Real-time Optimal Energy Management System for Plug-in Hybrid Electric Vehicles

by

Amir Taghavipour

A thesis
presented to the University of Waterloo
in partial fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2014

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

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

Abstract

Air pollution and rising fuel costs are becoming increasingly important concerns for the transportation industry. Hybrid electric vehicles (HEVs) are seen as a solution to these problems as they offer lower emissions and better fuel economy compared to conventional internal combustion engine vehicles. A typical HEV powertrain consists of an internal combustion engine, an electric motor/generator, and a power storage device (usually a battery). Another type of HEV is the plug-in hybrid electric vehicle (PHEV), which is conceptually similar to the fully electric vehicle. The battery in a PHEV is designed to be fully charged using a conventional home electric plug or a charging station. As such, the vehicle can travel further in full-electric mode, which greatly improves the fuel economy of PHEVs compared to HEVs.

In this study, an optimal energy management system (EMS) for a PHEV is designed to minimize fuel consumption by considering engine emissions reduction. This is achieved by using the model predictive control (MPC) approach. MPC is an optimal model-based approach that can accommodate the many constraints involved in the design of EMSs, and is suitable for real-time implementations. The design and real-time implementation of such a control approach involves control-oriented modeling, controller design (including high-level and low-level controllers), and control scheme performance evaluation. All of these issues will be addressed in this thesis.

A control-relevant parameter estimation (CRPE) approach is used to make the control-oriented model more accurate. This improves the EMS performance, while maintaining its real-time implementation capability.

To reduce the computational complexity, the standard MPC controller is replaced by its explicit form. The explicit model predictive controller (eMPC) achieves the same performance as the implicit MPC, but requires less computational effort, which leads to a fast and reliable implementation. The performance of the control scheme is evaluated through different stages of model-in-the-loop (MIL) simulations with an equation-based and validated high-fidelity simulation model of a PHEV powertrain.

Finally, the CRPE-eMPC EMS is validated through a hardware-in-the-loop (HIL) test. HIL simulation shows that the proposed EMS can be implemented to a commercial control hardware in real time and results in promising fuel economy figures and emissions performance, while maintaining vehicle drivability.

Acknowledgements

I would like to thank my supervisors, Prof. Azad and Prof. McPhee who guided me throughout this work.

I acknowledge the NSERC/Toyota/Maplesoft Industrial Research Chair program as well as OCE for financial support of this research.

I'm also grateful of my friends in Motion Research Group, with special thanks to Dr. Masoudi, and Mr. Sharif Razavian who helped me push my research ahead whenever I was having difficulties.

Dedication

To all my loved ones . . .

Table of Contents

List of Tables **xi**

List of Figures **xiii**

Nomenclature **xviii**

1 Introduction **1**

1.1 Motivation and Challenges 1

1.2 Model Predictive Control 4

1.3 Problem Statement 5

1.3.1 Control-oriented Modeling 6

1.3.2 Control Design 6

1.3.3 Control Scheme Evaluation 6

1.4 List of Contributions 7

1.5 Thesis Layout 11

2 Literature Review and Background **12**

2.1 Hybrid Powertrain Architectures 12

2.2 Energy Management Systems 14

2.3 Hybrid Powertrain Modeling 17

2.4 Model Predictive Control 18

2.5 Explicit Model Predictive Control 21
2.6 Control-relevant Parameter Estimation 23
2.7 Hardware-in-the-loop Simulation 25
2.8 Chapter Summary 27

3 High-fidelity Modeling of a Plug-in Hybrid Electric Powertrain 28
3.1 Introduction 28
3.2 Toyota Prius Plug-in Hybrid Powertrain 30
3.3 MapleSim model 32
3.3.1 Mean-value Internal Combustion Engine 32
3.3.2 Electric Machines 35
3.3.3 Lithium-ion Battery Pack 35
3.3.4 Power-split Device 36
3.3.5 Vehicle Model 36
3.4 Model Validation 36
3.4.1 Mean-value Internal Combustion Engine 37
3.4.2 Electric Machines 39
3.4.3 Lithium-ion Battery Pack 39
3.4.4 Power-split Device 44
3.4.5 Vehicle Model 44
3.5 Chapter Summary 45

4 Non-linear Model Predictive Control Design 46
4.1 Introduction 46
4.2 Theory of Model Predictive Control (MPC) 47
4.2.1 Problem Formulation 47
4.2.2 Stability Analysis 51
4.2.3 Robustness Analysis 53

4.3 NMPC Performance on the Low-fidelity Powertrain Model 54
4.3.1 Control-oriented Model 54
4.3.2 MPC Formulation for a PHEV 55
4.3.3 No Knowledge of Trip Information 56
4.3.4 Known Travelling Distance 57
4.3.5 Known Entire Trip Information 57
4.3.6 Manual Charge Depletion Charge Sustenance 59
4.3.7 Simulation Results 59
4.4 NMPC Performance Benchmarking 64
4.5 NMPC Performance on the High-fidelity Powertrain Model 65
4.5.1 Control-oriented Model inside the High-level Controller 69
4.5.2 NMPC Energy Management System 71
4.6 Low-level Controllers 72
4.6.1 Engine Control-oriented Model 73
4.6.2 Engine Controls Design 75
4.7 Results of Simulation 76
4.7.1 Without Emissions Control 77
4.7.2 With Emissions Control 79
4.8 Chapter Summary 85

5 Explicit Model Predictive Control Design 87
5.1 Introduction 87
5.2 eMPC Energy Management Strategy Design 89
5.2.1 Control-oriented model 90
5.2.2 Optimization problem formulation 91
5.2.3 Region reduction 95
5.2.4 Point location problem 96
5.3 Energy Management Polytopes 97

5.4 Stability notes . . . 101
5.5 eMPC Performance on the High-fidelity Powertrain Model . . . 106
5.5.1 No Knowledge of Trip Information . . . 108
5.5.2 Known Travelling Distance . . . 110
5.5.3 Discussions . . . 110
5.6 Control-relevant Parameter Estimation (CRPE) . . . 112
5.6.1 Battery Thevenin Model . . . 113
5.6.2 Battery Parameters Estimation . . . 115
5.6.3 CRPE Control-oriented Model . . . 118
5.7 CRPE-eMPC Formulation for a PHEV . . . 119
5.7.1 CRPE-eMPC Controls Regions . . . 120
5.7.2 CRPE-eMPC Stability Notes . . . 122
5.8 CRPE-eMPC Performance on the High-fidelity Powertrain Model . . . 124
5.8.1 No Knowledge of Trip Information . . . 124
5.8.2 Known Travelling Distance . . . 127
5.8.3 Discussions . . . 127
5.9 Chapter Summary . . . 129

6 Hardware-in-the-loop Simulation 132
6.1 Introduction . . . 132
6.1.1 ECU Validation Procedure . . . 133
6.1.2 Virtual Simulation Model Requirements . . . 135
6.1.3 Real-time Target Requirements . . . 136
6.2 Hardware Description . . . 137
6.2.1 MotoTron . . . 137
6.2.2 PXI Real-time Target . . . 138
6.2.3 CAN Bus . . . 139
6.3 Controls Implementation Notes . . . 141

6.4 eMPC on the Low-fidelity Model 142
6.5 CRPE-eMPC on the High-fidelity Model 145
6.6 Chapter Summary 150

7 Conclusions **151**
7.1 Summary 151
7.2 Recommendations for Future Research 153
7.2.1 Controls Design 153
7.2.2 Controls Validation 155

References **156**

APPENDICES **174**

A Model Parameters and Variables **175**

B Controls Design for Electric Drive **177**
B.1 Electric Drive Control 177
B.1.1 Clarke Transformation 179
B.1.2 Flux and Torque Estimation 181
B.1.3 Direct Torque Control 182
B.2 Power Electronics High-fidelity Model 184
B.3 Results of Simulation 186

C CAN Bus Arbitration ID **190**

List of Tables

3.1 Engine specifications 31
3.2 MG2 specifications 31
3.3 Battery-pack specifications 32
3.4 Identified values for the battery parameters 43

4.1 Fuel economy for different control strategies 63
4.2 Fuel consumption comparison 85

5.1 MIL with the high-fidelity powertrain model: Fuel economy for different control strategies 112
5.2 Identified values for the parameters of the equivalent dual-polarization circuit model 118
5.3 MIL with the high-fidelity powertrain model: Fuel economy for different control strategies 129

6.1 CAN message definition for the HIL simulation 140
6.2 eMPC MIL and HIL test using low-fidelity powertrain model: Fuel economy for different control strategies 145
6.3 CRPE-eMPC MIL and HIL test using high-fidelity powertrain model: Fuel economy for different control strategies 148
6.4 Fuel economy summary 150

A.1 Variables and Parameters Value 176

B.1 Switching table 183
B.2 Torque and flux hysteresis 184
B.3 Takahashi and Noguchi switching table 185

List of Figures

2.1 Series architecture schematic 13
2.2 Parallel architecture schematic 13
2.3 Power-split architecture schematic 14

3.1 The transmission of Toyota Prius plug-in hybrid 30
3.2 Toyota Prius plug-in hybrid high-fidelity simulation model in MapleSim . 33
3.3 PHEV powertrain model in Autonomie 37
3.4 The engine brake specific fuel consumption (BSFC) map 38
3.5 Equivalent thermal efficiency for different values of engine speed and manifold pressure 38
3.6 MG2 efficiency map 39
3.7 Time history of the simulated battery voltage for experimental data and simulated response using the identified parameters 42
3.8 Convergence trend of the objective function during the optimization process 43
3.9 Time history of the battery voltage for various discharge rates 44
3.10 Powertrain model in MapleSim and Autonomie: simulation result along 2 UDDS drive cycles 45

4.1 GPS information and segments for one UDDS drive cycle 58
4.2 Charge depletion/charge sustenance strategy: (a) SOC along 2 UDDS drive cycles (b) Power source torques 61
4.3 Blended mode strategy (Known travelling distance): (a) SOC along 2 UDDS drive cycles (b) Power source torques 61

4.4 Blended mode strategy (Entire trip information): (a) SOC along 2 UDDS drive cycles (b) Power source torques 62
4.5 Manually switching between EV and CS: (a) SOC along 2 UDDS drive cycles (b) Power source torques 62
4.6 Fuel economy versus ratio of weighting parameters in the cost function . . 64
4.7 DP result: (a) MG2 Torque (b) MG2 Speed (c) MG2 efficiency 65
4.8 DP result: (a) MG1 Torque (b) MG1 Speed (c) MG1 efficiency 66
4.9 DP result: (a) Engine Torque (b) Engine Speed 66
4.10 SOC comparison for DP and MPC in charge sustenance mode 67
4.11 Engine torque change with air to fuel ratio and ignition timing 74
4.12 Simulation procedure 77
4.13 Blended mode MPC strategy without emission control: (a) Velocity and Battery SOC (b) MG2 Torque (c) Engine Torque (d) Fuel Consumption . . 78
4.14 CDCS MPC strategy without emission control: (a) Velocity and Battery SOC (b) MG-2 Torque (c) Engine Torque (d) Fuel Consumption 80
4.15 Blended mode MPC strategy with emission control: (a) Velocity and Battery SOC (b) MG-2 Torque (c) Engine Torque (d) Fuel Consumption 81
4.16 CDCS MPC strategy with emission control: (a) Velocity and Battery SOC (b) MG-2 Torque (c) Engine Torque (d) Fuel Consumption 82
4.17 Blended mode MPC strategy with emission control: (a) Ignition Timing (b) Manifold Pressure (c) AFR (d) Conversion efficiency 83
4.18 CDCS MPC strategy with emission control: (a) Ignition timing (b) Manifold pressure (c) AFR (d) Conversion efficiency 84
4.19 Catalyst conversion efficiency for (a) CDCS (b) Blended mode strategies: without emission control(bullet marker)/with emission control (cross marker) 85

5.1 eMPC design procedure 90
5.2 Number of polytopes for different levels of E_{ref} and SOE_{ref} 98
5.3 Polytope set for $E_{ref} = 0$ and $SOE_{ref} = 60\%$ 98
5.4 Control actions for $E_{ref} = 0$ and $SOE_{ref} = 60\%$ based on different initial conditions (a) battery power (b) engine power (c) braking power 99

5.5 Cost function for $E_{ref} = 0$ and $SOE_{ref} = 60\%$ based on different initial conditions . . . 101
5.6 Spectral radius of $\tilde{A}$ for different levels of E_{ref} and SOE_{ref} . . . 103
5.7 The locus of switching system poles in z-plane . . . 106
5.8 Low-level controls implementation . . . 107
5.9 CDCS eMPC strategy with emission control: (a) Velocity and Battery SOC (b) MG2 Torque (c) Engine Torque . . . 108
5.10 CDCS eMPC strategy with emission control: (a) Engine Torque (b) HC conversion efficiency (c) Air-to-fuel ratio (AFR) . . . 109
5.11 Catalyst conversion efficiency for CDCS eMPC strategy . . . 109
5.12 Blended eMPC strategy with emission control: (a) Velocity and Battery SOC (b) MG2 Torque (c) Engine Torque . . . 110
5.13 Blended eMPC strategy with emission control: (a) Engine Torque (b) HC conversion efficiency (c) Air-to-fuel ratio (AFR) . . . 111
5.14 Catalyst conversion efficiency for Blended eMPC strategy . . . 111
5.15 Thevenin equivalent model of the battery . . . 113
5.16 Time history of the battery power, applying eMPC to the PHEV for UDDS driving cycle . . . 115
5.17 Power spectral density analysis of the chemistry-based battery power signal, when an eMPC scheme is applied to the PHEV . . . 116
5.18 Dual-polarization model of the battery . . . 117
5.19 Time history of the SOC profile for the dual-polarization equivalent and chemistry-based battery model . . . 117
5.20 Number of polytopes for different levels of E_{ref} and SOC_{ref} . . . 121
5.21 Polytope set for $E_{ref} = 0$ and $SOC_{ref} = 60\%$ . . . 121
5.22 Control actions for $V_{batt} = 180V$, $E_{ref} = 0$, and $SOC_{ref} = 60\%$ based on different initial conditions (a) battery power (b) engine power (c) braking power . . . 123
5.23 CDCS CRPE-eMPC strategy with emission control: (a) Velocity and Battery SOC (b) MG2 Torque (c) Engine Torque . . . 125

5.24 CDCS CRPE-eMPC strategy with emission control: (a) Engine Torque (b) HC conversion efficiency (c) Air-to-fuel ratio (AFR) 126
5.25 Catalyst conversion efficiency for CDCS CRPE-eMPC strategy 126
5.26 Blended CRPE-eMPC strategy with emission control: (a) Velocity and Battery SOC (b) MG2 Torque (c) Engine Torque 127
5.27 Blended CRPE-eMPC strategy with emission control: (a) Engine Torque (b) HC conversion efficiency (c) Air-to-fuel ratio (AFR) 128
5.28 Catalyst conversion efficiency for Blended CRPE-eMPC strategy 128
5.29 eMPC vs. CRPE-eMPC:(a)Battery power (b)Engine power (c)Battery power for constant $E = 10kWs$ (d)Engine power for constant $E = 10kWs$ (e)Battery power for constant $SOC = 50\%$ (e)Engine power for constant $SOC = 45\%$ 130

6.1 ECU validation procedure 134
6.2 Schematic of the HIL setup 139
6.3 High-fidelity model inside the real-time computer 142
6.4 Charge depletion/ charge sustenance strategy (a) vehicle velocity and battery depletion profile (b) demanded and propulsion power (c) demanded power and SOE indices 143
6.5 Blended mode strategy (a) vehicle velocity and battery depletion profile (b) demanded and propulsion power (c) demanded power and SOE indices . . 144
6.6 HIL simulation, CDCS CRPE-eMPC strategy with emission control: (a) Velocity and Battery SOC (b) MG2 Torque (c) Engine Torque 146
6.7 HIL simulation, CDCS CRPE-eMPC strategy with emission control: (a) Engine Torque (b) HC conversion efficiency (c) Air-to-fuel ratio (AFR) . . 147
6.8 HIL simulation, catalyst conversion efficiency for CDCS CRPE-eMPC strategy 147
6.9 HIL simulation, Blended CRPE-eMPC strategy with emission control: (a) Velocity and Battery SOC (b) MG2 Torque (c) Engine Torque 148
6.10 HIL simulation, Blended CRPE-eMPC strategy with emission control: (a) Engine Torque (b) HC conversion efficiency (c) Air-to-fuel ratio (AFR) . . 149
6.11 HIL simulation, catalyst conversion efficiency for blended CRPE-eMPC strategy 149

B.1 d-q representation of a permanent magnet synchronous motor 178
B.2 Direct torque control scheme 179
B.3 MG2 inverter schematic 182
B.4 Six sectors representation 183
B.5 Voltage vector selection scheme 184
B.6 Electric drive high-fidelity model in MapleSim 185
B.7 (a) MG2 torque along one UDDS drive cycle for Blended mode CRPE-eMPC strategy (b) MG2 Torque tracking error (c)zoomed view for $t \in [780, 860]s$ 186
B.8 (a) MG2 equivalent stator flux (b)MG2 torque (c) flux and torque variation indices (d) flux vector direction (e) dq current (f) dq voltage (g) Measured currents (h)Applied voltages (i) IGBTs on/off status for measurement sampling time of 20 ms 188
B.9 (a) MG2 equivalent stator flux (b)MG2 torque (c) flux and torque variation indices (d) flux vector direction (e) dq current (f) dq voltage (g) Measured currents (h)Applied voltages (i) IGBTs on/off status for measurement sampling time of 20 μs 189

C.1 Different byte allocation methods in CAN data frame 191
C.2 An example for different endianness definitions 191

Nomenclature

A_d	Vehicle frontal area
A_{th}	Engine throttle area
$\boldsymbol{B}$	Battery model input matrix
C	Thevenin equivalent capacitance
C_i	Dual polarization equivalent capacitance
C_D	Engine throttle discharge coefficient
D_{th}	Diameter of output vent
$\boldsymbol{E}$	Battery model state matrix
E	Tractive energy
E_{ref}	Reference tractive energy
$\boldsymbol{H}$	Hessian of homotopy objective function
H_f	Gasoline heat of combustion
I_a	Current in a-coordinate
I_b	Current in b-coordinate
I_c	Current in c-coordinate
I_C	Carrier equivalent inertia
I_d	Current in d-coordinate
I_e	Engine equivalent inertia
I_g	MG1 equivalent inertia
I_m	Motor equivalent inertia
I_q	Current in q-coordinate
I_r	Ring gear equivalent inertia
I_s	Sun gear equivalent inertia
I_S	Motor stator current
K	Gear ratio
K_i	Homotopy gain
N_c	Control horizon length
N_{cyl}	Engine number of cylinders
N_{eng}	Engine stroke number index
N_p	Prediction horizon length
P_0	Atmosphere pressure
P_{dem}	Demanded power
P_m	Engine intake manifold pressure
P_{mot}	MG2 power
P_{BAT}	Battery power
P_{BRK}	Brake power
P_{ENG}	Engine power
Q_{batt}	Battery capacity

R	Thevenin equivalent resitance
R_{air}	Air constant
R_{batt}	Battery resistance
R_i	Dual polarization equivalent resitance
R_S	Stator equivalent resistance
R_{tire}	Tire radius
T_0	Atmosphere temperature
T_d	Demanded torque
T_{des}	Desired engine torque
T_e	Engine torque
T_{est}	Estimated torque
T_{exh}	Exhaust gas temperature
$T_{exh,des}$	Desired exhaust temperature
T_f	Powertrain load torque
T_i	Polytope index
T_{ind}	Engine indicated torque
T_m	MG2 torque
T_{man}	Engine intake manifold temperature
T_r	Ring torque
T_s	Time constant
U^*	Input optimizer
U^{min}	Minimum values of all predicted inputs
U^{max}	Maximum values of all predicted inputs
V_{batt}	Battery model voltage
V_a	Voltage in a-coordinate
V_b	Voltage in b-coordinate
V_c	Voltage in c-coordinate
V_d	Voltage in d-coordinate
V_m	Engine intake manifold volume
V_{oc}	Battery voltage
V_q	Voltage in q-coordinate
V_D	Engine displacement
V_S	Motor stator voltage
$V(\boldsymbol{\Gamma})$	Homotopy objective function
$\boldsymbol{X}$	Vehicle traveled distance
Y^{min}	Minimum values of all predicted outputs
Y^{max}	Maximum values of all predicted outputs

b_λ	Flux variation index
b_T	Torque variation index
c_d	Drag coefficient
c_{e0}	Initial electrolyte concentration in region s, n, and p
d_{th}	Diameter of input vent
f_r	Tire rolling resistance coefficient
g	Gravity acceleration
$\boldsymbol{g}$	Gradient of homotopy objective function
i_{batt}	Battery model current
k	Electric motor efficiency index
m	Vehicle mass
$\dot{m}_{act,fuel}$	Actual fuel rate
$\dot{m}_{air}$	Air mass rate entering engine cylinders
$\dot{m}_f$	Fuel consumption rate
(m_{f_i})	Fuel consumption rate in a driving segment
$\dot{m}_{inj,fuel}$	Injected fuel rate
p	Motor number of poles
r_1	Ring teeth No. of PGS1
r_2	Ring teeth No. of PGS2
s_1	Sun teeth No. of PGS1
s_2	Sun teeth No. of PGS2
t_+	Li^+ transference number in the electrolyte
w_i	Weighting parameters
v_g	PI controller output voltage for MG1
v_m	PI controller output voltage for MG2
AFR	Air to fuel ratio
AFR_{des}	Desired engine air fuel ratio
PR	Electric power ratios
SOC	Battery state of charge
SOC_{high}	The highest battery state of charge
SOC_{low}	The lowest battery state of charge
SOC_{ref}	Battery reference state of charge
SOE	Battery state of energy
SOE_{ref}	Battery state of energy reference
α	Bound on the Lyapunov function
$\bar{\alpha}$	Engine speed coefficient
β	Stability constant
$\bar{\beta}$	Engine power coefficient
γ	Air heat capacity ratio

ϵ_s	Battery model volume fraction of separator region
ε	Homotopy desired tolerance
ε_λ	Flux hysteresis band
ε_T	Torque hysteresis band
ζ	Stability constant
η_{AFR}	Engine AFR-related efficiency
η_{conv}	Catalyst conversion efficiency
η_i	Engine thermal efficiency
η_m	MG2 efficiency
η_v	Engine volumetric efficiency
η_Δ	Engine ignition timing-related efficiency
θ	Engine throttle angel
θ_0	Totally closed throttle angle
κ	Homotopy step size
λ	Lagrange multiplier
λ_{dS}	Stator equivalent flux in d-coordinate
λ_{qS}	Stator equivalent flux in q-coordinate
λ_{pm}	Rotor permanent magnet equivalent flux
λ_S	Stator equivalent flux
ξ	Battery model vector of state variables
ξ_{exp}	Battery model vector of measured state variables
ρ	Spectral radius
ρ_{air}	Air density
σ	MG2 efficiency index
σ_n	Electronic conductivity of solid phase of electrode n
τ_f	Wall-wetting time constant
ω_e	Engine speed
ω_g	MG1 speed
ω_m	MG2 speed
ω_r	Ring speed
υ	Homotopy parameter
ϖ	Bound on the Lyapunov function
$\boldsymbol{\Gamma}$	Battery model vector of parameters
Δ	Angle of crankshaft at the ignition occurrence
ΔU^{min}	Minimum values of all predicted inputs variation
ΔU^{max}	Maximum values of all predicted inputs variation
ΔY^{min}	Minimum values of all predicted outputs variation
ΔY^{max}	Maximum values of all predicted outputs variation
Θ_i	Motor flux sectors

Chapter 1

Introduction

Green vehicles or environmentally-friendly forms of transportation, are making a revolution in the automotive industry. The high cost of fuel and harmful emissions from internal combustion engines have motivated a transformation in the transportation industry. In an effort to find alternatives, electricity is emerging as a clean and reliable solution for propelling future vehicles. Automotive engineers now dream of a fully electric fleet, though there remain some technical hindrances in the way of making this dream come true. Engineers first attempted to reduce the size of the engine while introducing a small number of electric devices in order to power these so-called hybrid electric vehicles (HEVs). Many other sources besides the fuel - such as mechanical, hydraulic, or even pneumatic devices - were considered as choices at this early stage of development; however, electrical equipment seemed to be more promising than its counterparts for passenger cars. Aside from the rigorous emissions standards, each year the market provides some incentives for customers to pay for extra equipment on HEVs, with a reward of better fuel economy.

1.1 Motivation and Challenges

Air pollution and rising fuel costs are important concerns for the transportation industry. Hybrid electric vehicles (HEVs) are seen as a solution to these problems. The main mission

of HEVs is to offer lower fuel consumption when compared to conventional vehicles powered only by internal combustion engines. The other sources of energy in HEV powertrains allow the engines to be smaller and more efficient, which translates into lower emissions and better fuel economy. HEV powertrains consist of an efficient engine, an electric motor/generator, and a power storage device that is usually a battery. There must be a device on board analogous to the fuel tank for storing electricity, which is the most challenging aspect of a hybrid or electric vehicle. In general, this device is a battery which introduces problems regarding price and durability of the component. In fact, automotive engineers are eagerly waiting for battery technology to progress to the point where long-lasting devices can be manufactured at a low cost, which would reduce the total cost of an electric vehicle. As the current battery technology matures, it will be possible to have larger and more efficient batteries installed on an HEV. This development would boost the appeal of HEVs ultimately paving the way for commercializing fully-electric vehicles with desired performance and range.

With the development of advanced battery technologies, the energy storage capacity of batteries has improved significantly. Plug-in hybrids have greater battery storage capacity as compared to that of conventional hybrids. In fact, the plug-in hybrid concept can be thought of as a bridge connecting the conventional HEV technology to the potential ultimate transportation solution: the electric vehicle. The plug-in hybrid electric drivetrain is designed to use (either full or partial) energy from the energy storage component to replace part of the primary energy source [1]. Since the battery in a plug-in hybrid vehicle (PHEV) can be fully charged using a conventional home electric plug, the vehicle can travel further in pure electric mode. As a result, the engine is turned off for a longer period of time during the trip, which explains the superior fuel economy of PHEVs compared to conventional hybrids. Moreover, since most urban trips are relatively short, there is a high probability of always having the engine off when driven in the city [2]. About half of all daily driving distance is less than 64 km (40 miles) [2]. If a vehicle is designed to travel 64 km (40 miles) in pure electric mode, that vehicle will spend half of its life as a pure electric vehicle (EV) [1]. Thus, it will rarely be necessary to start the engine on most urban trips which, again, leads to a better fuel economy of PHEVs compared to conventional hybrids.

According to the Electric Power Research Institute (EPRI), more than 40% of the U.S. generation capacity operates at a reduced load overnight, and it is during these off-peak hours when most PHEVs could be recharged. Recent studies show that if PHEVs replace one-half of all vehicles on the road by 2050, only an 8% increase in electricity generation (4% rise in capacity) will be required [3]. At current electricity rates, the incremental cost of charging a PHEV fleet overnight will range from $90 to $140 per vehicle per year, which translates to an equivalent gasoline production cost of about 60 cents to 90 cents per gallon [4]. Thus, PHEVs are a very interesting option for the future of transportation. PHEVs can be designed with different power source configurations in the drive train as well as different energy management systems (EMSs).

The EMS is responsible for deciding how much power should be produced by the internal combustion engine and how much should be stored/released from the auxiliary energy storage systems to achieve the desired power at the wheels, by enforcing the operating constraints, and in a way that optimizes fuel economy. The EMS design still remains a challenging and most critical problem even after more than a decade after hybrid electric vehicle introduction to the automotive market.

To further improve PHEVs performance, advanced control strategies are needed for deciding the amount of energy to be produced and stored [5]. Additionally, the performance of a PHEV is closely related to the way the battery is depleted.

The automotive industry, just like any other high-tech industry, where electronic controls are the main part of each product is looking to reduce their development procedure. Model-based control design is a time-saving and cost effective approach that results in an optimized and validated system. Since the controls design can be done with a single model of a complete system in an integrated software environment, there is no risk that individual components do not fit together optimally. On the other hand, these controllers should be real-time implementable to deal with the fast dynamics involved in different parts of a vehicle. As a result, the automotive industry is actively researching solutions to the challenges of implementing model-based approaches in terms of designing EMSs.

Recently, the automotive industry has seen a considerable change in the implementation

process of control systems, with the integration of rapid development tools for products. The adoption of these rapid-prototyping concepts for vehicle and component testing and validation, called hardware-in-the-loop (HIL) simulation, is a significant change that involves modeling the plant hardware (engine, transmission, vehicle dynamics, etc.) being controlled, and interfacing this model with the intended controller. The main benefit of HIL simulation is that one can see the interface between model and controller during all phases of testing and the development procedure, which means that the controller issues can be identified and fixed earlier in the design process. This advanced testing capability has been shown to significantly reduce the overall development time and greatly improve the quality and reliability of the final product. Furthermore, automotive suppliers can validate their individual subsystems using HIL systems which are identical to those used by the original equipment manufacturers (OEMs). These systems give the test engineer the ability to simulate a variety of time consuming or expensive scenarios for a vehicle prototype[6].

In this thesis, a real-time, model-based and near-optimal EMS for Toyota Prius plug-in hybrid powertrain is proposed and its performance is validated through hardware-in-the loop simulation.

1.2 Model Predictive Control

In this research, a model-based strategy is proposed using the model predictive control (MPC) concept. MPC seems to be appropriate for exploiting the potentials of modern concepts and fulfilling modern automotive requirements, since most of these requirements can be stated in the form of a constrained multi-input multi-output near-optimal control problem and MPC provides an approximate solution to this class of problems [7].

In a typical MPC framework, the sequence of optimal control actions/inputs to the system are determined based on the prediction of the behavior of the system over a finite time horizon [8]. This is achieved by solving an on-line constrained optimization problem repetitively, at each sampling instant during which the system state/output measurements

(or estimates) become available. Only the first input is then applied to the system and the optimization is repeated at the next sampling time when the new set of measurements are available [9].

The main drawback of this method is the extensive computational effort required for each time step of control. Traditionally, MPC has been used primarily for controlling chemical processes that involve relatively slow dynamics. Thanks to the faster processors now available, there is an obvious motivation for applying the model-based control method to fast systems especially for automotive systems. The application of MPC to HEVs has already been investigated [10, 11]; however, this method has not been applied to the design of an energy management strategy for a plug-in power-split HEV - the goal we seek in this research. In general, MPC is the only advanced control technology that has made a substantial impact on industrial control problems; its success is largely due to its almost unique ability to simply and effectively handle hard constraints on control and states [12]. Another reason to use MPC synthesis is that, when implemented in a receding horizon fashion, an optimization problem is solved at every time step. This enables the controller to adapt to actual working conditions.

Furthermore, the advances in predictive control, particularly the development of explicit model predictive control (eMPC) schemes and also developments in the field of numerical computation of optimal solutions, provide new possibilities.

For the implementation, it is desirable to solve the optimization problem off-line in an explicit manner via a multi-parametric program, instead of direct on-line implementation of the controller, which yields an explicit and piecewise affine (PWA) control law [13].

1.3 Problem Statement

The goal of the proposed research is to design a near-optimal and real-time EMS for PHEV powertrains. To this end, the issues listed below must be considered:

1.3.1 Control-oriented Modeling

Since the model predictive controller is a model-based approach, a sufficiently accurate model to represent the behavior of the system is needed. Moreover, this model should be sufficiently simple and fast in order to be suitable for the real-time controller implementation. There is an obvious trade-off between model complexity and computational efficiency; therefore, model reduction methods that take control issues into account should be considered. In this case, the control-relevant parameter estimation (CRPE) method is used to find an appropriate control-oriented model for use within the controller. At the initial stages of this research, simple models of the system can be considered to acquire insight into the problem, and then investigate systematic reduction procedures.

1.3.2 Control Design

The variety of components in the powertrain demands the use of two layers of control. A supervisory controller is designed based on the MPC concept. As shown in chapter 4, control parameters are involved in the design of a MPC and they must be carefully chosen to provide desirable performance. Since the commands of MPC cannot be applied directly to the power sources on board, a second layer of control will be required to ensure each component follows MPC commands. For the engine, the sliding mode approach is used for controlling the engine torque and emissions; for the simplified model of the electric drive, standard proportional-integral (PI) controllers are used. The direct torque control (DTC) approach for electric drive control will also be addressed. Indeed, this will only be possible after replacing the simple electric drive model with a high-fidelity power electronics model.

1.3.3 Control Scheme Evaluation

After the design stage, the performance of the control scheme needs to be evaluated. This task can be accomplished by using the following strategies: comparing MPC performance with a global optimization method; model-in-the-loop (MIL) simulation; and hardware-in-the-loop (HIL) simulation.

For the first part, dynamic programming (DP) is considered, which has been used extensively in the literature to find a global solution for HEV control strategies. Unfortunately DP cannot be implemented on-line so it will only be used as a benchmark for developing heuristic strategies. In the context of optimal control, DP and Pontryagin's minimum principle (PMP) are two different approaches to obtain optimal trajectories for deterministic optimal control problems. In the minimum fuel consumption problem of HEVs, the DP method guarantees a globally optimal solution by detecting all possible control options [14].

For a better estimation of control scheme performance in the real world, MIL simulation is considered by using a validated high-fidelity simulation model of a Toyota Prius plug-in hybrid developed in the MapleSim software. A distinguishing feature of this model is that it contains a chemistry-based model of the battery. Most of the battery models used in the literature are based on a simple circuit (voltage source and internal resistor) or look-up tables, whereas, in the high-fidelity PHEV simulation model, a more realistic physics-based model of the battery is included.

Upon completing the MIL simulation, we will proceed to develop an HIL system. At this stage, an electronic control unit (ECU) is programmed based on the proposed controller, and the energy management system performance is evaluated.

Controller parameters will pass through different stages of fine-tuning during each of these three simulation procedures.

1.4 List of Contributions

The proposed research contributions can be summarized as follows:

- Developed real-time, equation-based, and cross-validated high-fidelity simulation model of a PHEV powertrain.

To validate the proposed control scheme, a high-fidelity simulation model is required that includes all essential features of the real plant. Since this high-fidelity simulation

model is used as the virtual simulation model, it should be real-time capable so it can be used in the HIL control validation procedure.

This model is developed in the MapleSim software to use the symbolic programming capability of Maple. Solving a large number of differential-algebraic equations involved in the equation-based model of the powertrain can be cumbersome. However, the number of equations can be significantly reduced using the symbolic simplification in MapleSim. This enables the model to be run in real time.

One distinctive feature of the developed PHEV powertrain model is the chemistry-based model of the battery as opposed to the existing look-up table or circuit-based models of the batteries in the literature. This leads to more realistic estimation of PHEV fuel economy and range. The parameters of this model are identified using the experimental data base of the Autonomie software [15], which is widely used for energy management design in industry.

Since the high-fidelity simulation model is equation-based, it can be used for performing sensitivity analysis, optimization and systematic model reduction techniques to obtain control-oriented models.

- Model predictive control design and evaluation for a PHEV EMS.

In this research, a model-based strategy is proposed with the use of MPC concept. The application of MPC to HEVs has already been investigated. Wang et al. [11] integrated the MPC controller with a proposed real-time control system. The system can be used for all types of hybrid architectures consisting of an engine and an electric motor. They used a number of different performance metrics to evaluate the performance of the control system. By changing the operational weights in the cost function, the power control system can achieve different goals. The research was limited to series and parallel configurations. Borhan et al. [10] applied MPC to a non-plug-in power-split HEV; however, they ignored the dynamics of the powertrain, retaining only the faster dynamics (such as battery dynamics) in the model inside the controller. They concluded that the fuel economy achieved with MPC are better than those reported by the rule-based PSAT (Power System Analysis

Toolkit) simulation software. In fact, MPC has not been used to design an energy management strategy for a plug-in power-split HEV: the goal we seek in this research. It should be noted that the PHEV powertrain is different from that of conventional HEVs in terms of the initial conditions and constraints. In a PHEV, the battery capacity is significantly higher and it can be charged from a source external to the powertrain. A PHEV battery can be fully charged before the vehicle is started; this is an impossible scenario for a HEV. In a HEV powertrain, the battery state of charge (SOC) should be maintained within a definite range (for instance, between 0.60 and 0.65 [10]) and the final SOC value at the end of the simulation should be the same as initial SOC [1]. In PHEVs, the battery is generally discharged from a high level, and when the SOC drops to a reference value, the control strategy attempts to maintain the SOC as close to that value as possible. This reference value is lower than what it is in an HEV. The strategies that are applied to HEVs can be implemented on PHEVs, but should be modified for optimal performance. Therefore, the energy management strategies for HEVs and PHEVs are two different problems with different constraints. Furthermore, the chemistry-based model of the battery in the high-fidelity simulation model helps to more realistically estimate the MPC EMS performance.

- Developed and validated near-optimal and real-time implementable PHEV EMS using explicit model predictive control (eMPC) approach with simple and innovative control-oriented model for simpler stability analysis.

Despite MPC's near-optimal performance in improving fuel economy while handling the constraints on the problem, there are some real-time implementation problems. As a result, the eMPC approach is used. As mentioned earlier, eMPC solves the optimization problem off-line from which some look-up tables can be generated, so the problem is reduced from solving a quadratic programming problem to searching in the look-up tables while implementing the control algorithm. Therefore, eMPC can guarantee a real-time implementation, but the size of the look-up tables may be a concern for accommodating the data base onto commercial control hardware with limited memory. The eMPC approach is appropriate for relatively small problems, whereas the dynamics governing a power-split hybrid electric powertrain put some limitations on the number of state variables that can

be handled by eMPC in order to result in a fairly-small sized data base. A solution is to perform current model reduction techniques to come up with a control-oriented model with fewer state variables, yet the resultant model may contain time-varying parameters which makes stability analysis more challenging. To address this problem, a simple and innovative control-oriented model is proposed. The designed eMPC EMS with the proposed control-oriented model can be implemented faster than real time and result in promising fuel economy figures. According to the model-in-the-loop simulation results, the designed eMPC energy management can be implemented to the high-fidelity simulation model three times faster that its implicit MPC counterpart.

- Developed control-relevant parameter estimated (CRPE) control-oriented model to improve performance of eMPC EMS while maintaining its real-time capabilities.

MPC performance is closely related to the accuracy of the control-oriented model that is used to predict the behavior of the plant. To obtain a simple but sufficiently accurate model of a PHEV powertrain for control, control-relevant parameter estimation approach is utilized. Since the PHEV performance and electric range is related to the on-board battery, the control-relevant parameter estimation is used to improve the battery model inside the control-oriented model. To this end, the active frequency range of the battery excited by the EMS is targeted to provide the experimental data for performing the parameter estimation procedure. It is noteworthy that the number of state variables of the mentioned model should be kept low enough so that it would not affect the real-time implementation capabilities of the designed eMPC EMS. Model-in-the-loop simulation showed that the CRPE-eMPC EMS can reduce the fuel consumption by up to 6.5% when compared to the eMPC EMS without a control-relevant model.

- Implemented the CRPE-eMPC energy management strategy on a commercial control hardware with limited computational and memory capabilities.

Hardware-in-the-loop is an essential stage in controls validation procedure. After many steps of refinement, CRPE-eMPC energy management was modified in such a way that

it could be implemented on a commercial control hardware, and tested in the real-world setup. In fact, this modification affects the performance of the CRPE-eMPC EMS. The aforementioned high-fidelity simulation model is used as the virtual simulation model inside the real-time target which is connected to the control hardware via CAN bus.

HIL simulation shows that the proposed EMS can be implemented to commercial control hardware in real time and results in fuel economy improvement of up to 16.5% to the baseline Toyota Prius plug-in hybrid EMS while controlling the emissions. Note that if the emissions control is not considered in the control scheme, the fuel economy improvement is expected to be even higher.

1.5 Thesis Layout

Chapter 2 reviews the literature on PHEV control. Chapter 3 introduces the validated high-fidelity simulation model of the Toyota Prius plug-in hybrid electric powertrain that will be used in MIL and HIL tests for the designed controls performance evaluation.

In chapter 4, the EMS is designed based on the MPC approach. This EMS performance is evaluated by applying to the low-fidelity and high-fidelity simulation models of the PHEV powertrain and benchmarked by a dynamic programming approach. To conduct the MIL test using the high-fidelity simulation model, appropriate low-level controls for the engine and the electric drive are designed.

Chapter 5 discusses the design procedure of EMS using eMPC approach in order to speed up the simulations. Also, a control-oriented parameter estimation method is introduced to improve the control-oriented model accuracy which leads to better performance of eMPC controller.

In chapter 6, the eMPC EMS is programmed on a MotoTron electronic management system hardware to perform HIL testing, followed by conclusions and future work in chapter 7.

Chapter 2

Literature Review and Background

Triggered by rising fuel prices, stringent legal norms and increasing environmental awareness of the customer, car manufacturers are producing vehicles with high fuel efficiency and low emissions. This is possible due to new components and technologies that are introduced in automotive powertrains such as turbo charging, exhaust gas recirculation, continuous variable transmission, electronic throttle control, and hybrid electric powertrain.

In this chapter, different hybrid electric powertrain architectures are introduced. Then, studies on model predictive control approach as well as control-relevant parameter estimation method for obtaining more accurate control-oriented model are reviewed. Finally, the work on controls evaluation procedure through hardware-in-the-loop simulation is investigated.

2.1 Hybrid Powertrain Architectures

This section briefly introduces some known architectures for HEVs - namely series, parallel, and power split. In series architectures, the engine is not directly connected to the wheels. Instead, an electric motor propels the vehicle using the energy stored in the battery. Upon depletion of the battery, the engine is used to turn a generator, which recharges the battery.

Figure 2.1 shows a schematic of this architecture. Since the engine is separated from the wheels, it can be controlled in such a way so as to operate mostly inside its sweet spot with the least fuel consumption and emissions. However, this architecture suffers from a relatively low efficiency in comparison with other structures for passenger cars because of a multi-stage conversion of energy.

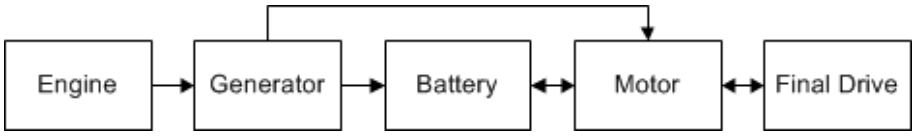

Figure 2.1: Series architecture schematic

In parallel architecture, the engine can provide part of the propulsion power directly. The electric motor assists the engine by means of mechanical coupling (see Figure 2.2).

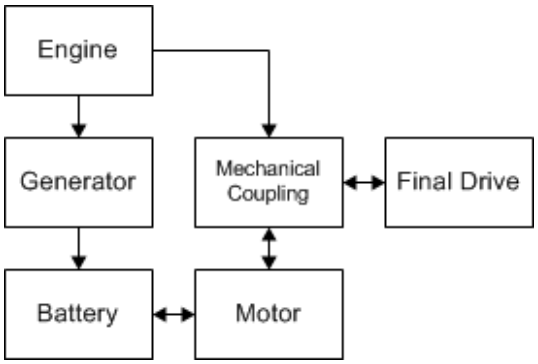

Figure 2.2: Parallel architecture schematic

Both of these architectures are capable of capturing part of the energy usually dissipated as heat while braking (regenerative braking). A power-split architecture combines the advantages of the two architectures mentioned above. In this case, the engine, generator, and motor are connected to each other by means of a planetary gear set (see Figure 2.3).

Among the different PHEV architectures, the power split has displayed the best fuel economy for urban driving conditions in comparison with series and parallel configurations

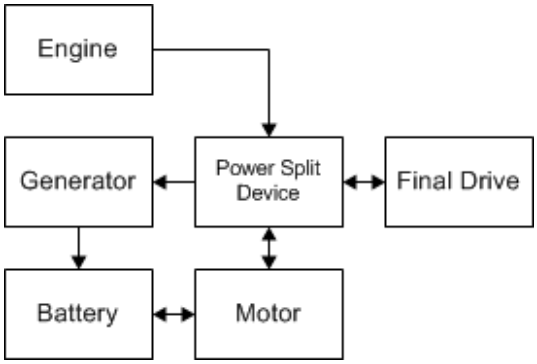

Figure 2.3: Power-split architecture schematic

[16]. In highway driving conditions, the power-split and parallel architectures showed, respectively, similar and better efficiency as compared with the series architecture.

2.2 Energy Management Systems

Designing the energy management system is as important as choosing the architecture of a hybrid electric powertrain; the best architecture may operate poorly if an inappropriate control strategy is used.

Over recent years, several strategies for HEV energy management have been proposed, including rule-based methods, dynamic programming (DP), stochastic dynamic programming (SDP), equivalent fuel consumption minimization (ECMS), and model predictive control (MPC). These strategies optimize the fuel economy by solving the problem where the fuel consumption explicitly appears in the cost function. Indeed, this will result in minimum fuel consumption if the full information problem can be solved over the whole driving cycle (as in the DP approach). However, information about the future driving cycle is not available during conventional driving. In addition, planning for the whole future driving cycle is computationally demanding. While the use of stochastic models (as in SDP and in stochastic MPC) alleviates some of these problems, the choice of stochastic model and its identification still poses several challenges [5].

One way to design an energy management strategy for a PHEV is to extend the strategies applied to non-plug-in HEVs [17].

On the other hand, there are two main strategies for PHEVs. The first is full electric drive plus charge sustaining mode. In this strategy, the battery, which is fully charged at first, provides a considerable range of full electric drive while the engine is off. This mode continues until the battery state of charge (SOC) drops to a predefined level. The engine then takes over and keeps the SOC as close as possible to that reference. The energy management strategy makes the engine operate efficiently with no deterioration in drivability performance. This mode continues until there is access to an electricity source in order to fully charge the battery again. It is noteworthy that the engine is allowed to start in full electric drive mode if the power demanded by the driver is more than what the battery and motor can provide. In this case, the strategy is charge depletion plus charge sustenance mode. Some studies have addressed battery health-conscious energy management scheme for PHEVs. For instance, Moura et al. [18] suggest that in order to minimize battery degradation, a PHEV energy management scheme should primarily deplete battery charge quickly, then blend the engine and battery power to avoid charge sustenance.

The second main strategy which is called blended mode, performs charge depletion mode throughout the driving schedule. The engine is started more frequently to reduce the rate of battery charge depletion. This strategy has the potential of improving fuel economy more than the full electric drive/change sustenance strategy, but it requires trip information.

Furthermore, several items must be considered in energy management system design including the choice of the correct objective function that should be minimized, forecasting the future load based on the information available at runtime, and the characteristics of the vehicle [19]. It should be noted that most of the strategies used in commercial HEVs are rule-based heuristics [20]. Rule-based approaches constrain the power split between different on-board power sources based on the current state of the powertrain (e.g., vehicle/engine speed, battery charge, and power demand) using maps or rule bases [21]. A set of rules can then be maintained to ensure that the states of the system are

as close as possible to the desired scheme. Those decisions can be made by using maps which can be constructed from engineering expertise and insight, or by using more formal methods such as optimization [22] or fuzzy logic [23]. The stochastic dynamic programming (SDP) method is quite appealing in this context because of its ability to optimize the system performance with respect to a probabilistic distribution of several different drive cycles [24]. However, this method has some computational complexities [25]. Moura et al. [26] derived an optimal energy management strategy for a PHEV (with a power-split architecture) based on stochastic dynamic programming, which rations battery charge by blending engine and battery power in a manner that improves engine efficiency and reduces total charge sustenance time. Freyermuth et al. [27] simulated and compared four different control strategies for a power-split PHEV with a 16 km All Electric Range (AER) battery pack. Using the Electric Vehicle/Charge Sustaining (EV/CS) strategy, the engine turns on only when the power demand is higher than the available power of the battery. The differential Engine Power strategy is similar to EV/CS, but the turn-on threshold for the engine is lower than the maximum power of the electrical system. Using the Full Engine Power strategy, if the engine turns on, it will supply all the power demand of the drive cycle and no power will drain from the battery. The aim of this strategy is to force the engine to operate at a higher power demand and, consequently, with a higher efficiency. The Optimal Engine Power strategy is similar to the previous strategy: it seeks to use the engine more efficiently (at a higher power demand) by restricting the engine operation close to the peak efficiency. Freyermuth et al. [27] conclude that EV/CS is equivalent to the Differential Engine Power strategy and Full Engine Power is the best of all (and much better than Optimal Engine Power).

More advanced control techniques are based on real-time optimization. Also referred to as causal systems, they rely on real-time feedback to optimize a cost function that is developed using the past information [17]. Route-based energy management algorithms require knowledge of future power demand, which is then used to specify the future power contribution of different sources of energy on board. This type of optimization can be performed off-line for drive cycles known a priori using deterministic dynamic programming (DDP) [28], and can also be performed on-line using optimal model predictive control [29].

Gong et al. [30, 31] suggest that it is possible to improve the control strategy of a PHEV if the trip information is determined a priori by means of recent advancements in intelligent transportation systems (ITS), which use data obtained from global positioning system (GPS) and geographical information system (GIS) infrastructure.

Unfortunately, it seems that the control software of powertrains fall short with respect to their complexity. While mostly strategies that are based on heuristics and look-up tables are implemented, it was shown that model predictive control has a large potential for control of automotive powertrains ([32, 33, 34, 35, 36]).

2.3 Hybrid Powertrain Modeling

The development procedure of HEVs needs a framework with a validated model to simulate emissions and fuel consumptions. This framework can be used for benchmarking the fuel economy of alternate hybrid powertrain technologies, component sizing, and performance evaluation of various supervisory control algorithms for energy management [37].

Designing such a control system requires a sufficiently accurate hybrid powertrain system plant model [38]. In terms of plug-in hybrid powertrain, the battery determines the vehicle full electric range and the emissions performance of the vehicle. Therefore, it is essential to have an accurate model of this component.

Development of the powertrain management system can be time-consuming, since the process relies on extensive testing of new control designs and calibration of control hardware [39]. Using model and hardware-in-the-loop (MIL/HIL) simulations makes the control evaluation and validation procedures simpler, especially at the early stages of controls design.

Furthermore, the control design usually involves a large number of iterations and evaluations over long driving cycles. Therefore, developing a fast-running model of hybrid powertrains, especially the engine and battery pack, with sufficient accuracy for control applications is necessary.

There are different approaches to find a suitable model for control evaluation purposes. Some models are based on look-up tables, derived from experiments. Although these models cannot capture transient behavior of the real plant, they are useful for running steady-state simulations. Some of the mentioned tools are ADVISOR [40], HE-VESIM [41], and PSAT [42].

Patil et al. [37] described a simulation framework to predict fuel economy of a series hydraulic hybrid vehicle for any drive cycle, developed in MATLAB/Simulink. The parameter data is chosen to represent a Class 6 medium-duty parcel delivery truck. A complete plant model of a power-split HEV is developed in [38], which is also validated against experimental results on the Ford test track. Kim et al. [43] developed a vehicle model and a controller for the 3rd generation Toyota Prius to reproduce real-world behaviors, and the simulation results were compared with the testing results. They showed that the developed vehicle model achieves fuel consumption that is close to the testing value, within 5%, and the operation of the engine model was similar to that of the real-world engine.

2.4 Model Predictive Control

Model predictive control (MPC) is an advanced control strategy which determines inputs for a given process that optimize the forecast process behavior with respect to a cost function. These inputs, or control actions, are calculated repeatedly at each sampling time using a dynamic process model (control-oriented model) designed for the prediction. To this end, the fast and reliable solution of quadratic programming (QP) problems in real-time becomes a crucial ingredient of most MPC algorithms. If the process model is linear and the cost function quadratic, a setup called linear MPC, the computational cost for finding near-optimal control actions even reduces to the solution of one single convex QP problem at each sampling instant [44].

One must be warned, that there exists a tradeoff between model accuracy and complexity of the optimization: the simpler the model (and performance index/constraints), the easier the solving of the optimization would be. Henceforth, while in building simulation

models one looks for the most accurate model to numerically reproduce the behavior of the process as faithfully as possible. However, prediction models used in MPC (as well as in any other model-based control design techniques, from pole-placement to Bode diagrams, etc.) are usually very simple, yet representative enough to capture the main dynamical relations.

In particular, complex non-linear systems can be approximated by linear and piecewise linear models whose associated optimal control problem can be solved by relatively simple numerical procedures. System identification is an excellent tool to obtain the most representative prediction model within a prescribed bound of model complexity [45].

The reason for the success of MPC in industrial applications is due to its ability to handle processes with many manipulated and controlled variables and constraints on them in a rather systematic manner [46]. Furthermore, MPC allows for the specification of an objective function which is optimized by the controller. Other advantageous MPC features are the capability of dealing with time delays [47], of rejecting measured and unmeasured disturbances [48], and by taking advantage from future information [49]. Finally, there is a philosophical attraction to MPC since it embodies both (receding horizon) optimization and feedback adjustment. In fact, model predictive control [50] has been developed to integrate the performance of optimal control with the robustness of feedback control. The main appeal of MPC is in being able to enforce pointwise-in-time constraints, while providing the control designer with direct capability to shape the transient response by adjusting the weights in the objective function being minimized. MPC controllers can handle continuous-valued and discrete-valued control inputs, accommodate system parameter changes or subsystem faults, as long as they are reflected in the prediction model [51].

Over the past few years, smart electronic devices that monitor and control the mechanical components have enabled major advances in automotive applications. Nowadays, cars have become complex systems in which electronic and mechanical subsystems are tightly connected and interact to achieve optimal performance. Automotive actuators have become mechatronic systems in which mechanical components coexist with electronics and computing devices. These mechatronic automotive systems are characterized by tight operating requirements such as high precision robustness, low power consumption,

fast transition time, significant non-linearities, as well as input and state constraints which need to be enforced during system operation. On the other hand, their dynamics may often be characterized by relatively low-dimensional dynamical models [51]. This has not only caused an increased interest in MPC for automotive applications, especially for drive line, but also for engine control [52]. In addition, MPC is suitable because of its receding horizon implementation that mimics human driving behavior [53].

There are several examples of using MPC approach for automotive applications in the literature including vehicle traction control [54], suspension [55], direct injection stratified charge engines [56], automotive powertrains [36, 57], magnetically actuated mass spring damper system [51], power converters [58, 59], and so on. Recent applications of MPC for powertrain control include control of diesel engines [52, 60], catalyst control [61, 62], transmission control [63], powertrain actuator control [51, 64], and hybrid electric powertrain EMSs [65].

Di Cairano et al. [47] have proposed a design flow to develop model predictive controllers targeted to automotive applications, that makes use of simulation model and experimental data to tune the different parameters of the controller. The procedure has been applied to an engine idle speed control. An improved idle speed controller can reduce the need for a spark reserve, and hence consistently improve the fuel efficiency. The results show that the MPC largely outperforms the available baseline controller.

Model predictive control (MPC) is widely used in the lateral control of vehicles. As previously mentioned, the effectiveness of the MPC is in the very easy and efficient handling of linear systems with states and control constraints. In the case of vehicle lateral dynamics control, limitations on steering angle and steering angle rate are typical constraints. By introducing appropriate weightings for states and control and by solving a QP problem, one can easily obtain an MPC controller [66].

Falcone et al. [49] proposed a non-linear MPC approach for coordination of active steering and braking in an autonomous vehicle navigating along a known trajectory.

2.5 Explicit Model Predictive Control

Classical model predictive control (MPC) requires on-line optimal solution of linear or quadratic programming problem at each sample time, using the current states. Such on-line computation is the main disadvantage that limits applications of MPC in general only for controlling slow processes [9]. For instance, the first generation of MPC algorithms was aimed at solving multi-variable constrained control problems typical to the oil and chemical industries [67]. However, computation speed is not the only limitation: the code implementing the solver might generate concerns due to software certification issues, a problem which is particularly acute in safety critical applications [46].

Explicit MPC (eMPC) techniques [68] can be used to synthesize the controller as a piecewise affine function. With this approach appropriately applied, the MPC can be implemented in a micro-controller without the need for an optimization solver and satisfying the stringent memory and chronometric constraints of automotive electronic control units (ECUs) [51].

eMPC allows one to solve the optimization problem off-line for a given range of operating conditions of interest. By exploiting multi-parametric programming techniques, eMPC computes the optimal control action off-line as an "explicit" function of the state and reference vectors, so that on-line operations reduce to a simple function evaluation. Such a function is piecewise affine in most cases, so that the MPC controller maps into some polyhedral regions that can be stored as a look-up table of linear gains [46].

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

Beside eMPC advantages, this approach suffers from some drawbacks. In practice, eMPC is limited to relatively small problems (typically 1-2 inputs, up to 5-10 states, up to 3-4 free control moves) but allows one to reach very high sampling frequencies (up to 1 MHz) and requires a very simple control code to be embedded in the system [46].

For a fixed number of states and reference signals, the complexity of the solution is given by the number of regions that form the explicit solution. This number mainly depends (exponentially, in the worst case) on the number of constraints (and also of binary variables/system modes in the hybrid case) included in the MPC problem formulation, and only mildly on the number of state variables. It also depends on the number of optimization variables. Bemporad [45] has introduced a table which shows such dependencies on random linear MPC problems. In the multi-parametric QP case, an upper-bound to the number of regions is 2 to the power of constraints number, which is the number of all possible combinations of active constraints at optimality [46].

One problem with the approach is that, as the horizon size, the number of states, and the number of constraints grow, the number of polyhedral regions grows quickly, making the look-up table approach difficult to implement in practice. Therefore, various eMPC design and search algorithms with some minor sacrifice of optimality were proposed [67].

Industrial problems addressed through eMPC techniques have been reported in technical papers, starting from what is probably the first work in this domain (traction control) [69]. The most suitable applications for eMPC are fast-sampling problems (in the order of 1-50 ms) and relatively small size. Most of the applications of eMPC have been reported in the automotive domain and electrical power converters.

Borrelli et al. [54] describe a hybrid model and a MPC strategy for solving a traction control problem. The problem is tackled in a systematic way from modeling to control synthesis and implementation. The resultant optimal controller is converted to its equivalent piecewise affine form by employing multi-parametric programming techniques, and finally experimentally tested on a car prototype. Nausa et al. [53] describe a systematic approach for the design of a parameterized adaptive cruise control (ACC), based on eMPC. A unique feature of the synthesized ACC is its parameterization in terms of key characteristics, which, after the parameterization, makes it easy and intuitive to tune, even for the driver.

Stewart and Borrelli [60] present the development and implementation of a practical eMPC approach that allows sub-controllers to receive and accommodate time-varying

setpoints and constraints from higher levels in standard industrial automotive controller hierarchies. The proposed approach was demonstrated on production ECU controlling a real 2.2 litre diesel engine in which the variable-geometry turbocharger (VGT) and exhaust gas recirculation (EGR) actuators were used to track setpoints on mass air flow (MAF) and manifold absolute pressure (MAP) sensors while respecting a time-varying constraint on engine-out NOx emissions. Widd et al. [70] present results on model predictive control of the combustion phasing in an homogeneous charge compression ignition (HCCI) engine based on a hybrid model formulation composed of several linearizations of a physics-based non-linear model. The explicit representation of the MPC was implemented experimentally and the performance during setpoint changes was compared to that of a switched state feedback controller. The hybrid MPC produced smoother transients without overshoot when the setpoint change traversed several linearizations.

Di Cairano et al. [5] have introduced an energy management strategy that focuses on optimizing the engine efficiency for a series HEV. The experimental results executed on a fully functional vehicle on a chassis-roll dynamometer using the UDDS cycle, show fuel economy improvements with respect to two base strategies.

2.6 Control-relevant Parameter Estimation

Effective system identification of highly interactive processes for multi-variable control purposes has been viewed as a challenging problem by many investigators [71, 72, 73]. Recently, control-relevant parameter estimation has become an important subject to control researchers, because of its potential to make significant improvements in the field of model-based control. It is suggested that model identification and controls design should not be performed independently, and this has led to the iterative design of a model-based controller. While conventional identification approaches emphasize obtaining an accurate model, the objective of the control-relevant identification is to find an approximate model that is appropriate for the design of a high-performance controller.

The goal of control-relevant parameter estimation is to make appropriate choices of

design variables in the estimation procedure so that the important properties of the plant with respect to the intended control application are retained in the estimated model [74]. Control-relevant parameter estimation problems naturally arise in reduced-order controller design ([75, 76]) and system identification [77]. Various control-relevant parameter estimation schemes ([78, 79, 80, 81]) have been proposed and also achieved good results in several industrial applications ([82, 83]).

Among them, most of the control-relevant identification schemes adopt a time domain identification technique and use frequency weighting functions in both identification and controller design stages ([83, 84, 85]). The frequency domain weighting function in the controller design stage is obtained from differences between estimated and predicted input/output signals during iteration. The cost function is updated at every iteration step and the results of identification considerably affect the controller gain tuning, which means that the cost function is changed at every step to minimize the cost function. Jun et al. [86] describe a MATLAB-based computer-aided design tool, IRA-HPC, which accomplishes integrated system identification and robustness analysis for receding horizon control (RHC), a model predictive control algorithm implemented on the Application Module of the Honeywell TDC 3000 distributed control system and shows its benefits in terms of simplifying the choices of design variables in integrated identification and control design. Lee and Rivera [74] has presented a novel integrated framework for multi-variable system identification and control system design leading to desirable models for the control of highly interactive multi-variable process systems. The resulting models serve as a useful nominal model for a high performance advanced control system, such as model predictive control. Verboven et al. [87] presents a computational approach for the frequency-domain identification of multi-variable, discrete-time transfer function models based on a cost function minimization. The algorithm is optimized for the parametric characterization of complex high-order multi-variable systems requiring a large number of model parameters, including sparse matrix methods for the reduction of computation time and memory requirements. The algorithm supports a multi-variable frequency-dependent weighting, which generally improves the quality of the transfer function model estimate. The overall approach is successfully demonstrated for a typical case encountered in experimental structural dynamics

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

2.7 Hardware-in-the-loop Simulation

Hardware-in-the-loop (HIL) systems have become efficient tools for strategy and interface software development. These HIL simulations have given the test engineer the ability to simulate a variety of scenarios that may be too difficult, time consuming or expensive to do on a vehicle prototype. The HIL systems allow control function development to be done and verified ahead of a vehicle build. The HIL systems are also used in the development of reasonable initial calibration values for the embedded software. The improved software quality and early verification of software leads to reduced vehicle commissioning time, which refers to the process whereby an early vehicle prototype is put together and each function is verified at a gross level, to see if a minimum level of functionality exists before being handed off to the various engineering teams for further development [6]. Automotive systems typically contain numerous feedback loops, each consisting of a physical system or plant to be controlled, related sensors, actuators, a controller, and a setpoint or more generally, a desired state trajectory. The controller is in fact a computer that uses input data from the sensors, and calculates appropriate controls which runs the actuators in such a way that the plant follows the desired state trajectory. Since the related dynamic equations are generally complex, it is rarely possible to find closed-form solutions.

Carrying out simulation studies within the virtual world of a desktop computer is a predominant solution to this problem. However, such off-line model-in-the-loop simulation provides no guarantee that the real-time performance constraints can be met in a sampled-time embedded computer with real-world I/O. Therefore, the HIL technique is employed to validate if a controller can meet the real-time performance requirements. In this technique, either a real plant to a simulated controller or a simulated plant is wired to a real controller. In either case, the presence of I/O and wires is the distinguishing feature. Simulation and control start with models of hardware because the actual hardware is not yet available.

Later, the models of that hardware can be removed and the actual hardware is connected. Thus, the hardware can be tested on-line several times as early as possible in the design process. Moreover, HIL requires dynamic models that include all significant interactions that exist within a particular plant [88].

The advent of microprocessor-based electronic control units (ECUs) for car engines and powertrain created a need for new tools for testing, calibrating, and validating these ECUs. HIL simulation met this need, and became a key technology for engine ECU testing and calibration [89]. Wagner and Furry describe a virtual simulation environment that enables such HIL simulation [90]. Kimura and Maeda use such HIL simulation for engine ECU development [91]. Lee et al. show the requirements that a HIL simulator must satisfy to be effective for ECU development [92]. They also present a formal process for developing such a HIL simulator that uses automatic code generation to streamline the transition of control system designs from pure simulation to a commercial embedded code [93]. Song and Grigoriadis utilize HIL simulation to validate the design of an advanced linear parameter-varying engine control system [94].

The use of HIL simulation for automotive ECU development is not limited to engine applications. In fact, HIL simulation has been used effectively for the development, calibration, and validation of transmission and driveline electronic control units. For instance, the authors in [95, 96] describe the process of developing a HIL simulator for the purpose of transmission ECU development. Schupbach and Balda [96] demonstrate the versatility of HIL simulation by using the same HIL setup to simulate the transmissions and drivelines of several distinct vehicles. Many transmissions use hydraulics as a means for actuation. Ferreira et al. develop a HIL simulation setup for a hydraulic system, laying particular emphasize on the inherent numerical stiffness of hydraulics models and the proper modeling efforts necessary for simulating such models in real time [97, 98].

Trigui et al. [99] applied the HIL approach to a parallel HEV configuration in order to analyze fuel reduction benefits due to hybridization without any influence of vehicle characteristics or engine technology improvement. Winkler et al. [100] showed a brief overview of the simulation of an HEV using the Micro Hybrid architecture which allows the use of Start/Stop operation as well as brake energy regeneration via the object-oriented

modeling language Modelica. The authors presented some simulation results using the new European driving cycle (NEDC) to show the impact of special sensitivity curves for control of the battery SOC. Hao et al. [101] presented a design procedure of HIL for electric vehicle powertrain including BMS (Battery Management System), MCU (Motor Control Unit) and VMS (vehicle management system), using dSPACE with MATLAB/Simulink, along with the RTW toolbox (real time workshop). A HIL verification environment for simulations of heavy-duty hybrids with full-scale hybrid electric power system is presented in [102]. This test facility has been used to measure the efficiencies of electric motors and hydraulic pumps, to characterize devices and validate plant models, and to make experimental HIL simulations in order to verify and validate HEV control algorithms.

2.8 Chapter Summary

In this chapter, the main architectures for a PHEV were introduced together with what have been done for designing an EMS for such a powertrain in the literature. The MPC and eMPC approaches were briefly reviewed to show their potential for designing optimal EMSs. Control-relevant parameter estimation was introduced and its advantages in terms of obtaining more accurate control-oriented models were mentioned. Finally, HIL testing importance for controls validation in the automotive industry was explained.

Chapter 3

High-fidelity Modeling of a Plug-in Hybrid Electric Powertrain

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

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

3.1 Introduction

A high-fidelity simulation model of the Toyota Prius plug-in hybrid powertrain developed in the MapleSim software is introduced in this chapter. MapleSim is an environment for multi-domain system simulation with direct access to system equations to make it more convenient to do sensitivity analysis, model reduction and optimization. Symbolic calculation and optimized code generation in MapleSim reduces simulation time and makes

the model more suitable for MIL and HIL tests. This physics-based model of a power-split plug-in powertrain contains a chemistry-based Lithium-ion battery pack, which can be distinguished from other models used in the literature since the performance of a PHEV greatly depends on its battery.

Developing a proper battery model, in which both simplicity and accuracy is retained, is one of the most challenging tasks. To enhance the fidelity of the model, a full chemistry-based Lithium-ion battery model developed by Newman and Tiedemann [103] and Doyle et al. [104], then simplified by Dao et al. [105], is used in this research. The model is an isothermal battery model based on concentrated solution theory, porous electrode theory, and the variations in electronic/ionic conductivities and diffusivities. The equations governing the dynamic behavior of the battery model are non-linear partial differential algebraic equations (PDAEs), which are not suitable for real time applications. As a result, a simplified version of the full battery model presented by Dao et al. [105], which preserves the accuracy and significantly reduces the computational cost, is utilized in this thesis. In their simplification procedure, Dao et al. exploited the nature of the battery equations along with a combination of several techniques such as volume-averaging, Galerkin's method, and curve-fitting. The resulting equations consist of 14 non-linear differential algebraic equations (DAEs) that carry both accuracy and simplicity for the battery simulation model. Developing any simulation model of a physical system requires parameters that should be identified with an admissible degree of accuracy. Accordingly, parameter identification of the Lithium-ion battery plays a key role in evaluating this electrochemical subsystem of the full vehicle model. Among parameter identification techniques, the homotopy optimization procedure presented by Vyasarayani et al. [106] seems to be very reliable due to its capability in approaching to global extremum.

To do cross-validation of the MapleSim model, the Autonomie software of Argonne National Lab has been used. Autonomie provides reliable component models that have been validated from a number of testing results [43]. It includes forward-looking models that allows advanced powertrain designers to develop realistic control strategies and assess component behaviors in a system environment by using models that are close to reality [42].

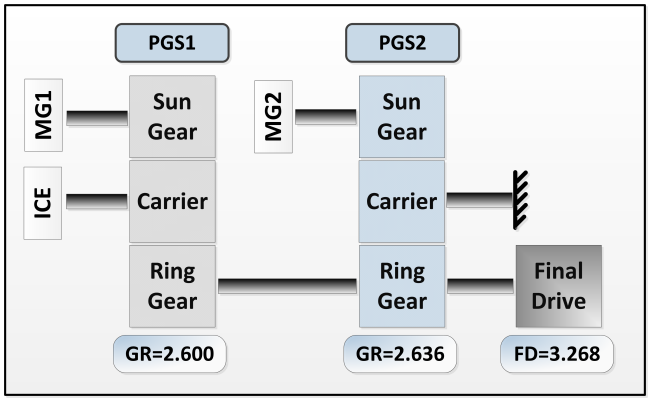

Figure 3.1: The transmission of Toyota Prius plug-in hybrid

3.2 Toyota Prius Plug-in Hybrid Powertrain

The powertrain of this vehicle is quite similar to the 3rd generation Toyota Prius except for the battery pack. The larger Lithium-ion battery pack provides longer full electric driving range while reducing the environmental footprint considerably. As usual, there are 2 electric motors (MG1 and MG2) which are connected to the engine and final drive with 2 planetary gear sets. The schematic of the power-split device [107] is shown in Figure 3.1. Note that GR is the ratio of ring gear teeth number to sun gear teeth number in each planetary gear set.

One of the two planetary gear sets splits the power flow from the engine like the 2nd version, but an additional gear is operated as a reduction gear for the motor [43]. MG1 and the engine are connected to the sun gear and carrier of the first planetary gear set, respectively [107]. The engine of this vehicle runs on the Atkinson cycle. To reduce mechanical loss, the engine is beltless and its water pump runs on the electric side of the powertrain similar to the air conditioning compressor. The engine main specifications are shown in Table 3.1 [108]. A PHEV might have subsequent engine stop-start events during its trip. Therefore, an exhaust heat recirculation system is designed to heat up the engine coolant quickly to its normal operating temperature in order to prevent cold start events, thereby improving fuel economy.

Table 3.1: Engine specifications

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

Table 3.2: MG2 specifications

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

The generator (MG1) and the traction motor (MG2) are air-cooled permanent magnet AC synchronous machines. MG1 can start the engine and charge the battery with the maximum power of 42 kW. The MG2 specification is demonstrated in Table 3.2 [108]. The addition of speed reduction planetary gear set requires the MG2 to operate at high speed levels.

The Prius Plug-in hybrid contains a newly-developed large capacity Lithium-ion battery pack (Table 3.3)[108]. This battery has two levels of output power depending on the vehicle driving mode. The electric mode (EV) range is up to 18 km according to Environmental Protection Agency (EPA) rating. After partial battery depletion, the energy management strategy switches to the hybrid mode (HV).

Table 3.3: Battery-pack specifications

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

3.3 MapleSim model

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

3.3.1 Mean-value Internal Combustion Engine

In the area of engine modeling, different model complexities are created for different applications [109, 110]. The engine modeling technique that most researchers have used for controls development has been a mean-value engine modeling approach [111, 112]. The PHEV high-fidelity simulation model uses a mean-value engine model which is accurate and fast enough for powertrain simulation. The mean value engine model consists of 4 parts: engine control unit (ECU), throttle body, intake/exhaust manifold, and combustion chamber. The throttle body receives the throttle angle command from the ECU and changes it to the air flow entering the intake manifold. In fact, all air that passes the throt-

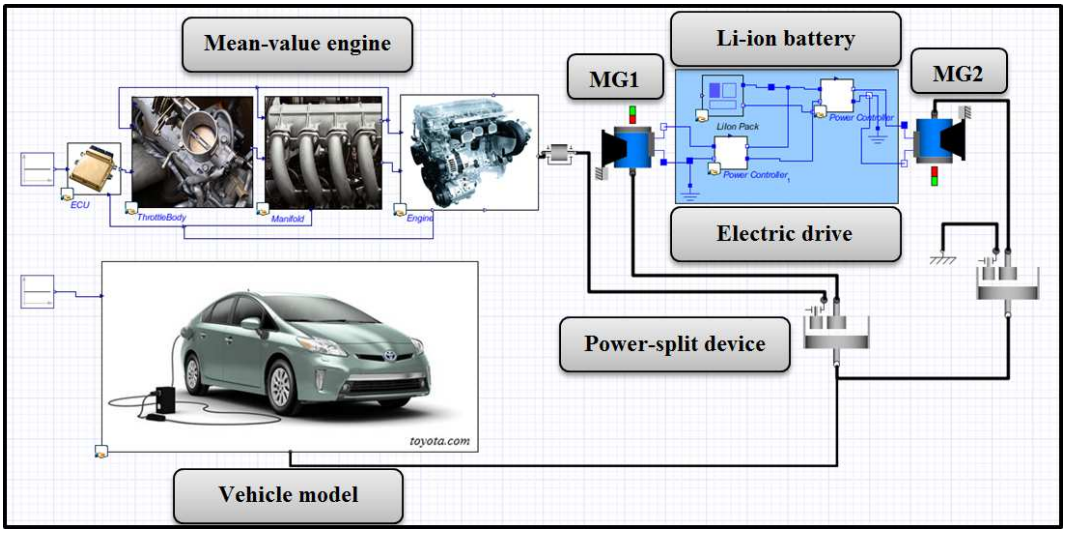

Figure 3.2: Toyota Prius plug-in hybrid high-fidelity simulation model in MapleSim

tle cannot enter the combustion chamber. The amount of air depends on intake manifold pressure and engine speed for naturally aspirated engines like what we have here. This effect is referred to as engine volumetric efficiency. In the combustion chamber, the power generated is calculated according to the air fuel ratio, mechanical, and thermal losses.

The first dynamic equation (3.1) is related to the manifold pressure [113]:

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

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

$$\begin{aligned} A_{th} = \tfrac{d_{th}.D_{th}}{2}(-\sqrt{1-(\tfrac{d_{th}}{D_{th}})^2} + \sqrt{1-(\tfrac{d_{th}cos\theta_0}{D_{th}cos\theta})^2}) \\ + \tfrac{D_{th}^2}{2}(sin^{-1}(\sqrt{1-(\tfrac{d_{th}}{D_{th}})^2}) - \tfrac{cos\theta_0}{cos\theta} sin^{-1}(\sqrt{1-(\tfrac{d_{th}cos\theta_0}{D_{th}cos\theta})^2})) \end{aligned} \tag{3.2}$$

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

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

where γ is air heat capacity ratio.

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

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

The engine generated torque can be estimated via:

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

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

The mean value engine model can be used to predict the indicated torque and fuel consumption based on throttle angle, air/fuel ratio and ignition timing. The main concern is to adjust the parameters in order to match the fuel consumption map of the mean-value engine model to the real Toyota Prius plug-in hybrid engine, which will be addressed later

in this chapter.

3.3.2 Electric Machines

There are components available in MapleSim to model the high-fidelity power electronics along with permanent magnet AC synchronous machines. But considering the full model of these components will increase the simulation time, and the powertrain model will no longer run in real time. As a result, we replace PM synchronous machines with DC motors with the same power rating. It is assumed that the electric drive is an ideal DC-DC converter, but the electric parts of the powertrain are modified to consider the associated efficiencies.

3.3.3 Lithium-ion Battery Pack

The Lithium-ion battery used in the simulation model of the PHEV is developed in MapleSim as a custom component by which the inputs and outputs are related using an acausal representation. The differential-algebraic equations are introduced into the custom component block along with the inputs and outputs required in the design of the control strategy for the PHEV simulation model. The differential equations are in the form of

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

where $\boldsymbol{E}$ is the state matrix, which is singular due to algebraic constraints, $\boldsymbol{x}$ is the state vector, $\boldsymbol{y}$ is the output vector, $\mathbf{f}$ is the column vector of non-linear functions, $\boldsymbol{B}$ is the input matrix, and $\boldsymbol{u}$ is the input to the simulation model, which is the current passing through the battery, i_{batt}. There are five differential equations and nine algebraic equations that should be solved simultaneously in each step of simulation. The battery state variables and parameters have been defined in [105]. The output vector $\boldsymbol{y}$ can be battery voltage, V_{batt}, and/or state of charge, SOC, depending on the control strategy.

3.3.4 Power-split Device

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

3.3.5 Vehicle Model

The vehicle model has 14 degrees of freedom (DOF), including 6 DOF for the chassis. The 4 suspension displacements and 4 wheel spins add 8 DOF to the model. Moreover, this vehicle is capable of being steered. Tires on this vehicle are modeled according to the magic formula by Pacejka including rolling resistance. Although the longitudinal dynamics of the vehicle is the most important DOF for assessing fuel consumption, the hybrid powertrain final drive is connected to the wheels of this vehicle model to make the simulation represent the full 3D vehicle motion and cover different maneuvers possible in standard drive cycles. The aerodynamic drag force is simulated using an external load acting on the vehicle's center of mass.

3.4 Model Validation

For validating the Toyota Prius plug-in hybrid high-fidelity simulation model, the parameters of the model should be identified using the experimental data available for each component. Most of the validation procedure was done using the Autonomie software experimental data base, which is widely used for energy management design in industry. Autonomie is the updated version of the PSAT software which was developed at Argonne National Laboratory. Autonomie is developed using MATLAB/Simulink and is a forward-looking model which simulates vehicle fuel economy, emissions, and performance in a realistic manner and employs a virtual driver who compares the trace speed and the actual vehicle speed and controls the vehicle with a torque input [114]. The schematic of a PHEV powertrain in Autonomie is shown in Figure 3.3.

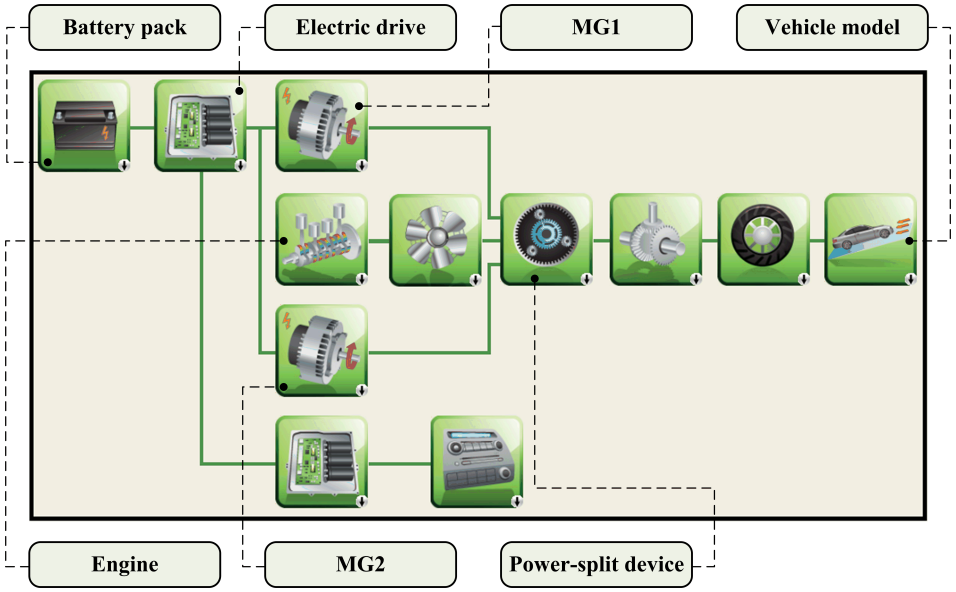

Figure 3.3: PHEV powertrain model in Autonomie

3.4.1 Mean-value Internal Combustion Engine

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

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

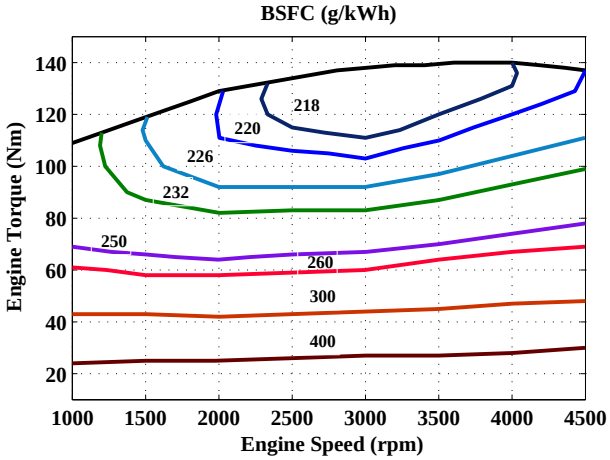

Figure 3.4: The engine brake specific fuel consumption (BSFC) map

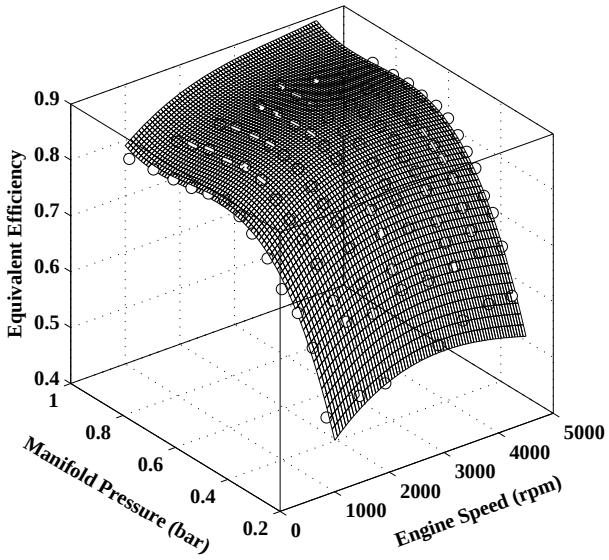

Figure 3.5: Equivalent thermal efficiency for different values of engine speed and manifold pressure

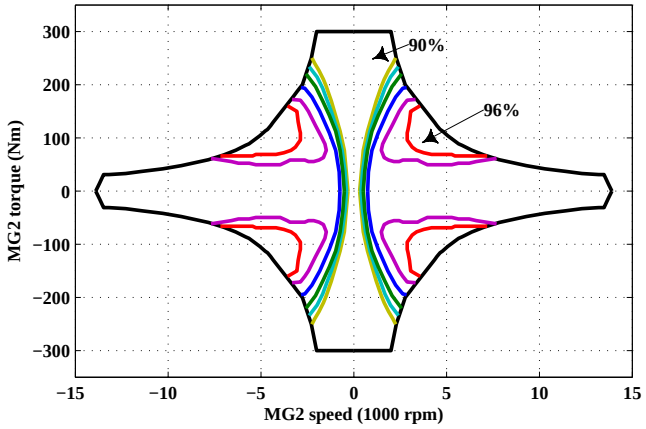

Figure 3.6: MG2 efficiency map

3.4.2 Electric Machines

As mentioned earlier, DC machines for modeling MG1 and MG2 are used. The parameters available for DC machines are inductance and resistance of the armature that only affects the transient response of the machine and also the torque and speed constants. There is no other model parameter to change the efficiency of the machine. To consider the efficiency map [107] of the traction motor (MG2), which is shown in Figure 3.6, an energy dissipation element (a mechanical brake) is added in connection of the machines to the power-split device. If the brake activation signal follows $\sigma T_m(1-\eta_m)$ the efficiency map given in Figure 3.6 can be reproduced in the MapleSim high-fidelity simulation model. Note that T_m is the motor torque, α is a constant related to the geometry of the brake, and η_m is the motor efficiency.

3.4.3 Lithium-ion Battery Pack

To have a more realistic battery model, the parameters for the model should be estimated based on experimental data, which in this chapter were extracted using the parameters of the full-order rigorous model used by Dao et al. [105]. In general, converging to local

minima is one of the most common issues when deterministic methods are utilized in optimization process of parameter identification. The homotopy method is an effective solution to obtain the global minimum of the optimization problem [106]. In homotopy optimization process, the original differential equations of the problem

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

in which ξ is the vector of state variables, $\mathbf{G}$ is the column vector of non-linear functions, and $\boldsymbol{\Gamma}$ is the column vector of the parameters to be identified, are modified by coupling the vector of experimental data, ξ_{exp}, to the original differential equation as [106]

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

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

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

with ξ^j and ξ_{exp}^j as the jth component of ξ and ξ_{exp}, respectively, is performed based on sensitivity equations used in evaluating the gradient and Hessian of the objective function during the identification process. The minimization procedure can be based on an iterative method such as Gauss-Newton algorithm, with a quadratic rate of convergency. Accordingly, the parameter vector will be updated based on a recurrence relation

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

in which κ denotes the step size, $\boldsymbol{g}$ is the gradient vector, and $\boldsymbol{H}$ is the Hessian of the objective function. The second term in the right-hand side is used as the search direction in the algorithm whose components are estimated using the following definitions [106]. For

gradient:

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

and Hessian can be approximated as

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

Starting from unity for υ and a large value of K_i, the experimental data and simulated response match for any set of parameters to be identified [106]. At the start of each iteration, the value of υ will be decreased by a specified decrement, then using the optimized parameters of the previous step as initial guess for $\boldsymbol{\Gamma}$, the parameters for the current are calculated. This process will be continued until $\upsilon = 0$, for which the optimized parameters are the same as the original equations. It makes the optimized parameters for different values of υ lie within the neighborhood of the ones corresponding to the global minima of the modified mathematical models, when approaching to the original problem for which $\upsilon = 0$.

The parameters selected from the full-chemistry battery model are the ones that are difficult to be estimated or measured, which is essential in parameter identification study. Volume fraction of separator region (ϵ_s), Li^+ transference number in the electrolyte (t_+), electronic conductivity of solid phase of electrode n (σ_n), and initial electrolyte concentration in region s, n, and p (c_{e0}) are identified based on the reference results for the battery voltage. The physical significance of the parameters can be found in the battery model developed by Dao et al. [105]. To apply the homotopy optimization procedure in identifying the battery parameters, the modified equations (3.8) need to be solved during the optimization process. It is important to note that consistent initial conditions, which satisfy the algebraic equations in the governing DAEs, play a key role in the accuracy of the solutions. Hence, the solver parameters should be elaborately adjusted so that the initial conditions can be updated in each step of the optimization stage for various values of the model parameters.

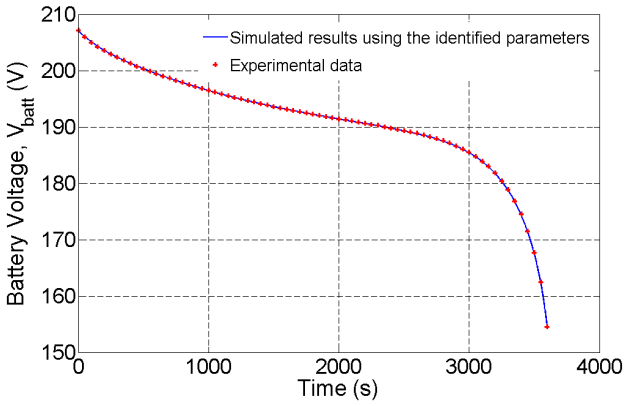

Figure 3.7: Time history of the simulated battery voltage for experimental data and simulated response using the identified parameters

Only under these conditions, the optimization procedure will converge and the optimized parameters lie within a reasonable range. The goal is to identify some of the battery model parameters, assuring that the difference between experimental and simulation results for the battery voltage is less than a desired tolerance ε. A constant discharge current is applied to the battery and simulation results versus experiments are shown in Figure 3.7. Figure 3.8 shows the convergence trend of the objective function during simulation process.

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

The value for the identified parameters are listed in Table 3.4. Based on the estimated parameters for the simplified model presented, the obtained values are in an acceptable range for the battery model. Including the identified parameters, the battery discharge voltage for various battery currents are shown in Figure 3.9, which signifies an appropriate dynamic behavior of the simulated battery model.

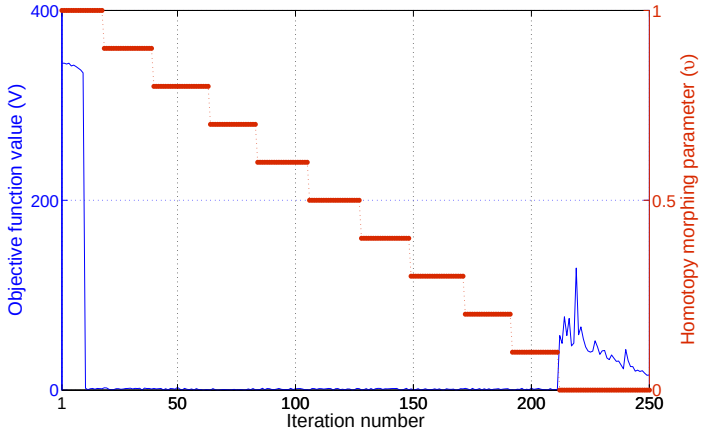

Figure 3.8: Convergence trend of the objective function during the optimization process

Table 3.4: Identified values for the battery parameters

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

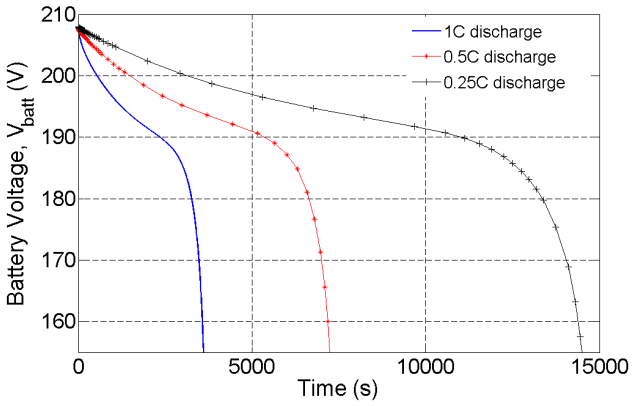

Figure 3.9: Time history of the battery voltage for various discharge rates

3.4.4 Power-split Device

The parameters related to the geometry of these gear sets such as gear teeth numbers and inertia are adjusted according to the real-world Toyota Prius plug-in powertrain.

3.4.5 Vehicle Model

Finally, the whole validated powertrain model can be simulated in MapleSim and compared to the Autonomie model. Figure 3.10 shows the MapleSim and Autonomie models simulation result based on a charge depletion / charge sustenance (CDCS) strategy along 2 successive UDDS drive cycles. In CDCS, the vehicle goes in pure electric mode first, so the battery is discharged from a high level, and when battery state of charge (SOC) drops to a reference value (30%), the control strategy tries to keep it as close as possible to that level. Indeed, if demanded power in the first part of the travel is more than what electric motor or battery can provide, the engine will compensate the remaining propulsion power. In this case, the Autonomie model shows 6% lower fuel consumption compared to the MapleSim model.

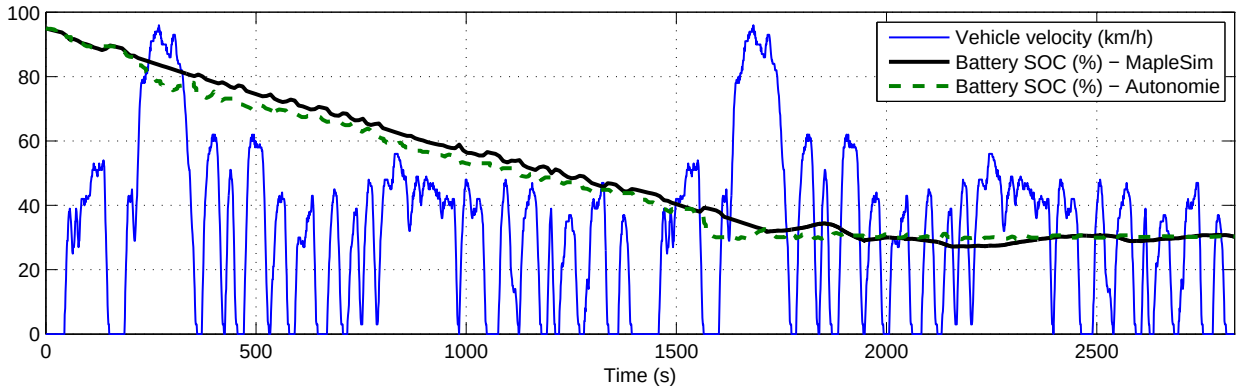

Figure 3.10: Powertrain model in MapleSim and Autonomie: simulation result along 2 UDDS drive cycles

3.5 Chapter Summary

In this chapter, a high-fidelity simulation model of the Toyota Prius plug-in hybrid powertrain was developed in MapleSim. To do cross-validation of the MapleSim model, the Autonomie software has been used. Furthermore, the homotopy optimization approach was applied to identify the parameters of the chemistry-based Lithium-ion battery pack model. Since the symbolic computation power of MapleSim can reduce the simulation time significantly, the developed model can be used to conduct model and hardware-in-the-loop simulations. Therefore, by developing the MapleSim high-fidelity PHEV model, one can perform controls evaluation and validation, or come up with a simple and sufficiently accurate control-oriented model to be used for model-based controls design with more confidence.

Chapter 4

Non-linear Model Predictive Control Design

In this chapter, an EMS is designed using the model predictive control approach. This energy management performance is evaluated by applying to the low-fidelity and high-fidelity simulation model of the PHEV powertrain and benchmarked by a dynamic programming approach. To perform MIL tests using the high-fidelity simulation model, appropriate low-level controls for the engine and the electric drive are designed.

4.1 Introduction

In this chapter, a model-based strategy for a PHEV is proposed with the use of MPC concept. MPC seems a proper method to exploit the potentials of modern concepts and to fulfill the automotive requirements since most of them can be stated in the form of a constrained multi-input multi-output optimal control problem and MPC provides an approximate solution of this class of problems [7].

In the first part of this chapter, different ways to design an energy management strategy for a PHEV based on different levels of trip information are addressed, and MPC is used to

design the strategy in the following cases: 1) No knowledge of trip information 2) Known travelling distance 3) Known entire trip information 4) Possibility of manually switching between electric mode and charge sustaining mode. Finally, the effect of weighting parameters inside the MPC cost function is investigated along with comparing fuel economy in different cases based on a UDDS drive cycle.

In the second part, a new control-oriented model is introduced in order to design a new supervisory controller for the PHEV high-fidelity simulation model along with engine and electric motor low-level controls.

For the engine low-level controller, a sliding mode control approach is considered to make the engine follow the MPC-prescribed torque trajectory while minimizing Hydrocarbon (HC) emissions, which results from multiple engine starts. This controller is robust and can guarantee a good performance in the real-world experiment. To design this energy management scheme, both CDCS and blended mode strategies are investigated. Finally, the performance of the proposed energy management scheme is evaluated by applying it to the high-fidelity simulation model. A more reliable trade-off between fuel economy and emissions using a near-optimal energy management scheme for both CDCS and blended mode strategies can be obtained.

4.2 Theory of Model Predictive Control (MPC)

4.2.1 Problem Formulation

The general design objective of MPC is to compute the trajectory of a future input to optimize the future behavior of the plant output. The optimization is performed within a limited time window based on the information of the plant at the start of the time window. If a quadratic objective function is used for the optimization, this defines a quadratic programming problem. To guarantee the best MPC performance, some requirements should be considered. The most important requirement is to have a sufficiently accurate yet simple enough model that captures all features of the original system behavior with a low

computational effort. The next step is to assess the measurable states and estimate the other states of the system. The last requirement is the instrument of implementing the planned activities namely electronic control unit.

A common notation expressions used in MPC is the moving horizon window, which is the time interval in which the optimization is applied. The length of this window is called the prediction horizon (N_p). It determines to what extent we wish to predict the future. The prediction horizon length affects the optimal controller performance [115]. The objective of solving a MPC problem is to find a vector that contains the variation of inputs in order to reach the desired trajectory of outputs. The length of this vector is called the control horizon (N_c). Although the optimal trajectory of the future control signal is completely described within the moving horizon window, if the actual control input to the plant only takes the first sample of the control signal while neglecting the rest of the trajectory, this principle is called a receding horizon control scheme. In the planning process, we need information about the state variables at every point in time in order to predict the future. This information is denoted by $x(t_i)$ which is a vector containing many relevant factors and is either measured directly or estimated. A good dynamic model will provide an accurate prediction of the future [31]. Meanwhile, an integrator is naturally embedded into the design, leading to the predictive control system tracking constant references and rejecting constant disturbances without steady-state errors. Another significant advantage of this approach is that, in practice, it requires neither the steady-state information about the control signals ($u(k) = u(k-1) + \Delta u(k)$) nor the steady-state information about the state variables. For a linear MPC problem, the model inside the controller is an augmented one containing an integrator for each output. Consider the following augmented discrete system:

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

where x, u , and y are the state, input, and output variables of the linear system. The

relation between the predicted output of the system inside the prediction window (Y), the time step k_i, the measured states at time t_i , and the designed variation of the inputs will be [31]:

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

where

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

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

where $y(k_i + 2 \mid k_i)$ means the predicted output on step $k_i + 2$ based on the measurement from step k_i and $\Delta u(k_i)$ is the variation of control input in that time step [31]. The performance of a control system can deteriorate significantly when the control signals from the original design meet with operational constraints. By means of a small modification, however, the degree of performance deterioration can be reduced if the constraints are incorporated in the implementation, which leads to the idea of constrained control. In order to modify the controller, all the constraints must be written in the form of variation in input signal. For the constraints on the amplitude of the inputs, the variation of the inputs, and the outputs, the following relation can be written:

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

where

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

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

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

In Eq. (4.3), I and 0 are identity and zero matrices of size $N_c \times N_c$, Y^{min}, Y^{max} are vectors containing the minimum and maximum of all predicted outputs, U^{min}, U^{max} are the minimum and maximum of all inputs, and ΔU^{min}, ΔU^{max} are the minimum and maximum allowable variations for all inputs. As mentioned before, the current optimization problem can be converted into a quadratic form. Assume that the cost function is written as follows:

$$\begin{aligned} J &= \tfrac{1}{2} \Delta U^T H \Delta U + \Delta U^T E \\ M\Delta U &\leq N \end{aligned} \tag{4.4}$$

where M and N are specified by the constraints of (4.3), and H, E are matrices derived based on the cost function. Note that the inputs for the MPC problem are the inputs variations over the length of the control horizon. A typical solution to this problem using Lagrangian multipliers can be found [116]:

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

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

Since this problem must be solved in every time step, a fast approach is required. Identifying active constraints in each time step would help accelerate the calculation procedure. In this research, a procedure is used that suggests an iterative approach to identify the active constraints in order to solve the problem and find the second term in Equation (4.5).

A well-designed controller must be stable and also have desirable performance when applied to a system. As shown in the literature, model predictive control display promising performance and the design parameters can be tuned in such a way that controller instability is avoided.

4.2.2 Stability Analysis

In this section, some theories for MPC stability issues are reviewed. MPC can easily destabilize the system if short prediction horizon is chosen. To avoid this situation, we can consider longer (even infinite) prediction horizons which adversely leads to higher computational effort. Another way to guarantee the stability for any length of prediction horizon is by considering terminal constraints to ensure that the system states converge to definite values at the end of the prediction horizon [117]. Though easiest to handle from the viewpoint of mathematical analysis, a significant drawback of this approach is that the terminal equality constraint can be quite severe; it is hard to satisfy and artificially imposing such a strict constraint can lead to substantial performance loss. For example, for such an approach to work, the underlying system needs to be reachable, instead of just being stabilizable [67]. However, adding constraints might make the optimization infeasible. To overcome this problem, the constraint can be relaxed so that the states of the system converge to a region rather than specific points. In other words, the terminal constraint points can be changed to terminal constraint sets. Chen and Allgower [118] presented an approach called quasi-infinite-horizon MPC, where a quadratic terminal penalty corresponding to the infinite horizon cost of the linearized system is imposed. Because a terminal constraint is used to force the state to lie within a prescribed terminal region, within which the system is stabilized by the linear feedback, feasibility alone implies the asymptotic stability.

Another possible approach is to penalize the terminal constraint points inside the cost function. It has been known for some time (e.g. [119]) that making the horizons infinite in predictive control leads to guaranteed stability, but it was not known how to handle constraints with infinite horizons. The key idea is to re-parameterize the predictive control problem with infinite horizons in terms of a finite number of parameters (instead of the infinite number of control decisions), so that the optimization can still be performed over a finite-dimensional space—in fact, it remains a quadratic programming (QP) problem. In [120], the opinion has been expressed that there is no longer any reason to use finite horizons—at least with linear models. In the case of changing the horizon length to infinity, it is necessary to reformulate the problem for both stable and unstable systems. The essential difference in the latter case is that the unstable modes must be driven to 0 within N_c steps; otherwise, these modes, which are uncontrolled, would become unbounded and result in an infinite cost value.

In [119, 121], it is shown that stability can sometimes be guaranteed with finite horizons even when there is no explicit terminal constraint. The finite horizon predictive control problem is associated with a time-varying Riccati difference equation, which is intimately related to the optimal value of the cost function. The Fake Algebraic Riccati Technique replaces this equation with an algebraic (time-invariant) Riccati equation that resembles that found in infinite-horizon LQ problems.

Besides the approaches using terminal penalty constraints, other approaches have been proposed. One notable alternative approach employs a contraction constraint, that requires the size of the state to shrink over the prediction horizon [122]. More generally, it chooses a positive-definite function of the state and requires this function to decrease over time in the optimization. To ensure feasibility, the size of the prediction horizon is not set a priori but is treated as an optimization variable. The whole computed input sequence can be implemented in open loop until the end of the horizon, as originally suggested, or the optimization can be repeated after some time as suggested in [123]. Extending the MPC formulation for constrained linear systems to non-linear systems is conceptually straightforward but met with practical difficulties. Most of the stability results for the constrained linear systems apply to non-linear systems without modification. In fact, many

of the earlier stability results for constrained optimal control [124, 125] were developed in the context of a general non-linear system. However, the implementation is greatly complicated by the computational complexity in finding a globally optimal solution to a non-convex optimization problem. Computational complexity remains as a major obstacle for designing a practically implementable non-linear MPC algorithm with guaranteed stability. Naturally the researchers focused on finding a formulation that does not require a globally optimal solution to be found, just a feasible solution. In Mayne and Michalska [125] once a feasible solution is found, the subsequent calculation preserves the feasibility and tries to merely improve the cost.

However, [126] shows that the closed-loop system with MPC controller is globally asymptotically stable if and only if the optimization problem is feasible.

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

4.2.3 Robustness Analysis

MPC, being a feedback control method, has some inherent robustness, which was analyzed by several researchers [127]. Nonetheless, when a quantitative description of the model uncertainty is available, it may be beneficial to consider all possible future trajectories under the given uncertainty description in the optimal control calculation. Lee and Yu [128] presented an argument indicating the deficiency of the open-loop formulation and presented an alternative formulation based on dynamic programming. With some modifications, they were able to formulate a MPC algorithm that solves a convex program at each time and guarantees robust stability. However, such an approach cannot be implemented directly since the possible control laws do not yield a finite-dimensional parametrization. Kothare, Balakrishnan and Morari [129] presented an interesting formulation where the minimization at each sample time searched over all linear state feedback laws to minimize the worst case error. The problem was formulated as a Linear Matrix Inequality (LMI), which is convex and can be solved through semi-definite programming. However, MPC integrates performance of optimal control with the robustness of feedback control.

4.3 NMPC Performance on the Low-fidelity Powertrain Model

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

4.3.1 Control-oriented Model

In order to design MPC EMS, a simple and sufficiently accurate model of a PHEV powertrain is needed, which is called a control-oriented model. Among the different architectures for a HEV, the power-split configuration seems to be the most efficient one for a limited capacity of battery [130]. In a power-split configuration, the engine, the electric motor and the generator are connected to each other by means of two planetary gear sets.

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

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

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

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

It should be noted that the parameters are adjusted according to a Toyota Prius plug-in hybrid (See Appendix A). In this system, there are 3 state variables: ring speed (ω_r) which is proportional to the vehicle velocity, engine speed (ω_e), and battery state of charge (SOC). There are 3 inputs: Engine torque (T_e), Motor torque (T_m) and Generator Torque (T_g). η_m and η_g represent motor and generator efficiency, respectively including DC/DC converter and DC/AC inverter efficiencies. T_d is the driver's demanded torque. When the battery is discharged k=1, and k=-1 for battery charging. The aforementioned system is considered as the control-oriented model.

4.3.2 MPC Formulation for a PHEV

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

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

The first term is related to keeping the state of charge around a predefined reference. The second term is intended to minimize the fuel consumption. w_1 and w_2 are weighting parameters that are chosen according to the predicted maximum value of the weighted variables. The effect of these weighting parameters on fuel consumption will be investigated later in this chapter. Fuel economy is estimated according to UDDS drive cycle. Also there are some constraints on this problem that are defined as follows:

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

For finding a simpler form of the controller and also using the linear MPC, the equations of the system are linearized for each time step around the operating point. Furthermore, the fuel consumption map of the engine is estimated as:

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

where $\bar{\alpha}$ and $\bar{\beta}$ are constants [131].

Now, different possible control strategies can be evaluated based on the trip information, by determining an appropriate reference SOC trajectory. This reference SOC is plugged into (4.9) and a quadratic programming problem is solved for each case as follows.

4.3.3 No Knowledge of Trip Information

When there is no knowledge about the trip information, the best strategy is to get use of the vehicle full electric range at early stages of driving. Basically, in this control strategy the battery energy is used until the SOC reaches a predefined level (charge depletion), so it is independent of the driving cycle and driving distance or any other information like initial SOC [132]. In fact, the engine might be started in charge depletion mode at some points, where demanded power exceeds what motor and battery can provide. When the

SOC drops to the lower limit, the strategy enters a loop run by MPC. This controller tries to keep the state of charge around the reference and simultaneously minimizes the fuel consumption (charge sustenance). Here, the lower limit is assumed to be 0.3 because of battery health factors. It should be noted that charge depletion/charge sustenance (CDCS) strategy might be very useful to extend battery life cycle [133].

4.3.4 Known Travelling Distance

In this case, we have knowledge of travelling distance to the next charging station. If travelling distance is less than the vehicle all electric range, the best strategy would be going in pure electric mode. Otherwise we should follow another strategy. As mentioned earlier, it was shown that making a delay in charge sustaining stage would improve fuel economy. Therefore, one can assume the battery SOC is linearly decreased with the vehicle travelled distance according to the following relation [134]:

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

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

To implement this strategy, we have to plug the linear trajectory into the cost function, and MPC minimizes fuel consumption while making state of charge follow the reference.

4.3.5 Known Entire Trip Information

Here, we assume to have access to some information about vehicle velocity determined by road traffic using GPS. The objective is to find an optimized SOC trajectory that can be found based on the longitudinal model of the PHEV to minimize the fuel consumption up to the next charging station (See [135]). To reach this goal, a curve is fit to the vehicle velocity data as shown in Figure 4.1. Then, the velocity schedule is divided into some appropriate segments. Based on the initial and final velocities in each segment and travelling time,

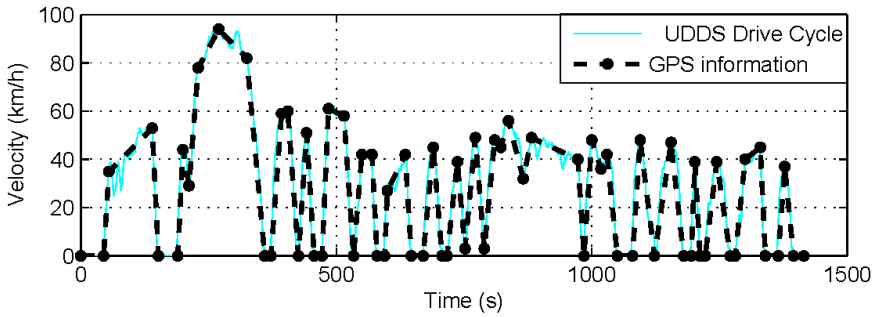

Figure 4.1: GPS information and segments for one UDDS drive cycle

the power demanded in each segment can be calculated. At this moment, an optimization parameter (PR) defined as in Equation (4.12) is introduced.

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

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

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

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

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

It should be mentioned that consideration of segments will improve the prediction for an optimized SOC trajectory. But, it definitely increases computational effort for optimization problem. So, there would be a trade off between number of segments and fuel economy

improvement.

4.3.6 Manual Charge Depletion Charge Sustenance

This strategy seems to be a more practical way to reduce fuel consumption, although its efficacy is limited and cannot be compared to the blended mode approach. In this strategy, the driver is able to manually switch between full electric and charge sustenance (CS) mode. The driver can find out switching occasion based on his/her experience and road traffic knowledge or using GPS information. According to the engine efficiency, it is better to drive in full electric mode in the city, where there are frequent stops and starts. While driving on the highway, it is suggested to switch to charge sustenance mode. It is evident that this switching is possible when there is enough energy stored in the battery. When the state of charge drops to the lowest allowed limit, it is no longer possible to do the switching. Now, we try to divide the UDDS drive cycle into some sections where the driver is willing to switch between modes. In the UDDS schedule, there is one micro cycle ($180 < t < 360$) with higher average speed. We assume that it happens while driving on the highway. In this section, the driver switches to CS mode, but for the rest of drive cycle, the vehicle is driven in EV mode, while there are lots of stop and start occasions or even micro cycles with lower average velocity ahead, before the SOC drops to its lowest allowed limit.

It is obvious that one can suggest different schedules for switching between 2 modes. But, here we try to show that this simple strategy, leads to a significant improvement in fuel economy without committing to any other complex optimization method. Moreover, GPS information can be used as a guideline for the driver to recognize appropriate time for switching between 2 modes.

4.3.7 Simulation Results

The simulation was done in the MATLAB environment. The demanded torque was calculated based on the UDDS drive cycle. This torque serves as one of the inputs to the

controller. EMS uses this input and a linearized model of the powertrain to predict the future contribution of each power source on board. Outputs of the controller are applied to the non-linear model of the powertrain (Equations (4.6) to (4.8)) so that we can find out system state variables like battery state of charge, vehicle velocity and especially fuel consumption. Results of simulation are presented in Figures 4.2 to 4.5. Figure 4.6 depicts fuel economy for different levels of trip information, for different values of the ratio of weighting parameters ($\frac{w_2}{w_1}$) introduced in Equation(4.9). Readers may refer to [115] for an study on prediction and control horizon length effects on fuel consumption. Note that for pure electric driving, the battery provides all the power needed to propel the vehicle. As a result, if the required propulsion power does not exceed what battery or electric motor can provide, there is no need to run the MPC controller.

In each case, control input plots are shown to prove that they meet the constraints on the problem. According to Figure 4.2-a the battery is fully charged at the beginning of the drive cycle and the vehicle goes in pure electric mode and the engine is off (Figure 4.2-b), until the state of charge drops to 0.3 which is the predefined reference state of charge. Fuel consumption in this period is zero. Now, the main controller switches to the MPC mode and tries to maintain the state of charge as close to the reference as possible while minimizing the fuel consumption. Fuel economy in this case is found to be 105 MPG. In this study, the vehicle's fuel economy is reported in miles per gallon (MPG). However, EPA considers miles per gallon gasoline equivalent (MPGe) as a measure of the average distance travelled per unit of energy consumed for alternative fuel vehicles, PHEVs and EVs. In this rating, 33.7 kWh of electricity is equivalent to one gallon of gasoline.

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

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

Figure 4.5-a shows the result of simulation for manually switching between EV and CS

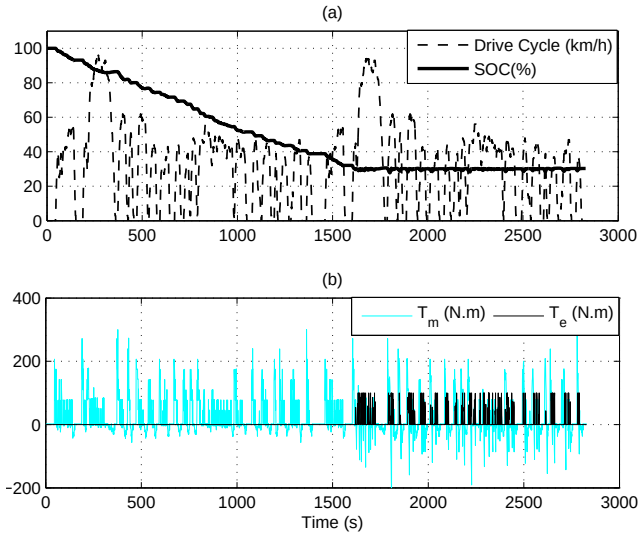

Figure 4.2: Charge depletion/charge sustenance strategy: (a) SOC along 2 UDDS drive cycles (b) Power source torques

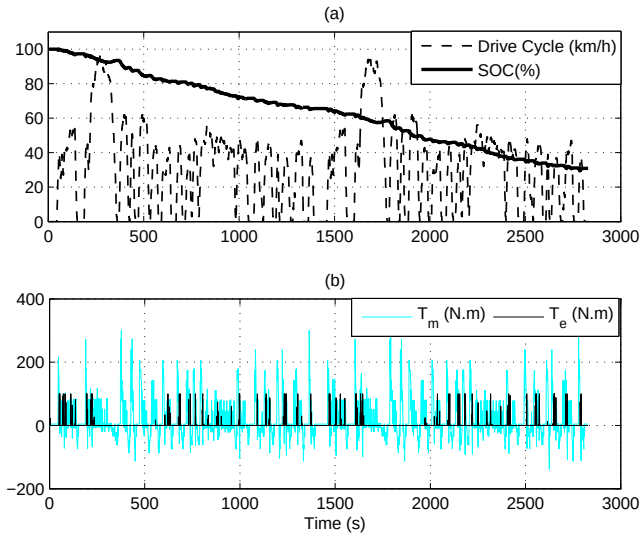

Figure 4.3: Blended mode strategy (Known travelling distance): (a) SOC along 2 UDDS drive cycles (b) Power source torques

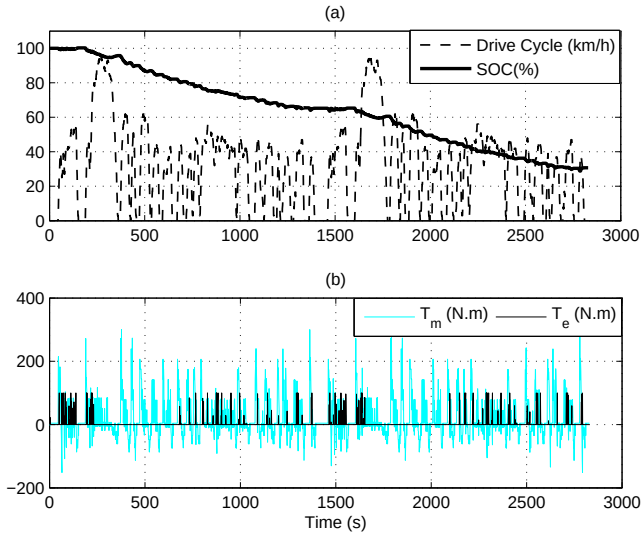

Figure 4.4: Blended mode strategy (Entire trip information): (a) SOC along 2 UDDS drive cycles (b) Power source torques

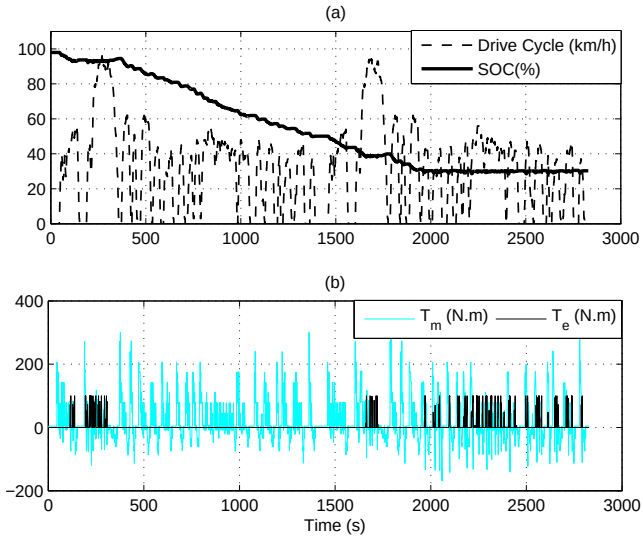

Figure 4.5: Manually switching between EV and CS: (a) SOC along 2 UDDS drive cycles (b) Power source torques

Table 4.1: Fuel economy for different control strategies

Control Strategy	MPG
Charge Depletion/Charge Sustenance	105
Blended Mode (Known travelling distance)	112
Blended Mode (Entire trip information)	123
Manually switching between EV and CS mode	121

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

The performance of the MPC controller depends on weighting parameters inside the cost function as well as predication and control horizon length. The effect of prediction and control horizon values is investigated in [115]. Figure 4.6 shows that CDCS strategy and the blended mode strategy for known trip distance, result in lower fuel economy in comparison with other two strategies. It is shown that manually switching between EV and CS suggests a very close or even better (for some weighting parameter values) fuel economy as compared with blended mode with entire trip information. However, the maximum possible fuel economy that can be reached by using the entire trip information is greater than other strategies. The maximum fuel economy for different control strategies is reviewed in Table 4.1. Indeed, one expects to see better fuel economy by decreasing w_2/w_1. But, we cannot see this improvement in fuel economy for some of w_2/w_1 values, due to the constraints and especially maintaining drivability.

Computationally, it took 35.4 second in real time for 2828 seconds of simulation (for two successive UDDS drive cycles) to be completed. The simulation is conducted on a

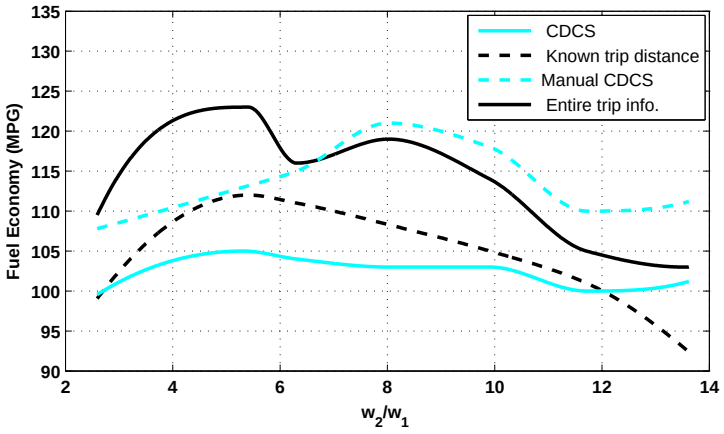

Figure 4.6: Fuel economy versus ratio of weighting parameters in the cost function

machine which is powered by a 3.16 GHz dual core CPU and a 4 GB memory. It would be even faster if the controller was implemented as a C-code. Also it takes 20.2 seconds in real time for finding the optimized SOC trajectory based on trip information on the same machine.

4.4 NMPC Performance Benchmarking

To compare MPC results, we solved a dynamic programming (DP) problem for this problem with the same dynamics and constraints. DP has been extensively used in the literature to find a global solution for HEVs control strategies. DP cannot be implemented on-line. Therefore it is just a benchmark for developing heuristic strategies. In the minimum fuel consumption problem of HEVs, the DP method guarantees a global optimal solution by detecting all possible control options. The results of DP in charge sustaining mode are illustrated in Figures 4.7 to 4.10. According to Figure 4.9, the average speed of the engine is more than what it is in MPC. This brings the operating points closer to the engine sweet spot. Therefore, the resultant fuel consumption is 326 g (123 MPG), although the engine never stops operating. The fuel consumption for MPC without considering input

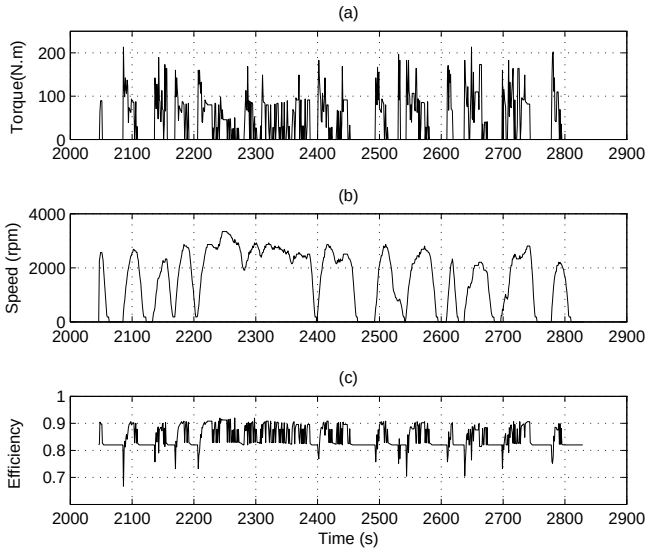

Figure 4.7: DP result: (a) MG2 Torque (b) MG2 Speed (c) MG2 efficiency

variation inside MPC cost function while $SOC_{ref} = 0.3$ is 381 g (105 MPG). This shows 14.4% increase in fuel consumption compared to DP result.

4.5 NMPC Performance on the High-fidelity Powertrain Model

In this part, an EMS based on MPC is developed and applied to the high-fidelity simulation model introduced in chapter 3. In order to apply the EMS to the MapleSim model, low-level controllers are required for the engine and electrical drive. Reducing engine emissions has been considered as an objective for designing low-level controls.

As mentioned earlier, PHEVs have the potential for considerable fuel consumption reductions, but possibly at the expense of increased tailpipe emissions due to multiple cold starts and improper use of the engine for PHEV specific operation [136]. It seems that the

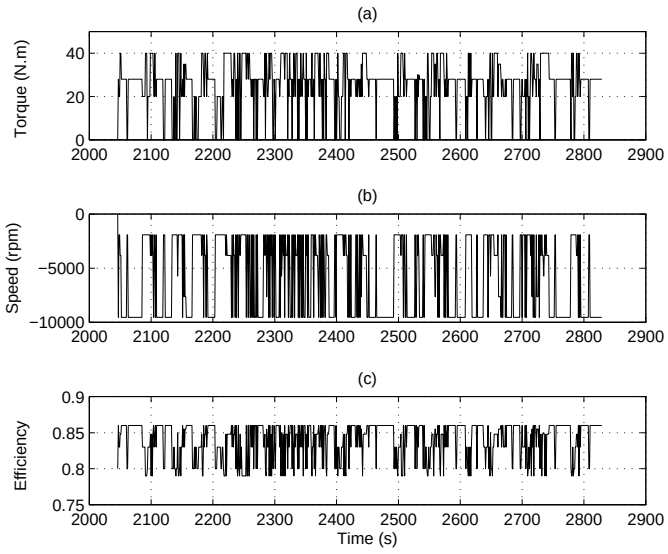

Figure 4.8: DP result: (a) MG1 Torque (b) MG1 Speed (c) MG1 efficiency

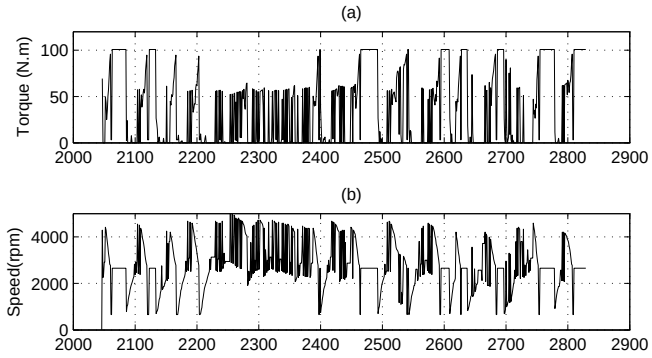

Figure 4.9: DP result: (a) Engine Torque (b) Engine Speed

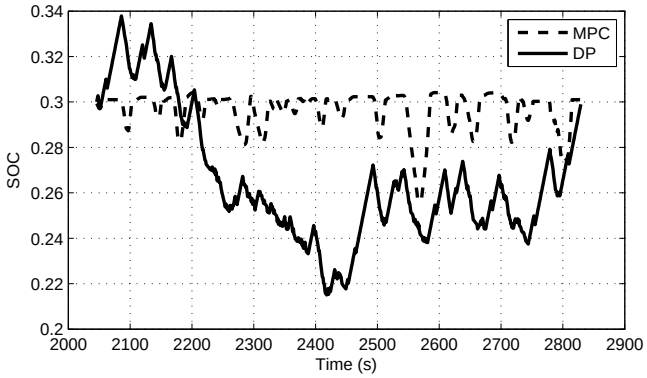

Figure 4.10: SOC comparison for DP and MPC in charge sustenance mode

catalyst temperature management for reduced tailpipe emissions is a challenging control problem due to the frequent and extended engine shut-down and catalyst cool-down [137].

Throughout the literature there are different approaches to address this emissions problem. For instance, it is possible to use a separate hardware in addition to the catalyst to resolve the concerns on different engine start events. In [138], the authors investigate the addition of hydrocarbon (HC) absorber traps and activated carbon fiber canister in traditional exhaust after-treatment system. These traps can store HC temporarily and release it after the temperature reaches the light-off temperature of exhaust after-treatment device. Therefore, the HC will be catalyzed by exhaust after-treatment even at low temperatures. The same happens to NOx that is translated to N_2.

A number of articles have presented models for a three-way catalytic converter (TWC) to describe the heat/mass transfer in the after-treatment system and conversion efficiency as a function of the catalyst brick temperature and air to fuel ratio [139, 140]. These models are primarily used for design and evaluation and are too complex for the development of control algorithms. Using simpler models of catalyst conversion efficiency makes it possible to design a specific model-based emissions control for the engine. For instance, [141] proposed a simplified model of an internal combustion engine to derive a sliding mode control law for emissions reduction. In [142] the authors used a PMP approach to do real-

time optimal control of cold start via an experimentally verified control-oriented model of the engine.

In most optimization problems for HEVs, minimizing fuel consumption is the only objective, and emissions limitation is considered as a constraint of the process; as long as emissions are within predefined limits, it does not influence the optimization process [143]. Recent HEV studies have considered emission reduction as a part of the control objective and analyzed the trade-off between fuel economy and emissions [144, 145]. Most of those studies considered minimization of fuel consumption and engine-out emissions instead of tailpipe emissions [144, 146]. Although reducing engine emissions can reduce tailpipe emissions as well, it is not the key factor. Since the conversion efficiency of a cold catalyst is very low, fast catalyst warm-up and sustainment are the key factors to minimizing tailpipe emissions [147]. In [148] the authors designed a control scheme using multi-objective genetic algorithms to develop a fuzzy controller to reduce fuel consumption and emissions of a parallel HEV simultaneously. Sagha et al. [149] proposed a modified equivalent consumption minimization strategy (ECMS) to include fuel consumption and also a NOx reduction control for a lightweight through-the-road architecture HEV. The control strategy has been able to reduce NOx emissions near to Euro 4 restrictions and also maintain CO and HC emissions below the restrictions of Euro 4 and 5 standards. Gao et al. [150] investigate the effect of an absorber which can substantially reduce the hydrocarbon and nitrogen oxide emissions by temporarily storing them until the three-way catalyst is sufficiently warm to remove them from the exhaust. It is shown that the mentioned absorber has a substantial effect on reducing emissions for a PHEV in comparison to a HEV.

Smith et al. [136] experimentally verify a vehicle supervisory control system for a pre-transmission parallel PHEV powertrain architecture, where tailpipe emissions from a PHEV test platform have been reduced through the development and refinement of vehicle supervisory control methods. The focus of the enhancements was to replace high engine torque demands during starting with clean electric motor torque through some rule-based methods. This approach proved very effective for the reduction of NOx emissions. However, the model-based control approaches suggest even better performance for different

operating conditions. In [137], the authors used dynamic programming approach and came up with an adaptive supervisory powertrain controller (SPC) that optimally adjusts the engine on/off, gear-shift, and power-split strategies under various Energy-to-Distance Ratios (EDR) and catalyst temperature conditions for a pre-transmission parallel plug-in hybrid electric compact SUV in order to achieve near-optimal fuel economy and emission performance. In fact, the authors used a simple simulation model of the powertrain to evaluate their proposed scheme. Also, they assumed that air to fuel ratio and spark ignition timing of the engine are controlled for the optimal performance.

The controls design procedure is quite similar to section 4.3 but with a modified cost function and of course a different control-oriented model.

4.5.1 Control-oriented Model inside the High-level Controller

In a power-split configuration, the engine, the electric motor and the generator are connected to each other by means of two planetary gear sets (See Figure 3.1). Since the driveline dynamics is faster than any other dynamic of the whole system, by neglecting the loss power in planetary gear sets, the relation between torques will be [151]:

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

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

$$\dot{SOC} = A\ SOC + B \begin{bmatrix} T_m \\ \omega_e \end{bmatrix} + \tilde{B} \begin{bmatrix} T_d \\ V \\ 1 \end{bmatrix}$$

$$y = C\ SOC + D \begin{bmatrix} T_m \\ \omega_e \end{bmatrix} + \tilde{D} \begin{bmatrix} T_d \\ V \\ 1 \end{bmatrix} \tag{4.15}$$

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

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

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

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

4.5.2 NMPC Energy Management System

The modified cost function is:

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

The first term is related to keeping the state of charge around reference. The second term is for minimizing the fuel consumption. Two last terms are considered for making control inputs as smooth as possible, where ΔT_m and ΔT_g are the variation of motor and generator torque. w_1, w_2 , w_3, and w_4 are weighting parameters chosen according to the predicted maximum value of the weighted variables. These weighting parameters have a significant effect on controller performance. Readers may refer to [135] for more details on determining weighting parameters. To solve this problem we keep the constraints on (4.9) unchanged:

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

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

4.6 Low-level Controllers

From the last section, setpoints for engine, motor and generator torque are available. To make these sources follow the setpoints, the low-level controls need to be tuned. For electric drives, a standard PI controller can be used. However, Appendix B reviews the design procedure of a direct torque controller for high-fidelity simulation model of the electric drive. But for the engine we would look for a different approach. Since we are to apply the engine torque setpoint originated from MPC to a mean value gasoline engine model which captures some non-linear phenomena in the engine, we need an appropriate method to be robust, able to ensure stability and maintain good system performance under adverse operating conditions. Sliding mode control (SMC) can be designed in such a way that there is good disturbance rejection and trajectory tracking, fast dynamic response, and good stability [152].

SMC is a reliable model-based control method for engine torque management in practical cases, since it is capable of handling the model uncertainties. Here, we look at emissions control as well; the engine low-level controller should be capable of tracking the designated engine torque while minimizing engine emissions. The main control input is the throttle angle. Other inputs, like injected fuel rate and ignition timing, highly affect the transient behavior of the engine. According to legislation to have the best fuel economy and emission for the engine, we are not allowed to change these two parameters away from their optimum values for a long time [153]. Therefore, throttle angle is generally more reliable and dominating engine input to change the steady state response in this case. But, we need to determine the amount of injected fuel as well to control air-fuel ratio which basically contributes to engine emissions.

Engine torque management can be done in two ways: The first approach is using engine torque sensor to measure crankshaft torque for feedback control. We consider the effects of the engine combustion torque, friction torque, pumping torque, and all accessory loads by following this approach. As mentioned before, spark timing and air fuel ratio affect transient engine torque response. However, doing torque control based on these two parameters as the major inputs, makes them remain away from the optimal ranges.

Thus, torque control in this way cannot guarantee low emissions, but it would reduce uncertainties especially in the case of engine aging. The second approach is to measure and control manifold pressure, since the engine torque is a function of cylinder air flow which in turn is a function of the manifold pressure. Indeed, air-fuel ratio is another parameter that determines engine combustion torque. In case of assuming constant air to fuel ratio and ignition timing, the control goal can be changed to make the manifold pressure follow the desired value. If the throttle is used to control manifold pressure, ignition timing and air to fuel ratio on combustion torque do affect the torque control. So the disadvantage of this second approach is a larger amount of calibration needed to find a proper conversion from desired torque to desired manifold pressure for all engine operating conditions. However, use of this strategy will not require a torque sensor [153]. In this research, we use the second approach along with controlling the air to fuel ratio to get a desirable emission performance [154].

4.6.1 Engine Control-oriented Model

To reduce emissions, we focus on maximizing catalytic converter efficiency for different operating conditions of the engine. The conversion efficiencies are generally measured over a range of air to fuel ratios and catalytic converter body temperatures, requiring extensive data fitting and look-up tables. The conversion efficiency can be described by the S-shaped Wiebe function as proposed in [142]:

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

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

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

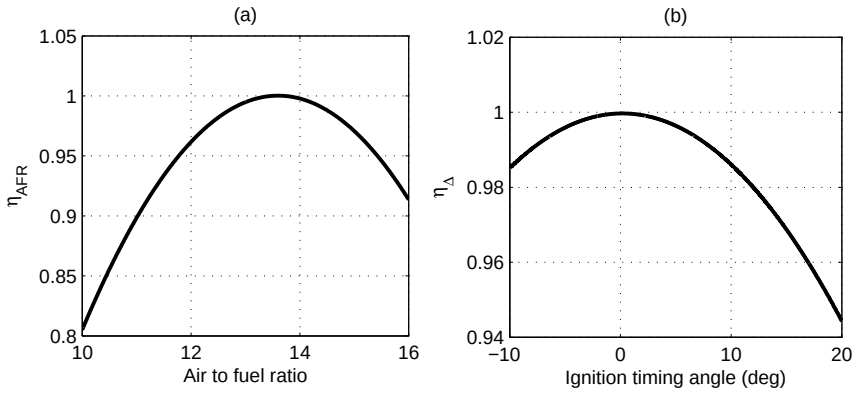

Figure 4.11: Engine torque change with air to fuel ratio and ignition timing

To estimate the combustion torque, we need to know how much fuel goes into the engine. In fact, all injected fuel cannot get into the cylinders because of vaporization. To consider this, we use another dynamic equation for wall-wetting effect:

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

where τ_f is a constant.The injected amount of fuel ($\dot{m}_{inj,fuel}$) is another control input in the problem. Now, the engine generated torque can be estimated via:

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

where η_i, and H_f are engine thermal efficiency (approximately a function of engine speed and manifold pressure) and gasoline heat of combustion. Readers may refer to [113] for a table of numeric parameters used in this engine model. Figure 4.11 shows the air to fuel ratio and ignition timing efficiencies (η_{AFR} and η_{Δ}).

Air to fuel ratio can be described as:

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

According to [155], the third dynamic equation to estimate the exhaust gas temperature can be written as:

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

where $AFI = cos(0.13(AFR - 13.5))$, $ST = 7.5\Delta + 600$ and Δ is the angle of crankshaft in degree at which the ignition occurs. Here, it is assumed that the catalytic converter body temperature is proportional to exhaust temperature [156].

4.6.2 Engine Controls Design

To control the engine torque and catalytic converter efficiency using sliding mode control, three different sliding surfaces should be defined [157]: S_1, S_2, and S_3 for torque, air to fuel ratio, and exhaust temperature control.

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

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

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

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

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

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

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

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

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

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

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

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

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

4.7 Results of Simulation

In the simulation procedure (Figure 4.12) there is a driver who follows the predefined drive cycle with gas and brake pedals. Command from gas pedal is calibrated to give the demanded torque, which directly goes to the control system. Also, the mechanical

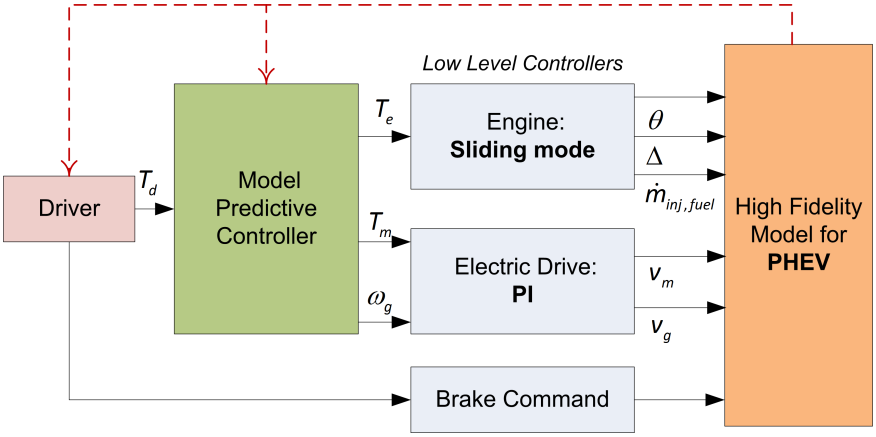

Figure 4.12: Simulation procedure

brake is an standard PI controller. The demanded torque and the vehicle velocity, which is found through Equation (4.16), are fed to the control system and the high-level controller calculates the control inputs every 2 seconds. Generator speed and torque, engine torque, battery SOC and its variation are other information that the high-level controller needs to predict proper controls for drivability, fuel economy and maintaining battery SOC around the predefined level. According to control inputs, new setpoints can be calculated for engine, motor and generator torque. Low-level controllers are in charge of tracking these setpoints as closely as possible and reducing emissions as explained below.

For evaluating controls in MATLAB environment, the high-fidelity MapleSim model is converted to an optimized S-function.

4.7.1 Without Emissions Control

Figure 4.13-a shows the vehicle follows two successive UDDS drive cycles for the blended mode strategy without controlling the emissions. As it is seen, the battery state of charge drops to the minimum possible level at the end of the trip. In fact, if the horizontal axis displays the travelled distance, the depletion trajectory will be a near linear profile. Therefore, the high-level controller has maintained drivability while keeping the final SOC above

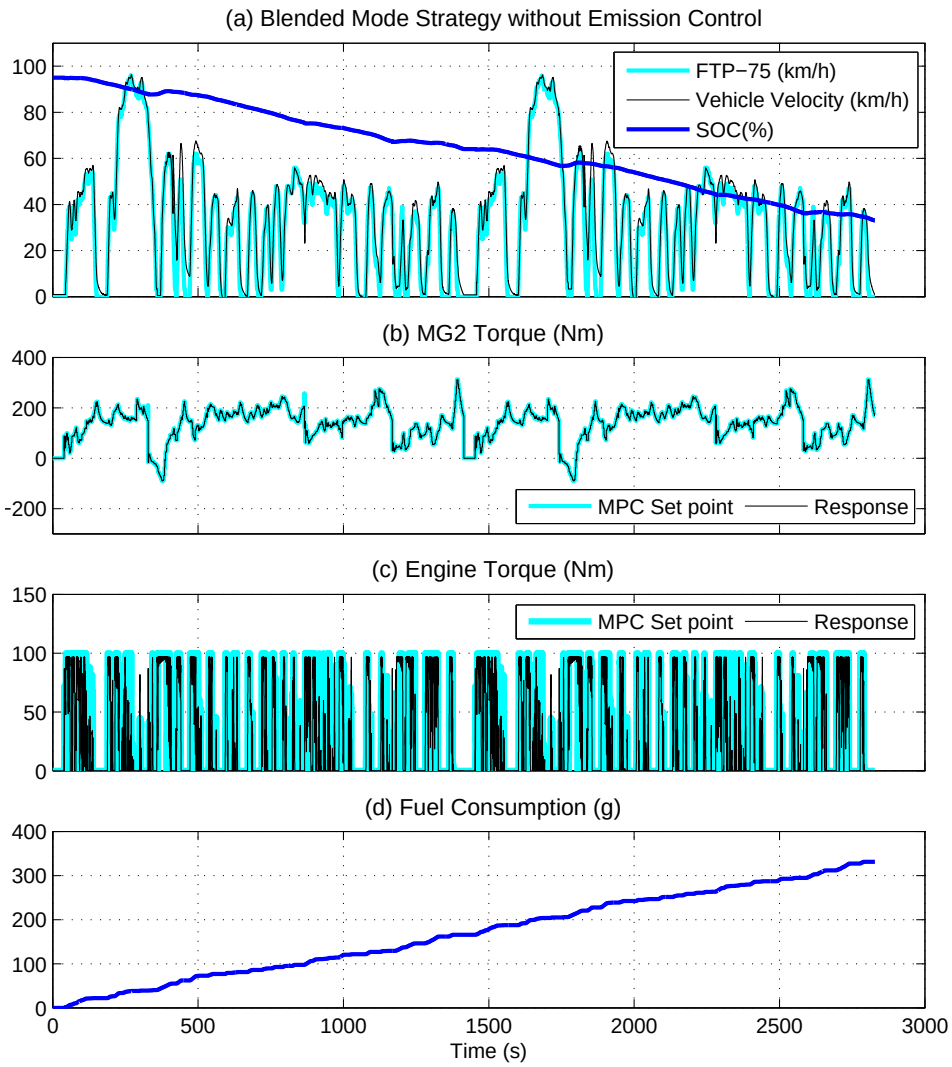

Figure 4.13: Blended mode MPC strategy without emission control: (a) Velocity and Battery SOC (b) MG2 Torque (c) Engine Torque (d) Fuel Consumption

the minimum level. Figure 4.13-b and Figure 4.13-c show that the low-level controllers have made the engine and MG2 torque follow the MPC-prescribed trajectory. According to Figure 4.13-d, the resultant fuel consumption is $1.66l/100km(142MPG)$.

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

4.7.2 With Emissions Control

Figure 4.15-a shows the drivability of the vehicle in blended mode while controlling the engine emissions. Figure 4.15-b shows more frequent operation of MG2 in comparison to the case where the engine emission is not controlled (Figure 4.13-b). The reason is the failure of the engine to provide adequate torque to propel the vehicle for all time steps. This failure is due to the extra heat loss that occurs inside the engine. As mentioned earlier, one way to have less emission is to keep the catalyst temperature high. To reach this goal, the temperature of the gas inside the exhaust manifold needs to be increased. Changing the ignition timing makes it possible to have more heat loss and exhaust gas with higher temperature at the expense of higher fuel consumption. In brief, to warm up the catalyst, a larger amount of the combustion energy should be dissipated as heat instead of producing mechanical energy inside the engine. Figure 4.15-b,c shows the SMC and PI controllers are successful in making the engine and MG2 follow the MPC-prescribed trajectories. Changing ignition timing and AFR result in a decrease of engine combustion torque which is shown by corresponding efficiencies in Equation(4.20). Therefore, Figure 4.15-d indicates more fuel consumption of $1.89l/100km(124MPG)$.

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

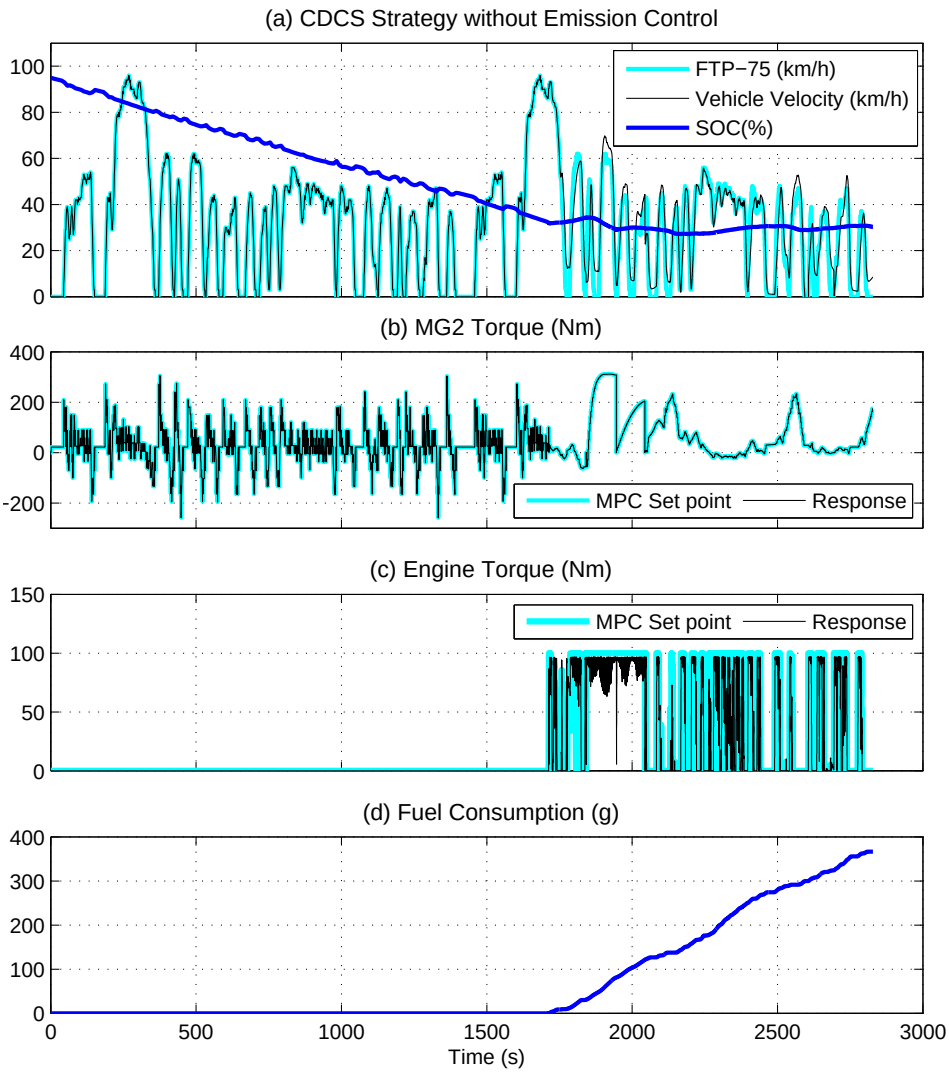

Figure 4.14: CDCS MPC strategy without emission control: (a) Velocity and Battery SOC (b) MG-2 Torque (c) Engine Torque (d) Fuel Consumption

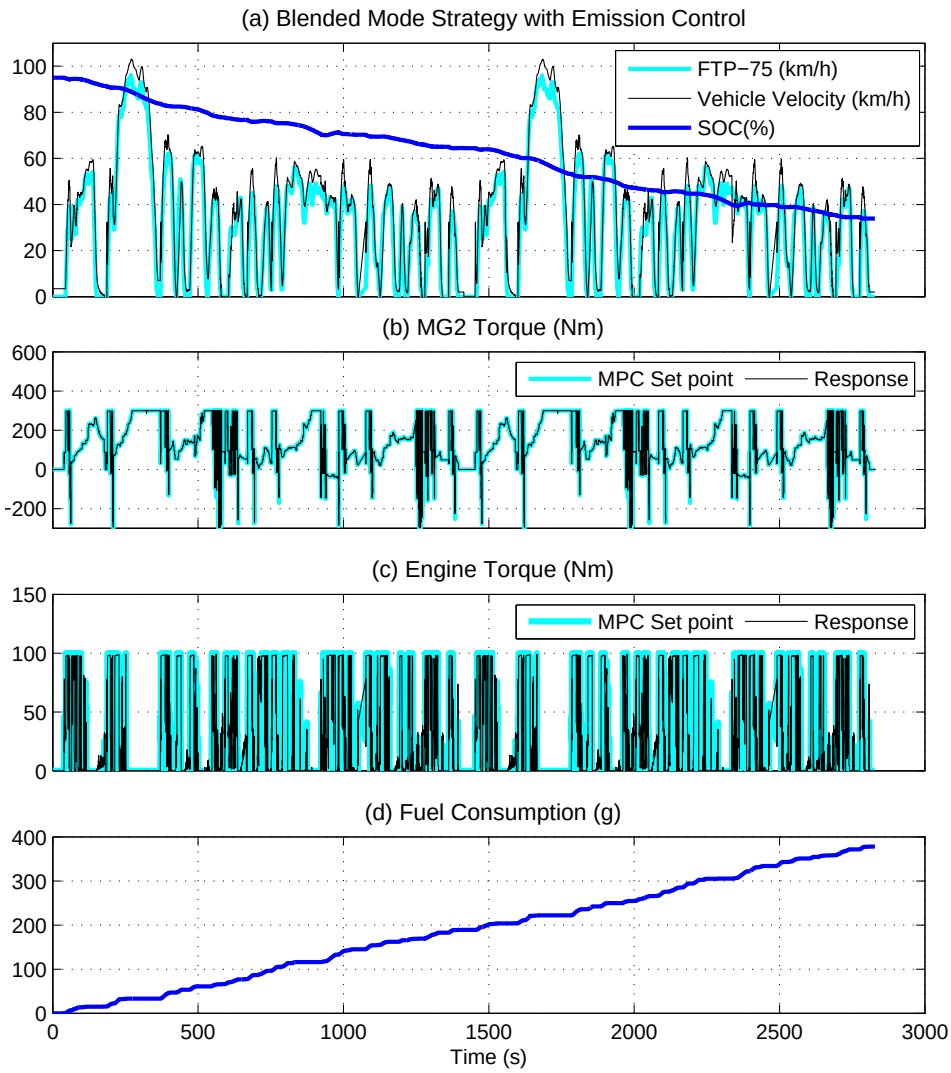

Figure 4.15: Blended mode MPC strategy with emission control: (a) Velocity and Battery SOC (b) MG-2 Torque (c) Engine Torque (d) Fuel Consumption

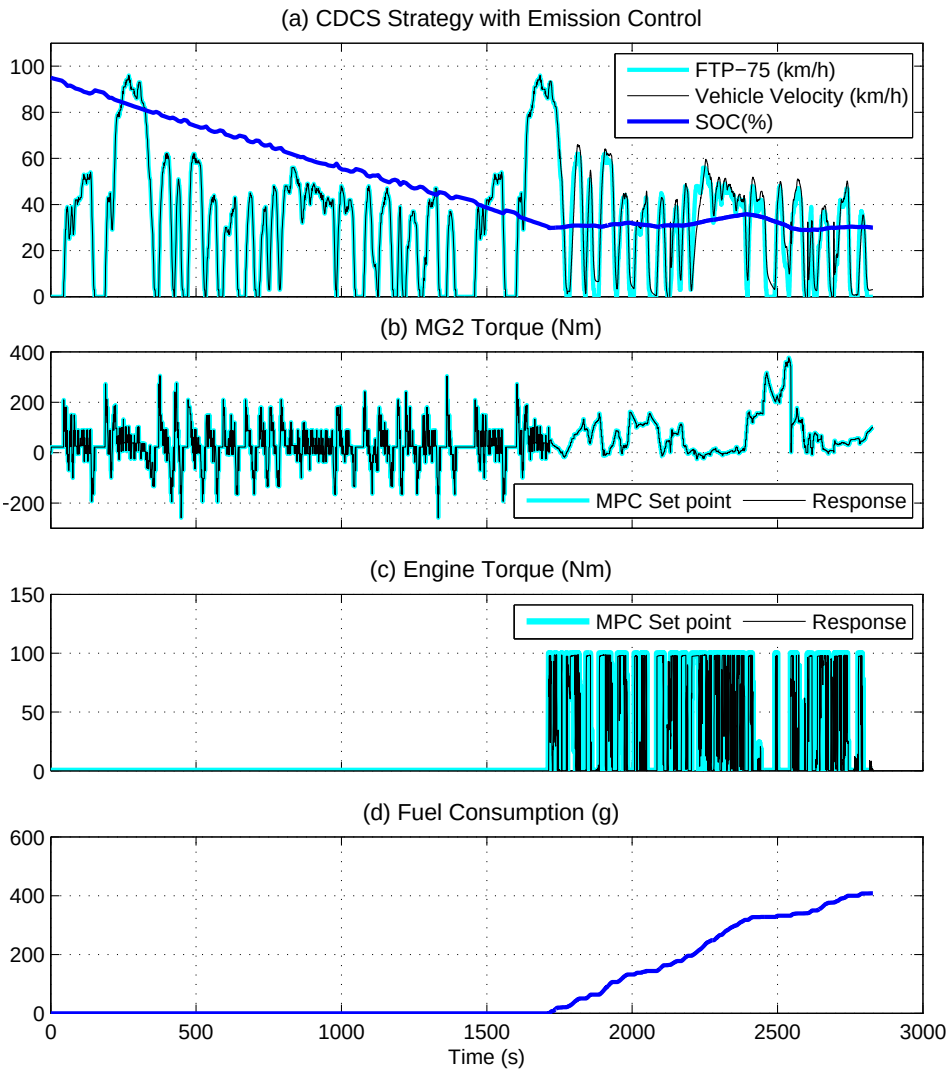

Figure 4.16: CDCS MPC strategy with emission control: (a) Velocity and Battery SOC (b) MG-2 Torque (c) Engine Torque (d) Fuel Consumption

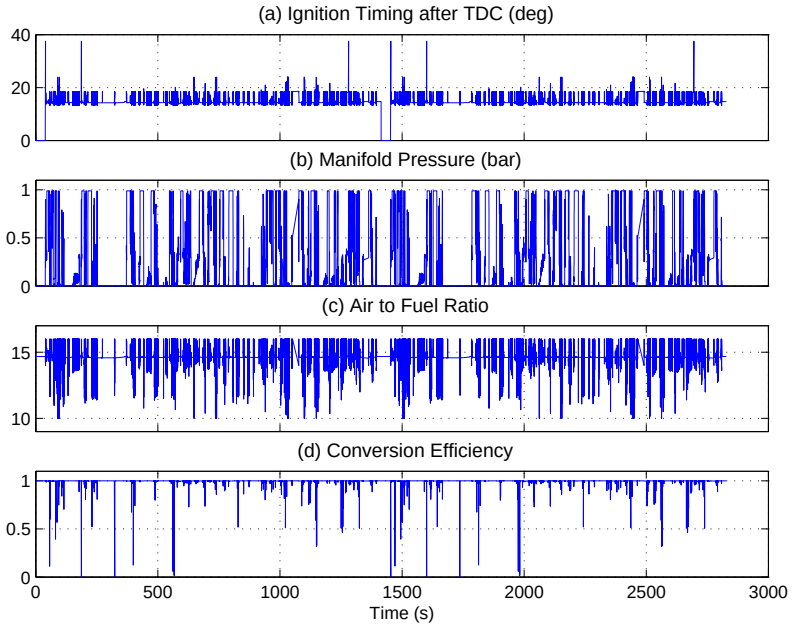

Figure 4.17: Blended mode MPC strategy with emission control: (a) Ignition Timing (b) Manifold Pressure (c) AFR (d) Conversion efficiency

As mentioned earlier, exhaust temperature is closely related to the catalytic converter body temperature that contributes to conversion efficiency. Ignition timing has a considerable effect on determining exhaust temperature. Figure 4.17-a shows ignition timing throughout the vehicle trip. For the sake of combustion stability, we confine the ignition timing within $\Delta \in [0, 20]$. As a result, exhaust temperature (T_{exh}) remains high enough for the whole trip.

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

Figure 4.18 demonstrates different results for engine emission control in the CDCS

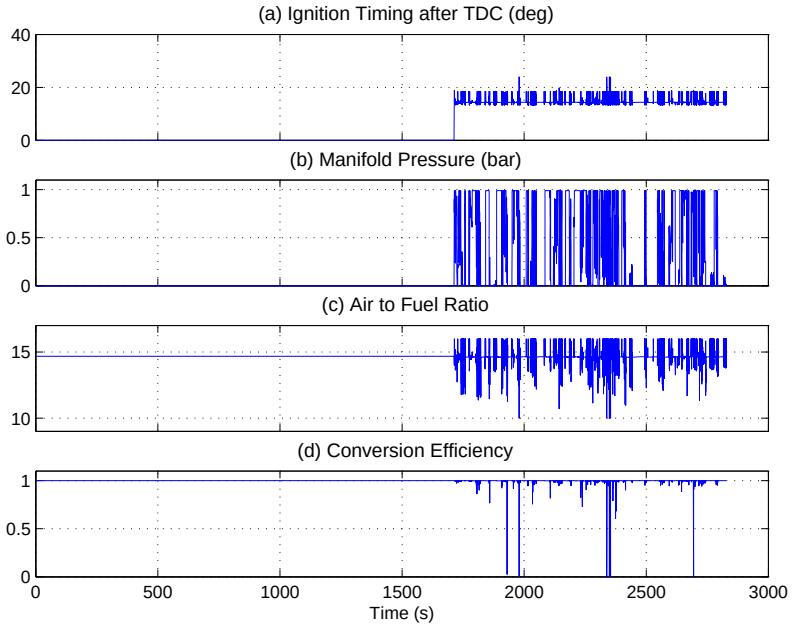

Figure 4.18: CDCS MPC strategy with emission control: (a) Ignition timing (b) Manifold pressure (c) AFR (d) Conversion efficiency

Table 4.2: Fuel consumption comparison

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

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

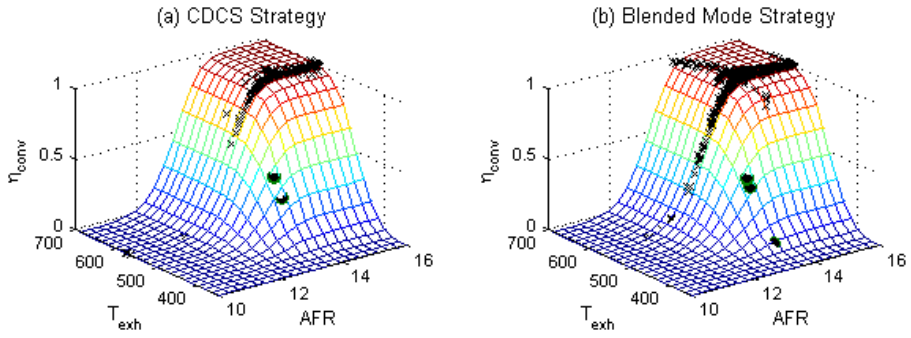

Figure 4.19: Catalyst conversion efficiency for (a) CDCS (b) Blended mode strategies: without emission control(bullet marker)/with emission control (cross marker)

strategy. It is evident that the engine emission is zero for the full electric mode of driving.

The results are summarized in Table 4.2. Figure 4.19 shows the distribution of catalyst operating point for two mentioned strategies. Moreover, the readers are referred to [158] for the performance comparison of the current high-level controls with that of the adaptive ECMS power management system.

4.8 Chapter Summary

In the first part of this chapter, an energy management strategy for a PHEV was designed using a discrete MPC concept for different levels of trip information. The system was chosen close to the specification of Toyota Prius Plug-in hybrid. The model inside the

controller was linearized and discretized, and the simulation time was comparable to the time needed for implementing the controller on-line in a practical situation. Simulation shows that the fuel economy can be improved by up to 17% by considering an optimized SOC trajectory based on entire trip information, in comparison to charge depletion/charge sustenance strategy, where there is no knowledge about the vehicle speed schedule.

In the second part, an energy management scheme was designed including high-level and low-level controllers to reduce fuel consumption and engine emissions of a power-split PHEV. The previous model predictive control EMS was modified with a different control-oriented model. To design a low-level controller for the gasoline engine, the sliding mode control approach was used to make the engine follow a desirable torque and emissions performance. The control scheme was applied to a high-fidelity simulation model of the vehicle, including a chemistry-based model of the lithium-ion battery developed in MapleSim 6.1, to get more realistic results. The simulation was done for both charge depletion/charge sustenance and blended mode strategies. The results showed a promising fuel consumption of 1.89 l/100 km (124 MPG) and 2.05 l/100km (115 MPG) for blended and CDCS strategies, respectively, while the engine emissions were controlled during the vehicle trip. It was shown that engine emissions control can increase fuel consumption by 13% on the average.

Chapter 5

Explicit Model Predictive Control Design

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

5.1 Introduction

Despite the demonstrated benefits of MPC, its capabilities are limited due to the computational effort required for solving the on-line optimization problem. This MPC shortcoming can be overcome by using the so-called explicit/multi-parametric MPC (eMPC or mp-MPC) methods. In eMPC, the on-line optimization problem involved in the MPC is solved off-line using multi-parametric programming approaches and the control variables and the value function of the optimization problem are derived as explicit functions of the system state variables, as well as the critical regions of the state-space where these functions are valid. This significantly reduces the computational effort required for the MPC implementation [9].

In this chapter, a near-optimal, real-time implementable solution for a PHEV EMS is proposed, using explicit model predictive control. Di Cairano et al. [5] have used eMPC solution for a series HEV. But, to the best of our knowledge, this is the first time that an explicit model predictive controller is designed and implemented for the Toyota Prius plug-in hybrid power-split architecture. In this way, there are some challenges for finding an appropriate control-oriented model. Using eMPC is only practical for relatively small problems, since the size of controls database is exponentially increased by the number of state variables. The control-oriented model should be very simple, but accurate enough to capture the complex dynamics of a power-split PHEV powertrain. Moreover, the control-oriented model and the optimization cost function should be chosen in such a way that they guarantee feasible solution beside optimality, stability and desirable performance for the controller. The proposed control system is a switched discrete-time one. As a result, a stability analysis is required to make sure that the control system keeps its performance for all possible PHEV operating points. Therefore, an innovative control-oriented model is introduced to be very simple and also addresses the aforementioned issues.

The performance of the eMPC controller is closely related to how accurate is the control-oriented model. In this chapter, a control-relevant parameter estimation method is proposed to obtain a better control-oriented model that is still simple but captures more dynamical relations of the powertrain. Based on the new model, the EMS is redesigned to see how effective is control-oriented model improvement.

First of all, the energy management strategy design and implementation is discussed by developing an appropriate control-oriented model. In section 5.3, the resultant polytopes from solving eMPC are presented in addition to the physical interpretation of different regions.

Section 5.4 discusses the stability of the closed-loop system. In section 5.5, the designed controller is applied to the simulation model, and the results are compared to the MPC approach results.

Then, the control-oriented parameter estimation approach is followed by a new control-oriented model to design another eMPC EMS. Finally, the performance of the newly-

designed eMPC EMS is validated through MIL simulation.

5.2 eMPC Energy Management Strategy Design

In this section, an energy management strategy for Toyota Prius plug-in hybrid is designed by using the eMPC approach. Bemporad et al. [68] presented a technique to determine the linear quadratic regulator for constrained systems through off-line multi-parametric linear programming (mp-LP) and multi-parametric quadratic programming (mp-QP). The control law was shown to be piecewise linear and continuous, and could be implemented as a look-up table, i.e., different linear state feedback laws were applied to different polyhedral regions. Therefore, the on-line control computation is reduced to determining the region associated with the current state and then applying the stored control law associated on that region. The design procedure of the controller can be divided into two different stages: off-line and on-line procedures. The objective of the off-line procedure is to populate some look-up tables using a control-oriented model which contain the appropriate control actions for different system operating points. In this procedure, a multi-parametric programming problem is solved with an initial condition. The result would be a polytope with a specific control action. Then, the whole state space is explored to find the other polytopes and control actions. There may be a huge number of look-up tables as the result of solving the optimization problem. In this case, region reduction methods can be used to remove some redundant constraints in the optimization problem in order to downsize the look-up tables and improve controller speed in the implementation stage. The on-line procedure happens during controls implementation. Here, a fast and efficient searching algorithm is needed to look up the mentioned tables to find which one of those polytopes contains the initial state variable. This algorithm is called point location. Once, the corresponding polytope is found, the eMPC law can be obtained. These steps are shown in Figure 5.1.

In the following subsections, each step is addressed separately.

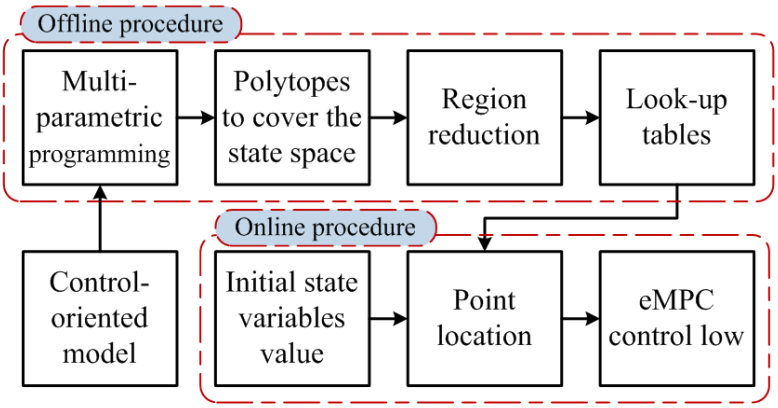

Figure 5.1: eMPC design procedure

5.2.1 Control-oriented model

A relatively simple model is required to take advantage of the explicit model predictive control approach. To end up with smaller look-up tables and make it possible to implement the controller to a commercial control hardware with a limited amount of memory and computational power, the following model inside the controller is used:

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

where $Z = \begin{bmatrix} SOE, & E \end{bmatrix}^T$ and $U = \begin{bmatrix} P_{BAT}, & P_{ENG}, & P_{BRK} \end{bmatrix}^T$.

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

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

There are two state variables in this model: battery state of energy (SOE) and tractive energy E. Battery SOE is defined as the ratio of battery stored/released energy to the

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

5.2.2 Optimization problem formulation

For hybridizing a vehicle, it is important to improve the fuel economy and emissions performance in comparison to the baseline vehicle while maintaining the vehicle drivability.

In a PHEV, fuel economy is closely related to the battery depletion trajectory. As a result, a tracking term inside the cost function is considered that SOE should follow at each time step (SOE_{ref}). To address drivability, another term is added inside the cost function to ensure that the hybrid powertrain provides adequate propulsion power to the driver's request (E_{ref}). On the other hand, the fuel consumption should be reduced as another important objective. The engine fuel consumption is assumed to be proportional to the engine generated power. Therefore, the engine power can be minimized as one of the assumed control actions. The cost function and constraints along the prediction horizon can be written as follows:

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

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

$$\mathbf{Q} = \begin{bmatrix} \omega_1 & 0 \\ 0 & \omega_2 \end{bmatrix}$$
$$\mathbf{R} = \begin{bmatrix} \omega_3 & 0 & 0 \\ 0 & \omega_4 & 0 \\ 0 & 0 & \omega_5 \end{bmatrix}$$

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

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

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

where $X = \left[\begin{array}{cccc} SOE, , & E & SOE_{ref}, & E_{ref} \end{array} \right]^T$, $X_{min} = \left[\begin{array}{cc} SOE_{min}, & E_{min} \end{array} \right]^T$, and $X_{max} = \left[\begin{array}{cc} SOE_{max}, & E_{max} \end{array} \right]^T$,

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

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

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

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

In multi-parametric programming, the objective is to find the optimizer U^* for a whole range of parameters X, i.e. $U^*(X)$ as an explicit function of the parameter X. The cost function is quadratic, so a multi-parametric quadratic programming (mp-QP) problem is solved. As shown in [159], we wish to solve problem (5.3) for all X within the polyhedral set of feasible values X_N. According to [160], if the multi-parametric quadratic program (5.3) is considered, then the set of feasible parameters X_N is convex, the optimizer U^* is continuous and piecewise affine (PWA), and the optimal value function J^* is continuous, convex and piecewise quadratic.

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

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

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

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

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

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

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

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

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

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

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

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

5.2.3 Region reduction

At the implementation stage, a small number of constraints defining a region is preferable since the controller quickly checks the constraints to find the appropriate control action. Therefore, computation of the minimal representations of the controller regions T_i where h_i and k_i are given according to (5.8) can significantly reduce the computational load in most multi-parametric programming solvers [161].

Starting with a given piecewise affine solution, in [162] the authors provide an approach to reduce the number of partitions by optimally merging regions where the affine gain is the same, so that the original solution is maintained but equivalently expressed with a minimal number of partitions. However, techniques for achieving a more drastic reduction of complexity require changing the solution, by accepting a certain level of sub-optimality with respect to the original problem formulation [46]. In [163, 164] the authors propose recursive rectangular partitions of the parameter space to determine a sub-optimal solution to general classes of multi-parametric programming problems. Based on a dynamic programming formulation of the finite-horizon optimal control problem, in [165], the au-

thors propose an approach to relax optimality within a prescribed bound in favor of the reduced complexity of the solution, which in the case of linear systems and piecewise affine convex costs leads to another form of computing approximate mp-LP (multi-parametric linear programming) solutions.

In this way, there are a couple of approaches to identify redundant constraints and remove them in order to reduce the number of regions. An ordinary way to address this problem is to solve n LPs (in the worst-case of $n-1$ constraints) for each region to detect and remove all redundant constraints according to [166]. Another approach is called ray shooting [167], which is suitable for the cases where the fraction of redundant constraints is low. On the other hand, the bounding box approach is most useful for polytopes with many easily detected redundant constraints. The region reduction that is used here is a combination of ray shooting and bounding box in order to find the redundant constraints even faster [168].

5.2.4 Point location problem

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

By exploiting the properties of multi-parametric linear and quadratic solutions, in [169], two new algorithms are proposed that avoid storing the redundant polyhedral regions, significantly reducing the on-line storage demands and computational complexity of the evaluation of control. In [170] the authors suggest to organize the hyperplanes defining the regions on a binary search tree (possibly further subdividing some of the regions), so that

the time to locate the state vector on-line within the partition becomes logarithmic in the number of stored cells, and memory space is saved.

Here, the well-known concept of interval trees [171] is used in order to find a list of candidates that are possible solutions to the point-location problem. Standard interval trees are efficiently ordered binary search trees for determining a set of possibly overlapping one-dimensional line segments that contain a given point or line segment. The mentioned line segments can be found through the bounding box approach. Then, a local search needs to be done on the list of candidates to determine the polytope to which the current state variable belongs [168]. All the optimization problem has been solved in multi-parametric toolbox [172]. After solving the mp-QP problem, we come up with many look-up tables and control actions. The following section discusses the resulting look-up tables further.

5.3 Energy Management Polytopes

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

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

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

Region(I) $E > E_{ref}$ and $SOE < SOE_{ref}$: In this part, the controller has to increase the battery state of charge and slow down the vehicle. There are two ways to do that.

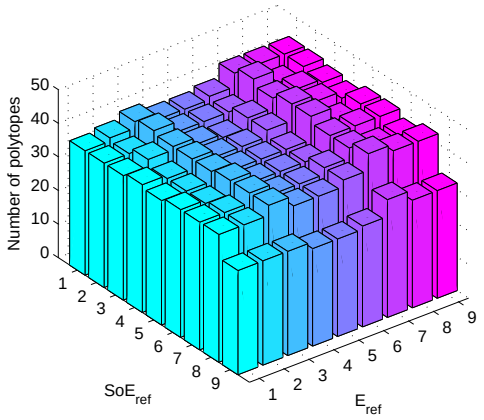

Figure 5.2: Number of polytopes for different levels of E_{ref} and SOE_{ref}

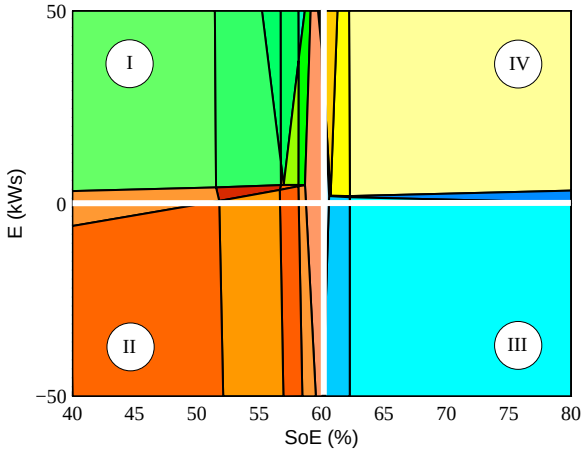

Figure 5.3: Polytope set for $E_{ref} = 0$ and $SOE_{ref} = 60\%$

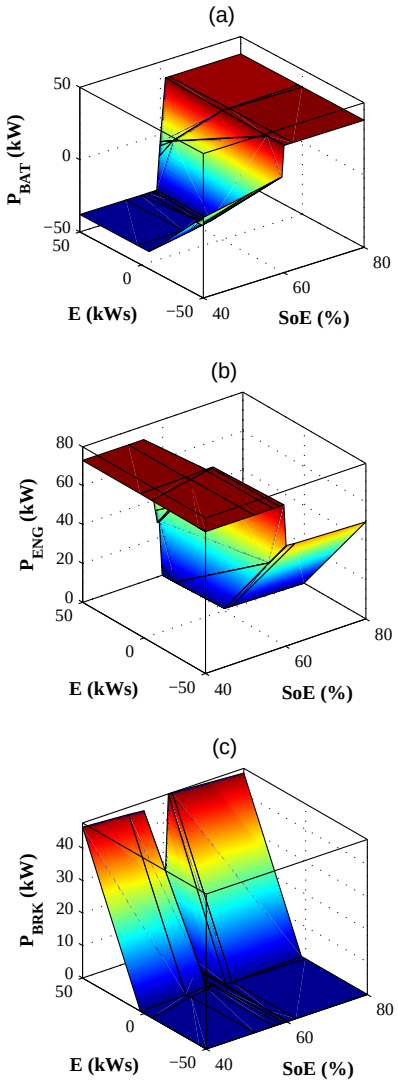

Figure 5.4: Control actions for $E_{ref} = 0$ and $SOE_{ref} = 60\%$ based on different initial conditions (a) battery power (b) engine power (c) braking power

One is to increase the engine power to charge the battery. The other one is to increase the braking power to get use of regenerative braking. Figure 5.4 shows that the controller uses both ways to get to the objective in region (I). The battery power should be negative indicating that it is being charged (Figure 5.4-a). Moreover, if $\delta E = |E - E_{ref}|$ is high, the braking power will be more (as shown in Figure 5.4-c)

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

Region(III) $E < E_{ref}$ and $SOE > SOE_{ref}$: In this region, we should accelerate the vehicle ($P_{BRK} = 0$). The electric drive can assist the engine since we have enough charge in the battery. Another objective of the controller is to minimize the engine power (in order to reduce fuel consumption). In region (III), there is no need to increase SOE so the electric drive can take care of propelling the vehicle. Moreover, the engine power is changing based on the magnitude of δE (as shown in Fig. 5.b)

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

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

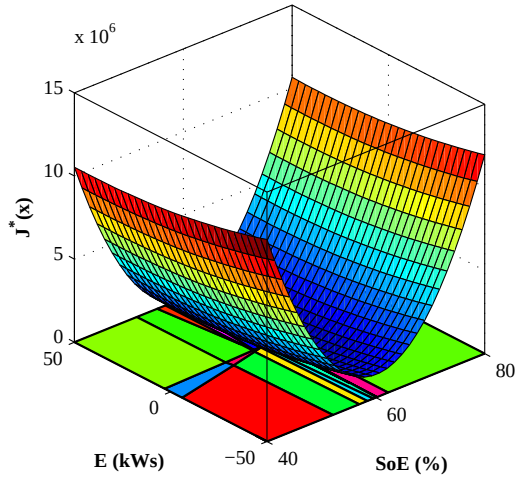

Figure 5.5: Cost function for $E_{ref} = 0$ and $SOE_{ref} = 60\%$ based on different initial conditions

5.4 Stability notes

As mentioned earlier, the closed-loop system with MPC controller is globally asymptotically stable if and only if the optimization problem is feasible.

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

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

that polytope is:

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

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

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

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

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

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

We show that if $\tilde{\mathbf{A}}$ is stable and $\tilde{\mathbf{B}}$ is bounded then the closed-loop system (5.10) is stable. For a discrete system, if $V_1(\tilde{Z}_k) > 0$ exists and $\Delta V_1(\tilde{Z}_{k+1}, \tilde{Z}_k) = V_1(\tilde{Z}_{k+1}) - V_1(\tilde{Z}_k) < 0$ then the system is exponentially stable in the sense of Lyapunov [173]. Since $\tilde{\mathbf{A}}$ is stable, we can find $P_1 > 0$ and $Q > 0$ such that:

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

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

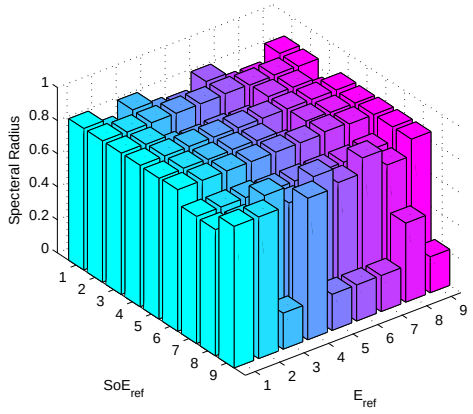

Figure 5.6: Spectral radius of $\tilde{A}$ for different levels of E_{ref} and SOE_{ref}

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

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

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

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

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

Therefore:

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

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

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

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

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

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

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

with a negative forward difference $\Delta\bar{V}(x_{k+1}, x_k) = \bar{V}(x_{k+1}) - \bar{V}(x_k) < 0$ when $x_k \in T_j \backslash 0$ and $x_{k+1} \in T_i$ [174].

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

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

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

Up to now, the stability of the closed-loop system of each set of polytopes is investigated, whereas the controller switches between different sets of polytopes to cover all operating points. A switched system is stable if all individual subsystems are stable and the switching is sufficiently slow, so as to allow the transient effects to dissipate after each switch. In [175], this property is formulated and justified using multiple Lyapunov techniques. In this work, the switching frequency depends on the dynamics of (SOE_{ref}, E_{ref}). As mentioned before, (SOE_{ref}, E_{ref}) are bounded values. As a result, we assume the following equations govern the dynamics of those reference values.

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

where ζ should be chosen in such a way that guarantees (5.19) stability and also make the switching system slower than the control-oriented model. For stability, ζ should be greater than unity, so that the poles of (5.18) are located inside the unity circle in z-plane. On the other hand, these poles should be far enough from the center of unity circle to slow down the system (5.18) response. We assume that ρ is the largest spectral radius of $\tilde{A}$ for all 81 sets of polytopes. The driving behavior determines ζ. If we choose $1/\zeta > \rho$, the switching system will be slower than the control-oriented model. As a result, if ζ is determined in such a way that the poles of the switching system are located inside the dark ring of Figure 5.7, the switched system will be stable.

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

$$\begin{aligned} X(k+1) = &\begin{bmatrix} \mathbf{I}_{2\times2} & \mathbf{O}_{2\times2} \\ \mathbf{O}_{2\times2} & (1/\zeta)\mathbf{I}_{2\times2} \end{bmatrix} X(k) \\ &+ \begin{bmatrix} \mathbf{B} \\ \mathbf{O}_{2\times3} \end{bmatrix} U(k) + \begin{bmatrix} \mathbf{O}_{2\times1} \\ \Gamma \end{bmatrix} \end{aligned} \tag{5.19}$$

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

$$\begin{bmatrix} \tilde{Z}_{k+1} \\ \tilde{Z}_{0,k+1} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{A}} & \mathbf{O}_{2\times2} \\ \mathbf{O}_{2\times2} & (1/\zeta)\mathbf{I}_{2\times2} \end{bmatrix} \begin{bmatrix} \tilde{Z}_k \\ \tilde{Z}_{0,k} \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{B}} \\ \Gamma \end{bmatrix} \tag{5.20}$$

Since the spectral radius of $(1/\zeta)\mathbf{I}_{2\times2}$ is less than unity and Γ is bounded ($\|\Gamma\| < 115$), according to the above discussion there is a $V_2(\tilde{Z}_{0,k}) = \tilde{Z}_{0,k}^T P_2 \tilde{Z}_{0,k} > 0$ such that $\Delta V_2(\tilde{Z}_{0,k+1}, \tilde{Z}_{0,k}) \leq 0$ where $P_2 > 0$.

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

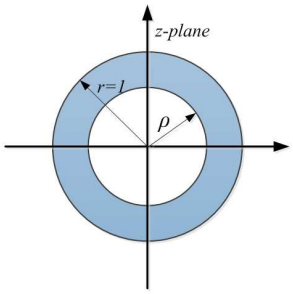

Figure 5.7: The locus of switching system poles in z-plane

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

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

5.5 eMPC Performance on the High-fidelity Powertrain Model

After finding the polytopes and the corresponding control actions, we need to implement the controller to the simulation model by using low-level controls. Basically, we have to change the provided power to torque and speed for different components. Figure 5.8 shows the procedure that is done at each control time step. At the beginning, we have E_{ref} and SOE_{ref} as well as initial SOE and demanded energy that are given to the eMPC controllers. By using the mentioned point location algorithm, the appropriate controls among the polytopes can be found. We are looking for T_e, T_m, and ω_g. By having T_e we can control the engine throttle to the desired engine torque. On the propulsion side, we

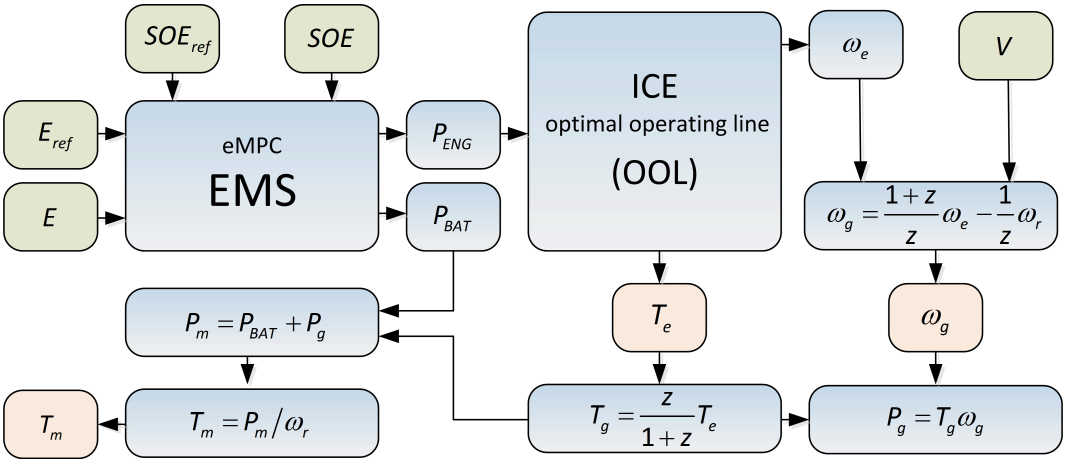

Figure 5.8: Low-level controls implementation

have P_{BAT} and P_{ENG}. Once we got P_{ENG}, we can use the optimal operating line of the engine, which gives us the most efficient operating point for the given P_{ENG}. Now, we have the engine speed and torque for the optimum operating point. The engine torque setpoint can directly be given to the engine low-level controller. If we measure the vehicle velocity, we will be able to get the MG1 speed setpoint by using the speed constraint relation on the first planetary gear set ($z = \frac{s_1}{r_1}$). Meanwhile, if we use static torque relation on the planetary gear set, we can find the MG1 torque based on the engine torque. Now MG1 power is calculated and we can find the MG2 power, since we have got the P_{BAT} from eMPC controller. By measuring the MG2 speed at the current time step, we are able to find the last setpoint value which is MG2 torque. Now, we can implement the controller to the simulation model.

In this section, the eMPC EMS is applied to the high-fidelity simulation model including the low-level controls (see section 4.6). The results of MIL simulation are reviewed in Figure 5.9 to Figure 5.14.

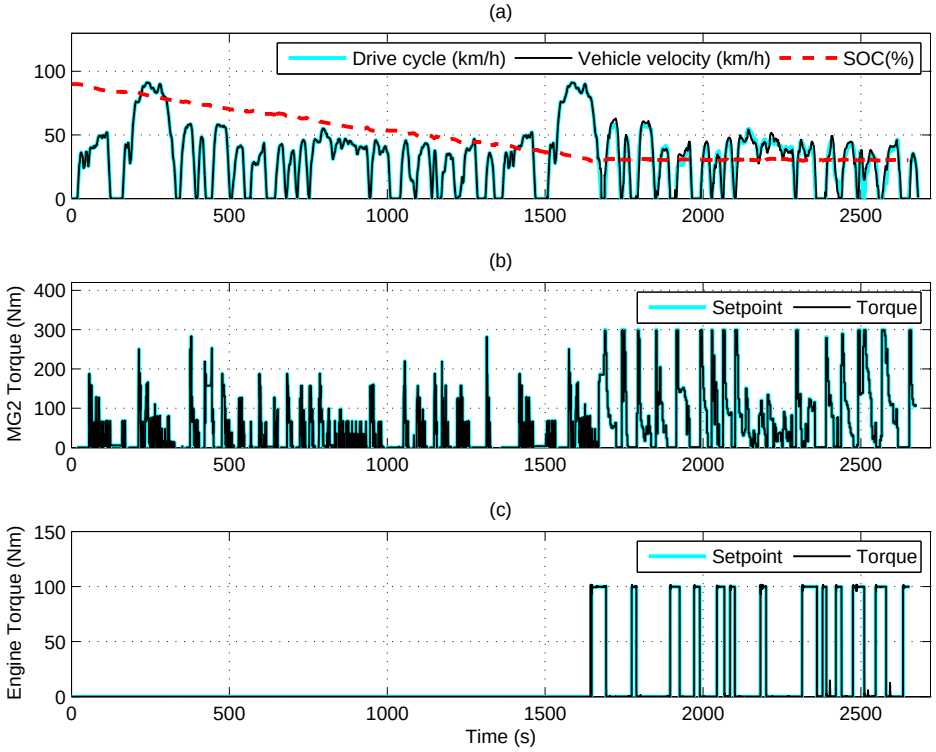

Figure 5.9: CDCS eMPC strategy with emission control: (a) Velocity and Battery SOC (b) MG2 Torque (c) Engine Torque

5.5.1 No Knowledge of Trip Information

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

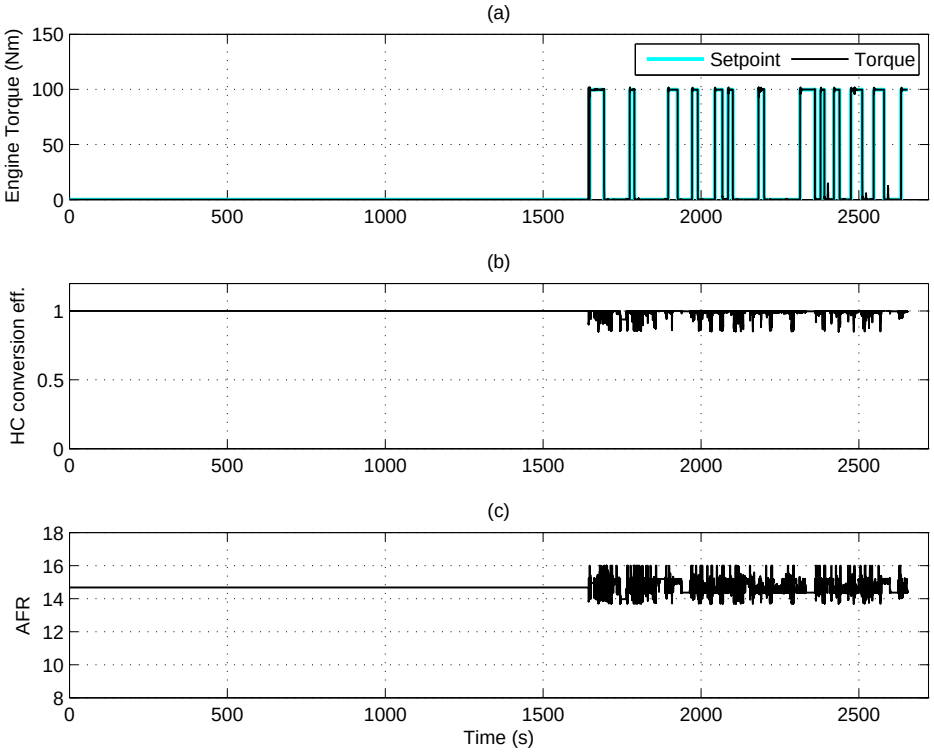

Figure 5.10: CDCS eMPC strategy with emission control: (a) Engine Torque (b) HC conversion efficiency (c) Air-to-fuel ratio (AFR)

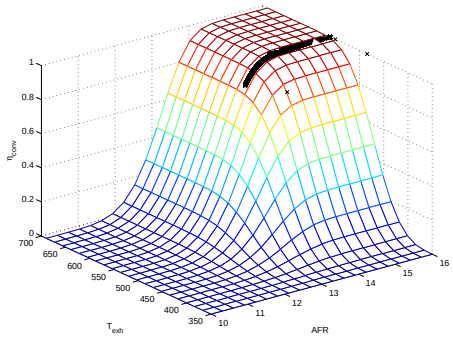

Figure 5.11: Catalyst conversion efficiency for CDCS eMPC strategy

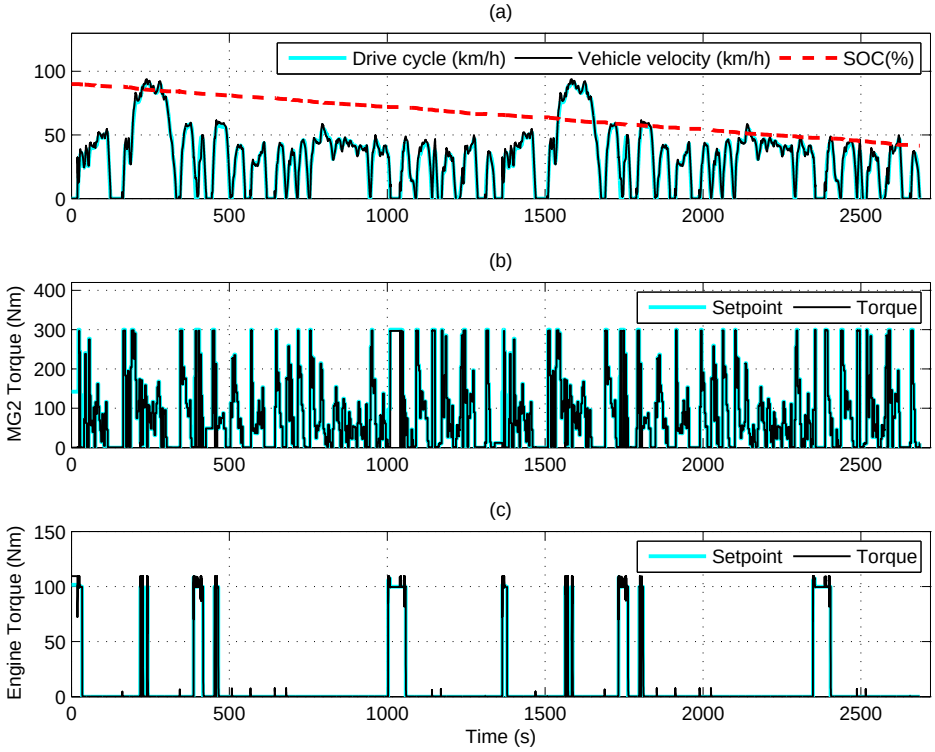

Figure 5.12: Blended eMPC strategy with emission control: (a) Velocity and Battery SOC (b) MG2 Torque (c) Engine Torque

5.5.2 Known Travelling Distance

Figures 5.12 to 5.14 show similar results for the blended mode strategy. Fuel consumption for this strategy is $1.81l/100km(130MPG)$ by considering engine emissions control.

5.5.3 Discussions

Fuel economy for MIL testing using the high-fidelity simulation model is compared in Table 5.1. Table 5.1 shows that explicit model predictive control reduces fuel economy by 1.74% and improves it by 4.84% for CDCS and blended mode strategies, respectively when compared to the MPC approach. But note that the primary purpose of considering

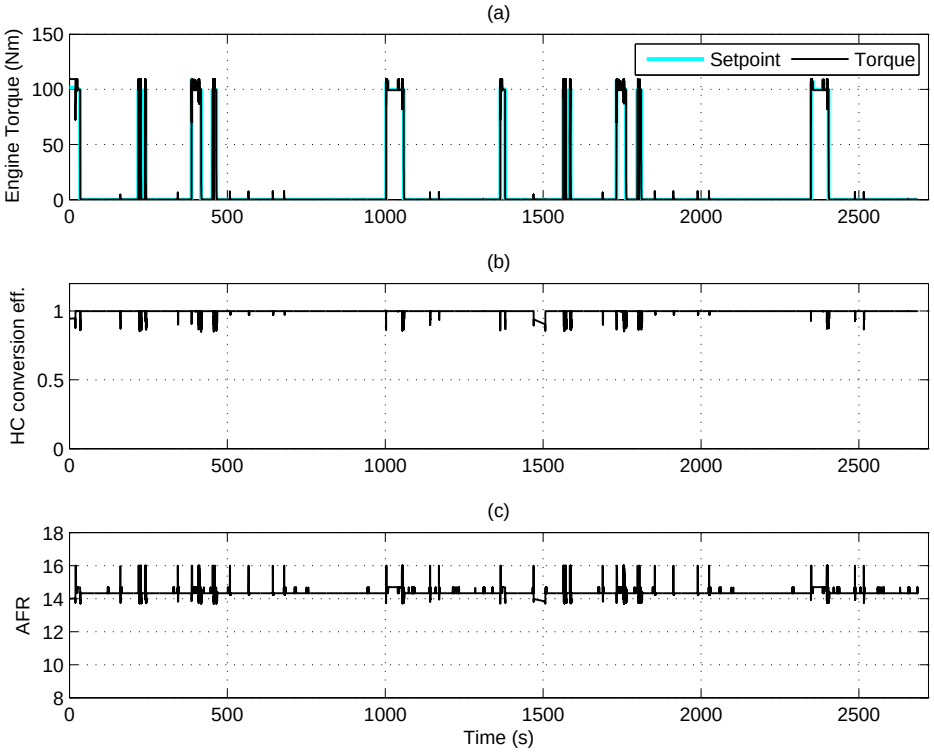

Figure 5.13: Blended eMPC strategy with emission control: (a) Engine Torque (b) HC conversion efficiency (c) Air-to-fuel ratio (AFR)

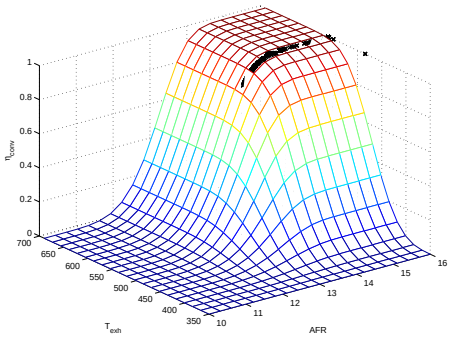

Figure 5.14: Catalyst conversion efficiency for Blended eMPC strategy

explicit model predictive control was to maintain the performance of MPC while make it faster during implementation stage. The MIL simulation takes 1885 s for simulating 2828 s, which is three times faster than MIL using the MPC EMS on the average. This shows that eMPC has a superior potential to be implemented to a commercial control hardware with limited computational power. As a measure of drivability performance, eMPC EMS can follow the designated drive cycle with the root mean square error of 0.87 km/h and 0.89 km/h for CDCS and Blended mode strategies, respectively.

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

Control Strategy	MPC (MPG)	eMPC (MPG)	improvement (%)
Charge Depletion/Charge Sustenance	115	113	-1.74
Linear blended mode	124	130	4.84

5.6 Control-relevant Parameter Estimation (CRPE)

Parameter estimation is an essential step in modeling. Control relevance in parameter estimation refers to the suitable selection of design variables in the estimation algorithm given that the intended purpose of the model is control system design. Its basis lies in the fact that if the bias distribution is appropriately shaped, then low-order representations of the plant will be obtained which will capture all the essential plant dynamics for control system design [176]. Real-time implementability of model predictive controller is closely related to how simple is the control-oriented model. Simple models may not be so accurate to capture the essential dynamics of the main plant. In this way, such simple models are preferable that their parameters are estimated in a way that captures all dynamics of the plant. But finding the parameters to make the control-oriented model cover all the operating points of the plant may not be feasible. Instead, one can do the parameter identification within a specific range of the plant operating points that is essential according to controls design

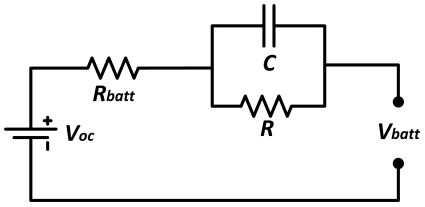

Figure 5.15: Thevenin equivalent model of the battery

requirements. The derived control-oriented model is only valid for an active frequency range that the controls require. The control-oriented model may not represent the plant for the whole range of its operating points, but it is simple and adequately accurate to pass the real-time implementability requirements needed for designing predictive controls.

5.6.1 Battery Thevenin Model

As mentioned earlier in this chapter, the performance of model predictive controller is closely related to the accuracy of the control-oriented model inside it. As a result, we address a method to improve PHEV control-oriented model accuracy in such a way that it remains simple enough to keep the size of eMPC look-up tables small enough for real-time implementation purposes. Since the battery pack plays the most important role in determining full electric range as well as PHEV fuel economy and emissions performance, we consider a more detailed model of this component inside the control-oriented model. The previous control-oriented model (5.1) contains two state variables. In this section, one extra state variable is added to the previous model. One added state variable belongs to the battery component. In the new control-oriented model, we replace the power-based model of the battery with the Thevenin equivalent model of it (Figure 5.15).

According to the definition of battery state of charge:

$$\dot{SOC} = -\frac{i_{batt}}{Q} = -\frac{1}{R_{batt}Q}(V_{oc} - V - V_{batt}) \tag{5.22}$$

where Q and V are battery pack capacity and the voltage across RC-component of Figure

5.15, respectively. One can find a differential equation for V as:

$$\dot{V} = -\frac{V}{RC} + \frac{1}{R_{batt}C}(V_{oc} - V - V_{batt}) \tag{5.23}$$

The control input in (5.22) and (5.23) is V_{batt}. On the other hand, we find an expression which can relate V_{batt} to P_{BAT} as battery power is directly related to the total propulsion power. Therefore,

$$P_{BAT} = i_{batt}V_{batt} = \frac{V_{batt}}{R_{batt}}(V_{oc} - V - V_{batt}) \tag{5.24}$$

which can be rewritten as:

$$V_{batt} = \frac{1}{2}\{(V_{oc} + V) - \sqrt{(V_{oc} + V)^2 + 4R_{batt}P_{BAT}}\} = f(V, P_{BAT}) \tag{5.25}$$

In order to get a linear control-oriented model, $f(V, P_{batt})$ can be linearized within operating range of V and P_{BAT}. If $f(V, P_{BAT}) = b_1V + b_2P_{BAT} + b_3$ the battery control-oriented model can be written as

$$\begin{aligned} \dot{SOC} &= -\tfrac{1}{R_{batt}Q}\{V_{oc} - b_3 - (1 - b_2)V - b_2P_{BAT}\} \\ \dot{V} &= -\tfrac{V}{RC} + \tfrac{1}{R_{batt}C}\{V_{oc} - b_3 - (1 - b_2)V - b_2P_{BAT}\} \end{aligned} \tag{5.26}$$

(5.26) in combination with (5.27) makes the new control-oriented model with 3 state variables

$$\dot{E} = P_{BAT} + P_{ENG} - P_{BRK} \tag{5.27}$$

Now, the four parameters of Thevenin equivalent model of battery should be estimated according to controls requirements in order to get more realistic behavior of the battery. This will be addressed in the following section.

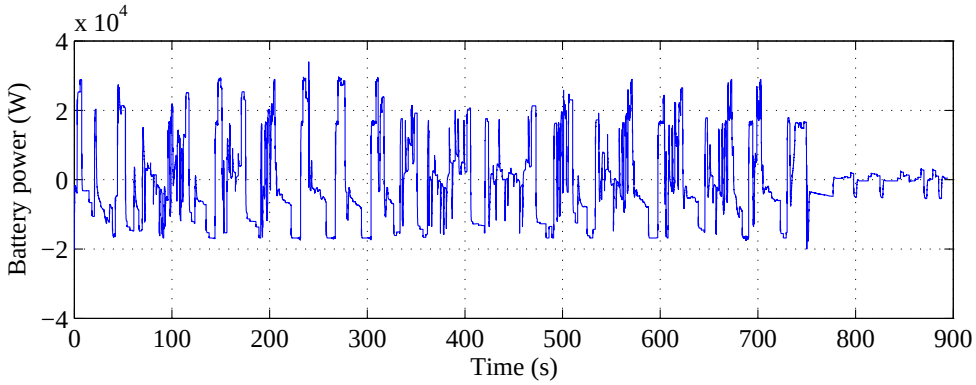

Figure 5.16: Time history of the battery power, applying eMPC to the PHEV for UDDS driving cycle

5.6.2 Battery Parameters Estimation

In this part, the battery parameters within the operating range of the EMS are estimated. To this end, we find the battery active frequency range based on a great deal of simulations that have been already done. Therefore, the experimental data needed for parameter estimation is derived from that specific frequency range. As the battery model parameters are identified, the new control-oriented model can be determined which is followed by a new eMPC EMS design. The new EMS is expected to have better performance over the previous one.

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

As shown in Figure 5.16, the power signal contains high-frequency oscillations, which makes the process of parameter estimation much harder.

To facilitate the identification process, the dominant frequency range of the power signal can be extracted using a spectrum analysis, after which a low-pass filter should be designed

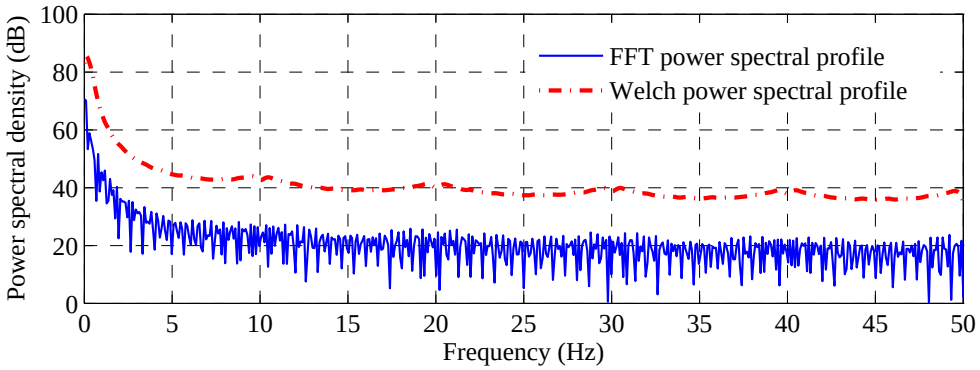

Figure 5.17: Power spectral density analysis of the chemistry-based battery power signal, when an eMPC scheme is applied to the PHEV

to regenerate the modified signal. Based on both Welch and Fast Fourier Transform (FFT) power spectral estimate (as shown in Figure 5.17), the frequencies higher than 5 Hz have less contribution in representing the battery power signal.

Accordingly, a digital filter with finite-duration impulse response such as least-square approaches, seems to be promising candidate in this case, by which the refined power signal to the battery can be regenerated. The state of charge is then extracted, applying the refined power signal to the high-fidelity battery model. Assuming the acquired input-output to the battery as the reference data, the parameters for the equivalent circuit model can be estimated [177]. In this work, a dual polarization equivalent circuit model with a double-RC circuit (Figure 5.18) is utilized, by which two key physical phenomena in lithium-ion batteries are incorporated in the equivalent system. The model, which includes polarizations of different time scales, has been experimentally examined for lithium-ion batteries in terms of dynamic performance and the state-of-the-charge estimation [178].

Although, Thevenin equivalent model is appropriate for control-oriented modeling, it cannot represent the physics of lithium-ion battery as properly as a dual-polarization model. As such, the parameters of the dual-polarization model are estimated first. Then, the dual-polarization model is approximated by a Thevenin equivalent model. In other words, the impendence of two RC circuits are approximated with only one RC circuit:

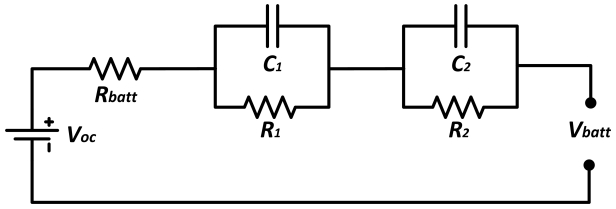

Figure 5.18: Dual-polarization model of the battery

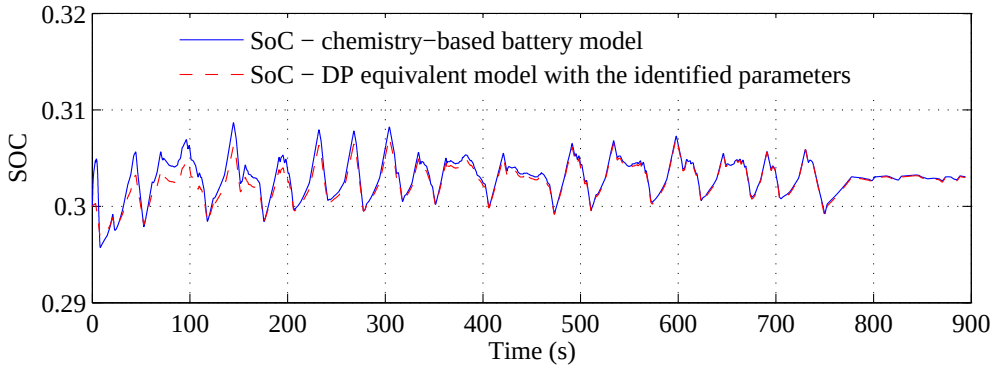

Figure 5.19: Time history of the SOC profile for the dual-polarization equivalent and chemistry-based battery model

$$\frac{R_1}{R_1C_1s+1}+\frac{R_2}{R_2C_2s+1}\cong\frac{R}{RCs+1} \tag{5.28}$$

Simulation results for the state of charge of the equivalent battery model, compared to the experimental ones derived from the full chemistry-based battery model, are shown in Figure 5.19. A great match signifies the efficacy of the homotopy optimization procedure in identifying the model parameters, which was used earlier in validating the PHEV simulation model.

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

Table 5.2: Identified values for the parameters of the equivalent dual-polarization circuit model

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

parameters are listed in Table 5.2.

5.6.3 CRPE Control-oriented Model

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

$$\bar{Z}(k+1) = \bar{\mathbf{A}}\bar{Z}(k) + \bar{\mathbf{B}}U(k) + \bar{F} \tag{5.29}$$

where $\bar{Z} = \begin{bmatrix} SOC, & V_{batt}, & E \end{bmatrix}^T$ and $U = \begin{bmatrix} P_{BAT}, & P_{ENG}, & P_{BRK} \end{bmatrix}^T$.

$$\bar{\mathbf{A}} = \begin{bmatrix} 1 & \bar{a}_1 & 0 \\ 0 & \bar{a}_2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$
$$\bar{\mathbf{B}} = \begin{bmatrix} \bar{a}_3 & 0 & 0 \\ \bar{a}_4 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$
$$\bar{\mathbf{F}} = \begin{bmatrix} \bar{a}_5 \\ \bar{a}_6 \\ 0 \end{bmatrix} \tag{5.30}$$

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

5.7 CRPE-eMPC Formulation for a PHEV

We can rewrite (5.31) in the way of (5.3) but here we added one parameter to the multi-parametric quadratic programming problem. So the new problem has 5 parameters, because of one added state variable in the new control-oriented model. Now, the control-relevant parameter estimated eMPC (CRPE-eMPC) energy management can be designed.

$$\min_U \sum_{j=1}^{N_p} \left\{ \bar{Y}^T(j)\bar{\mathbf{Q}}\bar{Y}(j) + U^T(j)\bar{\mathbf{R}}U(j) \right\}$$

$$s.t.$$

$$\begin{bmatrix} SOC_{min} \\ V_{batt,min} \\ E_{min} \end{bmatrix} \leq \begin{bmatrix} SOC \\ V_{batt} \\ E \end{bmatrix} \leq \begin{bmatrix} SOC_{max} \\ V_{batt,max} \\ E_{max} \end{bmatrix}$$

$$U_{min} \leq U \leq U_{max} \tag{5.31}$$

Where $\bar{Y} = \begin{bmatrix} SOC - SOC_{ref}, & E - E_{ref} \end{bmatrix}^T$, and N_p is the prediction horizon length.

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

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

5.7.1 CRPE-eMPC Controls Regions

By solving the multi-parametric quadratic problem, the number of polytopes can be seen in Figure 5.20:

Total number of polytopes is 8439. The resultant polytopes set for $E_{ref} = 0$ and $SOC_{ref} = 60\%$ is demonstrated in Figure 5.23.

As it is seen, there is one added dimension to the polytope set in Figure 5.21 as compared to Figure 5.3. For any given V_{batt}, any cross section which contains $(SOC - E)$ generally looks similar to Figure 5.3.

In this case, we cannot visualize the control actions like Figure 5.4. But for any cross section of Figure 5.21 for any given V_{batt}, the analysis on E and SOC remains valid.

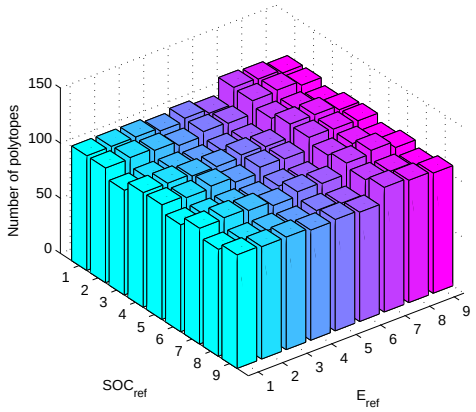

Figure 5.20: Number of polytopes for different levels of E_{ref} and SOC_{ref}

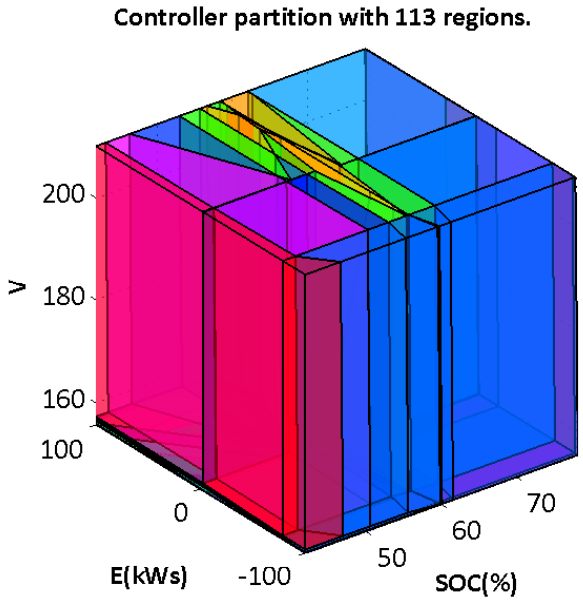

Figure 5.21: Polytope set for $E_{ref} = 0$ and $SOC_{ref} = 60\%$

Figure 5.22 shows the control actions versus different measured values of SOC and E (initial conditions) at current sampling instance for $V_{batt} = 180V$.

5.7.2 CRPE-eMPC Stability Notes

Here we can exactly follow what is mentioned in section 5.4. In each set of polytopes which belongs to a definite SOC_{ref}, E_{ref}, the controller drives the state variables to the mentioned reference values in finite time steps. To prove the local stability of the closed-loop system, we pick the polytope which contains $\bar{Z}_0$. The control corresponding to that polytope is:

$$\hat{\bar{U}} = \bar{f}_0 \hat{\bar{Z}} + \bar{g}_0 \tag{5.32}$$

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

$$\bar{Z}(k+1) = (\bar{\mathbf{A}} + \bar{\mathbf{B}}\bar{f}_0)\bar{Z}(k) + \bar{\mathbf{B}}\bar{g}_0 \tag{5.33}$$

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

$$\begin{aligned} \tilde{\bar{Z}}(k+1) &= (\bar{\mathbf{A}} + \bar{\mathbf{B}}\bar{f}_0)\tilde{\bar{Z}}(k) \\ +\bar{\mathbf{B}}\bar{g}_0 + (\bar{\mathbf{A}} + \bar{\mathbf{B}}\bar{f}_0 - I_{2\times 2})\bar{Z}_0 &= \tilde{\bar{\mathbf{A}}}\tilde{\bar{Z}}(k) + \tilde{\bar{\mathbf{B}}} \end{aligned} \tag{5.34}$$

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

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

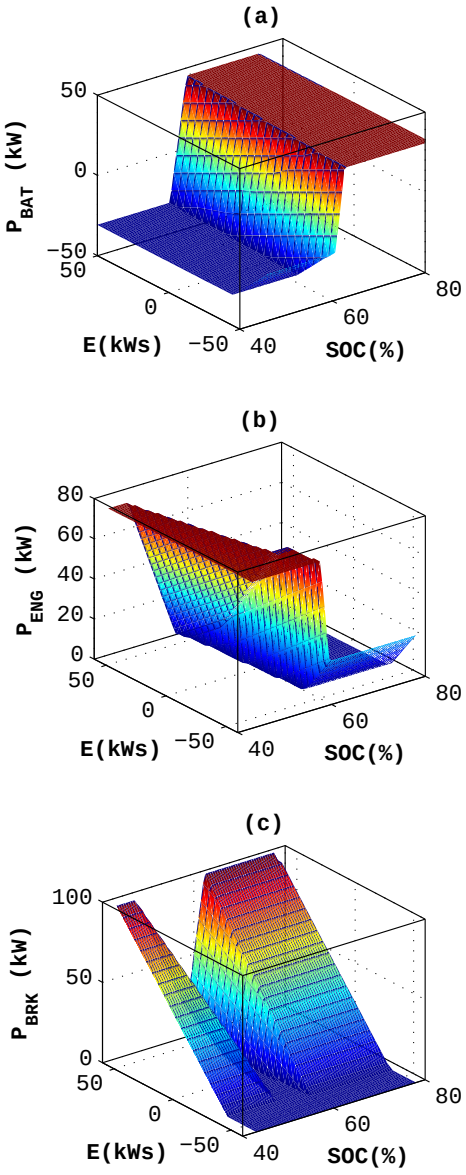

Figure 5.22: Control actions for $V_{batt} = 180V$, $E_{ref} = 0$, and $SOC_{ref} = 60\%$ based on different initial conditions (a) battery power (b) engine power (c) braking power

$$\bar{V}(\tilde{\bar{Z}}) = \sum_{j=1}^{N_p} \{\tilde{\bar{Z}}^T(j)\bar{\mathbf{Q}}\tilde{\bar{Z}}(j)\} > 0 \Rightarrow \Delta\bar{V}(\tilde{\bar{Z}}_{k+1}, \tilde{\bar{Z}}_k) < 0 \tag{5.35}$$

For investigating the stability of the switched system, we assume that (5.35) remains valid and $\bar{\rho}$ is the largest spectral radius of $\tilde{\bar{A}}$ for all 81 sets of polytopes. If $\bar{\zeta}$ is chosen greater than $\bar{\rho}^{-1}$, the switched system will be stable.

$$\begin{bmatrix} SOC_{ref}(k+1) \\ E_{ref}(k+1) \end{bmatrix} = \begin{bmatrix} 1/\bar{\zeta} & 0 \\ 0 & 1/\bar{\zeta} \end{bmatrix} \begin{bmatrix} SOC_{ref}(k) \\ E_{ref}(k) \end{bmatrix} + \Gamma \tag{5.36}$$

5.8 CRPE-eMPC Performance on the High-fidelity Powertrain Model

In this part, the CRPE-eMPC EMS is applied to the high-fidelity simulation model considering the low-level controls (see section 4.6). The results of MIL simulation are reviewed in Figures 5.23 to 5.28.

5.8.1 No Knowledge of Trip Information

Figure 5.23-a shows the vehicle drivability performance along two UDDS drive cycles for the CDCS strategy. Figures 5.23-b,c show the performance of low-level controls in tracking the setpoints determined by CRPE-eMPC EMS. Figure 5.24 demonstrates the emissions control performance. The sliding mode controller keeps the HC conversion efficiency around unity by controlling the air-fuel ratio and ignition timing as shown in Figure 5.25. PHEV fuel consumption is reduced to $1.97l/100km(119MPG)$ by using the CRPE-eMPC EMS.

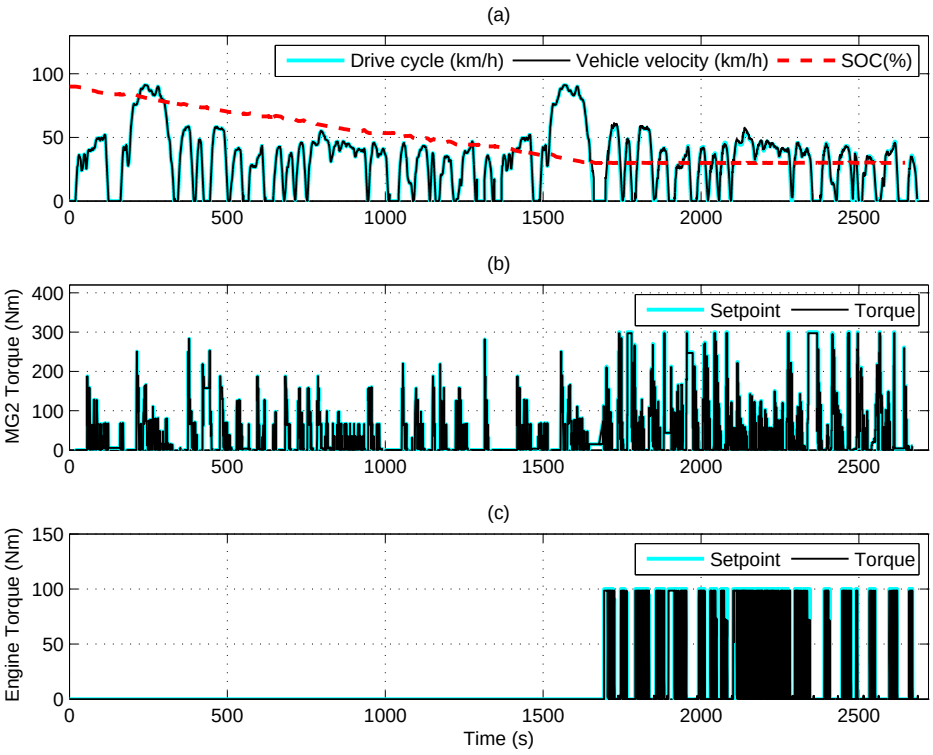

Figure 5.23: CDCS CRPE-eMPC strategy with emission control: (a) Velocity and Battery SOC (b) MG2 Torque (c) Engine Torque

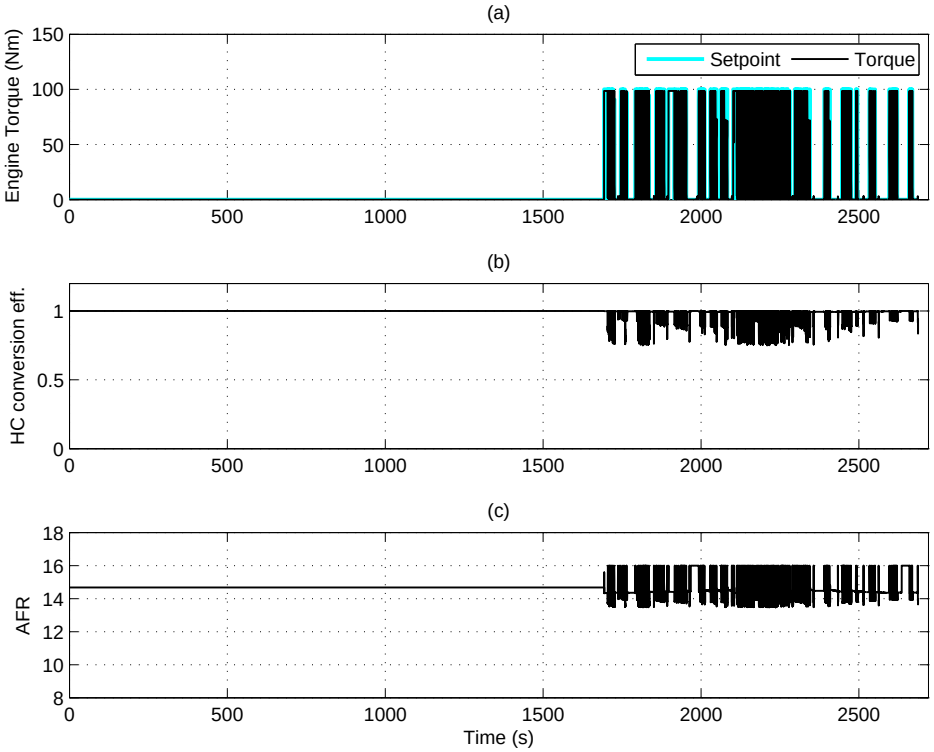

Figure 5.24: CDCS CRPE-eMPC strategy with emission control: (a) Engine Torque (b) HC conversion efficiency (c) Air-to-fuel ratio (AFR)

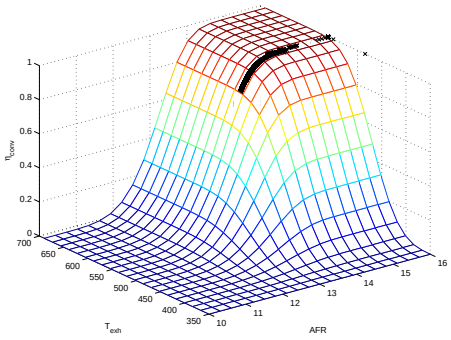

Figure 5.25: Catalyst conversion efficiency for CDCS CRPE-eMPC strategy

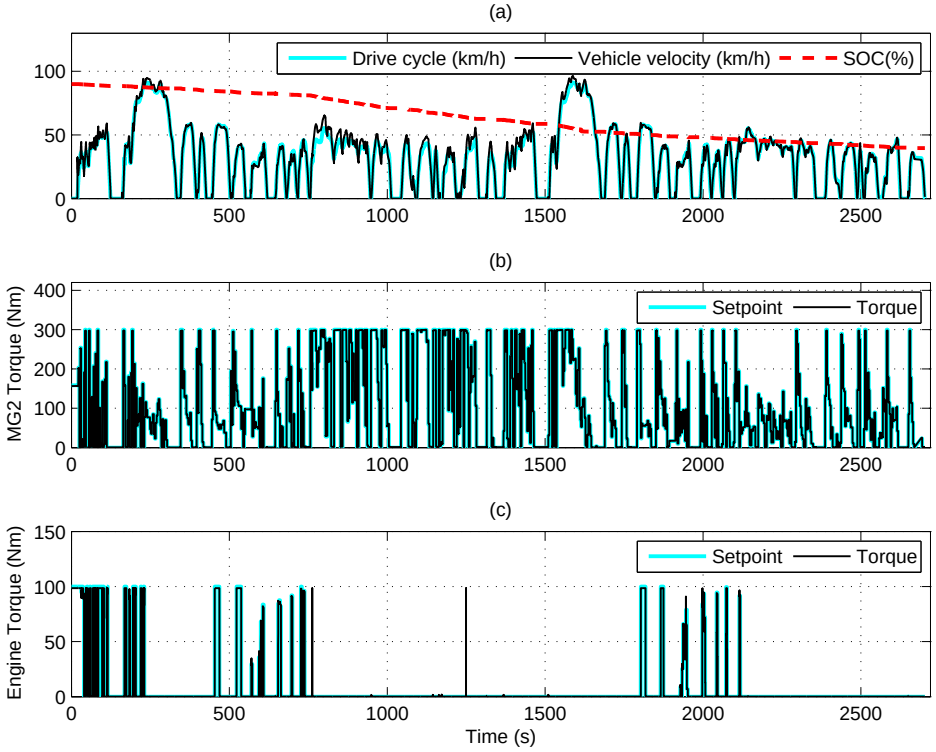

Figure 5.26: Blended CRPE-eMPC strategy with emission control: (a) Velocity and Battery SOC (b) MG2 Torque (c) Engine Torque

5.8.2 Known Travelling Distance

Figures 5.26 to 5.28 show the results for the blended mode strategy. Fuel consumption in this simulation is $1.68l/100km(140MPG)$ by considering engine emissions control.

5.8.3 Discussions

Fuel economy for MIL testing by using CRPE-eMPC EMS is reviewed in Table 5.3. It shows that CRPE-eMPC EMS improves fuel economy by 5.31% and 7.69% for CDCS and blended mode strategies, respectively in comparison to the eMPC high-level controller which was designed based on a control-oriented model with 2 state variables. The MIL simulation

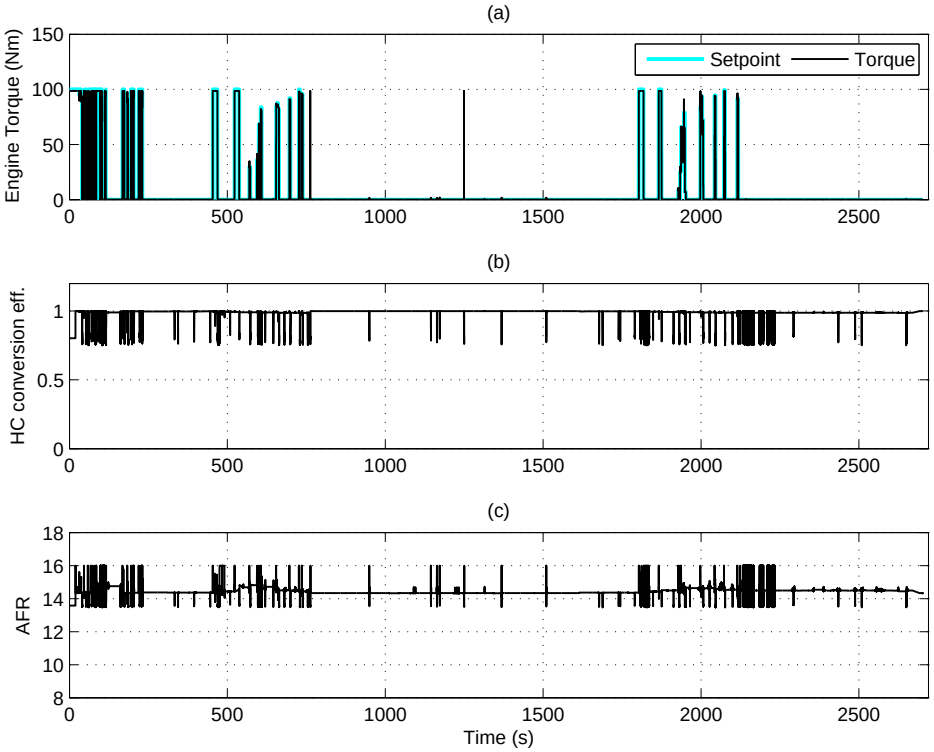

Figure 5.27: Blended CRPE-eMPC strategy with emission control: (a) Engine Torque (b) HC conversion efficiency (c) Air-to-fuel ratio (AFR)

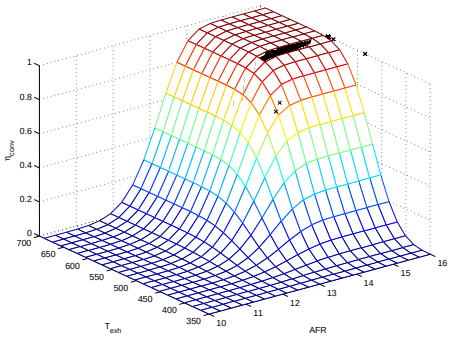

Figure 5.28: Catalyst conversion efficiency for Blended CRPE-eMPC strategy

Table 5.3: MIL with the high-fidelity powertrain model: Fuel economy for different control strategies

Control Strategy	eMPC (MPG)	CRPE-eMPC (MPG)	improvement (%)
Charge Depletion/Charge Sustenance	113	119	5.31
Linear blended mode	130	140	7.69

can still be performed 1.5 times faster than real time. Therefore adding one state variable to the previous control-oriented model (section 5.2.1) doesn't slow down the simulation. In fact, the performance of the controller is also improved in terms of fuel economy while maintaining the emissions performance. As a measure of drivability performance, CRPE-eMPC EMS can follow the designated drive cycle with the root mean square error of 0.45 km/h and 0.97 km/h for CDCS and Blended mode strategies, respectively.

Figure 5.29 shows why the fuel economy has been improved by using CRPE-eMPC approach for designing EMS. More accurate battery model inside the control-oriented model has led to greater P_{BAT} which has reduced the P_{ENG} and consequently reduced fuel consumption.

5.9 Chapter Summary

In this chapter, the explicit model predictive control approach was used to design an energy management strategy for a plug-in hybrid powertrain. In this way, a new control-oriented model was proposed with two state variables. We implemented the developed controller to a PHEV simulation model and reduced the simulation time by 44% and improved fuel economy by 16% on average in comparison to MPC.

According to the model-in-the-loop simulation results, the designed eMPC energy management can be applied to the high-fidelity simulation model three times faster that its implicit MPC counterpart, while maintaining the expected performance.

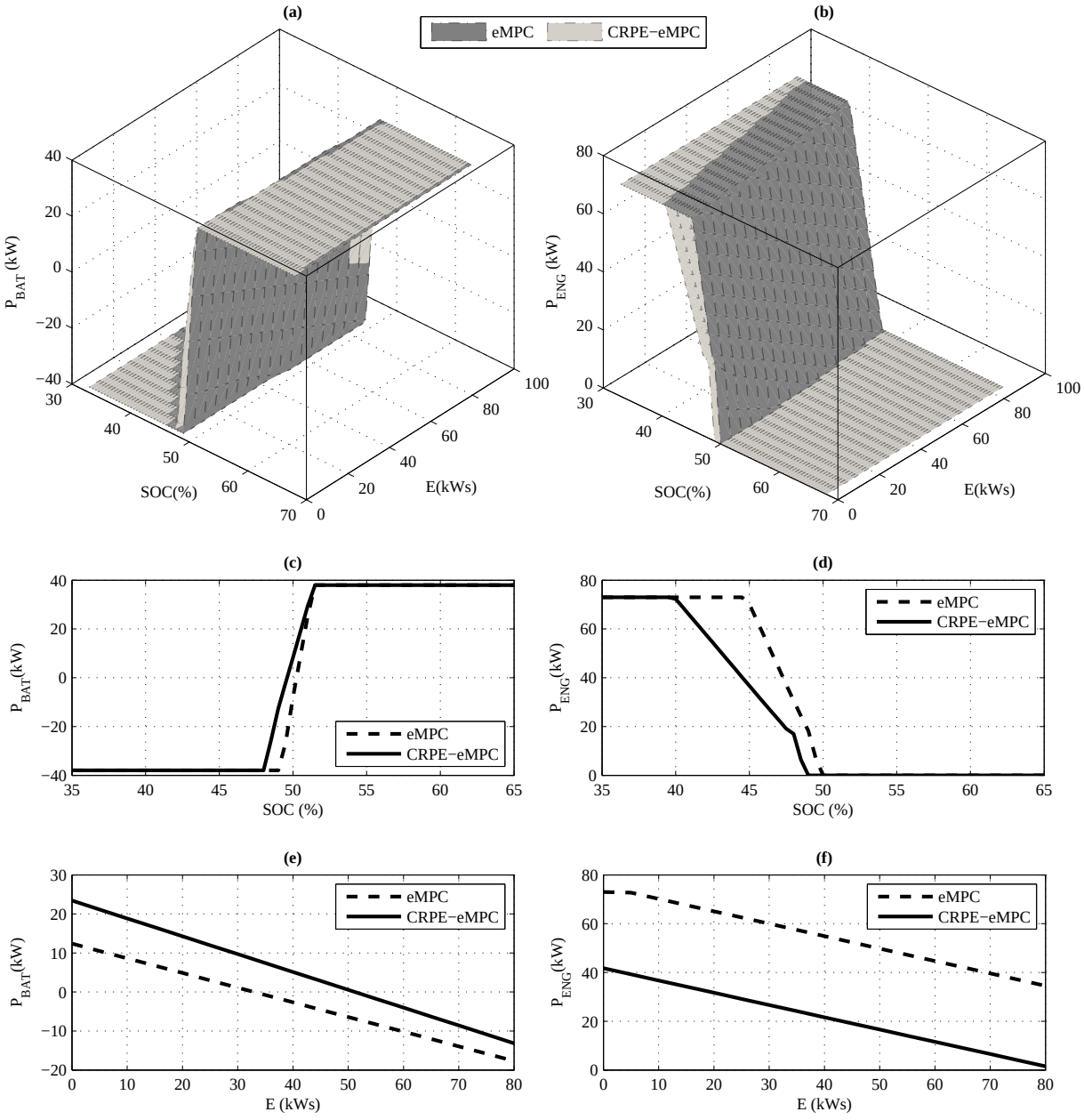

Figure 5.29: eMPC vs. CRPE-eMPC:(a)Battery power (b)Engine power (c)Battery power for constant $E = 10kWs$ (d)Engine power for constant $E = 10kWs$ (e)Battery power for constant $SOC = 50\%$ (e)Engine power for constant $SOC = 45\%$

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

Chapter 6

Hardware-in-the-loop Simulation

In this chapter, the essential elements for performing hardware-in-the-loop (HIL) testing are introduced. Then, the eMPC EMS is programmed on a MotoTron electronic control system (ECU) hardware and its performance is evaluated through HIL testing.

6.1 Introduction

An effective approach for rapid prototyping and evaluation of vehicle control system is real time hardware-in-the-loop testing. Real-time simulation and support for HIL are increasingly recognized as essential tools for engineering design [179]. This reduces dependence on prototype vehicles, especially in the early stages of a program, and subsequently reduces the time, effort and resources required to build and support them. The objective is to build the first full prototype car much later in a development program, and in such a way that it does not require major modification before going into mass production [180].

HIL simulation often requires significantly less hardware than physical prototyping, therefore being cheaper and quicker to build. HIL simulators often achieve fidelity levels unattainable through purely virtual simulation by prototyping those components whose dynamics or other attributes (e.g. transient emission formation in engines) are not fully

understood. On the other hand, HIL simulations of complex physical phenomena run faster than purely virtual simulations of the same phenomena (e.g. IC engine simulations based on Computational Fluid Dynamics). Systems that normally operate in highly variable environments (e.g., off-road vehicle suspension systems) can often be tested in controlled lab settings through HIL simulation, which significantly increases repeatability, and often makes it possible to simulate a given system over a much broader range of operating conditions than what is feasible via purely physical prototyping. Moreover, HIL simulation makes it possible to simulate destructive events without incurring a costly destruction. HIL simulators can be used to train human operators (e.g., airplane pilots) of safety-critical systems (e.g., supersonic aircraft) in significantly safer environments (e.g., flight simulators). It is noteworthy that HIL simulation allows different teams to develop different parts of a system in hardware without losing sight of integration issues, thereby enabling concurrent systems engineering [89]. In this chapter, we present the HIL testing procedure of the proposed eMPC EMS for Toyota Prius plug-in hybrid powertrain.

6.1.1 ECU Validation Procedure

The electronic control unit (ECU) strategy prove-out is done in successive steps on off-line simulations on a desktop, HIL, dynamometer, and vehicle, with each step bringing in additional "real" substitutes for the virtual models. ECU strategy procedure in this sequence has some advantages. First, it ensures that component-level testing is done prior to subsystem and system level testing. Second, it capitalizes on the fact that ECUs are usually available much sooner than vehicle hardware prototypes, enabling a large amount of testing to be completed prior to vehicle manufacturing [6]. Figure 6.1 shows the steps in ECU validation [181].

The off-line simulations used within the early phases of the development process are often called model-in-the-loop simulations (MIL). For the modeling of the vehicle and functions at the MIL stage, standard tools such as MATLAB/Simulink and MapleSim can be used. Next step is software-in-the-loop (SIL) simulation, where the functional model of an ECU is replaced by C-code and coding errors can be found independent of the future

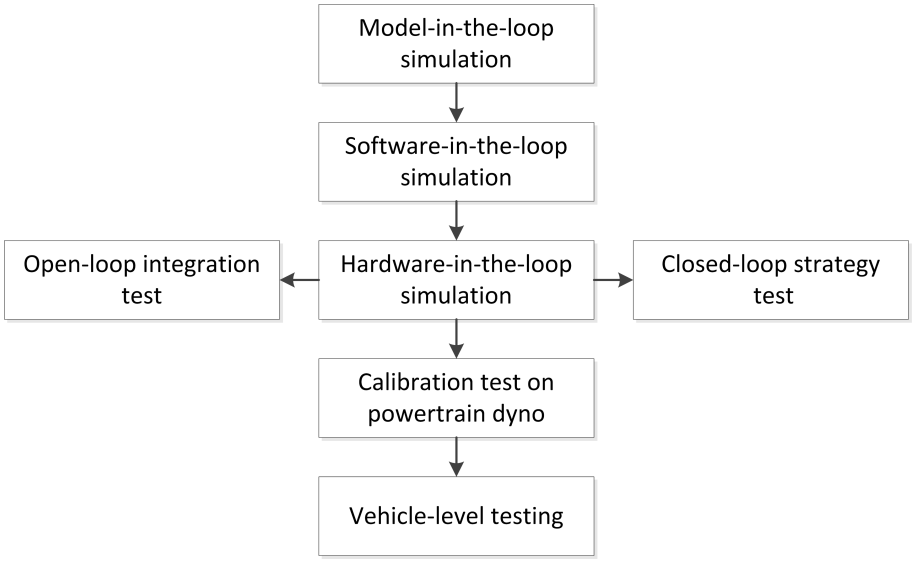

Figure 6.1: ECU validation procedure

ECU hardware. In the next step, actual hardware of the ECU is available and the tests can be supported by HIL simulation. The HIL simulation consists of two parts: open-loop integration and closed-loop strategy testing [181]. Open-loop test platform uses a simpler model inside a real-time computer in order to check the functionality of user inputs and low-level I/O interfaces, for instance, push button start, or gear shift command. Closed-loop test platform needs the dynamical model of the plant implemented to the real-time computer which provides feedback information from the plant.

After the software tests are successfully passed, the calibration of the ECUs can be done on the test-bench or in the vehicle. At this point, changes to the functions and system specifications are time consuming, expensive, and in most cases not possible [100].

The explained validation procedure presents some challenges for HEV control system unit such as [6]: (1) adequate computational power required to execute not just an engine plant model, but models of other controller units as well as a complete vehicle model, and (2) capability of extensive CAN communication support, since the HEV controllers that are communicating with the ECU should be modeled [182].

6.1.2 Virtual Simulation Model Requirements

Since the performance of the ECU is tested in a virtual vehicle environment, the appropriate vehicle dynamics need to be modeled. There are four important elements to the HIL system model that is used for ECU testing: physical model of the plant (such as engine, transmission, battery, motor, etc.), sensors (such as sensors for engine speed, engine intake manifold pressure, battery voltage, etc.), actuators (such as electronic throttle body, fuel injector, etc.), and external systems that interact with the vehicle model (such as human driver, road grade, etc.) [182]. The scope of the HIL system model is primarily driven by the level of functional testing required. On one hand, the HIL test bed could be used as an open-loop tester to verify some low-level (input/output driver level) ECU software functionality. In this case, the vehicle model could be constant values or uncorrelated signal traces that drive all the inputs of the controller. On the other hand, the HIL test bed could be used for verifying closed-loop dynamic functionality [6].

To use the HIL simulation, real-time capable simulation models are needed. Developing those models is a real challenge since they have to be accurate and fast enough at the same time [100]. The virtual models within a HIL simulator must typically meet two requirements. First, they must capture the essential dynamics of the virtually prototyped systems accurately enough to enable the HIL simulator to achieve its required design goals. Secondly, they must run in real time, a requirement that often translates into a bound on model complexity. These two requirements, that are, fidelity and simplicity, typically conflict [89]. The literature recognizes this conflict and considers a dynamic system model to be proper if it optimally balances these two requirements. The MapleSim high-fidelity simulation model in chapter 3 is physics-based and captures transients in comparison to look-up table models, and real-time capable at the same time, which has been cross-validated with experimental data available in the Autonomie software.

6.1.3 Real-time Target Requirements

Since the interactions between the physical and virtual components of a HIL simulator are bidirectional, it is crucial that the time frames of these components match exactly. Therefore, the virtual components must run in real time, which apply tight requirements on the HIL simulator's microprocessor, operator system, and integration routine. Furthermore, even with fast processors (such as field-programmable gate array or FPGA), running a HIL simulator in real time requires a special kind of operating system that executes integration steps at regular intervals signaled by clock interrupts. Also, the solver used for simulating the virtual components of a HIL setup should ensure the completion of every integration step within the real-time step corresponding to it. This can be difficult if the solver uses variable step-size integration, which explains the prevalence of fixed step-size integration routines in the context of HIL simulation [89].

Fixed step-size integration introduces some challenges for HIL simulators of hybrid discrete/continuous systems, as the transitions between the discrete states of such systems may occur during integration time steps. For instance, a clutch in a car transmission may engage halfway through an integration step. [183] discusses this difficulty and explores some of its possible remedies. Another common problem in HIL simulation is virtual model stiffness, defined as a large disparity between the characteristic speeds of different components of a virtual model. Stiff models can be seen in many disciplines, particularly mechatronics, where mechanical and electrical components typically exhibit markedly disparate response speeds [89]. Proper modeling techniques can often reduce model stiffness by eliminating fast dynamics from a given model in favor of slower dynamics [184]. This may not be feasible if capturing the relatively fast dynamics of a system is a simulation requirement. When the disparity between the fast and slow dynamics in a virtual model cannot be eliminated, it is common to integrate these dynamics separately at different integration rates. Such multirate integration may take place on one processor via multithreading, but is more often achieved using multiple processors. [185, 186] have discussed multirate integration issues with particular focus on different methods that each processor can use to synchronize with other processors.

6.2 Hardware Description

In the HIL simulation, the high-fidelity simulation model of the plant is solved in real-time using a powerful computer. A HIL simulation setup provides a more realistic environment for controller evaluation purposes, as it can take into consideration different aspects of the control loop that are neglected in model-in-the-loop simulations, such as communication issues and controller computational limitations. In this section, the details of the HIL simulation setup will be discussed. The two main components in an HIL setup are: 1) an independent processing unit to run the controller procedure, and 2) a powerful real-time processing unit to run the plant model. For our HIL simulation, the designed controller is programmed into an ECU, and the high-fidelity powertrain model is solved by a real-time target to provide the accurate sampling which the controller requires. The communication channel between the ECU and the plant (real-time target) is the Control Area Network (CAN) bus. The following parts contain details of the hardware used in this setup.

6.2.1 MotoTron

The HIL simulation results are more reliable when the controller prototype is the same as the controller used in the real plant. For EMS application, a MotoTron ECU is used to serve as the powertrain controller. This ECU is from the ECM-5554-112 family of controllers from Woodward that uses an 80MHz Motorola MPC5554 processor. The commercial version of this controller is used in automotive and marine applications. The automotive-based design of this ECU makes it an ideal choice for the HIL simulations. To program the controller code into the ECU, the code needs to be compiled by the MotoHawk Green Hill compiler. Then, the generated code can be programmed into the ECU by the MotoTune software. The controller used in this setup is a version that can be calibrated and provides controller tuning features in real time using MotoTune. This feature is specifically useful in tuning controller parameters without encountering the need to reprogram the controller itself. The easiest way to program the controller is to download the generated code into the controller via CAN bus using the MotoTune software. To this end, a USB-to-CAN

adapter is provided by MotoTune to facilitate the programming procedure. The controller code itself can be easily complied using Woodward's Green Hill compiler, which compiles the required code directly from a Simulink model.

6.2.2 PXI Real-time Target

To satisfy real-time requirements, and achieve enhanced accuracy of the simulations, it is necessary to use a real-time computer to solve the plant model deterministically. For this purpose, a PXI platform from National Instrument (NI) is used as the real-time target. The processing unit of this computer is a PXI-8110, which is powered by a 2.26 GHz quad-core CPU and has 2GB of RAM. This PXI platform runs the LabVIEW real-time operating system, which responds to an interrupt or performs a task before a specified deadline as opposed to non-real-time operating systems where tasks are prioritized based on different criteria such as maintaining the hardware/software functionality. Therefore, by making use of such real-time operating systems, the model can be solved with greater consistency, and the communication delay can be minimized. Our real-time target (NI PXI computer) runs LabVIEW real-time 2011 operating system.

To run a program on this platform, a LabVIEW program must be deployed. LabVIEW is a graphical programming language that facilitates communication with external hardware and expedites the development of multi-threaded applications. LabVIEW programs are made in Virtual Instrument (VI) files. These VI programs are made in the host laptop which runs a Windows version of LabVIEW. The VI programs are then utilized into the real-time target via Ethernet connection. Once the program is successfully deployed, the real-time target begins to run the program, and the user can see the outputs or send commands using the host computer. To use this platform for solving the powertrain high-fidelity model in real time, the model has to be converted into a C-code and then into a Digital Link Library (DLL) in order to be used in the LabVIEW environment. To this end, the MapleSim EMI component block generator toolbox provides seamless solution to convert the high-fidelity simulation model into an appropriate DLL file that can be used in the NI LabVIEW environment.

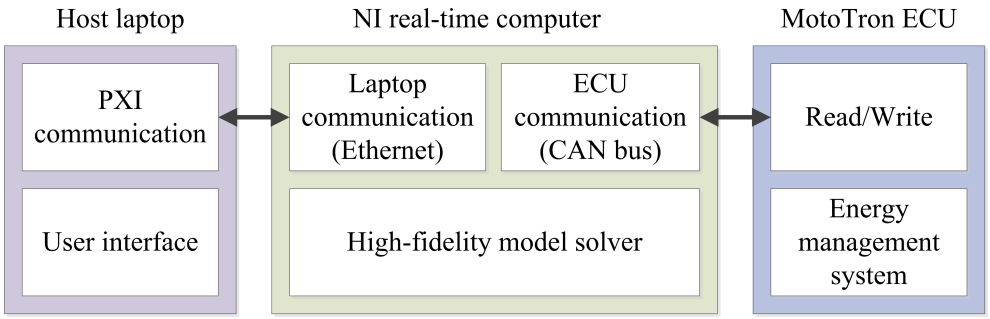

Figure 6.2: Schematic of the HIL setup

Major responsibilities of the real-time target are shown in Figure 6.2. Each core of the real-time target CPU runs a different application. The first core is responsible for running the application to PXI-host communication. This application is solely used to send and receive variables to and from the laptop host via Ethernet connection. The second CPU core runs the CAN communication application. The last core is responsible for solving the high-fidelity vehicle model.

6.2.3 CAN Bus

A HEV has several critical subsystems with individual control modules such as the engine, battery, driveline and brakes. The controllers communicate with each other and with the vehicle system controller on a CAN-based communication network. CAN is a standard message-based protocol, which was initially developed for in-vehicle communications, because of its robustness and ease of operation. The behaviors of these subsystems are strongly influenced by their individual controllers. Not all of these control modules were connected in the HIL setup. Those controllers that were not connected as hardware pieces were simulated as models along with the plant dynamics on the HIL system. So, the communication between controllers and the controller functionalities had to be modeled carefully to ensure a good compromise between functional accuracy and real-time constraints [6]. On a CAN bus, each of the nodes are directly connected to the bus, and there

Table 6.1: CAN message definition for the HIL simulation

CAN message				MotoHawk		LabVIEW	
message name	arbitration ID	message length	variable name	start bit	bit length	start bit	bit length
PXI to ECU	1	7 bytes	E_{ref}	48	8	8	8
			SOC_{ref}	40	8	16	8
			E	24	16	24	16
			SOC	16	8	40	8
			V_{batt}	8	8	48	8
ECU to PXI	2	7 bytes	P_{BAT}	40	16	8	16
			P_{ENG}	24	16	24	16
			P_{BRK}	8	16	40	16

is no central control unit to regulate the communications. Instead, CAN bus is a serial message-based protocol, where each node can send and receive messages when the bus is free. When two nodes start to send messages simultaneously, the message with higher priority prevails, and the lower-priority message waits until the bus is free. The priority of each message is identified by an arbitration ID (Appendix C), where lower IDs have the higher priority.

The EMS requires three readings from the plant: the current battery state of charge, the current demanded tractive energy, and the battery voltage. The three measurements are calculated by the real-time target by solving the high-fidelity simulation model. Also, it takes two readings from SOC_{ref} and E_{ref}. The real-time target then sends these four pieces of information, in a single CAN message to the ECU. The controller processes the information and calculates P_{BAT}, P_{ENG}, and P_{BRK} and send them back to the real-time target in another message.

Table 6.1 shows the variables, and the position of the variable in the CAN messages for ECU-PXI communication.

In the base CAN frame format (CAN 2.0 A protocol), the identifier portion of the message (arbitration ID) contains 11 bits following the start bit. The main data frame can

contain up to 8 bytes (64 bits). Combined with all other regulatory bits, a CAN message is comprised of up to 108 bits. Depending on the bit-rate of the CAN channel, a limited number of messages can be sent on a CAN bus. In this HIL setup, the CAN channels work with a bit rate of 500 kbps (kilo-bits per second); therefore, the maximum capacity of each CAN channel is roughly 4600 messages per second. The communication program on the real-time target runs at every 1ms and sends a message (PXI to ECU) in each run of the loop. The controller program also runs every 5ms and sends one message (ECU to PXI). Thus, 1200 messages are sent in each second, and this load occupies 26% of the CAN channel capacity.

6.3 Controls Implementation Notes

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

To solve this problem, the eMPC energy management was modified. The control action surfaces (See Figure 5.4 and 5.24) versus state variables were approximated with some new look-up tables. Using this technique, we reduced the size of the controller from 2 MB to 143 kB for the larger CRPE-eMPC EMS.

Figure 6.3 shows different parts of the high-fidelity simulation model inside the real-time computer.

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

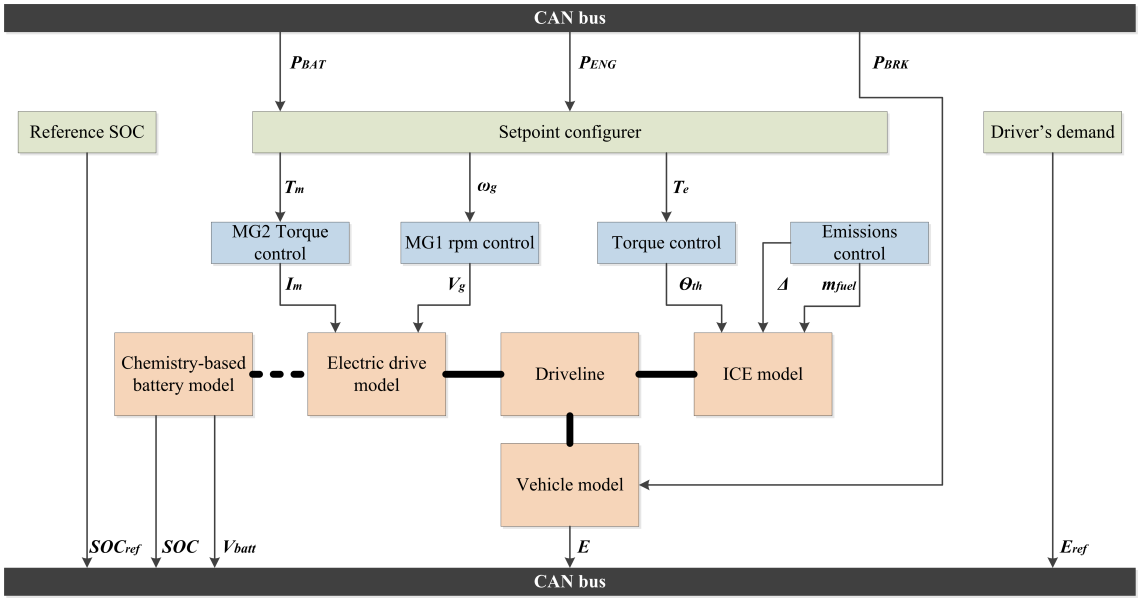

Figure 6.3: High-fidelity model inside the real-time computer

powertrain model, SOC, E, V_{batt}, SOC_{ref}, and E_{ref} are measured at each 1 ms and passed to the ECU via CAN bus.

6.4 eMPC on the Low-fidelity Model

In Figure 6.4-a, the vehicle drivability performance and battery state of charge for CDCS strategy are demonstrated. In Figure 6.4-b, we can see that the driver's demanded power is followed by propulsion power. This shows that the powertrain is able to provide the required propulsion power, so the vehicle velocity can follow the predefined UDDS schedule. Figure 6.4-c shows the index of demanded power as well as SOE index which are determined according to the number of polytopes set. Figure 6.5 shows these results for the blended mode strategy. Note that the engine operation has reduced the battery SOC depletion slope which results in better fuel economy as compared to CDCS strategy.

Table 6.2 shows the HIL fuel economy for applying eMPC EMS to low-fidelity power-

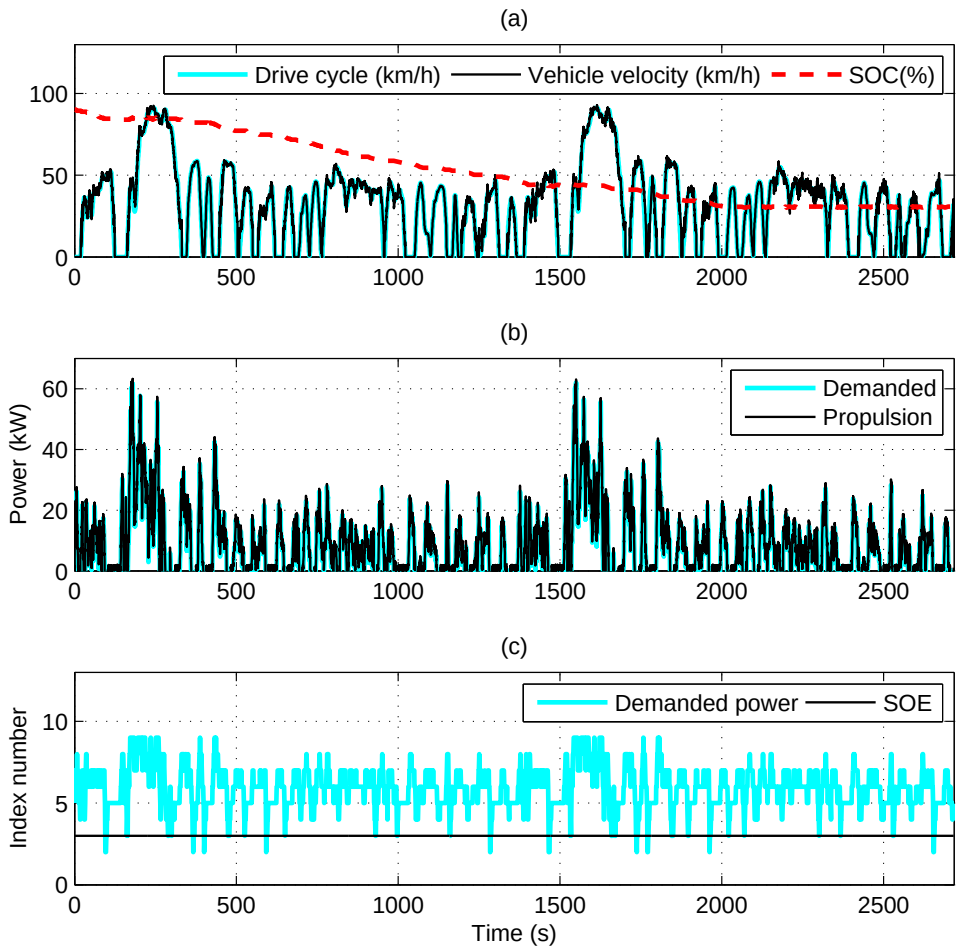

Figure 6.4: Charge depletion/ charge sustenance strategy (a) vehicle velocity and battery depletion profile (b) demanded and propulsion power (c) demanded power and SOE indices

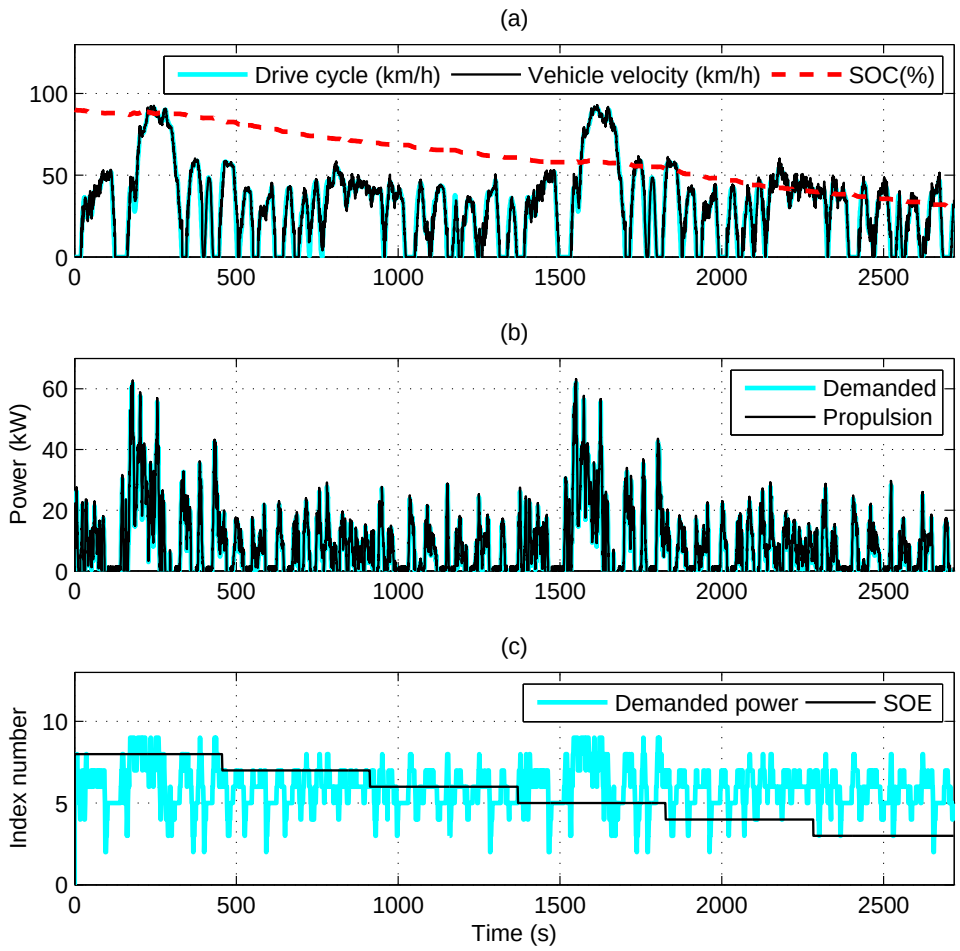

Figure 6.5: Blended mode strategy (a) vehicle velocity and battery depletion profile (b) demanded and propulsion power (c) demanded power and SOE indices

Table 6.2: eMPC MIL and HIL test using low-fidelity powertrain model: Fuel economy for different control strategies

Control Strategy	MIL (MPG)	HIL (MPG)
Charge Depletion/Charge Sustenance	119	116
Linear blended mode	133	127

train model.

Note that if we use the same controller and simulation model for MIL and HIL test, the simulation results should be the same. By comparing the results in this section to what was discussed in section 5.5, some discrepancies are seen in terms of vehicle drivability and fuel economy. The oscillations of the vehicle velocity shown in Figure 6.4 and Figure 6.5 as compared to Figure 5.9 and Figure 5.10, is due to switching between different polytope sets found by considering 9 divisions for E_{ref} and SOE_{ref} along the drive cycle. Fuel economy for CDCS and blended mode strategies in HIL testing are worsened by 2.5% and 4.5% as compared to MIL test. This error is due to replacing the eMPC data base and its search algorithm with the approximated look-up tables. As a measure of drivability performance, eMPC EMS can follow the designated drive cycle with the root mean square error of 0.91 km/h and 1.12 km/h for CDCS and Blended mode strategies, respectively. In brief, the difference between MIL and HIL simulation results is due to the difference between the originally designed and modified CRPE-eMPC EMS.

6.5 CRPE-eMPC on the High-fidelity Model

Figure 6.6-a shows the vehicle drivability performance along 2 UDDS drive cycles for CDCS strategy. The performance of the low-level controls in tracking the CRPE-eMPC EMS setpoints is shown in Figure 6.6-b,c. The emissions control performance is demonstrated in Figure 6.7 and Figure 6.8. PHEV fuel consumption is reduced to 2.06 l/100 km by using

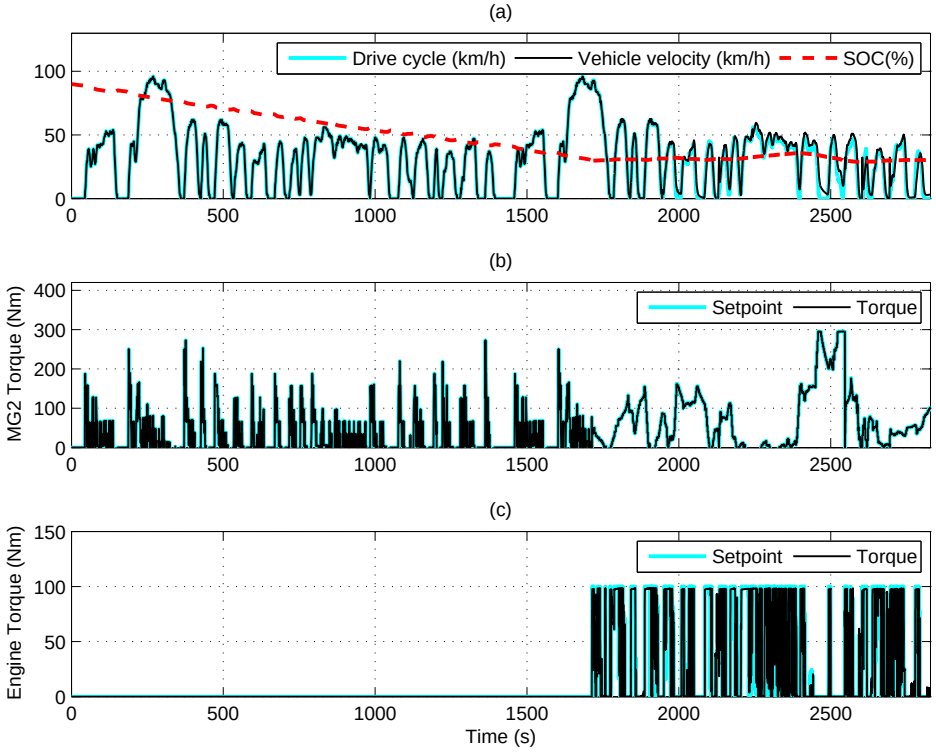

Figure 6.6: HIL simulation, CDCS CRPE-eMPC strategy with emission control: (a) Velocity and Battery SOC (b) MG2 Torque (c) Engine Torque

CRPE-eMPC EMS.

Figure 6.9 to Figure 6.11 show the results for blended mode strategy. Fuel consumption in this simulation is 1.73 l/100km by considering engine emissions control.

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

Presumably, the results of MIL test which was discussed in section 5.9 should be the same as what we get from HIL test. But fuel economy for CDCS and blended mode strategies in HIL testing are worsened by 5% and 3.6% as compared to the MIL test. As mentioned earlier, the current controller has been changed in comparison to the EMS validated in section 5.9, since the CRPE-eMPC controller was modified in order to implement

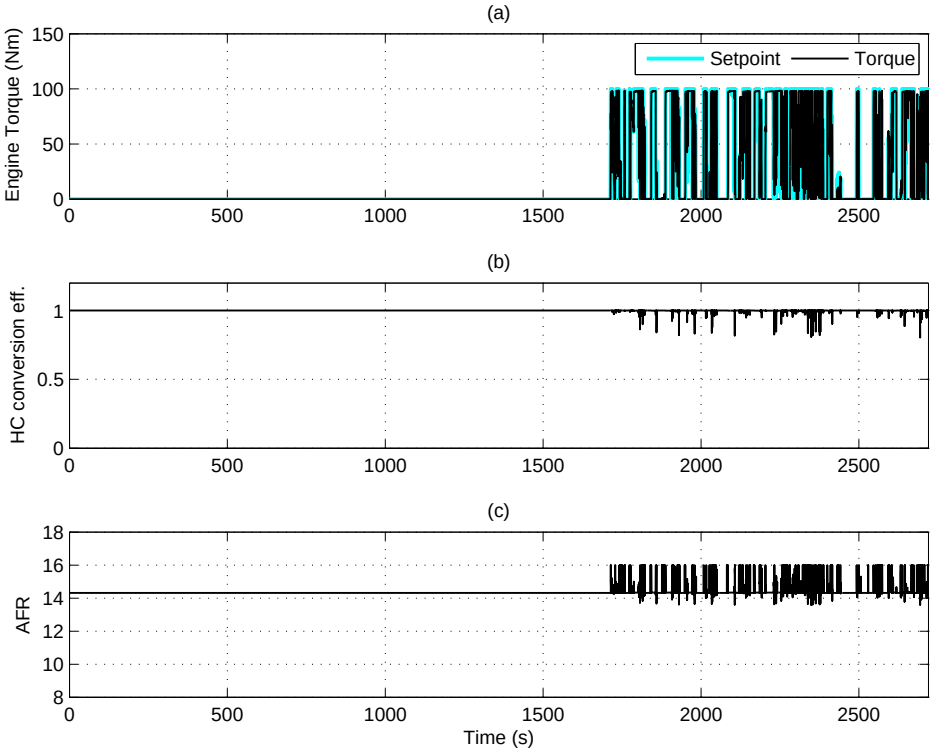

Figure 6.7: HIL simulation, CDCS CRPE-eMPC strategy with emission control: (a) Engine Torque (b) HC conversion efficiency (c) Air-to-fuel ratio (AFR)

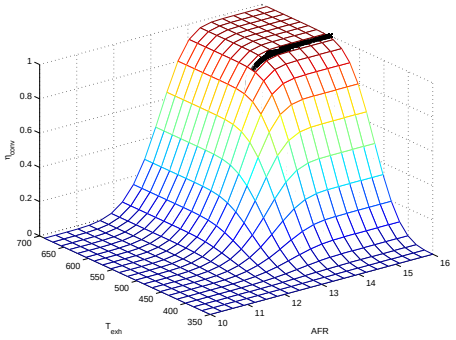

Figure 6.8: HIL simulation, catalyst conversion efficiency for CDCS CRPE-eMPC strategy

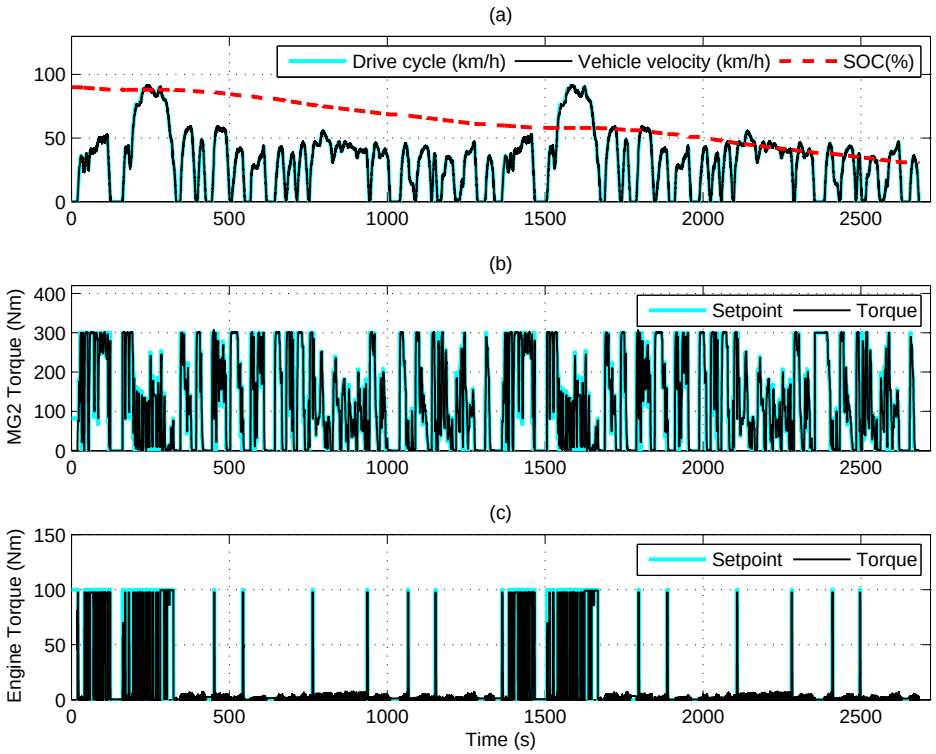

Figure 6.9: HIL simulation, Blended CRPE-eMPC strategy with emission control: (a) Velocity and Battery SOC (b) MG2 Torque (c) Engine Torque

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

Control Strategy	MIL (MPG)	HIL (MPG)
Charge Depletion/Charge Sustenance	119	113
Linear blended mode	140	135

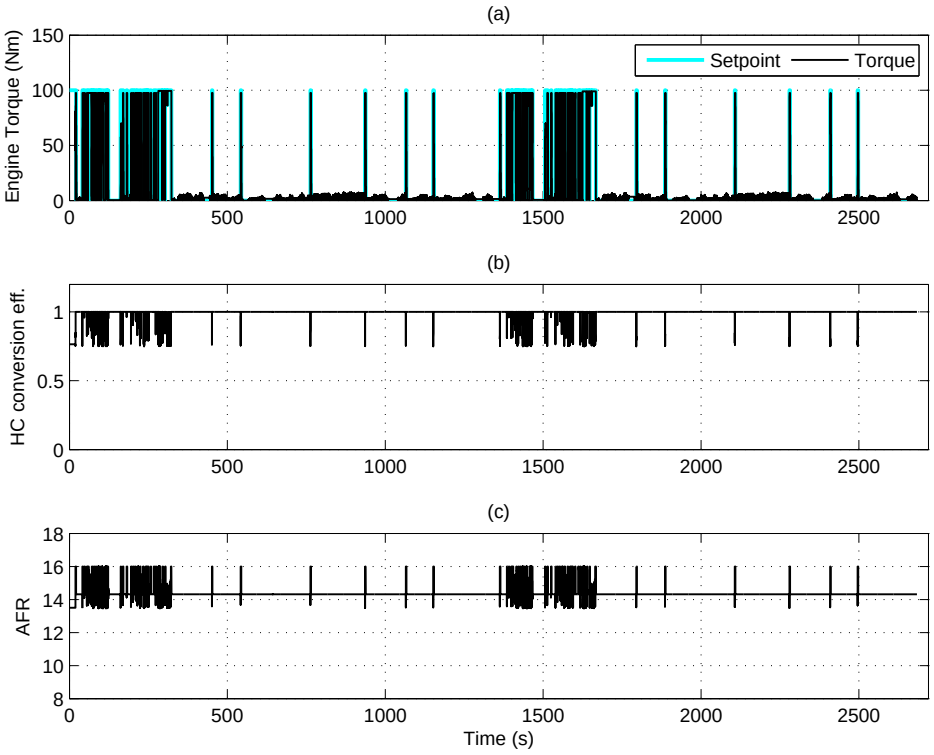

Figure 6.10: HIL simulation, Blended CRPE-eMPC strategy with emission control: (a) Engine Torque (b) HC conversion efficiency (c) Air-to-fuel ratio (AFR)

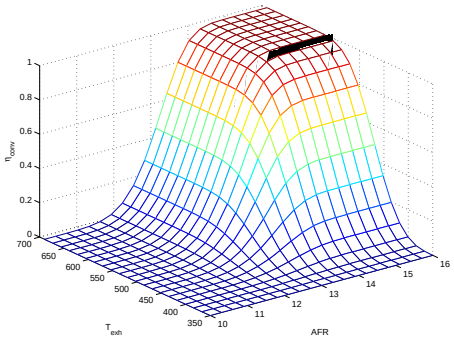

Figure 6.11: HIL simulation, catalyst conversion efficiency for blended CRPE-eMPC strategy

Table 6.4: Fuel economy summary

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

the EMS into MotoTron hardware. Note that if the emissions control is not considered in the control scheme, the fuel economy improvement is expected to reach up to 132 MPG and 156 MPG for CDCS and blended mode strategies, respectively.

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

6.6 Chapter Summary

In this chapter, the designed EMS performance was validated through HIL test. In order to implement the EMS to the control hardware with limited memory size and computational capability, some modifications were applied to the original control scheme. HIL simulation shows that the proposed EMS can be implemented to a commercial control hardware in real time and results in a promising fuel economy improvement up to 16.5% compared to the baseline strategy by controlling the emissions. Note that if the emissions control is not considered in the control scheme, the fuel economy improvement is expected to be even higher.

Chapter 7

Conclusions

7.1 Summary

In this thesis, a near-optimal EMS for a plug-in hybrid electric powertrain was proposed to minimize both fuel consumption and emissions, using a model predictive control (MPC) approach.

At first, a real-time, equation-based, and validated high-fidelity simulation model of a plug-in hybrid electric powertrain was developed in the MapleSim software in order to be used in controls performance evaluation procedure such as model-in-the-loop (MIL) and hardware-in-the-loop (HIL) simulations. The chemistry-based model of the battery in the powertrain model, leads to more realistic estimation of the PHEV fuel economy and range. The parameters of PHEV powertrain model were adjusted based on the experimental database of the Autonomie software, which is widely used for energy management design in industry. The symbolic programming capability of Maple allows reducing the number of equations involved in the powertrain model significantly, and makes it run in real time, which is essential for performing HIL tests.

For implementing the EMS to the high-fidelity simulation model, another level of controls were designed for the engine and electric drive, using sliding mode control and field-

oriented control approaches, respectively. One of the features of the engine control system is improving hydrocarbon conversion efficiency of the catalytic converter.

Due to some real-time implementation problems of MPC despite its near-optimal performance in improving fuel economy, another PHEV EMS was designed using explicit model predictive control (eMPC), which solves the optimization problem off-line to create some look-up tables. Therefore, the problem is reduced from solving a quadratic programming problem at each control sampling time, to searching in look-up tables while implementing the control algorithm. As a result, eMPC can guarantee real-time implementation for a fairly small control-oriented model. In order to keep the size of mentioned look-up tables small enough to be implemented to a commercial control hardware with limited amount of flash memory, a simple and innovative control-oriented model was proposed. According to MIL simulation results, the designed eMPC energy management can be implemented to the high-fidelity simulation model three times faster that its implicit MPC counterpart.

For further improvement of energy management strategy performance, a control-relevant parameter estimation (CRPE) approach was used to make the control-oriented model more accurate. The resultant CRPE-control-oriented model is accurate within a specific frequency range that is excited by the EMS. Based on the newly-developed control-oriented model, CRPE-eMPC EMS was designed. The MIL simulation revealed that the CRPE-eMPC EMS could reduce the fuel consumption up to 6.5% as compared to the eMPC EMS.

Finally, the CRPE-eMPC energy management strategy was implemented on MotoTron hardware with limited computation and memory capabilities. The aforementioned high-fidelity simulation model was used as the virtual simulation model inside the real-time target which was connected to the hardware via CAN bus. The HIL simulation showed that the proposed EMS could be implemented in real time and result in a promising fuel economy improvement up to 16.5% as compared to the baseline strategy by controlling the emissions. Note that if the emissions control is not considered in the control scheme, the fuel economy improvement is even higher.

The proposed research contributions can be summarized as follows:

- Developed real-time, equation-based, and cross-validated high-fidelity simulation model of a PHEV powertrain.
- Model predictive control design and evaluation for a PHEV EMS.
- Developed and validated near-optimal and real-time implementable PHEV EMS using explicit model predictive control (eMPC) approach with simple and innovative control-oriented model for simpler stability analysis.
- Developed control-relevant parameter estimated (CRPE) control-oriented model to improve performance of eMPC EMS while maintaining its real-time capabilities.
- Implemented the CRPE-eMPC energy management strategy on a commercial control hardware with limited computational and memory capabilities.

7.2 Recommendations for Future Research

Although the performance of the proposed EMS for improving fuel economy was shown through MIL and HIL simulations, there is still room for improvement in both controls design and validation stages. The recommended future research for each part is separately mentioned below:

7.2.1 Controls Design

Virtual Simulation Model Improvement

The proposed MapleSim model of PHEV in chapter 3, should be run in a real-time target with multi-threaded CPU in order to include the power electronics high-fidelity simulation model. Furthermore, the model parameters can be validated through the data from a prototype on a rig or a test bench of powertrain real components for even more realistic estimation of powertrain behavior.

Smart PHEV

As mentioned, PHEV performance is closely related to the battery depletion profile along the driving schedule. More information from the trip can significantly improve EMS performance. This improvement can be made in two ways:

- Short horizon vehicle velocity prediction in order to take advantage of MPC predictive feature by using global positioning system (GPS), geographic information system (GIS), and intelligent traffic systems (ITS).

- Optimized SOC depletion profile (as shown in chapter 4).

By considering the above items, one can design a smart EMS which pushes the PHEV performance to its ultimate limit.

Fast MPC Approach

Basically, two approaches to fast quadratic programming (QP) solution in MPC can be distinguished: first, the explicit, or off-line QP solution, as used in this thesis, which is limited to models with small state dimensions and few process inputs. Second, the on-line QP solution is the classical way to treat the sequence of QPs in MPC for varying initial process values.

As a result, for the designed MPC EMS in chapter 4, faster QP solvers (e.g. active set method) can be used. By using this fast MPC approach there is a possibility of improving MPC energy management real-time capabilities. Then, fast MPC performance can be compared to eMPC energy management as proposed in chapter 5.

Sensitivity Analysis

Sensitivity analysis can be done on the parameters of EMS such as control and prediction horizon length as well as cost function weighting parameters by using either classical MPC

or the explicit version. This analysis is helpful during the energy management calibration stage for expediting the procedure and taking advantage of a model-based control approach.

Control-oriented Model Improvement

Control-relevant parameter estimation (CRPE) for control-oriented model can be done for different driving schedules in order to find a more precise active frequency range for the battery. This leads to a more accurate control-oriented model and consequently guarantees EMS's best performance along any driving scenario. Moreover, CRPE can be performed for control-oriented models with higher order to obtain the most accurate control-oriented model while maintaining real-time capability of the EMS.

In this way, one can find a relation between control-oriented model fidelity and real-time capability of eMPC.

7.2.2 Controls Validation

Energy Management System Calibration

ECU calibration is an iterative process of measurement and calibration at runtime to optimally tune the parameters of the ECU algorithms. The parameters of the proposed EMS should be tuned during calibration procedure, for the best possible performance on a PHEV prototype.

Obviously, the model-based control approach can expedite this stage and make tuning of the parameters and weights more systematic.

Energy Management System on a PHEV Test Bench

In order to find real-world performance of the EMS, it should be implemented to the real components of a PHEV on a test bench.

References

[1] M. Ehsani, Y. Gao, and A. Emadi. *Modern Electric, Hybrid Electic and Fuel Cell Vehicles: Fundamentals, Theory, and Design*. Taylor and Francis Group LLC, 2nd edition, 2010.

[2] T. Markel. Plug-In HEV vehicle design options and expectations. *ZEV Technology Symposium, California Air Resources Board, Sacramento, CA, USA*, pages 111–122, 2006.

[3] Electric Power Research Institute (EPRI) Report. Technology primer: the plug-in hybrid electric vehicle. pages 1–2, 2007.

[4] K. Parks, P. Denholm, and T. Markel. Costs and emissions associated with plug-in hybrid electric vehicle charging in the xcel energy colorado service territory. *National Renewable Energy Laboratory Report, NREL TP-640-41410*, page 29, 2007.

[5] S. Di Cairano, W. Liang, I.V. Kolmanovsky, M.L. Kuang, and A.M. Phillips. Engine power smoothing energy management strategy for a series hybrid electric vehicle. *American Control Conference (ACC)*, pages 2101–2106, 2011.

[6] D. Ramaswamy, R. McGee, S. Sivashankar, A. Deshpande, and et al. A case study in hardware-in-the-loop testing: Development of an ecu for a hybrid electric vehicle. *SAE Technical Paper 2004-01-0303*, 2004.

[7] L. Del Re, F. Allgower, L. Glielmo, C. Guardiola, and I. Kolmanovsky. *Automotive Model Predictive Control: Models, Methods and Applications*. Springer, 2010.

[8] J.B. Rawlings and D.Q. Mayne. *Model predictive control: theory and design*. Nob Hill Publishing, LLC, Madison, Wisconsin, 2009.

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

[10] H.A. Borhan, A. Vahidi, A.M. Phillips, M.L. Kuang, and I.V. Kolmanovsky. Predictive energy management of a power-split hybrid electric vehicle. *American Control Conference*, 2009.

[11] Z. Wang. *Model Predictive Control for Hybrid Electric Vehicle*. PhD thesis, The Chinese University of Hong Kong, 2008.

[12] D.Q. Mayne. *Constrained optimal control*. European Control Conference, Plenary Lecture, 2001.

[13] A. Bemporad, W. Heemels, and B. De Schutter. On hybrid systems and closed-loop mpc systems. *IEEE Transactions on Automatic Control*, 47(5):863–869, 2002.

[14] N. Kim, D. Daeheung Lee, S.W. Cha, and H. Peng. Optimal control of a plug-in hybrid electric vehicle (phev) based on driving patterns. *EVS24 International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium, Stavanger, Norway*, 2009.

[15] Argonne National Laboratory. Autonomie. *Available from http://www.autonomie.net/index.html*, 2011.

[16] X. Li and S.S. Williamson. Efficiency and suitability analyses of varied drive train architectures for plug-in hybrid electric vehicle (phev) applications. *Vehicle Power and Propulsion Conference, IEEE VPPC*, pages 1–6, 2008.

[17] S.G. Wirasingha and A. Emadi. Classification and review of control strategies for plug-in hybrid electric vehicles. *IEEE Transactions on Vehicular Technology*, 60(1):111–122, 2011.

[18] S.J. Moura, J.L. Stein, and H.K. Fathy. Battery health-consious power management for plug-in hybrid vehicles via stochastic control. *IEEE Transactions on Control Systems Technology*, 21(3):679 – 694, 2010.

[19] M. Ceraolo, A. di Donato, and G. Franceschi. A general approach to energy optimization of hybrid electric vehicles. *IEEE Transaction on Vehicular Technology*, 57(3):1433 – 1441, 2008.

[20] L. Bergh, B. Simpkin, M. Abele, G. Heuer, A. Ferre, S. Vallejos, K. Nenniger, H. Berger, and M. Midl. Energy efficient vehicles for road transport ee-vert. 2009.

[21] B.K. Powell, K.E. Bailey, and S.R. Cikanek. Dynamic modeling and control of hybrid electric vehicle powertrain systems. *IEEE Control Syst. Mag.*, 18(5):17–33, 1998.

[22] A. Rousseau, S. Pagerit, and D. Gao. Plug-in hybrid electric vehicle control strategy parameter optimization. *Electric Veh. Symp.-23, Anaheim, CA*, 2007.

[23] A. Vahidi, A. Stefanopoulou, and H. Peng. Current management in a hybrid fuel cell power system: A model-predictive control approach. *IEEE Transactions on Control Systems Technology*, 14(6):1047–1057, 2006.

[24] J. Liu and H. Peng. Modeling and control of a power-split hybrid vehicle. *IEEE Transactions on Control Systems Technology*, 16(6):1242–1251, 2008.

[25] D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2 edition, 1995.

[26] S. J. Moura, H.K. Fathy, D.S. Callaway, and J.L Stein. A stochastic optimal control approach for powermanagement in plug-in hybrid electric vehicles. *IEEE Transactions on Control Systems Technology*, 19(3):545–555, 2011.

[27] V. Freyermuth, E. Fallas, and A. Rousseau. Comparison of production powertrain configuration options for plug-in hevs from fuel economy perspective. *SAE Technical Paper 2008-01-0461*, 2008.

[28] A. Brahma, Y. Guezennec, and G. Rizzoni. Optimal energy management in series hybrid electric vehicles. *American Control Conference*, 2000.

[29] C.C. Lin, H. Peng, J.W. Grizzle, and J.M. Kang. Power management strategy for a parallel hybrid electric truck. *IEEE Transactions on Control Systems Technology*, 11(6):839849, 2003.

[30] Q. Gong, Y. Li, and Z.R. Peng. Trip based optimal power management of plug-in hybrid electric vehicles using gas-kinetic traffic flow model. *American Control Conference*, 2008.

[31] L. Wang. *Model Predictive Control System Design and Implementation Using MATLAB*. Springer, 2009.

[32] J. V. G. Ortiz P. Ortner, P. Langthaler and L. del Re. Mpc for a diesel engine air path using an explicit approach for constraint systems. *Proc. IEEE International Conference on Control Applications,Munich, Germany*, pages 2760–2765, 2006.

[33] H. Hur, T. Nagata, and M. Tomizuka. Model-based optimal gear shift pattern scheduling and smooth gear shifting control. *Steuerung und Regelung von Fahrzeugen*

und Motoren - AUTOREG 2006, VDIBerichte Nr. 1931. VDI Wissensforum, pages 303–312, 2006.

[34] L. Johannesson, B. Egardt, and M. Asbogard. Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming. *IEEE Transaction on Intelligent Transportition Systems*, 8(1):71–83, 2007.

[35] F. Latteman, K. Neiss, S. Terwen, and T. Connolly. The predictive cruise control - a system to reduce fuel consumption of heavy duty trucks. *SAE Technical Paper 2004-01-2616*, 2004.

[36] B. Saerens, M. Diehl, J. Swevers, and E. Van den Bulck. Model predictive control of automotive powertrains - first experimental results. *47th IEEE Conference on Decision and Control (CDC)*, pages 5692–5697, 2008.

[37] C. Patil, B. Morris, C. Fortune, B. Surampudi, J. Redfield, and H. Gruenewald. Model-Based Approach to Estimate Fuel Savings from Series Hydraulic Hybrid Vehicle : Model Development and Validation. *SAE 2011-01-2274*, 2011.

[38] F.U. Syed, M. Kuang, L. Ming, J. Czubay, and H. Ying. Derivation and Experimental Validation of a Power-Split Hybrid Electric Vehicle Model. *IEEE Transactions on Vehicular Technology*, 55(6):1731–1747, 2006.

[39] Yongsheng He and Chan chiao Lin. Development and Validation of a Mean Value Engine Model for Integrated Engine and Control System Simulation. *SAE 2007-01-1304*, pages 776–790, 2007.

[40] T. Markel, A. Brooker, T. Hendricks, T. Johnson, K. Kelly, B. Kramer, M. OKeefe, S. Sprik, and K. Wipke. ADVISOR: A systems analysis tool for advanced vehicle modeling. *Journal of Power Sources*, 110(2):255–266, 2002.

[41] C. Lin. Integrated, feed-forward hybrid electric vehicle simulation in SIMULINK and its use for power management studies. *SAE 2001-01-1334*, 2001.

[42] A. Rousseau, P.Sharer, and M. Pasquier. Validation process of a HEV system analysis model: PSAT. *SAE 2001-01-0953*, 2001.

[43] N. Kim, A. Rousseau, and E. Rask. Autonomie Model Validation with Test Data for 2010 Toyota Prius. *SAE 2012-01-1040*, 2012.

[44] H.Ferreaua, P. Ortnerb, Peter Langthalerb, L. Del Reb, and M. Diehl. Predictive control of a real-world diesel engine using an extended online active set strategy. *Annual Reviews in Control*, 31(2):293–301, 2007.

[45] A. Bemporad. Model-based predictive control design: New trends and tools. *45th IEEE Conference on Decision and Control (CDC)*, page 66786683, 2006.

[46] A. Alessio and A. Bemporad. A survey on explicit model predictive control. *Lecture Notes in Control and Information Sciences*, 384:345–369, 2009.

[47] S. Di Cairano, D. Yanakiev, A. Bemporad, I. Kolmanovsky, and D. Hrovat. An mpc design ow for automotive control and applications to idle speed regulation. *47th IEEE Conference on Decision and Control (CDC)*, pages 5686–5691, 2008.

[48] S. Di Cairano and H. Tseng. Driver-assist steering by active front steering and differential braking: Design, implementation and experimental evaluation of a switched model predictive control approach. *49th IEEE Conference on Decision and Control (CDC)*, pages 2886–2891, 2010.

[49] P. Falcone, F. Borrelli, J. Asgari, H. Tseng, and D. Hrovat. Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on Control Systems Technology*, 15(3):566–580, 2007.

[50] C. Garcia, D. Prett, and M. Morari. Model predictive control: theory and practicea survey. *Automatica*, 25(3):335–348, 1989.

[51] S. Di Cairano, A. Bemporad, I. V. Kolmanovsky, and D. Hrovat. Model predictive control of magnetically actuated mass spring dampers for automotive applications. *International Journal of Control*, 80(11):1701–1716, 2000.

[52] P. Ortner and L. Del Re. Predictive control of a diesel engine air path. *IEEE Transactions on Control Systems Technology*, 15(3):449–456, 2007.

[53] G.J.L. Nausa, J. Ploegb, M.J.G. Van de Molengrafta, W.P.M.H. Heemelsa, and M. Steinbuch. Design and implementation of parameterized adaptive cruise control: An explicit model predictive control approach. *Control Engineering Practice*, 18(8):882–892, 2000.

[54] F. Borrelli, A. Bemporad, M. Fodor, and D. Hrovat. An mpc/hybrid system approach to traction control. *IEEE Transactions on Control System Technology*, 14(3):541–552, 2006.

[55] N. Giorgetti, A. Bemporad, H. E. Tseng, and D. Hrovat. Hybrid model predictive control application towards optimal semiactive suspension. *Proc. of IEEE Int. Symp. on Industrial Electronics, Dubrovnik, Croatia*, pages 391–398, 2005.

[56] N. Giorgetti, G. Ripaccioli, A. Bemporad, I. V. Kolmanovsky, and D. Hrovat. Hybrid model predictive control of direct injection stratified charge engines. *IEEE/ASME Transactions on Mechatronics*, 11(5):499–506, 2006.

[57] A. Lagerberg and B. Egardt. Model predictive control of automotive powertrains with backlash. *Proc. of the 16th IFAC World Congress, Prague, Czech Republic*, 2005.

[58] S. Richter, S. Mariethoz, and M. Morari. Highspeed on-line mpc based on a fast gradient method applied to power converter control. *American Control Conference (ACC)*, pages 4737–4743, 2010.

[59] S. Kouro, P. Cortes, R. Vargas, U. Ammann, and J. Rodriguez. Model predictive control - a simple and powerful method to control power converters. *IEEE Transactions on Industrial Electronics*, 56(6):1826–1838, 2009.

[60] G. Stewart and F. Borrelli. A model predictive control framework for industrial turbodiesel engine control. *48th IEEE Conference on Decision and Control (CDC)*, pages 5704–5711, 2008.

[61] R. Schallock, K. Muske, and J. P. Jones. Model predictive functional control for an automotive three-way catalyst. *SAE Int. J. Fuels Lubricants*, 2(1):242–249, 2009.

[62] S. Trimboli, S. Di Cairano, A. Bemporad, and I. Kolmanovsky. Model predictive control for automotive time-delay processes: An application to air-to-fuel ratio. *Proc. 8th IFAC Workshop Time-Delay Syst.*, pages 1–6, 2009.

[63] R. Amari, M. Alamir, and P. Tona. Unified mpc strategy for idle-speed control, vehicle start-up and gearing applied to an automated manual transmission. *Proc. 17th IFAC World Congr.*, pages 7079–7085, 2008.

[64] A. Grancharova and T. Johansen. Explicit model predictive control of an electropneumtaic clutch actuator using on/off valves and pulsewidth modulation. *Proc. Eur. Control Conf.*, pages 1–6, 2009.

[65] G. Ripaccioli, A. Bemporad, F. Assadian, C. Dextreit, S. Di Cairano, and I. Kolmanovsky. Hybrid modeling, identification, and predictive control: An application

to hybrid electric vehicle energy management. *Hybrid Systems: Computation and Control, ser. Lec. Not. in Computer Science. Springer*, 5469:321–335, 2009.

[66] Seung-Hi Lee, Young Ok Lee, Youngseop Son, and Chung Choo Chung. Proximate model predictive control strategy for autonomous vehicle lateral control. *11th International Conference on Control, Automation and Systems (ICCAS)*, pages 590–595, 2011.

[67] J.H.Lee. Model predictive control: Review of the three decades of development. *International Journal of Control, Automation, and Systems*, 9(3):415–424, 2011.

[68] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[69] F. Borrelli, A. Bemporad, M. Fodor, and D. Hrovat. A hybrid approach to traction control. *Di Benedetto, M.D., Sangiovanni-Vincentelli, A.L. (eds.) HSCC 2001. LNCS 2034*, 2034:162–174, 2001.

[70] A. Widd, L. Hsien-Hsin, J.C. Gerdesand P. Tunestal, and R. Johansson. Highspeed on-line mpc based on a fast gradient method applied to power converter control. *American Control Conference (ACC)*, pages 420–425, 2011.

[71] H. Andersen and M. Kummel. Evaluating estimation of gain directionality parts 1: Methodology and 2: A case study of binary distillation. *J. Process Control*, 2(2):59–86, 1992.

[72] E. Jacobsen and S. Skogestad. Inconsistencies in dynamic models for ill-conditioned plants application to low-order models of distillation columns. *Ind. Eng. Chem. Res.*, 33(3):631–640, 1994.

[73] C. Koung and J. MacGregor. Design of identification experiments for robust control: A geometric approach for bivariate processes. *Ind. Eng. Chem. Res.*, 32(8):1658–1666, 1993.

[74] Hyunjin Lee and D.E. Rivera. An integrated input signal design and control-relevant parameter estimation approach for highly interactive multivariable systems. *American Control Conference (ACC)*, 2006.

[75] D.E. Rivera and M. Morari. Control-relevant model reduction problems for siso h_2, h_∞, and μ-controller synthesis. *International Journal of Control*, 46(2):505–527, 1987.

[76] D.E. Rivera and M. Morari. Plant and controller reduction problems for closed-loop performance. *27th IEEE Conference on Decision and Control (CDC)*, pages 1143–1148, 1988.

[77] D.E. Rivera, C. Webb, and M. Morari. A control-relevant identification methodology. *Annual AIChE Meeting*, 1987.

[78] R.J.P. Schrama. *Approximate identi9cation and control design with application to a mechanical system*. Phd dissertation, Delft University of Technology, 1992.

[79] Z. Zang, R. Bitmead, and M. Gevers. h_2 iterative model refinement and control robustness enhancement. *30th IEEE Conference on Decision and Control (CDC)*, pages 279 – 284, 1991.

[80] R.G. Hakvoort, R.J.P. Schrama, and P.M.J. Van den Hof. Approximate identification with closed-loop performance criterion and application to lqg feedback design. *Automatica*, 30(4):679–690, 1994.

[81] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin. Iterative feedback tuningtheory and applications. *IEEE Control Systems Magazine*, 18(4):26–41, 1998.

[82] R.A. de Callafon, P.M.J. Van den Hof, and M.Steinbuch. Control relevant identification of a compact disc pick-up mechanism. *32nd IEEE Conference on Decision and Control (CDC)*, pages 2050–2055, 1993.

[83] A.G. Partanen and R.R. Bitmead. The application of an iterative identification and controller design to a sugar cane crushing mill. *Automatica*, 31(11):1547–1563, 1995.

[84] Z. Zang, R. Bitmead, and M. Gevers. Iterative weighted least-squares identification and weighted lqg control design. *Automatica*, 31(11):15771594, 1995.

[85] P. Michelberger, J. Bokor, L. Palkovics, E. Nandori, and P. Gaspar. Iterative identification and control design for uncertain parameter suspension system. *IFAC Transportation Systems. Preprints of the 8th IFAC/IFIP/IFORS Symposium*, 2(1):464–469, 1997.

[86] K.S. Jun, D.E. Rivera, E. Elisante, and V.E. Sater. A computer-aided design tool for robustness analysis and control-relevant identification of horizon predictive control with application to a binary distillation column. *Journal of Process Control*, 6(2-3):177–186, 1996.

[87] P. Verboven, P. Guillaume, and B. Cauberghe. Multivariable frequencyresponse curve fitting with application to modal parameter estimation. *Automatica*, 41(10):1773–1782, 2005.

[88] Sung Chul Oh. Evaluation of motor characteristics for hybrid electric vehicles using the hardware-in-the-loop concept. *IEEE Transactions on Vehicular Technology*, 54(3):817–824, 2005.

[89] H.K. Fathy, Z.S. Filipi, J. Hagena, and J.L. Stein. Review of hardware-in-the-loop simulation and its prospects in the automotive area. *Proc. SPIE 6228, Modeling and Simulation for Military Applications*, 2006.

[90] J.R. Wagner and J.S. Furry. A real-time simulation environment for the verification of automotive electronic controller software. *International Journal of Vehicle Design*, 13(4):365–377, 1993.

[91] A. Kimura and I. Maeda. Development of engine control system using real time simulator. *Proceedings of the 1996 IEEE international symposium on computer-aided control system design*, 1996.

[92] W. Lee, M. Yoon, and M. Sunwoo. A cost- and time-effective hardware-in-the-loop simulation platform for automotive engine control systems. *Proceedings of the Institute of Mechanical Eng., Part D: Journal of Automobile Engineering*, 217(1):41–52, 2003.

[93] W. Lee, S. Park, and M. Sunwoo. Towards a seamless development process for automotive engine-control system. *Control Engineering Practice*, 12(1):977–986, 2004.

[94] Q. Song and K.M. Grigoriadis. Diesel engine speed regulation using linear parameter varying control. *American Control Conference (ACC)*, pages 779–784, 2003.

[95] S. Raman, N. Sivashankar, W. Milam, W. Stuart, and S. Nabi. Design and implementation of hil simulators for powertrain control system software development. *American Control Conference (ACC)*, pages 709–713, 1999.

[96] R.M. Schupbach and J.C. Balda. A versatile laboratory test bench for developing powertrains of electric vehicles. *Proceedings of the Vehicular Technology Conference*, 56:1666–1670, 2002.

[97] J.A. Ferreira, F.G. Almeida, M.R. Quintas, and E.J.P De Oliveira. Hybrid models for hardware-in-the-loop simulation of hydraulic systems part 1: Theory. *Proceedings*

of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering, 218(1):465–473, 2004.

[98] J.A. Ferreira, F.G. Almeida, M.R. Quintas, and E.J.P De Oliveira. Hybrid models for hardware-in-the-loop simulation of hydraulic systems part 2: Experiments. *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, 218(1):475–486, 2004.

[99] R. Trigui, B. Jeanneret, B. Malaquin, F. Badin, and C. Plasse. Hardware in the loop simulation of a diesel parallel mild-hybrid electric vehicle. *IEEE Vehicle Power and Propulsion Conference (VPPC)*, pages 448–455, 2007.

[100] D. Winkler and C. Ghmann. Hardware-in-the-loop simulation of a hybrid electric vehicle using modelica/dymola. *Proceedings of the 22nd International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium (EVS-22), Yokohama, Japan*, 2006.

[101] Hu Hao, Xu Guoqing, and Zhu Yang. Hardware-in-the-loop simulation of electric vehicle powertrain system. *Asia-Pacific IEEE Power and Energy Engineering Conference (APPEEC)*, 2009.

[102] A. Hentunen, J. Suomela, A. Leivo, M. Liukkonen, and P. Sainio. Hardware-in-the-loop verification environment for heavy-duty hybrid electric vehicles. *IEEE Vehicle Power and Propulsion Conference (VPPC)*, pages 1 – 6, 2010.

[103] J. Newman and W. Tiedemann. Porous-electrode theory with battery applications. *AIChE Journal*, 21(1):25–41, 2004.

[104] M. Doyle, T.F. Fuller, and J. Newman. Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. *Journal of the Electrochemical Society*, 140(6):1526–1533, 1993.

[105] T. Dao, C. P. Vyasarayani, and J. McPhee. Simplification and order reduction of lithium-ion battery model based on porous-electrode theory. *Journal of Power Sources*, 198(0):329–337, 2012.

[106] C. P. Vyasarayani, T. Uchida, A. Carvalho, and J. McPhee. Parameter identification in dynamic systems using the homotopy optimization approach. *Multibody System Dynamics*, 26(4):411–424, 2011.

[107] Y. Mizuno, R. Ibaraki, K. Kondo, K. Odaka, H. Watanabe, T. Mizutani, K. Kaneshige, and D. Kitada. Development of New Hybrid Transmission for Compact-Class Vehicles. *SAE 2009-01-0726*, 2009.

[108] T.A. Burress, S.L. Campbell, C.L. Coomer, C.W. Ayers, A.A. Wereszczak, J.P. Cunningham, L.D. Marlino, L.E. Seiber, and Hua-Tay Lin. Evaluation of the 2010 toyota prius hybrid synergy drive system. *ORNL/TM-2010/253*, 2011.

[109] O. Grondin, R. Stobart, H. Chafouk, and J. Maquet. Modeling the Compression Ignition Engine for Control: Review and Future Trends. *SAE 2004-01-0423*, 2004.

[110] P.J.M. Schulten and D. Stapersma. Mean Value Modelling of the Gas Exchange of a 4-stroke Diesel Engine for Use in Powertrain Applications. *SAE 2003-01-0219*, 2003.

[111] M. Kao and J.J. Moskwa. Turbocharged Diesel Engine Modeling for Nonlinear Engine Control and State Estimation. *ASME Journal of Dynamic Systems, Measurement, and Control*, 117(4):20–30, 1995.

[112] I. Kolmanovsky, P. Moraal, M. van Nieuwstadt, and A.G. Stefanopoulou. Issues in Modelling Control of Intake Flow in Variable Geometry Turbocharged Engines. *Proceedings of 18th IFIP Conference on System Modeling and Optimization*, 1997.

[113] M. Saeedi. *A Mean Value Internal Combustion Engine Model in MapleSim*. Msc dissertation, University of Waterloo, Waterloo (ON), 2010.

[114] S. Bogosyan, M. Gokasan, and D.J. Goering. A Novel Model Validation and Estimation Approach for Hybrid Serial Electric Vehicles. *IEEE Transactions on Vehicular Technology*, 56(4):1485–1497, 2007.

[115] A. Taghavipour, N. L. Azad, and J. McPhee. An optimal power management strategy for power split plug-in hybrid electric vehicles. *International Journal of Vehicle Design*, 60(3-4):286 – 304, 2012.

[116] C.M. Ferris, O.L. Mangasarian, and S.J. Wright. *Linear programming with MATLAB*. the Society for Industrial and Applied Mathematics and the Mathematical Programming Society, 2007.

[117] S.S. Keerthi and E.G. Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: stability and moving-horizon approximations. *Journal of Optimization Theory and Applications*, 57(1):265–293, 1988.

[118] H. Chen and F. Allgower. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 14(1):1205–1217, 1998.

[119] R.R. Bitmead, M. Gevers, and V. Wertz. *Adaptive Optimal Control: The Thinking Man's GPC*. Prentice Hall, 1990.

[120] D.W. Clarke. *Advances in Model-Based Predictive Control*. Oxford University Press, 1994.

[121] E. Mosca. *Optimal, Predictive, and Adaptive Control*. Prentice Hall, 1995.

[122] E. Polak and T. H. Yang. Moving horizon control of linear systems with input saturation and plant uncertainty, parts 1 and 2. *International Journal of Control*, 58(3):613–663, 1993.

[123] S. L. de Oliveira Kothare and M. Morari. Contractive model predictive control for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 45(1):1053–1071, 2000.

[124] S. Keerthi and E. Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: stability and moving-horizon approximations. *Journal of Optimization Theory and Applications*, 57(1):265–293, 1988.

[125] D. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(1):814–824, 1990.

[126] A. Zheng and M. Morari. Stability of model predictive control with soft constraints. *IEEE Transactions on Automatic Control*, 40(10):1818–1823, 1995.

[127] J. A. Primbs and V. Netistic. A framework for robustness analysis of constrained finite receding horizon control. *American Control Conference (ACC)*, pages 2718–2722, 1998.

[128] J. H. Lee and Z. Yu. Worst-case formulation of model predictive control for systems with bounded parameters. *Automatica*, 33(5):763–781, 1997.

[129] V. Balakrishnan M. V. Kothare and M. Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, 1996.

[130] K. Muta, M. Yamazaki, and J. Tokieda. Development of new-generation hybrid system ths iidrastic improvement of power performance and fuel economy. *SAE Paper No. 2004-01-0064*, 2004.

[131] K. Nicholas, K. Hedrick, and F. Borrelli. Integrating traffic data and model predictive control to improve fuel economy. *IFAC Symposium on Control in Transportation Systems*, 2009.

[132] V. Marano, P. Tulpule, S. Stockar, S. Onori, and G. Rizzoni. Comparative study of different control strategies for plug-in hybrid electric vehicles. *SAE World Congress and Exhibition Detroit MI USA*, 2004.

[133] S. J. Moura, J. L. Stein, and H. K. Fathy. Battery health-consious power management for plug-in hybrid vehicles via stochastic control. *ASME Dynamic Systems and Control Conference*, 2010.

[134] C. Zhang, A. Vahidi, X. Li, and D. Essenmacher. Role of trip information preview in fuel economy of plug-in hybrid vehicles. *ASME Dynamic Systems and Control Conference*, 2009.

[135] A. Taghavipour, M. Vajedi, N.L. Azad, and J. McPhee. Predictive power management strategy for a phev based on different levels of trip information. *IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling (E-COSM'12)*, pages 326 – 333, 2012.

[136] D. Smith, H. Lohse-Busch, and D. Irick. A preliminary investigation into the mitigation of plug-in hybrid electric vehicle tailpipe emissions through supervisory control methods. *SAE Technical Paper*, pages No.2010–01–1266, 2010.

[137] D. Kum, H. Peng, and N.K. Bucknor. Optimal catalyst temperature management of plug-in hybrid electric vehicles. *American Control Conference (ACC)*, pages 2732 – 2738, 2011.

[138] Y. Tian, W. Sun, D. Qu, and L. Wang. Study of a exhaust after-treatment system applied to hybrid vehicle. *Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, pages 1–4, 2010.

[139] S.H. Chan, D.L. Hoang, and P.L. Zhou. Heat transfer and chemical kinetics in the exhaust system of a cold-start engine fitted with a three way catalytic converter. *Proceedings of the Institution of Mechanical Engineering Part D: Journal of Automobile Engineering*, 214(7):765 – 777, 2000.

[140] J. C. Wurzenberger, G. Auzinger, R. Heinzle, and R. Wanker. 1d modeling of reactive fluid dynamics, cold start behavior of exhaust systems. *SAE Technical Paper*, pages No.2006–01–1544.

[141] J.S. Souder and J.K. Hedrick. Adaptive sliding mode control of airfuel ratio in internal combustion engines. *International Journal of Robust and Nonlinear Control*, 14(6):525–541, 2011.

[142] B. Shaw, G. D. Fischer, and J. K. Hedrick. A simplied coldstart catalyst thermal model to reduce hydrocarbon emissions. *Proceedings of 15th Triennial World Congress of the International Federation of Automatic Control*, pages 1915 – 1921, 2002.

[143] M. Ceraolo, A. di Donato, and G. Franceschi. A general approach to energy optimization of hybrid electric vehicles. *IEEE Transaction on Vehicular Technology*, 57(3):1433–1441, 2008.

[144] C.C. Lin, H. Peng, J. W. Grizzle, and J.M. Kang. Power management strategy for a parallel hybrid electric truck. *IEEE Trans. Control Syst. Technol*, 11(6):839 849, 2003.

[145] I. Kolmanovsky, M. Nieuwstadt, and J. Sun. Optimization of complex powertrain systems for fuel economy and emissions. *IEEE International Conference on Control Applications, HI*, pages 833 – 839, 1999.

[146] S. M. Lukic and A. Emadi. Emissions and fuel economy trade-off for hybrid vehicles using fuzzy logic. *Math. Comput. Simul.*, 66(2):155 172, 2004.

[147] D. Kum, H. Peng, and N.K. Bucknor. Supervisory control of parallel hybrid electric vehicles for fuel and emission reduction. *Journal of Dynamic Systems, Measurement, and Control*, 133(6):833 – 839, 2011.

[148] M. Dorri and A. Shamekhi. Design of an optimal control strategy in a parallel hybrid vehicle in order to simultaneously reduce fuel consumption and emissions. *SAE Technical Paper*, pages No.2011–01–0894, 2011.

[149] H. Sagha, S. Farhangi, and B. Asaei. Modeling and design of a nox emission reduction strategy for lightweight hybrid electric vehicles. *35th Annual Conference of IEEE Industrial Electronics,(IECON '09)*, pages 334 – 339, 2009.

[150] Z. Gao, M. Kim, J. Choi, C.S. Daw, J.E. Parks II, and D.E. Smith. Cold-start emissions control in hybrid vehicles equipped with a passive adsorber for hydrocarbons and nitrogen oxides. *Proc. Inst. Mech. Eng., Part D (J. Automob. Eng.)*, 226(10):1396 1407, 2012.

[151] L. Guzzella and A. Sciarretta. *Vehicle Propulsion Systems: Introduction to Modeling and Optimization.* Springer, 2 edition, 2007.

[152] H.N. Iordanou and B. Surgenor. Experimental evaluation of the robustness of discrete sliding mode control versus linear quadratic control. *IEEE Transactions on Control Systems Technology*, 5(2):254–260, 1997.

[153] D.M. Lamberson. *Torque Management of Gasoline Engines*. Msc dissertation, University of California at Berkeley, Berkeley (CA), 2003.

[154] A. Taghavipour, N.L. Azad, and J. McPhee. Design and evaluation of a predictive powertrain control system for a plug-in hybrid electric vehicle to improve fuel economy and emissions. *IMechE, Part D: Journal of Automobile Engineering*, page Accepted for publication, 2014.

[155] P.R. Sanketi, J.C. Zavala, and J.K. Hedrick. Dynamic surface control of engine exhaust hydrocarbons and catalyst temperature for reduced cold start emissions. *Proc. of International Federation of Automatic Control (IFAC) Conference*, 2005.

[156] J.C. Zavala. *Engine Modeling and Control for Minimization of Hydrocarbon Cold-start Emissions in SI Engines*. Phd dissertation, University of California, Berkeley, Berkeley (CA), 2007.

[157] R. S. Razavian, A. Taghavipour, N.L. Azad, and J. McPhee. Design and evaluation of a real-time optimal control system for series hybrid electric vehicles. *International Journal of Electric and Hybrid Vehicles*, 4(3):260–288, 2012.

[158] M. Vajedi, A. Taghavipour, N.L. Azad, and J. McPhee. A comparative analysis of route-based power management strategies for real-time application in plug-in hybrid electric vehicles. *American Control Conference (ACC)*, pages 2612–2617, 2014.

[159] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. Explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[160] F. Borrelli. *Constrained Optimal Control Of Linear And Hybrid Systems*, volume 290. Lecture Notes in Control and Information Sciences, Springer, 2003.

[161] P. Tondel, T. A. Johansen, and A. Bemporad. An algorithm for multiparameteric quadratic programming and explicit mpc solution. *Automatica*, 39(5):489–497, 2003.

[162] T. Geyer, F.D. Torrisi, and M. Morari. Optimal complexity reduction of polyhedral piece- wise affine systems. *Automatica*, 44(1):1728–1740, 2008.

[163] T.A. Johansen. On multi-parametric nonlinear programming and explicit nonlinear model predictive control. *41th IEEE Conference on Decision and Control (CDC)*, pages 2768–2773, 2002.

[164] T.A. Johansen. Approximate explicit receding horizon control of constrained nonlinear systems. *Automatica*, 40(1):293–300, 2004.

[165] B. Lincoln and A. Rantzer. Relaxing dynamic programming. *IEEE Transactions Automatic Control*, 51(8):1249–1260, 2006.

[166] K. Fukuda. Polyhedral computation faq. *http://www.ifor.math.ethz.ch/staff/fukuda/*, 2000.

[167] S.W. Cheng and A. Janadan. Algotithms for ray-shooting and intersection searching. *Journal of Algorithms*, 13(3):670–692, 1992.

[168] M. Kvasnica. *Efficient Software Tools for Control and Analysis of Hybrid Systems*. Phd dissertation, ETH Zurich, 2008.

[169] M. Baotic, F. Borrelli, A.Bemporad, and M. Morari. Efficient on-line computation of constrained optimal control. *SIAM Journal on Control and Optimization*, 47(5):2470–2489, 2008.

[170] P. Tndel, T.A. Johansen, and A. Bemporad. Evaluation of piecewise affine control via binary search tree. *Automatica*, 39(5):945–950, 2003.

[171] M. De Berg, O. Schwarzkopf, M. Van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 2nd edition, 2000.

[172] M. Kvasnica, P. Grieder, and M. Baotic. Multi-parametric toolbox (mpt). *Available from http://control.ee.ethz.ch/ mpt/*, 2004.

[173] J.J. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, 1991.

[174] G. Ferrari-Trecate, F.A. Cuzzola, D. Mignone, and M. Morari. Analysis of discrete-time piecewise affine and hybrid systems. *Automatica*, 38(12):2139–2146, 2002.

[175] D. Liberzon. *Switching in Systems and Control*. Birkhauser Boston, 2003.

[176] D.E. Rivera. Control-relevant parameter estimation: A systematic procedure for prefilter design. *American Control Conference (ACC)*, pages 237–241, 1991.

[177] A. Seaman, T. Dao, and J. McPhee. A survey of mathematics-based equivalent-circuit and electrochemical battery models for hybrid and electric vehicle simulation. *Journal of Power Sources*, 256:410–423, 2014.

[178] H. He, R. Xiong, and J. Fan. Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach. *Energies*, 4(4):582–598, 2011.

[179] I.R. Kendall and R.P. Jones. An investigation into the use of hardwarein- the-loop simulation testing for automotive electronic control systems. *Control Engineering Practice*, 7(11):1343–1356, 1999.

[180] C. Xi-ming, X. Liang-fei, H. Bin, L. Xi-hao, and O. Ming-gao. Real time simulation of shev powertrain system. *Journal of System Simulation*, 16, 2004.

[181] R. McGee. Ford motor company hybrid electric escape powertrain control system development and verification utilizing hardware-in-the-loop technology. *dSPACE User Conference*, 2002.

[182] S. Raman, N. Sivashankar, W. Milam, W. Stuart, and S. Nabi. Design and implementation of hil simulators for powertrain control system software development. *American Control Conference (ACC)*, pages 709–713, 1999.

[183] K. Hagiwaraa, S. Terayamab, Y. Takedaa, K. Yodab, and S. Suzukia. Development of automatic transmission control system using hardware-in-the-loop simulation system. *JSAE Review*, 23(1):55–59, 2002.

[184] T. Ersal, H.K. Fathy, J.L. Stein, and L.S. Louca. Automated proper modeling: Theoretical developments and applications. *Proceedings of the SPIE Modeling and Simulation Symposium, Society of Optical Engineering*, 2006.

[185] R.M. Howe. Real-time multi-rate asynchronous simulation with single and multiple processors. *Proc. SPIE 3369, Enabling Technology for Simulation Science II*, 3369:331–342, 1998.

[186] D.M.Lane, G.J. Falconer, G. Randall, and I. Edwards. Interoperability and synchronisation of distributed hardware-in-the-loop simulation for underwater robot development: Issues and experiments. *IEEE International Conference on Robotics and Automation*, 3369:909–1014, 2001.

[187] Texas Instruments Europe. Field orientated control of 3-phase ac-motors. 1998.

[188] A. Manuel and J. Francis. Simulation of direct torque controlled induction motor drive by using space vector pulse width modulation for torque ripple reduction. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 2(9):4471–4478, 2013.

[189] I. Takahashi and T. Noguchi. A new quick-response and high-efficiency control strategy of an induction motor. *IEEE Transactions on Industry Applications*, 5:820–827, 1986.

APPENDICES

Appendix A

Model Parameters and Variables

Table A.1 reviews model parameters and variables.

$$I_v^{'} = m\frac{R_{tire}^2}{K} + I_m^{'}K + I_r^{'}K$$
$$I_g^{'} = I_g + I_s$$
$$I_e^{'} = I_e + I_C$$
$$T_d = mgf_rR_{tire} + 0.5\rho_{air}A_dc_d(\omega_r/K)^2R_{tire}^3$$
$$(\frac{r_2}{s_2})\omega_r r_1 + \omega_g s_1 = \omega_e(r_1 + s_1)$$

Table A.1: Variables and Parameters Value

Symbol	Unit	Value
$\bar{\alpha}$	$(kg/h)(rad/s)^{-2}$	0.02
$\bar{\beta}$	$(kg/h)W^{-1}$	1.86
$\dot{m}_f$	(kg/h)	-
I_g	kgm^2	0.1
I_s	kgm^2	0.1
I_e	kgm^2	0.5
I_C	kgm^2	0.1
I_m	kgm^2	0.1
I_r	kgm^2	0.1
R_{tire}	m	0.3
K	-	4.11
m	kg	1380
g	m/s^2	9.81
f_r	-	0.02
ρ_{air}	kg/m^3	1.2
A_d	m^2	2.5
c_d	-	0.2
r_1	-	78
s_1	-	30
r_2	-	78
s_2	-	30
V_{oc}	V	345.6
R_{batt}	Ω	0.93
Q_{batt}	kWh	4.4
T_{min-e}	Nm	0
T_{max-e}	Nm	142
T_{min-m}	Nm	-300
T_{max-m}	Nm	300
T_{min-g}	Nm	0
T_{max-g}	Nm	40
ω_{min-e}	rpm	0
ω_{max-e}	rpm	6000
ω_{min-m}	rpm	-13600
ω_{max-m}	rpm	13600
ω_{min-g}	rpm	-10000
ω_{max-g}	rpm	10000
SOC_{min}	-	0
SOC_{max}	-	1

Appendix B

Controls Design for Electric Drive

B.1 Electric Drive Control

Semiconductor improvements in both power and signal electronics have enabled the development of effective AC drive controllers with lower power dissipation hardware and more accurate control structures.

The electrical drive controls become more accurate since only the DC current and voltage are controlled, but the three-phase currents and voltages are also managed by so-called vector controls. One of the most efficient forms of vector control scheme is the Field-oriented Control (FOC). It is based on three major points: the machine current and voltage space vectors, the transformation of a three-phase speed- and time-dependent system into a two-coordinate time-invariant system, and effective Pulse Width Modulation pattern generation. Based on these factors, the control of an AC machine acquires every advantage of DC machine control and frees itself from the mechanical commutation drawbacks. Furthermore, by achieving a very accurate steady-state and transient response control, this control structure leads to high dynamic performance in terms of response times and power conversion [187].

AC motor control structures generally apply three 120 spatially displaced sinusoidal voltages to the three stator phases. In most classic AC drives, the generation of the three sine waves is based on motor electromechanical characteristics and on an equivalent model for the motor in its steady state. Furthermore, the control looks like three separate single-phase system controls rather than one control of a three-phase system. The Field-oriented Control (FOC) consists of controlling the stator currents represented by a

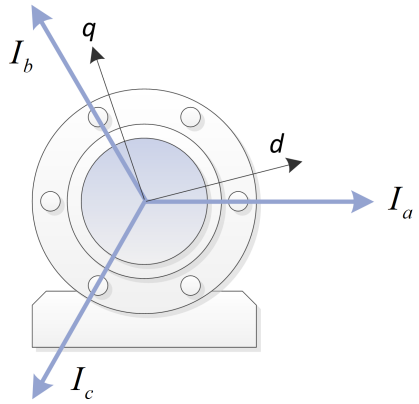

Figure B.1: d-q representation of a permanent magnet synchronous motor

vector. This control is based on projections that transform a three-phase time- and speed-dependent system into a two-coordinate (d and q coordinates) time-invariant system (Figure B.1). These projections lead to a structure similar to that of a DC machine controller. FOC-controlled machines need two constants as input references: the torque component (aligned with the q coordinate) and the flux component (aligned with the d coordinate). As Field-orientated Control is simply based on projections, the control structure handles instantaneous electrical quantities. This makes the control accurate in every working operation (steady-state and transient) and independent of the limited-bandwidth mathematical model [1].

However, FOC scheme is complex in terms of implementation due to its dependance on the motor parameters. These parameters should be continuously estimated, since their values are changing during the machine operation. On the other hand, Direct torque control (DTC) approach only requires the stator resistance to estimate the torque and flux.

In DTC approach, electromagnetic torque and flux can be controlled independently through a switching table. Selecting the inverter switching mode limits the torque and flux error in comparison to their desired values within hysteresis bands. The basic DTC scheme consists of two comparators for torque and flux which will be introduced in the following sections.

DTC control suggests a more robust scheme with simpler implements beside less dependency on the machine parameters in comparison to FOC approach. Nevertheless, it is difficult to control the torque and flux at low speed with this approach. High torque ripple,

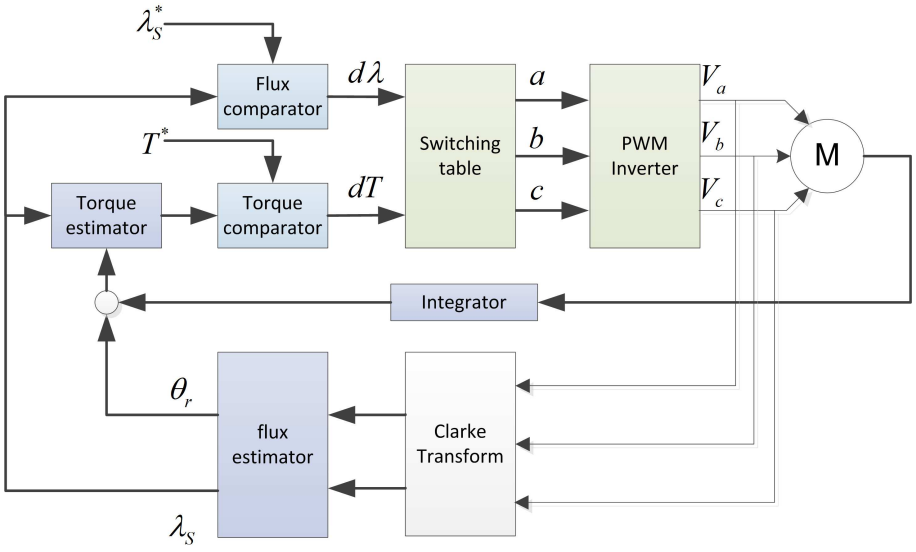

Figure B.2: Direct torque control scheme

a high sampling frequency required for digital implementation of hysteresis controllers, and current and torque distortion during the change of sectors are other DTC approach down sides. One way to minimize torque ripples is using the space vector modulation (SVM). The objective of DTC-SVM is to estimate a reference stator voltage vector and modulate it with SVM technique in order to drive power gates of the inverter with a constant switching frequency. As a result, the inverter can produce a voltage vector of any direction and magnitude. This means that the stator flux can be produced in any direction and magnitude which smoothen the resultant torque [188].

The DTC scheme is shown in Figure B.2.

B.1.1 Clarke Transformation

For stator voltage and currents we can write:

$$\begin{aligned} \vec{V_S} &= V_a e^{j0} + V_b e^{j\frac{2\pi}{3}} + V_c e^{j\frac{4\pi}{3}} \\ \vec{I_S} &= I_a e^{j0} + I_b e^{j\frac{2\pi}{3}} + I_c e^{j\frac{4\pi}{3}} \end{aligned} \tag{B.1}$$

Now we can change them to d-q coordinates. We call this, forward Clarke transformation. The forward Clarke transformation converts a 3-phase system (a, b, c) to a 2-phase coordinate system (d,q). Assuming that the a axis and the d axis are in the same direction, the quadrature-phase stator voltages and currents (V_d,V_q), (I_d,I_q) are related to the actual 3-phase stator currents as follows:

$$\begin{aligned} V_d &= \tfrac{2}{3}\{V_a - \tfrac{1}{2}(V_b + V_c)\} \\ V_q &= \tfrac{2}{3}\{\tfrac{\sqrt{3}}{2}(V_b - V_c)\} \\ I_d &= \tfrac{2}{3}\{I_a - \tfrac{1}{2}(I_b + I_c)\} \\ I_q &= \tfrac{2}{3}\{\tfrac{\sqrt{3}}{2}(I_b - I_c)\} \end{aligned} \tag{B.2}$$

For the non-power-invariant transformation, the quantities V_d and V_a are equal. If it is assumed that $V_a + V_b + V_c = 0$, the quadrature-phase components can be expressed utilizing only two phases of the 3-phase system for the voltage and the current:

$$\begin{aligned} V_d &= V_a \\ V_q &= \tfrac{1}{\sqrt{3}}\{V_a + 2V_b\} \\ I_d &= I_a \\ I_q &= \tfrac{1}{\sqrt{3}}\{I_a + 2I_b\} \end{aligned} \tag{B.3}$$

So the inverse Clarke transformation will be:

$$\begin{aligned} V_a &= V_d \\ V_b &= -\tfrac{1}{2}V_d + \tfrac{\sqrt{3}}{2}V_q \\ V_c &= -\tfrac{1}{2}V_d - \tfrac{\sqrt{3}}{2}V_q \\ I_a &= I_d \\ I_b &= -\tfrac{1}{2}I_d + \tfrac{\sqrt{3}}{2}I_q \\ I_c &= -\tfrac{1}{2}I_d - \tfrac{\sqrt{3}}{2}I_q \end{aligned} \tag{B.4}$$

B.1.2 Flux and Torque Estimation

Using vector notation of Kirchhoff voltage law and by assuming the same resistance for each phase in stator, the equation of the stator winding for each phase can be written as [1]:

$$\vec{V_S} = R_S\vec{I_S} + \frac{d}{dt}\vec{\lambda}_S \tag{B.5}$$

By considering integration period of T_s, the corresponding flux can be approximated in d-q coordinates as:

$$\begin{aligned} \lambda_{dS} &= (V_d - R_S I_d)T_s + \lambda_{dS0} \\ \lambda_{qS} &= (V_q - R_S I_q)T_s + \lambda_{qS0} \end{aligned} \tag{B.6}$$

where R_S is the stator equivalent resistance. λ_{dS0} and λ_{qS0} are the initial value of flux in the d and q directions at each sampling time. The magnitude and the phase of the flux can be found as:

$$\begin{aligned} \lambda_S &= \sqrt{\lambda_{dS}^2 + \lambda_{qS}^2} \\ \theta_r &= tan^{-1}(\tfrac{\lambda_{qS}}{\lambda_{dS}}) \end{aligned} \tag{B.7}$$

Now, the phase difference between the rotor and stator equivalent flux vectors can be calculated as:

$$\delta = \theta_r - \int \omega_m dt \tag{B.8}$$

The motor produced torque is proportional to the cross product of the rotor and stator equivalent flux vectors. As a result, the torque magnitude can be estimated as:

$$T_{est} = \frac{3}{2} p \lambda_S \lambda_{pm} sin\delta \tag{B.9}$$

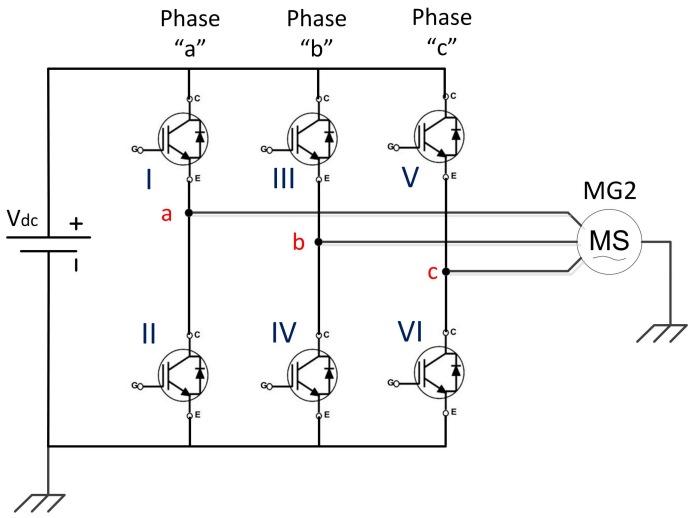

Figure B.3: MG2 inverter schematic

B.1.3 Direct Torque Control

According to Figure B.2, the torque and flux errors are required for DTC scheme. Based on the setpoint torque magnitude (T^*), the stator desired flux (λ_S^*) is calculated.

$$\lambda_S^* = \frac{2T^*}{3p\lambda_{pm}} \tag{B.10}$$

The schematic of the IGBTs of the electric drive is shown in Figure B.3. By turning the corresponding insulated gate bipolar transistors(IGBTs)on and off, different voltage values can be applied to the three phases of the motor.

Table B.1 shows how inverter switching scheme can produce different voltage vectors. These voltage vectors are presented in Figure B.4 which make six sectors on the motor phase plane (Θ_i).

Based on the sector where the stator flux vector is located at each time step, a specific voltage vector can be chosen to change the flux and torque magnitude. Choosing the appropriate voltage vector can alternate λ_S magnitude and δ in order to remove flux ($d\lambda$) and torque (dT) error, respectively. Note that δ increase to 90° leads to the maximum produced torque magnitude. Assuming λ_S is located in sector Θ_1, choosing $\vec{v}_2$ increases the torque, but $\vec{v}_6$ leads to torque decrease.

Table B.1: Switching table

voltage vector	a	b	c
v_0	0	0	0
v_1	1	0	0
v_2	1	1	0
v_3	0	1	0
v_4	0	1	1
v_5	0	0	1
v_6	1	0	1
v_7	1	1	1

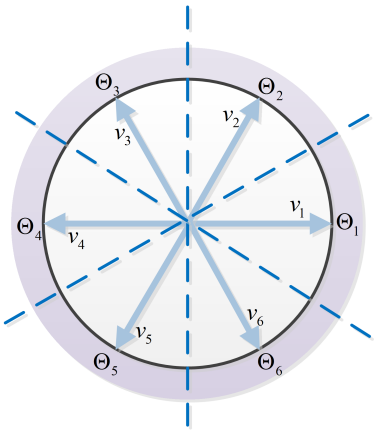

Figure B.4: Six sectors representation

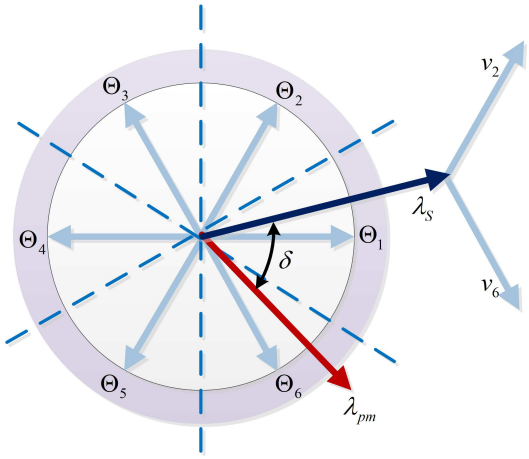

Figure B.5: Voltage vector selection scheme

Table B.2: Torque and flux hysteresis

$b_\lambda = 1$	for	$d\lambda < -\varepsilon_\lambda$
$b_\lambda = 0$	for	$d\lambda > \varepsilon_\lambda$
$b_T = 1$	for	$dT < -\varepsilon_T$
$b_T = 0$	for	$dT = 0$
$b_T = -1$	for	$dT > \varepsilon_T$

Torque (b_T) and flux (b_λ) variation indices can be found based on the torque (dT) and flux ($d\lambda$) errors according to Table B.2.

For different values of b_λ and b_T, one can populate the Takahashi and Naguchi switching table [189] in order to control the motor torque.

In the next section, the design scheme is applied to the high-fidelity model of electric drive which is developed in the MapleSim software.

B.2 Power Electronics High-fidelity Model

The permanent magnet synchronous motor and the associated inverter average model of MG2 is shown in Figure B.6.

Table B.3: Takahashi and Noguchi switching table

Flux variation	Torque variation	Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6
$b_\lambda = 1$	$b_T = -1$	v_6	v_1	v_2	v_3	v_4	v_5
	$b_T = 0$	0	0	0	0	0	0
	$b_T = +1$	v_2	v_3	v_4	v_5	v_6	v_1
$b_\lambda = 0$	$b_T = -1$	v_5	v_6	v_1	v_2	v_3	v_4
	$b_T = 0$	0	0	0	0	0	0
	$b_T = +1$	v_3	v_4	v_5	v_6	v_1	v_2

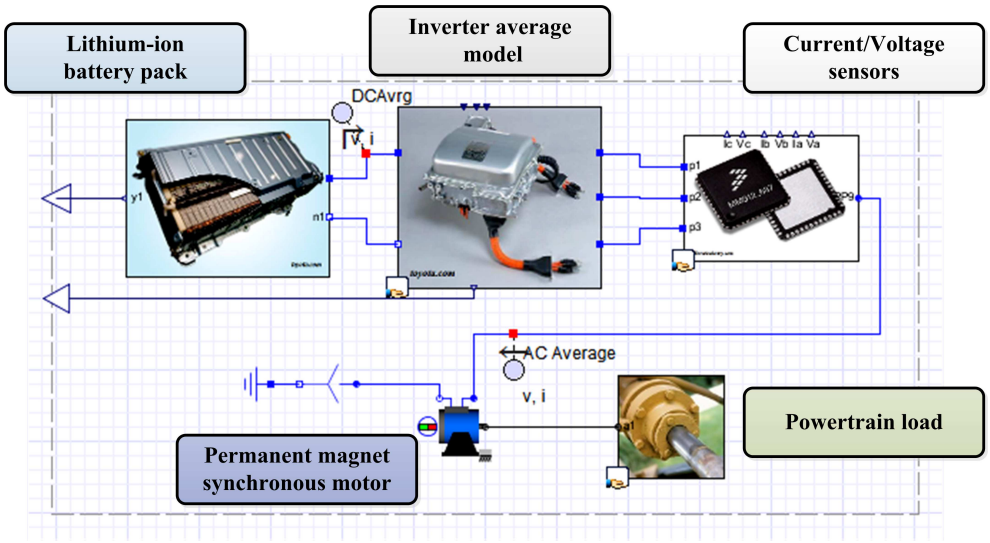

Figure B.6: Electric drive high-fidelity model in MapleSim

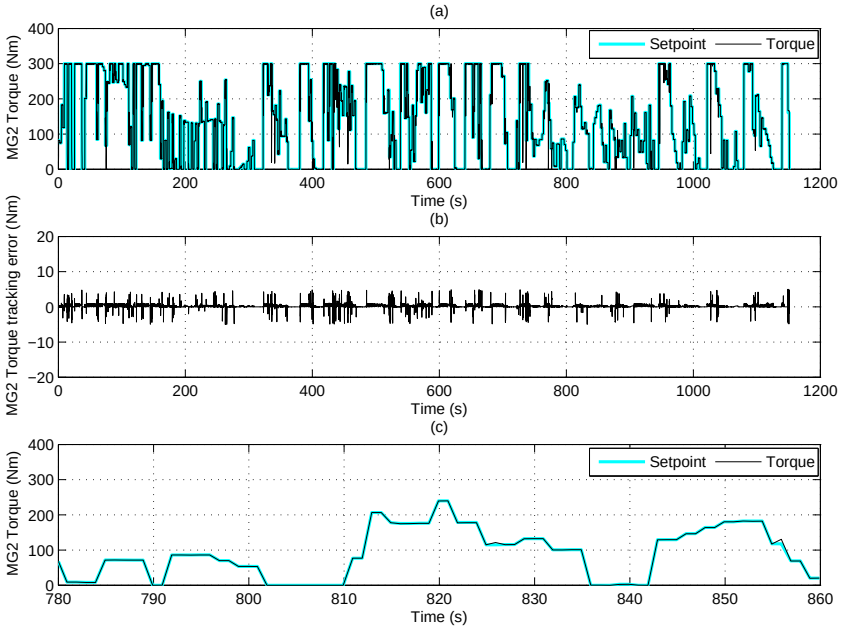

Figure B.7: (a) MG2 torque along one UDDS drive cycle for Blended mode CRPE-eMPC strategy (b) MG2 Torque tracking error (c)zoomed view for $t \in [780, 860]s$

This model consists of three main parts: inverter average model, current/voltage sensors, and permanent magnet synchronous motor. The chemistry-based model of the Lithium-ion battery pack is considered to determine the maximum available DC voltage at each time step (V_{dc} in Figure B.3). The exerted torque load on the electric drive from the powertrain is also shown in Figure B.6. We can also consider the same model for MG1.

B.3 Results of Simulation

The control scheme should be able to follow all MG2 torque setpoints which are discussed in chapter 4,5, and 6. In this section, the performance of the control scheme is evaluated by considering the torque setpoints of Figure 6.11.

The result of simulations for the motor torque is shown in Figure B.7. Figure B.7-b demonstrates the setpoint tracking performance of designed control scheme within 80s.

Figure B.8 presents the result for different variables which are involved in controlling the motor torque with data acquisition frequency of 50 Hz. Figure B.9 shows how fast the switches are turning on and off to make the motor produce the desired torque within smaller time period of 40 ms. Here, the data acquisition frequency is 5 kHz.

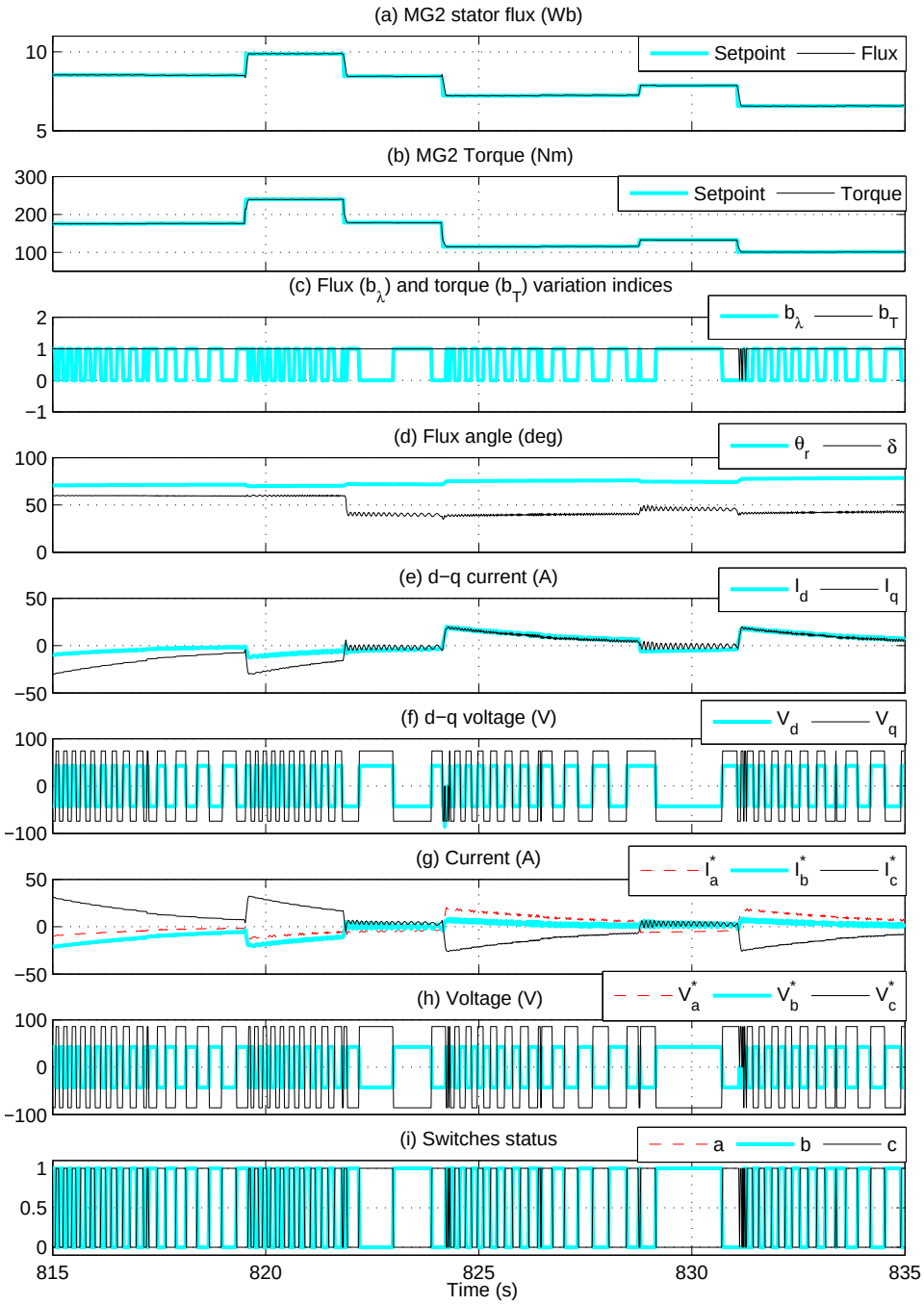

Figure B.8: (a) MG2 equivalent stator flux (b)MG2 torque (c) flux and torque variation indices (d) flux vector direction (e) dq current (f) dq voltage (g) Measured currents (h)Applied voltages (i) IGBTs on/off status for measurement sampling time of 20 ms

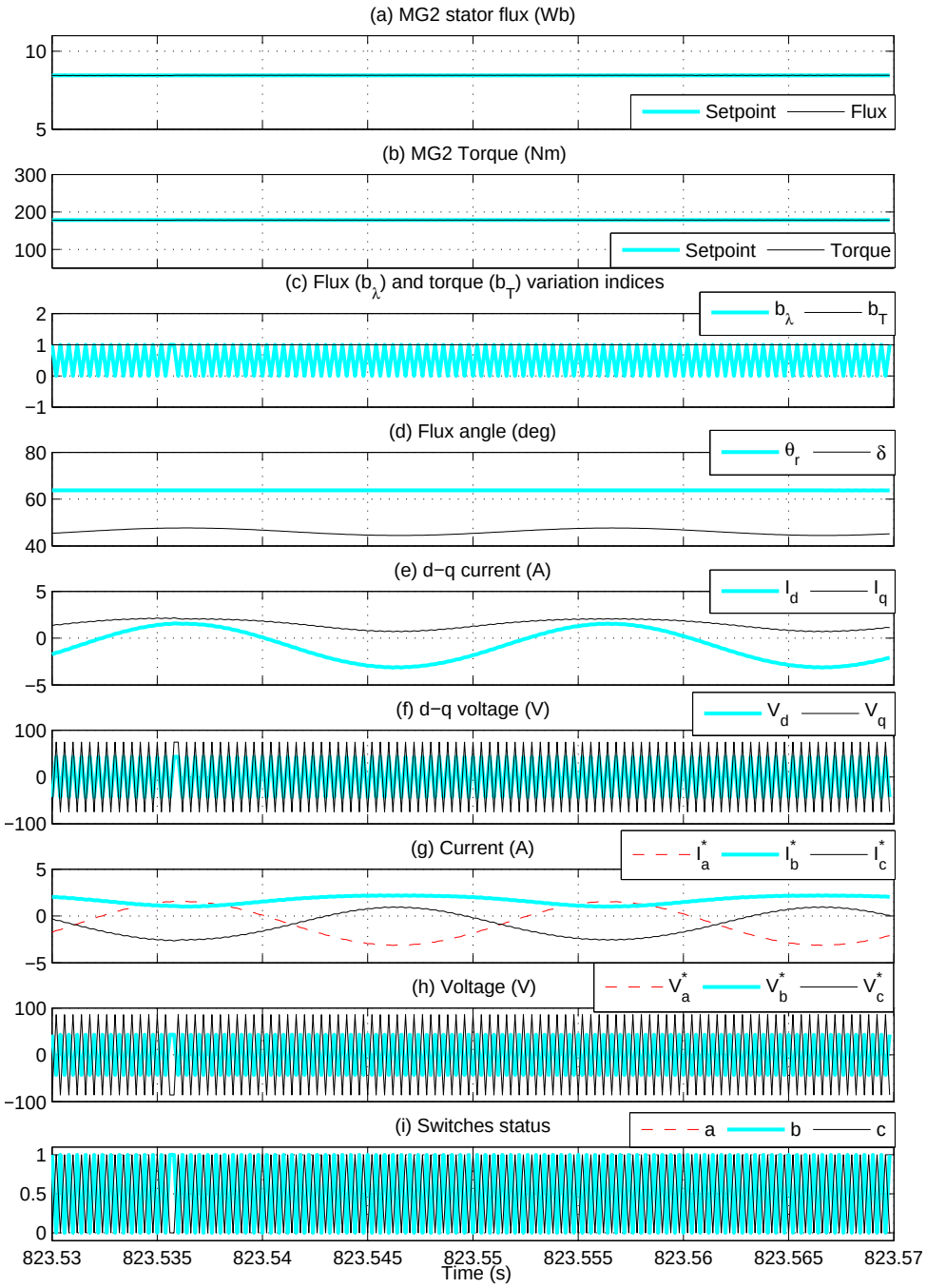

Figure B.9: (a) MG2 equivalent stator flux (b)MG2 torque (c) flux and torque variation indices (d) flux vector direction (e) dq current (f) dq voltage (g) Measured currents (h)Applied voltages (i) IGBTs on/off status for measurement sampling time of 20 μs

Appendix C

CAN Bus Arbitration ID

On a CAN bus, each of the nodes are directly connected to the bus, and there is no central control unit to regulate the communications. Instead, CAN bus is a serial message-based protocol, where each node can send and receive messages when the bus is free. The priority of each message is identified by an arbitration ID, where lower IDs have the higher priority.

The arbitration ID also serves as the name tag for each message. When a node transmits a message on the CAN bus, the message is received by every node on the bus. Each node can then ignore the message, or do a specific task based on the ID and the contents of the message. The other part of a CAN message is the data frame. A CAN data frame is defined byte-wise, i.e., the message consists of groups of bytes that contain an integer number. Thus, to send a variable, it should be scaled to an integer number, based on its range and required accuracy. When the variable is transmitted and received, it is scaled back to its original format. When a variable requires more than one data byte to be transmitted (when its range exceeds [0,255]), it is divided into a number of bytes. Careful attention is required during the processes of turning the variable into separate bytes and the ordering of bits in each byte. In a CAN message, the bits are sent one by one as a serial signal. When the whole message is sent, it is interpreted as a number of bytes. However, the way the bits are grouped into bytes shows inconsistency between different devices and software packages. For example, the Woodward compiler takes the first eight bits of the message as the byte 7, but LabVIEW considers the same order of bits as byte 0 (see Figure C.1).

There are also various ways to interpret the multi-byte numbers. This is referred to as the endianness. In little endian format (Intel), the least significant byte is first and has lower index, whereas in big endian format (Motorola), the least significant byte is sent last

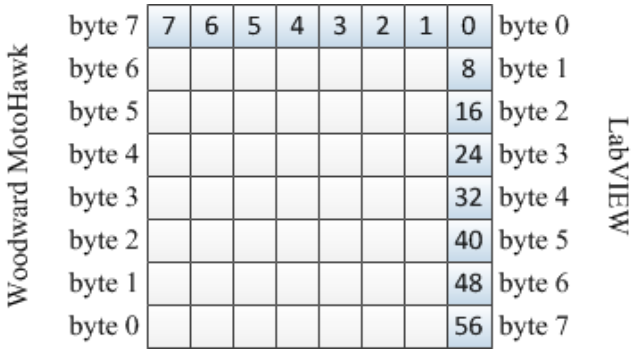

Figure C.1: Different byte allocation methods in CAN data frame

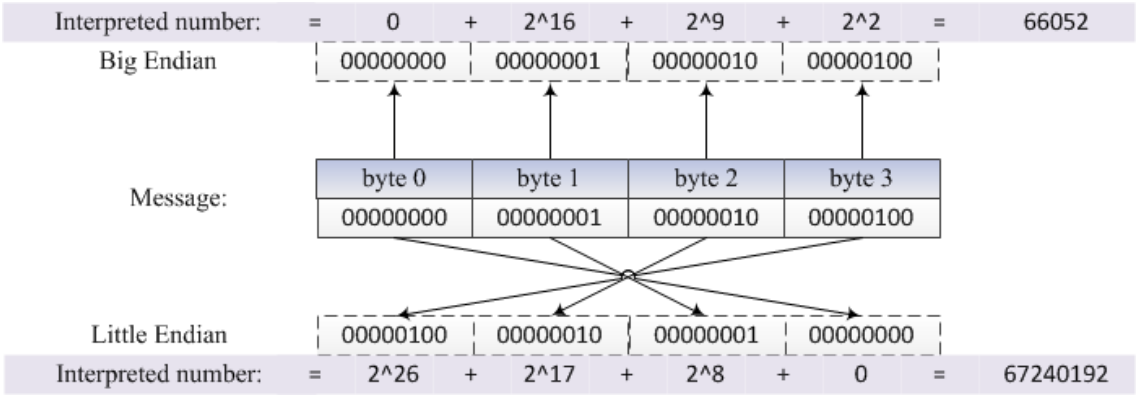

Figure C.2: An example for different endianness definitions

and has the highest index. The difference in endianness is illustrated in Figure C.2. In our setup, the little endian mode is used to interpret data bytes.