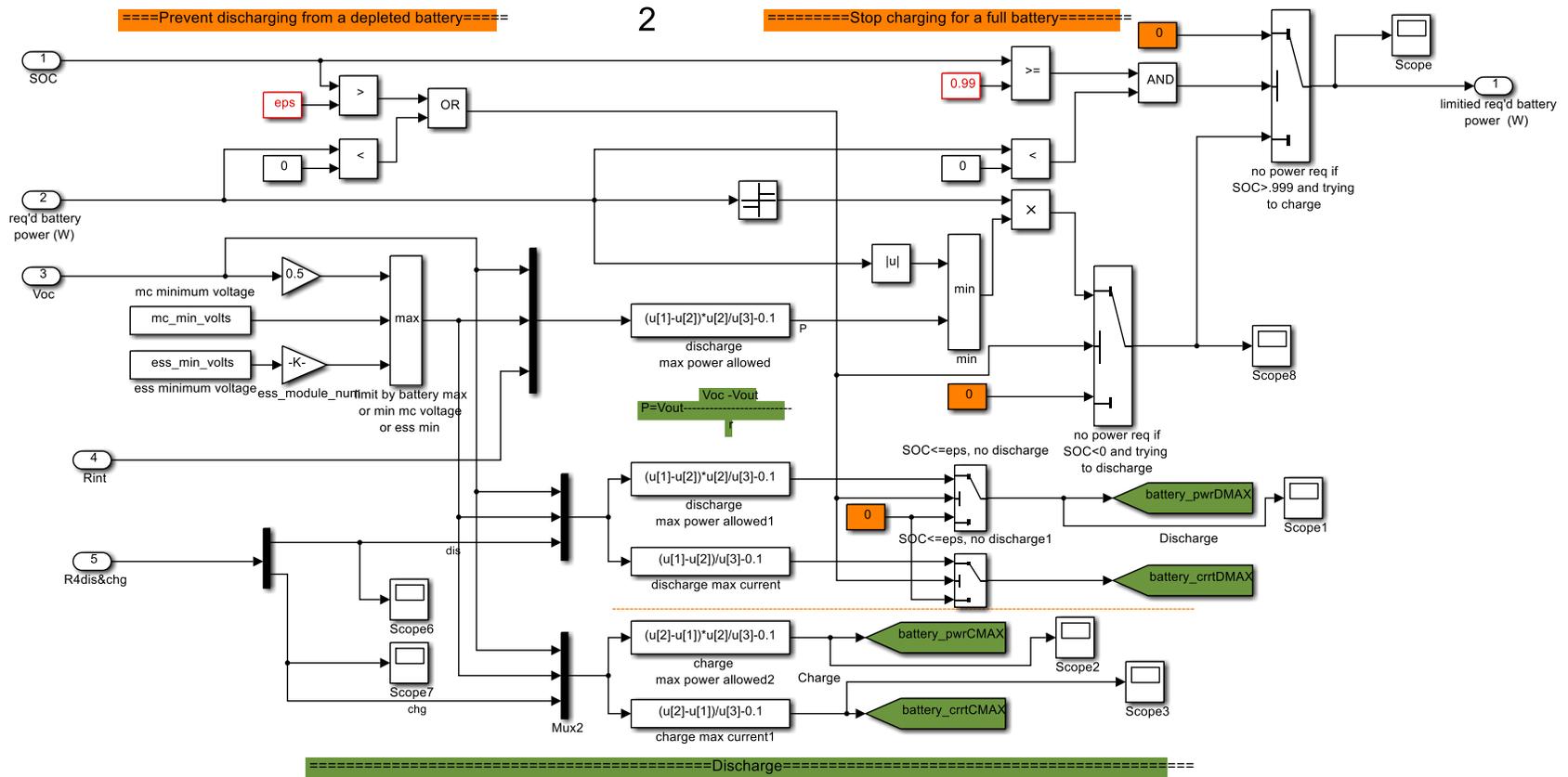


Figure B. 11 HEV battery limited power model

3

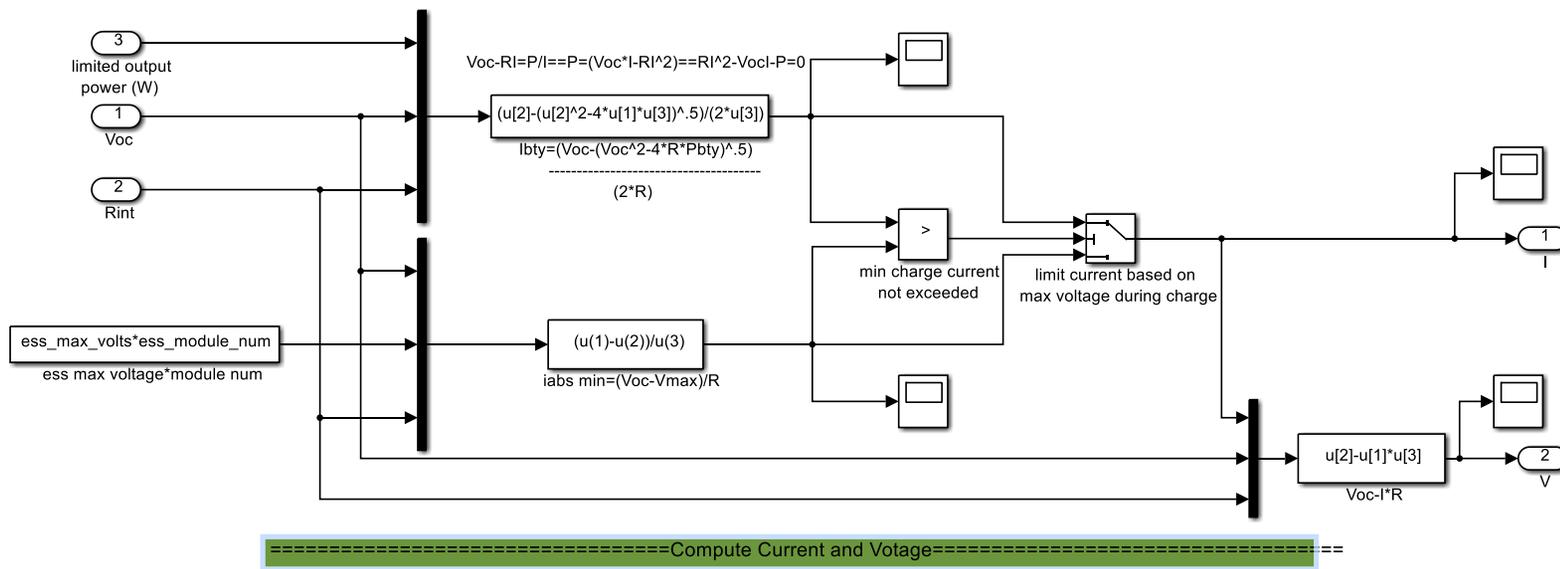


Figure B. 12 HEV battery output current model

4

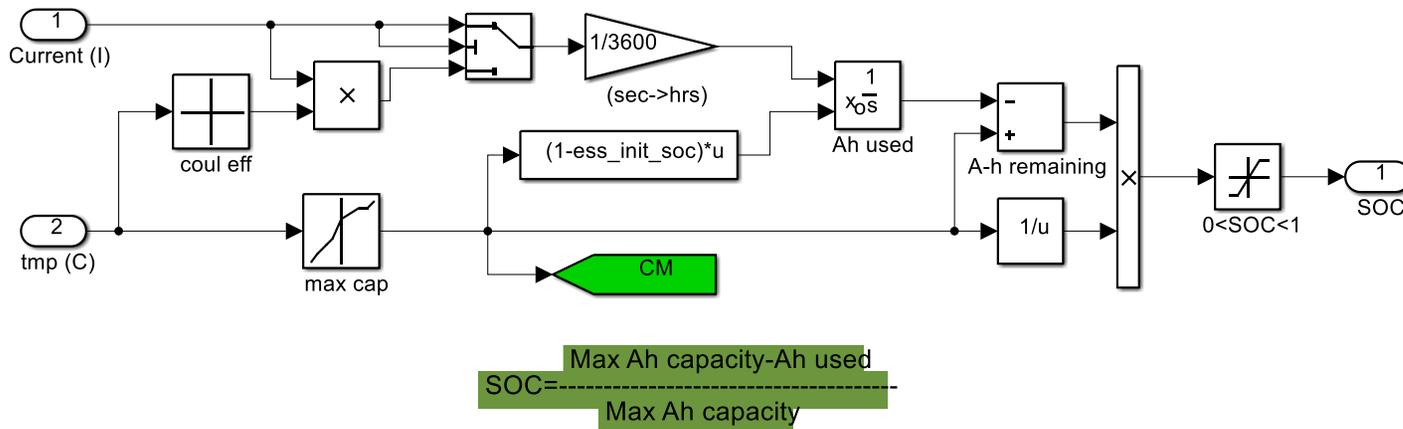


Figure B. 13 HEV battery SOC model

5

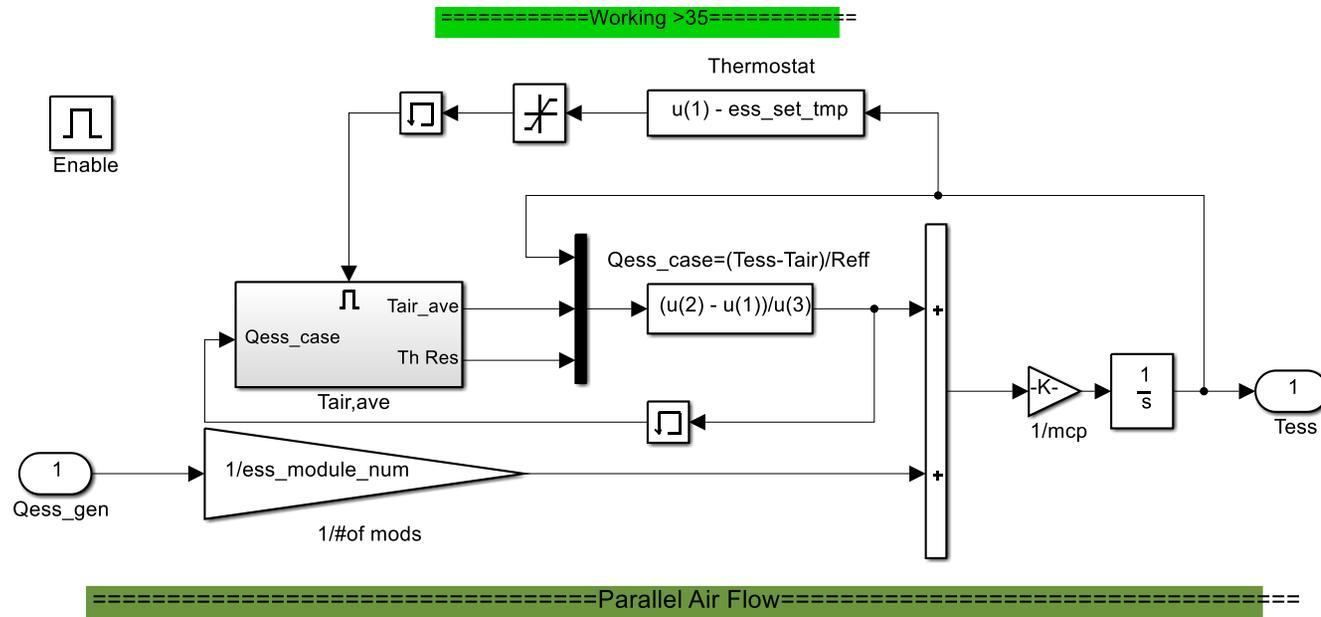


Figure B. 14 HEV battery air flow/temperature model

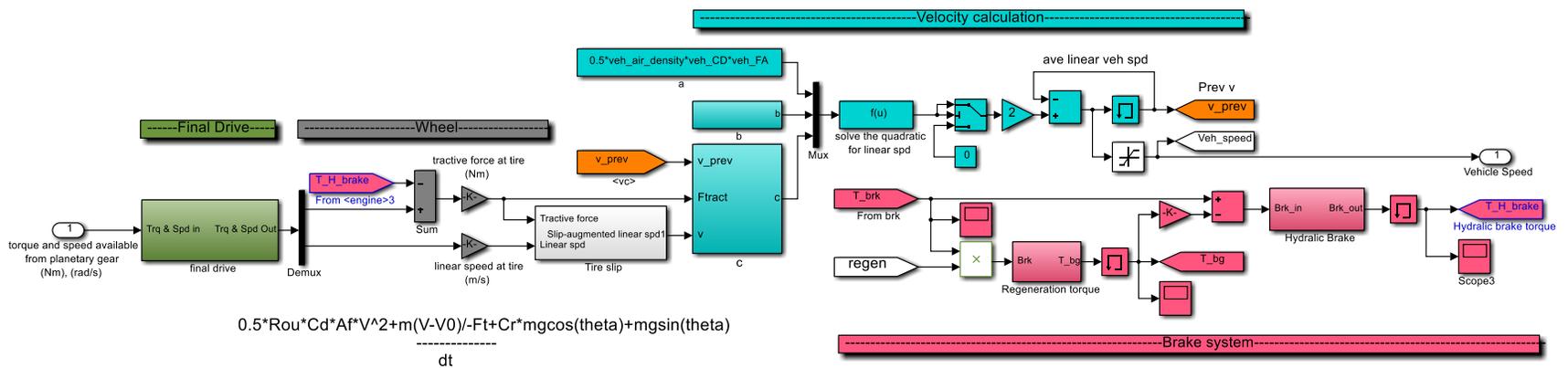


Figure B. 15 HEV vehicle dynamics model

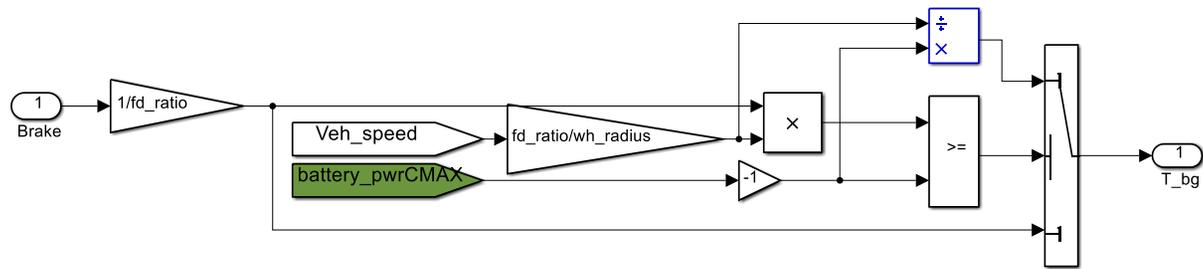


Figure B. 16 HEV braking regeneration model

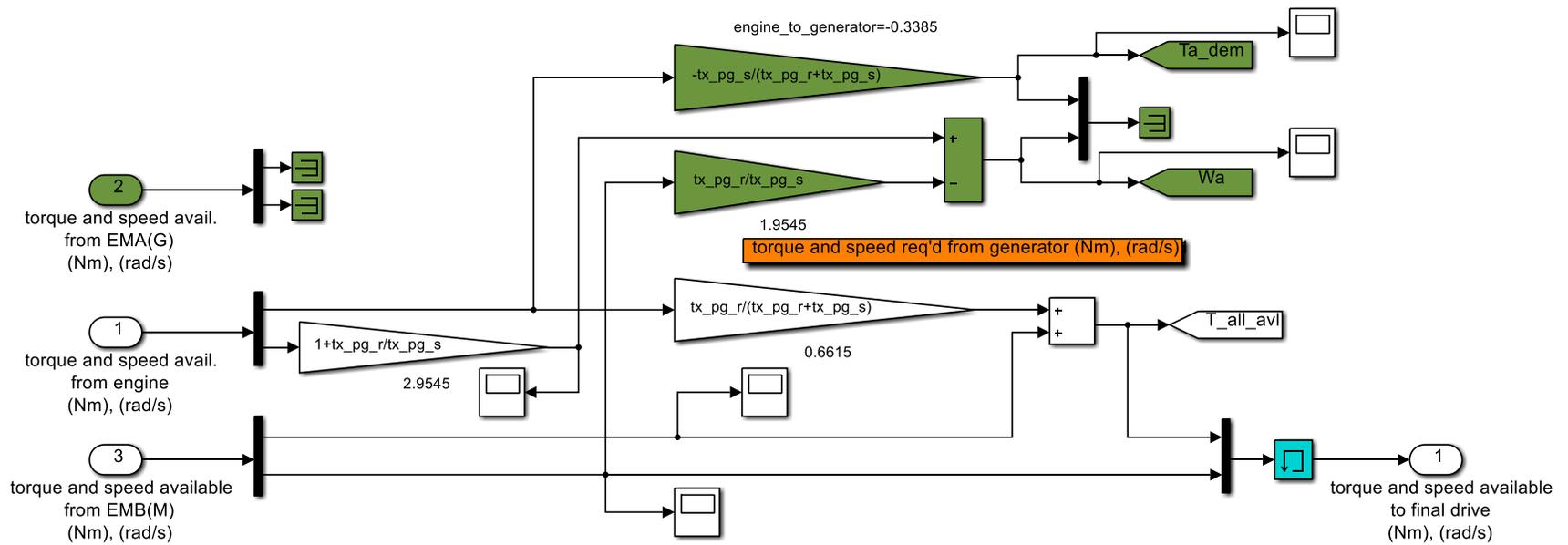


Figure B. 17 HEV planetary gear model

## APPENDIX C: System functions

### C.1 Adaptive recursive prediction algorithm

```
% This m code is based on s-function
steps=50;% 50-step prediction;
Beta1=20;% weighting factor for current;

y_data_length=3; % desired torque
u_data_length=y_data_length+1; % pedal position

ne=4*(y_data_length+u_data_length); % number of equations for Least-Squares
solution.
sda=y_data_length+1+steps+ne; % total sampled data

%=====
I=eye(2*y_data_length+1,2*y_data_length+1);

%=====initialize the value of u_input and y_input=====

    yin=block.InputPort(1).Data;
    y_input=block.Dwork(1).Data;
    y_input(1,:)=[];
    y_input=[y_input;yin];
    block.Dwork(1).Data=y_input;

    uin=block.InputPort(2).Data;
    u_input=block.Dwork(2).Data;
    u_input(1,:)=[];
    u_input=[u_input;uin];
    block.Dwork(2).Data=u_input;

y_input1=y_input(sda-y_data_length:sda-1,1);% y data vector(desired torque);
u_input1=u_input(sda-u_data_length+1:sda,1); % up data vector(pedal
position);
phiT=[y_input1;u_input1];

y_input2=y_input(sda-y_data_length:sda-1,1);% y data vector;
u_input2=u_input(sda-u_data_length+1:sda,1); % u data vector;

phiT1=[y_input2;u_input2];

%====let bk=Dwork(4)=====
bk=block.Dwork(4).Data;% current coefficients (denoted as big theta in
dissertation)
%=====

%====let pk=Dwork(5)=====
pp=block.Dwork(5).Data; % current recursive P in dissertation
for j=1:(2*y_data_length+1)
    for i=1:(2*y_data_length+1)
        pk(i,j)=pp(i+(j-1)*(2*y_data_length+1));
    end
end
end
```

```

%%=====calculation for trace=====
ppt_lmt = 2;
error_max=30;
Alpha1=block.Dwork(7).Data;% weighting factor for past data

%=====gradient=====
target_F_norm=400;%block.Dwork(8).Data; %old
F_norm=block.Dwork(6).Data; %new
error1=(target_F_norm-F_norm);

norm0=block.Dwork(9).Data;
norm0(1,:)=[];
norm0=[norm0;error1];
block.Dwork(9).Data=norm0;
norm_error=mean(norm0);

% block.Dwork(8).Data=block.Dwork(6).Data;

if norm_error<-error_max
    norm_error=-error_max;
elseif norm_error>error_max
    norm_error=error_max;
end
ppt=norm_error;

%=====Conditions for update=====Conditions for update=====
if uin<=0.1 && ppt<=1*ppt_lmt && ppt>=-1*ppt_lmt %If pedal position is less
than 1%, coefficient doesn't need to be updated.
    bk1=bk;
    pk1=pk;
    Alpha1=1;
else
    Alpha1=Alpha1-(5.61e-006)*ppt; % adaptive scheme

    ratio=(Beta1/Alpha1)^2;
    bk1=(1-pk*phiT*phiT'/(phiT'*pk*phiT+1/ratio))*(bk+ratio*pk*phiT*yin); %
yin is the current new desired torque data
    pk01=(1/Alpha1^2)*pk-(1/Alpha1^2)*(phiT'*pk)'*(phiT'*pk)/
(phiT'*pk*phiT+1/ratio);
    pk1=0.5*(pk01+pk01');
end

F_norm=norm(pk1,'fro');

yp=phiT1'*bk1;% for output

block.Dwork(3).Data=yp; % prediction
block.Dwork(4).Data=bk1; % regressive gains
block.Dwork(5).Data=ppl; % P
block.Dwork(6).Data=F_norm; % Norm
block.Dwork(7).Data=Alpha1; % weighting factor for past

```

## C.2 Prediction error variance algorithm

```
% This a s-function based m code
dimension=512;
yin=(block.InputPort(1).Data)^2;
y_input=block.Dwork(1).Data;
y_input(1,:)=[];
y_input=[y_input;yin];
block.Dwork(1).Data=y_input;

sigma=sqrt(sum(y_input)/dimension);% traditional calculation
```

## C.3 MPC-LQT control algorithm

```
% this m code is s-function based.

Pre_in=block.InputPort(1).Data;
Pre_input=block.Dwork(1).Data;
Pre_input(1,:)=[];
Pre_input=[Pre_input;Pre_in];
block.Dwork(1).Data=Pre_input;

Te0=block.InputPort(2).Data(1);
we0=block.InputPort(2).Data(2);
TA0=block.InputPort(2).Data(3);
wA0=block.InputPort(2).Data(4);
TB0=block.InputPort(2).Data(5);
wB0=block.InputPort(2).Data(6);

PbatMAX=block.InputPort(3).Data(1);
PbatMIN=block.InputPort(3).Data(2);
Dis_Chg=block.InputPort(3).Data(3);
Cbat=block.InputPort(3).Data(4);
Voc=block.InputPort(3).Data(5);
Rint=block.InputPort(3).Data(6);
CM=block.InputPort(3).Data(7);
SOC0=block.InputPort(3).Data(8);
Tbg0=block.InputPort(3).Data(9);

fuel_m=block.InputPort(4).Data;
Tdes0=block.InputPort(5).Data(1);
engineon=block.InputPort(5).Data(2);
flowrate=block.InputPort(4).Data;
equalflowrate=block.InputPort(5).Data;

LowHeatingValue=54;
gS=44;gR=86; % gear numbers
% Matlab for discrete-time tracking system

speedlimit=0.001;
taoe=10;% Making motors 1000 times faster than IC engine.
yitae=42;
AC11=-1/taoe;
AC12=0;AC13=0;AC14=0;AC15=0;AC16=0;

taoA=0.015;
```

```

AC21=-1/taoA*gS/(gS+gR);
AC22=-1/taoA;
AC23=0;AC24=0;AC25=0;AC26=0;

taoB=0.015;
if abs(wB0)<=speedlimit
    AC31=0;
else
    AC31=-1/taoB*(gR/(gR+gS)-we0/wB0);
end
AC32=0;
AC33=-1/taoB;
AC34=0;AC35=0;AC36=0;

mveh=1750;JA=0.1;JB=0.1;Je=0.1;rw=0.3876;fd=3.2667;

E11=Je+JA*( (gS+gR)/gS)^2;
E12=-JA*gR*(gS+gR)/gS/gS;
E22=JB+mveh*(rw/fd)^2+JA*(gR/gS)^2;
ED=E11*E22-E12*E12;

AC41=E22/ED;
AC42=(E22*(gS+gR)+E12*gR)/(gS*ED);
AC43=-E12/ED; AC44=0;
AC45=E12*1.2*0.37*3.12*rw^3*wB0/ED/fd/fd/fd;
AC46=0;

AC51=-E12/ED;AC52=-(E12*(gS+gR)+E11*gR)/gS/ED;
AC53=E11/ED;AC54=0;
AC55=-E11*1.2*0.37*3.12*rw^3*wB0/ED/fd/fd/fd;
AC56=0;

% [mm,nn]=size(Pre_input);
% Tdes=Pre_input(mm,1);

% demB=-1/CM/sqrt(Voc*Voc-4*Rint*(TA0*wA0+TB0*wB0))
% demA=-1/CM/sqrt(Voc*Voc-4*Rint*(TA0*wA0+TB0*wB0))
% demAB=-1/CM/sqrt(Voc*Voc-4*Rint*(TA0*(we0*(gS+gR)-wB0*gR)/gS+TB0*wB0));
% AC61=0;
% AC62=demAB*(we0*(gS+gR)-wB0*gR)/gS;
% AC63=demAB*wB0;
% AC64=demAB*TA0*(gR+gS)/gS;
% AC65=demAB*(TB0-TA0*gR/gS);
pbat=Tdes0*wB0-Te0*we0-Tbg0;
if pbat>=PbatMAX
    pbat=PbatMAX;
elseif pbat<=PbatMIN
    pbat=PbatMIN;
else
end
dem=1/CM/sqrt(Voc*Voc-4*Rint*pbat);
AC61=dem*we0;
AC62=0;AC63=0;
AC64=dem*Te0;
AC65=-dem*Tdes0;

```

```

AC66=0;

MatrixAC=[AC11,AC12,AC13,AC14,AC15,AC16;...
          AC21,AC22,AC23,AC24,AC25,AC26;...
          AC31,AC32,AC33,AC34,AC35,AC36;...
          AC41,AC42,AC43,AC44,AC45,AC46;...
          AC51,AC52,AC53,AC54,AC55,AC56;...
          AC61,AC62,AC63,AC64,AC65,AC66];
% MatrixA=exp(MatrixAC*0.01)
MatrixA=eye(6)+MatrixAC*0.01;

if abs(we0)<=79 || engineon==0
    BC11=0;
else
    % BC11=fuel_m;
    BC11=1/taoe*yitae*LowHeatingValue/we0;
end
BC12=0;BC21=0;BC22=0;BC31=0;
if wB0==0
    BC32=0;
else
    BC32=1/taoB/Cbat/wB0*(1-2*Rint/Voc/Voc/Cbat);
end
BC41=0;BC42=0;BC51=0;BC52=0;BC61=0;BC62=0;

MatrixBC=[BC11,BC12;BC21,BC22;BC31 BC32;BC41 BC42;BC51 BC52;BC61 BC62];
% MatrixB0=exp(MatrixAC*0.01)*MatrixBC*0.01
% MatrixB=MatrixBC*0.01;
MatrixB=pinv(MatrixAC)*(exp(MatrixAC*0.01)-eye(6))*MatrixBC;
MatrixC=[0 -gR/gS 1 0 0 0];

% state weighting matrix Q
MatrixQ=1; % performance index
MatrixR=[35,0;0,300]; % performance index control
MatrixF=1; % performance index weighting matrix F
maxsteps=5;

%% Times4SOC=2000;
x1(1)=Te0; %% initial condition on state x1
x2(1)=TA0; %% initial condition on state x2
x3(1)=TB0; %% initial condition on state x3
x4(1)=we0; %% initial condition on state x4
x5(1)=wB0; %% initial condition on state x5
x6(1)=SOC0;

xk=[x1(1);x2(1);x3(1);x4(1);x5(1);x6(1)];

% note that if kf =10 then
% k=[k0, kf]=[0 1 2 3 ,...,10]
% then we have 11 points and an array x1 should have subscript
% x1(N) with N=1 to 11. This is because x(0) is illegal in array
% definition in Matlab. let us use N=kf+1
k0=0; % the initial instant k_0]
kf=maxsteps-1; % the final instant k_f
N=kf+1; %
    for i=1:1:N-1

```

```

        zk(i)=Pre_input(i);
    end
    zkf=Pre_input(maxsteps);
    [n,n]=size(MatrixA); % fixing the order of the system matrix A
    I=eye(n); % identity matrix I
    E=MatrixB*inv(MatrixR)*MatrixB'; % the matrix E=BR^(-1)B'
    V=MatrixC'*MatrixQ*MatrixC;
    W=MatrixC'*MatrixQ;
    % solve matrix difference Riccati equation
    % backwards starting from kf to k0
    % use the form P(k)=A'P(k+1)[I+EP(k+1)]^(-1)A+V
    % first fix the final condition S P(k_f)=F;
    % g(k_f)= C'FZ(k_f)
    % note that P, Q, R, F are all symmetric ij=ji
    pkplus1=MatrixC'*MatrixF*MatrixC;
    gkplus1=MatrixC'*MatrixF*zkf;

    p11(N)=pkplus1(1);p12(N)=pkplus1(2);p13(N)=pkplus1(3);
    p14(N)=pkplus1(4);p15(N)=pkplus1(5);p16(N)=pkplus1(6);
    p21(N)=pkplus1(7);p22(N)=pkplus1(8);p23(N)=pkplus1(9);
    p24(N)=pkplus1(10);p25(N)=pkplus1(11);p26(N)=pkplus1(12);
    p31(N)=pkplus1(13);p32(N)=pkplus1(14);p33(N)=pkplus1(15);
    p34(N)=pkplus1(16);p35(N)=pkplus1(17);p36(N)=pkplus1(18);
    p41(N)=pkplus1(19);p42(N)=pkplus1(20);p43(N)=pkplus1(21);
    p44(N)=pkplus1(22);p45(N)=pkplus1(23);p46(N)=pkplus1(24);
    p51(N)=pkplus1(25);p52(N)=pkplus1(26);p53(N)=pkplus1(27);
    p54(N)=pkplus1(28);p55(N)=pkplus1(29);p56(N)=pkplus1(30);
    p61(N)=pkplus1(31);p62(N)=pkplus1(32);p63(N)=pkplus1(33);
    p64(N)=pkplus1(34);p65(N)=pkplus1(35);p66(N)=pkplus1(36);
    %
    g1(N)=gkplus1(1);g2(N)=gkplus1(2);g3(N)=gkplus1(3);
    g4(N)=gkplus1(4);g5(N)=gkplus1(5);g6(N)=gkplus1(6);
    %
    for k=N-1:-1:1,
        Pk=MatrixA'*pkplus1*inv(I+E*pkplus1)*MatrixA+V;
        Lk=pinv(MatrixR+MatrixB'*pkplus1*MatrixB)*MatrixB'*pkplus1*MatrixA;
        gk=(MatrixA-MatrixB*Lk) '*gkplus1+W*zk(k);
        p11(k)=Pk(1,1);    p12(k)=Pk(1,2);    p13(k)=Pk(1,3);    p14(k)=Pk(1,4);
        p15(k)=Pk(1,5);    p16(k)=Pk(1,6);
        p21(k)=Pk(2,1);    p22(k)=Pk(2,2);    p23(k)=Pk(2,3);    p24(k)=Pk(2,4);
        p25(k)=Pk(2,5);    p26(k)=Pk(2,6);
        p31(k)=Pk(3,1);    p32(k)=Pk(3,2);    p33(k)=Pk(3,3);    p34(k)=Pk(3,4);
        p35(k)=Pk(3,5);    p36(k)=Pk(3,6);
        p41(k)=Pk(4,1);    p42(k)=Pk(4,2);    p43(k)=Pk(4,3);    p44(k)=Pk(4,4);
        p45(k)=Pk(4,5);    p46(k)=Pk(4,6);
        p51(k)=Pk(5,1);    p52(k)=Pk(5,2);    p53(k)=Pk(5,3);    p54(k)=Pk(5,4);
        p55(k)=Pk(5,5);    p56(k)=Pk(5,6);
        p61(k)=Pk(6,1);    p62(k)=Pk(6,2);    p63(k)=Pk(6,3);    p64(k)=Pk(6,4);
        p65(k)=Pk(6,5);    p66(k)=Pk(6,6);

        pkplus1=Pk;
        %
        g1(k)=gk(1);    g2(k)=gk(2);    g3(k)=gk(3);    g4(k)=gk(4);
        g5(k)=gk(5);    g6(k)=gk(6);

    end
    gkplus1=gk;
end

```

```

% calculate the feedback coefficients L and Lg(k)
%  $L(k) = (R+B'P(k+1)B)^{-1}BP(k+1)A$ 
%  $Lg(k) = [R+B'P(k+1)B]^{-1}B'$ 
%
for k=N-1:-1:1,
    Pk=[p11(k),p12(k),p13(k),p14(k),p15(k),p16(k);...
        p21(k),p22(k),p23(k),p24(k),p25(k),p26(k);...
        p31(k),p32(k),p33(k),p34(k),p35(k),p36(k);...
        p41(k),p42(k),p43(k),p44(k),p45(k),p46(k);...
        p51(k),p52(k),p53(k),p54(k),p55(k),p56(k);...
        p61(k),p62(k),p63(k),p64(k),p65(k),p66(k)];
    gk=[g1(k);g2(k);g3(k);g4(k);g5(k);g6(k)];
    Lk=pinv(MatrixR+MatrixB'*pkplus1*MatrixB)*MatrixB'*pkplus1*MatrixA1;
    Lgk=pinv(MatrixR+MatrixB'*pkplus1*MatrixB)*MatrixB';

    l11(k)=Lk(1,1);    l12(k)=Lk(1,2);    l13(k)=Lk(1,3);    l14(k)=Lk(1,4);
    l15(k)=Lk(1,5);    l16(k)=Lk(1,6);
    l21(k)=Lk(2,1);    l22(k)=Lk(2,2);    l23(k)=Lk(2,3);    l24(k)=Lk(2,4);
    l25(k)=Lk(2,5);    l26(k)=Lk(2,6);

    lg11(k)=Lgk(1,1);    lg12(k)=Lgk(1,2);    lg13(k)=Lgk(1,3);
    lg14(k)=Lgk(1,4);    lg15(k)=Lgk(1,5);    lg16(k)=Lgk(1,6);
    lg21(k)=Lgk(2,1);    lg22(k)=Lgk(2,2);    lg23(k)=Lgk(2,3);
    lg24(k)=Lgk(2,4);    lg25(k)=Lgk(2,5);    lg26(k)=Lgk(2,6);

    pkplus1=Pk;
    gkplus1=gk;
end

for k=1:N-2,
    Lk=[l11(k),l12(k),l13(k),l14(k),l15(k),l16(k);...
        l21(k),l22(k),l23(k),l24(k),l25(k),l26(k)];
    Lgk=[lg11(k),lg12(k),lg13(k),lg14(k),lg15(k),lg16(k);...
        lg21(k),lg22(k),lg23(k),lg24(k),lg25(k),lg26(k)];
    Lgkplus1=[lg11(k+1),lg12(k+1),lg13(k+1),lg14(k+1),lg15(k+1),lg16(k+1);...
        lg21(k+1),lg22(k+1),lg23(k+1),lg24(k+1),lg25(k+1),lg26(k+1)];
    xk=[x1(k);x2(k);x3(k);x4(k);x5(k);x6(k)];
    xkplus1=(MatrixA1-MatrixB*Lk)*xk+MatrixB*Lgkplus1*gk;
    x1(k+1)=xkplus1(1);
    x2(k+1)=xkplus1(2);
    x3(k+1)=xkplus1(3);
    x4(k+1)=xkplus1(4);
    x5(k+1)=xkplus1(5);
    x6(k+1)=xkplus1(6);
end
%
% solve for optimal control
%  $u(k) = -L(k)x(k) + Lg(k)g(k+1)$ 
% for k=1:N
for k=1:N-1,
    Lk=[l11(k),l12(k),l13(k),l14(k),l15(k),l16(k);...
        l21(k),l22(k),l23(k),l24(k),l25(k),l26(k)];
    Lgk=[lg11(k),lg12(k),lg13(k),lg14(k),lg15(k),lg16(k);...
        lg21(k),lg22(k),lg23(k),lg24(k),lg25(k),lg26(k)];
    gkplus1=[g1(k+1);g2(k+1);g3(k+1);g4(k+1);g5(k+1);g6(k+1)];
    xk=[x1(k);x2(k);x3(k);x4(k);x5(k);x6(k)];

```

```

    uk=-Lk*xk+Lgk*gkplus1;
    uk1(k)=uk(1);
    uk2(k)=uk(2);
end
%
block.Dwork(2).Data=uk1(1);
block.Dwork(3).Data=uk2(1);
block.Dwork(6).Data(1)=x1(1);% x1: Te
block.Dwork(6).Data(2)=x1(2);% x1: Te

block.Dwork(7).Data(1)=x2(1);% x2: TA
block.Dwork(7).Data(2)=x2(2);% x2: TA

block.Dwork(8).Data(1)=x3(1);% x3: TB
block.Dwork(8).Data(2)=x3(2);% x3: TB

block.Dwork(9).Data(1)=x4(1);% x4: we
block.Dwork(9).Data(2)=x4(2);% x4: we

block.Dwork(10).Data(1)=x5(1);% x5:wB
block.Dwork(10).Data(2)=x5(2);% x5:wB

block.Dwork(11).Data(1)=x6(1);% x6:SOC
block.Dwork(11).Data(2)=x6(2);% x6:SOC

```

## **BIBLIOGRAPHY**

## BIBLIOGRAPHY

- [1] A. Sciarretta, and L. Guzzella, "Control of hybrid electric vehicles," *IEEE control systems*, vol. 27, no. 2, April, 2007, pp. 60-70.
- [2] M. Debert, G. Colin, Y. Chamaillard, L. Guzzella, A. Ketfi-Cherif and B. Bellicaud, "Predictive energy management for hybrid electric vehicles - Prediction horizon and battery capacity sensitivity," *IFAC Symposium on Advances in Automotive Control*, Munich, Germany, July, 2010, pp.270-275.
- [3] S. K. Tae, C. Manzie, and R. Sharma, "Model predictive control of velocity and torque split in a parallel hybrid vehicle," *IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, USA, Oct. 2009, pp. 2014-2019.
- [4] Z. X. Fu, "Real-time prediction of torque availability of an IPM synchronous machine drive for hybrid electric vehicles," In *Proceedings of the IEEE International Conference on Electric Machines and Drives*, San Antonio, Texas, USA, May, 2005, pp. 199–206.
- [5] N. Shidore, and T. Bohn, "Evaluation of cold temperature performance of the JCS-VL41M PHEV battery using battery HIL," *SAE technical paper 2008-01-1333*, Detroit, April, 2008.
- [6] E. Samadani, J. Lo, M. Fowler, R. Fraser, et al., "Impact of temperature on the A123 Li-Ion battery performance and hybrid electric vehicle range," *SAE technical paper 2013-01-1521*, Detroit, April, 2013.
- [7] A. Pesaran, "Battery thermal management in EVs and HEVs: issues and solutions," *Advanced Automotive Battery Conference*, Las Vegas, NV, USA, Feb. 2001.
- [8] M. Back, M. Simons, F. Kirschaum, and V. Krebs, "Predictive control of drivetrains," *IFAC 15<sup>th</sup> triennial world congress*, Barcelona Spain, 2002, pp. 1506-1506.
- [9] M. Koot, B. de Jager, J. Kessels, W. Heemels, and P. van den Bosch, "Energy management strategies for vehicle power nets," *Proc. Of the 2004 American control conference*, Boston, USA, vol. 5, Jun-July, 2004, pp. 4072-4077.

- [10] M. Koot, J. Kessels, B. de Jager, W. Heemels, P. van den Bosch, and M. Steinbuch, "Energy management strategies for vehicular electric power systems," *IEEE transactions on vehicular technology*, vol. 54, no. 3, May. 2005, pp. 771-782.
- [11] T.J. Boehme, M. Schori, B. Frank, M. Schultalbers, and W. Drewelow, "A predictive energy management for hybrid vehicles based on optimal control theory," *American control conference*, Washington, DC, USA, June, 2013, pp. 5984-5989.
- [12] O. Ellabban, and H. Abu-Rub, "Predictive torque control of an induction motor fed by a bidirectional quasi z-source inverter," *IEEE international conference on Industrial Electronics Society*, Vienna, Austria, Nov. 2013, pp. 5854-5859.
- [13] A. Marouf, M. Djemaï, C. Sentouh and P. Pdlo, "Driver torque and road reaction force estimation of an electric power assisted steering using sliding mode observer with unknown inputs," *IEEE international conference on Intelligent Transportation Systems*, Madeira Island, Portugal, Sept. 2010, pp. 354-359.
- [14] R.C. Chabaan, "Torque estimation in electrical power steering systems," *IEEE Vehicle Power and Propulsion Conference*, Dearborn, USA, Sept. 2009, pp. 790-797.
- [15] S. Sekizawa, S. Inagaki, T. Suzuki, S. Hayakawa, N. Tsuchida, T. Tsuda, and H. Fujinami, "Modeling and recognition of driving behavior based on stochastic switched ARX model," *IEEE Transactions on Intelligent Transportation Systems*, vol.8, no. 4, Dec. 2007, pp. 593-606.
- [16] S. Taguchi, T. Suzuki, S. Hayakawa, and S. Inagaki, "Identification of hybrid system based on probability weighted multiple ARX model," *ICROS-SICE international Joint Conference*, Fukuoka, Japan, Aug. 2009, pp. 4837-4842.
- [17] H. Okuda, N. Ikami, T. Suzuki, Y. Tazaki, and K. Takeda, "Modeling and analysis of driving behavior based on a probability-weighted ARX model," *IEEE Transactions on Intelligent Transportation systems*, vol. 14, no. 1, Mar. 2013, pp. 98-112.
- [18] M. Sundbom, P. Falcone, and J. Sjöberg, "Online driver behavior classification using probabilistic ARX models," *IEEE proceedings of the 16<sup>th</sup> International Conference on Intelligent Transportation Systems*, The Hague, The Netherlands, Oct. 2013, pp. 1017-1112.
- [19] R. Vidal, and B. D. O. Anderson, "Recursive identification of switched ARX hybrid models: exponential convergence and persistence of excitation," *IEEE 43<sup>rd</sup> Conference on Decision and Control*, Atlantis, Bahamas, vol. 1, Dec. 2004, pp.32-37.

- [20] R. Wang, S.M. Lukic, "Review of driving conditions prediction and driving style recognition based control algorithms for hybrid electric vehicles," *IEEE Vehicle Power and Propulsion Conference (VPPC)*, Chicago, IL, USA, Sept. 2011, pp. 1-7.
- [21] S. Sun, X. Xu, "Variational inference for infinite mixtures of gaussian processes with applications to traffic flow prediction," *IEEE Intelligent Transportation Systems*, vol.12, Jun. 2011, pp.466-475.
- [22] E. Meeds, and S. Osindero, "An Alternative Infinite Mixture of Gaussian Process Experts," *Advances in Neural Information Processing Systems*, vol. 18, 2006, pp. 883-890.
- [23] G. Dhiman, K. Mihic, and T. Rosing, "A system for online power prediction in virtualized environments using Gaussian mixture models," *DAC'10 proceedings of the 47<sup>th</sup> design automation conference*, Ahaheim, California, USA, June. 2010, pp. 807-812.
- [24] C. Lin, H. Peng, and J.W. Grizzle, "A Stochastic Control Strategy for Hybrid Electric Vehicles," *American Control Conference*, Boston, MA, USA, vol. 5, Jun. 2004, pp. 4710-4715.
- [25] J. A. Oliva, C. Weihrauch, T. Bertram, "A model-based approach for predicting the remaining driving range in electric vehicles," *Annual conference of the prognostics and health management society*, New Orleans, LA, USA, Vol. 4, Oct. 2013, pp. 1-11.
- [26] Y. Ji, J. Hao, N. Reyhani, and A. Lendasse, "Direct and recursive prediction of time series using mutual information selection," *The 8th International Workshop on Artificial Neural Networks (IWANN)*, Barcelona, Spain, June, 2005. pp. 1010–1017.
- [27] H. Cheng, P. Tan, J. Gao, and J. Scripps, "Multi-step ahead time series prediction," *In Proceedings of the Tenth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, Singapore, vol. 3918, April, 2006, pp.765-774.
- [28] M. B. Stojanović, M. M. Bžić, and M. M. Stanković, "Mid-term load forecasting using recursive time series prediction strategy with support vector machines," *Electronics and Energetics*, vol. 23, no. 3, Dec. 2010, pp. 287-298.
- [29] A. Sorjamaa, and A. Lendasse, "Time series prediction using DirRec strategy," *European Symposium on Artificial Neural Networks*, Bruges, Belgium, April, 2006, pp. 143-148.

- [30] J. Tikka, A. Lendasse, and J. Hollmén, “Analysis of fast input selection: application in time series prediction,” *Internet corporation for Assigned Names and Numbers (ICANN)*, Athens, Greece, Sept. 2006, pp. 161-170.
- [31] S. B. Taieb, A. Sorjamaa, and G. Bontempi, “Multiple-output modeling for multi-step-ahead time series forecasting,” *Neurocomputing*, vol. 73, no. 10-12, Jun. 2010, pp. 1950-1957.
- [32] J. Tikka, J. Hollmén, and A. Lendasse, “Input selection for long-term prediction of time-series,” *In Proceedings of the 8th International Workshop on Artificial Neural Networks (IWANN)*, Barcelona, Spain, June, 2005, pp. 1002-1009.
- [33] G. Bontempi, “Long term time series prediction with multi-input multi-output local learning,” *In Proceedings of the 2nd European Symposium on Time Series Prediction (ESTSP)*, Helsinki, Finland, Feb. 2008, pp.145-154.
- [34] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse, “Methodology for long-term prediction of time series,” *Neurocomputing*, vol. 70, no. 16-18, 2007, pp. 2861-2869.
- [35] F. Yang, Z. Yin, H. X. Liu, and B. Ran, “Online recursive algorithm for short-term traffic prediction,” *Transportation Research Record*, vol. 1879, no. 1-8, 2004, pp. 1-8.
- [36] X. Chen, Y. Wang, I. Haskara, and G. Zhu, “Optimal Air-to-Fuel Ratio Tracking Control with Adaptive Biofuel Content Estimation for LNT Regeneration,” *IEEE Transactions on Control Systems Technology*, 2013.
- [37] X. Chen, G. Zhu, X. Yang, D. L. S. Hung, and X. Tan, “Model-based estimation of flow characteristics using an ionic polymer-metal composite beam,” *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 3, 2013, pp. 932-943.
- [38] T. Kailath (Series Editor), “Recursive Estimation Methods,” *In System Identification: Theory for the User (2nd Edition)*, New Jersey 07632: Prentice-Hall, Inc., Englewood Cliffs, 1987, pp. 306.
- [39] S. Joe Qin, “Recursive PLS algorithms for adaptive data modeling,” *Computers and Chemical Engineering*, vol. 22, 503514, 1998, pp. 503-514.
- [40] C. Musardo, G. Rizzoni, Y. Guezennec, and B. Staccia, “A-ECMS: An adaptive algorithm for hybrid electric vehicle energy management,” *European Journal of Control*, vol. 11, no. 4-5, 2005, pp. 509-524.

- [41] J. Liu and H. Peng, "Modeling and Control of a Power-Split Hybrid Vehicle," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, 2008, pp. 1242-1251.
- [42] J. Gao, G. Zhu, E. G. Strangas, and F. Sun, "Equivalent fuel consumption optimal control of a series hybrid electric vehicle," *In Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 23, 2009, pp.1003-1018.
- [43] H. Cohn and C. Umans, "Group-theoretic approach to fast matrix multiplication," *In Proceedings of the 44th Annual Symposium on Foundations of Computer Science*, Cambridge, MA, Oct. 2003, pp.438-449.
- [44] H. Cohn, R. Kleinberg, B. Szegedy, and C. Umans, "Group-theoretic algorithms for matrix multiplication," *In Proceedings of the 46th Annual Symposium on Foundations of Computer Science*, Pittsburgh, PA, USA, Oct. 2005, pp. 379-388.
- [45] J. Yang, and G. Zhu, "Adaptive recursive prediction of the desired torque of hybrid powertrain," *IEEE transactions on vehicular technology*, vol. 64, no.8, 2015, pp.3402-3413.
- [46] J. Yan, Y. Zhang, and C. Wang, "Li-ion cell operation at low temperatures," *J. Electrochem. Soc.* 2013, Volume 160, Issue 4, Pages A636-A649.
- [47] J. Jaguemont, L. Boulon, and Y. Dubé, "Characterization and modeling of a hybrid electric vehicle lithium - ion battery pack at low temperatures," *IEEE transactions on vehicular technology*, 2015.
- [48] G. Beauregard, D. Karner, and S. Ohba, "Current-compensated discharge limiter for industrial battery-powered vehicle," *SAE Technical Paper 2001-01-2494*, 2001.
- [49] R. Carlson, et al, "On road evaluation of advanced hybrid vehicles over a wide range of ambient temperatures", *EVS-23- paper # 275*, Anaheim, CA, December 2007.
- [50] S. Ebbesen, P. Elbert, and L. Guzzella, "Battery state-of-health perceptive energy management for hybrid electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 7, 2012, pp. 2893-2900.
- [51] Y. Cheng, R. Trigui, C. Espanet, A. Bouscayrol, and S. Cui, "Specifications and design of a PM electric variable transmission for Toyota Prius II," *IEEE Trans. Veh. Technol.*, vol. 60, no. 9, Nov. 2011, pp. 4106–4114.

- [52] J. Kim, and B.H. Cho, "State-of-charge estimation and state-of-health prediction of a li-ion degraded battery based on an EKF combined with a per-unit system," *IEEE transactions on vehicular technology*, vol. 60, no. 9, Nov. 2011, pp. 4249-4260.
- [53] Z. Wang, W. Li, and Y. Xu, "A novel power control strategy of series hybrid electric vehicle," *In Proceedings of the IEEE/RSJ International Conference on Intelligent robots and systems*, New York, Oct., 29-Nov., 2, 2007, pp.96-102.
- [54] H. He, J. Gao, and Y. Zhang, "Fuel cell output power-oriented control for a fuel cell hybrid electric vehicle," *In Proceedings of the American Control Conference*, New York, USA, June 11-13, 2008, pp.1605-610.
- [55] J. Yang, and G. Zhu, "Predictive boundary management control of a hybrid powertrain," *ASME 2014 Dynamic systems and control conference*, San Antonio, TX, USA, Oct. 22, 2014.
- [56] K.B. Wipke, M.R. Cuddy, and S. D. Burch, "ADVISOR 2.1: a user-friendly advanced powertrain simulation using a combined backward/forward approach," *IEEE Transaction on Vehicular Technology*, vol. 48, no. 6, 1999, pp.1751-1761.
- [57] T. Markel, A. Brooker, T. Hendricks, and et al., "Advisor: a system analysis tool for advanced vehicle modeling," *Journal of Power Sources*, vol. 110, no. 2, 2002, pp. 255-266.
- [58] J. M. Miller, and M. Everett, "An assessment of ultra-capacitors as the power cache in toyota THS-II, GM-allision AHS-2 and ford FHS hybrid propulsion systems," *In Proceedings of 20th Annu. Appl. Power Electron Conf. Expo.*, 2005, vol. 1, pp. 481-490.
- [59] S. Abe, "Development of the hybrid vehicle and its future expectation," *SAE, Warrendale, PA*, Tech. Rep. 2000-01-C042, 2000.
- [60] V.Johnson, "Battery performance models in ADVISOR," *Journal of Power Sources*, vol. 110, no.2, 2002, pp. 321-329.
- [61] A. A. Pesaran, "Battery thermal models for hybrid vehicle simulations," *Journal of Power Sources*, vol. 110, 2002, pp. 377-382.
- [62] A. Vasebi, M. Partovibakhsh, S. Monhammad Taghi Bathaee, "A novel combined battery model for state-of-charge estimation in lead-acid batteries based on extended Kalman filter," *Journal of Power Sources*, vol. 174, 2007, pp. 30-40.

- [63] S. Lee, J. Kim, J. lee, B.H. Cho, "State-of-charge and capacity estimation of lithium-ion battery using a new open-circuit voltage versus state-of-charge," *Journal of Power Sources*, vol. 185, 2008, pp. 1367-1373.
- [64] Z. He, M. Gao, C. Wang, L. Wang and Y. Liu, "Adaptive state of charge estimation for li-ion batteries based on an unscented Kalman filter with an enhanced battery model," *Energies*, vol. 6, 2013, pp. 4134-4151.
- [65] H. Wang, Y. Liu, H. Fu and G. Li, "Estimation of state of charge of batteries for electric vehicles," *International Journal of Control and Automation*, vol. 6, no. 2 April, 2013, pp. 185-194.
- [66] USABC, "Electric vehicle battery test procedure manual," *DOE/ID-1049*, Jan. 1996.
- [67] Z. M. Salameh, M. A. Casacca, and W. A. Lynch, "A mathematical model for lead-acid batteries," *IEEE Transactions on Energy Conversion*, vol. 7, no. 1, 1992, pp. 93-97.
- [68] W. Chang, "The state of charge estimation methods for battery: a review," *Hindawi Publishing Corporation, ISRN applied mathematics*, vol. 2013, pp. 1-7.
- [69] X. Tang, X. Mao, L. Jian, etc., "Li-ion battery parameter estimation for state of charge," *American control conference*, San Francisco, CA, USA, June, 2011, pp. 941-946.
- [70] W. He, N. Williard, C. Chen, and M. Pecht, "State of charge estimation for electric vehicle batteries using unscented kalman filter," *Microelectronics Reliability*, vol. 53, 2013, pp. 840-847.
- [71] J. Wang, B. Cao, Q. Chen, and F. Wang, "Combined state of charge estimator for electric vehicle battery pack," *Control Engineering Practice*, vol. 15, 2007, pp. 1569-1576.
- [72] G. Ripaccioli, D. Bernardini, S. Di Cairano, et al., "A stochastic model predictive control approach for series hybrid electric vehicle power management," *American control conference*, Baltimore, MD, USA, June 30, 2010, pp. 5844-5849.
- [73] H. Borhan, A. Vahidi, L. Wei, et al., "Nonlinear model predictive control of a power-split hybrid electric vehicle: influence of inclusion of powertrain dynamics," *Dynamic systems and control conference*, Arlington, VA, USA, Oct. 31, 2011, pp.1-8.

- [74] G. Paganelli, S. Delprat, T. Guerra, J. Rimaux, and J. Santin, "Equivalent consumption minimization strategy for parallel hybrid powertrains," *IEEE vehicular technology conference*, 2002, vol. 4, pp. 2076-2081.
- [75] A. Sciarretta, M. Back, and L. Guzzella, "Optimal control of parallel hybrid electric vehicles," *IEEE transactions on control systems technology*, 2004, vol. 12, no. 3, pp. 352-363.
- [76] H. A. Borhan and A. Vahidi, "Model predictive control of a power-split hybrid electric vehicle with combined battery and ultra-capacitor energy storage," *American control conference*, Baltimore, MD, USA, June 30, 2010, pp. 5031-5036.
- [77] A. Phillips, M. Kuang, H. Borhan, A. Vahidi and I. Kolmanovsky, "Predictive energy management of a power-split hybrid electric vehicle," *American control conference*, St. Louis, MO, USA, June 10, 2009, pp. 3970-3976.
- [78] K. Yu, J. Yang, and D. Yamaguchi, "Model predictive control for hybrid vehicle ecological driving using traffic signal and road slope information," *Control theory and technology*, 2015, vol. 13, no. 1, pp. 17-28.
- [79] F. Yan, J. Wang, and K. Huang, "Hybrid electric vehicle model predictive control torque-split strategy incorporating engine transient characteristics," *IEEE transactions on vehicular technology*, 2012, vol. 61, no. 6, pp. 2458-2467.
- [80] S. Barsali, C. miulli, and A. Possenti, "A control strategy to minimize fuel consumption of series hybrid electric vehicles," *IEEE transactions on vehicular technology*, 2004, vol.19, no. 1, pp. 187-195.
- [81] S. J. Moura, h. K. Fathy, D. S. Callaway, and J. L. Stein, "A stochastic optimal control approach for power management in plug-in hybrid electric vehicles," *IEEE transactions on vehicular technology*, 2011, vol.19, pp. 545-555.
- [82] J. Yang, and G. Zhu, "Stochastic predictive boundary management for a hybrid powertrain," *Submitted to IEEE transactions on vehicular technology*, 2015.
- [83] H. A. Borhan, A. Vahidi, A. M. Philips, M. L. Kuang, and I. V. Kolmanovsky, "Predictive energy management of a power-split hybrid electric vehicle," *American control conference*, St. Louis, MO, USA, June 10, 2009, pp.3970-3976.

- [84] D. S. Naidu, *Optimal control systems*, *CRC press*, Boca Raton, London, New York, Washington, D.C. 2003.