

2021

Powertrain Fuel Consumption Modeling and Benchmark Analysis of a Parallel P4 Hybrid Electric Vehicle Using Dynamic Programming

Aaron Robert Mull
West Virginia University, armull@mix.wvu.edu

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>



Part of the [Acoustics, Dynamics, and Controls Commons](#), and the [Navigation, Guidance, Control, and Dynamics Commons](#)

Recommended Citation

Mull, Aaron Robert, "Powertrain Fuel Consumption Modeling and Benchmark Analysis of a Parallel P4 Hybrid Electric Vehicle Using Dynamic Programming" (2021). *Graduate Theses, Dissertations, and Problem Reports*. 10154.

<https://researchrepository.wvu.edu/etd/10154>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

**POWERTRAIN FUEL CONSUMPTION MODELING AND BENCHMARK
ANALYSIS OF A PARALLEL P4 HYBRID ELECTRIC VEHICLE USING
DYNAMIC PROGRAMMING**

Aaron Mull

**Thesis submitted to the Statler College of Engineering and Mineral Resources at West
Virginia University in Partial fulfillment of the requirements for the degree of**

**Master of Science
In
Mechanical Engineering**

Andrew C. Nix Ph.D., Committee Chairperson

Scott Wayne, Ph.D.

Mario Perhinschi, Ph.D.

Department of Mechanical and Aerospace Engineering

Morgantown, West Virginia

2021

**Keywords: Hybrid-Electric Vehicle, Fuel Consumption Modeling, Benchmark Analysis,
Dynamic Programming, Optimal Control Strategy,**

Copyright 2021 Aaron Mull

Abstract

POWERTRAIN FUEL CONSUMPTION MODELING AND BENCHMARK ANALYSIS OF A PARALLEL P4 HYBRID ELECTRIC VEHICLE USING DYNAMIC PROGRAMMING

Aaron Mull

As regulations on the emission of greenhouse gasses continue to tighten on the automotive industry, the production of hybrid electric vehicles has gained significant popularity in recent years. With the increase in production, there has been a parallel demand in the advancement of both mechanical hardware and control system implementation used in these vehicles. A critical factor in the efficient operation of a hybrid electric vehicle is the energy management strategy where the goal is to maximize the efficient use of fuel energy to propel the vehicle. Designing a fuel-efficient control system is a complex challenge due to the degrees of freedom that exist in the control of a hybrid electric vehicle. Several methods exist for the real-time implementation of control strategies that employ heuristic or optimization-based algorithms; however, these control strategies typically rely on the results of offline optimization as a benchmark against which the control strategies are evaluated. Offline energy management optimization strategies require a pre-defined driving schedule for which the operation of the powertrain can be evaluated to determine the globally optimal control policy. The goal of this work is to develop a hybrid electric vehicle model that is suitable for use in a dynamic programming algorithm that provides the benchmark for optimal control of the hybrid powertrain. The benchmark analysis employs dynamic programming by backward induction to determine the globally optimal solution by solving the energy management problem starting at the final timestep and proceeding backwards in time. This method requires the development of a backwards facing model that propagates the wheel speed of the vehicle for the given drive cycle through the driveline components to determine the operating points of the powertrain. Although dynamic programming only searches the solution space within the feasible regions of operation, the benchmarking model must be solved for every admissible state at every timestep leading to strict requirements for runtime and memory. The backward facing model employs the quasi-static assumption of powertrain operation to reduce the fidelity of the model to accommodate these requirements. Verification and validation testing of the dynamic programming algorithm is conducted to ensure successful operation of the algorithm and to assess the validity of the determined control policy against a high-fidelity forward-facing vehicle model with a percent difference of fuel consumption of 1.2%. The benchmark analysis is conducted over multiple drive cycles to determine the optimal control policy that provides a benchmark for real-time algorithm development and determine control trends that can be used to improve existing algorithms. The optimal combined CS fuel economy of the vehicle is determined by the dynamic programming algorithm to be 32.99 MPG, a 52.6% increase over the stock 3.6L 2019 Chevrolet Blazer.

Acknowledgements

I would like to acknowledge the members of my thesis committee. First, my committee chair, advisor, and lead faculty advisor for the WVU EcoCAR team, Dr. Andrew Nix, for his guidance and dedication to both my academic affairs and the EcoCAR program as a whole. Dr. Nix provided me with the opportunity to combine my leadership skills and engineering knowledge as the engineering manager for the WVU EcoCAR team and provided me with his unwavering support in my educational activities. I commend Dr. Nix's ability to be more than just an advisor, but a true mentor both in and out of the lab. I thank Dr. Scott Wayne for providing significant expertise in vehicle powertrain modeling in and out of the classroom setting. The hybrid powertrain modeling course provided by Dr. Wayne laid the groundwork for understanding hybrid vehicle modeling and simulation that is applied in this work. Third, I thank Dr. Mario Perhinschi who provides a wealth of knowledge to the university in control systems and artificial intelligence. The concepts taught in his classes have been invaluable to the understanding of controls engineering that I have today.

I would like to extend a special thanks to the WVU EcoCAR team General Motors mentor Dr. William Cawthorne for his involvement with the team in his investment in myself as a professional. With Bill's guidance, I was able to implement several industry standard practices into the standard work for the WVU EcoCAR team that will benefit the program for years to come. I thank Bill for freely sharing his wealth of knowledge in controls and modeling as well as in the area of team management. Bill was especially helpful with helping me hone the necessary skills to secure my first engineering position after graduation and to start my career as an automotive engineer at General Motors.

I would like to acknowledge Dr. Brian Woerner for his involvement and leadership in the EcoCAR program. Under his guidance, Lane Department involvement in the EcoCAR program at WVU is at an all-time high.

I thank every one of the graduate leadership team of the WVU EcoCAR program. Firstly, project manager Benton Morris who took me under his wing as an undergraduate and ultimately taught me how to successfully manage the team. I would not have been able to attain such a well-rounded knowledge of each of the swimlanes without Benton's help managing the propulsion systems integration team. Second, the propulsion controls and modeling team leads, Thomas Harris, Jared Diethorn and Holden Fraser, working with each of you showed me how much I enjoyed working with vehicle controls and modeling. Whether it was writing lengthy modeling results documents with Thomas or late-night troubleshooting sessions with Jared, I am confident that the team would be significantly worse off without the knowledge each of you brought to the table. Third, I thank the CAVs team leadership, specifically Clay Vincent and Alex Colon who jumped into the program at a difficult time and worked diligently to get their systems up to par with what the competition required. The work done by this group significantly contributed to the successes of the WVU EcoCAR Team thus far. Finally, I thank the communications managers, Haleigh Resetar and Ryan Alexander, for giving the work done by the team more meaning than ever. Whether we are presenting to an elementary school classroom or politician, the impacts of the work accomplished by the WVU EcoCAR team has never reached a larger audience with a better message. I thank both of you for your efforts to make our work go the extra mile.

I also thank the undergraduates that have been part of the team throughout my time as engineering manager. Whether you were actively engaged with the work that I was doing, or succeeding in your activities on your own, I thank you for getting our team to where it is today. I would like to

specifically thank Jason Pulliam for his help in the design and integration of the electrical systems of the competition vehicle, James Keller for his help with the integration of mechanical systems, Billy Cawthorne for his overall helpfulness of doing whatever needed done, and Colin Kellett for putting in the extra work to be ready to receive to torch in becoming the next engineering manager for the team.

Dedication

This thesis is dedicated to my parents. Their unwavering support has been instrumental in my continued pursuit in education. I thank them every day for pushing me to have a strong work ethic and helping me understand my mistakes and move past failures to reach the achievements that I have throughout my educational career.

Table of Contents

1.0	Introduction.....	1
1.1	Hybrid Electric Vehicles	2
1.2	Hybrid Electric Vehicle Architectures	5
1.2.1	Classifications	5
1.2.2	Configurations.....	6
1.3	EcoCAR Mobility Challenge	8
1.4	Team Selected Architecture	9
1.5	HEV Supervisory Control and Energy Management Strategies	13
1.6	Importance of Benchmark Analysis for HEVs Powertrain Operation.....	15
2.0	Overview of HEV Energy Management Strategies	17
2.1	Heuristic Control Methods	18
2.1.1	Rule-Based.....	18
2.1.2	Fuzzy Logic	19
2.2	Optimal Control Methods.....	20
2.2.1	Real-Time Optimization: Equivalent Consumption Minimization Strategy	21
2.2.2	Global Optimization: Dynamic Programming.....	22
3.0	Hybrid Vehicle Modeling Fundamentals.....	24
3.1	Model Based Design and Core Concepts.....	24
3.1.1	Model Types	28

3.1.2	Model Structures	29
3.1.3	Model Fidelity Levels	31
3.2	Powertrain Simulation Environments and Tools	33
3.3	Powertrain Subsystem Models	34
3.3.1	Vehicle dynamics	34
3.3.2	Internal Combustion Engine	37
3.3.3	Energy Storage System	40
3.3.4	Electric Motor	41
3.3.5	Driveline	42
3.3.6	Driver	45
3.3.7	Overall Structure	46
4.0	Benchmark Analysis Methodology	48
4.1	Dynamic Programming History and Fundamentals	50
4.1.1	The Basic Problem	51
4.1.2	Bellman's Principal of Optimality	52
4.1.3	Dynamic Programming by Backwards Induction	53
4.1.4	Statement of the Optimization Problem	54
4.2	DP Problem Formulation	56
4.3	Benchmarking Model	58
4.3.1	Drive Cycle Data and Shift Schedule	58

4.3.2	Electric Motor Torque Lookup Table	60
4.3.3	Dynamic Programming Algorithm	62
5.0	Results and Discussion	73
5.1	Model Verification	74
5.1.1	Verification Cycle	74
5.1.2	Verification Tests	76
5.1.3	Cost Function Evaluation	83
5.2	DP Solution Simulink Validation	98
5.3	Benchmark Analysis	102
5.3.1	EMC City	102
5.3.2	EMC Highway	114
5.3.3	US06	125
5.3.4	UDDS	135
5.3.5	HWFET	143
5.4	Summary of Benchmark Analysis Results	152
5.5	Computational Considerations	155
6.0	Conclusions and Recommendations	156
7.0	Contributions and Future Work	157
8.0	References	159
9.0	Appendix	166

9.1 DP algorithm and Backward-Looking Model MATLAB Code: 166

List of Figures

Figure 1: Example of Engine Operational Point Shifted with Electric Motor (Left: Opportunity Charging) (Right: Motor Assist).....	4
Figure 2: Electric Motor Locations for a Parallel HEV	7
Figure 3: WVU Team Competition Vehicle Architecture	11
Figure 4: Flowchart of an HEV Supervisory Controller [6].....	14
Figure 5: Control Strategy Classification [10].....	18
Figure 6: V-Model of the Model Based Design Process	25
Figure 7: Mobility Challenge Energy Consumption Drive Cycles: a) {top} EMC City Cycle b) {bottom} EMC Highway Cycle.....	27
Figure 8: Basic Structure of a HEV model	29
Figure 9: Free-Body Diagram of Vehicle Moving in One Dimension [32]	35
Figure 10: Vehicle Body 1DOF Longitudinal Block.....	36
Figure 11: Illustration of the Components of a Four Stroke Spark-Ignition Internal Combustion Engine [1].....	38
Figure 12: a) Engine Fuel Consumption Map {left}. B) Engine Efficiency Map {right}.....	39
Figure 13: MathWorks Simulink Mapped SI Engine [33].....	40
Figure 14: MathWorks Simulink Datasheet Battery [35].....	41
Figure 15: MathWorks Simulink Mapped Motor [36]	42
Figure 16: Driveline Torque Path for Parallel P4 HEV	43
Figure 17: Torque Converter Schematic [2].....	44
Figure 18: MathWorks Simulink Longitudinal Driver [40]	46
Figure 19: Overall Structure of Simplified Powertrain Plant Model.....	47
Figure 20: Initial Stage Calculation of the Benchmarking Model.....	62

Figure 21: ICE Torque Determination Logic.....	67
Figure 22: Intermediate Stage Calculations of the DP Algorithm.....	70
Figure 23: Example of Intermediate Stages of the DP Algorithm at Stage $k = T - 2$	70
Figure 24: Example of Intermediate Stages of the DP Algorithm at Stage $k = T - 3$	71
Figure 25: Leaking Effect in DP Algorithm.....	72
Figure 26: Uniform State Transition Effect in DP Algorithm.....	73
Figure 27: Verification Drive Cycle Data.....	75
Figure 28: Shift Schedule of Verification Drive Cycle.....	76
Figure 29: ICE Only Verification Test Powertrain Power Output.....	78
Figure 30: CD Verification Test Powertrain Power Output.....	81
Figure 31: Fuel Only Cost Function Verification Test Powertrain Power Output.....	84
Figure 32: Operating Points of the Powertrain Components for the Fuel Only Cost Function Verification Test.....	86
Figure 33: Quadratic SOC Penalty Cost Function Verification Test Powertrain Power Output...	88
Figure 34: Sensitivity Analysis of Quadratic SOC Penalty Factor Weight Factor.....	89
Figure 35: Powertrain Operating Points for SOC Penalty Cost Function Verification Test.....	90
Figure 36: Motor and Braking Penalty Cost Function Verification Test Powertrain Power Output	92
Figure 37: Powertrain Operating Points for Motor and Braking Cost Function Verification Test	93
Figure 38: Overproduction Penalty Cost Function Verification Test Powertrain Power Output ..	95
Figure 39: Powertrain Operating Points for Overproduction Cost Function Verification Test....	96
Figure 40: Simulink Validation Model Torque Production Compared to DP Algorithm.....	99

Figure 41: Simulink Validation Model Vehicle State and Dynamics Compared to DP Algorithm	100
Figure 42: EMC City Drive Cycle Speed, Torque and Power Requirements	103
Figure 43: EMC City Shift Schedule	104
Figure 44: EMC City Powertrain Power Output	105
Figure 45: Operating Points of ICE and P4 Motor for EMC City Drive Cycle.....	107
Figure 46: Efficiency Distribution of ICE Only Operation for EMC City Drive Cycle.....	108
Figure 47: EMC City Optimal Operating Efficiency Distribution	108
Figure 48: Torque Production Analysis of Cruise Portion of EMC City Drive Cycle.....	110
Figure 49: Torque Production Analysis of Stop and Go Portion of EMC City Drive Cycle	111
Figure 50: EMC City Torque Split Ratio in Positive Torque and Power Regime.....	112
Figure 51: EMC City Torque Split Ratio in Negative Torque and Power Regime	113
Figure 52: Energy Analysis for EMC City Drive Cycle.....	113
Figure 53: EMC Highway Drive Cycle Speed, Torque and Power Requirements.....	115
Figure 54: EMC Highway Shift Schedule	116
Figure 55: EMC Highway Powertrain Power Output.....	117
Figure 56: Operating Points of the ICE and P4 Motor for the EMC Highway Drive Cycle	118
Figure 57: Efficiency Distribution of ICE Only Operation for EMC Highway Drive Cycle.....	119
Figure 58: EMC Highway Optimal Operating Efficiency Distribution.....	120
Figure 59: Torque Production Analysis of Cruise Portion of EMC Highway Drive Cycle	121
Figure 60: Torque Production of Highway Driving Portion of the EMC Highway Cycle.....	122
Figure 61: EMC Highway Torque Split Ratio in Positive Torque and Power Regime	123
Figure 62: EMC Highway Torque Split Ratio in Negative Torque and Power Regime	124

Figure 63: Energy Analysis for EMC Highway Drive Cycle	124
Figure 64: US06 Drive Cycle Speed, Torque and Power Requirements	126
Figure 65: US06 Shift Schedule	126
Figure 66: US06 Drive Cycle Powertrain Power Output.....	128
Figure 67: US06 Drive Cycle Powertrain Operating Points.....	129
Figure 68:Efficiency Distribution of ICE Only Operation for US06 Drive Cycle	130
Figure 69: US06 Optimal Control Policy Operating Efficiency Distribution	130
Figure 70: Torque Production Analysis of Cruise Portion of US06 Drive Cycle	131
Figure 71: Torque Production Analysis of Stop and Go Portion of US06 Drive Cycle.....	132
Figure 72: US06 Torque Split Ratio in Positive Torque and Power Regime	133
Figure 73: US06 Torque Split Ratio in Negative Torque and Power Regime.....	133
Figure 74: Energy Analysis for US06 Drive Cycle	134
Figure 75: UDDS Drive Cycle Speed Torque and Power Requirements	135
Figure 76: UDDS Shift Schedule.....	136
Figure 77: UDDS Drive Cycle Powertrain Power Output.....	137
Figure 78: UDDS Drive Cycle Powertrain Operating Points	138
Figure 79: Efficiency Distribution of ICE Only Operation for UDDS Drive Cycle	139
Figure 80: UDDS Optimal Control Policy Operating Efficiency Distribution.....	139
Figure 81: Torque Production Analysis of UDDS Drive Cycle	140
Figure 82: UDDS Torque Split Ratio in Positive Torque and Power Regime	141
Figure 83: UDDS Torque Split Ratio in Negative Torque and Power Regime.....	141
Figure 84: Energy Analysis for UDDS Drive Cycle	142
Figure 85: HWFET Drive Cycle Speed, Torque, and Power Requirements	144

Figure 86: HWFET Shift Schedule.....	145
Figure 87: HWFET Powertrain Power Output	146
Figure 88: HWFET Drive Cycle Powertrain Operating Points	147
Figure 89: Efficiency Distribution of ICE Only Operation for HWFET Drive Cycle	147
Figure 90: HWFET Optimal Operating Efficiency Distribution	148
Figure 91: Torque Production Analysis of HWFET Drive Cycle	149
Figure 92: HWFET Torque Split Ratio in Positive Torque and Power Regime	150
Figure 93: HWFET Torque Split Ratio in Negative Torque and Power Regime.....	150
Figure 94: Energy Analysis for HWFET Drive Cycle.....	151
Figure 95: Optimal Torque Split Ratio as a Function of Torque Request for Multiple Drive Cycles	153
Figure 96: Optimal Torque Split Ratio as a Function of Power Request for Multiple Drive Cycles	154

List of Tables

Table 1: Verification Test DP Algorithm Parameters.....	77
Table 2:ICE Only Verification Test Energy Summary.....	79
Table 3: CD Verification Test Energy Summary	82
Table 4: Fuel Only Verification Test Energy Summary	86
Table 5:Quadratic SOC Penalty Cost Function Verification Test Energy Summary	90
Table 6: Motor and Brake Penalty Cost Function Verification Test Energy Summary	93
Table 7: Overproduction Penalty Cost Function Verification Test Energy Summary	96
Table 8: Cost Function Evaluation Summary.....	97

Table 9:EMC City Information and DP Algorithm Initialization Parameters	104
Table 10: EMC City Energy Summary	114
Table 11: EMC Highway and DP Algorithm Initialization Parameters	116
Table 12: EMC Highway Energy Summary	125
Table 13: US06 Cycle Information and DP Algorithm Initialization Parameters	127
Table 14: US06 Energy Summary	134
Table 15: UDDS Drive Cycle Information and DP Algorithm Initialization Parameters	136
Table 16: UDDS Energy Summary	143
Table 17: HWFET and DP Algorithm Initialization Parameters	145
Table 18: HWFET Energy Summary	151
Table 19: DP Algorithm Powertrain Performance Results.....	152

List of Acronyms

ADAS	Advanced Driver Assistance System
A-ECMS	Adaptive Equivalent Consumption Minimization Strategy
ANN	Artificial Neural Network
AVTC	Advanced Vehicle Technology Competition
AWD	All-Wheel Drive
BEV	Battery Electric Vehicle
BSFC	Brake Specific Fuel Consumption
CAN	Controller Area Network
CAVs	Connected and Automated Vehicles
CD	Charge Depleting
CS	Charge Sustaining
CVT	Continuous Variable Transmission
DP	Dynamic Programming
eAWD	Electric All-Wheel Drive
ECMS	Equivalent Consumption Minimization Strategy
EMC	EcoCAR Mobility Challenge
EPA	Environmental Protection Agency
ERAD	Electrified Rear Axle Drive
E-REV	Extended-Range Electric Vehicle
ESS	Energy Storage System
FCO	Fuel Cut Off
GM	General Motors
HEV	Hybrid Electric Vehicles
HIL	Hardware-in-the-loop
HSC	Hybrid Supervisory Controller
HV	High Voltage
ICE	Internal Combustion Engine
MBD	Model-Based Design
MIL	Model-in-the-loop
MPC	Model Predictive Control
PCM	Propulsion Control and Modeling
PHEV	Plug-in Hybrid Electric Vehicle
PMP	Pontrygin's Minimum Principle
PSI	Propulsion Systems Integration
RESS	Rechargeable Energy Storage System
SI	Spark-Ignition
SIL	Simulation-in-the-loop
SOC	State of Charge
V2V	Vehicle to Vehicle
V2X	Vehicle to everything
WVU	West Virginia University
4S	Four Stroke

List of Symbols

<u>Symbol</u>	<u>Description</u> <u>English</u>	<u>Units</u>
A	Roadload coefficient A	-
a	Longitudinal acceleration	m/s ²
B	Roadload coefficient B	-
Battery Capacity	Energy capacity of the ESS	Whr
C	Roadload coefficient C	-
\mathcal{C}	Control space	-
$F_{roadload}$	Roadload Force	N
F_{total}	Total force	N
g	Cost-to-go matrix	-
g_{gear}	Intermediate Cost-to-go matrix	-
g_k	Cost-to-go matrix at time k	-
g_T	Cost-to-go matrix at time T	-
$GR_{diff,ICE}$	Front differential gear ratio	-
$GR_{diff,P4}$	Rear differential gear ratio	-
GR_{trans}	Transmission gear ratio	-
grav	Acceleration due to gravity (9.81)	m/s ²
i	State before transition at stage k	-
j	State after transition at stage k	-
J_k	Total path cost at arbitrary stage k	-
J_π	Total path cost of control policy π	-
J_{π^*}	Total path cost of control policy π^*	-
k	Arbitrary stage	-
m	Vehicle mass	kg
\dot{m}_{fuel}	Fuel flow rate	g/s
N_{gear}	Transmission gear number	-
P_{batt}	ESS power	W
$P_{batt,max}$	Maximum ESS power limit	W
$P_{batt,min}$	Minimum ESS power limit	W
P_{elec}	Electric power lookup table for P4 motor	W
P_{loss}	Power loss lookup table for P4 motor	W
P_{mech}	Mechanical power lookup table for P4 motor	W
Ped_{Accel}	Accelerator Pedal Position	%
r_{wheel}	Wheel radius	M
S	State space	-
SOC	State of charge of the ESS	%
SOC_{max}	Maximum SOC of the ESS	%
SOC_{min}	Minimum SOC of the ESS	%
$Stat_{TC,clutch}$	Torque converter clutch status	-
$Stat_{TC,Lock}$	Torque converter lock status	-
T	Final timestep of time horizon	-
t	Time	seconds

u_k	Control input at stage k	-
V	Longitudinal speed	m/s
\dot{V}	Longitudinal acceleration	m/s ²
$V_{downshift}$	Transmission downshift speed threshold	m/s
$V_{upshift}$	Transmission upshift speed threshold	m/s
x_k	State at arbitrary stage k	-
<u>Greek</u>		
γ	Road inclination angle	deg
η_{ICE}	ICE Efficiency	%
η_{mot}	Electric Motor Efficiency	%
η_{total}	Total System Fuel Efficiency	%
η_{trans}	Transmission Efficiency	%
π	Control policy	-
π^*	Optimal control policy	-
$\tau_{c,ICE}$	ICE component torque	Nm
$\tau_{c,ICE,max}$	Maximum ICE component torque	Nm
$\tau_{c,ICE,min}$	Minimum ICE component torque	Nm
$\tau_{c,ICE,min run}$	Minimum running ICE component torque	Nm
$\tau_{c,ICE,min FCO}$	Fuel cut off ICE component torque	Nm
$\tau_{c,mot}$	P4 motor component torque	Nm
$\tau_{c,mot,max}$	Maximum P4 motor component torque	Nm
$\tau_{c,mot,min}$	Minimum P4 motor component torque	Nm
$\tau_{c,remain}$	Remaining torque in terms of ICE component torque	Nm
$\tau_{w,brk}$	Mechanical brake wheel torque	Nm
$\tau_{w,ICE}$	ICE wheel torque	Nm
$\tau_{w,mot}$	Electric motor wheel torque	Nm
$\tau_{w,remain}$	Remaining wheel torque	Nm
$\tau_{w,req}$	Drive cycle wheel torque requirement	Nm
ϕ_{TC}	Torque converter speed ratio	-
ψ_{TC}	Torque converter torque ratio	-
ω_{ICE}	ICE component speed	RPM
$\omega_{ICE,idle}$	ICE idle speed	RPM
$\omega_{ICE,max}$	Maximum ICE speed	RPM
$\omega_{ICE,min}$	Minimum ICE speed	RPM
ω_{mot}	P4 motor speed	RPM
$\omega_{mot,max}$	Maximum P4 motor speed	RPM
$\omega_{mot,min}$	Minimum P4 motor speed	RPM
ω_{turb}	Torque converter turbine speed	RPM
ω_{wheel}	Wheel speed	RPM

1.0 Introduction

The objective of this research is to develop a hybrid electric vehicle model that can be used to conduct a benchmark analysis by implementing a dynamic programming algorithm to identify the optimal control policy for a hybrid electric drivetrain on prescribed drive cycles. The results from the benchmark analysis can be used to compare and improve existing hybrid supervisory control strategies by providing a frame of reference to the optimal performance of the hybrid drivetrain.

In this work, the motivation behind hybrid powertrain technology is discussed by providing the justification behind the transition from conventional vehicle that rely solely on internal combustion engines, to the hybrid electric vehicles that utilize electric motors and batteries to supplement torque production. The background of hybrid vehicle technology and the range of technology available to support these powertrains is discussed along with an overview of the work done by the West Virginia University EcoCAR Team to develop a hybridized 2019 Chevrolet Blazer to participate in the premier North American Advanced Vehicle Technology Competition (AVTC) the EcoCAR Mobility Challenge (EMC). Part of the development of the hybrid vehicle includes extensive hybrid supervisory control implementation which is the primary inspiration for the contents of this work.

An overview of energy management strategies is discussed to define the options that are available to control hybrid vehicles and provide a frame of reference for how the benchmark analysis results are useful in the design of such control systems. The fundamentals of the mathematical models used by the EcoCAR team are discussed in depth to provide the necessary background to understand the models developed for the benchmark analysis in this work. Each subsystem is examined and the available methods for balancing accuracy and complexity are investigated. The

structure, logic and justification of the benchmarking model are discussed in detail such that this work can be easily reproduced to model nearly any parallel hybrid electric vehicle with the appropriate component data. This work provides the essential background for the development of a dynamic programming algorithm and associated powertrain models such that highly efficient control systems can be developed to maximize the usability a given hybrid powertrain.

The dynamic programming algorithm is discussed in depth along with the mathematical models used for both the electric and conventional powertrain to determine the optimal control policies for multiple drive cycles. The methods to verify and validate the benchmarking model to ensure that the algorithm successfully operates and determines solutions that adequately represent control actions taken by the hybrid 2019 Chevrolet Blazer. The results for multiple drive cycles are analyzed in depth to identify the trend of control actions the optimal control policy constitutes and discuss the motivations as to why the control policy takes certain actions.

1.1 Hybrid Electric Vehicles

From the original prototype motor vehicle developed by Carl Benz in 1886 to the widely produced modern-day automobiles, the internal combustion engine (ICE) has been used as the primary generator of propulsive power. Under operation, the range of torques and speeds that correspond to the maximum efficiency of the ICE is narrow, and in most consumer automobiles, the ICE frequently operates outside of this envelope to honor the torque and power requests of the driver. The increasingly large number of ICE operating inefficiently across the world has begun causing consequential problems for the environment and hydrocarbon resource supplies. Deteriorating air quality, global warming issues and depleting petroleum resources have forced regulatory entities to implement ever more strict emissions regulations for automotive manufacturers. Rising to the

challenge to meet these regulations, innovation in the automotive design field has influenced hybrid electric vehicles (HEVs) popularity more than ever across the world. HEVs utilize a combination of conventional vehicle components, such as an engine and transmission, and electric vehicle components, such as an electric motor and battery pack, to provide propulsive power to the wheels of the vehicle. By electrifying the powertrain, higher fuel efficiency and reduction in emissions can be achieved when compared to conventional vehicles [1].

HEVs stand as the middle ground between conventional vehicles and fully electric vehicles by exhibiting many of the attractive features of either powertrain while improving on their shortcomings. Most notably, HEVs boast higher fuel efficiency and reduced emissions compared to conventional vehicles but are not restricted to long charging periods that would be experienced with battery electric vehicles (BEVs). These features of HEVs are achieved by two primary operational strategies: regenerative braking and engine operation optimization. Regenerative braking takes advantage of the energy that is traditionally lost as heat by friction brakes and instead uses deceleration events as an opportunity to capture electrical energy by utilizing the electric motors to reduce vehicle speed. The energy captured in regenerative braking events is stored in a high voltage (HV) battery pack, often referred to as the energy storage system (ESS), for later use.

Modern day engines operate around 40% efficiency at peak performance while most electric motors are capable of over 90% efficiency for most of their operational range. Considering the operation of an ICE, the efficiency envelope as a function of torque and speed, referred to as an engine map, will generally be populated with operational points that are solely dependent on the request of the driver. With the additional source of torque in an HEV, the electric motor, the operational points of the engine can be shifted to a more efficient point for a given driver demand as shown in the highly simplified example in Figure 1.

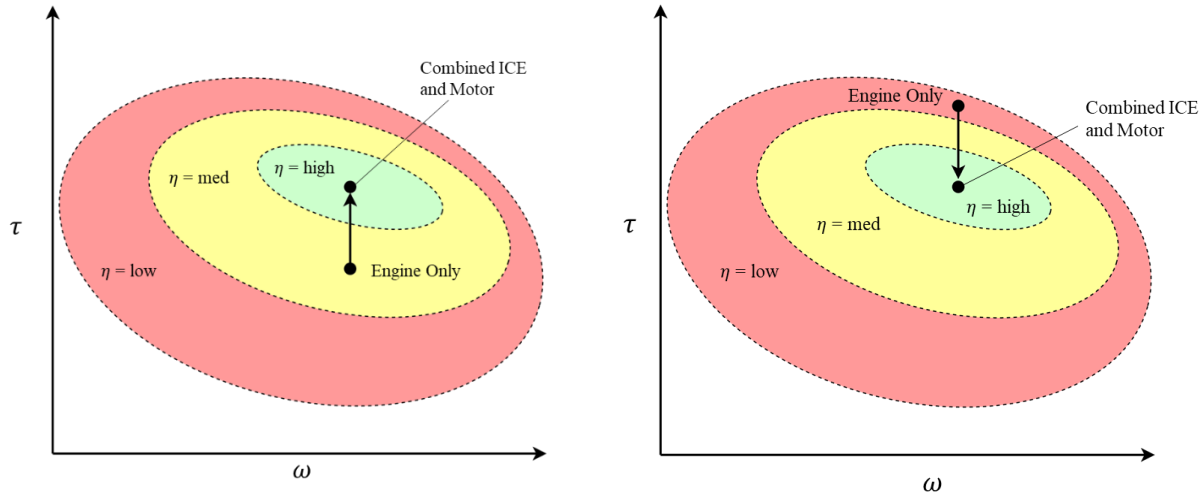


Figure 1: Example of Engine Operational Point Shifted with Electric Motor (Left: Opportunity Charging) (Right: Motor Assist)

In the example on the left, the engine only operational point would fall into the medium efficiency region. In a hybrid powertrain, the electric motor could be used to command negative torque, requiring the engine to produce more torque than the driver requested utilizing a strategy called opportunity charging. This would shift the operational point of the engine up into the high efficiency range. Similarly, in the example on the right, the driver torque request exceeds the efficient region of operation and the engine only operating point would fall in the low efficiency region. With electric motor assist, the ICE could produce less torque and operate in its most efficient regions and allow the electric motor to provide positive propulsive torque to meet the driver command. Depending on the HEV architecture, the use of both an electric motor and ICE, the operational point can be shifted up and down along the torque axis and also left and right along the speed axis.

There are additional benefits for plug-in hybrid electric vehicles (PHEVs) where the vehicle operator has the ability to charge the ESS from the grid, which allows for the propulsive power of the vehicle to be generated in a highly efficient plant rather than the inefficient ICE. Because of this, PHEVs generally have a larger ESS and more powerful electric motors that make the vehicle

capable of operating in a completely electric mode known as charge depleting (CD) operation or engine supporting mode known as charge sustaining (CS) operation. Traditional HEVs have smaller battery packs that are specifically designed to charge and discharge quickly as well as less powerful electric motors that are meant to assist engine operation in CS mode rather than support a fully electric or CD operational mode.

1.2 Hybrid Electric Vehicle Architectures

Conventional vehicles consist of an ICE coupled to the wheels through a transmission. Generally, a dry clutch or hydrodynamic torque converter decouples the engine from the transmission when needed. In these vehicles, there is a direct relationship between driver commands and the engine or friction brakes. HEVs introduce electric motors as additional torque sources into the powertrain. The location and capabilities of electric motors in the vehicles make up a wide spectrum of electrified vehicle architectures.

1.2.1 Classifications

Most vehicles on the road today can be categorized as one of the following classifications: conventional vehicle, micro hybrid, mild hybrid, full hybrid, plug-in hybrid, or battery electric vehicle. Micro hybrids incorporate engine start/stop capabilities into vehicles with non-electrified drivetrains. These systems allow the ICE to shut down and restart thus reducing fuel consumption and emissions. Micro hybrids typically do not employ the use of an ESS but instead utilize higher performance starter motors than conventional vehicles. Mild hybrids generally couple an ICE and electric motor that allow the engine to be turned off whenever possible and employ both power assist and regenerative braking. A mild hybrid will be equipped with a full hybrid powertrain (ESS, inverter, electric motor) however do not employ an exclusively electric-only or CD mode [2]. Full

hybrid electric vehicles typically have higher power motors and larger capacity battery packs when compared to mild hybrids and may employ a CD operating mode. It is important to note that HEV classifications should be viewed as a spectrum of capabilities rather than strict buckets that every vehicle can be sorted into. For example, a vehicle that exhibits most characteristics of a full hybrid but may not necessarily employ engine start/stop functionality while stopped at a traffic light.

1.2.2 Configurations

HEVs can be outfitted with electric motors in various configurations that vastly change the necessary control strategy as well as the overall performance of the vehicle. There are various types of HEV configurations but each type generally traces back to the following two configurations: series and parallel [2].

In a series configuration, the engine is completely mechanically decoupled from the wheels of the vehicle and thus the electric motor(s) are the only means for producing torque to the wheels of the vehicle. The output shaft of the engine is generally coupled to a separate electric motor that is exclusively used to generate energy for the ESS. Series HEVs typically employ large battery packs and powerful electric motors similar to what would be present in a BEV, with a small, fuel efficient engine. The control strategy for the engine is relatively simple as the engine can be run at whatever speed and torque the system deems to be either the most efficient or necessary for the current conditions. A major drawback of the series HEV architecture is that these architectures exhibit increased weight due to the full BEV drivetrain plus engine and generator motor. [3]

Parallel HEVs use the engine and electric motors together to provide propulsive power to the wheels of the vehicle. The position of the electric motor in a parallel configuration is indicated by

a position number P0-P4. An overview of the location of electric machines in a parallel hybrid electric vehicle are shown in Figure 2.

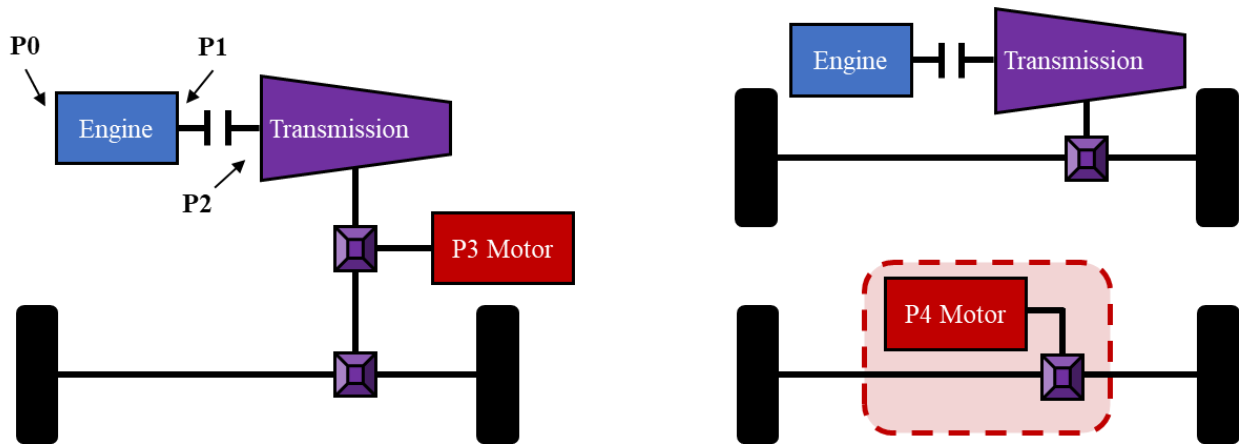


Figure 2: Electric Motor Locations for a Parallel HEV

A P0 electric motor is mounted on the front-end accessory drive of the engine and is connected to the engine via a belt. P1 electric motors are connected directly with the crankshaft, pre-clutch, of the ICE. A P2 electric motor is located post-clutch on the engine but pre-transmission. Electric motors in a P3 configuration are post-transmission and connected to the driveshaft of the vehicle through a gearbox. Finally, a P4 electric motor drives an independent axle and is only coupled to the conventional drivetrain through the wheels.

Series architectures have the advantage of high efficiency as propulsive power is supplied through energy efficient electric motors and the freedom to select engine speed and load to keep the ICE operating at peak efficiency. These configurations are preferable for urban stop and go driving conditions; however, a series architecture requires two energy conversions, mechanical to electrical and electrical to mechanical, which introduces losses. In some cases these losses are so significant that the vehicle consumes more fuel than a conventional vehicle, specifically during highway driving conditions [2]. Parallel architectures, however, lose the ability to always operate the ICE efficiently as the engine operating point is at the discretion of the driver. Parallel

architectures are generally more efficient in highway driving conditions but less efficient in city driving conditions. Some HEVs combine series and parallel configurations by utilizing power split devices and clutches to allow the use of the configuration that is best suited for the operating condition of the vehicle. These configurations are referred to as series-parallel configurations and take advantage of the benefits of both configurations.

1.3 EcoCAR Mobility Challenge

The EcoCAR Mobility Challenge (EMC) is the current Advanced Vehicle Technology Competition (AVTC) sponsored by Argonne National Laboratories, MathWorks, and General Motors (GM). AVTCs started in 1988 with the Methanol Marathon [4]. The EMC is the premier automotive student design competition between 10 North American universities. Each team was provided a 2019 Chevrolet Blazer RS with the GM 3.6L LGX in an all-wheel drive (AWD) conventional vehicle configuration. Spanning over four years, teams are tasked to convert their Blazer into a HEV. The main goals of the competition are to improve fuel economy, reduce emission, maintain stock vehicle performance, and improve vehicle automation features. A final competition is held at the end of each year, throughout the four-year competition, where teams have the opportunity to present their work and show off their vehicle in dynamic vehicle testing events.

There are three technical swimlanes that make up the majority of work within the EMC: propulsion systems integration (PSI), propulsion controls and modeling (PCM), and connected and automated vehicles (CAVs). The West Virginia University (WVU) team has established sub-teams made up of engineering students from various backgrounds such as mechanical engineering, electrical engineering, computer science, and computer engineering. The PSI team is responsible for

mechanical and electrical integration of vehicle systems such as the engine, transmission, ESS, electric motor, etc. The work of the PSI team extends to the design and integration all supporting systems for these components including the mounting, thermal, and electrical systems. The PCM team is responsible for vehicle architecture modeling as well as vehicle controls development and implementation. The models developed by the PCM team are used to drive architecture selection decisions as well as provide various simulation environments such as model in the loop (MIL) or hardware in the loop (HIL) where vehicle controls can be developed and tested. The vehicle controls developed by the PCM team include component functionality, energy management strategies, diagnostic systems, and fault mitigation systems. The CAVs team is responsible for designing and implementing advanced driver assistance systems (ADAS) with vehicle to vehicle (V2V) and vehicle to everything (V2X) communication capabilities. These systems are utilized to develop advanced vehicle features such as adaptive cruise control and lane keep assist.

Over the course of the competition, the team must work together to produce a fully functional, road worthy, HEV. The competition facilitates a team oriented and results-driven environment where students learn skills that are directly useful in industry-level jobs.

1.4 Team Selected Architecture

During Year 1 of the EMC, the WVU team conducted historical EcoCAR architecture research to learn about the challenges of integration and control of various hybrid architectures. Along with this research, the PCM team conducted analysis in the MIL environment to determine the size requirements for major powertrain components such as the engine and electric motor and to provide preliminary powertrain performance results based on potentially donated components. Based on these analysis as well as various rule sets from the competition, the team selected a P4

HEV architecture for the 2019 Chevrolet Blazer competition vehicle due to fuel economy modeling results as well as integration considerations given the available timeframe.

The WVU Team selected the GM 4-cylinder 2.5L LCV engine rated for a maximum torque of 255 Nm and a maximum power of 148 kW. The engine is paired to the 9-speed M3D 9T50 transmission with accumulator. The HV battery pack or ESS selected is the GM HEV4 battery pack with a total energy storage capacity of 1.5 kWh with a nominal voltage of 300V and peak power of 50 kW. The P4 electric motor came paired with an inverter from Magna Powertrain in a system called the electrified rear axle drive (ERAD), also referred to as the electric all-wheel drive (eAWD) system. The ERAD system is a production powertrain component in the European Volvo V60 hybrid which is a similar size vehicle as the Blazer. The ERAD system has a peak power of 50 kW, peak torque of 200 Nm, maximum continuous power of 20 kW, maximum continuous torque of 90 Nm, and a maximum motor speed of 12000 rpm. The gear ratio of the integrated differential gearbox is 9.17. An overview of the WVU team architecture is shown in Figure 3.

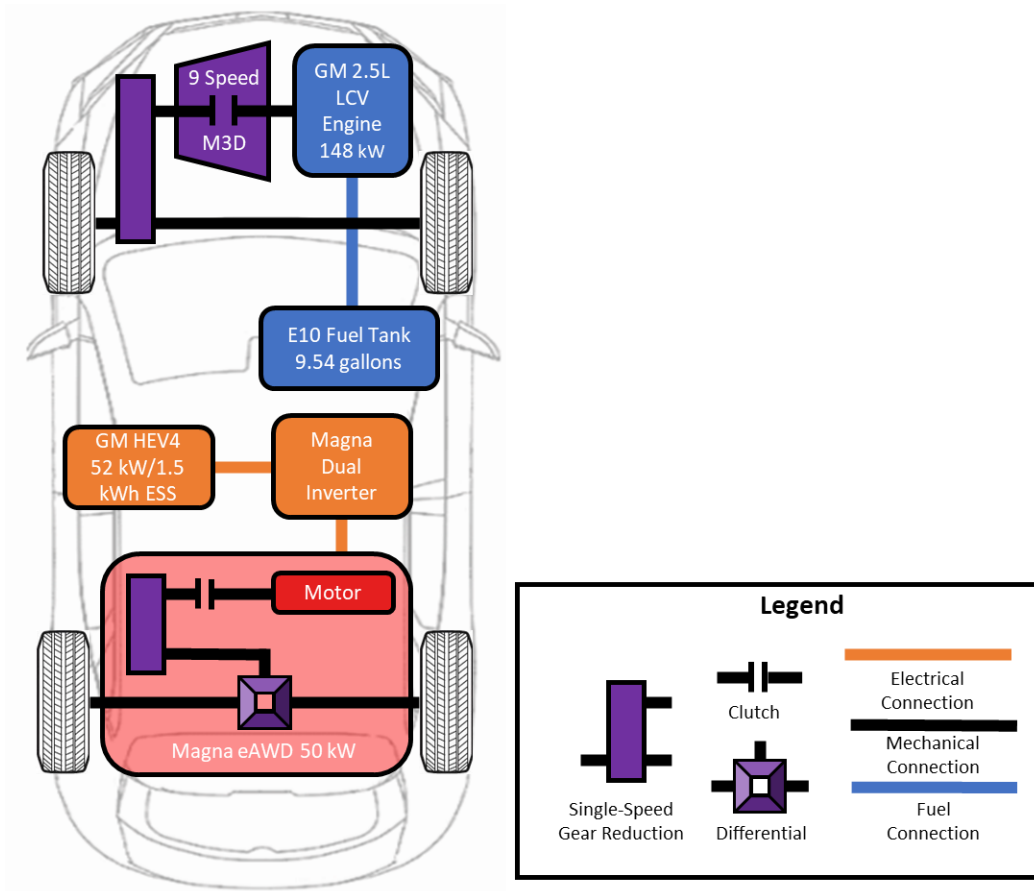


Figure 3: WVU Team Competition Vehicle Architecture

The P4 architecture supports three operating modes: FWD with opportunity charging, FWD with regenerative braking, and AWD. In FWD with opportunity charging, the engine produces excess torque to the front axle while the P4 traction motor “drags” the rear axle by producing negative torque. When producing negative torque, the electric motor is spun thus generating power that is stored in the ESS. In FWD with regenerative braking, the engine is supplying all of the positive propulsive torque for the vehicle. In situations where the driver requests a deceleration event, the electric motor is used to produce negative torque to meet the braking needs of the vehicle. This negative torque produced by the electric motor generates power that can be stored in the ESS. It is important to note that the amount of energy captured in a regenerative braking scenario is greater if the vehicle is slowed gradually, i.e., small negative torque command. In AWD both the electric

motor and engine produce positive torque to meet the needs of the driver. The operating mode of the vehicle can transition rapidly to meet the current driving conditions. Smooth transitions between these operating modes are a major consideration of the PCM team with respect to the ride quality of the competition vehicle.

HEVs generally operate in two operational strategies: CS and CD as mentioned in Section 1.2.1. A HEV operating in CD mode favors the electric powertrain over the ICE or may run in a purely electric mode until a low threshold of ESS state of charge (SOC) is reached. At this SOC low threshold the vehicle will begin operating in a CS mode where the vehicle will maintain the SOC at a specified setpoint and avoid large variations from this setpoint. CD modes are commonly used in PHEVs as these vehicles generally have ESSs with a larger capacity, while CS modes are necessary for any HEV to ensure that the SOC of the ESS does not reach critically low levels. It is also important to note that for fuel economy results of HEVs, the vehicle is evaluated on a known drive cycle with the starting and ending SOC within a specified bound. This is commonly referred to as CS fuel economy.

In the WVU team competition vehicle, the engine (148 kW) is capable of producing nearly three times the power compared to the electric powertrain (50 kW). This limits the electric powertrain's ability to meet driver demands without the help of the engine. In addition, the selected ESS has a usable capacity of only 1.5 kWh with simulation results developed by the PCM team showing an all-electric range of only about 2 miles [5]. For these reasons, the team did not implement a CD mode in the 2019 Chevrolet Blazer competition vehicle. Based on the available power from the electric powertrain, the team competition vehicle would operate exclusively in a CS mode with the electric motor augmenting the operation of the engine. Specifically, the electric powertrain would be used to shift the operating point of the engine to more efficient regions.

1.5 HEV Supervisory Control and Energy Management Strategies

When considering the control of the HEVs powertrain components, it is easy to assume that the problem is as simple as utilizing the electric motors as much as possible due to their high operating efficiencies. For HEVs equipped with CD modes, this would be the case. The strategy is to use the electric powertrain until the SOC reaches the low threshold. However, from a CS point of view, the control problem becomes far more complicated when attempting to achieve improved fuel economy over a conventional vehicle. An effective energy management strategy is essential to ensuring the efficient operation of the vehicle [2].

To understand the energy management problem, first consider a conventional vehicle powertrain. The ICE operates by converting chemical energy into mechanical energy to generate all of the energy required for the vehicle operation. The energy generated in the engine drives the accessories, driveline components, and ultimately the vehicle. For a given driver input and current driving conditions such as vehicle speed and grade, the operating condition of the engine (speed and torque) is determined by a single degree of freedom: transmission gear ratio. The energy management strategy for the conventional vehicle is the choice of this ratio. In a HEV, the additional torque path from the electric powertrain components creates a new degree of freedom, the ratio of energy supplied by the engine and electric motor, commonly referred to as the torque split ratio. Additionally, the electric motor can replace or supplement the mechanical brakes for deceleration requests from the driver. The additional freedom afforded by the electric components necessitates the use of an energy management strategy to improve the overall efficiency of the vehicle [2].

To implement the energy management strategy, the powertrain is commanded using a supervisory control algorithm. The flowchart of a general supervisory control system can be seen in Figure 4.

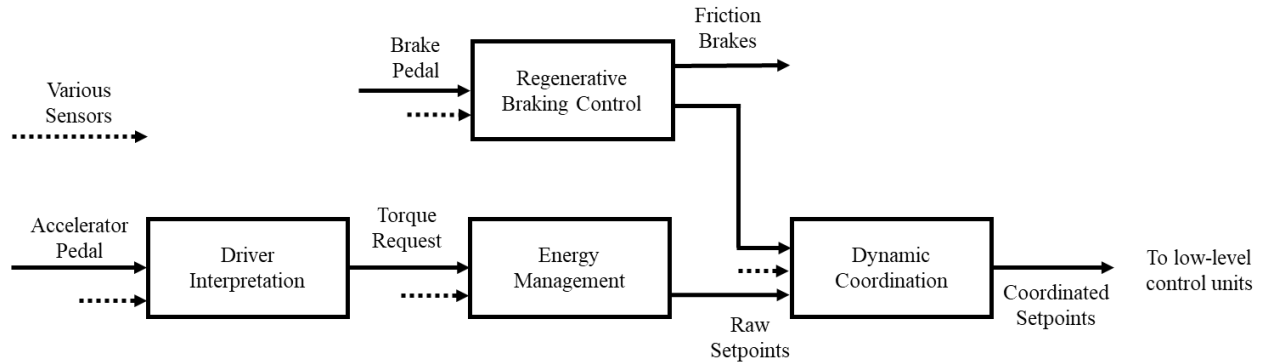


Figure 4: Flowchart of an HEV Supervisory Controller [6]

When the accelerator pedal is pressed, the first step in the torque control structure is the driver interpretation function that yields the total torque required at the wheels. The torque request is typically evaluated as a look-up table as a function of vehicle speed. In general, 100% pedal would correspond to the maximum torque available in the powertrain while 0% pedal would result in creep torque or engine braking depending on the speed of the vehicle. These look-up tables are commonly referred to as pedal maps and generally have a nonlinear relationship between 0% and 100% pedal positions. The torque request is then sent to the energy management function which decides where torque should be produced following a specified strategy. Popular energy management strategies typically fall under the designation of heuristic or optimal control strategies. These energy management strategies are discussed in further detail in Section 2.0 of this thesis. The output of the energy management system are the raw setpoints for powertrain components.

The inclusion of regenerative braking capabilities necessitates an additional torque path in the supervisory control strategy. When the brake pedal is pressed, the regenerative braking control function governs the use of the electric motor and friction brakes to reduce the vehicles speed. The

functionality of this system includes functional considerations such as not utilizing regenerative braking if SOC is too high as well as safety critical considerations of ensuring the driver is receiving enough brake force to slow the vehicle. It is important to note that any commands to the electric motor from the regenerative braking control function go through the dynamic coordination function. Coordinating the torque requests of powertrain components for both positive and negative torque requests is critical not only for driver comfort, but also for safe operation of powertrain components [6]. A critical aspect of this coordination is torque shaping where the component torque requests may be passed through various filters to remove harmful transients. For example, as the vehicle is driven down the road at very low SOC, the driver abruptly applies 100% brake pedal. The output of the regenerative braking control function may be to command the maximum amount of torque from the electric motor; however, this command may damage the motor by switching quickly from positive torque to maximum negative torque. In this example, the torque shaping functionality would smooth the transition in torque values for the electric motor to prevent damage. After passing the component torque commands through the dynamic coordination function, the supervisory controller passes the coordinated setpoints to the component level controllers in the vehicle. This transfer of information takes place on electrical connections called a controller area network (CAN) bus. It is common for vehicles to have multiple CAN busses that are used to transfer sensor readings, diagnostic information, powertrain control information and any other information that may be relevant to vehicle operation between electronic control units.

1.6 Importance of Benchmark Analysis for HEVs Powertrain Operation

With the rise of HEVs in automotive markets across the world, each boasting improved fuel economy over their conventional counterparts, the question arises of the limit on how much

improvement is possible with these powertrains. In fact, the efficient control and implementation of HEV energy management strategies pose a global optimization problem. Many energy management strategies generally employ some level of optimization to achieve improvements in fuel economy. These strategies must often balance the level of optimization achieved with the available computation resources. Heuristic energy management strategies generally obtain optimization information offline which is then employed as rule based or fuzzy logic strategies that require minimal computational resources to make informed decisions. Because of the offline nature of the optimization, these energy management strategies must optimize vehicle operation over a known drive cycle. Because of this, heuristic energy management strategies typically operate significantly worse than the global optimal fuel efficiency when exposed to the unknown conditions of normal driving. Several online optimization strategies exist that perform simplified optimization problems based on instantaneous vehicle information. These strategies offer significant performance improvement over heuristic approaches but at the cost of computational load. Often, a trade-off is made between the complexity of the optimization problem and the computational resources necessary for finding the optimal solution. Regardless of the technique used to employ an energy management strategy, it can be stated that the goal is always to improve over a similar conventional powertrain. This goal represents the low threshold of achievement for a HEV powertrain. The upper threshold would be the global optimal solution.

Based on the tradeoff nature of online optimal control algorithms between complexity and computation resources, it is clear that calculating the global optimal solution online is not feasible with current technology. However, by calculating the global optimal solution offline and using that information to design and calibrate online algorithms affords the ability to assess their performance against the upper threshold of achievement. For example, when designing an energy

management system, it is useful to know that a particular algorithm performed 30% better than a conventional counterpart, but if the algorithm is over 90% error from the global optimal solution, it becomes clear that additional improvements can be made. Dynamic Programming (DP) offers this ability to determine the global optimal solution for a given HEV architecture over a specified drive cycle. [7], [8], [9]

2.0 Overview of HEV Energy Management Strategies

Several families of energy management strategies have been investigated in existing literature. These strategies generally follow one of two trends, heuristic-based and model-based optimization methods [2]. These two trends are representative of the use of optimal control theory in the design of the controller or not. Heuristic energy management strategies are primarily based on intuition and logical relationships between variables and thus little optimization occurs. Heuristic control is popular among automotive manufactures and is widely adopted in modern HEVs. Model-based optimization methods, also known as optimal control, make use of optimal control theory to derive the controller. Optimal control strategies are currently subject to research and are gradually being introduced in the industry [6]. Heuristic and optimal control strategies have two distinct subgroups as shown in Figure 5.

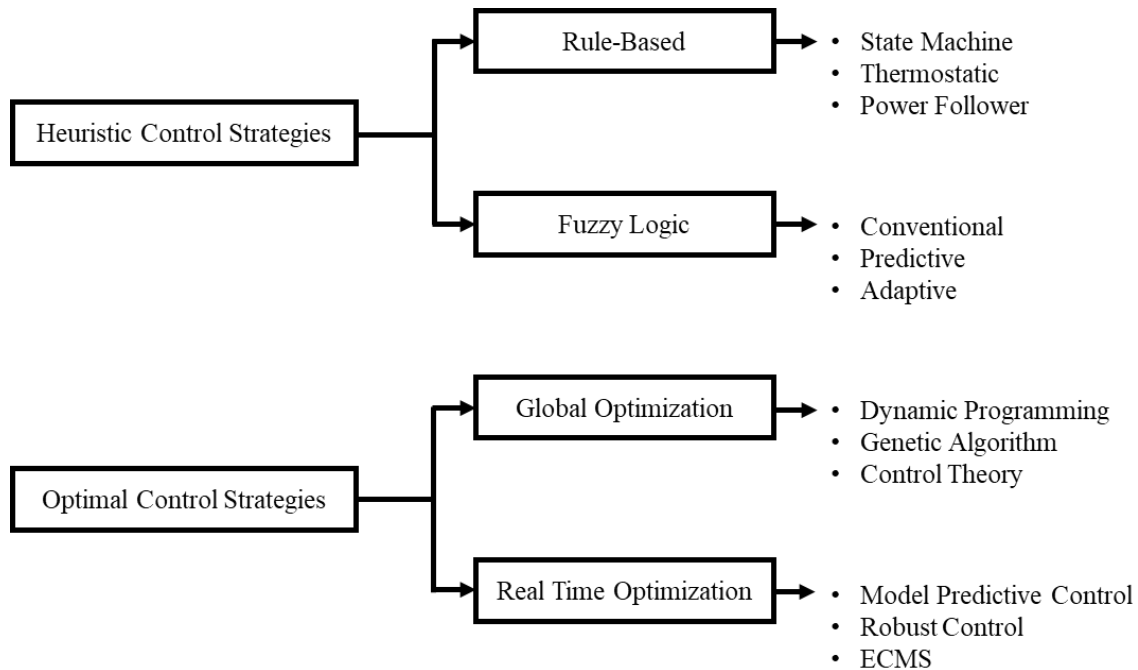


Figure 5: Control Strategy Classification [10]

2.1 Heuristic Control Methods

The most attractive characteristic of heuristic control methods is their effectiveness in real-time implementation. Since these strategies rely on predefined sets of rules and logic rather than minimization or optimization, heuristic controllers require little computational resources to make decisions. These types of controllers generally fall into two categories: rule-based and fuzzy logic [2].

2.1.1 Rule-Based

Rule based control is the traditional control methodology used in the automotive industry typically consisting of “if-then” and “switch” logic based on simulation data, intuition, or some other set of prescribed behavior based on constraints and conditions [10]. Bowles, Peng, and Zhang [11] discuss a PHEV with a continuous variable transmission (CVT) that operates with specific rules that govern the vehicles operating mode based on operational constraints and predetermined

efficiency calculations. The operational constraints are based on the stall speed and maximum torque available from the ICE. Additionally, Bowles *et al.* performed an offline optimization to generate lookup tables for CVT ratio and the engine throttle that correspond with fuel economy minimization. Based on these results, the group found that the engine operated inefficiently at low speeds so the motor would be used in these scenarios. As speed increased, the engine would be used to produce torque to meet the driver demand but if the demand exceeded the capabilities of the engine, the electric motor would assist. When SOC is low an additional recharge load was added to the engine at a specified power. Butler, Ehsani and Kamath [12] researched the implementation of two rule-based control strategies on a HEV architecture. The first operates such that the ICE runs at a constant fuel throttle angle and the electric machine makes up the difference between the torque requested by the driver and the torque produced by the ICE while the second allows the ICE to run over its entire speed range and makes the ICE throttle angle a function of speed to meet the steady-state road load. The general principal behind each strategy is that the electric motor provides power for propulsion during the transients and the ICE provides propulsion during a cruising condition. Rule-based energy management systems have been widely researched and shown to be practical and successfully implemented to control hybrid powertrains [13],[14],[15]. These strategies, however, require careful tuning to specific driving scenarios to achieve efficient performance. This major drawback leads to reduced performance in application of these control systems in consumer vehicles.

2.1.2 Fuzzy Logic

Fuzzy set theory contrasts conventional binary logic of “does” or “does not” by implementing varying degrees of belonging to a set. The use of fuzzy set theory to control systems is referred to as fuzzy logic. Fuzzy logic has been implemented in a multitude of fields where classic control

has long dominated such as home appliances, automobiles and chemical processes. Fuzzy control gives users the ability to implement expert knowledge through automation, provide robust nonlinear control and reduce development and maintenance time [16].

Hajimiri and Salmasi [17] implemented a fuzzy logic energy management system to a HEV with embedded features of predictive control to improve fuel economy and emission. The implementation of this controller showed improvement over a power follower rule-based approach. In [18], Xu *et al.* applied a fuzzy control strategy to a parallel HEV energy management system. Compared to a rule-based control strategy, the fuzzy logic controller showed more engine operation point distribution in the high load area where the efficiency is high. Many other examples of fuzzy logic use in powertrain control are available in literature [19],[20],[21] that show improvements over traditional rule-based control strategies. It is important to note that these control strategies are sub-optimal as any optimization is conducted offline and used to design the governing “if-then” statements of the fuzzy logic controllers.

2.2 Optimal Control Methods

Energy management of a HEV can be posed as an optimization problem over a finite time horizon whose solution can be found from optimal control theory. The methods used are aimed at finding a control law for a given system such that a specific optimality criterion, usually defined as an integral performance index, is achieved [2]. In an optimal strategy, an appropriate cost function is created which is minimized at each time step. There are two main areas of optimal control methods: real-time optimization and global optimization.

2.2.1 Real-Time Optimization: Equivalent Consumption Minimization Strategy

In real-time optimization, the cost function is minimized at each timestep of the online controller. In these controllers, simple mathematical models of the system are generally used to keep the execution time within a short time window. Potentially the most popular real-time optimal control strategy is Equivalent Consumption Minimization Strategy (ECMS) which operates on the premise that in a CS HEV the differences between initial and final SOC is small. The ESS is as an energy buffer that if discharged will need to be charged at some point in the future. This creates a specific cost for utilizing electrical stored energy and draws an equivalence between using a certain quantity fuel or stored electrical energy [2].

ECMS has been successfully implemented as an effective energy management strategy in literature. In [22], Sciarretta, Back, and Guzzella showed that ECMS has the potential to reduce fuel consumption by up to 30% of the conventional vehicle. In urban driving conditions, Sciarretta *et al.* showed that the potential fuel savings to be on the order of 50% over a conventional vehicle.

An ECMS algorithm requires careful tuning of its equivalence factor to enable the fuel savings shown by Sciarretta *et al.* To expand the flexibility of ECMS, three main methods have been examined to create an adaptive-ECMS (A-ECMS) algorithm with a dynamically changing equivalence factor: drive cycle prediction, drive pattern recognition, and SOC feedback. These algorithms employ various novel tools such as artificial neural networks (ANN) to assist in the determination of this adaptive equivalence factor [9]. Most A-ECMS work is still conducted in literature and has not yet been applied to consumer HEVs. [23]

2.2.2 Global Optimization: Dynamic Programming

Achieving a global optimal solution is generally directly correlated with highly complex and computationally expensive numerical solutions. One of the most popular methods for solving the optimal control problems for HEVs is Dynamic Programming (DP). DP is based on Bellman's principal of optimality which reduces a multi-step decision making problem into a series of single-step problems. These single step problems may be solved either forward in time or backwards from the last step to the first. The goal of DP is to minimize an incrementally increasing cost function at each step. After determining the control actions that lead to the minimum cost, the control actions can be observed to determine specific control policies that may be implemented in an online control strategy. DP offers dramatically reduced computation time compared to brute force methods as it only searches over admissible state or control values. It is important to note however, DP still requires the storage of all valid state transition costs [10].

DP algorithms must accomplish two main functions. The algorithm must calculate and store the cost-to-go from every feasible state to every other admissible state for each timestep through the use of a state transition function and cost function. The algorithm must then use this information to determine the set control actions, constituted by the state transition function, that minimizes the overall cost from the first timestep to the final timestep.

DP must be implemented offline as the knowledge of the full drive cycle *a priori* is required. Because of this, DP can be used to determine performance benchmarks. The results of the DP algorithm are used to determine appropriate control policies and the maximum efficiency a specific powertrain would be capable of achieving. Lin, Peng, Grizzle and Kang [24] implemented DP to solve the control optimization problem of a hybrid electric truck for both fuel economy and

emissions considerations. In this work, the cost function considered the minimization of fuel consumption and emissions. Additionally, the cost function is augmented with a quadratic penalty factor to constrain the optimization algorithm to CS performance. Lin *et al.* were able to reduce the performance gap from a rule-based strategy to the DP results by 50-70% by implementing control policies determined from the DP analysis.

DP algorithms typically require a highly simplified vehicle model due to the computational resources required to calculate the cost-to-go matrices. Wang and Lukic [25] discuss a method for reducing the fidelity of a model to be used appropriately in a DP algorithm. This method is applied in a case study to evaluate the Toyota Prius with DP. After simplifying the Toyota Prius model available in Advisor2002, Wang *et al.* performed validation testing of the simplified vehicle model against the Advisor2002 model. From this analysis, an error of ~2.0 % was reported between the high fidelity Advisor2002 model and the simplified vehicle model. Wang and Lukic then executed the DP algorithm to analyze the hybrid system for optimal performance. From the results of the DP analysis, a rule-based controller was developed that yielded a 27% increase in overall efficiency.

The benefits of DP have been assessed for various HEV architectures. Dokuyucu and Cakmakci [26] applied DP for a parallel HEV where they utilized a DP algorithm to identify optimal control performance for an energy management system and traction control system both as separate systems as well as together in a concurrent system. With the knowledge of the optimal control strategy of each system separately, the researchers were able to conclude that the concurrent controller design results were promising and worthy of future work. Patil, Filpi and Fathy [27] investigated its benefits for series HEV controller design. Patil *et al.* utilized DP as a benchmark to interpret the results of two online heuristic controllers more appropriately. All of the above

noted the benefits of extracting data from DP to improve and assess online control strategies. [9] [8] [7]

3.0 Hybrid Vehicle Modeling Fundamentals

Vehicle operation fundamentals mathematically describe vehicle behavior based on the general principals of mechanics. A vehicle consists of thousands of individual components that must effectively and reliably work together. To describe the behavior of a vehicle fully would require extensive and sophisticated mathematical knowledge as well as immense computational power. Although technically possible, the derivation of a complete vehicle model is not necessary to extract the benefits of a mathematical model.

In this section, the concepts and strategies used in vehicle powertrain modeling, performance analysis, and/or control system development are discussed. Various examples are included from the WVU EcoCAR Mobility Challenge team's propulsion system controls and modeling activities. This information is included to provide a background for the activities of the propulsion control and modeling (PCM) sub-team of the WVU EcoCAR team and for this thesis.

3.1 Model Based Design and Core Concepts

Model-based design (MBD) has transformed the way engineers and scientists work by moving design tasks from the lab and field to the desktop. Modeling and simulation play a key part in the design of HEVs by affording engineers the ability to rapidly develop models for various drivetrains and components. These models may be of very high fidelity and accurately model communication interfaces between components or may be low fidelity and only model relevant powertrain

components. The flexibility of various types, structures and fidelity levels reinforces the benefit of adopting MBD for HEVs.

The MBD methodology typically follows a system engineering process called the v-model shown in Figure 6.

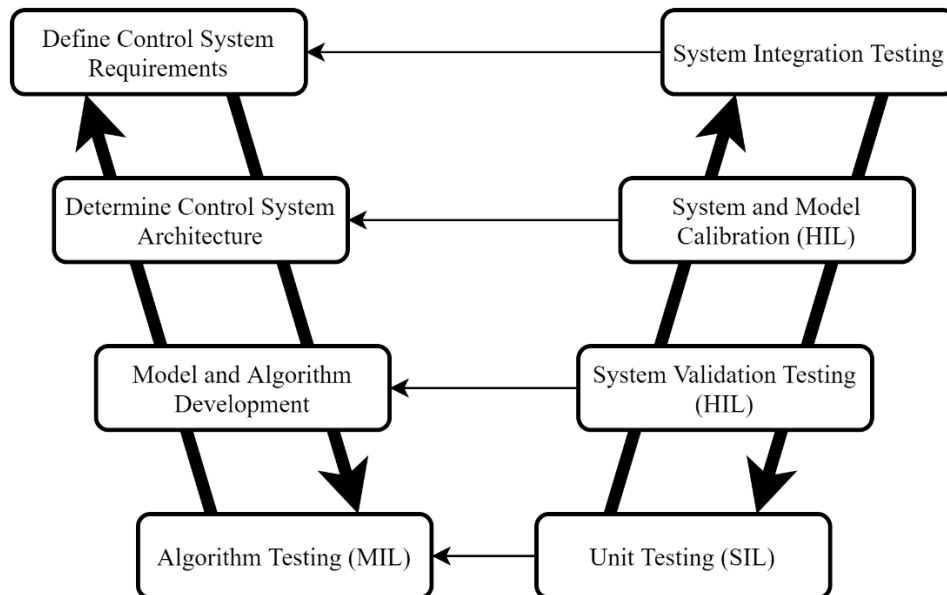


Figure 6: V-Model of the Model Based Design Process

Starting at the top left of the V-Model, high-level designs and system requirements are established. As one progresses through the process, the above steps are re-evaluated to ensure that the objectives and analysis remain valid. In the model-in-the-loop (MIL) portion, the model is evaluated to ensure full functionality of critical model components such as the controller, plant and driver. At the simulation-in-the-loop (SIL) stage the completed controller software is incorporated into the simulation of the plant model. In this environment the controller can be validated through rigorous testing but at this stage it does not run in real-time. At the hardware-in-the-loop stage (HIL) the controller software is downloaded to the physical hardware and connected to the simulated plant. It is at this stage that the model must run in real-time. With all of the above steps completed and all requirements passing, the controller can be implemented into the physical

plant and validated to ensure it operates appropriately. It is important to note that the path of the V-model is not a straight line from left to right. Each step in the model should be revisited when necessary to support the verification and validation of the initial design and any implemented changes throughout the MBD process.

A critical input to most powertrain modeling activities is the drive cycle. A drive cycle is a predefined set of data that specifies vehicle speed versus time. A range of standardized drive cycles are available from automotive regulatory agencies such as the environmental protection agency (EPA). These data sets represent a specific driving profile with some exhibiting vehicle maneuvers representative of urban driving, highway driving, or most commonly a combination of the two. Drive cycles give automotive designers and regulators the ability to directly assess different vehicles and architectures performance over a standardized set of data. An example of the EMC drive cycles is shown in Figure 7.

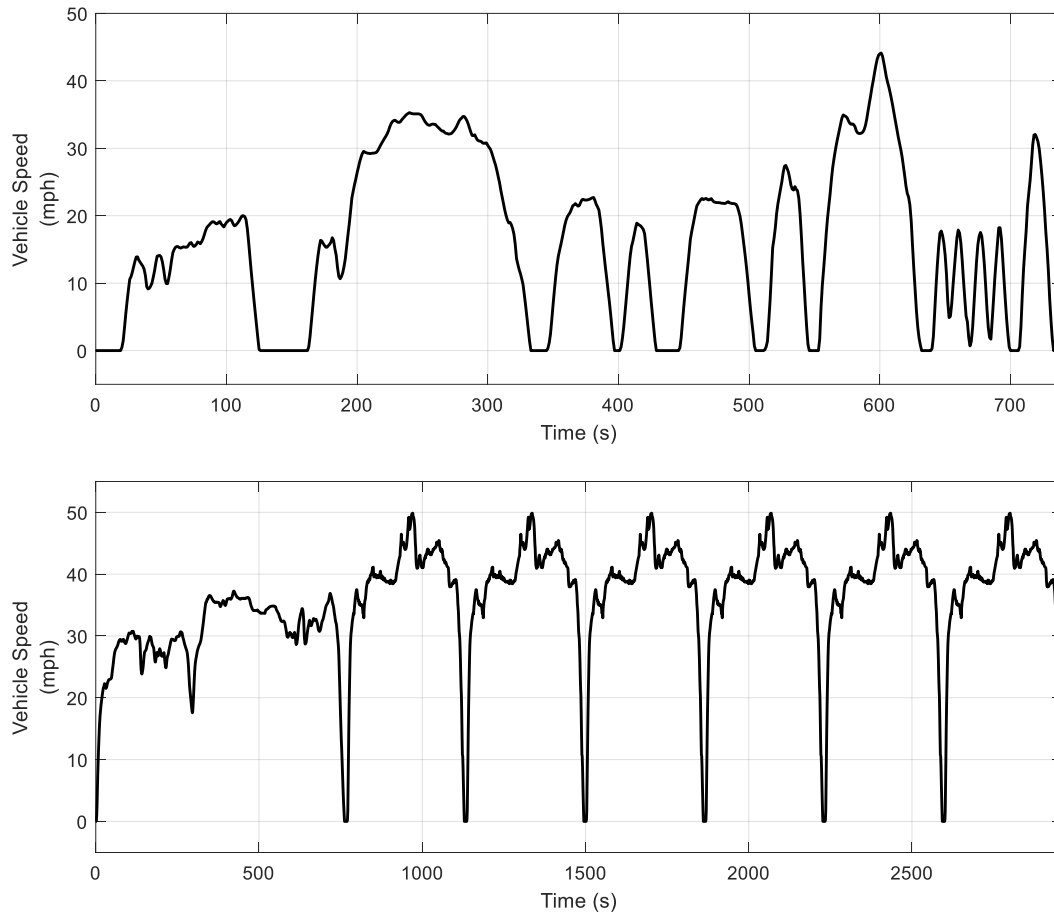


Figure 7: Mobility Challenge Energy Consumption Drive Cycles: a) {top} EMC City Cycle b) {bottom} EMC Highway Cycle

In the EMC City cycle, the average speed is low and there are numerous sharp acceleration and decelerations with extended idle times to represent driving in traffic and signalized intersections. In the EMC Highway cycle, there are extended periods of driving between stops as well as higher average speeds to be representative of vehicle operation on a highway or interstate. These two cycles are used by the EMC organizers as the scored cycle for which every team's competition vehicle's fuel economy and emission are evaluated. A similar process occurs in the United States by the EPA for automotive manufactures to ensure every vehicle meets the regulations set forth by the agency.

3.1.1 Model Types

There are two main categories of model types for HEV model-based design: empirical and physics based. The type of model governs the degree of abstraction from the physical principles that govern the system in a real-world environment.

Empirical models make use of measured or otherwise captured data from an existing system. This data may be fit to represent a new system that does not exist, such as the Willans Line ICE scaling technique [28] or simply implemented as a look-up table to be used with other powertrain components. A major benefit of empirical models is that they offer quick and low-cost simulation solutions. Many models that are required to run online utilize empirical models to ensure a fast enough runtime. A drawback of empirical models is that the data they are based on is collected under specific operating conditions. Due to this, it is not guaranteed that the model will accurately reflect the system performance under different conditions. For example, empirical engine models generally operate under the quasistatic assumption and thus transient operation is not accurately modeled.

In physics-based models, the state variables of the system are calculated according to the physical laws that govern the system in the real-world application. These types of models offer a more accurate representation of the true system but at the expense of higher computational load. Physics-based models are inherently better at modeling the system in a wide range of operating conditions when compared to empirical models. Many physics-based models are packaged as a black box such that the user need only provide information to the input to receive outputs. This simplifies the overall modeling effort but may lead to inefficiency if the black box outputs more information than desired for a specific simulation. An example of this is in Mathworks Simulink Powertrain

Blockset. There are a multitude of useful physics-based models for automotive components provided as ready to go black box models. One of these models is for the high voltage battery. For a simulation determining component torque requests, the user is only interested in electrical parameters such as current, voltage and SOC however, the model will calculate temperature values that are never used, adding to the computation time. For use in any online application, physics-based models must be tailored to a specific purpose or heavily simplified to keep the runtime at an appropriate level.

3.1.2 Model Structures

When designing a model to simulate the performance of a HEV, the structure of the model may vary depending on the type of simulation being conducted. Most HEV models will have three overarching sections: driver, controller and plant. A general basic structure of a model is shown in Figure 8.

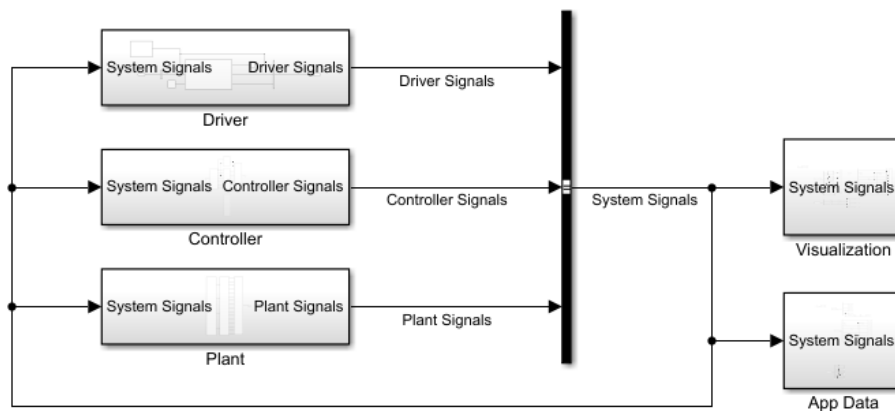


Figure 8: Basic Structure of a HEV model

The purpose of the driver model is to communicate the requirements of the drive cycle to the rest of the HEV model by determining torque requests, pedal positions, or some other control variable.

The controller model is where the supervisory control strategy logic is implemented. This logic includes the implemented energy management strategy, regenerative braking control, dynamic coordination strategies, and any other necessary supervisory control information. Finally, the plant model contains all of the components that make up the powertrain components, driveline components, vehicle dynamics, and any other mathematical model relevant to the vehicle. There are often other partitions used to visualize or record data from the model, however these are optional and do not directly affect the mathematics of the model.

When considering the structure of the overall model, it is important to understand the “flow” of information from the inputs of the modeled system to the outputs. There are two main structures for the flow of information: forward-looking and backward-looking structures. The choice of the direction of the flow of information is specific to the application of the model as the two directions are suited for different tasks.

Forward-looking models are structured to capture the causal nature of real driving conditions. In this structure, the simulated driver views the specified drive cycle and produces driver requests in the form of pedal positions, torque requests or some other request variable. This request is sent to the powertrain to produce torque with those components. That torque in-turn acts on the driveline to increase or decrease the speed of the vehicle. This speed is fed back to the simulated driver which influences the requests for the next time step. Forward-looking models are beneficial in control system design as they closely simulate the communications that will occur in a real vehicle under operation. It is important to know that in forward-looking model structures, the driver generally bases acceleration or deceleration requests on the error between current speed and the drive cycle, thus the drive cycle will likely never be followed exactly. Typically, when conducting

simulations, the driver model will be required to be within a narrow error threshold of the actual drive trace.

Backward-looking models however take a different approach that eliminates the error produced by the driver. In a backward-looking model the driver is essentially an extension of the drive trace. With the speed of the drive cycle known, the torque required to meet the specific speeds may be determined prior to conducting the simulation, or at each time step. With the required torque known at each timestep, the internal states of the vehicle components to achieve the path can be derived. With the known torque to achieve the specific speed of the drive cycle, the required torque can be propagated through the driveline to the vehicle components and determine their operating conditions. In a backward-looking model, the operating states of the vehicle need to be known a priori. In the HEV example, the torque split value needs to be known before the required torque can be appropriately accounted for when propagating torque values through the driveline to the powertrain components. Because of this, the backward-looking structure is ill-suited to develop control strategies; however, backward-looking models are useful for defining overall operating trends and performing high-level analyses of powertrain operation under different driving conditions. Backward-looking models are often used for determining global optimal solutions for the HEV control optimization problem using methods such as DP.

3.1.3 Model Fidelity Levels

Model fidelity defines the level of detail captured by the simulation model. Models can be implemented with various levels of fidelity depending on the application. The level of fidelity of the model is strongly related to the model type with physics-based models generally providing higher levels of fidelity than empirical models. The fidelity level can also be directly associated

with the runtime of a simulation as it can be stated that high fidelity models will have a longer runtime than low fidelity.

The key aspect of model fidelity is that the selection of the appropriate level depending on the model's purpose and design task being performed. For a HEV powertrain there are multiple mechanical, electrical, and chemical processes that occur simultaneously, each of which can be modeled with varying degrees of fidelity. For example, depending on the application requirements, an ESS may be modeled as a constant voltage source, higher-order RLC circuit, a set of lookup tables derived from captured test data, or a complex multi-physics model simulating electro-chemical reactions [10]. Each of these models has a place for a specific application, but an important aspect of a model to consider is a uniform level of fidelity across the board. It generally does not make sense to have a very high-fidelity battery and engine model in simulations that consider the design of a vehicle's suspension system.

Selection of model fidelity is influenced heavily by the application for which the model will be used. For many online applications such as model predictive control (MPC) or ECMS, the associated model must be low enough fidelity such that the algorithm can be solved within the real-time window. Similarly, for global optimal solvers such as DP and genetic algorithms, although they are executed offline, time constraints or computational limits may require lower fidelity models. High fidelity models are exceptionally useful in the late stages of control system development for uses in hardware-in-the-loop (HIL) environments. The WVU EcoCAR Team utilizes a dSPACE mid-size HIL simulator to test their hybrid supervisory controller (HSC). The HIL simulator relies on a high-fidelity full vehicle model and interfaces with the implemented control system on the HSC. The full vehicle model not only simulates the vehicle dynamics under

operation but simulates the CAN traffic between the HSC and the low-level powertrain controllers in the vehicle.

3.2 Powertrain Simulation Environments and Tools

There are many available simulation environments and tools available for applications to automotive systems. Some of these exist as black box solutions for powertrain simulation that can quickly be parameterized to provide capabilities for rapid prototyping and low fidelity results. Others provide a means to develop high fidelity models that can be customized to meet the needs of the user. Certain tools may be particularly useful in a specific simulation environment, such as MIL or HIL, and may provide a means to transition to another tool when progressing through the steps of MBD. The various tools used in each step of the MBD V-model are often referred to as a “toolchain”. For the WVU EcoCAR Team, the primary tools in the PCM and CAVs subteams’ toolchain are Mathworks MATLAB and Simulink packages as well as dSPACE’s Control Desk.

The PCM subteam primarily uses Mathworks Simulink package for all powertrain and controls development within the MIL environment. Simulink is a graphical simulation environment that provides a means to design and simulate systems before moving to hardware without having to write C, C++, or other low level computational code [29]. In Year 1 of the EMC the PCM team developed a high fidelity forward-looking full vehicle model utilizing proprietary powertrain information from GM as well as the physics based models in the Simulink Powertrain Blockset [30]. With this model, the team was able to assess the performance of various vehicle architectures for the team competition vehicle long before ever receiving the physical vehicle. Based on these simulations, the team identified a set of vehicle technical specifications, determined the necessary

vehicle architecture to meet consumer needs, sized major powertrain components, and began developing the hybrid supervisory control system.

As the PCM team transitions to more advanced testing methods such as SIL and HIL, dSPACE ControlDesk software is used. ControlDesk is a universal modular experiment and instrumentation for electronic control unit development [31]. ControlDesk allows for real-time implementation of the PCM team's supervisory control code onto physical hardware for use in the team competition vehicle. The tool provides seamless integration with Simulink by implementing the ability to directly build Simulink code onto dSPACE hardware. In the HIL environment, the PCM team uses a dSPACE mid-size simulator to simulate the plant of the full vehicle model with the physical HSC in the loop. This functionality provides the team with a means to debug supervisory control code before implementing this code into the competition vehicle.

3.3 Powertrain Subsystem Models

This section outlines the formulation of key automotive component models, and describes the corresponding model used by the EcoCAR team. The majority of the EcoCAR Team's component models are of the empirical type.

3.3.1 *Vehicle dynamics*

A real automobile moves through 3D space as a load varying distributed mass over a sprung suspension with various levels of traction on each wheel. The automobile is subject to various perturbations from windspeeds and experiences different forces and moments based on temperatures, body flex, material, etc. Development of a model with high enough fidelity to capture the dynamics of a real-world vehicle would require immense computational resources.

When modeling the powertrain and driveline behavior of the vehicle, it is often only necessary to consider 1-D motion in the longitudinal direction of the vehicle. The 1-D model generally provides enough fidelity to appropriately model powertrain performance and efficiencies and provides a foundation that can be expanded to include additional dimensions of motion if necessary.

The movement behavior of a vehicle along its moving direction is completely determined by all of the forces acting on its direction [1]. Figure 9 shows the free-body diagram of a vehicle modeled as a point mass moving in 1-D along an arbitrary grade.

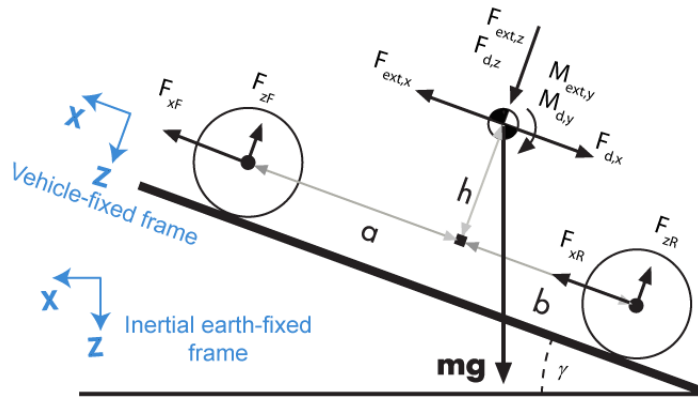


Figure 9: Free-Body Diagram of Vehicle Moving in One Dimension [32]

Applying Newton's second law to the free-body diagram, the longitudinal motion of the vehicle can be described as shown in Equations (3.1) and (3.2):

$$F_b = ma \quad (3.1)$$

$$F_b = F_{xF} + F_{xR} - F_{d,x} + F_{ext,x} - mg \sin(\gamma) \quad (3.2)$$

where m is the mass of the vehicle, a is the longitudinal acceleration, and F_b is the total force. The term F_b consists of longitudinal force components acting on the front axle (F_{xF}), longitudinal force components acting on the rear axle (F_{xR}), longitudinal drag force ($F_{d,x}$), external forces acting along the longitudinal axis ($F_{ext,x}$) and gravitational considerations acting against the vehicle on a

grade ($m * grav * \sin(\gamma)$). Each of the force components are computed based on the equations in [32] that correspond to the Mathworks Powertrain Blockset Vehicle Body 1DOF Longitudinal block as shown in Figure 10.

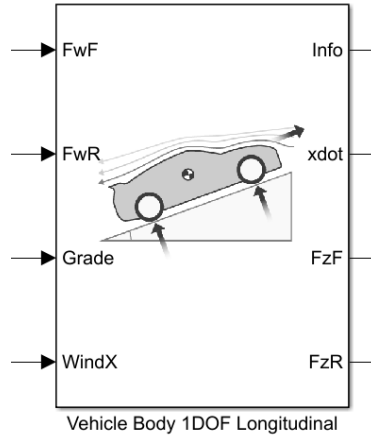


Figure 10: Vehicle Body 1DOF Longitudinal Block

In the EcoCAR team forward looking full vehicle model this block is parameterized with mass, drag coefficient, frontal area, vehicle dimensions, etc. to match the team competition vehicle. The block receives inputs of longitudinal forces on the front axle (FwF), rear axle (FwR), road grade (Grade) and longitudinal wind speeds (WindX). The outputs of vehicle speed (xdot), normal force on the front (FzF) and (FzR) axles as well as the “info” signal which contains additional information related to the vehicle such as drag, acceleration, etc.

The Mathworks block shown above provides flexibility for parameterization and the inclusion of perturbations such as grade and wind speed which can prove useful in forward looking models. However, for backward looking models, the solution of Equations (3.1) and (3.2) can be difficult to obtain. Alternatively, the roadload method can be used to generate a polynomial that describes the forces on the vehicle resisting its longitudinal motion as shown in Equation (3.3):

$$F_{roadload}(t) = A + B * V(t) + C * V(t)^2 \quad (3.3)$$

where the forces resisting longitudinal motion ($F_{roadload}(t)$) are calculated using vehicle speed (V) at each timestep (t) and the coefficients A , B and C . Coefficients A and B represent the static and dynamic friction forces due to bearing losses and tire/road rolling resistance, and C represents the aerodynamic resistance. The values of these coefficients are obtained in a standard procedure called a “coastdown” test. In this test, the vehicle is accelerated on a test track or dynamometer to a high speed, then allowed to coast to a stop while the velocity is recorded. With the mass known and coastdown velocity data, equation (3.3) can be derived via curve fitting [10]. The main model used by the EcoCAR team employs the Mathworks Simulink Powertrain Blockset Vehicle Dynamics 1DOF Longitudinal block, however, to allow for more flexibility in modeling efforts, GM provided the roadload coefficients for the team competition vehicle.

3.3.2 Internal Combustion Engine

The ICE is a highly complex machine that controls combustion reactions to convert the chemical energy of an air/fuel mixture into thermal energy. Some of this thermal energy is harnessed by a mechanical linkage to provide a power output. The most common engine used in automobiles today is the 4 stroke (4S), spark-ignited (SI) ICE. An illustration of the major components that make up a one of these engines is shown in Figure 11. The 4S-SI ICE relies on a multitude of subsystems to support the operation of the engine, but the basic principles of operation are as follows. The first stroke is the induction stroke where the inlet valve is opened, and the exhaust valve is closed. A fuel/air mixture is drawn into the cylinder via the inlet manifold as the piston travels downward. Next, the inlet and exhaust valves are both closed, and the piston begins moving upward, compressing the fuel/air mixture in the compression stroke. As the piston approaches top dead center, the spark plug produces a spark that ignites the mixture. Combustions propagates through the fuel/air mixture causing temperature and pressure to rise thus pushing the piston down

the cylinder. As the cylinder begins moving back up the cylinder, the exhaust valve opens allowing the combusted fuel/air mixture to be removed from the cylinder as exhaust. The next motion downward of the piston is the induction stroke of the next cycle [1].

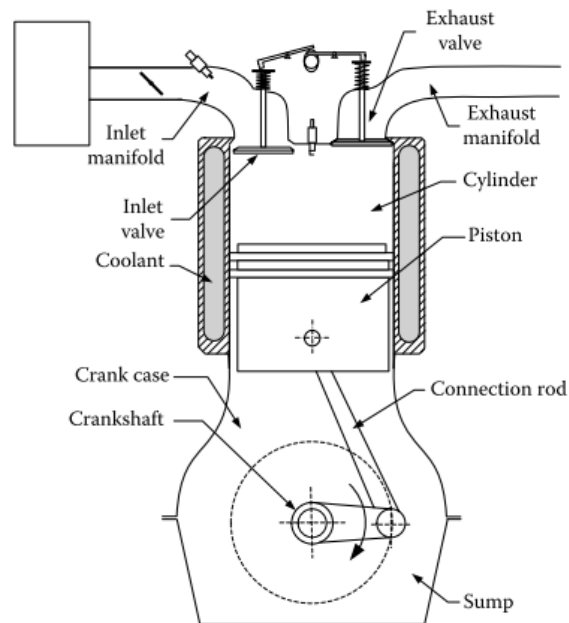


Figure 11: Illustration of the Components of a Four Stroke Spark-Ignition Internal Combustion Engine [1]

It is through this process that the ICE is able to provide torque for either conventional vehicles or HEVs. When modeling the operation of an ICE, there are a wide variety of approaches that can be used with varying levels of fidelity. A few of these modeling types may be the following in order of increasing complexity: static map, static map and lumped-parameter dynamic model, one-dimensional fluid-dynamic model and three-dimensional fluid-dynamic model. The latter two approaches are only necessary for detailed studies focused specifically on the engine subsystem, while the first two methods are well suited for use in models that include the engine as part of a comprehensive system (powertrain or vehicle).

The static map approach assumes the engine to be a perfect actuator which responds immediately to commands. The torque availability is modeled as a function of engine speed and performance

metrics of fuel consumption and emissions are tabulated as functions of torque and speed. The data is typically based on collected data for existing engines or scaled from some other collected data meaning that the majority of static map models are of the empirical type. The data collected to parameterize these static maps is generally proprietary but to illustrate the structure of these maps, Figure 12 is included [2].

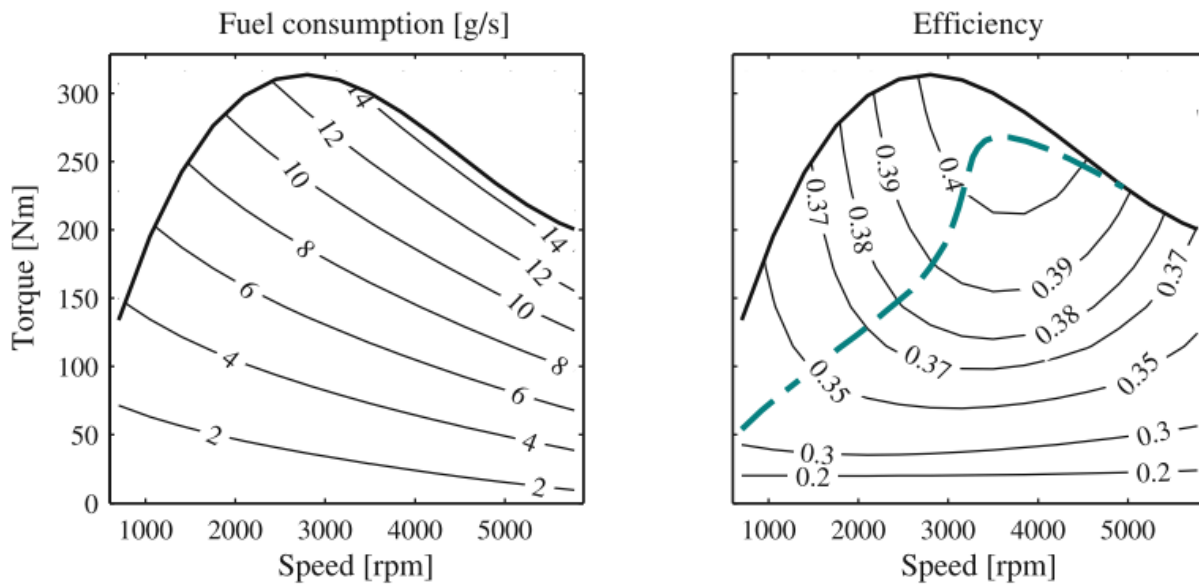


Figure 12: a) Engine Fuel Consumption Map {left}. B) Engine Efficiency Map {right}

Using the data shown in Figure 12, the performance of an ICE can be modeled and used for energy analysis of a conventional or hybrid vehicle. It is important to note that the data collected to populate these tables is steady state operation and thus models that rely on these maps are often referred to under the “quasistatic” assumption. These maps may be paired with a transfer function to model the dynamic limitations of the torque output.

The EcoCAR Team forward looking full fidelity model employs the Mathworks Simulink Powertrain Blockset Mapped SI Engine [33] as shown in Figure 13.

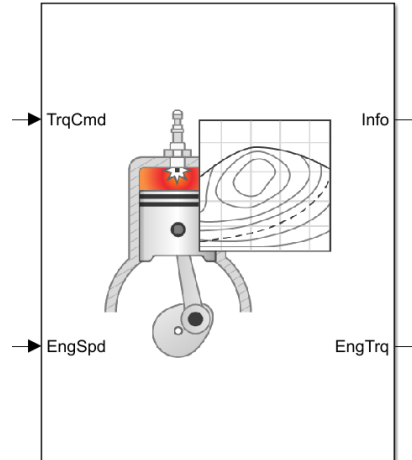


Figure 13: MathWorks Simulink Mapped SI Engine [33]

The mapped SI engine block receives inputs of the torque command (TrqCmd) and engine speed (EngSpd) and outputs the actual engine torque (EngTrq) as well as the “info” signal that contains relevant information for temperatures, emissions, fuel consumption, etc. This block is parameterized utilizing proprietary GM information for the 2.5L LCV Engine including engine component geometry, dynamic information, and performance data tables. In the vehicle model, the mapped SI engine block is supported by the SI controller block from MathWorks [34] to model the subsystems that support the engine under operation.

3.3.3 Energy Storage System

The energy storage system (ESS) is one of the key components that support the HEVs electrified drivetrain. The main purpose of the ESS is to convert chemical energy into electrical energy in discharge scenarios and convert electrical energy into chemical energy during charging events. It is important that the ESS is able to do both of these conversions with a high level of efficiency. These electrochemical processes that occur within this system are complex and difficult to model, however the performance of the ESS significantly impacts the overall performance of a HEV. Similar to the ICE, the ESS can be modeled in a variety of ways from employing empirical lookup

tables to a complex multi-physics model of the electrochemical processes. A trade-off in modeling battery systems must be made between the inclusion of depth and the amount of computation effort required.

In the EcoCAR full vehicle model, the ESS is modeled after the GM HEV4 rechargeable energy storage system (RESS). In the MathWorks Simulink model, the Datasheet Battery block from the Powertrain Blockset is employed [35] as shown in Figure 14.

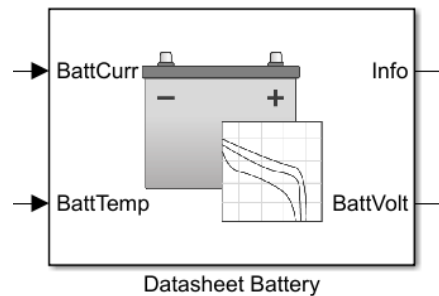


Figure 14: MathWorks Simulink Datasheet Battery [35]

This battery model implements a lithium-ion battery that is parameterized using GM data for the HEV4 battery pack. The inputs to this block are the battery current (BattCurr), which is generally determined from the torque required at the electric motor, and the battery temperature (BattTemp). This block outputs the battery voltage (BattVolt) as well as the “info” bus which contains numerous relevant operating parameters of the battery pack. Relevant parameters such as open-circuit voltage (OCV), internal resistance, state of charge (SOC) and battery temperature are determined using lookup tables.

3.3.4 Electric Motor

The electric motors integrated into the HEV drivetrain can be modeled similar to the ICE by utilizing maps of torque and efficiency. The relation between torque at the shaft and electric power

is provided by an efficiency map, which can be expressed as a function of speed and torque. The map may include the interactions with power electronics, such as the inverter, between the main electric bus and motor or a static efficiency value may be included to account for these components.

The EcoCAR team full vehicle model employs the MathWorks Simulink Powertrain Blockset Mapped Motor [36] block as shown in Figure 15.

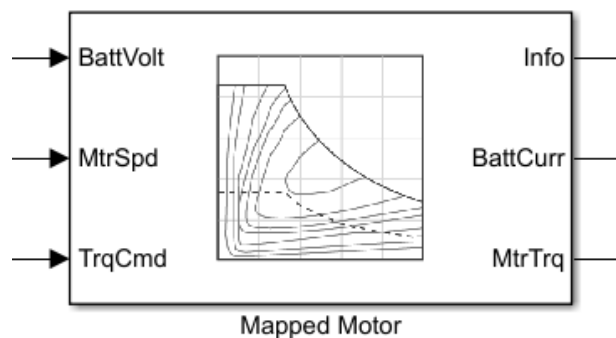


Figure 15: MathWorks Simulink Mapped Motor [36]

The Mapped Motor block is parameterized with data provided by Magna Powertrain to represent the P4 electric motor that makes up the eAWD system. This data includes the maximum torque versus speed curve and the electrical losses. With this information the inputs of battery voltage (BattVolt), motor speed (MtrSpd) and torque command (TrqCmd) are used to determine the required battery current (BattCurr), produced motor torque (MtrTrq) and the “info” bus that contains relevant information for the operation of the electric motor.

3.3.5 Driveline

The driveline of the vehicle represents the components of the vehicle that deliver torque from the torque generating devices such as the ICE and electric motor to the road surface. The friction

brakes are also included within the driveline model. The torque path of the driveline model for a parallel P4 HEV is shown in Figure 16.

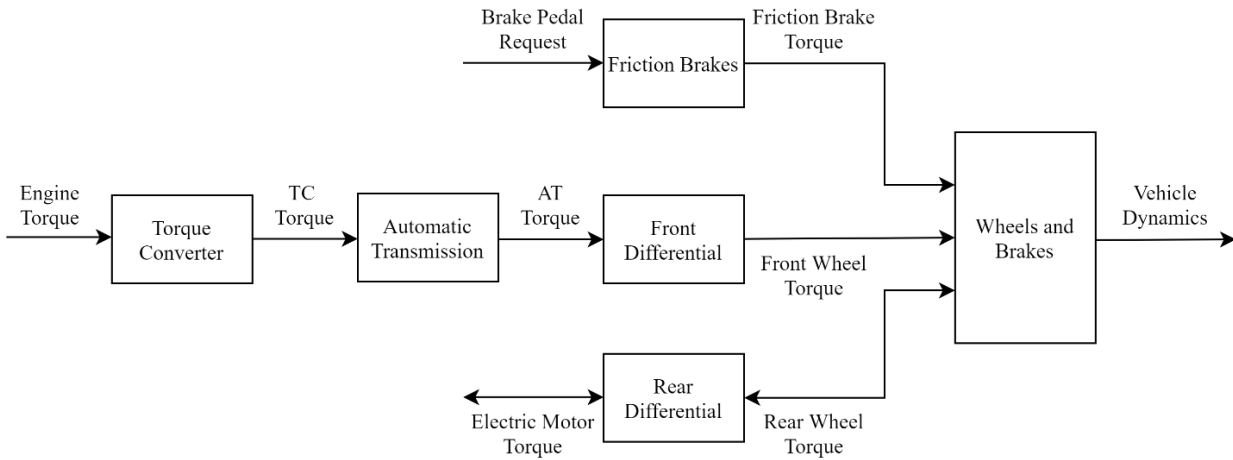


Figure 16: Driveline Torque Path for Parallel P4 HEV

Starting from the ICE's torque path, the torque converter is located on the output shaft of the engine. The torque converter is a fluid coupling device that is used to transmit motion from the engine to the transmission input shaft. It multiplies the torque similar to a reduction gear but unlike most other mechanical joints, it provides extremely high damping capabilities. The torque converter is made up of three co-axial elements: a pump (or impeller) connected to the engine shaft, a turbine connected to the transmission, and a stator in between as shown in Figure 17.

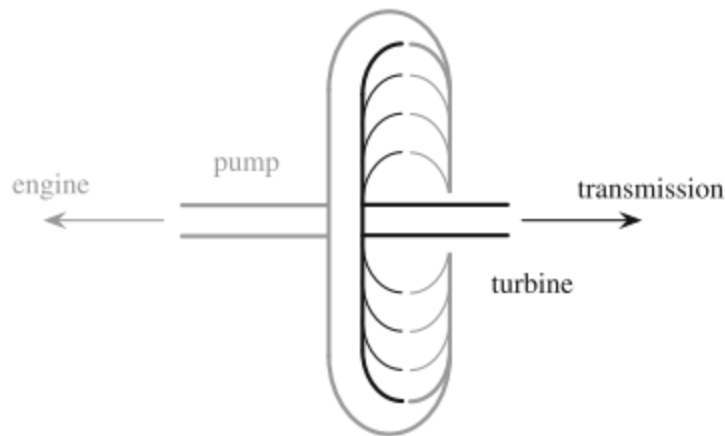


Figure 17: Torque Converter Schematic [2]

The fluid in the torque converter is moved by the pump which drags the turbine and depending on the modification of the flow by the stator transmits torque from the engine to the transmission [2]. The EcoCAR full vehicle model employs the MathWorks Simulink Powertrain Blockset Torque Converter [37]. This block is parameterized with data from the 2019 Chevrolet Blazer to match the WVU Team competition vehicle.

The next component in the conventional drivetrain torque path is the automatic transmission. The EcoCAR Team vehicle employs the GM M3D 9T50 9-speed transmission. In the full vehicle model, the MathWorks Simulink Powertrain Blockset Ideal Fixed Gear Transmission [37] is used. This model utilizes parameterized data as well as basic physical relationships to determine the speed and torque relationship between the inputs and outputs of the transmission. This block also requires the gear command from the HSC to determine the current gear selected and when shifts should occur. As the selection of the gear ratio is one of the degrees of freedom that the HEV can use to improve fuel economy, it is necessary for the transmission model to allow for the simulation of different ratios during a drive cycle.

The front and rear differentials are modeled utilizing the MathWorks Simulink Powertrain Blockset Open Differential [38] which is parameterized with inertial and damping data supplied from GM and Magna, respectively. This component is relatively simple as it utilizes a single gear ratio to transmit power from the source to the shafts attached to the wheels of the vehicle.

The friction brakes are modeled as a simple relationship between the deceleration pedal request and the maximum braking pressure. For a 25% braking request, the friction brakes apply 25% of the maximum available braking pressure. This braking pressure value is used by the MathWorks Simulink Powertrain Blockset Longitudinal Wheel [39] model with the disk brake type selected. The wheel model is a physics-based model that accounts for friction braking in the vehicle as well the wheel torque delivered and tire friction related calculations. In the EcoCAR Team model, the driveline subsystem is closely integrated into the vehicle dynamics model as all of the vehicle dynamics operational inputs are either outputs from the driveline or environmental factors such as temperature, grade, and windspeed.

3.3.6 Driver

The driver is an integral part of a forward-looking model that can be modeled utilizing a vast variety of methods. The primary responsibility of the driver model is to interpret the drive cycle and issue commands to the vehicle to meet that cycle within a reasonable amount of error. In the EcoCAR Model, the MathWorks Simulink Powertrain Blockset Longitudinal Driver Model [40] is used as shown in Figure 18.

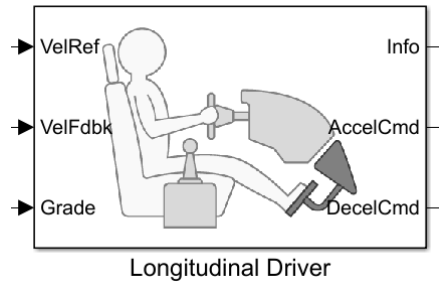


Figure 18: MathWorks Simulink Longitudinal Driver [40]

The driver subsystem receives three input signals: the reference velocity of the drive trace (VelRef), the actual vehicle velocity (VelFdbk) and the road grade (Grad). Based on the error between the actual and drive cycle velocity, the driver model outputs acceleration commands (AccelCmd) and deceleration commands (DecelCmd) as well as an “info” signal that contains additional information related to the driver. Configuration parameters to the driver model include driver response time, preview distance, vehicle aerodynamic drag coefficient, rolling resistance coefficient, vehicle weight and driveline resistances.

3.3.7 Overall Structure

In this section, an in depth look at each of the powertrain and driveline component models has been discussed. Understanding how each of these components work is critical to designing a comprehensive HEV model that adequately captures the vehicle’s performance. Another aspect of the comprehensive model is where each of these components reside in the mathematical model. The EcoCAR Team full fidelity vehicle model contains a number of supporting subsystems that are necessary to the controls development side of their work; however, for the simulation of powertrain performance, these subsystems are not required. For the purposes of the model development for this these, the EcoCAR model has been stripped down to include only the

subsystems and logic required to adequately model the powertrain performance of the team competition vehicle. The structure of this simplified model is shown in Figure 19.

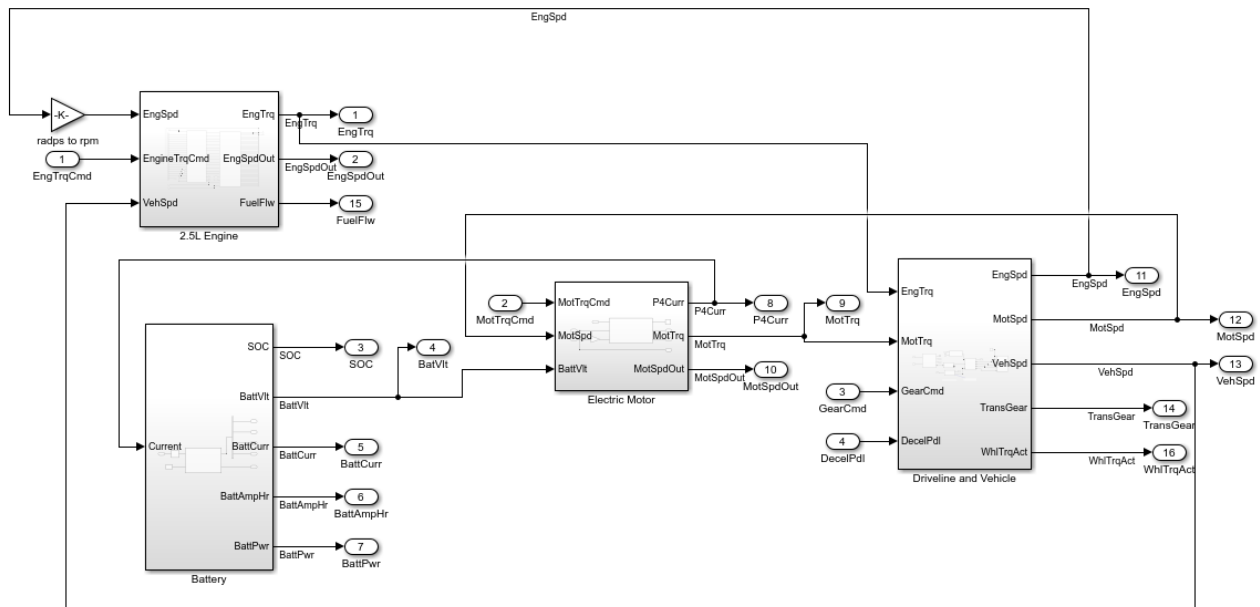


Figure 19: Overall Structure of Simplified Powertrain Plant Model

In the simplified model, the driver subsystem issues commands to meet the prescribed drive cycle. These commands are interpreted by the controller subsystem which in-turn issues commands to the components within the plant model. The engine torque command is issued to the 2.5L engine subsystem while the engine speed and vehicle speed are part of the feedback loop to this system. The 2.5L Engine subsystem issues the produced engine torque to the driveline subsystem and other informational signals such as the engine speed and fuel flow. The electric motor receives a motor torque command while the motor speed and battery voltage are part of the feedback loop. The electric motor subsystem produces a motor torque to the driveline subsystem as well as battery current to the ESS subsystem. Based on the current from the electric motor, the ESS determines its SOC and other operational parameters such as battery current, capacity and power. Additionally, the battery subsystem supplies the battery voltage to the electric motor subsystem.

The driveline receives the torque commands from the engine and motor and propagates those values through the relevant driveline components which ultimately calculate the torque and force values acting on the wheels of the vehicle. Those torques and forces are used by the vehicle dynamics subsystem to produce a vehicle speed. That vehicle speed is part of the feedback loop to the driver which is used to issue the next command. [7], [8], [41]

4.0 Benchmark Analysis Methodology

As mentioned in the introduction, the goal of the WVU EcoCAR Team is to design and build a vehicle with better than stock fuel economy, reduced emissions, and similar vehicle performance. The selected P4 parallel HEV architecture provides the powertrain with flexibility in meeting the driver's torque requests. This flexibility can be exploited to achieve a reduced fuel consumption by successfully implementing a supervisory control strategy that employs an efficient and robust energy management strategy. As described in Section 2.0, there are numerous energy management strategies that could be used to achieve a range of fuel economy results.

It is known that many of these strategies may increase fuel economy above the stock vehicle, but the knowledge of the global optimum is useful for ensuring that the selected control strategy is maximizing the usefulness of the flexibility. Additional operational benchmarks that are independent of any specific control strategy are also key to objectively evaluating the performance of any hybrid supervisory control strategy. The following points summarize the key information to be collected to evaluate the architecture and selected energy management strategy:

1. The optimal control policies (component operating profiles) for the operating regimes of city and highway driving
2. The optimal fuel economy from application of those control policies

3. The optimal power split between electrical and mechanical power
4. The ideal operating regime to employ under specific driving conditions.

This data will provide an insightful look at the potential gains from implementing this architecture into the EcoCAR team competition vehicle. This information can be used to guide the development and implementation for real-time supervisory control strategies. Because of this, the benchmark analysis of the WVU EcoCAR Team competition vehicle is the focus of this work.

In order to conduct the analysis for the parallel P4 hybrid-electric 2019 Chevrolet Blazer, it is necessary to consider all events over the driving cycle. Decisions made at or near the beginning of the cycle will influence the ending conditions. For example, when starting from a high SOC, it would seem to make sense to use the electric motor heavily due to its high efficiency. However, this discharge event near the beginning of the cycle would require a corresponding charge event near the end of the cycle which may lead to overall lower efficiency. In real-world conditions, it is not possible to take these future events into consideration due to the forward-looking nature of these control systems. It is also not possible to search through all feasible paths in real-time with the computational technology available in today's automobiles. As mentioned in Section 2.2.2 global optimization methods are well-suited for this type of analysis.

The identification of optimal control policies over a given drive cycle is fundamentally a constrained global optimization problem. The control policies to be found are optimal over a complete drive cycle and thus drive cycle time is an additional variable in the optimization. This allows the optimization problem to be formulated as a path search problem. Because of this, of all other methods available, dynamic programming was selected for the benchmarking analysis.

Dynamic programming offers dramatically reduced computation time compared to other graph search methods. As this is an off-line global optimization problem, the computation time is not as much of a constraint as it would be in an on-line optimization problem; however, it is desirable to keep the computation time at a reasonable threshold to improve the usability of the program. DP was selected over other comparable graph and tree search methods due to the relative complexity of the cost calculation and the overlapping nature of the subproblems. Analyzing a powertrain operating over a drive cycle is more complex than a typical path search problem, as the transition costs change depending on the particular path selected.

An added benefit is that DP implicitly excludes infeasible solution paths. This characteristic is particularly important as the use of backward-looking model requires that it follows a driving cycle perfectly. This requirement can yield many infeasible solutions where particular components or operating states cannot meet the required load.

4.1 Dynamic Programming History and Fundamentals

Before considering the application of Dynamic Programming to the HEV powertrain control problem, it is important to understand the fundamental mathematical concepts behind DP. DP was developed by mathematician Richard Bellman in the early 1950's as part of work on stochastic multi-stage decision-making algorithms for the United States Air Force [42]. Since its inception, DP has been widely adopted in inventory management, scheduling, process planning, and other multi-stage decision making problems [43].

4.1.1 The Basic Problem

Consider a discrete-time deterministic system, the states evolve over time as described by the transition function:

$$x_{k+1} = f(x_k, u_k), \quad \{k = 0, \dots, T\} \quad (4.1)$$

where x_k , the state variable at stage k , exists in a space S_k , and u_k , the control input at time k that modifies x_k , exists in space C_k .

The set of control inputs, termed a *policy*, consists of a sequence of functions:

$$\pi = \{u_0, u_1, \dots, u_{T-1}\} \quad (4.2)$$

where each u_k is constrained to take values in a subset of C_k , depending on the current state x_k .

The specific constraints applied are part of the formulation of a particular DP problem, and serve to eliminate infeasible control inputs. The set C_k is called an admissible policy.

Each transition of x_k between different values incurs a cost. The cost represents the effort of moving from one state to another and serves to differentiate the paths. In a literal path-search problem, the cost represents the distance required for each path step. The cost can also be used to enforce constraints, by highlighting infeasible state transitions and ensuring they will not be selected during the path search. The cost function describing the state transition and implementing constraints is given by $g_k(x_k, u_k)$.

The cost of a given path is additive over time since it accumulates over each stage of the problem.

The total cost of a path can be expressed as:

$$J(x_0) = g_T(x_T) + \sum_{k=0}^{T-1} g_k(x_k, u_k) \quad (4.3)$$

thus, the cost of following a given policy π starting from state x_0 is:

$$J_\pi(x_0) = g_T(x_T) + \sum_{k=0}^{T-1} g_k(x_k, \pi_k) \quad (4.4)$$

An optimal policy can be found, denoted π^* , that minimizes this cost, such that:

$$J_{\pi^*}(x_0) = \min \{ J_\pi(x_0) \} \quad (4.5)$$

4.1.2 Bellman's Principal of Optimality

The cornerstone of DP is Bellman's Principal of Optimality, which states that "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision" [44].

This property suggests that if the following optimal policy:

$$\pi^* = \{u_0^*, u_1^* \dots, u_{T-1}^*\} \quad (4.6)$$

passes through the state x_i at time $k = i$, and with the desire to find the optimal policy to get from x_i to x_T , one would do so by minimizing the truncated cost function:

$$J_\pi(x_i) = g_T(x_T) + \sum_{k=i}^{N-1} g_k(x_k, u_k) \quad (4.7)$$

and would find that the optimal policy is simply the truncated policy:

$$\pi_i^* = \{ u_i^*, u_{i+1}^* \dots, u_{T-1}^* \} \quad (4.8)$$

Practically, this means that the overall optimal control policy can be derived by sequentially determining the set of optimal policies for a series of smaller sub-problems. Applying this to a road trip example, if the fastest route from New York, New York to Los Angeles, California passes through Morgantown, West Virginia, then the principal of optimality implies that the Morgantown

to Los Angeles section of the overall drive is also the fastest route from Morgantown to Los Angeles.

4.1.3 *Dynamic Programming by Backwards Induction*

There are several approaches to the practical implementation of the DP to solve for global optimal solutions. The selection of which approach to adopt depends on the particular characteristics of the problem that is being solved. In the HEV powertrain optimal control problem where knowledge of the drive cycle is known *a priori*, the optimal policy can be found with a DP algorithm that employs backward induction. This is where the problem is initially considered from the final, or terminal, point and the cost-to-go is calculated at each successive stage, working backwards in time toward the initial point.

First, the final stage ($k = T - 1$) is considered, and an optimal control policy is determined for this step. This is referred to as the “tail sub-problem”. Next, the optimal policy for the tail sub-problem involving the final two stages ($k = T - 2:T - 1$) is determined. This process is continued until the policy for the full problem ($k = 0: 1$) has been identified.

This type of problem can be efficiently solved using a recursive algorithm, as follows. Starting at the final stage ($k = T - 1$), the minimum cost is found to be:

$$J_{T-1}^*(x_{T-1}) = \min \{ g_{T-1}(x_{T-1}, u_{T-1}) \} \quad (4.9)$$

The minimum cost of the next stage ($k = T - 2$) is therefore:

$$J_{T-2}^*(x_{T-2}) = \min \{ g_{T-2}(x_{T-2}, u_{T-2}) + J_{T-1}^* \} \quad (4.10)$$

Working backwards from here toward the initial stage $k = 0$, the total cost can therefore be found by the recursive equation:

$$J_k^*(x_k) = \min \{ g_k(x_k, u_k) + J_{k+1}^* \} \quad k = \{0, \dots, T - 1\} \quad (4.11)$$

where the cost $J_0^*(x_0)$ is the optimal cost of the overall control policy π^* .

4.1.4 Statement of the Optimization Problem

For the benchmark analysis for the optimal control of a parallel P4 hybrid-electric 2019 Chevrolet Blazer, the basic objective function can be expressed as:

$$J = \min \sum_{k=0}^T \dot{m}_{fuel}(k) \quad (4.12)$$

where k is the discrete time step of the drive cycle, \dot{m}_{fuel} is the fuel consumption rate of the engine (quasi-static over each timestep k), and 0 and T are the beginning and end points, respectively. It is important to note that the minimization here is not instantaneous fuel consumption, but instead the fuel consumption over the full drive cycle. Minimization of the instantaneous fuel consumption would simply drain the energy storage system (ESS) by maximizing electric motor use and then revert to conventional vehicle operation. Minimization of the fuel consumed over the full drive cycle takes a deeper look into where it makes the most sense to charge and discharge the ESS via the electric motor.

For the benchmark analysis results to be directly comparable to outputs obtained via other control methods, or other powertrain architectures, one of two requirements must be met: CS operation must be maintained, or an energy equivalent fuel economy must be determined. In this work, CS operation is strictly enforced due to the convenience afforded by the necessity to pick the starting state when determining the optimal control policy. It is important to note that the explicit selection of the CS SOC as the starting point may artificially skew the optimization as the optimal policy from a different starting point with an included energy conversion may yield higher fuel economy.

To implement this requirement, the change in ESS SOC between the beginning and end of the drive cycle must be kept at zero, regardless of the level of SOC variation during the drive. Note that this restriction is required for comparison and would not normally be present in real-world operation. This generates the constraint:

$$\sum_{k=0}^T P_{batt}(k) = 0 \quad (4.13)$$

The charge sustaining constraint is not enforced explicitly in the mathematics of the backwards looking model, instead it is addressed in the DP algorithm formulation.

A key constraint is imposed by the driver torque request $\tau_{w,req}$; the powertrain cannot produce any more torque at the wheels than the driver commands in order to perfectly follow the drive cycle. To reduce the complexity of the problem, the independent variable is selected to be the wheel torque produced by the electric motor, $\tau_{w,mot}$. The wheel torque from the engine ($\tau_{w,ICE}$) or mechanical brakes ($\tau_{w,brk}$), depending on whether the vehicle is accelerating or decelerating become the dependent variables. In order to represent the operating regimes of positive or negative acceleration in the parallel configuration the following constraints are used.

$$\tau_{w,req}(k) = \tau_{w,ICE}(k) + \tau_{w,mot}(k), \quad \tau_{w,req}(t) > 0 \quad (4.14)$$

$$\tau_{w,req}(k) = \tau_{w,brk}(k) + \tau_{w,mot}(k) + \tau_{w,ICE}(k), \quad \tau_{w,req}(t) < 0 \quad (4.15)$$

As mentioned above, the trivial solution for optimal fuel economy would be to never consume any fuel and only discharge the ESS to power the vehicle. Implementing a charge sustaining (CS) requirement is useful not only to provide a means for comparing fuel economy results, but also to avoid the trivial solution [45]. CS behavior can be strictly enforced by selecting the initial state to be equal to the predefined final state. Additional constraints are imposed on the optimization due to the physical limitations of the vehicle powertrain. There are finite limits to the amount of

instantaneous power the ESS, electric motor, and ICE can supply at a given vehicle speed, stemming from the limits on both torque output and rotational speed of the components. These are hard limits, meaning the simulation result is infeasible if it must exceed them. To deal with these cases, the algorithm checks the component operating points and power levels as they are calculated and discards any solutions that are tagged as infeasible. These constraints are as follows where the subscript c identifies a component torque:

$$SOC_{min} \leq SOC(k) \leq SOC_{max} \quad (4.16)$$

$$\tau_{c,ICE,min} \leq \tau_{c,ICE}(k) \leq \tau_{c,ICE,max} \quad (4.17)$$

$$\omega_{ICE,min} \leq \omega_{ICE}(k) \leq \omega_{ICE,max} \quad (4.18)$$

$$\tau_{c,mot,min} \leq \tau_{c,mot}(k) \leq \tau_{c,mot,max} \quad (4.19)$$

$$\omega_{mot,min} \leq \omega_{mot}(k) \leq \omega_{mot,max} \quad (4.20)$$

4.2 DP Problem Formulation

As backward induction dynamic programming requires the use of a discrete-time system, the powertrain and vehicle are modeled as such. The drive cycle is discretized to a time resolution of 1 second per step. Each time step, referred to as a “stage” in DP terminology, is represented by the variable k . The SOC is selected as the state variable, thus:

$$x_k = f(SOC) \quad (4.21)$$

The selection of this particular quantity is done to take advantage of an important aspect of the backwards-induction method of DP. In this case, the final state of the system at time $k = T$ becomes the initial state (at stage $k = 1$), CS operation can be enforced. With the SOC as the state variable, the maximum and minimum SOC constraints can be easily imposed as these values are

inherently defined by the boundaries of the discretized state space. The state space is discretized with a step size based on the balance of execution time, quality of results, and memory.

The motor wheel torque and the current transmission gear are selected as the control variables, thus:

$$u_k = [\tau_{w,mot}(k), N_{gear}(k)] \quad (4.22)$$

The motor torque is useful to control as it is included in the driver torque request for either an acceleration or deceleration request as shown in Equations (4.14) and (4.15). Utilizing the motor torque as a control input effectively navigates the space for the torque split between the ICE and electric motor as well as determining the most efficient use of regenerative braking. The transmission gear ratio is used as a control input to investigate the effect of the transmission gear ratio on the fuel efficiency of the ICE. Identifying the optimal control policy of the transmission can assist in not only determining changes to the energy management strategy, but also potentially the shift schedule of the transmission. These control inputs influence the following state transition function:

$$SOC_{k+1} = \frac{-P_{batt}\Delta t}{Battery\ Capacity} + SOC_k \quad (4.23)$$

where P_{batt} is the battery current required to provide the requested motor torque, Δt is the time step of the simulation in hours and *Battery Capacity* is the total energy capacity in Whr of the ESS. The negative sign is used to preserve the flow direction convention of positive current represents charge of the ESS and negative current represents discharge of the ESS.

4.3 Benchmarking Model

4.3.1 Drive Cycle Data and Shift Schedule

The benchmarking model is a backwards facing model, which means component speeds and torques are propagated from the wheels through the drivetrain and to the powertrain components. These torques and speeds are based on the drive cycle selected for this analysis. First, the drive cycle must be resampled to the discrete time step selected for the DP algorithm. In this analysis, the original drive cycle data has a time step of 0.1 seconds. The drive cycle is resampled at 1 second by selecting the values of the drive cycle that correspond to the 1 second timestep and ignoring the other entries.

Once the drive cycle has been resampled, the roadload polynomial equation is used to determine the force at the wheels at each vehicle speed as described in Equation 3.3. The coastdown test for the 2019 Chevrolet Blazer competition vehicle resulted in the following polynomial to describe the roadload force:

$$F_{roadload}(k) = 118.56 + 3.54 V(k) + 0.54 V(k)^2 \quad (4.24)$$

where $F_{roadload}(t)$ is the roadload force at the wheels, in N, for each time step and $V(t)$ is vehicle speed. Applying Newton's law for rigid bodies, the total force at the wheels of the vehicle can be determined as:

$$F_{total}(k) = m\dot{V}(k) + F_{roadload}(k) \quad (4.25)$$

where $F_{total}(t)$ is the total force at the wheels for each time step, m is the mass of the WVU EcoCAR team competition vehicle, and $\dot{V}(t)$ is the acceleration of the vehicle at each timestep. The wheel torque, $\tau_{wheel}(t)$, of the vehicle is then calculated as follows:

$$\tau_{w,req}(k) = F_{total}(k) * r_{wheel} \quad (4.26)$$

where r_{wheel} is the rolling radius of the tires of the 2019 Chevrolet Blazer. The rotational wheel speed must also be known for each time step. The wheel speed is calculated from the linear velocity of the vehicle as follows:

$$\omega_{wheel}(k) = \frac{V(k)}{r_{wheel}} * \frac{30}{\pi} \quad (4.27)$$

where $\omega_{wheel}(k)$ is the wheel speed in RPM.

In the EMC, the WVU team is not permitted to modify the low-level controls of the M3D automatic transmission. This constrains the operation of the transmission to the shift map of the conventional 2019 2.5L LCV Blazer. As part of the benchmark analysis, the constraint of the conventional shift schedule is implemented to capture the optimal fuel economy that can be achieved given the WVU EcoCAR Team's constraints. The gear shift schedule is determined based on the current vehicle speed and accelerator pedal position. The speed of the vehicle is inherently known *a priori* due to DP's reliance on a predefined drive cycle. The accelerator pedal position is determined by implementing the 2.5L LCV and M3D pedal map. This map specifies the accelerator pedal position as a function of vehicle speed and wheel torque requirement as shown:

$$Ped_{Accel}(k) = f(V(k), \tau_{w,req}(k)) \quad (4.28)$$

where $Ped_{Accel}(k)$ is the accelerator pedal position at time k . With the accelerator pedal known the upshift and downshift speed thresholds can be determined from tabulated data extracted from the low-level control system for the M3D transmission:

$$V_{upshift}(k) = f(Ped_{Accel}(k), V(k)) \quad (4.29)$$

$$V_{downshift}(k) = f(Ped_{Accel}(k), V(k)) \quad (4.30)$$

where $V_{upshift}(k)$ is the upshift speed threshold and $V_{downshift}(k)$ is the downshift speed threshold. The speed thresholds are used with the following control logic to determine the gear setting at each timestep of the simulation:

$$N_{gear}(t) = \begin{cases} N_{gear}(k-1) - 1 & \text{if } V(k) < V_{downshift}(k) \\ N_{gear}(k-1) & \text{if } V_{downshift}(k) \leq V(k) \leq V_{upshift}(k) \\ N_{gear}(k-1) + 1 & \text{if } V_{upshift}(k) < V(k) \end{cases} \quad (4.31)$$

where $N_{gear}(k)$ is the selected gear number at time t and $N_{gear}(k-1)$ is the selected gear number from the previous timestep $k-1$. This logic is applied multiple times per timestep to ensure that the appropriate gear is selected. For example, in a large deceleration event, the transmission may shift from gear 9 to gear 7 within a single timestep. This logic is implemented as a conditional statement that repeats until the vehicle speed is between the downshift and upshift thresholds for the given gear and accelerator pedal position.

4.3.2 Electric Motor Torque Lookup Table

In the DP algorithm, the state transition function is used to determine the battery power requirement to move from one SOC to another. The battery power value is the primary mechanism that drives the calculation of the cost-to-go for each transition, but obtaining the fuel consumed based on this battery power requires multiple subsequent calculations. The first calculation in this sequence is the determination of produced electric motor component torque based on the battery power requirement for the transition. This value is determined using empirical data in the form of a lookup table, however the performance data of the electric motor supplied by the manufacturer was in the format of power loss as a function of torque and speed. This data was converted into the electric motor torque lookup table using the following process.

First the speed and torque breakpoints of the empirical loss data were used to populate a table of mechanical power. Using the power loss data provided by the electric motor manufacturer as well as the mechanical power calculated from that data's breakpoints, the electrical power at the breakpoints of the power loss data can be determined through the following relationship:

$$P_{elec}(\tau_{c,mot}, \omega_{mot}) = P_{mech}(\tau_{c,mot}, \omega_{mot}) + P_{loss}(\tau_{c,mot}, \omega_{mot}) \quad (4.32)$$

where P_{elec} is the electrical power, P_{mech} is the mechanical power, and P_{loss} is the power loss. Each as a function of electric motor component torque $\tau_{c,mot}$ and speed ω_{mot} . With the electrical power table assembled, the breakpoints for the torque lookup table must be defined. As electric motor speed will remain as an input to the final lookup table, the identical breakpoints to the original power loss table are used. The power breakpoints for the table are determined from the electric power table by selecting the corresponding power for the maximum positive and negative torques at every given speed. Additional power breakpoints are arbitrarily added at power levels below those determined by the maximum mechanical power to fill the lookup table with more intermediate values. An algorithm is developed to perform a 1D lookup for each of the power values breakpoints for a given vehicle speed. The resulting table generated from this algorithm is a two-dimensional lookup table that identifies the electric motor torque as a function of battery power and electric motor speed.

It is critically important to note that based on this structure, positive torque and regenerative braking setpoints are selected. In automotive vehicle development, the method for decelerating the vehicle with regenerative braking capabilities is referred to in two ways: blended and over the top braking. Blended braking allows the electric motor and mechanical brake controller to communicate and determine the most efficient braking strategy. Over the top braking is where the amount of braking supplied by the mechanical brakes is fixed and the amount of regenerative

braking that occurs on top of that can change. In the EMC, teams are only allowed to apply over the top braking capabilities, however, due to the inability to model the driver brake pedal inputs, the DP model implements a blended braking system. This is a critical deviation from the capabilities of the EcoCAR team competition vehicle and thus, the regenerative braking commands should not be considered as part of the energy management strategy design.

4.3.3 Dynamic Programming Algorithm

The following steps describe the operation of the developed DP algorithm. In this model, i indicates the state grid position at k , while j indicates the state grid position at stage $k + 1$. The number of stages is indicated by T . To initialize the model, the transition cost from $k = N - 1$ to $k = N$ is calculated. The transition costs for each stage to every other stage are saved in the cost-to-go matrix, $g(i, j)$. Figure 20 is included as a visual representation of this initialization step in the benchmarking model.

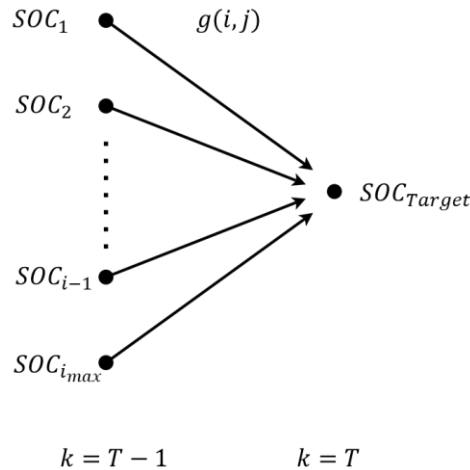


Figure 20: Initial Stage Calculation of the Benchmarking Model

In this initial stage, the transition cost from each SOC in the state grid to the target SOC is calculated as follows.

4.3.3.1 Electric Powertrain Model

Rearranging the state transition equation to solve for battery power yields the following:

$$P_{batt}(i,j) = \frac{[SOC(j) - SOC(i)] BatteryCapacity}{\Delta t} \quad (4.33)$$

In the initial stage, j represents the target SOC while i is each SOC in the state grid. Note that a negative battery power is a discharge from a higher SOC (at i) to a lower SOC (at j). The physical constraints of the electrical system (ESS and P4 motor) in the WVU team competition vehicle are matched at a power limit of 50 kW. Because of this pairing, it is convenient to impose the electrical system power limit constraints on the battery power in the following piecewise equation:

$$P_{batt} = \begin{cases} NaN & \text{if } P_{batt} \leq P_{batt,min} \\ P_{batt} & \text{if } P_{batt,min} < P_{batt} < P_{batt,max} \\ NaN & \text{if } P_{batt,max} \leq P_{batt} \end{cases} \quad (4.34)$$

where $P_{batt,min}$ and $P_{batt,max}$ are the minimum and maximum battery power limits, respectively.

The value “NaN” is useful in the DP model as it simplifies the removal of infeasible solutions as the value of “NaN” will persist through any mathematical operation and thus the transition will never be considered by the DP algorithm. With the entirety of the drive cycle known *a priori*, the rotational speed of the electric motor is determined from the rotational speed of the wheels of the vehicle as follows:

$$\omega_{mot} = \omega_{wheel} * GR_{diff,P4} \quad (4.35)$$

where ω_{mot} is the P4 electric motor speed at stage k and $GR_{diff,P4}$ is the differential gear ratio of the P4 differential. It is important to note that the speed of the electric motor is independent of the state and thus takes a single value at each stage. The torque produced by the electric motor is determined by a lookup table generated from the powerloss data provided by Magna for the eRAD P4 electric motor as discussed in Section 4.3.2. The electric motor component torque produced for

the transition from state i to state j , $\tau_{c,mot}$, is a function of the battery power and electric motor speed for that same state transition:

$$\tau_{c,mot} = f(-P_{batt}, \omega_{mot}) \quad (4.36)$$

Note that the negative sign is used to preserve the battery power convention where negative power represents power discharged from the battery to produce positive propulsive torque from the electric motor.

4.3.3.2 Conventional Powertrain Model

With the produced motor torque known and the vehicle wheel torque requirements known *a priori*, the remaining torque to be produced by either the ICE or friction brakes is determined as follows:

$$\tau_{w,remain} = \tau_{w,req} - (\tau_{c,mot} * GR_{diff,P4}) \quad (4.37)$$

where $\tau_{w,remain}$ is the remaining wheel torque after subtracting the wheel torque of the electric motor and $\tau_{w,req}$ is the required wheel torque given by the roadload equation.

The modeling of the conventional powertrain starts with the speed of the torque converter. The torque converter model utilized in the benchmark analysis is of very low fidelity to accommodate the runtime and memory considerations of the model. The turbine speed of the torque converter is propagated from wheel speed through the differential and transmission as follows:

$$\omega_{turb} = \omega_{wheel} * GR_{trans} * GR_{diff,ICE} \quad (4.38)$$

where ω_{turb} is the turbine speed of the torque converter. The locking state of the torque converter is then determined on the condition that the turbine is spinning faster than the target idle speed ($\omega_{idle,ICE}$) of the engine as well as the clutch status:

$$Stat_{TC,Lock} = \begin{cases} 0 & \text{if } \omega_{turb} < \omega_{idle,ICE} \\ 1 & \text{if } \omega_{idle,ICE} \leq \omega_{turb} \end{cases} \quad (4.39)$$

$$Stat_{TC,clutch} = \begin{cases} 0 & \text{if } \tau_{w,req} = 0 \\ 1 & \text{if } \tau_{w,req} \neq 0 \end{cases} \quad (4.40)$$

where $Stat_{TC,Lock}$ is the status of the torque converter lockup condition for the transition from state i to state j in gear N_{gear} with a value of 0 corresponding to the torque converter being unlocked and 1 being locked. $Stat_{TC,clutch}$ is the status of the torque converter clutch with 1 corresponding to the clutch engaged and 0 being disengaged. The speed of the ICE is then determined from the following piecewise function:

$$\omega_{ICE} = \begin{cases} \omega_{ICE,idle} & \text{if } Stat_{TC} = 0 \\ \omega_{turb} & \text{if } Stat_{TC} = 1 \end{cases} \quad (4.41)$$

where $\omega_{idle,ICE}$ is the target idle speed of the ICE. The speed ratio of the torque converter is determined by dividing the turbine speed by the impeller (ICE shaft) speed as shown:

$$\phi_{TC} = \frac{\omega_{turb}}{\omega_{c,ICE}} \quad (4.42)$$

where ϕ_{TC} is the speed ratio of the torque converter. Using data provided by GM, the torque ratio can be determined as a function of speed ratio by implementing a 1D lookup table:

$$\psi_{TC} = f(\phi_{TC}) \quad (4.43)$$

The remaining wheel torque is then transformed into a component torque request for the internal combustion engine by multiplying by the transmission gear ratio, differential gear ratio, transmission efficiency, and torque ratio, GR_{trans} , $GR_{diff,ICE}$, η_{trans} , and ψ_{TC} respectively:

$$\tau_{c,remain} = \tau_{w,remain} * GR_{trans}(N_{gear}) * GR_{diff,ICE} * \eta_{trans}(N_{gear}) * \psi_{TC} \quad (4.44)$$

As shown in Equation (4.44), the component torque commands for the engine become a function of transmission gear ratio given by the index N_{gear} in addition to the indexes i and j . This additional dimension captures the proportional component torque requirement from the engine and transmission combination to the wheel torque request for each state transition.

Next, a series of 1D lookup tables are used to identify the maximum and minimum admissible operating points of the ICE where the speed of the ICE is the independent variable as follows:

$$\tau_{c,ICE,max} = f_1(\omega_{ICE}) \quad (4.45)$$

$$\tau_{c,ICE,min\ run} = f_2(\omega_{ICE}) \quad (4.46)$$

$$\tau_{c,ICE,min\ FCO} = f_3(\omega_{ICE}) \quad (4.47)$$

where $\tau_{c,ICE,max}$ is the maximum torque of the ICE, $\tau_{c,ICE,min\ run}$ is the minimum running torque of the ICE, and $\tau_{c,ICE,min\ FCO}$ is the fuel cut off torque of the ICE. The $\tau_{c,ICE,max}$ lookup table is based on the maximum torque curve for the 2.5L LCV engine, $\tau_{c,ICE,min\ run}$ is the torque required to sustain idle for the engine, and $\tau_{c,ICE,min\ FCO}$ is the fuel cut off (FCO) torque at which fuel flow to the engine is zero. The conditions based around these parameters are that the ICE cannot exceed the maximum torque for a given speed, the ICE cannot operate below minimum running torque while the torque converter clutch is disengaged, and the ICE cannot operate below FCO torque while the torque converter clutch is engaged. If the calculated remaining ICE torque exceeds the maximum component torque, the ICE component torque is automatically assigned the value NaN. The ICE component torque produced is implemented in the MATLAB model as a piecewise function, but due to the length of the function, a flow diagram is instead used to show the logic implemented to determine this value. The ICE torque determination flow diagram is shown in Figure 21.

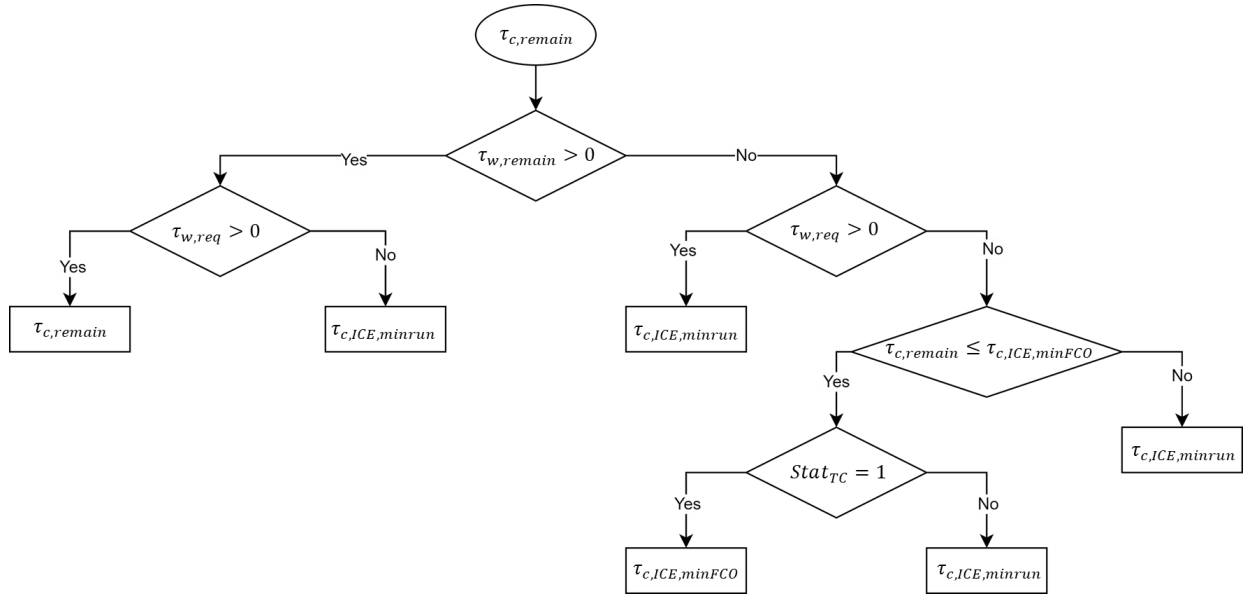


Figure 21: ICE Torque Determination Logic

The ICE torque determination logic implements two critical constraints to the operation of the ICE. The first is that positive torque required from the ICE at wheel speeds below idle require the unlocked dynamics of the torque converter to be considered. Essentially, the torque produced by the engine will be less than the request due to the torque multiplication of the torque converter in an unlocked state. The other constraint is that the ICE cannot operate above its maximum torque.

4.3.3.3 Additional Constraints

With the operational points of the electric motor and engine known, two additional physical constraints must be imposed. The first is the requirement for mechanical braking to only apply “negative” torque to slow the vehicle down. Before calculating the braking torque, the component torques for both the electric motor and ICE must be converted to wheel torques by propagating the torques and speeds through each axle’s driveline. The conversion from component to wheel torque for the P4 electric motor and ICE are as follows, respectively:

$$\tau_{w,mot} = \tau_{c,mot} * GR_{diff,P4} \quad (4.48)$$

$$\tau_{w,ICE} = \tau_{c,ICE} * GR_{diff,ICE} * GR_{trans} * \psi_{TC} * \eta_{trans} \quad (4.49)$$

The braking torque is then determined from the following relationship:

$$\tau_{w,brk} = \tau_{w,req} - \tau_{w,mot} - \tau_{w,ICE} \quad (4.50)$$

The second is the required drive cycle torque not met. This constraint is mathematically expressed by the following equation:

$$\tau_{w,req} = \tau_{w,mot} + \tau_{w,ICE} + \tau_{w,brk} \quad (4.51)$$

A violation of these constraints is imposed on the ICE torque value by assigning the location of the violation a “NaN”. This is done due to the convenience of further calculations being able to handle NaN values by simply skipping over them.

4.3.3.4 Cost Function

With the operational points of the engine known, the fuel flow rate can be determined by interpolating the fuel flow rate map as follows:

$$\dot{m}_{fuel} = f(\tau_{c,ICE}, \omega_{ICE}) \quad (4.52)$$

Where \dot{m}_{fuel} is the fuel flow rate for the torque ($\tau_{c,ICE}$) and speed (ω_{ICE}) for the transition from state i to state j in the gear N_{gear} . With the fuel flow rate known, the cost to go matrix can be constructed. The DP algorithm is designed to assess the performance of the powertrain for both a defined shift schedule as well as the additional dimension for gear selection. For the second case, the famous curse of dimensionality for DP algorithms becomes an issue. To address this, an intermediate minimization is applied to identify the most efficient gear for each transition from i to j which effectively removes the additional dimension N_{gear} . This is accomplished by utilizing the following:

$$g_{gear}(i, j, N_{gear}) = \dot{m}_{fuel} \quad (4.53)$$

where $g_{gear}(i, j, N_{gear})$ is the intermediate cost-to-go matrix for the transition from state i to state j in the gear N_{gear} . The final cost-to-go matrix is found by identifying the minimum of the intermediate cost-to-go matrix for a given transition from i to j , effectively minimizing along the axis of the gear number. The cost-to-go matrix is formulated as follows:

$$g(i, j) = \min (g_{gear}) \quad (4.54)$$

where $g(i, j)$ is the cost-to-go matrix from state i to state j . If the shift schedule is predefined, the final cost to go matrix, $g(i, j)$, will simply be equal to the intermediate cost to go matrix, $g_{gear}(i, j, N_{gear})$. Finally, the total path cost for the initial step of the DP algorithm is simply equal to the cost-to-go matrix for every admissible transition from state i to j :

$$J(i, k) = g(i, k) \quad for \quad k = T - 1 \quad (4.55)$$

4.3.3.5 Remaining Stages of DP Algorithm

This concludes the initial stage of the DP algorithm. For each remaining stage of the DP algorithm ($k = T - 1, \dots, 1$), Equations (4.33) through (4.54) are carried out for every transition from state i to state j as shown in Figure 22.

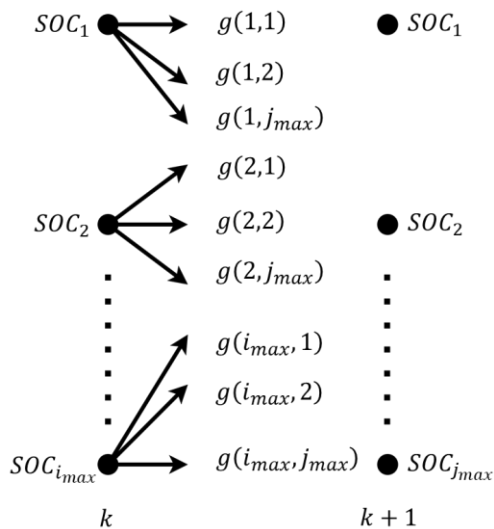


Figure 22: Intermediate Stage Calculations of the DP Algorithm

For intermediate stages ($k = T - 2, \dots, 1$) the total path cost, $J(i, k)$, is calculated according to Bellman's Principal of Optimality as described in Section 4.1.2 by the following relationship:

$$J(i, k) = \min [g(i, j) + J(j, k + 1)] \tag{4.56}$$

Using this relationship, the total path cost is minimized by identifying the combination of the current cost-to-go from state i to state j and the total path cost leading to that transition with the lowest cost at stage k . This method can be visualized as shown in Figure 23.

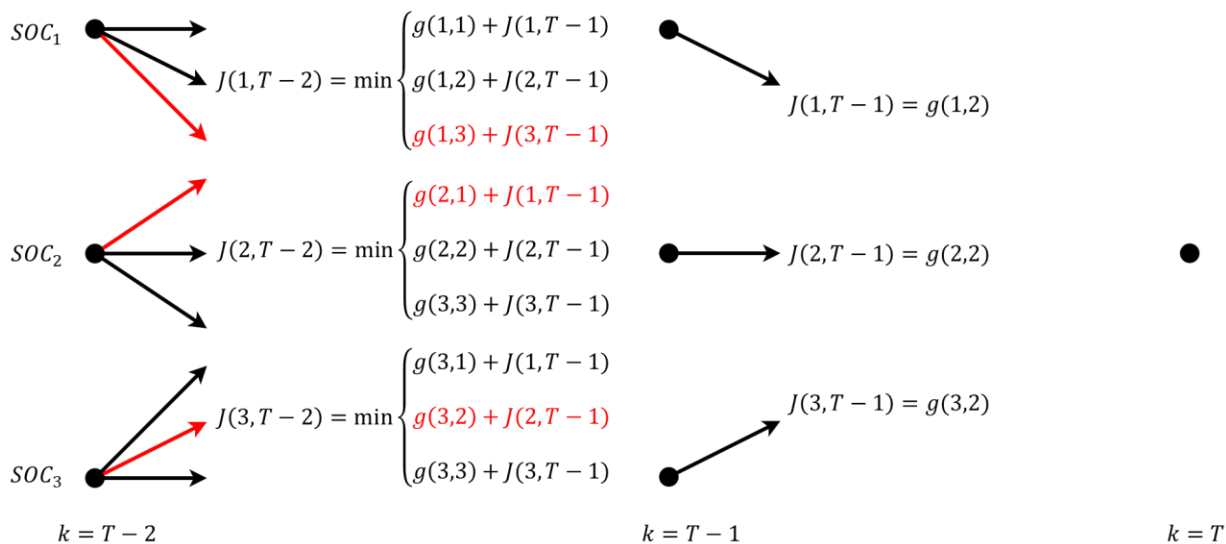


Figure 23: Example of Intermediate Stages of the DP Algorithm at Stage $k = T - 2$

The red path indicates the selected transition path with the minimum total cost. This method implies that if at stage $k = T - 2$ the optimized control actions transition from the state SOC_3 to state SOC_2 , then at an earlier stage $k = T - 3$, any transition from state SOC_i to SOC_3 , the optimal control policy will include the previously determined control actions taken at stage $k = T - 2$. This methodology is visualized in the continued example shown in Figure 24.

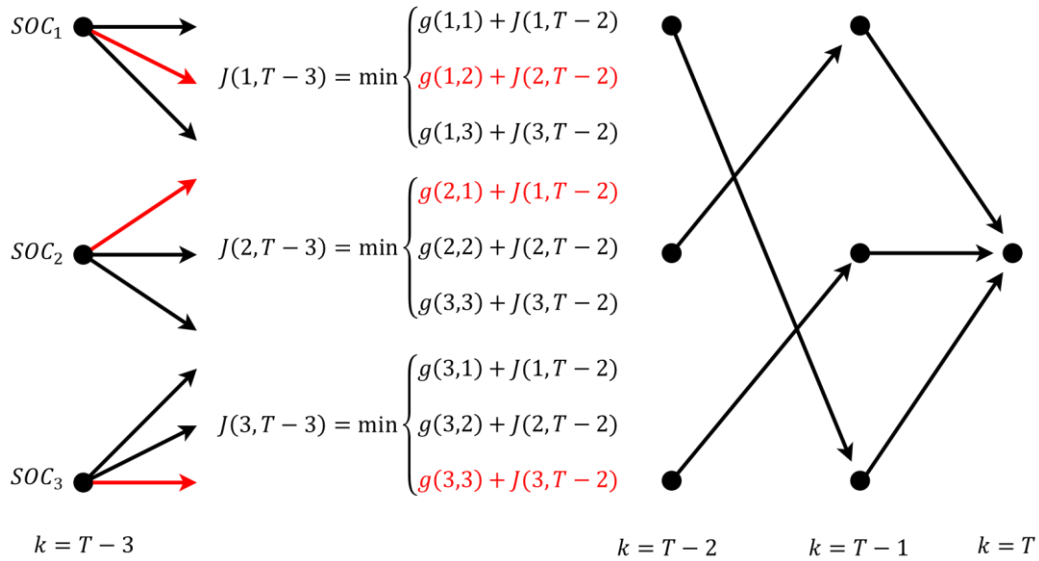


Figure 24: Example of Intermediate Stages of the DP Algorithm at Stage $k = T - 3$

After reaching the final stage at $k = 1$, the total cost and control actions that constitute the global optimal solution have been calculated. In contrast with the final time step, the starting SOC is difficult to enforce charge sustaining criteria. In this work, CS operation is imposed directly by selecting the initial state to be equal to the CS target SOC then following the optimal policy determined by the DP algorithm to the terminal SOC.

4.3.3.6 Known Obstacles in DP Algorithms

Due to the nature of DP, constraints are typically imposed by assigning high costs to infeasible solutions, or in the case of this work, costs that do not exist and are thus never considered. A problem that is known to occur in DP algorithms called “leaking” or the large propagation of large

penalties into the path of the optimal solution [46]. This phenomenon can be visualized as shown in Figure 25.

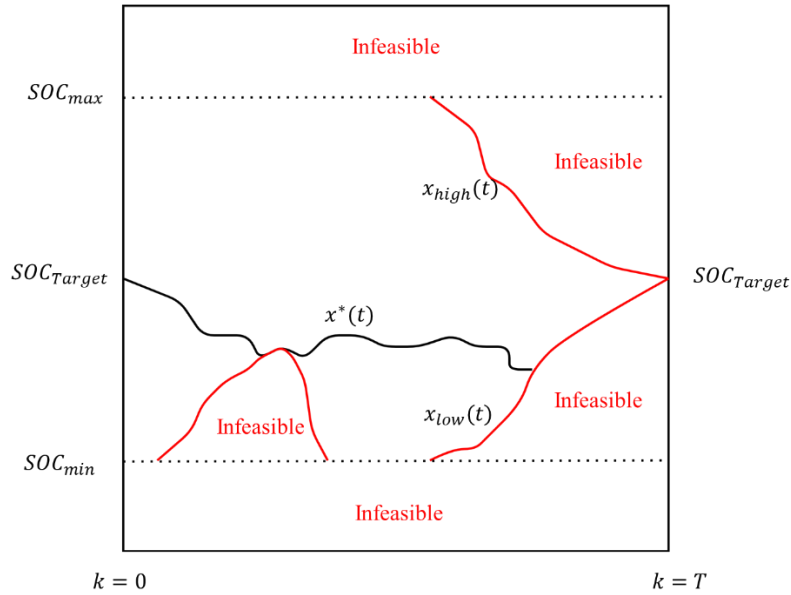


Figure 25: Leaking Effect in DP Algorithm

Leaking may cause the optimal solution trajectory to make bad decisions to avoid crossing the boundary into the infeasible region. Figure 25 shows the infeasible boundary receding from $k = T$ until reaching the low and high thresholds of SOC as the DP algorithm is solved backwards in time. In this example the infeasible boundary “leaks” back into the state space before receding back to the minimum SOC boundary. An example of why this may happen is because at the minimum SOC boundary, the torque requirement is so large that the ICE cannot satisfy the requirement on its own. Because of this, the cost of this transaction becomes massive. Now for subsequent points at the following stage, the previous infeasible point may be the only admissible transition although the cost is large. This has a cascading effect that causes the infeasible boundary to “leak” into the state space between the SOC boundaries until conditions (torque requirements) allow feasible transitions to push the infeasible boundary back out.

Another effect can also be recognized between feasible solutions where the distribution of total cost is such that for each step, every state i will transition to state j if admissible. This effect is referred to as uniform state transition where regardless of initial state, the solution will make poor decisions to reach the states associated with the optimal policy due to the collapse of different potential solutions. The uniform state transition effect can be visualized as shown in

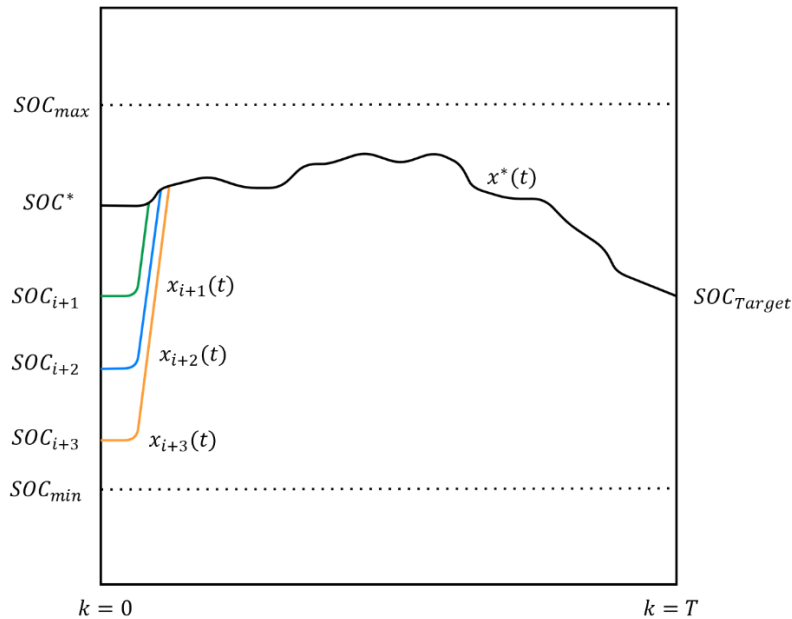


Figure 26: Uniform State Transition Effect in DP Algorithm

These two effects can be identified by what looks like discontinuities in the optimal solution determined by the DP algorithm.

5.0 Results and Discussion

This section outlines the results of the benchmark analysis using the DP algorithm. Included is the verification of successful operation and fuel consumption values determined by the DP algorithm. The results of the DP algorithm are validated against a higher fidelity Simulink model to identify where the simplified DP model significantly differs from the high-fidelity model it was derived

from. The results of the DP algorithm are discussed for both the predetermined shift schedule of the GM M3D transmission and intermediate minimization of gear selection problem.

5.1 Model Verification

Due to the complexity of HEV powertrain systems, the development of a powertrain model and the determination of an optimal performance solution analytically were not possible without making significant assumptions and simplifications. Drive cycles developed for the EcoCAR Mobility Challenge and the automotive industry are typically relatively long and include complex driving scenarios. To reduce the complexity of the verification process, a short verification cycle was developed where the optimal solution could be reasonably predicted. During the process of verification, aspects of the DP algorithm are modified to force the algorithm into specific conditions.

5.1.1 Verification Cycle

The verification cycle starts with the vehicle at rest where the operation of the ICE in idle can be verified. The cycle continues with a segment of constant acceleration to 65 mph, then a segment of sustained cruising at 65 mph, then a constant deceleration event, resulting with the vehicle at rest. A second acceleration to 25 mph, sustained cruise at 25 mph, and deceleration is defined for the second half of the cycle. The total distance of the verification cycle is 0.79 miles. The vehicle speed, wheel torque and wheel power of the verification cycle as determined by the are shown in Figure 27.

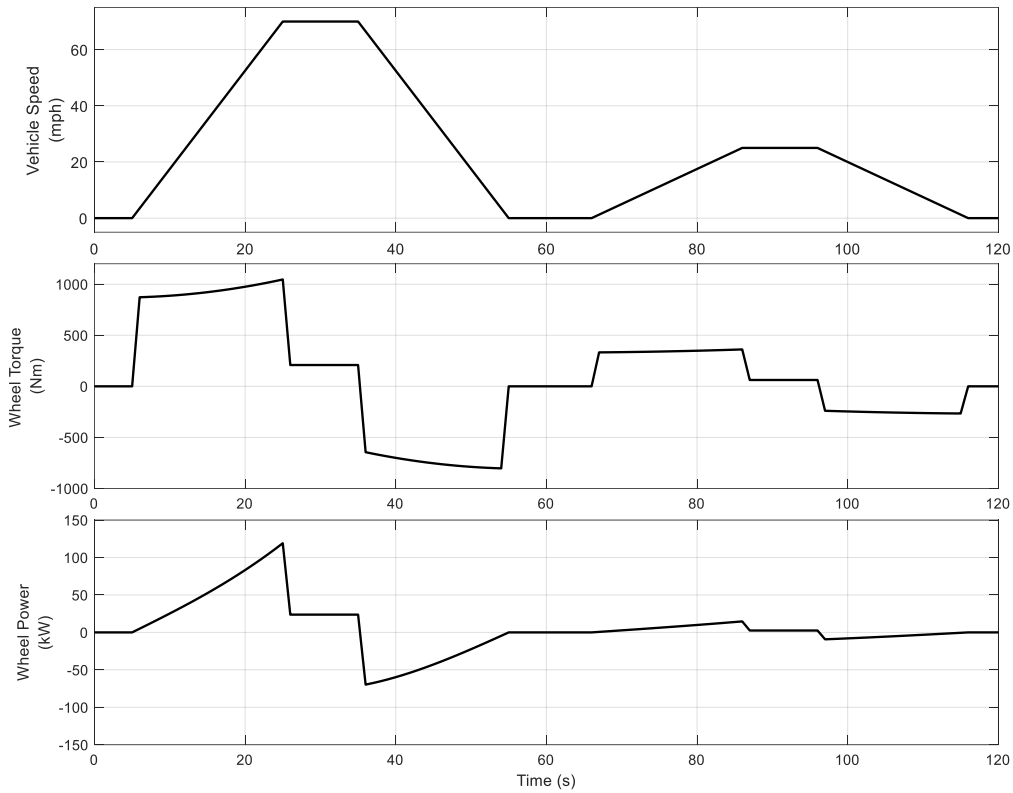


Figure 27: Verification Drive Cycle Data

The validation cycle captures one aggressive driving acceleration where the driver demand is high for an acceleration to 65 mph and one modest acceleration to 25 mph. The deceleration events are similar with the 65-0 mph deceleration being more aggressive while the 25-0 mph is more conservative. The two cruise scenarios are similar with the major difference being the amount of sustained power required for each. The shift schedule for the verification test is shown in Figure 28.

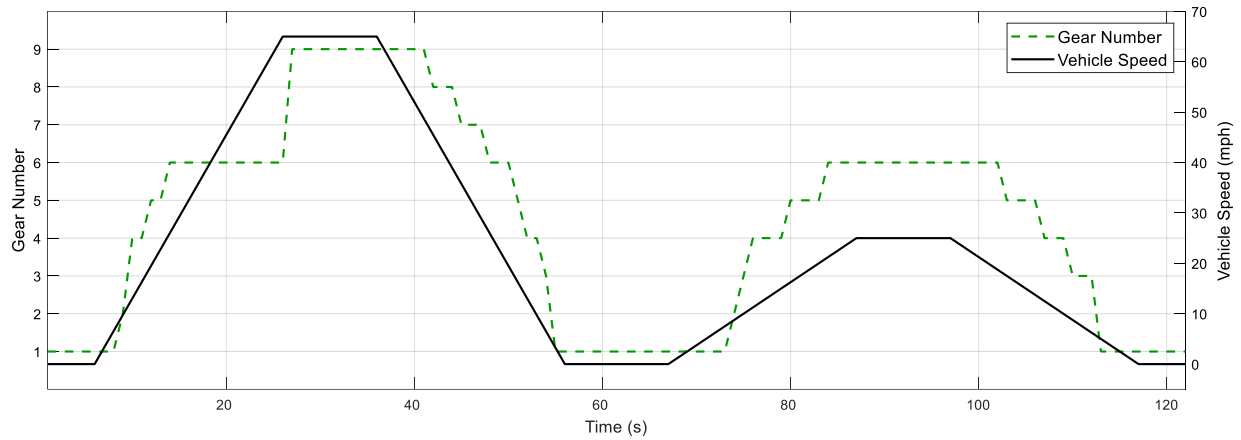


Figure 28: Shift Schedule of Verification Drive Cycle

The shift schedule is derived based on the pedal map and shift logic of the 2.5L LCV engine and M3D transmission combination. It should be noted that this shift schedule is discretized at the same 1s timestep as the drive cycle and thus for higher fidelity simulations and real-world situations the shift schedule for this same cycle may vary.

5.1.2 Verification Tests

To first verify the capabilities of the DP algorithm to determine optimal routes, the algorithm was tested against the trivial solutions of the HEV optimal control in two scenarios. The first scenario is conventional vehicle operation where the use of the electric powertrain is highly penalized and thus the required torque requirement for the drive cycle is supplied by only the ICE. The second scenario is CD operation of the HEV. In this scenario, the electric motor is utilized as much as possible to meet the required torque of the drive cycle. For these tests, the grid spacing of the state space is reduced to achieve faster runtime as the fidelity of the results is less important in the verification analysis. The summary of DP algorithm operating parameters are shown in Table 1.

Table 1: Verification Test DP Algorithm Parameters

Parameter	Value
maximum SOC	60%
minimum SOC	30%
number of grid points	2000
Δ SOC	0.015 %
ΔP_{batt}	810 W

5.1.2.1 ICE Only Verification

The ICE only verification test is conducted to verify that the conventional powertrain is capable of following the verification drive cycle without the help of the electric motor. In the real world 2019 Chevrolet Blazer, the 2.5L LCV and M3D transmission is a commercially available front wheel drive powertrain option and thus the engine and transmission pair are appropriately sized to provide adequate propulsion to satisfy the driving cycle. It is important to note that due to the increase of mass from electric powertrain, the conventional vehicle would have better performance than the ICE only model evaluated in this test. ICE only operation is achieved in the DP algorithm by modifying the cost function to penalize the use of the electric motor as follows:

$$g(i, j) = \min (\dot{m}_{fuel} + |\tau_{c,mot}|) \quad (5.1)$$

From this constraint, the DP algorithm will select the lowest amount of fuel consumed without any torque supplied from the electric powertrain. It should be noted that a scaling factor could potentially be necessary to ensure that the penalty on the electric motor torque outweighs the cost of fuel, however with fuel consumed in grams and motor torque in Nm and the large ΔP_{batt} from the discretized state grid, the penalty for electric motor use outweighs the penalty of fuel consumed. The powertrain power outputs for the ICE only verification test is shown in Figure 29.

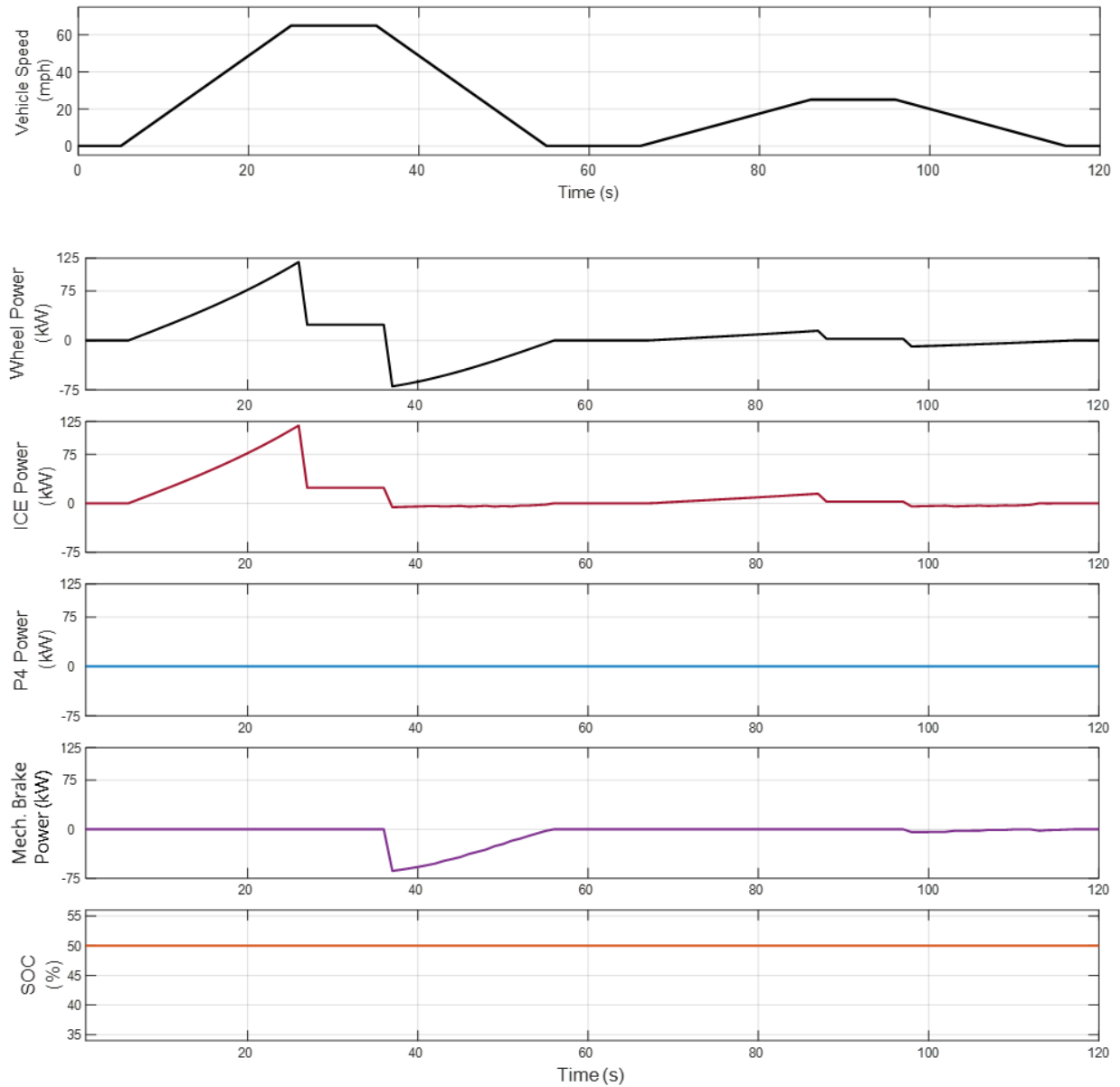


Figure 29: ICE Only Verification Test Powertrain Power Output

In this figure, the vehicle speed and wheel power requirements are shown with the associated power produced by the ICE, P4 electric motor, mechanical friction brakes and the corresponding SOC of the ESS. The results of the ICE only verification test show that the algorithm selected the solution that used the ICE to meet exactly the positive torque requirements and the combination of ICE drag torque and mechanical brakes to satisfy the negative torque requirements of the

verification cycle. A summary of the energy analysis results for the ICE only verification test is shown in Table 2.

Table 2: ICE Only Verification Test Energy Summary

Parameter	Value
Fuel Economy (MPG)	16.86
Fuel Used (g)	132.07
Total Fuel Energy (MJ)	5.67
Total Mechanical Energy (MJ)	2.44
Total Electrical Energy (MJ)	0

As the entirety of mechanical energy is provided by the ICE, a significant amount of fuel is consumed in this test. It should be noted that due to the short length and aggressive driving captured in the verification drive cycle, the fuel economy is low when compared to the stock 2019 Chevrolet Blazer with the 2.5L LCV engine. Additionally, the total electrical energy is zero, which for charge sustaining operation will also be true, but for this scenario, the SOC never changes because the electric motor is never used. The mechanical losses of the P4 gearbox are not considered as the Magna eRAD unit contains a clutch that effectively allows the rear axle to free spin if the P4 motor is not engaged. These results lay the baseline of the lowest “optimal” fuel economy for the hybridized vehicle as the optimal use of the electric motor will improve the fuel economy.

Although the fidelity of the DP algorithm is reduced for the verification tests, the ICE was able to achieve exactly the torque requirements for the cycle. The structure of the algorithm calculates ICE torque based on the remaining torque after subtracting the produced torque by the electric motor. In this scenario, the optimal operating points are zero torque from the electric motor and

thus the torque requests are identical to the required torque of the drive cycle. This benefit is not present in the CD verification test.

5.1.2.2 CD Verification

With the selected gear number fixed based on the method described in Section 4.3.1, the DP algorithm has demonstrated the ability to select the optimal control of the conventional powertrain, the next step is to implement the electric drivetrain and assess the improvements. The DP algorithm relies on the CS constraint to select an optimal control policy that differs from the trivial solution of motor only operation. By removing the CS constraint, the optimal control policy would be to use the electric motor to provide the propulsive torque and keep the engine operating at a low idle.

As described in Section 4.3.3.5, the CS constraint is imposed in the forward-looking analysis after the total path cost is calculated for each point in the state grid. The CS constraint is imposed by manually selecting the initial point to be the CS SOC and following the minimum total path costs to the resulting CS SOC at the terminal point of the algorithm. By selecting the initial point to be the minimum total path cost in the state grid, the trivial (CD) solution is selected. The results of the CD verification test are shown in Figure 30.

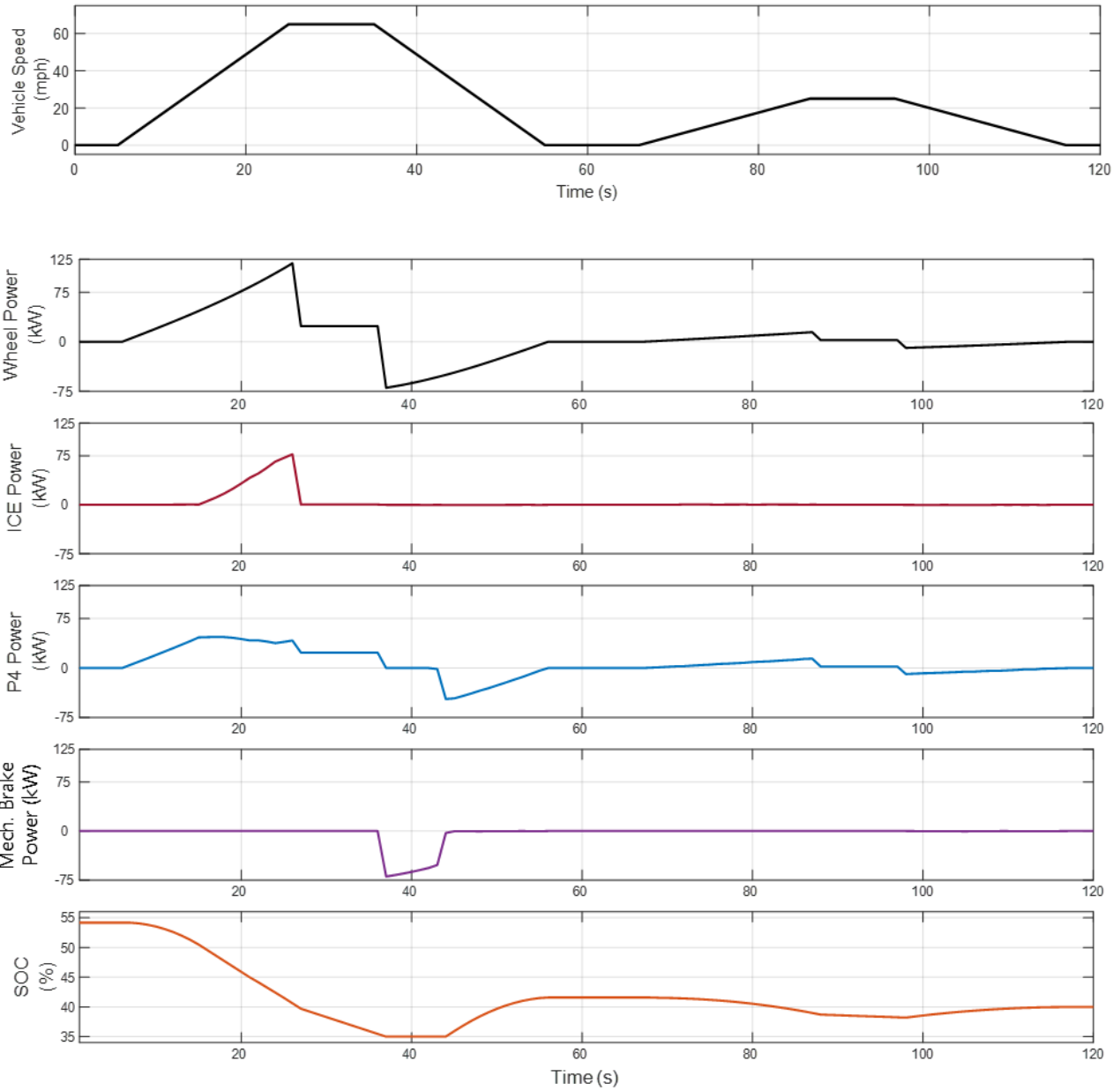


Figure 30: CD Verification Test Powertrain Power Output

The control policy begins by using the electric motor for all propulsive torque, but the upper torque limit of the motor is reached requiring the ICE to contribute. After reaching the cruise portion of the drive cycle, the motor once again takes over. During the first braking event, the mechanical brakes are used to meet the torque requirement initially before the electric motor commands negative torque to capture regen. For the remainder of the drive cycle, the electric motor is able to

provide the required torque. This test verifies both the DP algorithm’s ability to select the optimal solution for the hybrid powertrain operation as well as the successful calculation of motor operation and constraints. A summary of the energy analysis results for the CD verification test are shown in Table 3.

Table 3: CD Verification Test Energy Summary

Parameter	Value
Fuel Economy (MPG)	29.65
Fuel Used (g)	75.10
Total Fuel Energy (MJ)	3.22
Total Mechanical Energy (MJ)	2.44
Total Electrical Energy (MJ)	0.76

As this test constituted CD operation, the SOC dropped over 14% from the initial state to the final state resulting in a significant amount of electrical energy depleted. It is critically important to note that the reported fuel economy is not adjusted for the energy used during the test. The goal of this verification test was to allow the DP algorithm to pick the control policy that utilized the electric motor to follow the required torque for the drive cycle. Because of this, the ability for the ICE to enter fuel cut off was disabled for this test as the DP algorithm would select the motor torque required to drag the ICE in fuel cut off as much as possible. This makes debugging difficult as the motor and ICE torque fluctuate throughout the control policy. This is relevant because the minimum possible fuel consumption for this powertrain is not identified in this control policy.

Converse to the torque production of the ICE, the electric motor is not likely to produce the exact amount of torque required by the drive cycle. This is due to the direct relationship between the state grid and motor torque as described in Section 4.3.3.1. In the DP test, the electric motor

typically slightly overproduces torque and makes up the difference with the mechanical brakes. By ensuring the state grid is discretized at a fine step size, the error between the required torque and produced motor torque is low enough that the additional use of the mechanical brakes is negligible.

5.1.3 Cost Function Evaluation

With the DP algorithm and powertrain component models verified for realistic operation, the effects of various cost functions are evaluated. In literature, several cost functions have been used with DP algorithms to determine optimal control policies in HEV powertrain energy management strategies.

5.1.3.1 Fuel Only Cost Function

The most common cost function is the minimization of fuel consumption only, as discussed in [2] [25] [45], given by the following function:

$$g(i, j) = \min(\dot{m}_{fuel}) \quad (5.2)$$

where g is the cost-to-go from state i to state j . This cost function identically specifies the goal of the DP algorithm, minimizing the fuel used over the drive cycle. The optimal policy determined for the fuel only cost function is shown to be impacted by leaking and uniform state transition in the results presented in Figure 31.

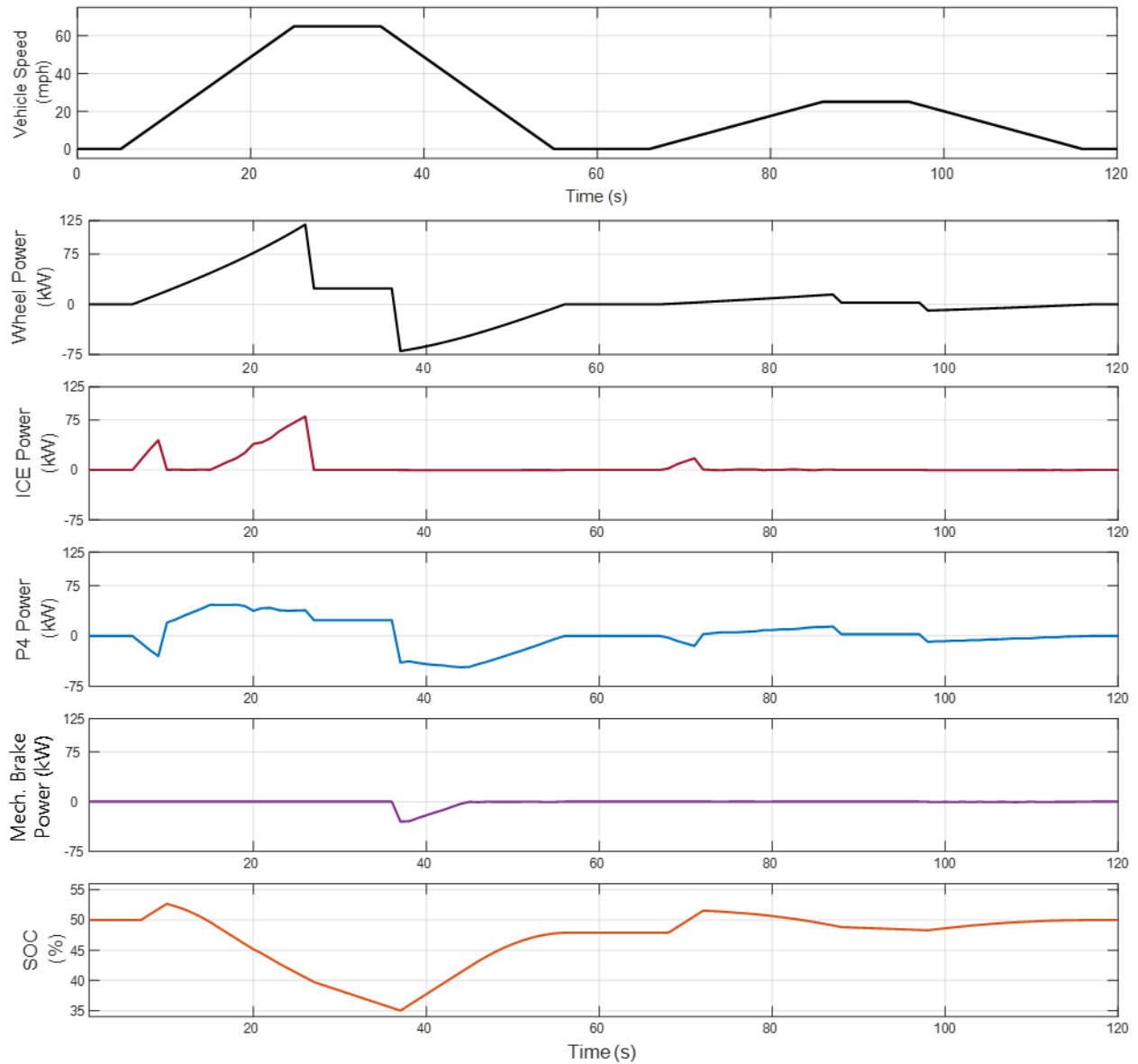


Figure 31: Fuel Only Cost Function Verification Test Powertrain Power Output

The undesired effects discussed in Section 4.3.3.6 can be identified at 8s and 74s into the drive cycle where the ICE power and P4 power spike to counteract each other. Not only are these control actions highly inefficient, but in a real-world application, control actions like these could damage components. Careful inspection of the total path cost matrix shows that the transition cost from i to state j are far outweighed by the total path cost at state j . This forces the solution trajectory in

the direction of the state j with the lowest total cost regardless of the cost-to-go for the transaction from i to j .

Although the proposed solution from the DP algorithm is physically infeasible, the operating points of the powertrain components is shown. As the fuel consumption, power loss, and efficiency maps of both the ICE and P4 motor used in the benchmark analysis are protected by confidentiality agreements, the operating points of these components are shown overlaid with constant power lines and the maximum torque line as a frame of reference. It should be noted that the included line is a publicly available reported maximum torque determined from a 2016 Chevrolet Colorado [47] and differs from the maximum torque used to model the powertrain as described in Section 3.3.2.

It can be stated that the ICE generally uses more fuel as speed and torque increase. However, the most efficient use of fuel energy by the ICE occurs roughly in the region of 70% of maximum torque between 2000 and 4000 RPM reaching a maximum operating efficiency less than 40% with most maximum efficiency points accumulating closer to 35% efficiency. This efficiency diminishes in all directions outside of this envelope. The operating points of the powertrain are shown in Figure 32.

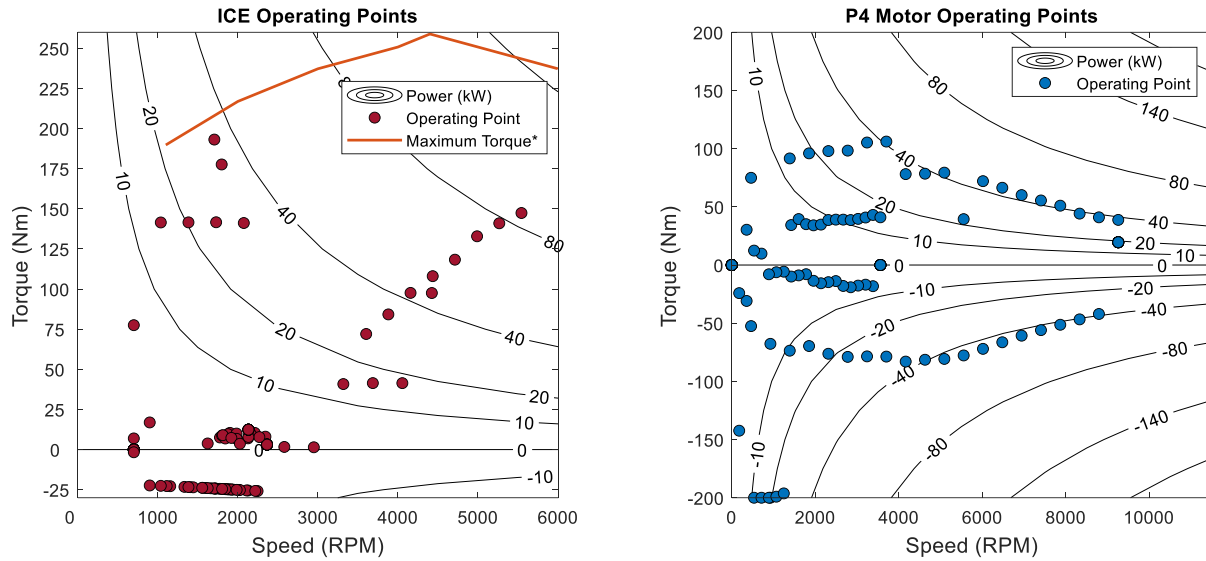


Figure 32: Operating Points of the Powertrain Components for the Fuel Only Cost Function Verification Test

The operating points show the ICE operating in three distinct regions, near idle, high load, and fuel cut off (FCO). Additionally, the electric motor is used significantly throughout the test for both propulsive torque and regenerative braking. The distribution of operating points is useful to understand to identify how the other cost function tests deviate from this solution. The summary of the energy analysis of this solution is shown in Table 4.

Table 4: Fuel Only Verification Test Energy Summary

Parameter	Value
Fuel Economy (MPG)	28.08
Fuel Used (g)	79.30
Total Fuel Energy (MJ)	3.41
Total Mechanical Energy (MJ)	2.44
Total Electrical Energy (MJ)	0

The overall performance of the DP algorithm using the fuel only cost function is desirable from an energy standpoint. The algorithm operated nearly as efficiently as the CD operating mode while

maintaining CS operation. Other cost functions are assessed to address the problems of the fuel only cost function.

5.1.3.2 Quadratic SOC Penalty Factor Cost Function

The use of an added quadratic penalty factor for the SOC location was successfully implemented by Lin *et al.* in [45]. The adoption of this term into the cost function is as follows:

$$g(i, j) = \min(\dot{m}_{fuel} + \alpha(SOC - SOC_{target})^2) \quad (5.3)$$

where α is a tunable weight factor. The quadratic term is used to allow the DP algorithm to select the CS optimal solution without manually selecting an initial value. The results of the verification drive cycle for the quadratic SOC penalty cost function applied to the DP algorithm is shown in Figure 33.

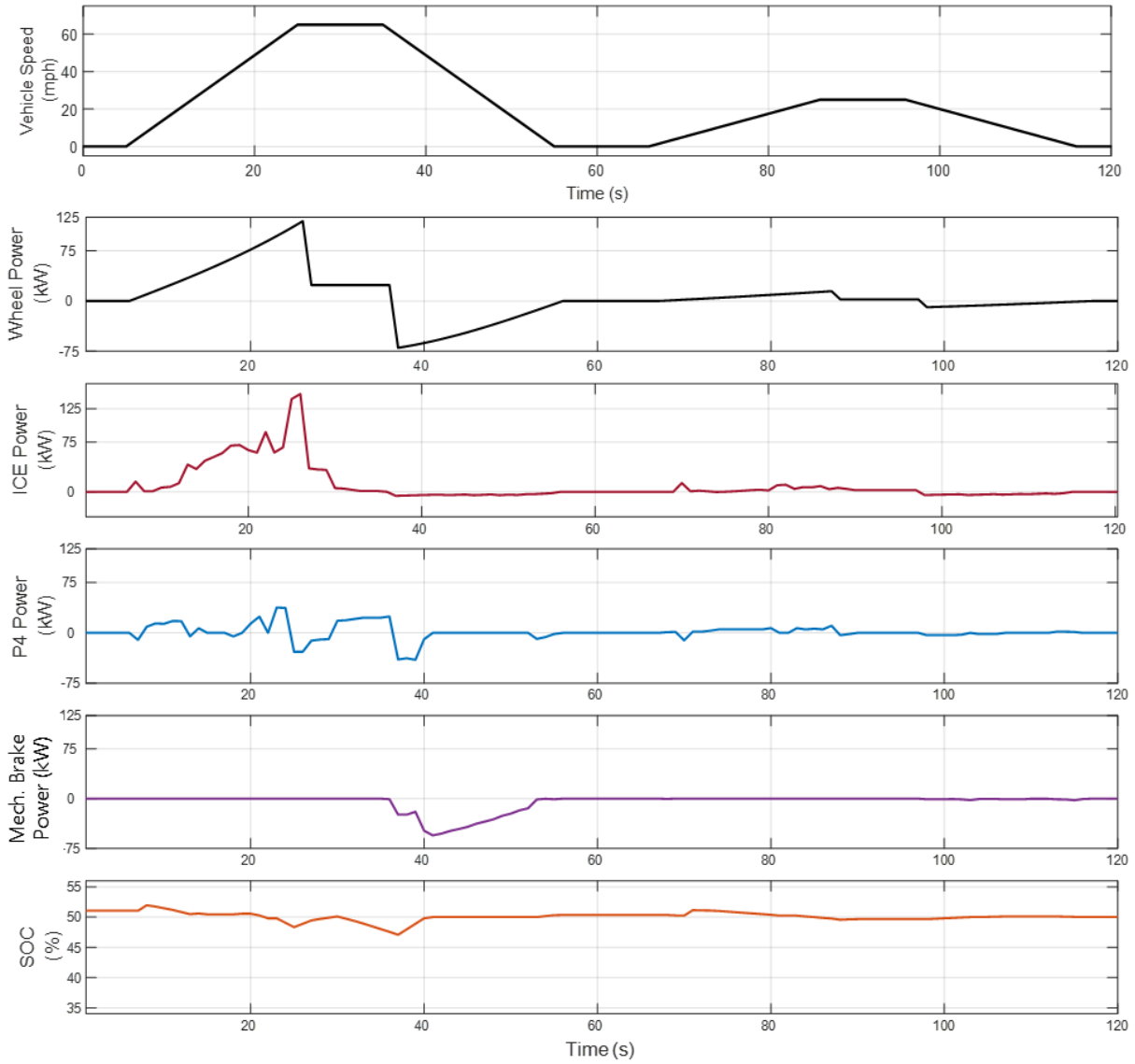


Figure 33: Quadratic SOC Penalty Cost Function Verification Test Powertrain Power Output

The results show that the SOC penalty cost function demonstrated its ability to guide the DP algorithm to a near CS solution. This was achieved through the careful tuning of the weight factor, α via trial and error with results summarized in Figure 34.

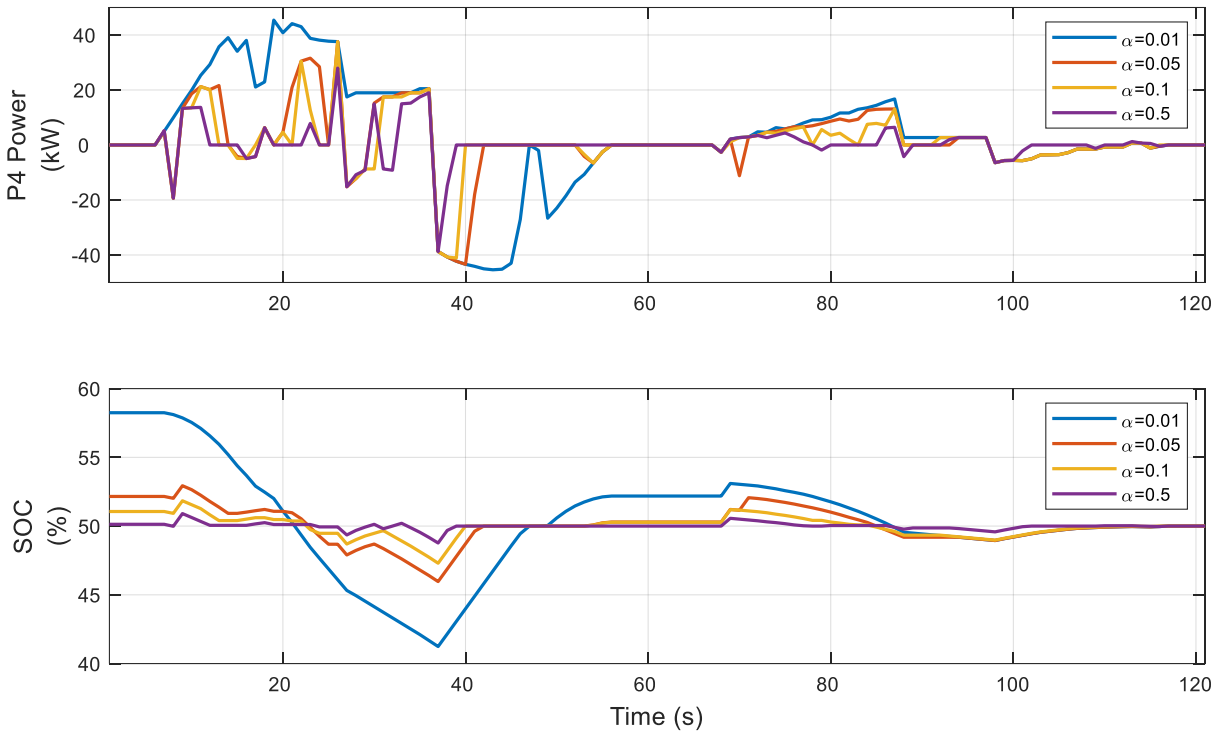


Figure 34: Sensitivity Analysis of Quadratic SOC Penalty Factor Weight Factor

For an α value that is too low, the CS control policy is not determined as shown by the value 0.01, while a large α will lead the DP algorithm to select a solution that never uses the electric motor. In this test, a weight factor of 0.1 was determined to balance CS operation with use of the electric powertrain.

The control policy is still prone to the leaking effect which is still present at 8s and 74s but with less effect when compared to the fuel only cost function. With this cost function, the use of the electric motor is too heavily penalized to allow the motor to provide useful positive torque. This can be identified in Figure 35 where the operating profiles for the ICE and electric motor are shown.

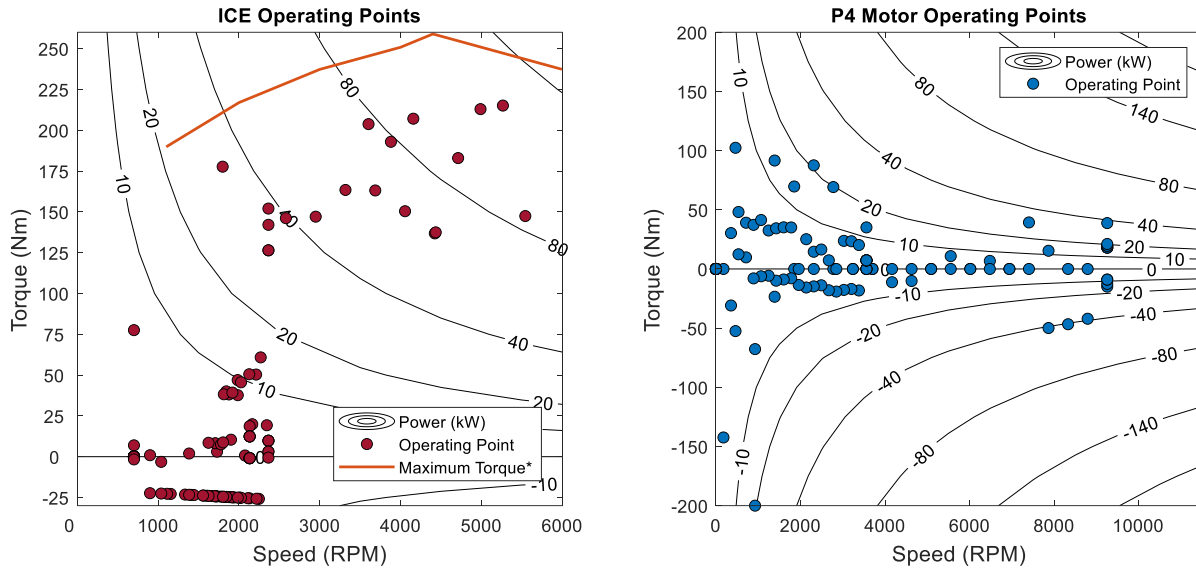


Figure 35: Powertrain Operating Points for SOC Penalty Cost Function Verification Test

Because of the penalty on the use of the electric motor, the ICE is relied on more heavily, especially at low load, thus burning more fuel. This is further expressed in the energy analysis of this verification test as shown in Table 5.

Table 5: Quadratic SOC Penalty Cost Function Verification Test Energy Summary

Parameter	Value
Fuel Economy (MPG)	19.93
Fuel Used (g)	111.73
Total Fuel Energy (MJ)	4.80
Total Mechanical Energy (MJ)	2.44
Total Electrical Energy (MJ)	0.06

The addition of this penalty factor shows only an 18.6% increase in fuel economy over the conventional powertrain only but while using 60 kJ of energy from the battery pack. As the quadratic SOC penalty term did not achieve an appropriate control policy or remove the effects of leaking, other penalty factors were investigated.

5.1.3.3 Motor and Braking Penalty Factor Cost Function

In [10], the use of an alternative penalty term is added to constraint the cost function as follows:

$$g(i, j) = \min(\dot{m}_{fuel} + \gamma(\alpha|\tau_{mot}|) + (1 - \gamma) * \beta|\tau_{brake}|) \quad (5.4)$$

where α and β are tunable weight factors and γ is 1 when the required wheel torque is positive and 0 when the required wheel torque is negative. This cost function is meant to penalize overuse of the electric motor for positive torque requirements and penalizing the overuse of the mechanical brakes for negative torque requirements. The addition of a small penalty for using the electric motor for positive torque is meant to influence the algorithm to select the efficient ICE operational states with the lowest corresponding motor torque. The penalty on the mechanical brakes is imposed to influence the DP algorithm to select solutions that maximize regenerative braking. Various simulations were carried out to tune the weight factors in a sensitivity analysis similar to what was used for the tuning of the weight factor for the quadratic SOC penalty cost function. From this analysis the values of the weight factors were determined to be 0.01 and 0.001 for α and β , respectively. Results for the use of the motor and braking penalty factor with these values for the weight factors are shown in Figure 36.

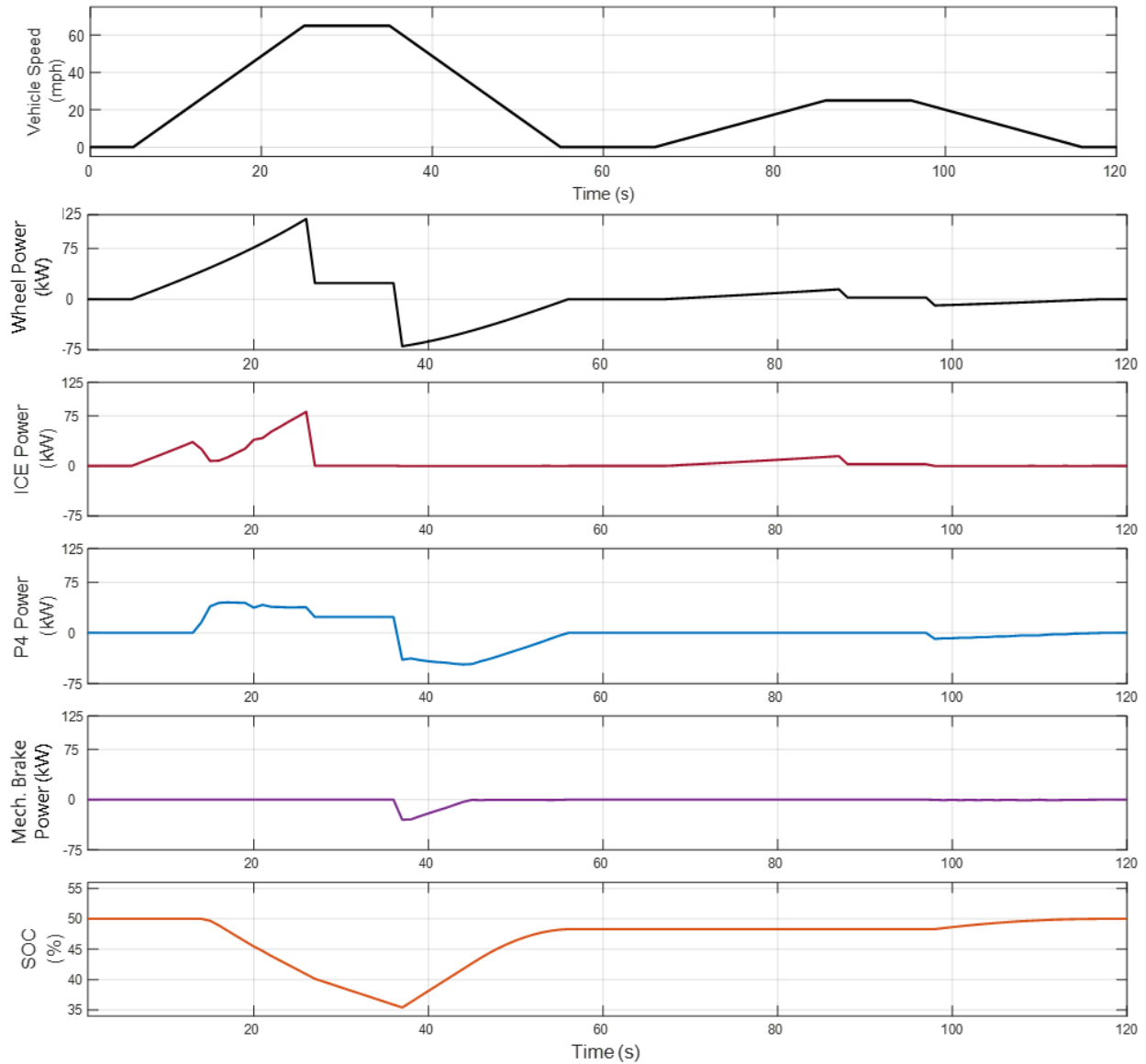


Figure 36: Motor and Braking Penalty Cost Function Verification Test Powertrain Power Output

The motor and braking penalty cost function showed that the impacts of leaking and uniform state transition were present in solutions for low values of the motor torque weight factor. Increasing this value eliminated the undesirable effects of these issues, but also affected the usability of the electric motor when positive torque is required. The operating profiles of the powertrain components shown in Figure 37 give more insight into these drawbacks of the motor and braking penalty cost function.

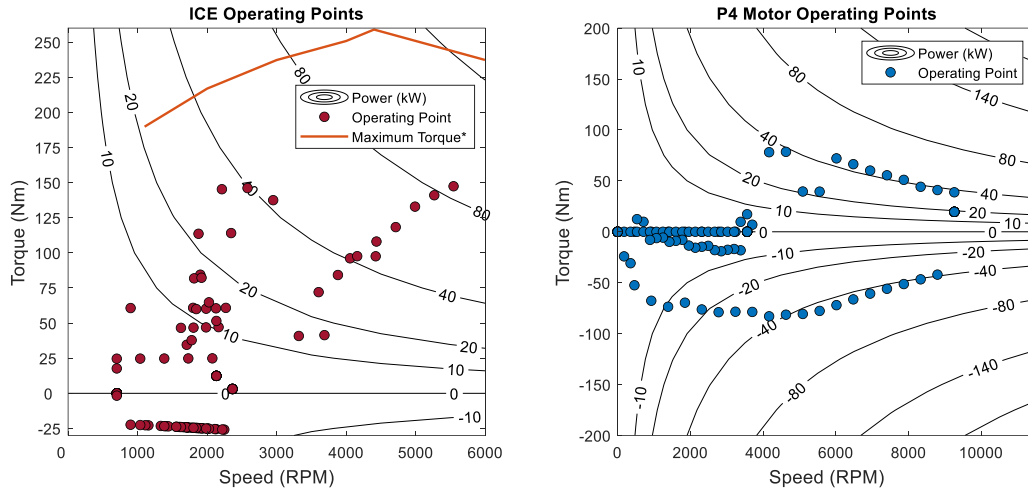


Figure 37: Powertrain Operating Points for Motor and Braking Cost Function Verification Test

The ICE is allowed to operate at low load at the beginning of the first acceleration event and throughout the entire second event. Regeneration torque however is appropriately used throughout the drive cycle. The summary of the energy analysis for this test is shown in Table 6.

Table 6: Motor and Brake Penalty Cost Function Verification Test Energy Summary

Parameter	Value
Fuel Economy (MPG)	23.42
Fuel Used (g)	95.08
Total Fuel Energy (MJ)	4.08
Total Mechanical Energy (MJ)	2.44
Total Electrical Energy (MJ)	0

The energy analysis showed a clear improvement in fuel economy when compared to the quadratic SOC penalty function with a 17.5% decrease in fuel consumption. Another benefit of the motor and brake penalty function is that CS operation can be strictly enforced resulting in zero net electrical energy used. This cost function yielded fuel results within 16.6% difference of the CD maximum electric motor use solution.

5.1.3.4 Torque Overproduction Penalty Factor Cost Function

Drawing from the improvements of the motor and brake penalty cost function, a final cost function was developed for use in the DP algorithm. This cost function implemented an overproduction of torque penalty factor to the cost function as shown:

$$g(i, j) = \min(\dot{m}_{fuel} + \alpha(|\tau_{w,ICE}| + |\tau_{w,mot}| + |\tau_{w,brake}|)) \quad (5.5)$$

where α is a tunable weight factor. With this cost function, all production of torque is penalized in the cost function. This effectively converts the goal of the DP algorithm to minimize the fuel consumed with the lowest possible production of torque. The careful calibration of the weight factor is critical to ensuring that the fuel term of the cost function is the driving force behind the decision making of the DP algorithm, while the additional overproduction term differentiates the total path cost. This term eliminates the uniform transition cost effect by making each cost unique and equally penalizing the use of each torque source in the vehicle powertrain. The powertrain power output results for an α value of 0.008 is shown in Figure 38.

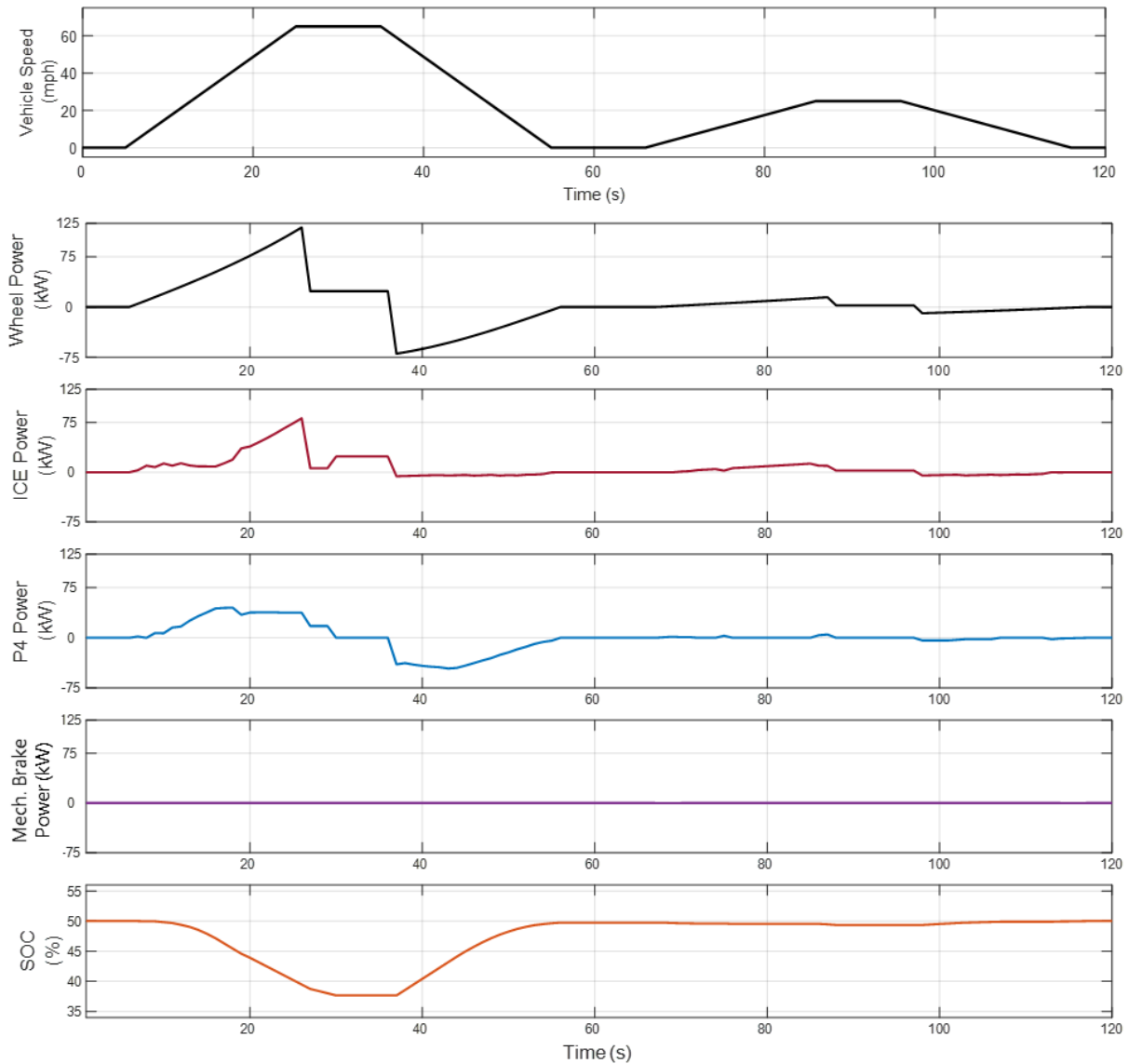


Figure 38: Overproduction Penalty Cost Function Verification Test Powertrain Power Output

The overproduction penalty factor strikes an appropriate balance between ICE and motor torque production, specifically in low speed and low gear regimes. The electric motor maximizes the amount of regenerative braking extracted from the drive cycle. The operating points of the powertrain components for the overproduction penalty cost function is shown in Figure 39.

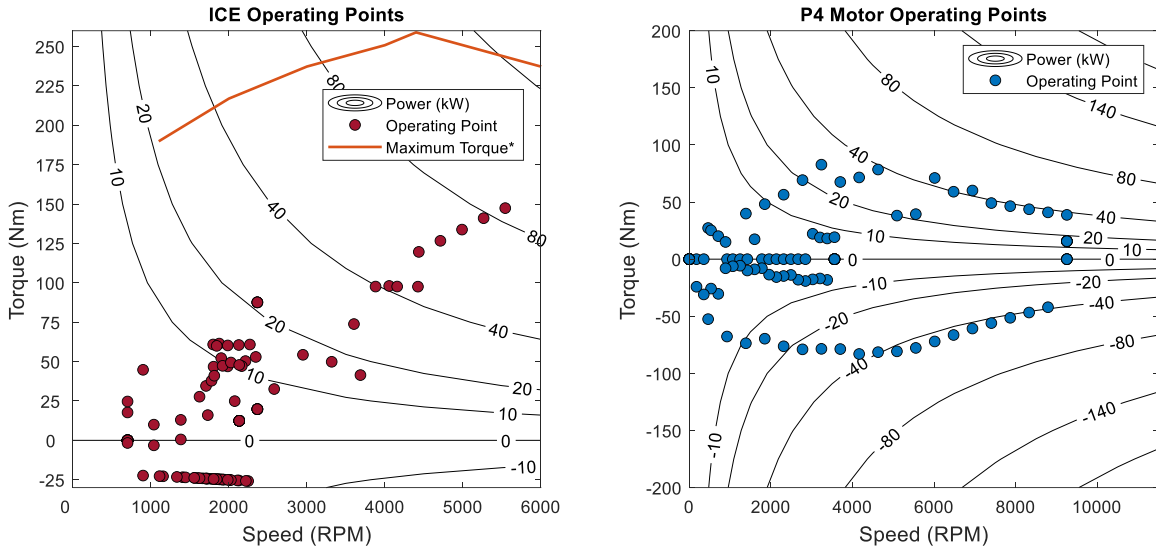


Figure 39: Powertrain Operating Points for Overproduction Cost Function Verification Test

The operating points show improved use of the electric motor as well as the shift of ICE operating points to more efficient regions. The negative effects of leaking and uniform transition cost are avoided and realistic operation of the powertrain is achieved. The energy analysis results for the overproduction penalty cost function are shown in Table 7.

Table 7: Overproduction Penalty Cost Function Verification Test Energy Summary

Parameter	Value
Fuel Economy (MPG)	25.03
Fuel Used (g)	88.96
Total Fuel Energy (MJ)	3.82
Total Mechanical Energy (MJ)	2.44
Total Electrical Energy (MJ)	0

Of the modified cost functions, the overproduction penalty cost function has the highest performance with respect to fuel economy with a 48.8% decrease in fuel consumption when compared to the engine only operation and within 10.9% of the CD operation of the vehicle.

5.1.3.5 Summary of Cost Function Evaluation

The resulting performance of each of the cost functions evaluated in this analysis are shown in Table 8.

Table 8: Cost Function Evaluation Summary

Cost Function	Total Fuel Energy (MJ)	Total Electrical Energy (MJ)	Drawback
Fuel Only	3.41	0	Unrealistic control actions
Quadratic SOC Penalty	4.80	0.06	Reduced range of SOC use
Motor and Braking Penalty	4.08	0	Reduced use of electric motor
Torque Overproduction	3.82	0	Penalized opportunity charging

Based on these results, the overproduction penalty factor was selected to augment the penalty function in the benchmark analysis of the hybrid 2019 Chevrolet Blazer.

The effects of leaking and uniform transition costs present difficult challenges to overcome without additional terms in the cost function. As such, additional terms in the cost function are to be viewed as soft constraints compared to the hard constraints of powertrain limit violation. The main soft constraint in the benchmark analysis conducted in this work is the overproduction of torque. This constraint shifts the goal of the DP algorithm from producing the optimal control policy that minimizes fuel consumption to the optimal control policy that minimizes fuel consumption with the lowest overproduction of torque. The largest implication of this constraint is the penalization of opportunity charging. While analyzing the results for the selection of the cost function for the benchmark analysis, the fuel consumption only cost function would command opportunity charging for massive magnitude torques for short periods which are physically infeasible. Otherwise, the solutions determined were nearly identical to the solution obtained with the torque

overproduction penalty. Because of this, it is concluded that it is not an efficient decision to use opportunity charging in this powertrain and that the use of the overproduction penalty is admissible for the global optimal solution of the fuel consumption minimization problem.

5.2 DP Solution Simulink Validation

It is necessary to ensure that the highly simplified vehicle model developed in the MATLAB environment adequately represents the general performance of the high-fidelity vehicle model from which it was developed. In this work, the torque commands that correspond to the optimal control policy determined by the DP algorithm are used in an open loop feed-forward powertrain model in Simulink. This model was developed directly from the full fidelity vehicle model used by the WVU PCM team by eliminating non-powertrain related interactions and rebuilding the supervisory control system to interpolate and issue the torque commands determined by the DP algorithm. The validation Simulink model is of considerably higher fidelity than the DP powertrain models including the full functionality of MathWork's Powertrain Blockset models for the ICE, Battery, Electric Motor, Torque Converter, Automatic Transmission, Front and Rear Differentials, Wheels and Brakes, Shaft Compliance, and 1DOF Vehicle Dynamics. The results for the powertrain torque production are shown in Figure 40.

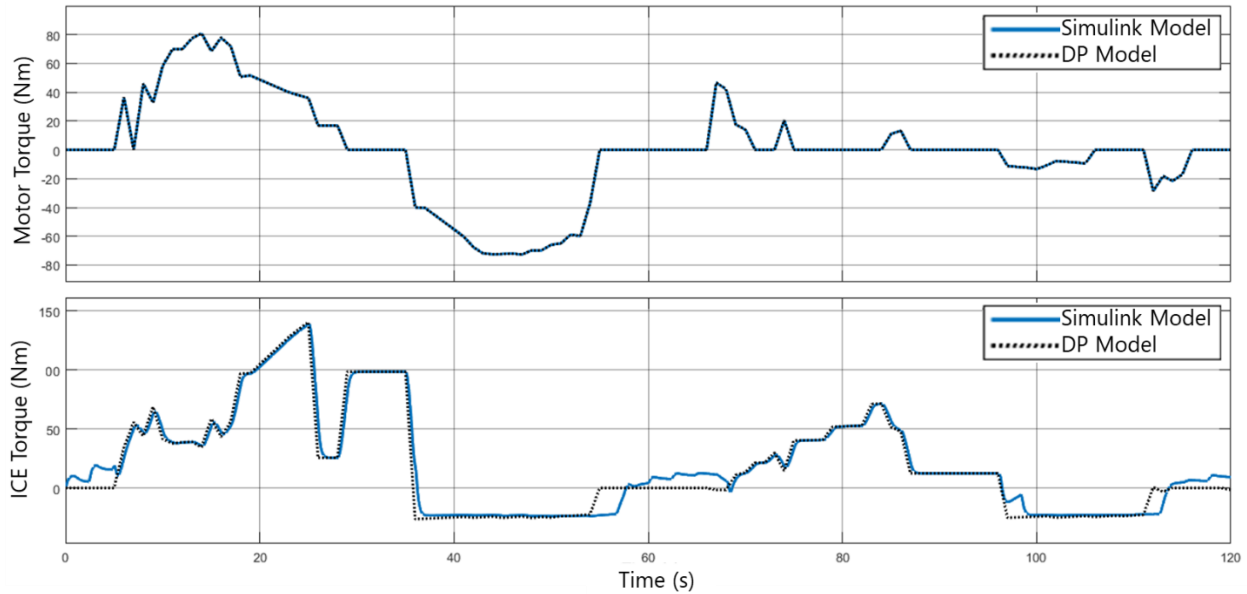


Figure 40: Simulink Validation Model Torque Production Compared to DP Algorithm

The electric motor was able to perfectly follow the commands issued by the hybrid supervisory subsystem, likely due to the simple nature of the electric motor model. The validation model implements the maximum torque curve and the loss data provided by the electric motor manufacturers to model the electric motor. This same data was used to generate the flipped torque lookup table implemented in the DP algorithm. The ICE torque production from the validation model reasonably follows the commands from the DP algorithm with an expected transient lag due to the lumped-parameter dynamics modeling present in the full fidelity ICE model. There are significant differences at idle and after returning from fuel cut off (FCO) conditions in the ICE model. This is likely due to the highly simplified torque converter adopted for the DP algorithm model. In the full fidelity model, there is a longer time constant for the dynamics of the torque converter to lock and unlock to facilitate idle and FCO. Although there are obvious differences, the DP algorithm's powertrain model successfully selects operating conditions that can be reasonably reproduced in a higher fidelity model.

After determining that the higher fidelity powertrain is capable of reproducing the control actions determined by the DP algorithm, the resulting vehicle state and key operating parameters must be validated to show that the DP model can reasonably relate powertrain operation to vehicle dynamics and operation. The resulting battery SOC, vehicle speed, and fuel flow rate comparison from the validation Simulink model are shown in Figure 41.

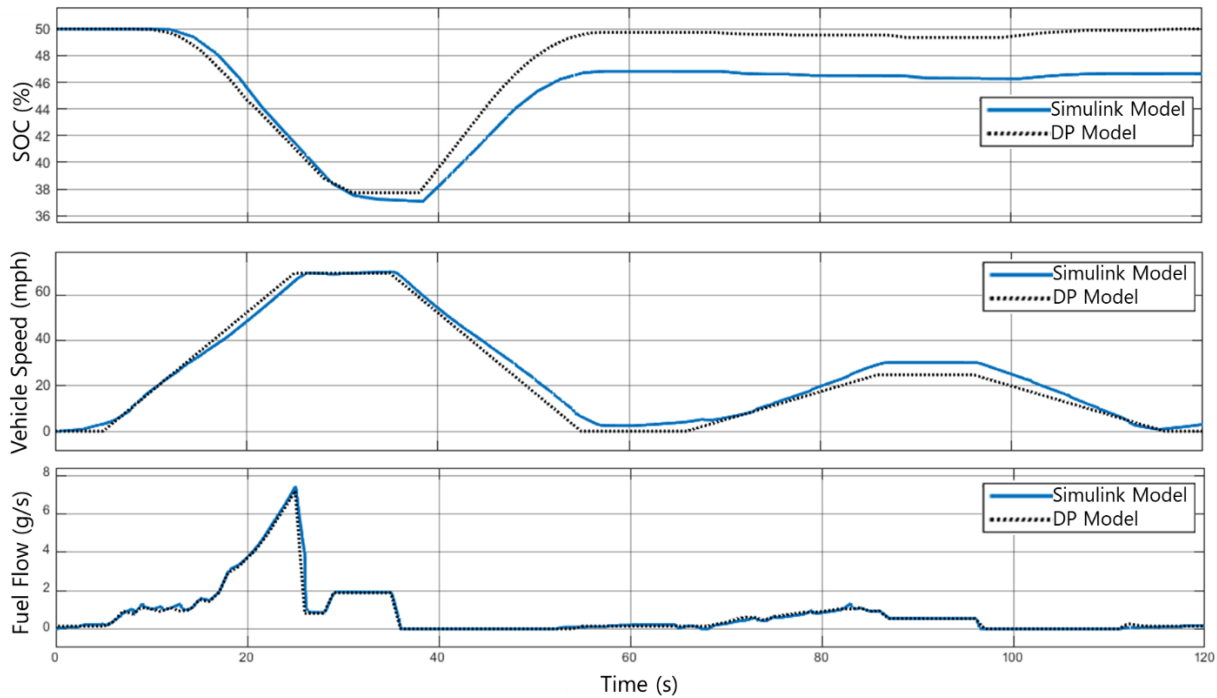


Figure 41: Simulink Validation Model Vehicle State and Dynamics Compared to DP Algorithm

The overall vehicle state and dynamics results are that the validation model reasonably reproduces the modeled dynamics and states used in the DP algorithm. The SOC has the largest deviation from the DP results with the final SOC at 46.8%, likely due to the more robust modeling of the battery pack in the validation model. In the DP algorithm, the change of SOC is directly correlated to the power of the electric motor. In the validation model, the battery pack model implements lookup tables for relevant battery operating parameters such as open circuit voltage, internal resistance, chemical losses, etc. The large differences in how the SOC of the battery is modeled are what drives the large differences between results. In discharge events, the high-fidelity model

produces a similar curve as the DP algorithm, while in regeneration, much less energy is recovered into the battery pack due to the additional modeling of electrical, chemical, and mechanical losses.

The control actions determined by the DP algorithm are shown to accurately propel the vehicle within a reasonable margin of error. The general trend of the initial acceleration event is followed well by the validation simulation with a total speed error of 3 mph. Following the initial acceleration, cruise and deceleration, the vehicle does not come fully to rest in the validation model likely because of the open loop nature of the validation model. The model explicitly follows the braking commands issued by the hybrid supervisory subsystem and with the minimization of overproduced torque, the mechanical brakes and negative torque from the electric motor do not supply sufficient braking to stop the vehicle with the additional vehicle dynamics modeled. During the second acceleration event, the top speed is overshoot by 6 mph, but cruise is adequately maintained. Overall, the speed profile of the validation model adequately follows the DP algorithm within acceptable margins of error.

Finally, the fuel consumption comparison gives insight to how well the DP algorithm models the key cost function parameter. The comparison between the validation model and DP algorithm show that the fuel values are extremely similar with the total fuel consumed by the validation algorithm being 90.05g while the DP algorithm uses 88.96g. These results show that although the DP algorithm ICE model is highly simplified compared to the full fidelity validation model, the quasistatic nature of the two models still govern the overall results of the two models. Overall, the results of the validation testing in Simulink show that the operating conditions determined by the DP algorithm are admissible and adequately capture the general steady state dynamics of the WVU EcoCAR team competition vehicle.

5.3 Benchmark Analysis

With the DP algorithm verified to produce an optimal set of control actions that have been validated to appropriately model the performance of the hybrid 2019 Chevrolet Blazer, the benchmark analysis can be conducted to extract the optimal control policies over relevant drive cycles such that the results can be implemented into the control strategy adopted by the WVU EcoCAR team. The benchmark analysis will examine 5 drive cycles to determine the optimal control policies and relevant parameters that are beneficial with the design of the control system. The drive cycles are the EMC City EMC Highway, US06, UDDS, and HWFET drive cycles. The EMC drive cycle are the criteria on which the WVU EcoCAR team is evaluated for energy consumption in the EMC. The optimization of fuel efficiency on target cycles is a common practice for automotive manufacturers to maximize the advertised performance of vehicles and meet corporate requirements by set forth by environmental protection agencies. Similarly, if the team vehicle's control system is optimized to the target of the EMC competition, the vehicle will perform better while under evaluation of the competition. The additional drive cycles in the benchmark analysis each offer a specific road condition that makes each one unique. The performance of the powertrain over these cycles can give insight for how an online control should handle specific situations.

5.3.1 EMC City

The EMC City cycle is designed to be representative of city driving conditions with two long driving events with variable speed followed by a multitude of quick acceleration and braking representative to stop and go traffic from a stop light. The speed and time profile with associated

wheel torque and power requirements determined from the vehicle glider model described in Section 4.3.1 are shown in Figure 42.

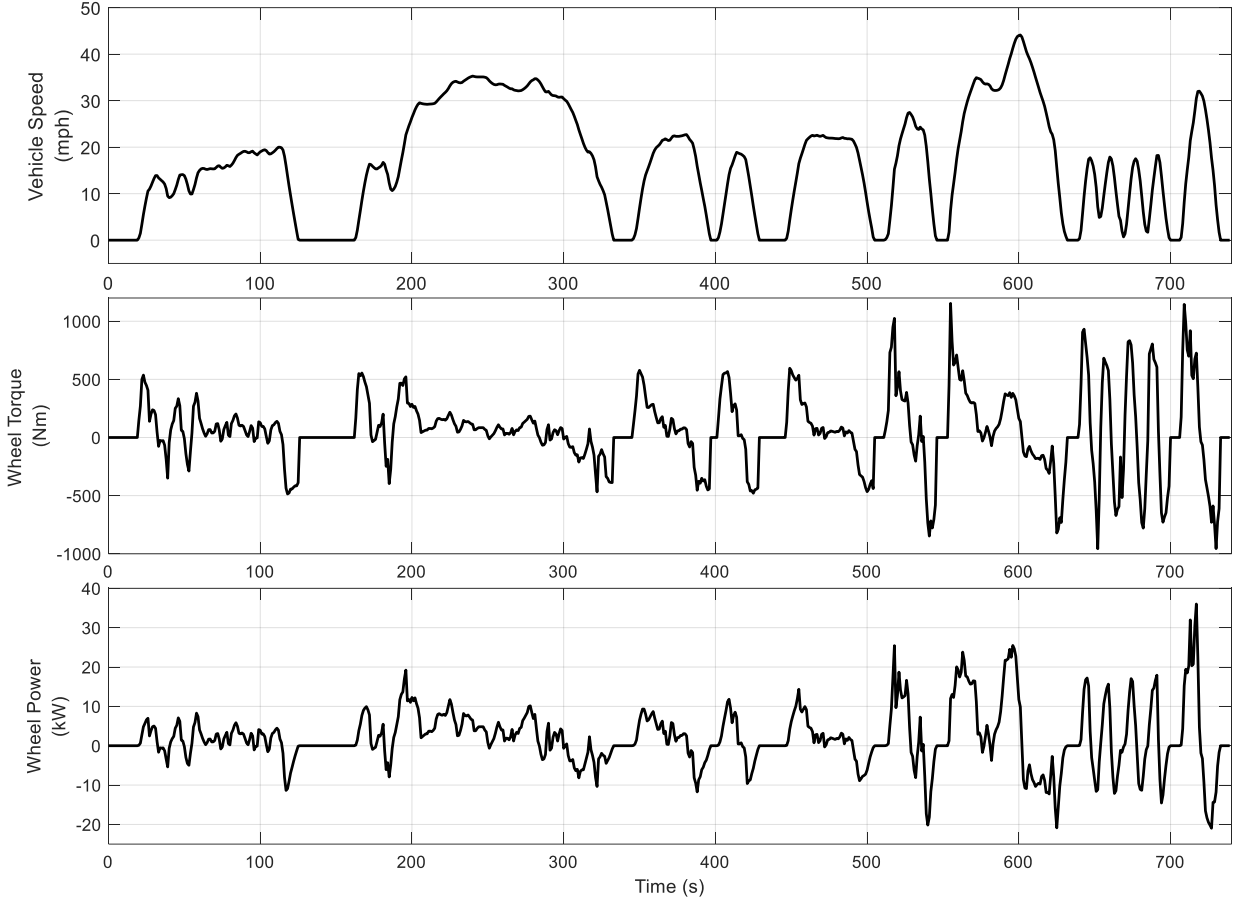


Figure 42: EMC City Drive Cycle Speed, Torque and Power Requirements

The torque and power requirements are relatively low compared to the verification cycle with the largest demands occurring near the end of the cycle. The shift schedule of the cycle is determined according to the strategy discussed in Section 4.3.1 and is shown in Figure 43.

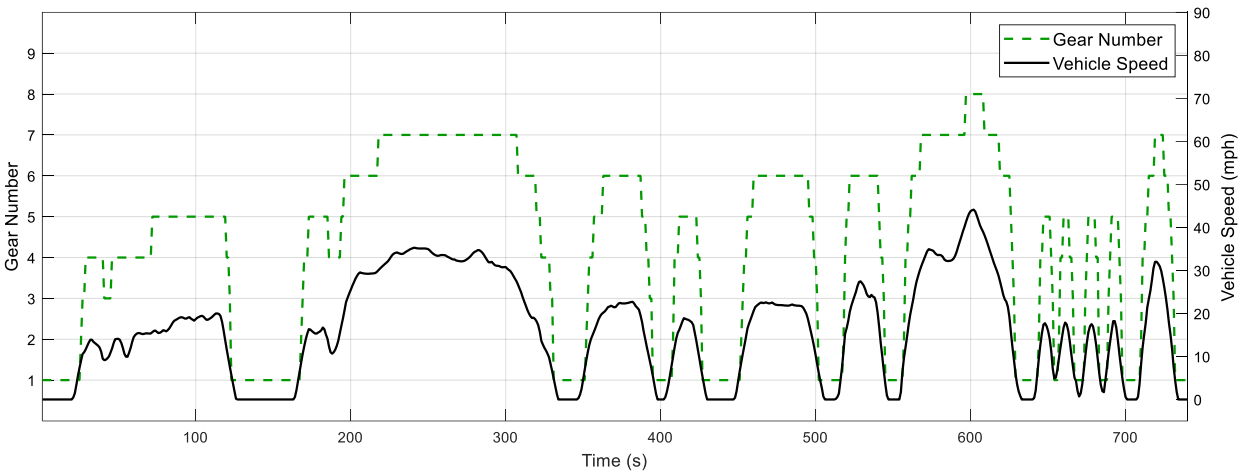


Figure 43: EMC City Shift Schedule

Similar to the torque and power requirements, the shift schedule is relatively relaxed compared to the verification cycle. Relevant information for the EMC City drive cycle and DP algorithm initialization are shown in Table 9.

Table 9: EMC City Information and DP Algorithm Initialization Parameters

DP Algorithm		EMC City	
Parameter	Value	Parameter	Value
Maximum SOC (%)	80	Distance (mi)	3.34
Minimum SOC (%)	20	Total Time (s)	740
Target SOC (%)	50	Sample Time (Hz)	1
Number of Grid Points	5,000	Maximum Acceleration (m/s ²)	2.09
Δ SOC (%)	0.012	Maximum Deceleration (m/s ²)	-1.86
ΔP_{batt} (W)	648	Mechanical Energy Required (MJ)	3.87

Discretizing 5,000 points within the state grid strikes a balance between runtime and memory requirements. The constraints of the ESS state that the SOC should remain between 20% and 80% and although the optimal control policy likely exists within a far more narrow window of SOC,

the entire feasible range of the ESS is included. The powertrain power output results are shown in Figure 44.

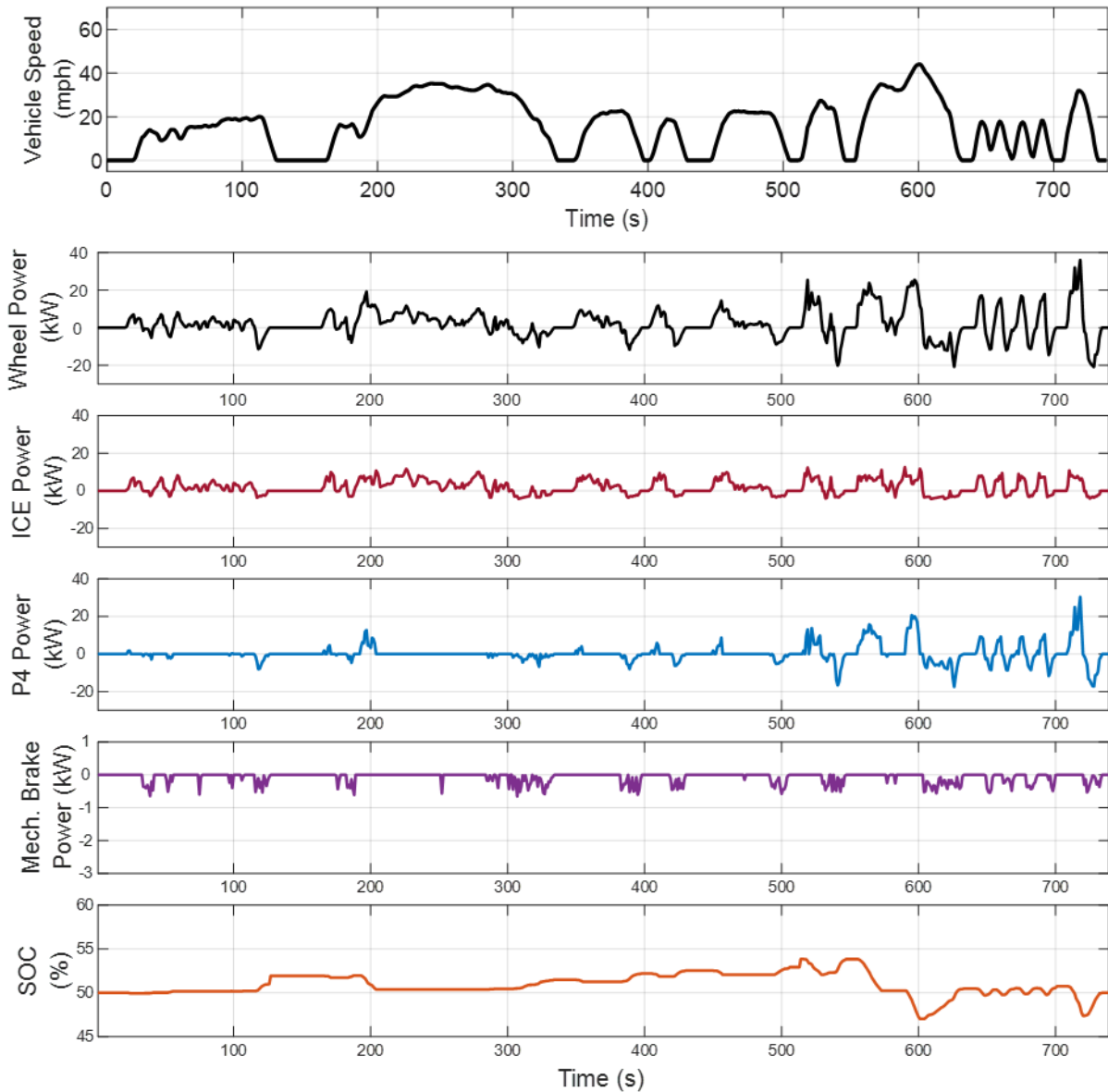


Figure 44: EMC City Powertrain Power Output

Early in the EMC City drive cycle, the ICE is used for the majority of propulsive torque while the electric motor is typically used for low power requirements. The braking torque is primarily supplied by the electric motor with the mechanical brakes used only to supplement the braking effort. Later in the cycle, the electric motor is used much more heavily for both positive and

negative torque as the vehicle accelerates and brakes more frequently. The largest deviation from the target SOC is to 47.02% during the stop and go portion of the drive cycle.

As the fuel consumption, power loss, and efficiency maps of both the ICE and P4 motor used in the benchmark analysis are protected by confidentiality agreements, the operating points of these components are shown overlaid with constant power lines and the maximum torque line as a frame of reference. It should be noted that the included line is a publicly available reported maximum torque determined from a 2016 Chevrolet Colorado [47] and differs from the maximum torque used to model the powertrain as described in Section 3.3.2.

It can be stated that the ICE generally uses more fuel as speed and torque increase. However, the most efficient use of fuel energy by the ICE occurs roughly in the region of 70% of maximum torque between 2000 and 4000 RPM reaching a maximum operating efficiency less than 40% with most maximum efficiency points accumulating closer to 35% efficiency. This efficiency diminishes in all directions outside of this envelope.

The selected operating points for the powertrain components are shown in Figure 45.

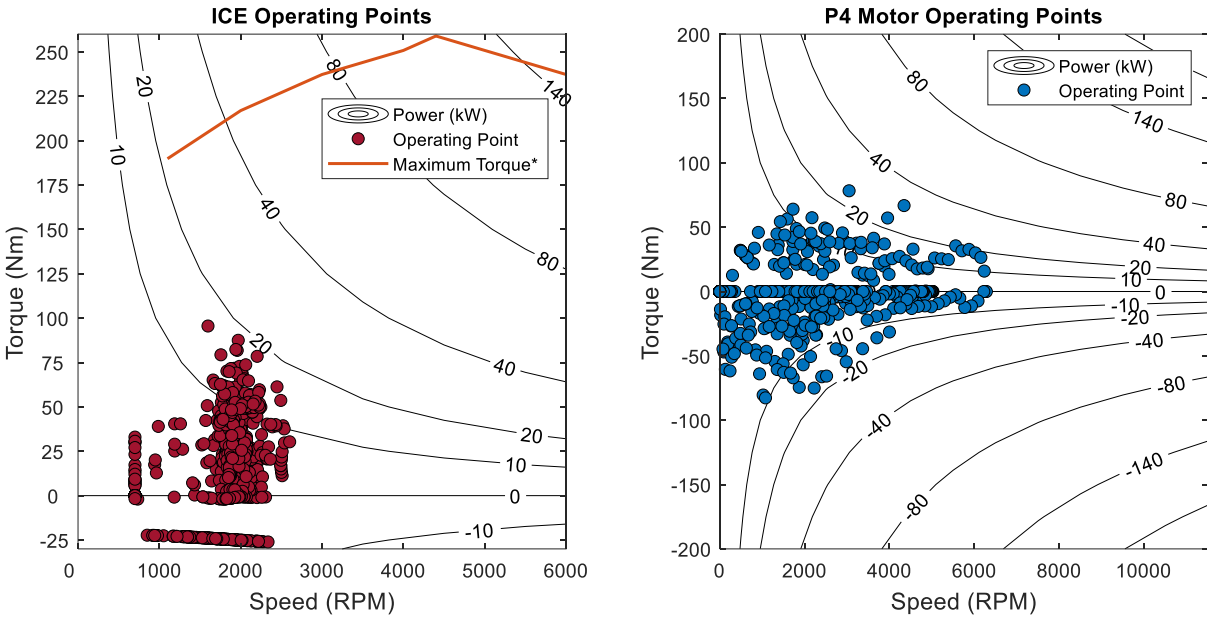


Figure 45: Operating Points of ICE and P4 Motor for EMC City Drive Cycle

The ICE can be identified operating at its minimum admissible running points such as minimum running power near the zero constant power line, and FCO in the negative region of torque. The ICE is shown to frequently operate around 2000 RPM with the majority of torque commands between 0 and 75 Nm of component torque produced. The ICE operates outside of this region a few times with speeds reaching up to 2500RPM and torques of up to 100Nm. The electric motor typically operates between ± 60 Nm over a large range of speeds.

To examine the operating efficiency of the powertrain, a set of histograms were developed to look at the distribution of operating points. The efficiency values are determined as follows:

$$\eta_{ICE} = \frac{P_{ICE}}{P_{fuel}} \quad (5.6)$$

$$\eta_{mot} = \frac{P_{mot}}{P_{loss}} \quad (5.7)$$

$$\eta_{total} = \frac{P_{wheel}}{P_{fuel}} \quad (5.8)$$

where η_{ICE} , η_{mot} and η_{total} are the efficiency of the ICE, electric motor, and total system, respectively. The ICE efficiency determined from ICE only operation of the vehicle over the EMC City drive cycle is used to demonstrate the efficiency improvements from implementing the electric drivetrain in urban conditions. The distribution of ICE efficiencies for ICE only operation over the EMC City drive cycle is shown in Figure 46.

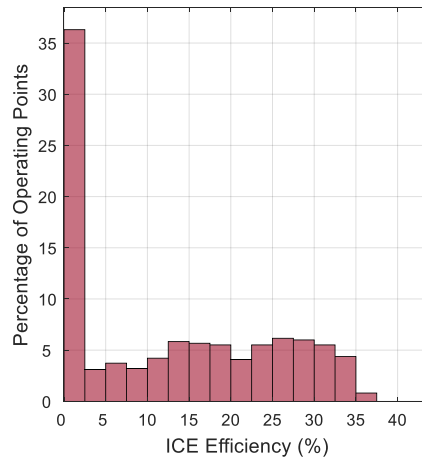


Figure 46: Efficiency Distribution of ICE Only Operation for EMC City Drive Cycle

The distribution of operating efficiencies for hybrid vehicle operation are shown in Figure 47.

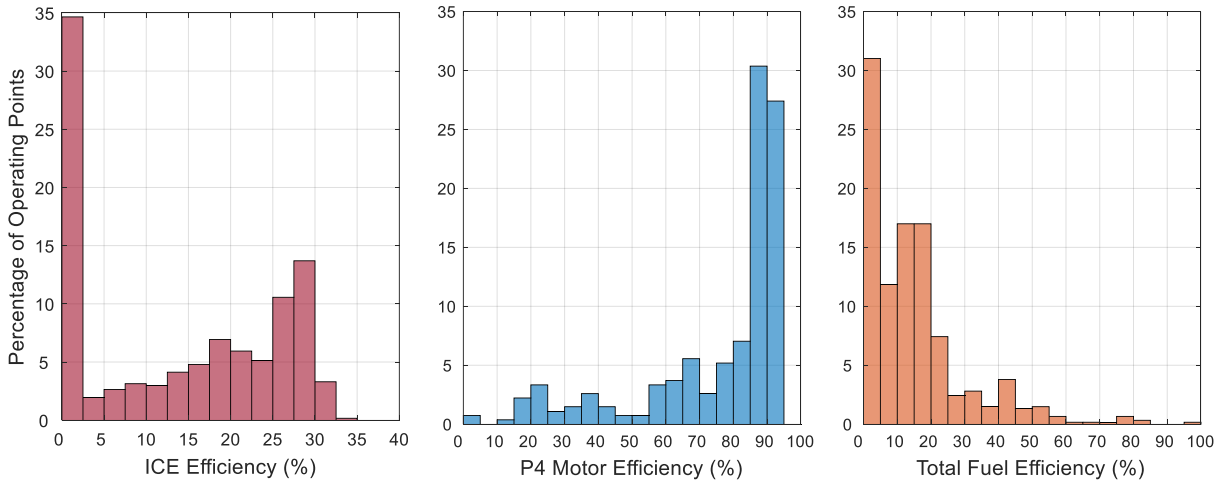


Figure 47: EMC City Optimal Operating Efficiency Distribution

Comparing the ICE only operation to hybrid operation, there is a clear shift in ICE efficiency from a nearly uniform distribution to a significant skew to the right between 25-30% efficiency. There are noticeable peaks between 25 and 30% with few operating points achieving efficiency greater than 30%. It is critical to note that for 18% of the drive cycle, the engine was in FCO and as such no fuel was burned. These points are not captured in the histogram as there is no associated efficiency for these points. The efficiency distribution for motor operation is significantly skewed to the right with peaks between 85-90% efficiency that cover 57% of the operating points. It is important to note that for 62% of the drive cycle the motor is not used and thus no efficiency value is assigned to these points. Considering the system as a whole, the total fuel efficiency is skewed to the left with the majority of points operating below 20% efficiency. This distribution is expected due to the effects of losses from the driveline and energy conversions.

By examining the torque split decisions of the optimal policy determined by the DP algorithm, the criteria for operating at specific torque split ratios can be assessed. The torque split ratio is a key part of energy management in parallel HEVs and is often communicated in various forms. For consistency, in this work, the torque split ratio is defined as follows:

$$TSR = \frac{\tau_{w,ICE}}{\tau_{Req}} \quad (5.9)$$

where TSR is the torque split ratio between ICE wheel torque and the required torque by the cycle. The torque split ratio defined as such represents the amount of required torque to follow the drive cycle that is produced by the ICE. In the EMC City drive cycle, two distinct driving conditions are captured. The first condition is a low-speed cruise with slight perturbations to the speed. Figure 48 shows the torque production of the hybrid powertrain for the second cruise condition of the EMC City Cycle.

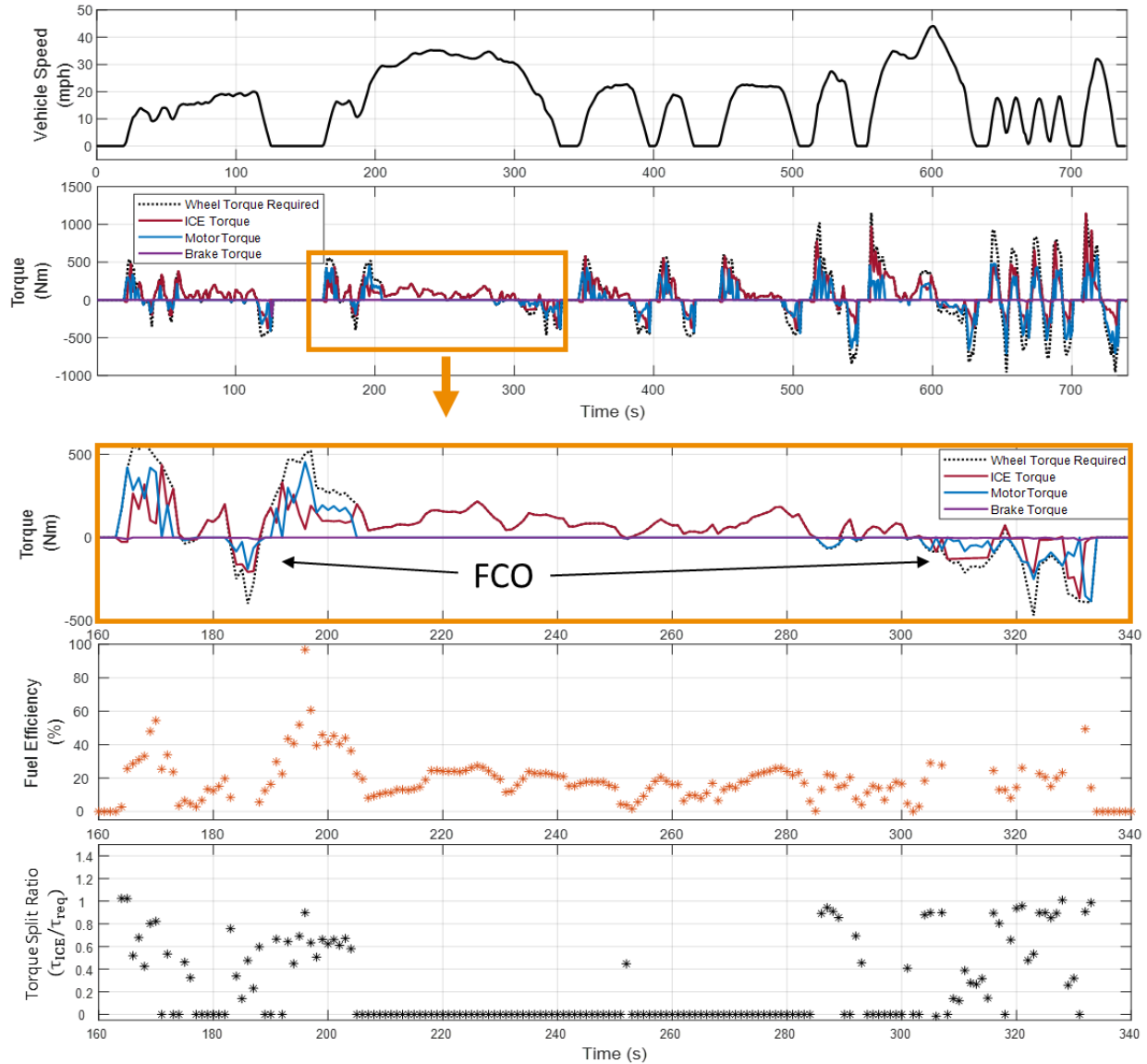


Figure 48: Torque Production Analysis of Cruise Portion of EMC City Drive Cycle

From the torque visualization of the torque split ratio of the cruise portion of the EMC City drive cycle, the electric motor is primarily used to assist the ICE at low speeds and high torque requirements. There is a corresponding significant increase in instantaneous fuel efficiency as the electric motor is used to provide positive propulsive torque. As speed increases, the ICE takes over to produce the entirety of required torque. For negative torque requirements, the ICE goes into FCO, as shown by the callout in Figure 48, while the electric motor supplies the remaining braking torque with regenerative braking.

During the stop and go traffic driving condition portion of the EMC City drive cycle, the torque requirements are much higher with large speed variation. The vehicle is quickly accelerated and decelerated multiple times throughout this portion of the drive cycle. The torque production analysis of the stop and go point of the EMC City drive cycle is shown in Figure 49.

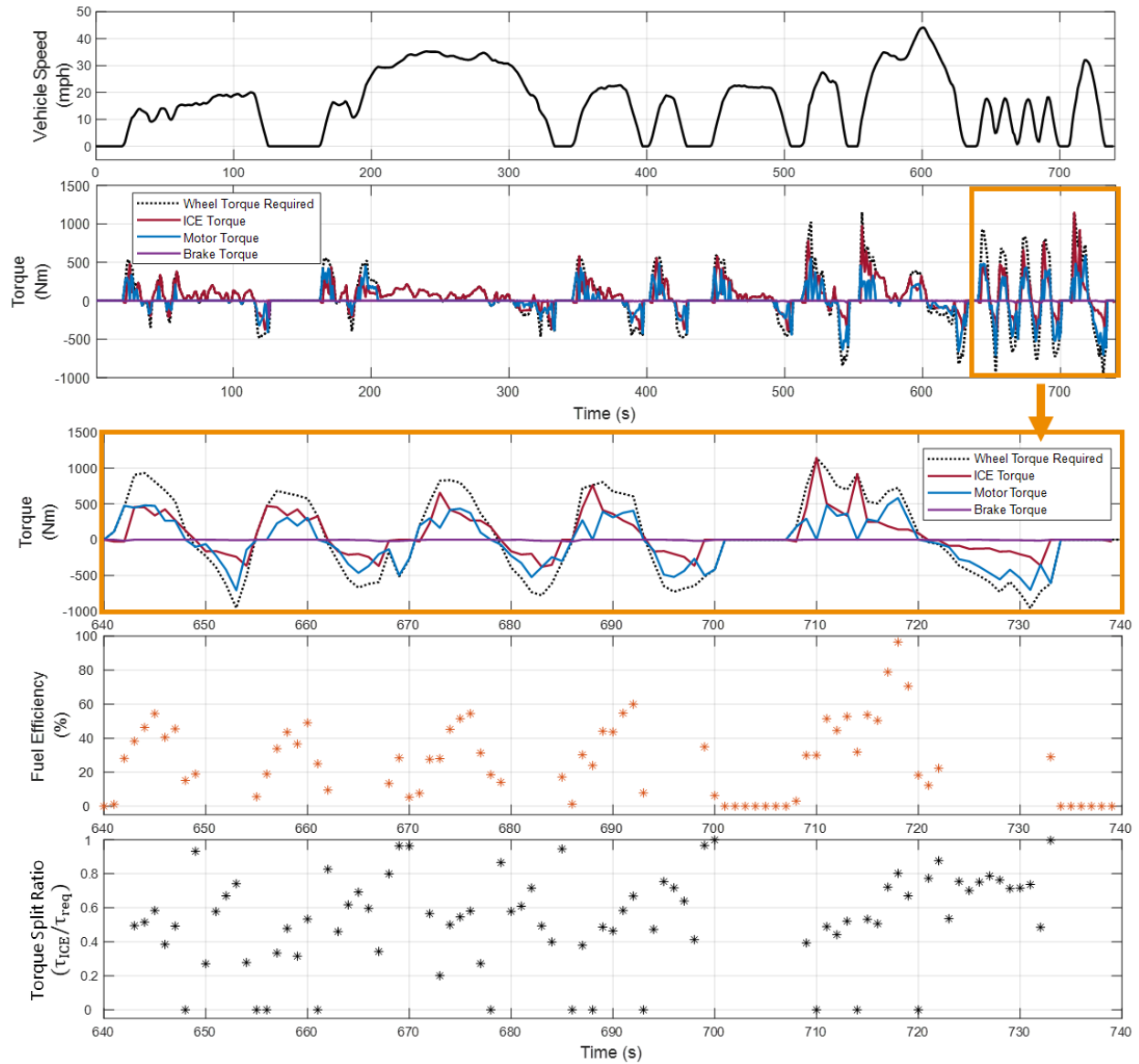


Figure 49: Torque Production Analysis of Stop and Go Portion of EMC City Drive Cycle

Throughout the examined portion of the drive cycle in Figure 49, the electric motor is used significantly to produce both positive and negative torque to the wheels of the vehicle. The general

trend during braking is that the ICE is pushed into FCO and the electric motor is used to capture regenerative braking. Throughout this portion of the drive cycle, there is significant increase in instantaneous fuel efficiency as the electric motor is heavily used.

Investigating the relationship between the power produced by the ICE and the torque split ratio reveals more about the control policy determined by the DP algorithm. The torque split ratio in the positive torque and power regime are shown in Figure 50.

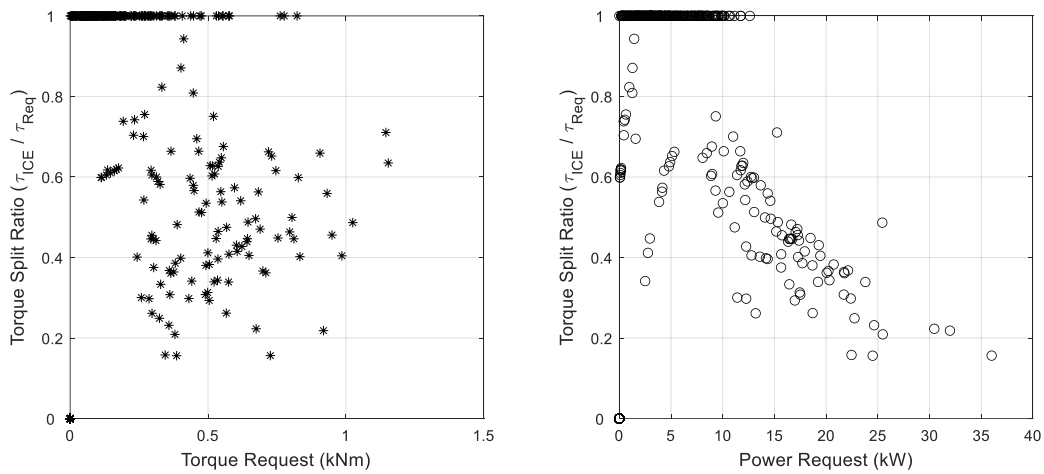


Figure 50: EMC City Torque Split Ratio in Positive Torque and Power Regime

As noted in the results in Figure 49, the TSR is 1 for low torque requests, specifically torque requests under 112 Nm. For higher torque requests there is a loose correlation with smaller values of TSR. In general, the TSR splits between 20-80% of torque between the ICE and electric motor with few operating points outside of that envelope. At very low power requests the ICE generally provides 100% of the torque request, but as the power requirement increases, there is a strong decreasing trend in the torque split ratio. This suggests that at high power requests, the electric motor plays a much larger role in the torque production in the optimal control policy. The TSR in the negative torque and power regime is shown in Figure 51.

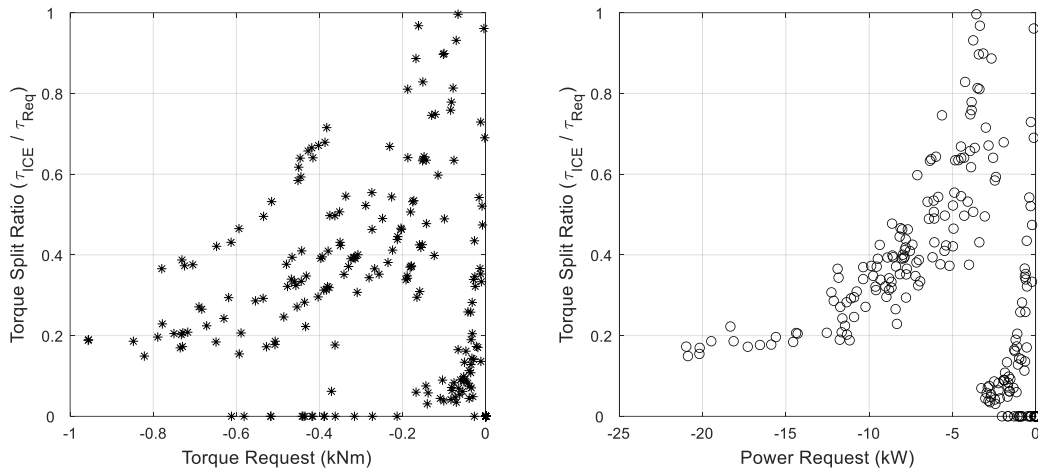


Figure 51: EMC City Torque Split Ratio in Negative Torque and Power Regime

In the negative regime, the TSR is much less useful as it indicates the amount of braking supplied by the ICE while in FCO which is consequential based on the speed of the vehicle, not an amount that is dependent of decisions of the energy management system.

The energy consumption of the vehicle for the EMC City drive cycle is shown in Figure 52.

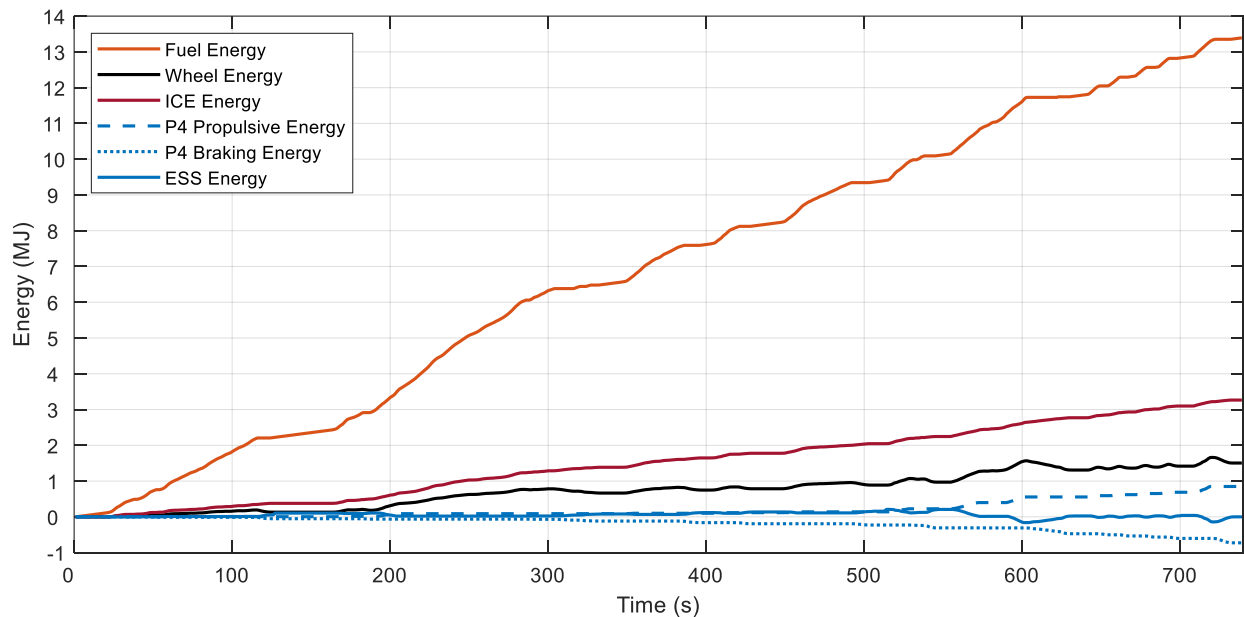


Figure 52: Energy Analysis for EMC City Drive Cycle

A performance summary of the energy used by the optimal control policy as determined by the DP algorithm is shown in Table 10.

Table 10: EMC City Energy Summary

Parameter	Value
Fuel Economy (MPG)	30.18
Fuel Used (g)	311.20
Fuel Energy (MJ)	13.36
Net Wheel Energy (MJ)	3.87
Net Electrical Energy (MJ)	0
Kinetic Energy Recovered (MJ)	0.73

There is significantly more fuel energy consumed throughout the EMC City drive cycle. This result is expected as the ICE does not have engine start/stop functionality thus continuously burning fuel unless in FCO. As expected, the energy produced by the ICE is far lower than the total available fuel energy due to the inherent inefficiencies associated with the ICE. The produced ICE energy exceeds the net energy at the wheels due to the losses associated with drivetrain component efficiencies. Comparing hybrid performance with non-hybrid performance, the 0.73 MJ of energy recovered using regenerative braking throughout the drive cycle that is, in turn, used for propulsive torque saved a total of 3.0 MJ of fuel energy. The resulting fuel economy of the vehicle following the optimal control policy is 30.11 MPG.

5.3.2 EMC Highway

The EMC Highway drive cycle is designed to be representative of highway driving conditions with the beginning portion of the cycle representing the portion of a drive to reach a highway and the remaining repeating portion representative of merging onto the highway, cruising on a

populated road, and exiting the highway multiple times. The speed and time profile with associated wheel torque and power requirements are shown in Figure 53.

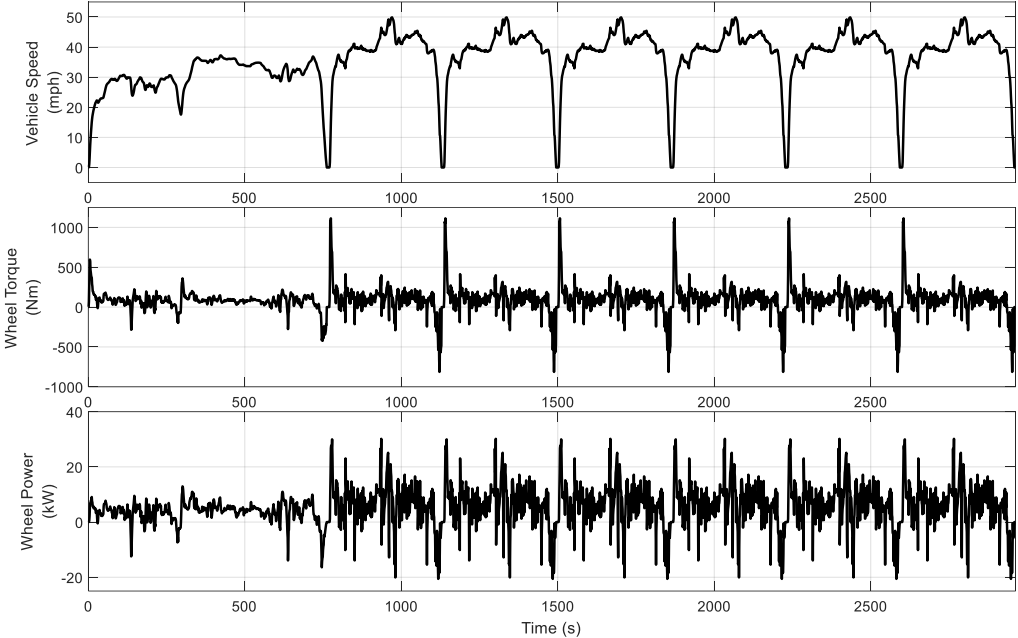


Figure 53: EMC Highway Drive Cycle Speed, Torque and Power Requirements

The torque and power requirements are very different for the two driving conditions. The first condition is much more relaxed while the highway driving portion of the cycle has similar torque and power requirement transients with higher cruise requirements. The shift schedule of the cycle is determined according to the strategy discussed in Section 4.3.1 and is shown in Figure 54.

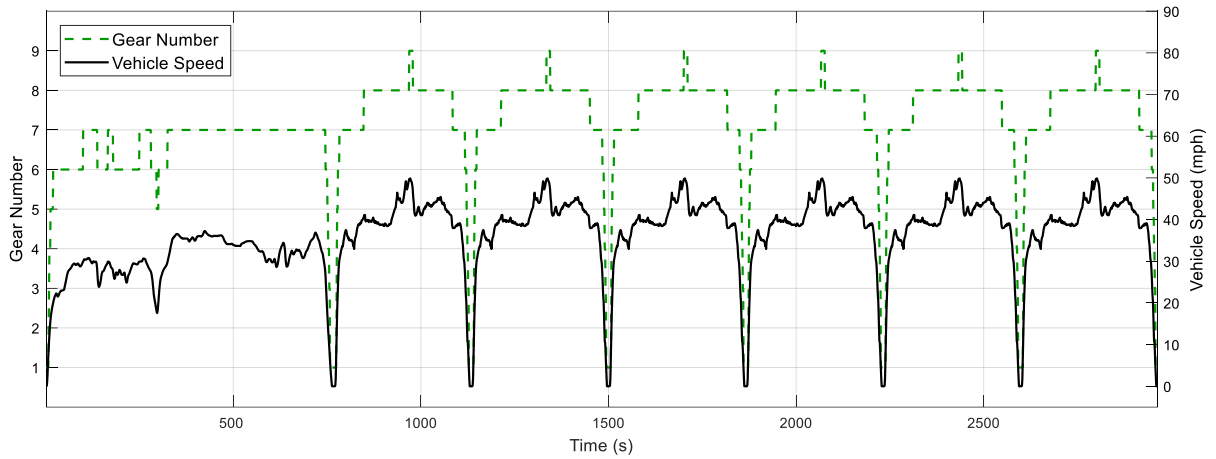


Figure 54: EMC Highway Shift Schedule

Due to the length of the EMC highway cycle, the transmission appears to shift directly into higher gears; however, at low speeds the transmission remains in lower gears. During the acceleration events, the transmission cycles through lower gears quickly as speed increases. Relevant information for the EMC Highway drive cycle and DP algorithm initialization are shown in Table 11.

Table 11: EMC Highway and DP Algorithm Initialization Parameters

DP Algorithm		EMC Highway	
Parameter	Value	Parameter	Value
Maximum SOC (%)	80	Distance (mi)	29.63
Minimum SOC (%)	20	Total Time (s)	2962
Target SOC (%)	50	Sample Time (Hz)	1
Number of Grid Points	5,000	Maximum Acceleration (m/s ²)	1.99
Δ SOC (%)	0.012	Maximum Deceleration (m/s ²)	-1.60
ΔP_{batt} (W)	648	Mechanical Energy Required (MJ)	21.90

As shown, the number of grid points and SOC limits are identical to the analysis of the EMC City benchmark analysis. The powertrain power output results for the EMC Highway drive cycle are shown in Figure 55.

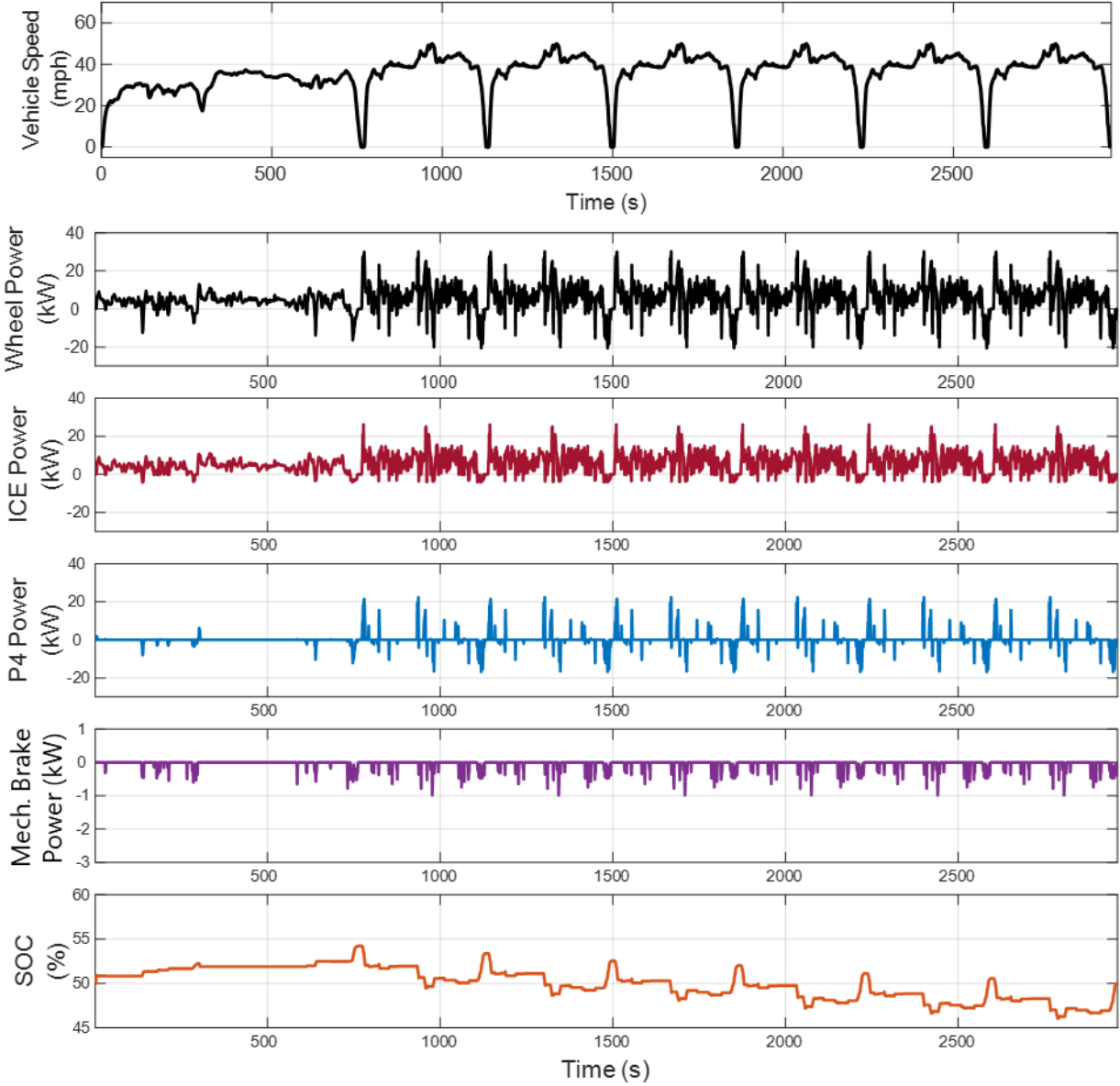


Figure 55: EMC Highway Powertrain Power Output

Similar to the EMC City Drive cycle, the ICE is used for the majority of propulsive torque while the vehicle experiences small perturbations in speed while the electric motor is primarily used

when the vehicle is significantly accelerating. The mechanical brakes are not significantly used throughout the cycle as the electric motor is able to provide all of the required braking torque. The SOC dips slightly early in the cycle but is recovered before transitioning to the second part of the drive cycle. The peaks in SOC reach 54.2% due to the significant braking at those location. The rise in SOC is short lived as the motor is quickly discharged to assist with the acceleration event. Near the end of the cycle, the largest deviation from the target SOC occurs at 46.1%.

To identify the operating conditions of the ICE and electric motor, the torque and speed values of the optimal control policy are shown in Figure 56. It should be noted that the static maps used for fuel consumption and operating efficiency are protected by confidentiality agreements, thus only the maximum torque line is included. It should be noted that the included line is a publicly available reported maximum torque determined from a 2016 Chevrolet Colorado and differs from the maximum torque used to model the powertrain as described in Section 3.3.2.

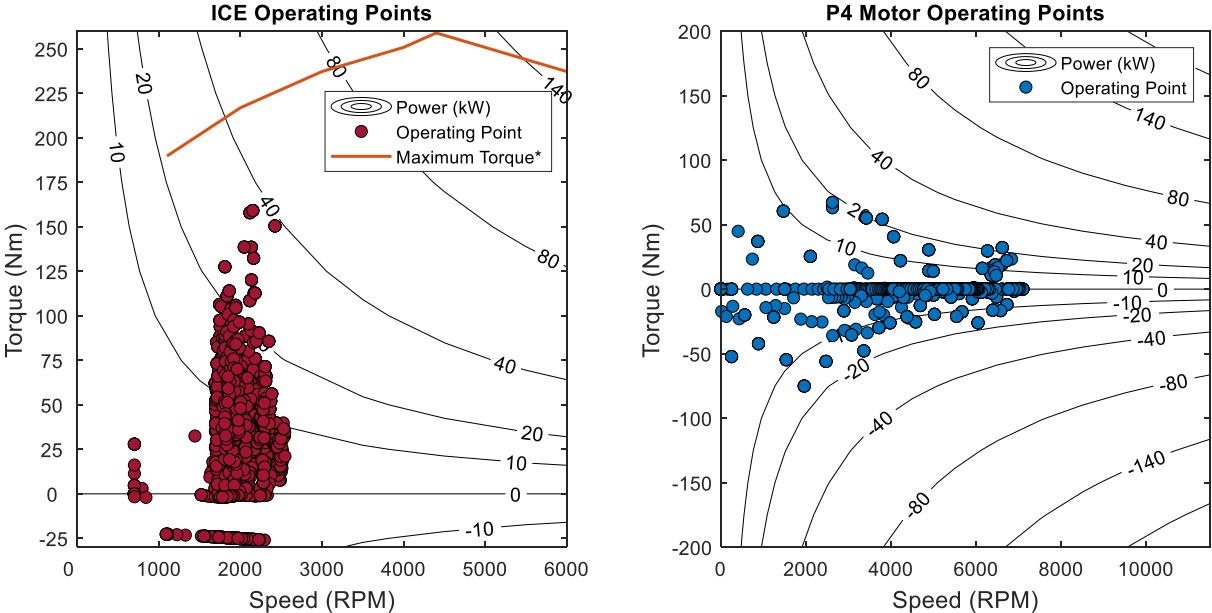


Figure 56: Operating Points of the ICE and P4 Motor for the EMC Highway Drive Cycle

The ICE is shown to operate at its minimum operating points at both minimum running power near the zero-power line and at FCO in the negative torque region of the plot. The ICE operates at torques and speeds similar to the EMC City cycle, between 1700 and 2500RPM, but at higher torque values. In contrast to operation for the EMC City cycle, there is a higher concentration of ICE operational points above 10kW of power. Further contrasting the EMC City cycle, the electric motor is seldom used for positive propulsive torque. In the negative operating region of the electric motor, there is a higher concentration of operating points that range from 0 to -75Nm of torque at various speeds.

The ICE efficiency determined from ICE only operation of the vehicle over the EMC Highway drive cycle is used to demonstrate the efficiency improvements from implementing the electric drivetrain. The distribution of ICE efficiencies for ICE only operation over the EMC Highway drive cycle is shown in Figure 57.

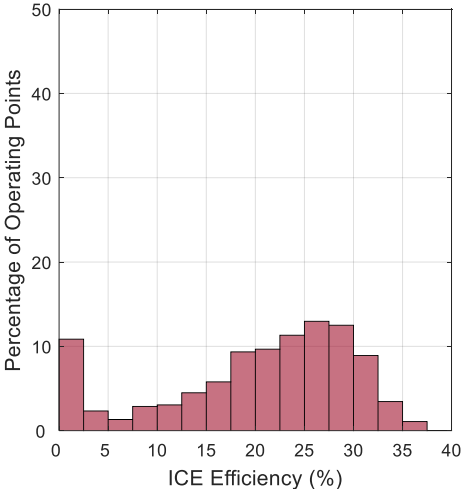


Figure 57: Efficiency Distribution of ICE Only Operation for EMC Highway Drive Cycle

Considering the efficiency of the powertrain operating in hybrid mode, the following set of histograms in Figure 58 investigate the distribution of operating point efficiency throughout the EMC Highway drive cycle.

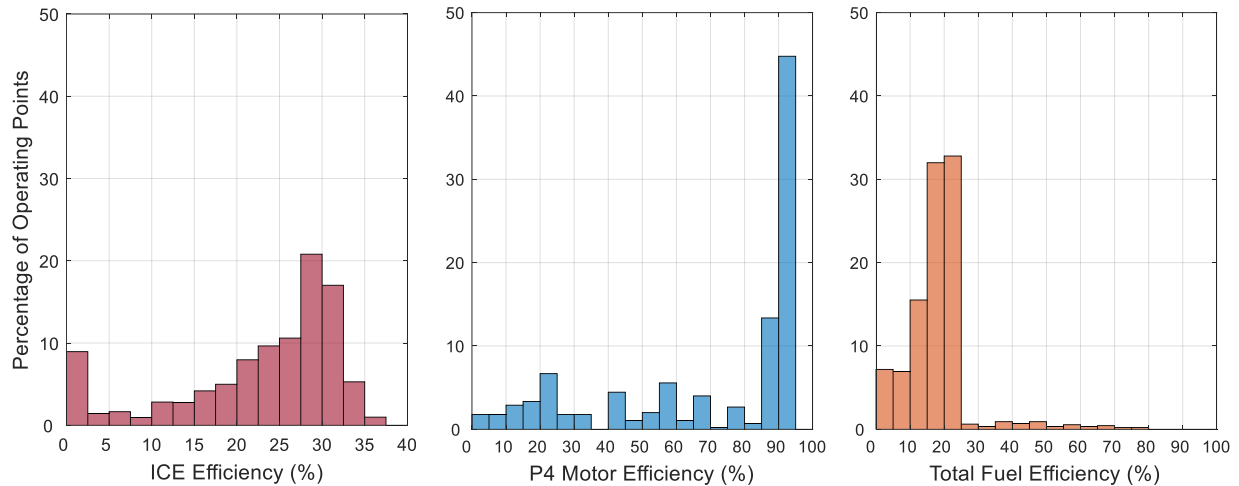


Figure 58: EMC Highway Optimal Operating Efficiency Distribution

For ICE only operation, there is a skew to the right, however the efficiency is clearly improved by operating in hybrid mode with clear peaks between 27.5-32.5% efficiency making up 40% of the operating points of the ICE. It is important to note that the ICE operated in FCO for only 8% of the drive cycle which is not captured by the histogram. The electric motor efficiency is largely distributed above 85% for nearly 54% of operating points in the cycle; however, compared to the electric motor performance in the EMC City drive cycle, there is a much higher concentration of low operating efficiency in the electric motor with several peaks occurring below 50% efficiency. Additionally, the electric motor was not used to provide propulsive torque for 85% of the drive cycle which is not represented in the histogram. Considering the entire systems fuel economy, the overall system fuel efficiency operates between 15 and 25% efficiency for more than 67% of the drive cycle. In contrast with the EMC City drive cycle results, there is not significant operation of the system above this range of efficiencies.

Understanding that in the EMC Highway drive cycle the electric motor is seldom used, identifying where the electric motor provides propulsive torque is critical to understand where the gains in

efficiency occur. Figure 59 shows the production of torque by the hybrid powertrain for the beginning segment of the EMC Highway drive cycle.

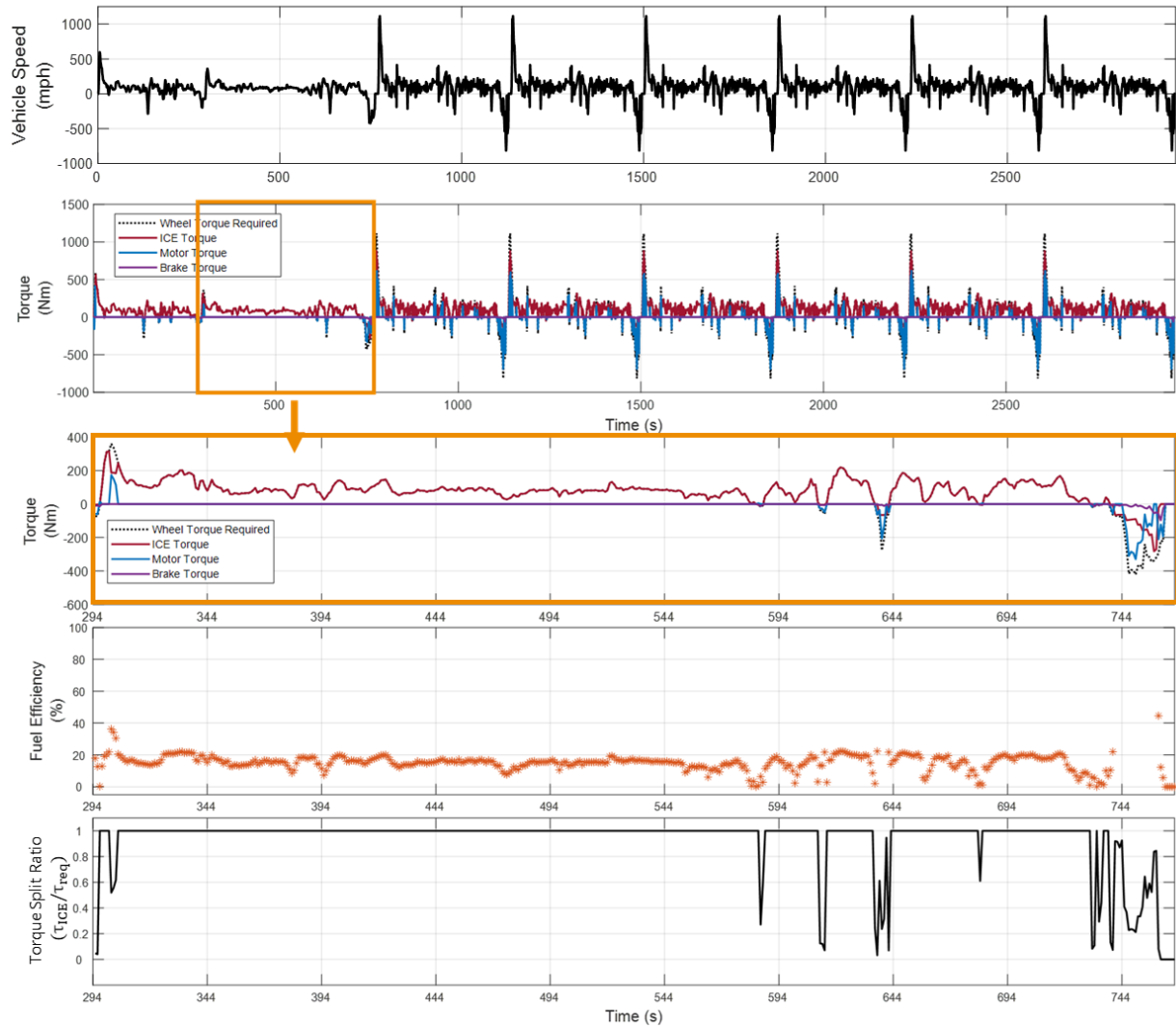


Figure 59: Torque Production Analysis of Cruise Portion of EMC Highway Drive Cycle

For sharp accelerations the electric motor is used for positive propulsive torque, however, during the majority of the cruise portion of the drive cycle, the ICE generates all of the torque to the wheels. There are several occasions where negative torque is required to follow the drive cycle, and the system responds by pushing the ICE into FCO and the electric motor to decelerate the vehicle where it is able to capture a portion of the energy. As the drive cycle transitions into the

higher demand highway driving segment, there remains to be little use of the electric motor as shown in Figure 60.



Figure 60: Torque Production of Highway Driving Portion of the EMC Highway Cycle

As the vehicle accelerates near the beginning of the highway driving segment of the drive cycle, the electric motor is used significantly to provide positive propulsive torque for the vehicle. The typical torque split during acceleration events falls between 0.4 and 0.8, meaning the ICE provides between 40 and 80% of the torque. Throughout the highway segment braking torque is required in short intervals. Similar to the previous driving segment, the system pushes the ICE into FCO while

the electric motor provides a portion of the braking. In both driving segments, there are significant peaks in fuel efficiency as the electric motor is used to provide positive propulsive torque.

The torque split ratio of the positive torque and power operating regime is shown in Figure 61.

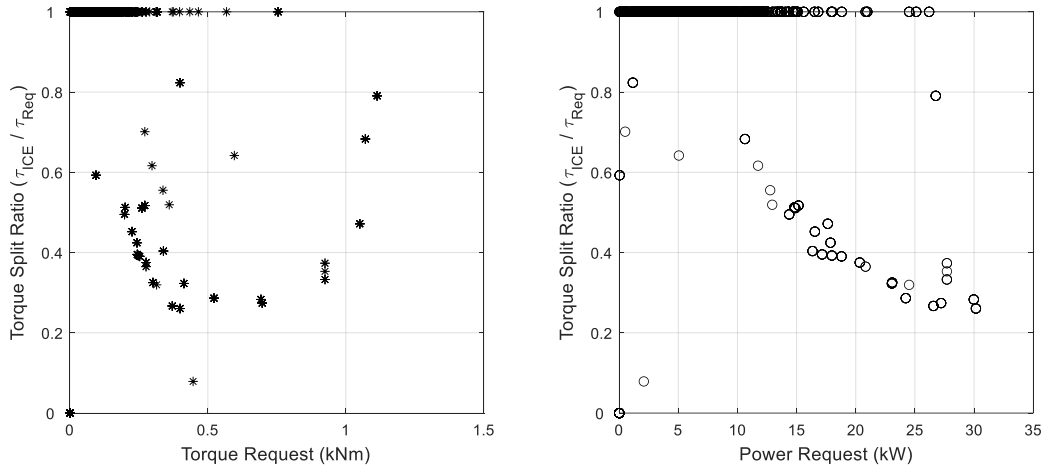


Figure 61: EMC Highway Torque Split Ratio in Positive Torque and Power Regime

The ICE operates on its own for a large majority of the cycle predominately at torque requests below 200Nm with a single outlier at 97Nm. For higher torque requests the TSR is generally between 0.2 and 0.6 with few operating points extending above 0.8 or below 0.2. There is little operation of the electric motor at power requests below 10kW. Similar to the trend from the EMC City drive cycle, there is a strong decreasing relationship for TSR values as the power request increases. The torque split ratio in the negative torque and power operating regime is shown in Figure 62.

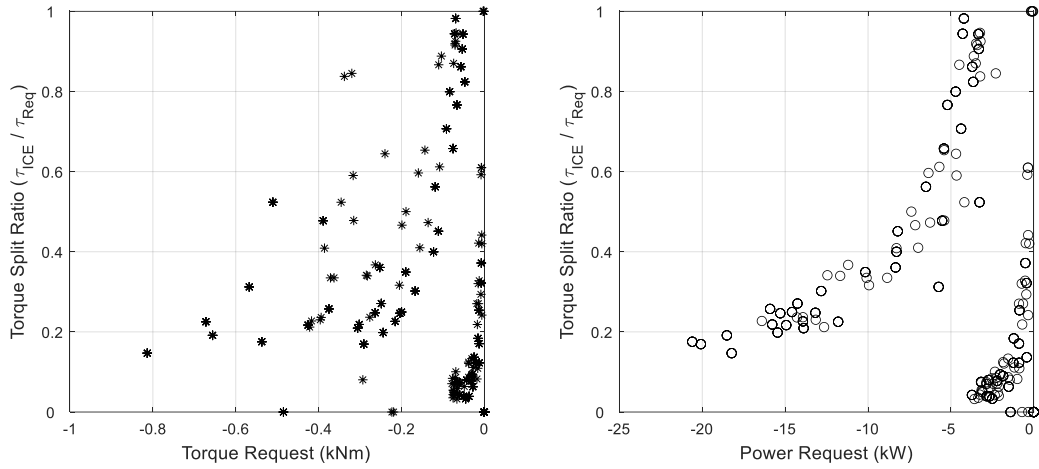


Figure 62: EMC Highway Torque Split Ratio in Negative Torque and Power Regime

In the negative torque and power regime, the results of the TSR are generally consequential. The energy consumption of the vehicle over the EMC Highway drive cycle is shown in Figure 63

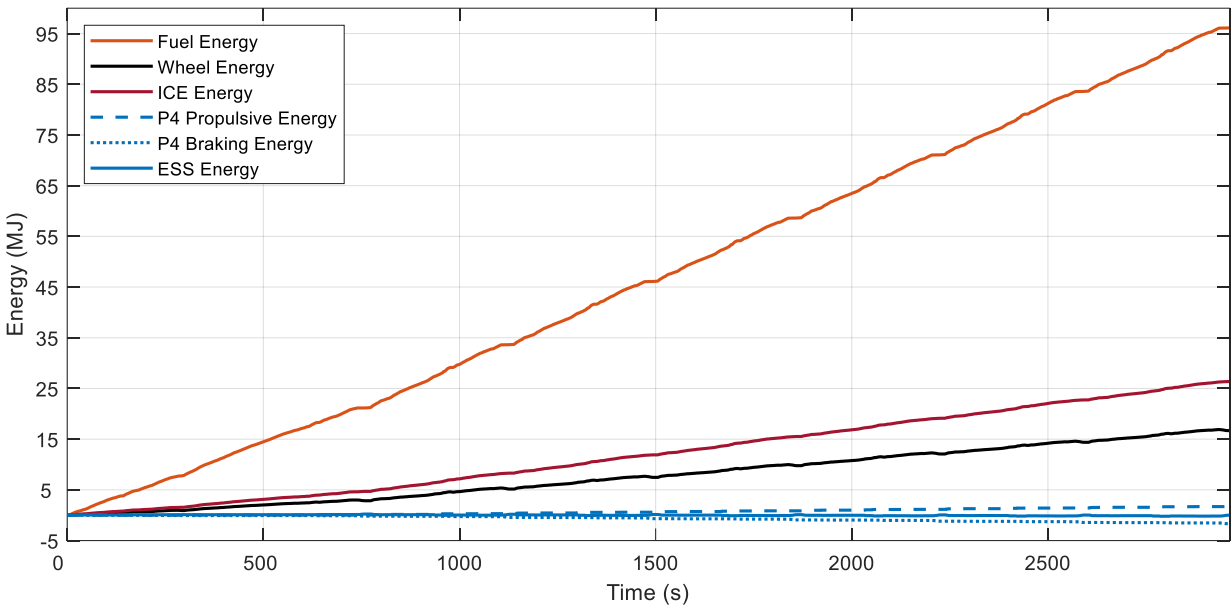


Figure 63: Energy Analysis for EMC Highway Drive Cycle

Table 12 summarizes the relevant overall performance metrics and energy values for the EMC Highway drive cycle.

Table 12: EMC Highway Energy Summary

Parameter	Value
Fuel Economy (MPG)	37.24
Fuel Used (g)	2238.2
Fuel Energy (MJ)	96.11
Net Wheel Energy (MJ)	21.88
Net Electrical Energy (MJ)	0
P4 Propulsive Energy (MJ)	1.68 MJ

Throughout the EMC Highway drive cycle, there is significant fuel consumption. There is a significant amount of kinetic energy recovered throughout the drive cycle although the electric motor does not have many operating points. During the times that the electric motor operated, the ICE consumed less fuel leading to a difference of 11.8 MJ of fuel energy savings from hybrid operation compared to ICE only operation. The resulting fuel economy of the vehicle following the optimal control policy is 37.24 mpg.

5.3.3 US06

To provide a comprehensive benchmarking analysis, the DP algorithm is used to determine optimal control policies for other popular EPA drive cycles. The US06 cycle is an aggressive drive cycle that captures both urban and highway driving conditions [48]. The speed and time profile of the US06 drive cycle with associated wheel torque and power requirements are shown in Figure 64.

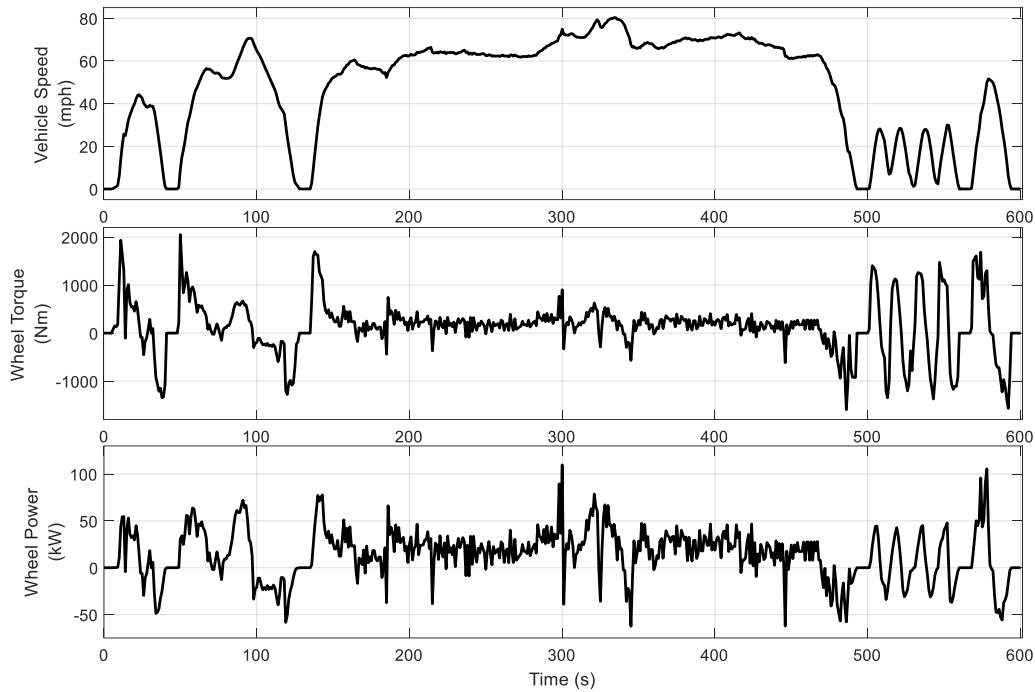


Figure 64: US06 Drive Cycle Speed, Torque and Power Requirements

The torque and power requirements for the US06 drive cycle are considerably more demanding than the previously examined drive cycles. The associated shift schedule for the US06 drive cycle is shown in Figure 65.

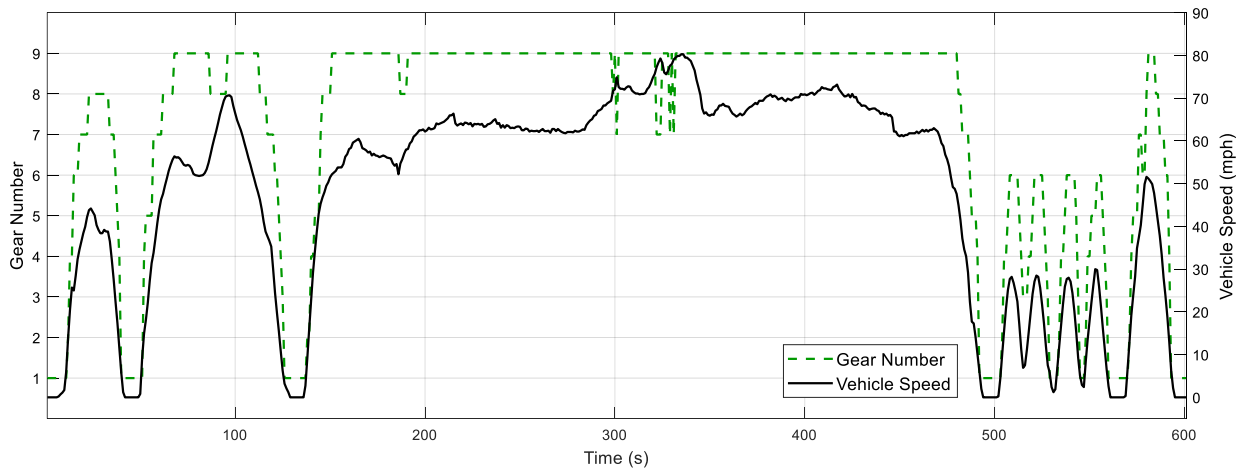


Figure 65: US06 Shift Schedule

Additional relevant parameters for the US06 drive cycle and DP algorithm initialization parameters are shown in Table 13.

Table 13: US06 Cycle Information and DP Algorithm Initialization Parameters

DP Algorithm		US06	
Parameter	Value	Parameter	Value
Maximum SOC (%)	80	Distance (mi)	8.01
Minimum SOC (%)	20	Total Time (s)	600
Target SOC (%)	50	Sample Time (Hz)	1
Number of Grid Points	5,000	Maximum Acceleration (m/s ²)	3.76
Δ SOC (%)	0.012	Maximum Deceleration (m/s ²)	-3.08
ΔP_{batt} (W)	648	Mechanical Energy Required (MJ)	14.43

This drive cycle was analyzed due to the sharp accelerations and higher speeds to provoke greater power and torque demands from the vehicle. The powertrain power output results for the US06 drive cycle are shown in Figure 66.

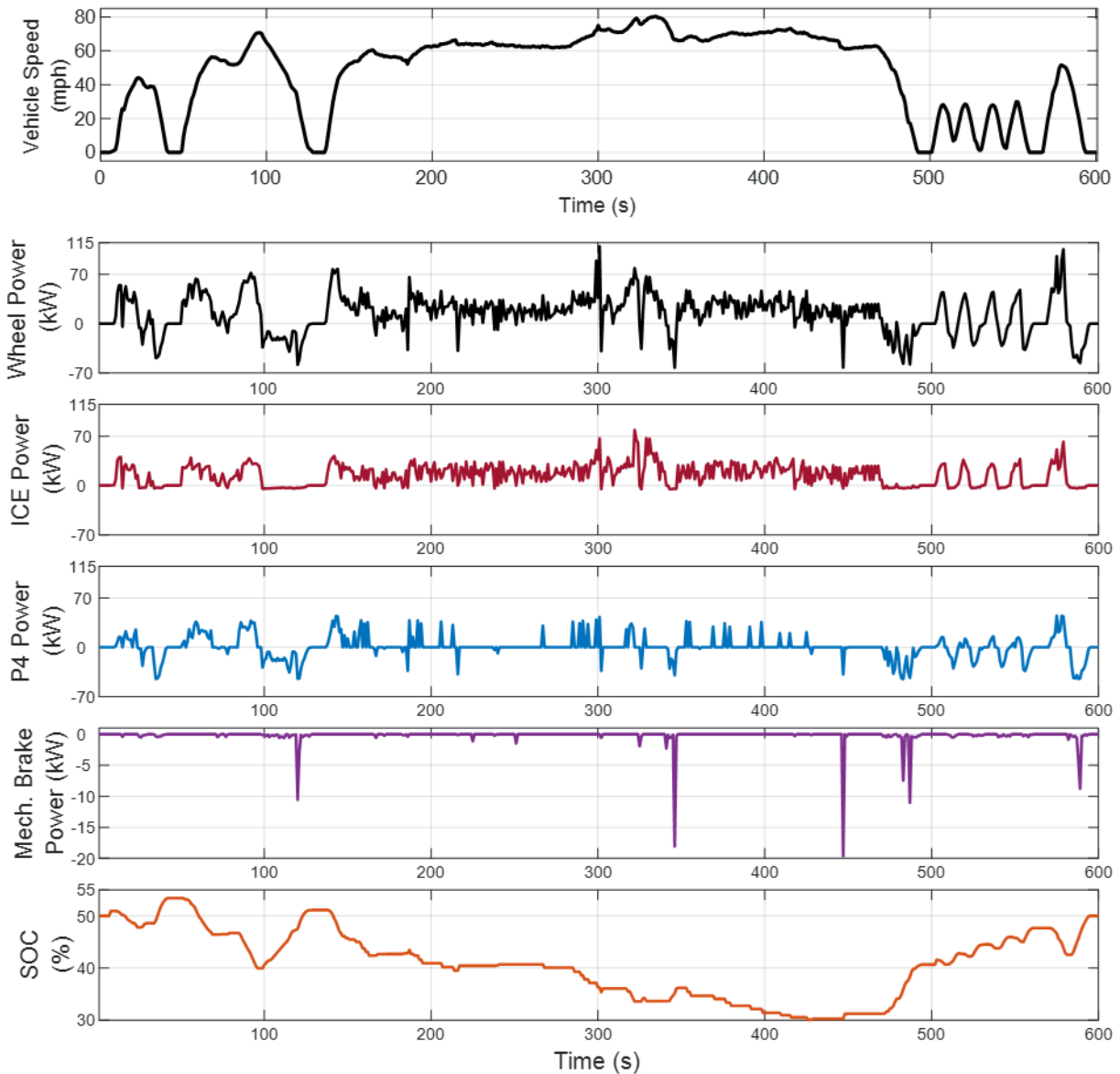


Figure 66: US06 Drive Cycle Powertrain Power Output

The demanding nature of the US06 drive cycle results in significant use of the ICE and electric motor together. Many of the power outputs are near the component maximums to be able to achieve the torque requirements of the drive cycle. The US06 drive cycle yields the largest deviation from the CS target, reaching a maximum of 53.4% and a minimum of 30.5%. The braking maneuvers in this drive cycle also require the use of the mechanical brakes several times.

The operating points of the ICE and electric motor are shown in Figure 67.

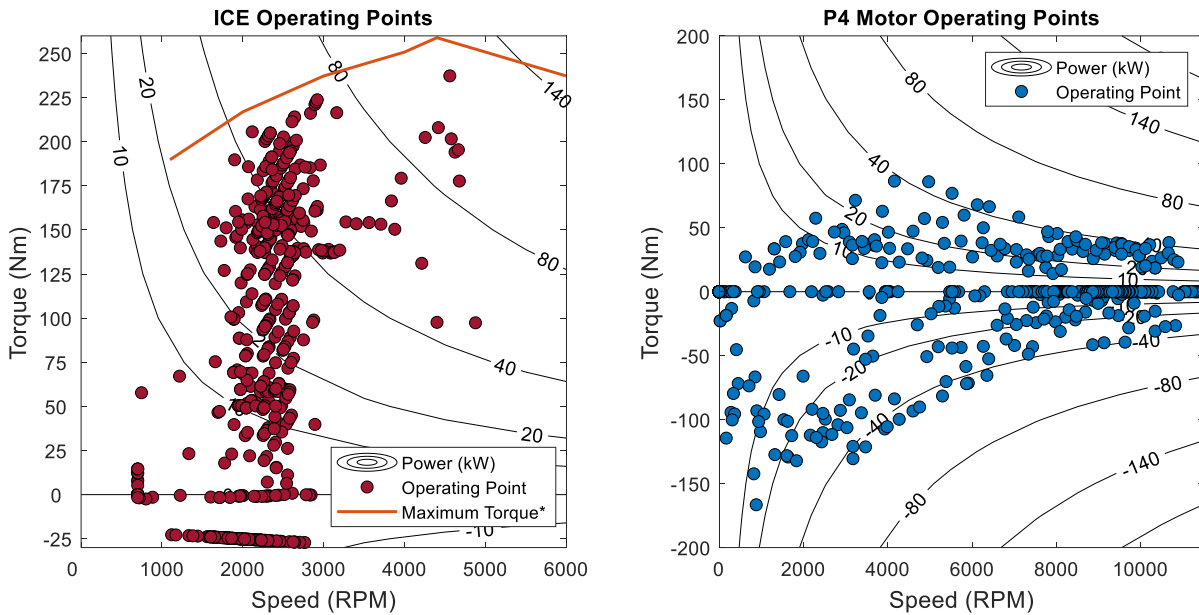


Figure 67: US06 Drive Cycle Powertrain Operating Points

Throughout the US06 drive cycle, the ICE operates under a much higher load. There are several operating points that extend into high power regions of operation. In the US06 cycle, there is little idle time and thus the ICE has few operating points at idle speeds; however, there is considerable use of FCO throughout the cycle. Considering the operating points of the electric motor, the maximum capabilities of the eRAD system are used to provide sufficient torque to achieve the drive cycle. Due to the high speeds of the US06 drive cycle, the electric motor exceeds its maximum speeds. In the real-world vehicle, the disconnect clutch in the eRAD system would open allowing the shaft to free spin at high speeds. This transaction is observed as the zero-torque production at the highest speeds on the plot of electric motor operating points. The ICE efficiency determined from ICE only operation of the vehicle over the UDDS drive cycle is shown in Figure 68.

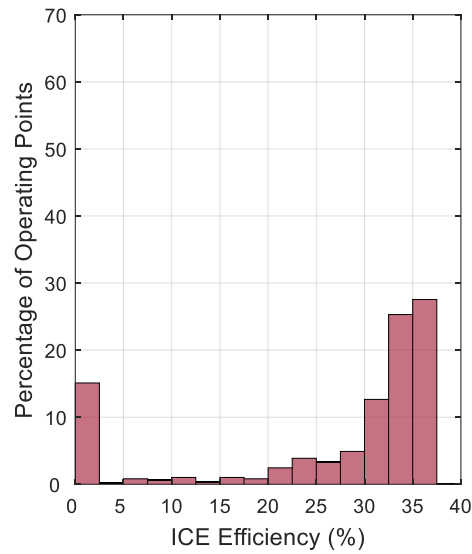


Figure 68: Efficiency Distribution of ICE Only Operation for US06 Drive Cycle

The efficiency distribution of the hybrid powertrain operation is shown in Figure 69.

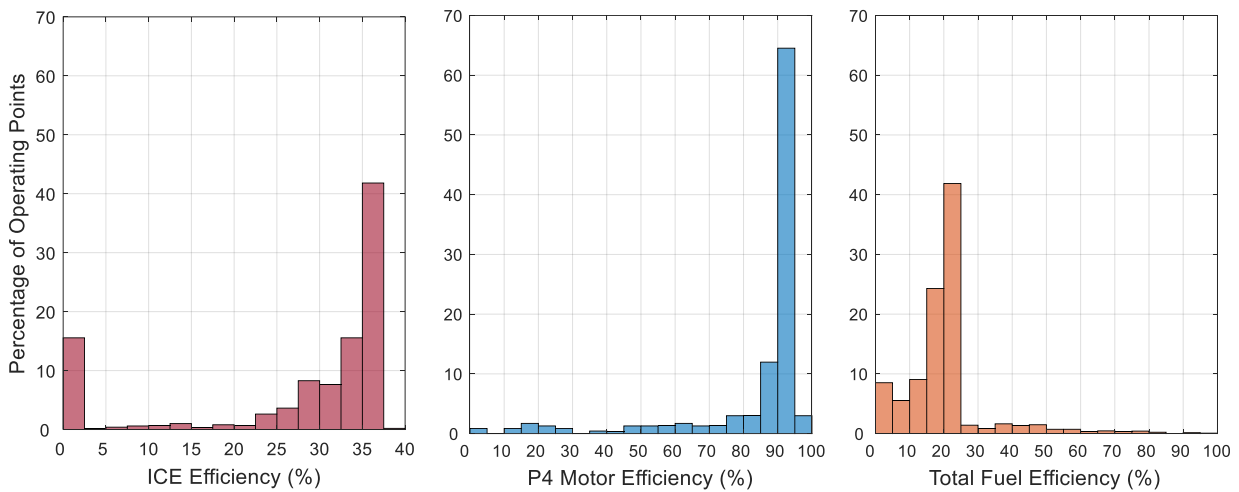


Figure 69: US06 Optimal Control Policy Operating Efficiency Distribution

The high-power requirements of the US06 drive cycle allow the ICE to operate at high efficiency in engine only operating modes. The additional torque from the electric motor improves the efficiency by allowing the ICE to operate within its highest efficiency region for 45% of the drive cycle. The remaining efficiency distribution is largely skewed toward higher ICE efficiencies with the exception of idle which only accounts for 10% of the operating points of the ICE. A critical

consideration of this drive cycle is that the ICE operates in FCO region for 17.5% of the cycle. The electric motor operates above 90% efficiency for 65% of operating points with the remaining distribution skewed toward higher efficiencies, however, the electric motor is not used for 60.1% of the drive cycle. The resulting overall efficiency is between 20 and 25% efficiency for half of the drive cycle.

The cruise portion of the US06 drive cycle has significantly higher torque requirements than previous cycles examined in this analysis as shown in Figure 70.

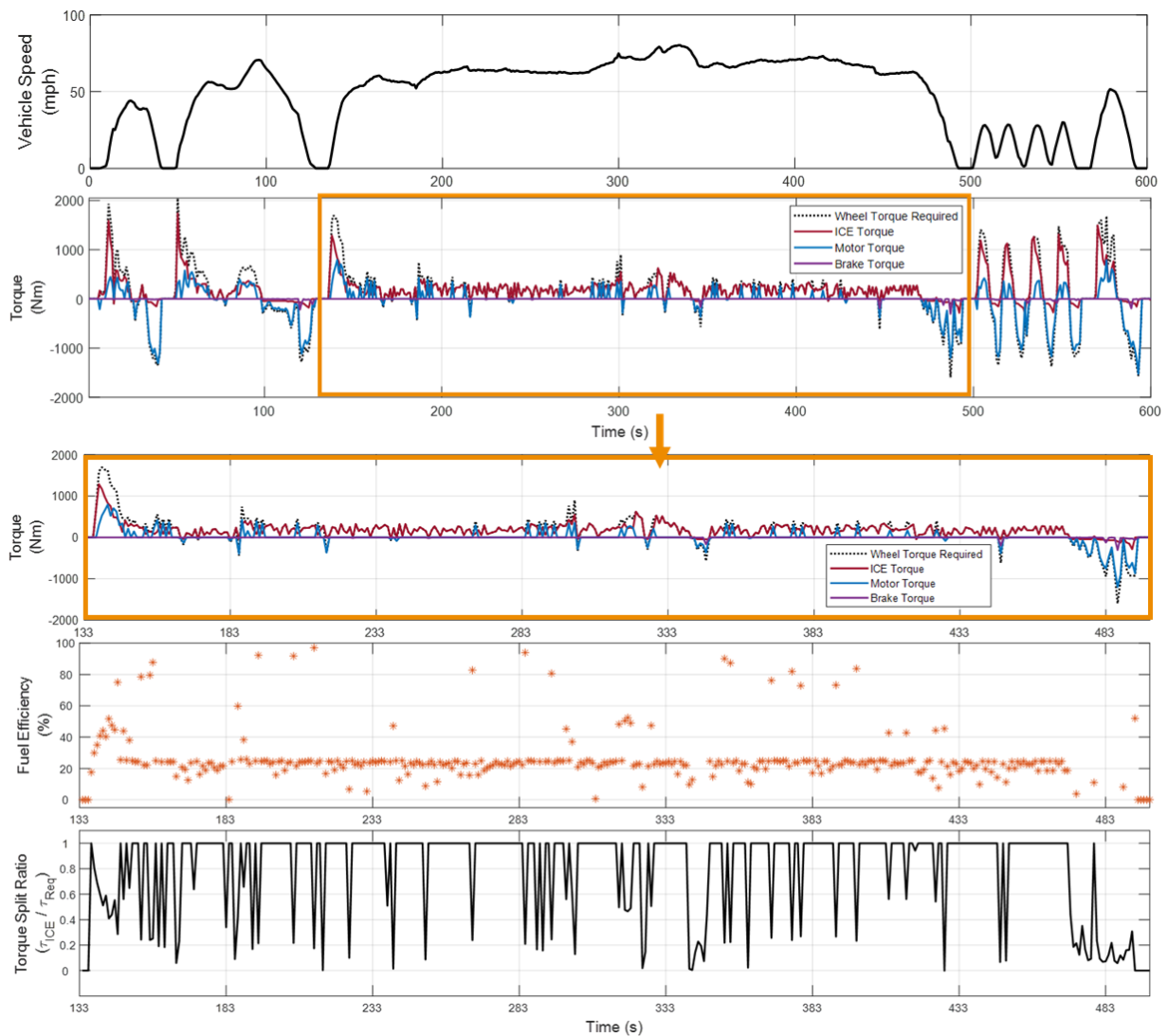


Figure 70: Torque Production Analysis of Cruise Portion of US06 Drive Cycle

Throughout the cruise portion of the US06 drive cycle, the ICE is the primary producer of torque, however, the electric motor contributes large portions of the torque at various points throughout this driving condition. This is likely due to the excessive regenerative braking available in the US06 drive cycle. Due to the availability of electrical energy, the motor is used to keep the ICE operating at its highest efficiency. The majority of regenerative braking occurs near the end of the US06 drive cycle where multiple sharp acceleration and braking events occur. The torque production near the end of the US06 cycle is shown in Figure 71.

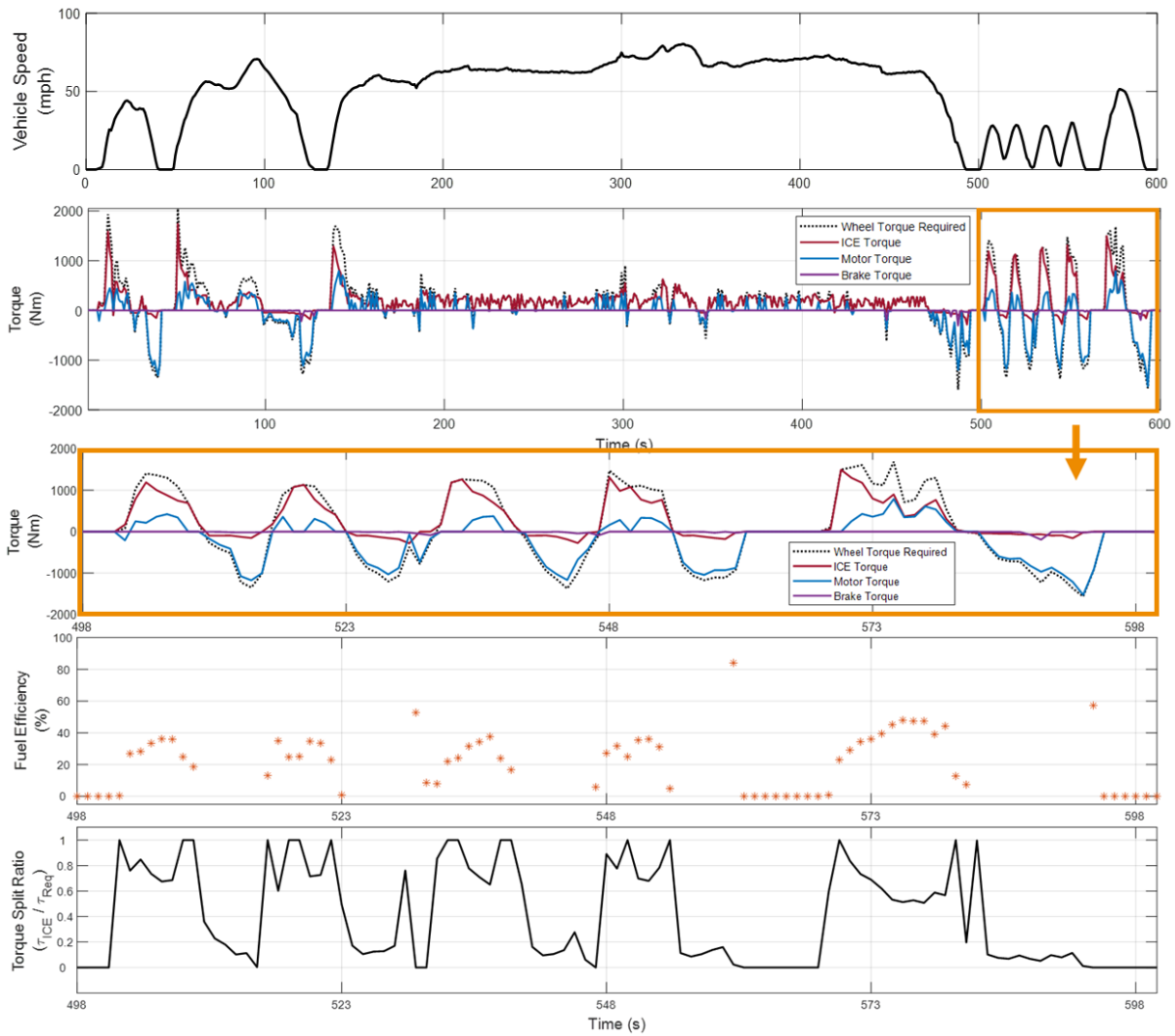


Figure 71: Torque Production Analysis of Stop and Go Portion of US06 Drive Cycle

In this portion of the US06 drive cycle, there is significant use of the electric motor to assist with the sharp accelerations and decelerations. Throughout this portion of the drive cycle, substantial regenerative braking energy is captured by the electric motor. Looking deeper into the amount of torque split, the torque split ratio of the positive and negative operating regime is shown in Figure 72 and Figure 73, respectively.

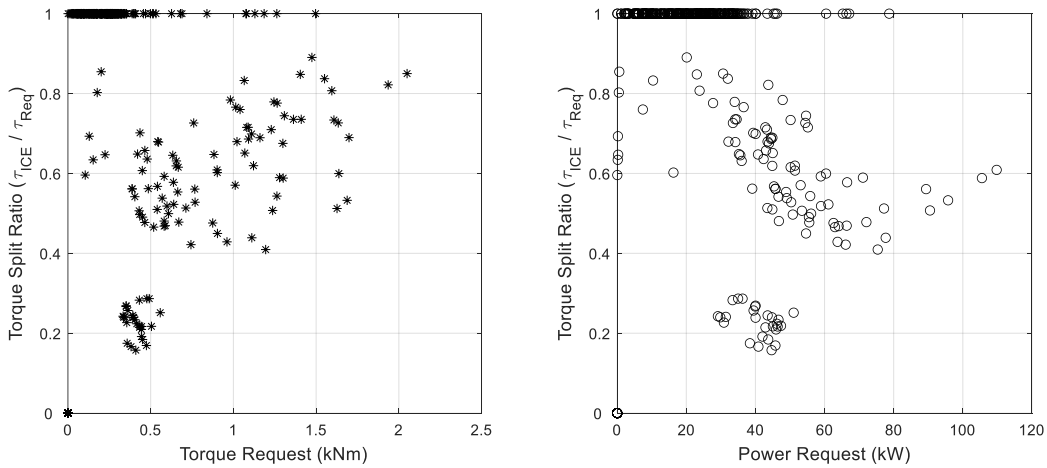


Figure 72: US06 Torque Split Ratio in Positive Torque and Power Regime

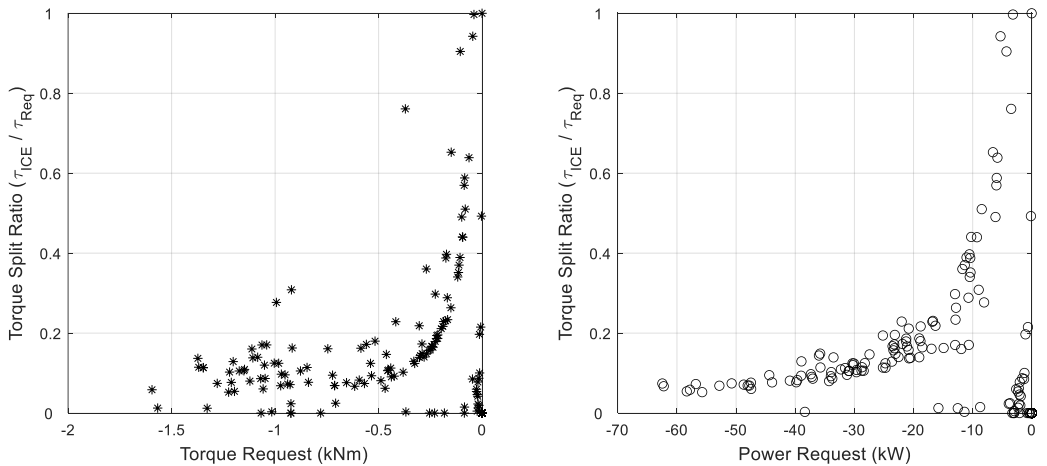


Figure 73: US06 Torque Split Ratio in Negative Torque and Power Regime

In Figure 72, there is a clear grouping of low TSR values around 500Nm and 40kW. These values occur during the electric motor assist throughout the cruise portion of the US06 drive cycle shown in Figure 70. The remaining points in the positive regime follow a similar trend as previous drive

cycles with a decreasing relationship as power request increases. In the negative regime, TSR is less useful to discuss as it is a result of the ICE in FCO and electric motor used for regen.

The energy consumption of the vehicle over the US06 drive cycle is shown in Figure 74.

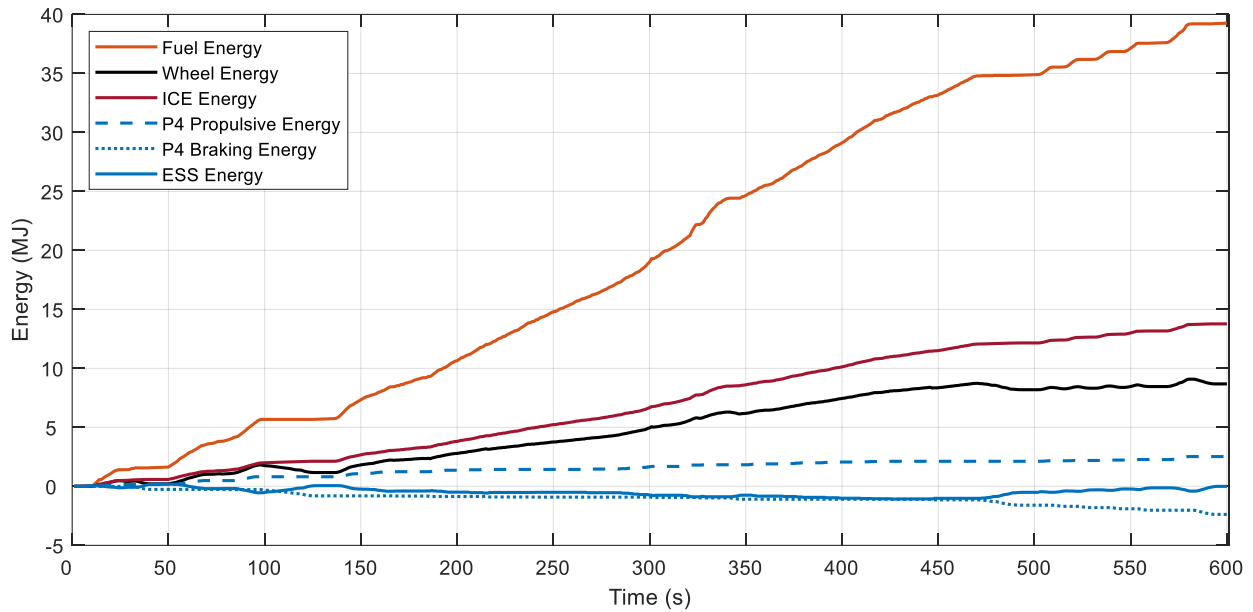


Figure 74: Energy Analysis for US06 Drive Cycle

Table 14 summarizes the relevant overall performance metrics and the energy values for the US06 drive cycle.

Table 14: US06 Energy Summary

Parameter	Value
Fuel Economy (MPG)	24.64
Fuel Used (g)	914.00
Fuel Energy (MJ)	39.24
Net Wheel Energy (MJ)	14.43
Net Electrical Energy (MJ)	0
P4 Propulsive Energy (MJ)	2.37

The fuel economy of this drive cycle is much lower than other cycles examined in this work. This is due to the aggressive driving that occurs throughout this drive cycle. Figure 74 shows that the fuel energy far exceeds the energy generated at the wheels. However, although significant amounts of fuel were consumed over the drive cycle, the ICE operated near peak efficiency for the majority of the drive cycle. This was accomplished with the help of the electric motor using the significant amount of kinetic energy extracted from the drive cycle. With the electric motor assist, the wheel energy produced exceeded the energy produced from the ICE.

5.3.4 UDDS

The UDDS drive cycle simulates the urban driving conditions in stop-and-go traffic [48]. The speed and time profile of the UDDS drive cycle with associated wheel torque and power requirements are shown in Figure 75.

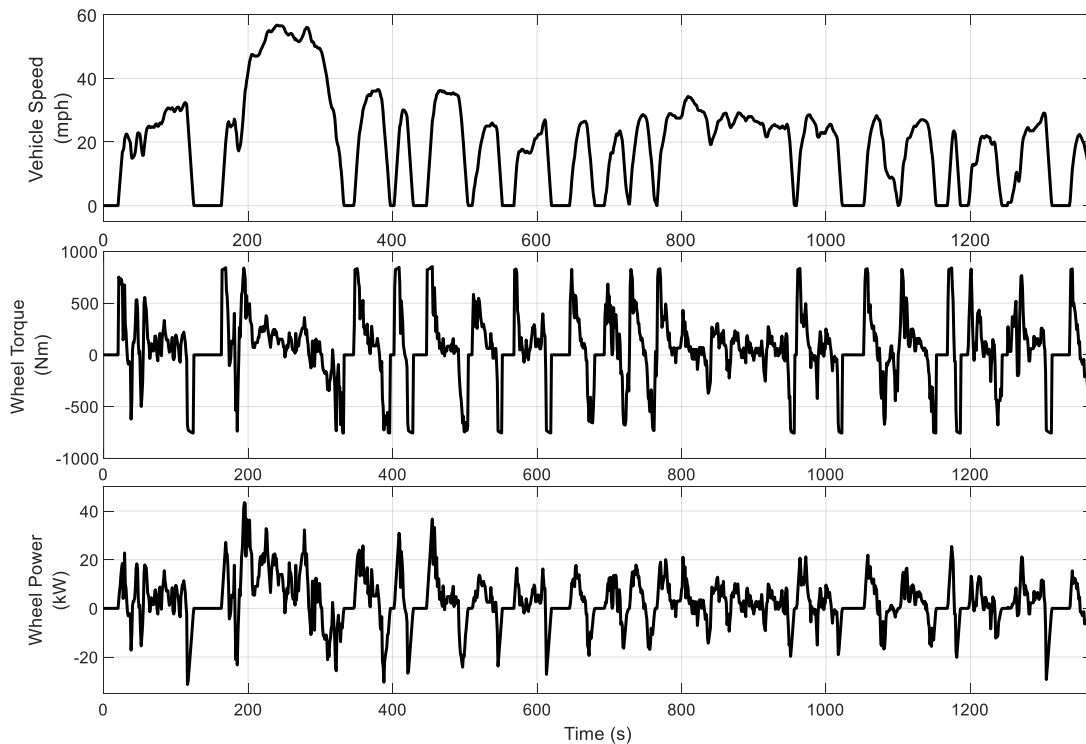


Figure 75: UDDS Drive Cycle Speed Torque and Power Requirements

The torque and power requirements for the UDDS drive cycle are relatively consistent with the EMC City drive cycle. Both drive cycles capture urban driving conditions; however, UDDS is much longer with many more rapid acceleration and deceleration events. The shift schedule for the UDDS drive cycle is shown in Figure 76.

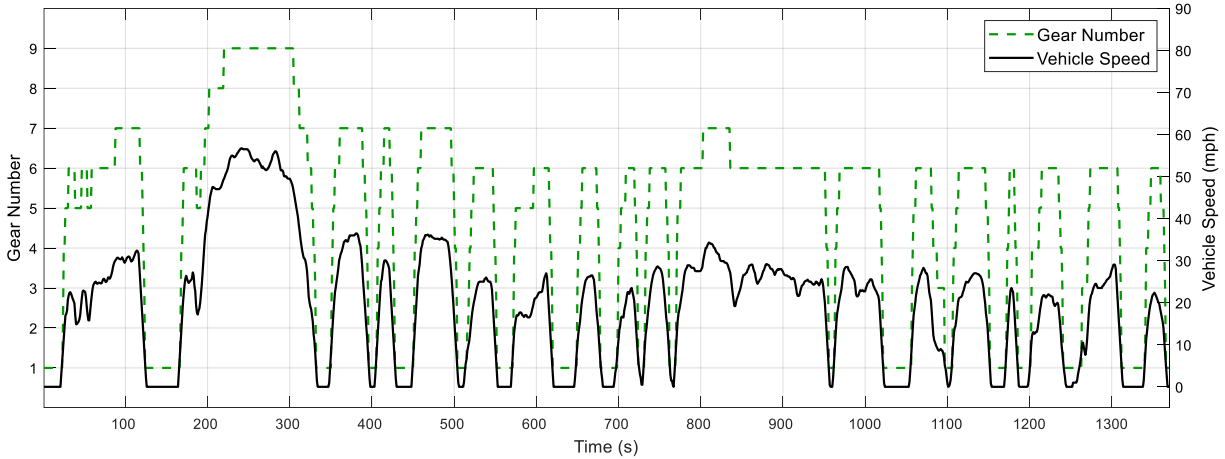


Figure 76: UDDS Shift Schedule

Additional relevant parameters for the US06 drive cycle and DP algorithm initialization parameters are shown in Table 15.

Table 15: UDDS Drive Cycle Information and DP Algorithm Initialization Parameters

DP Algorithm		UDDS	
Parameter	Value	Parameter	Value
Maximum SOC (%)	80	Distance (mi)	7.45
Minimum SOC (%)	20	Total Time (s)	1370
Target SOC (%)	50	Sample Time (Hz)	1
Number of Grid Points	5,000	Maximum Acceleration (m/s ²)	1.48
Δ SOC (%)	0.012	Maximum Deceleration (m/s ²)	-1.48
ΔP_{batt} (W)	648	Mechanical Energy Required (MJ)	9.46

The stop and go nature of the UDDS drive cycle is generally detrimental to conventional vehicles as the ICE operates inefficiently at low load and low speeds. This drive cycle provides insight to how beneficial the electric powertrain can be for urban conditions. The powertrain power output results for the UDDS drive cycle are shown in Figure 77.

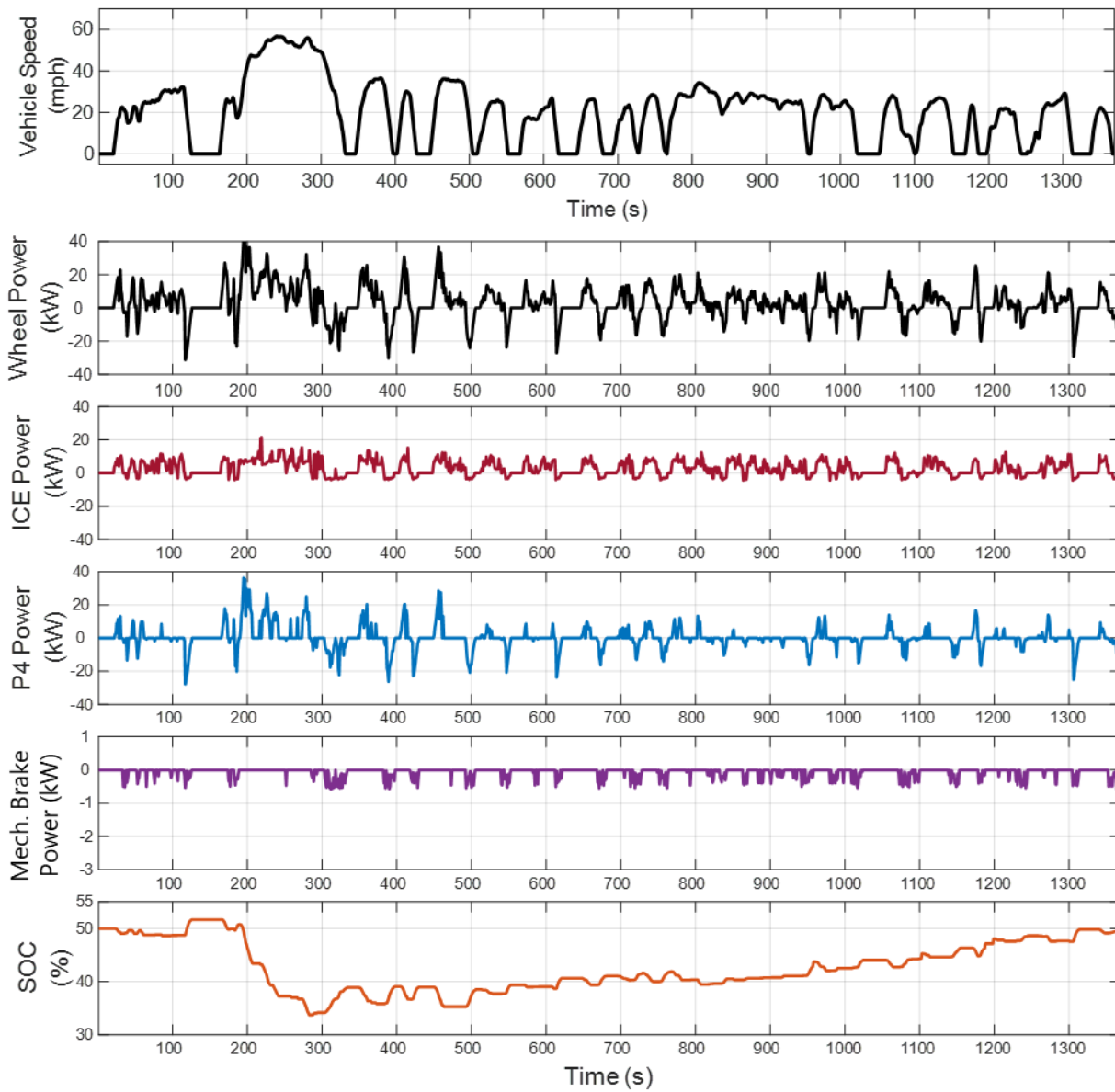


Figure 77: UDDS Drive Cycle Powertrain Power Output

In this drive cycle, there is significant use of the electric motor with the various acceleration and braking portions of the drive cycle. On several occasions, the ICE is assisted by the electric motor, likely due to the amount of regenerative braking available in the drive cycle. Due to this availability, the electric motor can be used more often outside of vehicle launch assist. The operating points of the powertrain are shown in Figure 78.

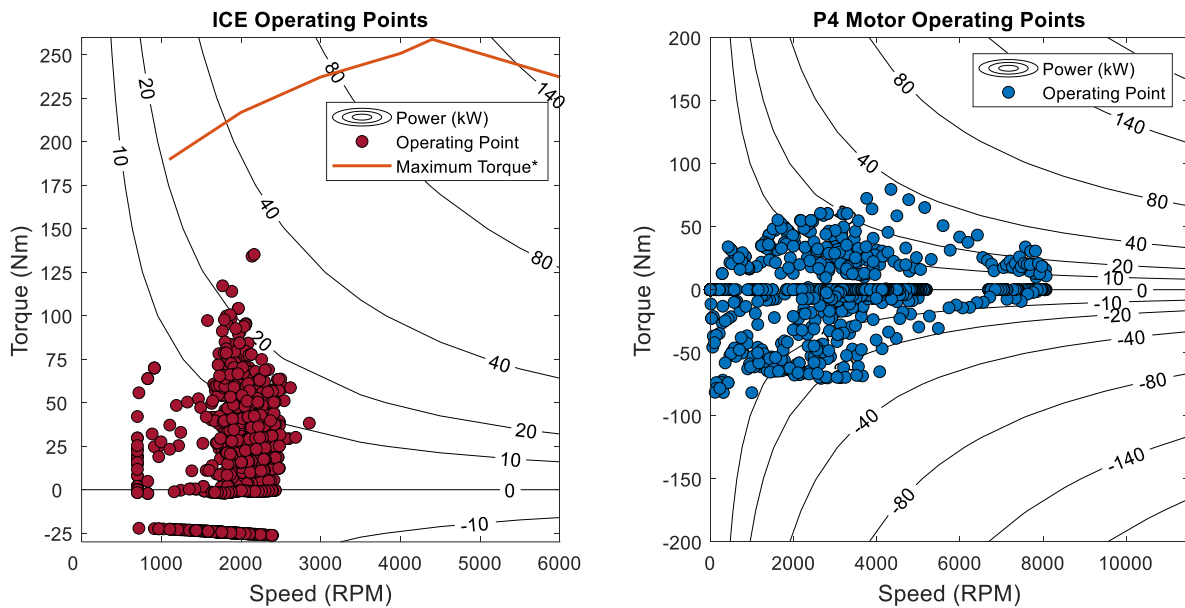


Figure 78: UDDS Drive Cycle Powertrain Operating Points

In the UDDS drive cycle, the ICE operates at low power as expected by the low torque requirements to meet the drive cycle. The electric motor operates frequently in both the positive and negative torque regimes as it is heavily used to launch and brake the vehicle. The ICE efficiency determined from ICE only operation of the vehicle over the UDDS drive cycle is shown in Figure 79.

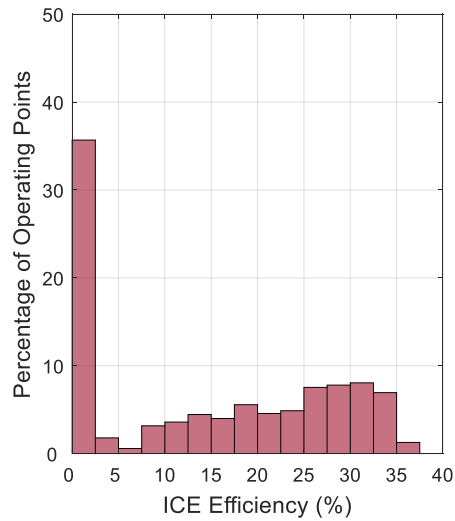


Figure 79: Efficiency Distribution of ICE Only Operation for UDDS Drive Cycle

The efficiency distribution of the powertrain components operating in hybrid mode is shown in Figure 80.

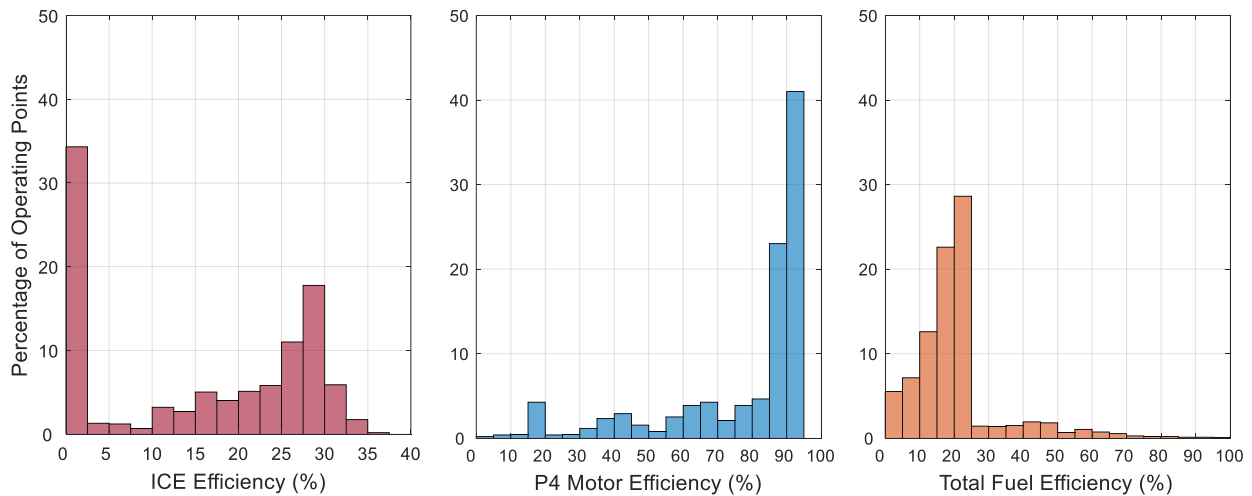


Figure 80: UDDS Optimal Control Policy Operating Efficiency Distribution

The distribution of ICE operating efficiencies shifts from a near uniform distribution for ICE only operation to a significant peak between 25 and 30% for hybrid operation, with a normal distribution spread across the admissible efficiency range. As expected, due to the stop and go nature of the UDDS drive cycle, the ICE is at idle for 26% of the drive cycle where it burns fuel and produces no propulsive torque while the ICE is in FCO for only 15.7% of the drive cycle. The electric motor

operates above 85% efficiency for over 65% of the drive cycle with few points operating at low efficiencies. The electric motor is not used for 61% of the drive cycle. The overall operating efficiency of the system is distributed between primarily between 10 and 20% efficiency with few points operating at higher efficiencies. A portion of the UDDS drive cycle is examined to identify the torque production by powertrain components. This analysis is shown in Figure 81.

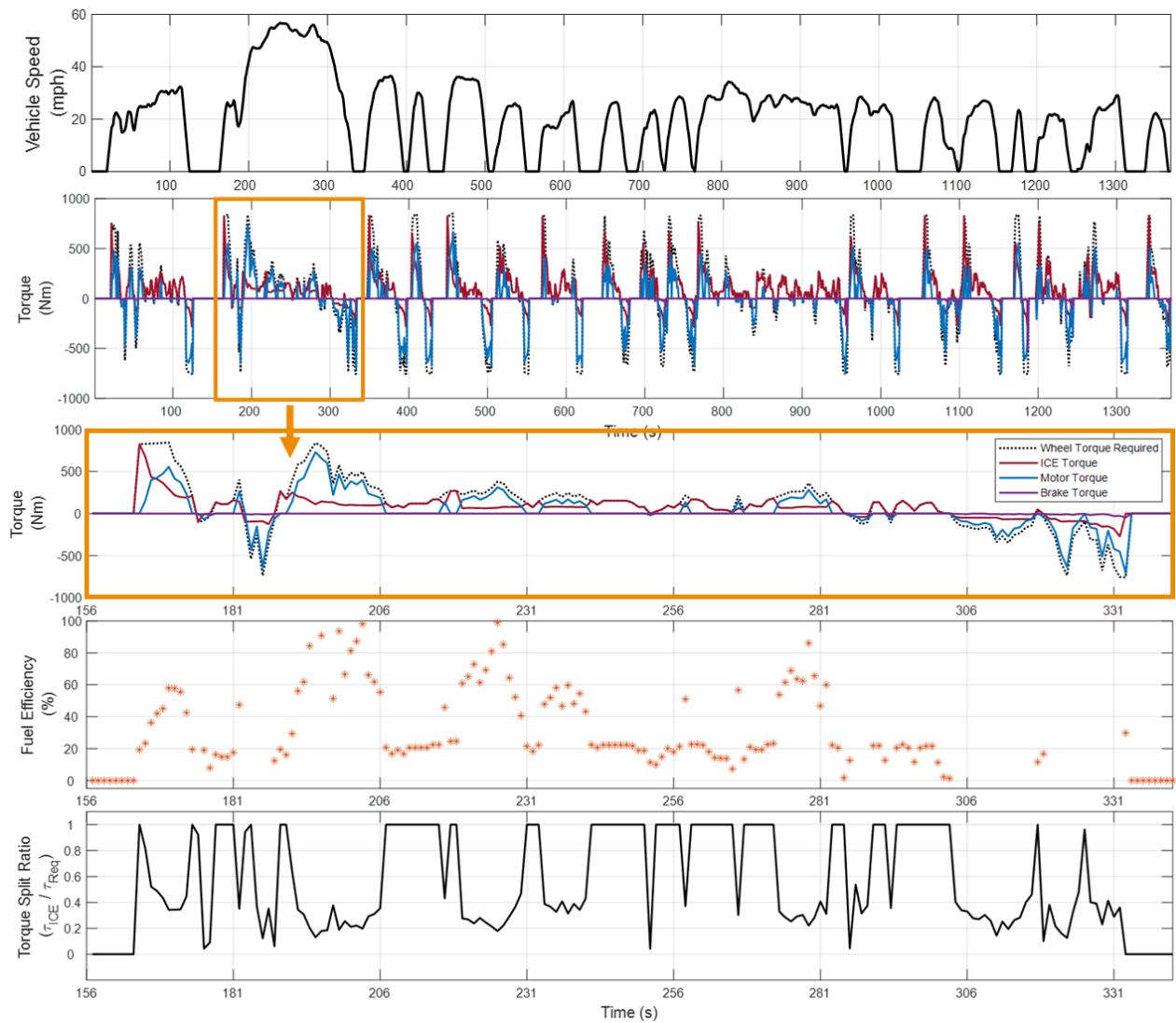


Figure 81: Torque Production Analysis of UDDS Drive Cycle

Similar to the EMC City drive cycle, the electric motor is used as the vehicle is launch, however, throughout the cruising portion of the drive cycle, the electric motor produces substantial torque

where in other drive cycles it typically does not. This is likely due to the abundance of regenerative braking energy the electric motor is able to extract from the cycle. This occurs as more energy recovered using regenerative braking allow for more use of the electric motor to provide propulsive torque due to the CS constraint. The torque split ratio in both positive and negative operating regimes is shown in Figure 82 and Figure 83, respectively.

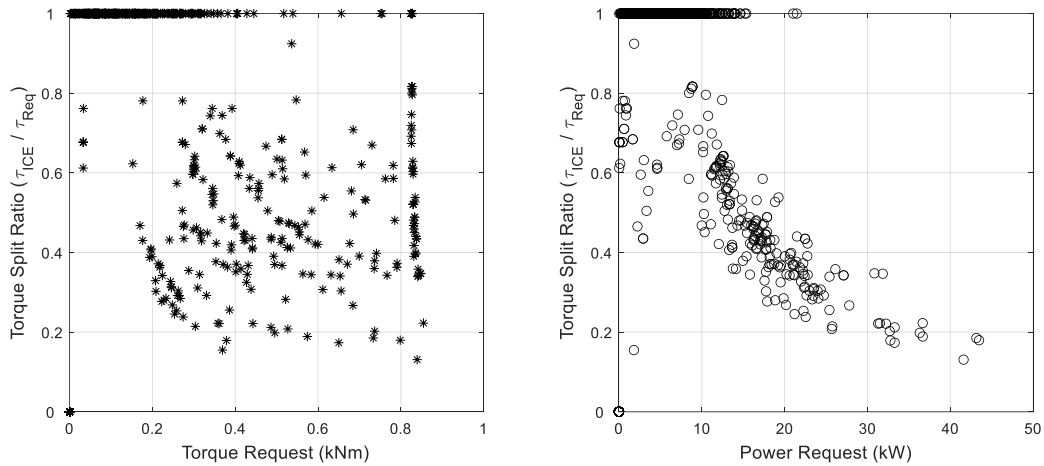


Figure 82: UDDS Torque Split Ratio in Positive Torque and Power Regime

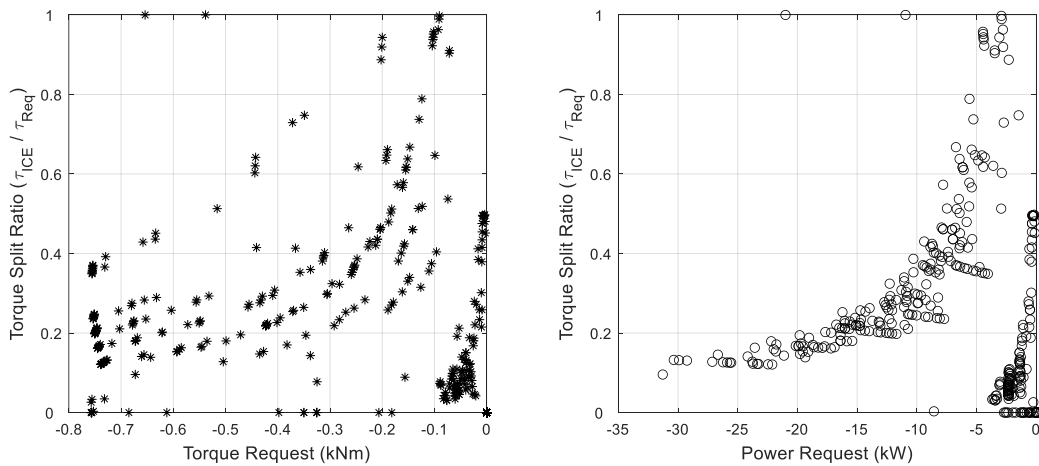


Figure 83: UDDS Torque Split Ratio in Negative Torque and Power Regime

In the positive torque and power regime, there is a similar trend as previously examined in other drive cycles for ICE only operation at low torque and power requirements, however, in the UDDS

cycle, the largest torque requests correspond to lower values of TSR. These large torque requests are likely vehicle launches. A similar trend as in previous drive cycles is identified with decreasing TSR values for increasing power requests. The negative torque and power regime follow the same trend as previously discussed in other drive cycles with a decreasing relationship for increasing magnitude negative torque and power values. The energy analysis for the UDDS drive cycle is shown in Figure 84.

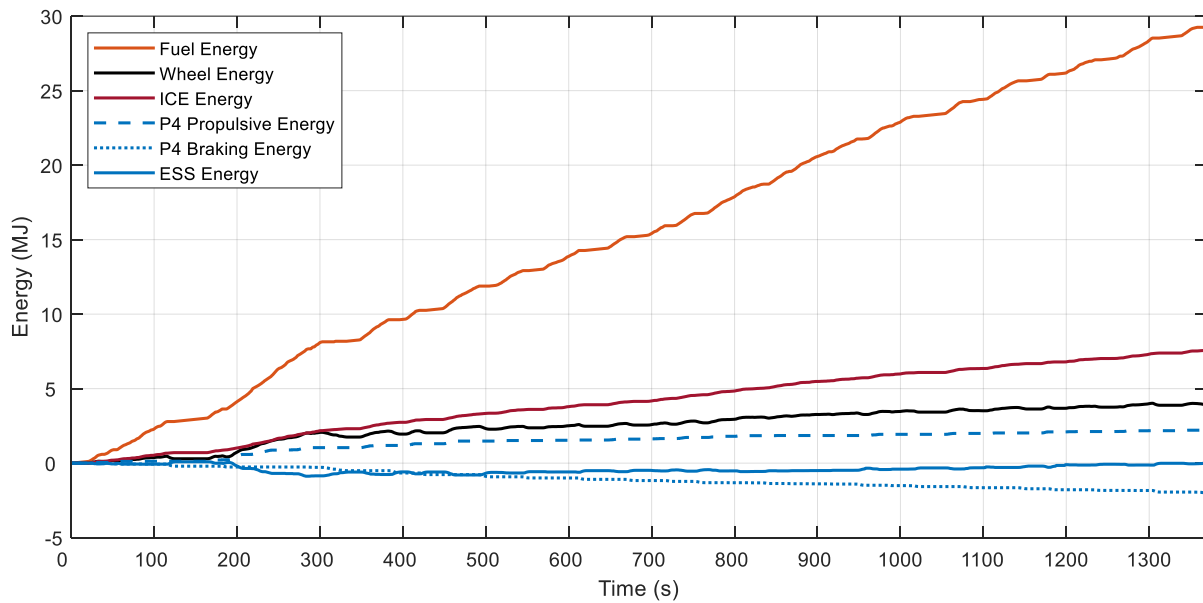


Figure 84: Energy Analysis for UDDS Drive Cycle

Table 16 summarizes the relevant overall performance metrics and the energy values for the UDDS drive cycle.

Table 16: UDDS Energy Summary

Parameter	Value
Fuel Economy (MPG)	30.74
Fuel Used (g)	681.61
Fuel Energy (MJ)	29.29
Net Wheel Energy (MJ)	9.46
Net Electrical Energy (MJ)	0
P4 Propulsive Energy (MJ)	1.98

As expected, the fuel economy of this drive cycle is comparable to the EMC City drive cycle. The 1.98 MJ of kinetic energy captured by regenerative braking was used to save a total of 7.80 MJ of fuel energy from the comparison of ICE only operation to hybrid operation of the vehicle. The resulting fuel economy of the vehicle following the optimal control policy determined for the UDDS drive cycle is 30.74 MPG.

5.3.5 *HWFET*

The HWFET drive cycle represents highway driving conditions under 60 mph [48]. The speed and time profile with associated wheel torque and power requirements are shown in Figure 85.

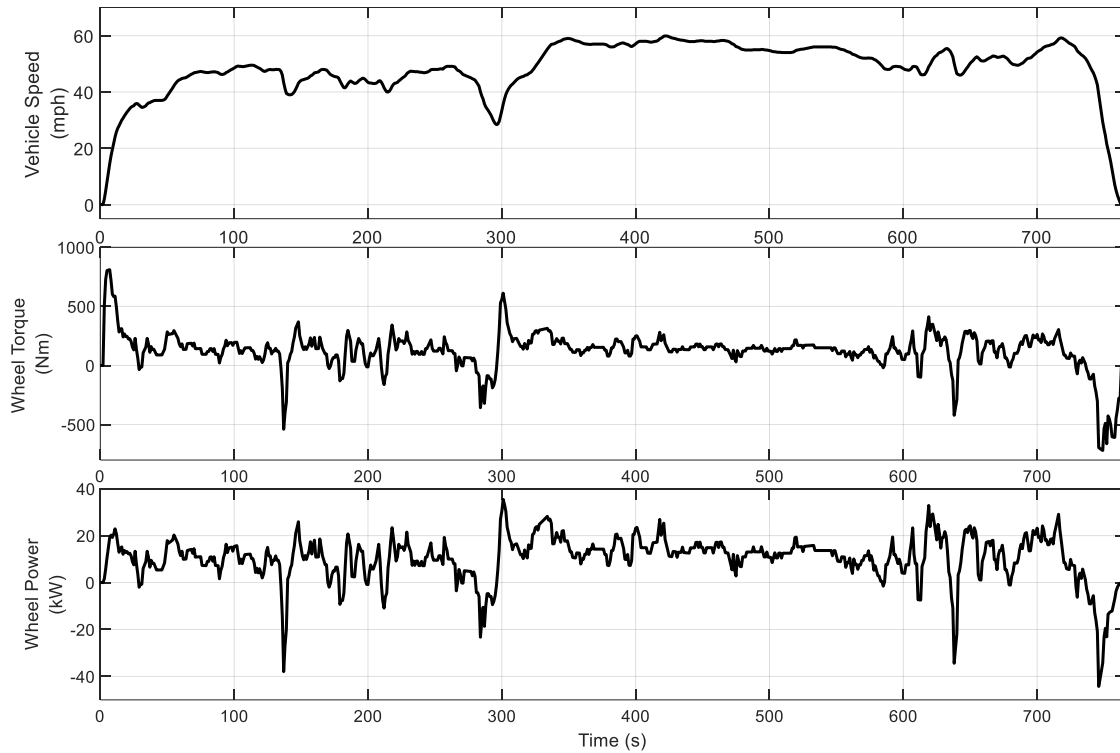


Figure 85: HWFET Drive Cycle Speed, Torque, and Power Requirements

The torque and power requirements of the HWFET drive cycle are comparable to the EMC Highway drive cycle; however, in this drive cycle there is far less regenerative braking opportunities and longer cruise conditions. The drive cycle is also significantly shorter than the EMC Highway drive cycle. The shift schedule of the HWFET drive cycle is determined according to the strategy discussed in Section 4.3.1 and is shown in Figure 86.

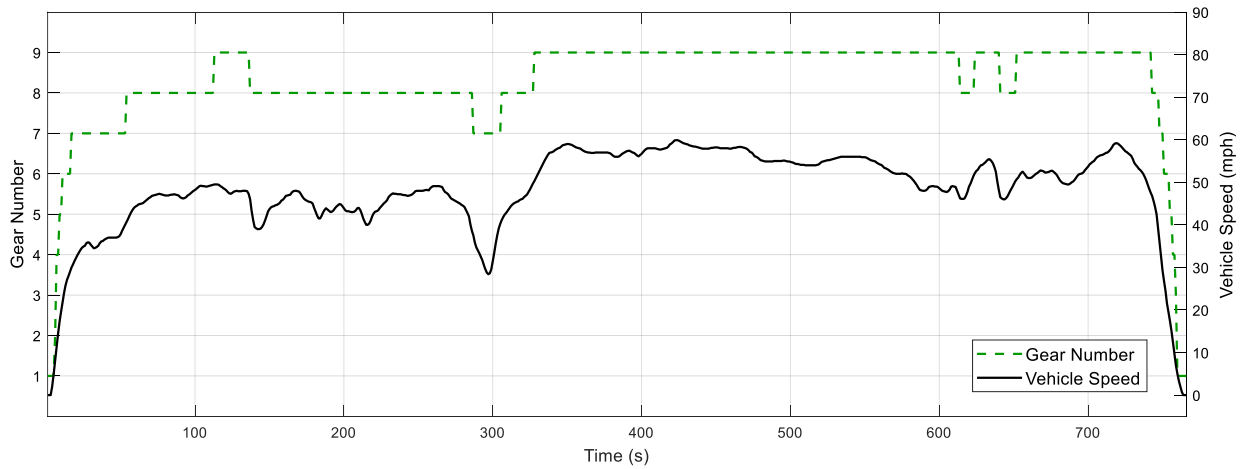


Figure 86: HWFET Shift Schedule

After the vehicle reaches cruising speed there is little variation in the gear position. Relevant information for the HWFET drive cycle is shown in Table 17.

Table 17: HWFET and DP Algorithm Initialization Parameters

DP Algorithm		HWFET	
Parameter	Value	Parameter	Value
Maximum SOC (%)	80	Distance (mi)	10.26
Minimum SOC (%)	20	Total Time (s)	766
Target SOC (%)	50	Sample Time (Hz)	1
Number of Grid Points	5,000	Maximum Acceleration (m/s ²)	1.43
Δ SOC (%)	0.012	Maximum Deceleration (m/s ²)	-1.48
ΔP_{batt} (W)	648	Mechanical Energy Required (MJ)	9.61

The powertrain power output results for the EMC Highway drive cycle are shown in Figure 87.

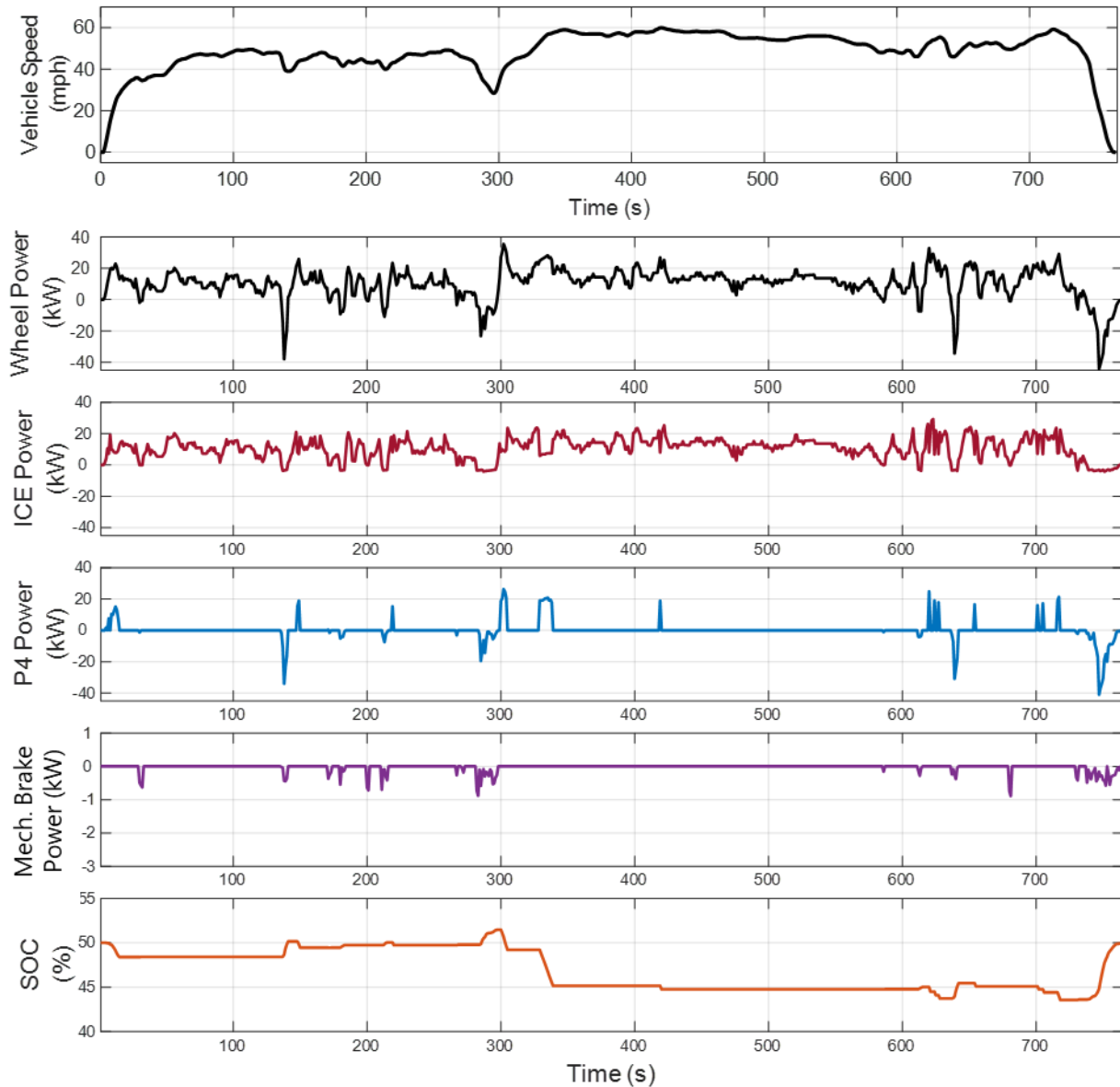


Figure 87: HWFET Powertrain Power Output

In the HWFET drive cycle, the electric motor is primarily used to launch the vehicle at the start of the test and provide braking. There is very little regenerative braking available in the HWFET drive cycle and thus the torque production of the electric motor is limited. The SOC reaches a maximum value of 51.48% and a minimum of 43.57%. The operational points of the powertrain for the HWFET cycle are shown in Figure 88.

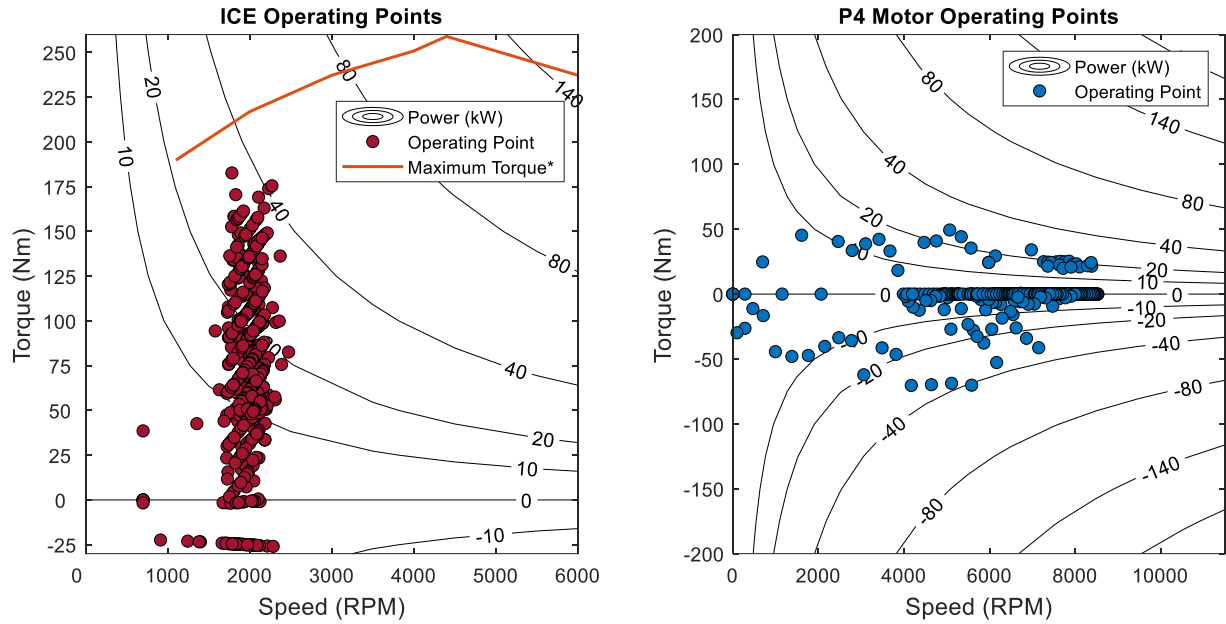


Figure 88: HWFET Drive Cycle Powertrain Operating Points

Throughout the HWFET cycle, there is little variation in speed and thus the ICE operates within a narrow window of speeds over a large range of torques. The electric motor is shown to have operated significantly at high speed. This occurred during the second acceleration event where the majority of electrical energy was spent throughout the drive cycle. The ICE efficiency determined from ICE only operation of the vehicle over the UDDS drive cycle is shown in Figure 89.

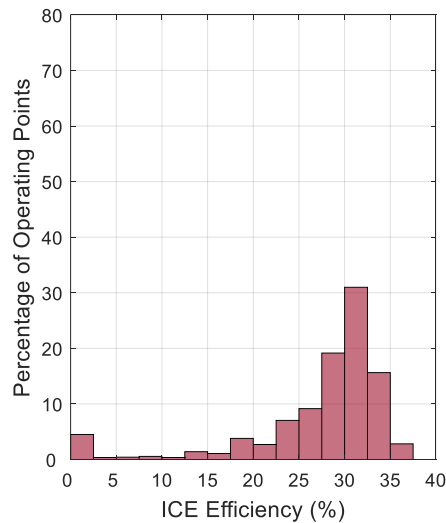


Figure 89: Efficiency Distribution of ICE Only Operation for HWFET Drive Cycle

The operating efficiency distribution for the powertrain is shown in Figure 90.

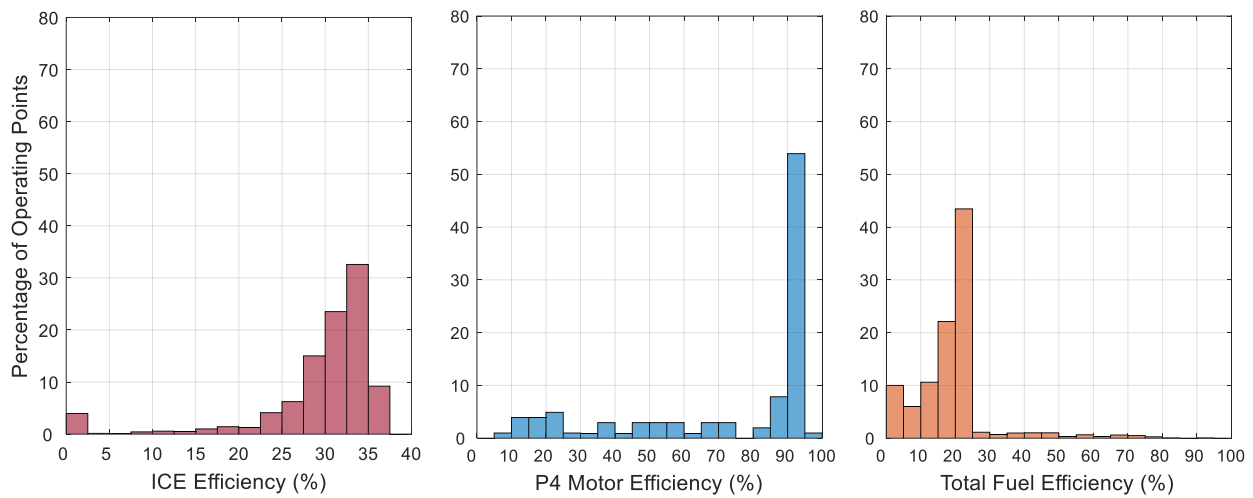


Figure 90: HWFET Optimal Operating Efficiency Distribution

For both ICE only and hybrid operation the operating efficiency of the ICE is skewed to the right. There is improvement for HEV operation with increased ICE efficiency with operating between 30-35% efficiency for over 55% of the drive cycle. Throughout the HWFET drive cycle the ICE was at idle for less than 2% of operating points and in FCO for only 7.7% of the cycle. The electric motor operated above 90% efficiency for 53% of the drive cycle with near uniform distribution for lower efficiencies. This is likely due to the little use of the electric motor as it did not operate for 86% of the drive cycle. The total system operated between 15 and 25% for the nearly 87% of the drive cycle. This low overall efficiency is due to the reliance on the ICE to provide the majority of propulsive torque throughout the HWFET drive cycle. The torque evaluation for the HWFET drive cycle is shown in Figure 91.

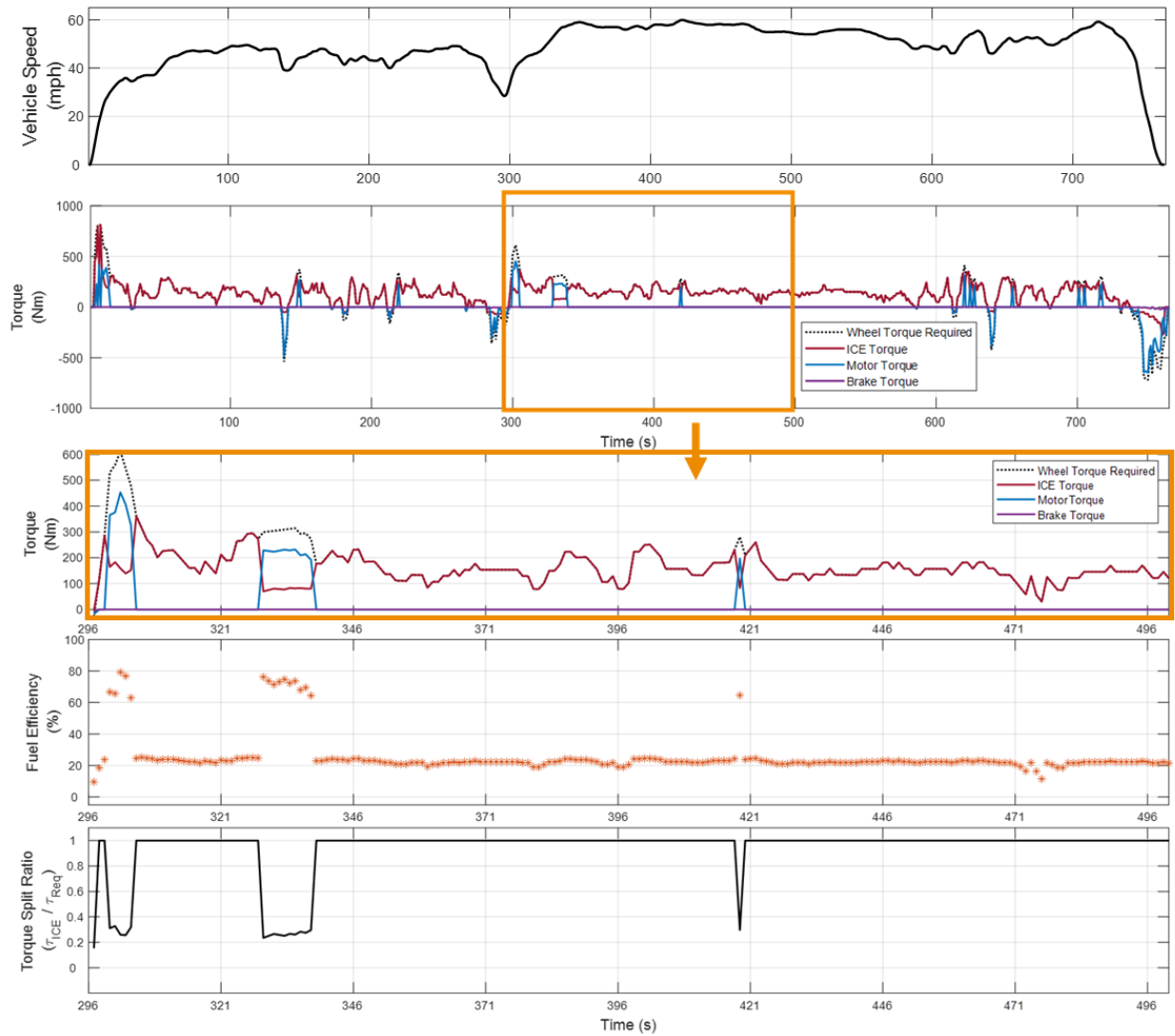


Figure 91: Torque Production Analysis of HWFET Drive Cycle

A similar trend as previous drive cycles is exhibited where the electric motor is only used for the large torque requests corresponding with major acceleration events. An interesting transaction occurs near 330s where the electric motor provides significant propulsive torque. This location is likely a large fuel consumption power for the ICE and thus was the best location to spend the limited electrical energy available in the cycle. The torque split ratio in the positive torque and power regime is shown in Figure 92.

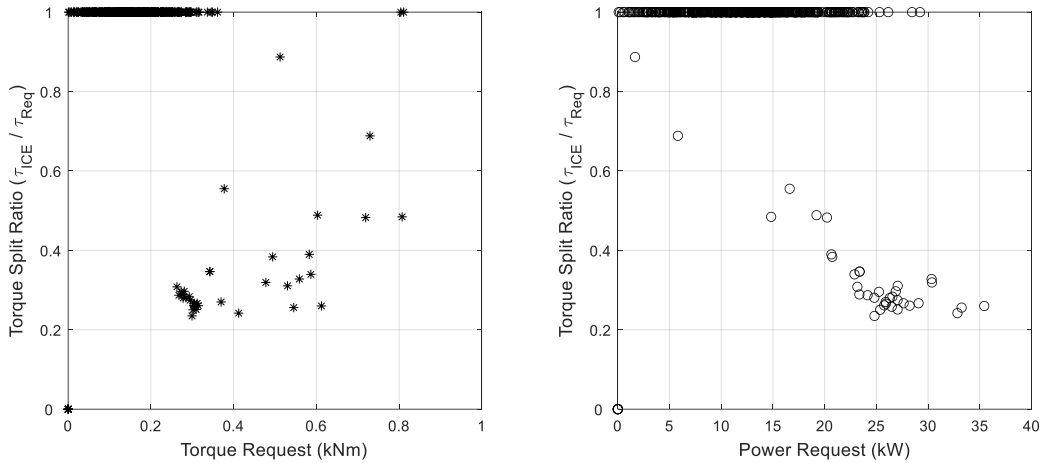


Figure 92: HWFET Torque Split Ratio in Positive Torque and Power Regime

Similarly, to previous drive cycles, the ICE operates on its own for torque requests under 250Nm and powers under 15kW. Crossing that threshold, the electric motor provides significant propulsive torque, between 20 and 60% of the request. Similar to previous drive cycles the relationship between increasing power and decreasing TSR is preserved. The torque split ratio in the negative torque and power regime is shown in Figure 93.

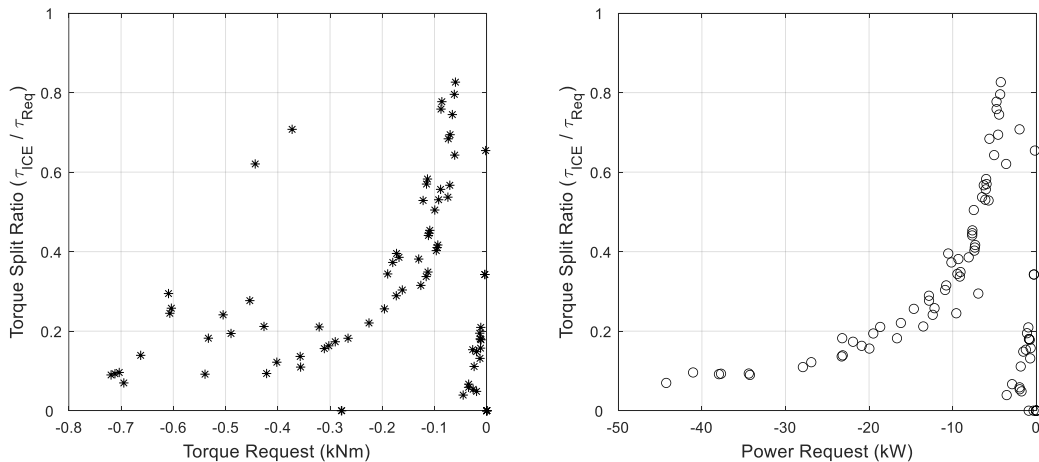


Figure 93: HWFET Torque Split Ratio in Negative Torque and Power Regime

In the negative torque and power regime, there is little significant of the TSR value as its value depends on the ICE being in FCO and regenerative braking from the electric motor. These values are not control actions as much as consequences of the powertrain operating appropriately.

The energy consumption of the vehicle over the EMC Highway drive cycle is shown in Figure 94.

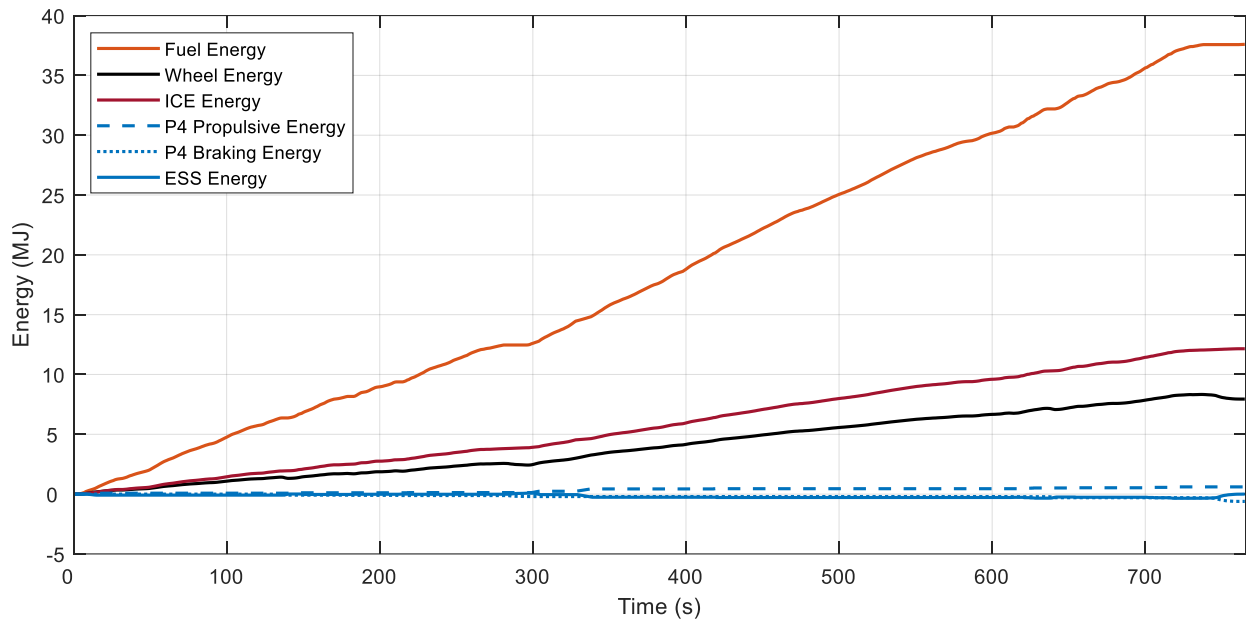


Figure 94: Energy Analysis for HWFET Drive Cycle

Table 18 summarizes the relevant overall performance metrics and energy values for the HWFET drive cycle.

Table 18: HWFET Energy Summary

Parameter	Value
Fuel Economy (MPG)	32.95
Fuel Used (g)	875.57
Fuel Energy (MJ)	37.60
Net Wheel Energy (MJ)	9.61
Net Electrical Energy (MJ)	0
P4 Propulsive Energy (MJ)	0.61

Throughout the HWFET drive cycle, there is significant fuel consumption. Due to the substantial reliance on the ICE throughout the cycle the fuel economy gain is low. There is little energy

produced by the electric motor in both braking and propulsion with a total of 0.61 MJ of energy extracted using regenerative braking. This results in a fuel energy savings of 3.47 MJ. The resulting fuel economy of the vehicle following the optimal control policy is 32.95 MPG.

5.4 Summary of Benchmark Analysis Results

In addition to the hybrid vehicle benchmarking, the non-hybrid fuel economy results for the Table 19 presents a summary of the performance results determined from the benchmark analysis.

Table 19: DP Algorithm Powertrain Performance Results

Parameter	EMC City	EMC Highway	US06	UDDS	HWFET
Non-Hybrid Fuel Economy (MPG)	24.64	33.17	19.15	23.26	29.16
Fuel Economy (MPG)	30.18	37.24	24.63	30.74	32.95
Fuel Used (g)	311.19	2238.5	914.20	681.61	875.57
Total Fuel Energy (MJ)	13.36	96.12	39.26	29.29	37.60
Total Wheel Energy (MJ)	3.87	21.89	14.43	9.46	9.61
Total ICE Energy (MJ)	3.27	26.36	13.77	7.56	12.15
Maximum SOC (%)	53.85	54.22	53.41	51.66	51.48
Minimum SOC (%)	47.02	46.09	30.07	33.74	43.57

According to the EMC rules, the combined fuel economy of the vehicle is determined based on the following relationship:

$$\text{EMC Combined Fuel Economy} = \frac{1}{\left(\frac{0.55}{\text{EMC City}} + \frac{0.45}{\text{EMC Highway}}\right)} \quad (5.10)$$

with this relationship the combined fuel economy of the hybrid 2019 Chevrolet Blazer developed by the WVU EcoCAR team is 32.99 MPG compared to the 21.6 MPG of the stock 3.6L 2019 Chevrolet Blazer.

Throughout the benchmark analysis, the torque split ratio was examined as a function of torque request and power request. The relationship between the optimal torque split ratio and these parameters is often used to determine rule sets for heuristic controllers. To summarize the relationships determined from each drive cycle, a composite plot of the optimal torque split ratio is generated to identify overall control policies that may be useful for general operation of the vehicle. This composite plot is shown in Figure 95 and Figure 96.

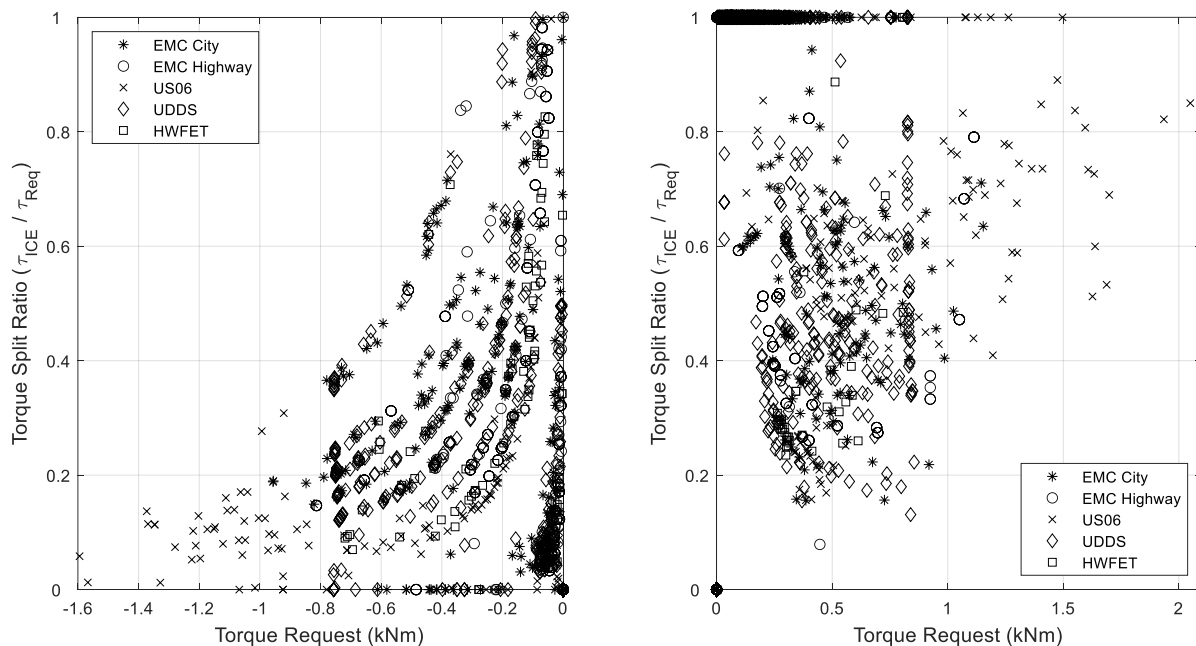


Figure 95: Optimal Torque Split Ratio as a Function of Torque Request for Multiple Drive Cycles

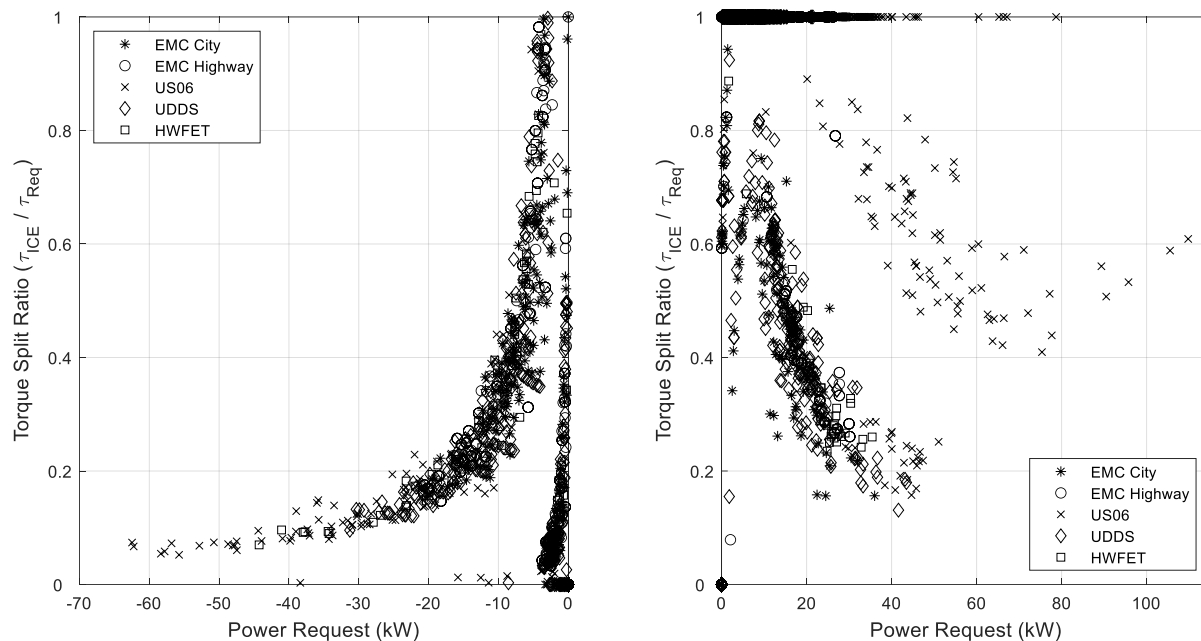


Figure 96: Optimal Torque Split Ratio as a Function of Power Request for Multiple Drive Cycles

Several general control policies can be established from these composite figures. In the positive regime of Figure 95, the torque split ratio never drops much below 0.6 unless the torque request exceeds roughly 200Nm. There is a clear relationship between power request and torque split ratio in both the positive and negative regimes. The data shown in Figure 95 and Figure 96 can be used to generate a quasi-optimal torque split ruleset by using curve fitting tools to generate trendlines for this data.

For urban driving conditions captured by the EMC City and UDDS drive cycles, the ICE operates over a range of low speeds and torques with the electric motor used to assist the engine in stop-and-go scenarios. In highway driving conditions captured by the EMC Highway and HWFET drive cycles, the ICE operates within a tighter envelope around 2000RPM compared to urban driving with higher torque production. The electric motor is not heavily relied on for torque production in the highway driving scenarios. The seldom times that the electric motor does operate in this condition typically correspond to large power requests. The optimal control policy of the US06

drive cycle does not follow the trends of either urban or highway driving. The aggressive nature of this drive cycle result in wide operating envelopes of both powertrains. The ICE operates over nearly its entire admissible range of speeds and torques with several values reaching the maximum available torque from the ICE and electric motor. This operating strategy is likely due to the vehicle model attempting to keep up with the aggressive driving conditions of the US06 drive cycle.

5.5 Computational Considerations

The step size of 5,000 steps, the memory requirement for the 5,000 by 5,000 arrays are 200MB each. At each stage, 24 5,000 by 5,000 matrices are computed with 150 smaller arrays and variables with the total memory corresponding to the length of the drive cycle. The shortest drive cycle is US06 at 600 seconds which has a memory requirement of 8.58GB and the longest drive cycle is EMC Highway at 2962 seconds with a memory requirement of 9.97GB. Doubling the step size to 10,000 steps substantially increases the runtime and memory requirements. A 10,000 by 10,000 matrix requires 800MB each with US06 and EMC Highway requiring a total of 24.43GB and 26.47GB, respectively. A PC equipped with 32GB of memory can execute the DP algorithm with 10,000 steps providing higher resolution in the state grid, however, the additional runtime is significant.

For the EMC City drive cycle, the runtime for 5,000 steps is 26.5 minutes while 10,000 steps have a runtime of 105.0 minutes, nearly four times longer than 5,000 steps. This difference in runtime is further recognized in the EMC Highway drive cycle with the 5,000 and 10,000 step runtimes of 102 minutes (1.7 hours) and 436 minutes (7.3 hours), respectively. Examining the difference in fuel used results, the fuel used for the optimal policy determined for EMC City drive cycle with a step number of 5,000 and 10,000 steps is 286.3g and 284.6g, respectively, with a percent difference

of 0.55%. It is clear that for the considerable increase in runtime and memory requirements, the additional resolution of the optimal control policy is not necessary for the benchmarking activities presented in this work.

6.0 Conclusions and Recommendations

The objective of this work was to develop and execute a benchmark analysis for a hybrid 2019 Chevrolet Blazer using DP by backward induction. The end goal was to determine the global optimal control policy consisting of the TSR and transmission gear number for a variety of drive cycles to provide a frame of reference for the maximum possible performance of the powertrain. Throughout this work, several important takeaways and recommendations were discovered and are discussed in this section.

- The development of an appropriate backward facing model is critical to the efficient operation of a DP algorithm.
- The model must be of high enough fidelity to adequately captures the performance of a vehicle while being low enough fidelity to have extremely fast execution time.
- Keeping as many of the calculations as possible in matrix form gave the DP algorithm incredible speed compared to using loops.
- Memory requirements are a key design parameter for the DP algorithm.
- Increased state grid resolution results in increased memory requirements and runtime.
- There are diminishing returns on increasing state grid resolution.
- Modifying the constraints can be useful for verification as well as generating more benchmark data for the system.
- The cost function for the DP algorithm can include soft constraints.

- The cost function represents the overall goal of the DP algorithm optimal policy selection process.
- An appropriately selected cost function should mitigate the negative effects caused by leaking and uniform transition cost.
- The ability of the DP algorithm to select an appropriate set of control actions is highly dependent on the selection of a cost function.
- In the WVU EcoCAR Team competition 2019 Chevrolet Blazer, the electric motor is primarily used to provide propulsive torque as the primary method from zero speed and between 60-80% of torque requests at low speeds.
- For higher speeds, the ICE remains the primary torque producer even as speed fluctuates.
- The ICE should be pushed into FCO as much as possible in deceleration events and the electric motor should be used to make up the remaining braking torque as regenerative braking.
- Mechanical brake usage should be minimized as they essentially waste braking energy that should be captured by the electric motor.

7.0 Contributions and Future Work

The central contribution of this work is the development of a proof-of-concept energy management strategy analysis algorithm based on dynamic programming. The results from this work serve as the highest attainable performance of the powertrain in the WVU EcoCAR team competition vehicle which can be used to assess the online energy management algorithms being developed by the team. This work also lays the framework for a powertrain performance benchmarking tool that

can be used for a multitude of different hybrid vehicle architectures and components due to the versatility and format of parameterizing the model.

The work presented in this thesis can be expanded along several avenues. One of these avenues are the development of online rule-based algorithms from the optimal control policy determined in the benchmark analysis. This rule-based controller can be developed from the relationship between the torque split ratio, driver request, and vehicle speed by interpolating the results of the benchmark analysis. Another avenue of future work can be to include the shift schedule as part of the DP analysis such that the most efficient gear is selected as part of the optimal control policy. This would require modification of several parts of the DP algorithms framework, but the analysis could prove useful to understand how the use of the electric motor effects the relationship between the engine and transmission. Finally, the backwards looking model and DP algorithm may be modified via the cost function to determine the optimal control policy for other criteria other than fuel consumption. The optimization of energy use, ICE efficiency, and emissions all have the potential to change the optimal control strategy to develop additional frames of reference to be used as design criteria for an online control algorithm.

8.0 References

- [1] M. Ehsani, Y. Gao, and A. Emadi, *Modern Electric, Hybrid Electric, and Fuel Cell Vehicles*, 2nd ed. CRC Press, 2017.
- [2] S. Onori, L. Serrao, and G. Rizzoni, *Hybrid Electric Vehicles*, 1st ed. Springer London, 2016.
- [3] U.S. Department of Energy, “Autonomie, Hybrid Electric Vehicles.” https://www.autonomie.net/references/hev_26.html (accessed Aug. 19, 2021).
- [4] Argonne National Laboratory, U.S Department of Energy, “AVTC History.” <https://avtcservices.org/avtc-history/> (accessed Jul. 16, 2021).
- [5] West Virginia University EcoCAR Team, “Architecture Selection Report,” 2019.
- [6] L. Guzzella and A. Sciarretta, *Vehicle Propulsion Systems*, vol. 91, no. 5. Springer International Publishing, 2012.
- [7] D. George, “Hybrid Electric Vehicle Torque Split Algorithm for Reduction of Engine Torque Transients,” West Virginia University Libraries, 2018.
- [8] N. Connelly, “Generation and Sensitivity Analysis of Transmission Shift Schedule for Hybrid-Electric Vehicle,” West Virginia University Libraries, 2018.
- [9] T. P. Harris, “Implementation of Radial Basis Function Artificial Neural Network into an Adaptive Equivalent Consumption Minimization Strategy for Optimized Control of a Hybrid Electric Vehicle,” West Virginia University Libraries, 2020.
- [10] S. Kaban, Z. Dong, and C. Crawford, “Performance Modeling and Benchmark Analysis of

- an Advanced 4WD Series-Parallel PHEV Using Dynamic Programming,” University of Victoria, 2010.
- [11] P. Bowles, H. Peng, and X. Zhang, “Energy management in a parallel hybrid electric vehicle with a continuously variable transmission,” in *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, 2000, vol. 1, no. June, pp. 55–59 vol.1, doi: 10.1109/ACC.2000.878771.
- [12] K. L. Butler, M. Ehsani, and P. Kamath, “A Matlab-based modeling and simulation package for electric and hybrid electric vehicle design,” *IEEE Trans. Veh. Technol.*, vol. 48, no. 6, pp. 1770–1778, 1999, doi: 10.1109/25.806769.
- [13] P. Pisu and G. Rizzoni, “A Comparative Study Of Supervisory Control Strategies for Hybrid Electric Vehicles,” *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 506–518, May 2007, doi: 10.1109/TCST.2007.894649.
- [14] H. Banvait, S. Anwar, and Y. Chen, “A rule-based energy management strategy for Plug-in Hybrid Electric Vehicle (PHEV),” in *2009 American Control Conference*, 2009, pp. 3938–3943, doi: 10.1109/ACC.2009.5160242.
- [15] A. M. Phillips, M. Jankovic, and K. E. Bailey, “Vehicle system controller design for a hybrid electric vehicle,” in *Proceedings of the 2000. IEEE International Conference on Control Applications. Conference Proceedings (Cat. No.00CH37162)*, 2000, vol. 1, pp. 297–302, doi: 10.1109/CCA.2000.897440.
- [16] D. Driankov, H. Hellendoorn, and M. Reinfrank, *An Introduction to Fuzzy Control*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993.

- [17] M. H. Hajimiri and F. R. Salmasi, "A Fuzzy Energy Management Strategy for Series Hybrid Electric Vehicle with Predictive Control and Durability Extension of the Battery," in *2006 IEEE Conference on Electric and Hybrid Vehicles*, Dec. 2006, pp. 1–5, doi: 10.1109/ICEHV.2006.352279.
- [18] B. Xu, M. Li, S. Yang, B. Guo, and H. Cui, "Design and Simulation of Fuzzy Control Strategy for Parallel Hybrid Electric Vehicle," in *2010 International Conference on Intelligent System Design and Engineering Application*, Oct. 2010, vol. 1, pp. 539–543, doi: 10.1109/ISDEA.2010.287.
- [19] L. Majdi, A. Ghaffari, and N. Fatehi, "Control strategy in hybrid electric vehicle using fuzzy logic controller," in *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2009, pp. 842–847, doi: 10.1109/ROBIO.2009.5420563.
- [20] Q. Zeng and J. Huang, "The design and simulation of fuzzy logic controller for parallel hybrid electric vehicles," *Proc. IEEE Int. Conf. Autom. Logist. ICAL 2007*, pp. 908–912, 2007, doi: 10.1109/ICAL.2007.4338695.
- [21] Shijing Xu, "Investigation of EMS based on fuzzy logic controller for an ICE/battery/UC hybrid electric vehicle," in *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*, Aug. 2011, pp. 4041–4044, doi: 10.1109/AIMSEC.2011.6010016.
- [22] A. Sciarretta, M. Back, and L. Guzzella, "Optimal Control of Parallel Hybrid Electric Vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 12, no. 3, pp. 352–363, May 2004, doi: 10.1109/TCST.2004.824312.
- [23] H. Kazemi, Y. P. Fallah, A. Nix, and S. Wayne, "Predictive AECMS by Utilization of

- Intelligent Transportation Systems for Hybrid Electric Vehicle Powertrain Control,” *IEEE Trans. Intell. Veh.*, vol. 2, no. 2, pp. 75–84, Jun. 2017, doi: 10.1109/TIV.2017.2716839.
- [24] Chan-Chiao Lin, Huei Peng, J. W. Grizzle, and Jun-Mo Kang, “Power management strategy for a parallel hybrid electric truck,” *IEEE Trans. Control Syst. Technol.*, vol. 11, no. 6, pp. 839–849, Nov. 2003, doi: 10.1109/TCST.2003.815606.
- [25] R. Wang and S. M. Lukic, “Dynamic programming technique in hybrid electric vehicle optimization,” in *2012 IEEE International Electric Vehicle Conference*, Mar. 2012, pp. 1–8, doi: 10.1109/IEVC.2012.6183284.
- [26] H. I. Dokuyucu and M. Cakmakci, “Concurrent design of energy management and vehicle stability algorithms for a parallel hybrid vehicle using Dynamic Programming,” in *2012 American Control Conference (ACC)*, Jun. 2012, pp. 535–540, doi: 10.1109/ACC.2012.6315397.
- [27] R. M. Patil, Z. Filipi, and H. K. Fathy, “Comparison of Supervisory Control Strategies for Series Plug-In Hybrid Electric Vehicle Powertrains Through Dynamic Programming,” *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 2, pp. 502–509, Mar. 2014, doi: 10.1109/TCST.2013.2257778.
- [28] M. Sorrentino, F. Mauramati, I. Arsie, A. Cricchio, C. Pianese, and W. Nesci, “Application of Willans Line Method for Internal Combustion Engines Scalability towards the Design and Optimization of Eco-Innovation Solutions,” in *SAE Technical Papers*, Sep. 2015, vol. 2015-Septe, no. September, doi: 10.4271/2015-24-2397.
- [29] The MathWorks Inc, “Simulink, Simulation and Model-Based Design.” <https://www.mathworks.com/products/simulink.html> (accessed Jul. 29, 2021).

- [30] The MathWorks Inc, “Simulink, Powertrain Blockset.”
<https://www.mathworks.com/products/powertrain.html> (accessed Jul. 29, 2021).
- [31] dSPACE GmbH, “ControlDesk.”
<https://www.dspace.com/en/inc/home/products/sw/experimentandvisualization/controldesk.cfm> (accessed Jul. 29, 2021).
- [32] The MathWorks Inc, “Powertrain Blockset Toolbox, Vehicle Body 1DOF Longitudinal.”
<https://www.mathworks.com/help/autoblks/ref/vehiclebody1doflongitudinal.html>
(accessed Aug. 03, 2021).
- [33] The MathWorks Inc, “Powertrain Blockset Toolbox, Mapped SI Engine.”
<https://www.mathworks.com/help/vdynblks/ref/mappedsiengine.html> (accessed Aug. 03, 2021).
- [34] The MathWorks Inc, “Powertrain Blockset Toolbox, SI Controller.”
https://www.mathworks.com/help/autoblks/ref/sicontroller.html?searchHighlight=SIcontroller&s_tid=srchtitle (accessed Aug. 03, 2021).
- [35] The MathWorks Inc, “Powertrain Blockset Toolbox, Datasheet Battery.”
https://www.mathworks.com/help/autoblks/ref/datasheetbattery.html?searchHighlight=datasheetbattery&s_tid=srchtitle (accessed Aug. 04, 2021).
- [36] The MathWorks Inc, “Powertrain Blockset Toolbox, Mapped Motor.”
https://www.mathworks.com/help/autoblks/ref/mappedmotor.html?searchHighlight=Mappedmotor&s_tid=srchtitle (accessed Aug. 04, 2021).
- [37] The MathWorks Inc, “Powertrain Blockset Toolbox, Torque Converter.”

- [https://www.mathworks.com/help/autoblks/ref/torqueconverter.html?searchHighlight=torque converter&s_tid=srchtitle](https://www.mathworks.com/help/autoblks/ref/torqueconverter.html?searchHighlight=torque%20converter&s_tid=srchtitle) (accessed Aug. 04, 2021).
- [38] The MathWorks Inc, “Powertrain Blockset Toolbox, Open Differential.” [https://www.mathworks.com/help/vdynblks/ref/opendifferential.html?searchHighlight=open differential&s_tid=srchtitle](https://www.mathworks.com/help/vdynblks/ref/opendifferential.html?searchHighlight=open%20differential&s_tid=srchtitle) (accessed Aug. 04, 2021).
- [39] The MathWorks Inc, “Powertrain Blockset Toolbox, Longitudinal wheel with disc, drum, or mapped brake - Simulink.” [https://www.mathworks.com/help/vdynblks/ref/longitudinalwheel.html?searchHighlight=Longitudinal Wheel&s_tid=srchtitle](https://www.mathworks.com/help/vdynblks/ref/longitudinalwheel.html?searchHighlight=Longitudinal%20Wheel&s_tid=srchtitle) (accessed Aug. 04, 2021).
- [40] The MathWorks Inc, “Powertrain Blockset Toolbox, Longitudinal speed-tracking controller.” [https://www.mathworks.com/help/vdynblks/ref/longitudinaldriver.html?searchHighlight=Longitudinal Driver&s_tid=srchtitle](https://www.mathworks.com/help/vdynblks/ref/longitudinaldriver.html?searchHighlight=Longitudinal%20Driver&s_tid=srchtitle) (accessed Aug. 04, 2021).
- [41] D. A. Ward, “Development of a powertrain control algorithm for a compound-split diesel hybrid-electric vehicle,” West Virginia University Libraries, 2012.
- [42] R. Bellman and E. Lee, “History and development of dynamic programming,” *IEEE Control Syst. Mag.*, vol. 4, no. 4, pp. 24–28, Nov. 1984, doi: 10.1109/MCS.1984.1104824.
- [43] D. K. Smith and D. P. Bertsekas, “Dynamic Programming and Optimal Control. Volume 1,” *J. Oper. Res. Soc.*, vol. 47, no. 6, p. 833, Jun. 1996, doi: 10.2307/3010291.
- [44] R. E. Bellman and S. E. Dreyfus, *Applied Dynamic Programming*, vol. 26, no. 3. Princeton University Press, 1962.

- [45] J. Pu, C. Yin, and J. Zhang, “Energy management strategy for parallel hybrid electric vehicles,” *Chinese J. Mech. Eng. (English Ed.)*, vol. 18, no. 2, pp. 215–219, 2005, doi: 10.3901/cjme.2005.02.215.
- [46] O. Sundström, D. Ambühl, and L. Guzzella, “On Implementation of Dynamic Programming for Optimal Control Problems with Final State Constraints,” *Oil Gas Sci. Technol. – Rev. l’Institut Français du Pétrole*, vol. 65, no. 1, pp. 91–102, 2010, doi: 10.2516/ogst/2009020.
- [47] General Motors, “2.5L LCV - Powertrain OEM Sales,” 2016.
- [48] U.S. Environmental Protection Agency, “Dynamometer Drive Schedules.” <https://www.epa.gov/vehicle-and-fuel-emissions-testing/dynamometer-drive-schedules> (accessed Sep. 14, 2021).

9.0 Appendix

9.1 DP algorithm and Backward-Looking Model MATLAB Code:

```
% Load Component Data
run('ComponentData.m') %File that contains all of the component data that I
should need

% Load Drive Cycle Data
SelectedCycle =input('Select Drive Cycle\n 1 = EMC City\n 2 = EMC Highway\n 3
= US06\n 4 = UDDS\n 5 = HWFET\n 6 = FU505\n 9 = Verification\n\n');

%Script that resamples the selected drive cycle and provides necessary drive
cycle information
run('DriveCycleDiscretize.m')

% Simulation Parameters
T = length(Trq_Whl); %number of stages

%Start Timer to keep track of runtime and provide estimated time remaining
tstart = tic;

% --- Discretize the State Space Grid --- %
% Discretize State Space Grid
SOCmin =20; %Minimum SOC in State Space [%]
SOCmax =80; %Maximum SOC in State Space [%]
SOC_Target = 50; %Target SOC [%]
Steps = 5000; %Number of steps in state space

%Assemble the SOC vector
SOCa = [SOCmin:(SOCmax-SOCmin)/Steps:SOC_Target-((SOCmax-SOCmin)/Steps)];
%Minimum SOC to Target SOC [%]
SOCb = [SOC_Target+((SOCmax-SOCmin)/Steps):(SOCmax-SOCmin)/Steps:SOCmax];
%Target SOC to Maximum SOC [%]
SOC = [SOCa; SOC_Target; SOCb]; %SOC grid of state space [%]

%Build initial SOC vector that only includes the transition from the target
SOC
targ_SOC_pos = find(SOC==SOC_Target); %Position of target SOC in SOC vector
[%]
SOC_init = zeros(size(SOC)); %Layout zero matrix with the appropriate size
[%]
SOC_init(SOC == SOC_Target) = SOC_Target; %Place the target SOC in the
appropriate position [%]
SOC_init(SOC_init==0) = NaN; %Convert remining zeros to NaN

%Simulation Setup Information
deltaSOC = (SOCmax-SOCmin)/Steps; %SOC step size for the given state grid [%]
deltaPwr = deltaSOC.*(BattEnergyCapacity*3600*0.01); %Battery Power step size
for the given state grid [W]
AdmissibleSOCTransition = 0.93; %Maximum transition size for SOC(i) to SOC(j)
[%]
```

```

% --- Determine Shift Schedule --- %
% Initialize Gear
gear = 1; %Start in first gear [N_gear]
for x=2:T
% Determine the Gear we should be in based on the Pedal Map
% Calculate Pedal Position
AccPdlPos =
interp2(VehSpd_Bpts,WhlTrq_Bpts,AccelPdl_table,VehSpd(x)*3.6,Trq_Whl(x))*0.01;
%Accelerator Pedal Position [dec]
AccPdlPos(isnan(AccPdlPos))=0; %Find NaN locations from above interpolation
and set them to zero
AccPdl(x) = AccPdlPos;

%Calculate Upper and Lower Speed Thresholds
UpperSpdThreshold = interp2(Gear_Up_bpt,Pdl_Pos_Up_bpt,Up_Shft_Spd,gear(x-
1),AccPdlPos); %Upper Spd Thresh [m/s]
LowerSpdThreshold = interp2(Gear_Dn_bpt,Pdl_Pos_Dn_bpt,Dn_Shft_Spd,gear(x-
1),AccPdlPos); %Lower Spd Thresh [m/s]

%Interpolate the selected gear based on the speed thresholds.
gear(x) = gear(x-1) + (1.*(UpperSpdThreshold<VehSpd(x))) + (-
1.*(LowerSpdThreshold>VehSpd(x))); %Selected gear [N_gear]

%Check to determine if additional shifts are required.
check = 1; %Initialize the check criterai
while check == 1
gearx = gear(x); %Previously determined selected gear

%Use previously determined gear to redefine the up and downshift thresholds
UpperSpdThresholdCheck =
interp2(Gear_Up_bpt,Pdl_Pos_Up_bpt,Up_Shft_Spd,gearx,AccPdlPos); %Upper Spd
Thresh [m/s]
LowerSpdThresholdCheck =
interp2(Gear_Dn_bpt,Pdl_Pos_Dn_bpt,Dn_Shft_Spd,gearx,AccPdlPos); %Lower Spd
Thresh [m/s]

%Determine the gear based on the up and downshift thresholds
gear(x) = gearx + (1.*(UpperSpdThresholdCheck<VehSpd(x))) + (-
1.*(LowerSpdThresholdCheck>VehSpd(x)));
%Check if the newly determined gear is the same as previously determined gear
if gearx==gear(x)
check = 0; %True condition changes the value of check
else %Otherwise additional iterations are needed
end
end
end

% --- Preallocate Space --- %
TransTo = zeros(size(SOC,1),T-2);
Loc = zeros(size(SOC,1),T-2);

%% ----- Start of Algorithm -----
----- %

```

```

% Start by assessing the transition to the final stage
k = T-1;

%Display to user the current stage
fprintf('The Current stage is k = %.2f\n',k)

% --- Electric Powertrain Calculations --- %
Pwr_Mot = (SOC_init'-SOC).*(BattEnergyCapacity*3600*0.01); %Motor Power [W]
Pwr_Mot(abs(Pwr_Mot)>=50000) = NaN; %Apply the Motor/Battery Power
Constraints
Spd_Mot = Spd_Whl(k)*Ndiff_P4*(pi/30).*ones(size(Pwr_Mot)); %Motor Speed
[rad/s]
% Trq_Mot =
[interp2(ElecPwr_bpts,P4Mtr_spd_bpts,P4_Trq_Pwr,Pwr_Mot,Spd_Mot)']; %Motor
Component Torque [Nm]
Trq_Mot = interp2(PwrBpts,P4Mtr_spd_bpts,PwrTrq_tbl,0.95*Pwr_Mot,Spd_Mot);
%Motor Component Torque [Nm]
Trq_Mot_Whl = Trq_Mot.*Ndiff_P4; %Motor Wheel Torque [Nm]

% Remaining Torque to be supplied by either the ICE or the friction brakes
Trq_Rem = Trq_Whl(k)-Trq_Mot_Whl; %Remaining Wheel Torque [Nm]

% --- Conventional Powertrain Calculations --- %
%Transmission Efficiency Lookup Table
TransEff =
reshape(interp1(transeff_Spdbpts,transeff_table',Spd_Whl(k)),[1,1,9]);

% Calculate Torque Converter Turbine Speed
Spd_turb =
Spd_Whl(k).*ones(size(Trq_Rem_ICE)).*TransGearRatio(gear(k)).*Ndiff; %Turbine
Speed - Propogated back through wheel speed [RPM]

Stat_Lock = (Spd_turb>=Spd_ICE_Idle).*(Spd_turb<=Spd_ICE_limit); %Torque
converter lock status (1 - locked, 0 - unlocked)
Stat_Clutch = (Trq_Whl(k)~=0); %Torque Converter clutch status (1 - closed, 0
- open)

% Calculate the speed of the ICE with following piecewise equation
Spd_ICE = (Spd_turb.*(Stat_Lock==1))+...

(Spd_ICE_Idle.*ones(size(Trq_Rem_ICE)).*(Stat_Lock==0)).*(VehSpd(k)<5)+...
(Spd_ICE_limit.*ones(size(Trq_Rem_ICE)).*(Stat_Lock==0)).*(VehSpd(k)>=5));
%ICE Speed [RPM]

%Additional Torque Converter Parameters
SR = Spd_turb./Spd_ICE; %Speed ratio of TC
TR = interp1(SpdRatio_bpts,TrqRatio,SR); %Torque ratio of TC

%Calculate the Engine Operational Constraints
Trq_ICE_max = interp1(ICE_spd_bpts,max_trq_line,Spd_ICE); %Maximum ICE
component torque [Nm]
Trq_ICE_minrun = interp1(ICE_spd_bpts,min_trq_line,Spd_ICE); %Minimum running
ICE component torque [Nm]
Trq_ICE_fco = interp1(f_tbrake_n_bpt,min_trq_fco,Spd_ICE); %Fuel cut off ICE
component torque [Nm]

```

```

Trq_Rem_ICE =
    (Trq_Rem)./TransGearRatio(gear(k))./Ndiff./TransEff(gear(k))./TR; %Remaining
torque in terms of ICE component torque [Nm]
Trq_Rem_ICE(Trq_Rem_ICE>Trq_ICE_max) = NaN; %Impose maximum ICE torque
production constraint

% ICE torque calculation
Trq_ICE = Trq_Rem_ICE.*(Trq_Rem>0).*(Trq_Whl(k)>0)+...
Trq_ICE_minrun.*(Trq_Rem<=0).*(Trq_Whl(k)>0)+...
Trq_ICE_minrun.*(Trq_Rem>0).*(Trq_Whl(k)<=0)+...
Trq_ICE_minrun.*(Trq_Rem<=0).*(Trq_Whl(k)<=0).*(Trq_Rem_ICE>Trq_ICE_fco).*(Stat
at_Lock==1)+...
Trq_ICE_minrun.*(Trq_Rem<=0).*(Trq_Whl(k)<=0).*(Trq_Rem_ICE<=Trq_ICE_fco).*(Stat
at_Lock==0)+...
Trq_ICE_fco.*(Trq_Rem<=0).*(Trq_Whl(k)<=0).*(Trq_Rem_ICE<=Trq_ICE_fco).*(Stat
at_Lock==1); %ICE component torque [Nm]

Trq_ICE_Whl =
    (Trq_ICE).*TransGearRatio(gear(k)).*Ndiff.*TransEff(gear(k)).*TR.*(Stat_Clutch
== 1); % ICE wheel torque [Nm]

%Torque requirement not met and negative mechanical braking torque only
constraints
Trq_brk = Trq_Whl(k)-(Trq_Mot_Whl)-(Trq_ICE_Whl); %Calculate mechanical brake
wheel torque [Nm]
Trq_ICE(Trq_brk>0.0001)=NaN; %Impose constraint

% --- Special Case --- %
% For situations where the produced torque by the ICE is not sufficient, the
selected gear is modified to achieve the desired torque
if sum(~isnan(Trq_ICE), 'all')==0 %Checks if ICE torque is insufficient
lowestgearreached=0; %Initialize logical for determining if already at lowest
gear
while sum(~isnan(Trq_ICE), 'all')==0 %Start conditional loop to identify
admissible ICE torque
% Check if lowest gear has been reached, determines direction of gear shift
(0 - downshift, 1 - upshift)
if gear(k)==1
lowestgearreached = 1; %Lowest gear has been reached
elseif gear(k) == 9
lowestgearreached = 0; %Lowest gear has not been reached
end

% Change gear
if lowestgearreached == 0
gear(k) = gear(k)-1;
else
gear(k) = gear(k)+1;
end

% Recalculate the conventional powertrain operating parameters
% Calculate Torque Converter Turbine Speed

```

```

Spd_turb =
Spd_Whl(k).*ones(size(Trq_Rem_ICE)).*TransGearRatio(gear(k)).*Ndiff; %Turbine
Speed - Propogated back through wheel speed [RPM]

Stat_Lock = (Spd_turb>=Spd_ICE_Idle).*(Spd_turb<=Spd_ICE_limit); %Torque
converter lock status (1 - locked, 0 - unlocked)
Stat_Clutch = (Trq_Whl(k)~=0); %Torque Converter clutch status (1 - closed, 0
- open)

% Calculate the speed of the ICE with following piecewise equation
Spd_ICE = (Spd_turb.*(Stat_Lock==1))+...

(Spd_ICE_Idle.*ones(size(Trq_Rem_ICE)).*(Stat_Lock==0)).*(VehSpd(k)<5)+...
(Spd_ICE_limit.*ones(size(Trq_Rem_ICE)).*(Stat_Lock==0)).*(VehSpd(k)>=5));
%ICE Speed [RPM]

%Additional Torque Converter Parameters
SR = Spd_turb./Spd_ICE; %Speed ratio of TC
TR = interp1(SpdRatio_bpts,TrqRatio,SR); %Torque tatio of TC

%Calculate the Engine Operational Constraints
Trq_ICE_max = interp1(ICE_spd_bpts,max_trq_line,Spd_ICE); %Maximum ICE
component torque [Nm]
Trq_ICE_minrun = interp1(ICE_spd_bpts,min_trq_line,Spd_ICE); %Minimum running
ICE component torque [Nm]
Trq_ICE_fco = interp1(f_tbrake_n_bpt,min_trq_fco,Spd_ICE); %Fuel cut off ICE
component torque [Nm]

Trq_Rem_ICE =
(Trq_Rem)./TransGearRatio(gear(k))./Ndiff./TransEff(gear(k))./TR; %Remaining
torque in terms of ICE component torque [Nm]
Trq_Rem_ICE(Trq_Rem_ICE>Trq_ICE_max) = NaN; %Impose maximum ICE torque
production constraint

% ICE torque calculation
Trq_ICE = Trq_Rem_ICE.*(Trq_Rem>0).*(Trq_Whl(k)>0)+...
Trq_ICE_minrun.*(Trq_Rem<=0).*(Trq_Whl(k)>0)+...
Trq_ICE_minrun.*(Trq_Rem>0).*(Trq_Whl(k)<=0)+...
Trq_ICE_minrun.*(Trq_Rem<=0).*(Trq_Whl(k)<=0).*(Trq_Rem_ICE>Trq_ICE_fco).*(St
at_Lock==1)+...
Trq_ICE_minrun.*(Trq_Rem<=0).*(Trq_Whl(k)<=0).*(Trq_Rem_ICE<=Trq_ICE_fco).*(S
tat_Lock==0)+...
Trq_ICE_fco.*(Trq_Rem<=0).*(Trq_Whl(k)<=0).*(Trq_Rem_ICE<=Trq_ICE_fco).*(Stat
_Lock==1); %ICE component torque [Nm]

Trq_ICE_Whl =
(Trq_ICE).*TransGearRatio(gear(k)).*Ndiff.*TransEff(gear(k)).*TR.*(Stat_Clutc
h == 1); % ICE wheel torque [Nm]

%Torque requirement not met and negative mechanical braking torque only
constraints
Trq_brk = Trq_Whl(k)-(Trq_Mot_Whl)-(Trq_ICE_Whl); %Cacluate mechanical brake
wheel torque [Nm]
Trq_ICE(Trq_brk>0.0001)=NaN; %Impose constraint
end

```

```

end

%Calculate fuel flow
EngFuelFlw = interp2(ICE_spd_bpts,ICE_trq_bpts,fuel_table,Spd_ICE,Trq_ICE);
%This is my map that I created

% --- Cost Function Calculations --- %
%cost-to-go-matrix with additional torque overproduction term
g_gear =
EngFuelFlw+(0.008.*(abs(Trq_Mot_Whl)+abs(Trq_ICE_Whl)+abs(Trq_brk)));%

%Remove arbitrary gear selected (FUNCTIONALITY NOT USED)
g = zeros(size(SOC,1));
[g,gloc] = min(g_gear,[],3);
g(g==0) = nan;

% visualization of operating parameters, selects the associated values from
the gear minimization (FUNCTIONALITY NOT USED)
% Linear indexes of selected variables (FUNCTIONALITY NOT USED)
ind = reshape(1:numel(gloc),size(gloc))+(numel(gloc).*(gloc-1));

% Operating parameters of the selected gear (FUNCTIONALITY NOT USED)
Selected_Trq_ICE = Trq_ICE(ind);
Selected_Trq_brk = Trq_brk(ind);
Selected_Trq_Rem_ICE = Trq_Rem_ICE(ind);
Selected_EngFuelFlw = EngFuelFlw(ind);
Selected_Spd_ICE = Spd_ICE(ind);

%Calculate the total cost of the path, at the first stage (k=N, the total
path cost is equal to the cost-to-go matrix)
[J(:,k),Jloc] = min(g,[],2,'linear');

%Visualization of operating parameters for transition from k=N-1 to k=N
vis_Trq_Mot(:,k) = Trq_Mot(Jloc); %Electric motor component torque [Nm]
vis_Trq_ICE(:,k) = Selected_Trq_ICE(Jloc); %ICE component torque [Nm]
vis_Trq_brk(:,k) = Selected_Trq_brk(Jloc); %Brake wheel torque [Nm]
vis_Trq_Rem_ICE(:,k) = Selected_Trq_Rem_ICE(Jloc); % Remaining ICE component
torque [Nm]
vis_selectedgear(:,k) = gloc(Jloc); % Selected gear
vis_Trq_Rem(:,k) = Trq_Rem(Jloc); % Remaining wheel torque [Nm]
vis_EngFuelFlw(:,k) = Selected_EngFuelFlw(Jloc); %ICE fuel flow [g/s]
vis_Spd_ICE(:,k) = Selected_Spd_ICE(Jloc); % ICE speed [RPM]
vis_Stat_Clutch(:,k) = double(Stat_Lock(Jloc)); %Clutch lock status

%Timer and estimated time remaining calculations
tt = [];
tk = toc(tstart);
tt = [tt tk];
timerem = mean(tt)*(k);
tt = [];

%% ----- Continue Algorithm for T-2,T-3...1 -----
----- %
for stage=2:T-1

```

```

%Start Timer to keep track of runtime and provide estimated time remaining
tstart = tic;

%Current Stage
k=T-stage;

%Display current stage and estimated time remaining to user
fprintf('The Current stage is k = %.2f\n Estimated time remaining %.0f
minutes\n\n\n\n',k,timerem/60)

%Calculate the transition for every i to each j
% --- Electric powertrain calculations --- %;
Pwr_Mot = -(SOC'-SOC).*(BattEnergyCapacity*3600*0.01); %Motor Power [W]
Pwr_Mot(abs(Pwr_Mot)>=50000) = NaN; %Apply the Motor/Battery Power
Constraints
Spd_Mot = Spd_Whl(k)*Ndiff_P4*(pi/30).*ones(size(Pwr_Mot)); %Motor Speed
[rad/s]
% Trq_Mot =
[interp2(ElecPwr_bpts,P4Mtr_spd_bpts,P4_Trq_Pwr,Pwr_Mot,Spd_Mot)']; %Motor
Component Torque [Nm]
Trq_Mot = interp2(PwrBpts,P4Mtr_spd_bpts,PwrTrq_tbl,0.95*Pwr_Mot,Spd_Mot);
%Motor Component Torque [Nm]
Trq_Mot_Whl = Trq_Mot.*Ndiff_P4; %Motor Wheel Torque [Nm]

% Remaining Torque to be supplied by either the ICE or the friction brakes
Trq_Rem = Trq_Whl(k)-Trq_Mot_Whl; %Remaining Wheel Torque [Nm]

% --- Conventional powertrain calculations --- %
%Transmission Efficiency Lookup Table
TransEff =
reshape(interp1(transeff_Spdbpts,transeff_table',Spd_Whl(k)),[1,1,9]);

% Calculate Torque Converter Turbine Speed
Spd_turb =
Spd_Whl(k).*ones(size(Trq_Rem_ICE)).*TransGearRatio(gear(k)).*Ndiff; %Turbine
Speed - Propogated back through wheel speed [RPM]

Stat_Lock = (Spd_turb>=Spd_ICE_Idle).*(Spd_turb<=Spd_ICE_limit); %Torque
converter lock status (1 - locked, 0 - unlocked)
Stat_Clutch = (Trq_Whl(k)~=0); %Torque Converter clutch status (1 - closed, 0
- open)

% Calculate the speed of the ICE with following piecewise equation
Spd_ICE = (Spd_turb.*(Stat_Lock==1))+...

(Spd_ICE_Idle.*ones(size(Trq_Rem_ICE)).*(Stat_Lock==0)).*(VehSpd(k)<5)+...
(Spd_ICE_limit.*ones(size(Trq_Rem_ICE)).*(Stat_Lock==0)).*(VehSpd(k)>=5));
%ICE Speed [RPM]

%Additional Torque Converter Parameters
SR = Spd_turb./Spd_ICE; %Speed ratio of TC
TR = interp1(SpdRatio_bpts,TrqRatio,SR); %Torque ratio of TC

%Calculate the Engine Operational Constraints

```



```

Trq_ICE_max = interp1(ICE_spd_bpts,max_trq_line,Spd_ICE); %Maximum ICE
component torque [Nm]
Trq_ICE_minrun = interp1(ICE_spd_bpts,min_trq_line,Spd_ICE); %Minimum running
ICE component torque [Nm]
Trq_ICE_fco = interp1(f_tbrake_n_bpt,min_trq_fco,Spd_ICE); %Fuel cut off ICE
component torque [Nm]

Trq_Rem_ICE =
    (Trq_Rem)./TransGearRatio(gear(k))./Ndiff./TransEff(gear(k))./TR; %Remaining
torque in terms of ICE component torque [Nm]
Trq_Rem_ICE(Trq_Rem_ICE>Trq_ICE_max) = NaN; %Impose maximum ICE torque
production constraint

% ICE torque calculation
Trq_ICE = Trq_Rem_ICE.*(Trq_Rem>0).*(Trq_Whl(k)>0)+...
Trq_ICE_minrun.*(Trq_Rem<=0).*(Trq_Whl(k)>0)+...
Trq_ICE_minrun.*(Trq_Rem>0).*(Trq_Whl(k)<=0)+...
Trq_ICE_minrun.*(Trq_Rem<=0).*(Trq_Whl(k)<=0).*(Trq_Rem_ICE>Trq_ICE_fco).*(Stat
at_Lock==1)+...
Trq_ICE_minrun.*(Trq_Rem<=0).*(Trq_Whl(k)<=0).*(Trq_Rem_ICE<=Trq_ICE_fco).*(Stat
at_Lock==0)+...
Trq_ICE_fco.*(Trq_Rem<=0).*(Trq_Whl(k)<=0).*(Trq_Rem_ICE<=Trq_ICE_fco).*(Stat
_Lock==1); %ICE component torque [Nm]

Trq_ICE_Whl =
    (Trq_ICE).*TransGearRatio(gear(k)).*Ndiff.*TransEff(gear(k)).*TR.*(Stat_Clutc
h == 1); % ICE wheel torque [Nm]

%Torque requirement not met and negative mechanical braking torque only
constraints
Trq_brk = Trq_Whl(k)-(Trq_Mot_Whl)-(Trq_ICE_Whl); %Cacluate mechanical brake
wheel torque [Nm]
Trq_ICE(Trq_brk>0.0001)=NaN; %Impose constraint

% --- Special Case --- %
% For situations where the produced torque by the ICE is not sufficient, the
selected gear is modified to achieve the desired torque
if sum(~isnan(Trq_ICE),'all')==0 %Checks if ICE torque is insufficient
lowestgearreached=0; %Initialize logical for determining if already at lowest
gear
while sum(~isnan(Trq_ICE),'all')==0 %Start conditional loop to identify
admissible ICE torque
% Check if lowest gear has been reached, determines direction of gear shift
(0 - downshift, 1 - upshift)
if gear(k)==1
lowestgearreached = 1; %Lowest gear has been reached
elseif gear(k) == 9
lowestgearreached = 0; %Lowest gear has not been reached
end

% Change gear
if lowestgearreached == 0
gear(k) = gear(k)-1;
else
gear(k) = gear(k)+1;
end

```

```

% Recalculate the conventional powertrain operating parameters
% Calculate Torque Converter Turbine Speed
Spd_turb =
Spd_Whl(k).*ones(size(Trq_Rem_ICE)).*TransGearRatio(gear(k)).*Ndifff; %Turbine
Speed - Propogated back through wheel speed [RPM]

Stat_Lock = (Spd_turb>=Spd_ICE_Idle).*(Spd_turb<=Spd_ICE_limit); %Torque
converter lock status (1 - locked, 0 - unlocked)
Stat_Clutch = (Trq_Whl(k)~=0); %Torque Converter clutch status (1 - closed, 0
- open)

% Calculate the speed of the ICE with following piecewise equation
Spd_ICE = (Spd_turb.*(Stat_Lock==1))+...

(Spd_ICE_Idle.*ones(size(Trq_Rem_ICE)).*(Stat_Lock==0)).*(VehSpd(k)<5)+...
(Spd_ICE_limit.*ones(size(Trq_Rem_ICE)).*(Stat_Lock==0)).*(VehSpd(k)>=5));
%ICE Speed [RPM]

%Additional Torque Converter Parameters
SR = Spd_turb./Spd_ICE; %Speed ratio of TC
TR = interp1(SpdRatio_bpts,TrqRatio,SR); %Torque tatio of TC

%Calculate the Engine Operational Constraints
Trq_ICE_max = interp1(ICE_spd_bpts,max_trq_line,Spd_ICE); %Maximum ICE
component torque [Nm]
Trq_ICE_minrun = interp1(ICE_spd_bpts,min_trq_line,Spd_ICE); %Minimum running
ICE component torque [Nm]
Trq_ICE_fco = interp1(f_tbrake_n_bpt,min_trq_fco,Spd_ICE); %Fuel cut off ICE
component torque [Nm]

Trq_Rem_ICE =
(Trq_Rem)./TransGearRatio(gear(k))./Ndifff./TransEff(gear(k))./TR; %Remaining
torque in terms of ICE component torque [Nm]
Trq_Rem_ICE(Trq_Rem_ICE>Trq_ICE_max) = NaN; %Impose maximum ICE torque
production constraint

% ICE torque calculation
Trq_ICE = Trq_Rem_ICE.*(Trq_Rem>0).*(Trq_Whl(k)>0)+...
Trq_ICE_minrun.*(Trq_Rem<=0).*(Trq_Whl(k)>0)+...
Trq_ICE_minrun.*(Trq_Rem>0).*(Trq_Whl(k)<=0)+...
Trq_ICE_minrun.*(Trq_Rem<=0).*(Trq_Whl(k)<=0).*(Trq_Rem_ICE>Trq_ICE_fco).*(S
tat_Lock==1)+...
Trq_ICE_minrun.*(Trq_Rem<=0).*(Trq_Whl(k)<=0).*(Trq_Rem_ICE<=Trq_ICE_fco).*(S
tat_Lock==0)+...
Trq_ICE_fco.*(Trq_Rem<=0).*(Trq_Whl(k)<=0).*(Trq_Rem_ICE<=Trq_ICE_fco).*(Stat
_Lock==1); %ICE component torque [Nm]

Trq_ICE_Whl =
(Trq_ICE).*TransGearRatio(gear(k)).*Ndifff.*TransEff(gear(k)).*TR.*(Stat_Clutc
h == 1); % ICE wheel torque [Nm]

%Torque requirement not met and negative mechanical braking torque only
constraints

```

```

Trq_brk = Trq_Whl(k)-(Trq_Mot_Whl)-(Trq_ICE_Whl); %Calculate mechanical brake
wheel torque [Nm]
Trq_ICE(Trq_brk>0.0001)=NaN; %Impose constraint
end
end

%Calculate fuel flow
EngFuelFlw = interp2(ICE_spd_bpts,ICE_trq_bpts,fuel_table,Spd_ICE,Trq_ICE);
%This is my map that I created

%Cost Function Calculations
%cost-to-go-matrix as a function of gear (FUNCTIONALITY NOT USED)
g_gear =
EngFuelFlw+(0.008.*(abs(Trq_Mot_Whl)+abs(Trq_ICE_Whl)+abs(Trq_brk)));;% <-
This one works

%Remove gear variable by finding minimum cost in terms of gear selection
(FUNCTIONALITY NOT USED)
g = zeros(size(SOC,1));
[g,gloc] = min(g_gear,[],3);
g(g==0) = nan;

% This is for the visualization of operating parameters, selects the
associated values from the gear minimization (FUNCTIONALITY NOT USED)
%Linear indexes of selected variables (FUNCTIONALITY NOT USED)
ind = reshape(1:numel(gloc),size(gloc))+(numel(gloc).*(gloc-1));

Spd_ICE = Spd_ICE.*(ones(size(Trq_ICE))); %ICE speed [RPM]
Stat_Lock = Stat_Lock.*(ones(size(Trq_ICE))); %Torque converter locking
status
Selected_Trq_ICE = Trq_ICE(ind); %ICE component torque [Nm]
Selected_Trq_brk = Trq_brk(ind); %Mechanical brake component torque [Nm]
Selected_Trq_Rem_ICE = Trq_Rem_ICE(ind); %Remaining ICE component torque [Nm]
Selected_EngFuelFlw = EngFuelFlw(ind); %Engine fuel flow [g/s]
Selected_Spd_ICE = Spd_ICE(ind); %ICE speed [RPM]

%Calculate total path cost & parameter visualization
for ii=1:size(SOC,1) % Assess each transition for every i
jj = find(abs((SOC(ii)-SOC))<=AdmissibleSOCTransition); %Determine the
corresponding admissible j values
totalg = g(ii,jj)'+J(jj,k+1); % Combined cost of current cost-to-go and total
path cost
[J(ii,k),Jloc] = min(g(ii,jj)'+J(jj,k+1),[],1); %Determine the minimized
combined cost for best transition
if isnan(J(ii,k)) == 1 %Determine if transition exists
%If transition does not exist, populate variable for NaN condtion
TransTo(ii,k) = nan;
vis_Trq_Mot(ii,k) = nan;
vis_Trq_ICE(ii,k) = nan;
vis_Trq_brk(ii,k) = nan;
vis_Trq_Rem_ICE(ii,k) = nan;
vis_Trq_Rem(ii,k) = nan;
vis_EngFuelFlw(ii,k) = nan;
vis_Spd_ICE(ii,k) = nan;
vis_Stat_Clutch(ii,k) = nan;
else

```

```

%If transition does exist, populate variables with corresponding transition
minloc = 1;
here = jj(1)+Jloc-1;
TransTo(ii,k) = here(minloc);
vis_Trq_Mot(ii,k) = Trq_Mot(ii,here(minloc));
vis_Trq_ICE(ii,k) = Selected_Trq_ICE(ii,here(minloc));
vis_Trq_brk(ii,k) = Selected_Trq_brk(ii,here(minloc));
vis_Trq_Rem_ICE(ii,k) = Selected_Trq_Rem_ICE(ii,here(minloc));
vis_Trq_Rem(ii,k) = Trq_Rem(ii,here(minloc));
vis_EngFuelFlw(ii,k) = Selected_EngFuelFlw(ii,here(minloc));
vis_Spd_ICE(ii,k) = Selected_Spd_ICE(ii,here(minloc));
vis_Stat_Clutch(ii,k) = Stat_Lock(ii,here(minloc));
end
end

%Timer and estimated time remaining
tk = toc(tstart);
tt = [tt tk];
timerem = mean(tt)*(k);
end

%% ----- Determine the Best Path and Calculate Operating Parameters -
----- %
BestPath = [find(SOC==50)]; %Select starting point to be the CS target SOC
%Follow the TransTo variable to determine the best path
for search = 2:T-2
BestPath(search) = TransTo(BestPath(search-1),search-1);
end
%The final entry in best path is the CS target
BestPath(T-1) = find(SOC==SOC_Target);
%Determine the SOC that corresponds to the best path
BestSOC = SOC(BestPath); %Best SOC [%]

%Calculate operating parameters following the best path
for time = 1:T-1
BestTrq_Mot(time) = vis_Trq_Mot(BestPath(time),time); %Optimal motor
component torque [Nm]
BestTrq_ICE(time) = vis_Trq_ICE(BestPath(time),time); %Optimal ICE component
torque [Nm]
BestTrq_brk(time) = vis_Trq_brk(BestPath(time),time); %Optimal brake wheel
torque [Nm]
BestGear(time) = gear(time); %Selected gear
BestTrq_Rem(time) = vis_Trq_Rem(BestPath(time),time); %Optimal remaining
wheel torque [Nm]
BestTrq_Rem_ICE(time) = vis_Trq_Rem_ICE(BestPath(time),time); %Optimal
remaining ICE component torque [Nm]
FuelFlow(time) = vis_EngFuelFlw(BestPath(time),time).*(264.17/743000);
%Optimal Fuel flow [gallon/s]
BestSpd_ICE(time) = vis_Spd_ICE(BestPath(time),time); %Optimal ICE speed
[RPM]
TransEff1 = interp1(transeff_Spdbpts,transeff_table',Spd_Whl(time));
%Transmission Efficiency as a function of optimal ICE parameters
BestTransEff(time) = TransEff1(BestGear(time)); %Optimal transmission
efficiency
BestTransRatio(time) = TransGearRatio(BestGear(time)); %Optimal transmission
ratio

```

end