

Graduate Theses, Dissertations, and Problem Reports

2021

# Implementation Of Fuzzy Logic Control Into An Equivalent Minimization Strategy For Adaptive Energy Management Of A Parallel Hybrid Electric Vehicle

Jared Alexander Diethorn West Virginia University, jadiethorn@mix.wvu.edu

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

Part of the Acoustics, Dynamics, and Controls Commons, Controls and Control Theory Commons, and the Navigation, Guidance, Control, and Dynamics Commons

#### **Recommended Citation**

Diethorn, Jared Alexander, "Implementation Of Fuzzy Logic Control Into An Equivalent Minimization Strategy For Adaptive Energy Management Of A Parallel Hybrid Electric Vehicle" (2021). *Graduate Theses, Dissertations, and Problem Reports.* 10204. https://researchrepository.wvu.edu/etd/10204

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.

# IMPLEMENTATION OF FUZZY LOGIC CONTROL INTO AN EQUIVALENT CONSUMPTION MINIMIZATION STRATEGY FOR ADAPTIVE ENERGY MANAGEMENT OF A PARALLEL HYBRID ELECTRIC VEHICLE

**Jared Diethorn** 

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: Equivalent Consumption Minimization Strategy, Brute Force Analysis, Fuzzy Logic Controller, Adaptive Control, Hybrid Electric Vehicle, Charge Sustainability

**Copyright 2021 Jared Diethorn** 

#### Abstract

## IMPLEMENTATION OF FUZZY LOGIC CONTROL INTO AN EQUIVALENT CONSUMPTION MINIMIZATION STRATEGY FOR ADAPTIVE ENERGY MANAGEMENT OF A PARALLEL HYBRID ELECTRIC VEHICLE

#### Jared Diethorn

As government agencies continue to tighten emissions regulations due to the continued increase in greenhouse gas production, automotive industries are seeking to produce increasingly efficient vehicle technology. Electric vehicles have been introduced by the industry, showing promising signs of reducing emissions production in the automotive sector. However, many consumers may be hesitant to purchase fully electric vehicles due to several uncertainty variables including available charging stations. Hybrid electric vehicles (HEVs) have been introduced to reduce problems while improving fuel economy. HEVs have led to the demand of creating more advanced controls software to consider multiple components for propulsive power in a vehicle. A large section in the software development process is the implementation of an optimal energy management strategy meant to improve the overall fuel efficiency of the vehicle. Optimal strategies can be implemented when driving conditions are known a prior. The Equivalent Consumption Minimization Strategy (ECMS) is an optimal control strategy that uses an equivalence factor to equate electrical to mechanical power when performing torque split determination between the internal combustion engine and electric motor for propulsive and regenerative torque. This equivalence factor is determined from offline vehicle simulations using a sensitivity analysis to provide optimal fuel economy results while maintaining predetermined high voltage battery state of charge (SOC) constraints. When the control hierarchy is modified or different driving styles are applied, the analysis must be redone to update the equivalence factor. The goal of this work is to implement a fuzzy logic controller that dynamically updates the equivalence factor to improve fuel economy, maintain a strict charge sustaining window of operation for the high voltage battery, and reduce computational time required during algorithm development. The adaptive algorithm is validated against global optimum fuel economy and charge sustaining results from a sensitivity analysis performed for multiple drive cycles. Results show a maximum fuel economy improvement of 9.82% when using a mild driving style and a 95% success rate when maintaining an ending SOC within 5% regardless of starting SOC. Recommendations for modification of the fuzzy logic controller are made to produce additional fuel economy and charge sustaining benefits from the parallel hybrid vehicle model.

#### Acknowledgements

I would like to thank my faculty advisor, committee chair, and head faculty advisor for the West Virginia University EcoCAR team, Dr. Andrew Nix, for his continued guidance and support throughout my academic career. Dr. Nix accepted me as a graduate research assistant in the EcoCAR program with little knowledge of my work ethic and only a recommendation from a fellow graduate student. He does not shy away from a task and devotes countless hours to both his students and the EcoCAR program to ensure everyone is performing at their best. I thank Dr. Scott Wayne for providing guidance in problems related to modeling complex dynamic vehicle systems. The hybrid vehicle course taught by Dr. Wayne expanded my knowledge of MATLAB/Simulink, specifically modeling a hybrid vehicle from scratch. Concepts learned in his class widened my knowledge of the automotive industry, which helped push me to fulfill a leadership role with the WVU EcoCAR team. I would also like to thank Dr. Mario Perhinschi for his continued effort in providing expert level knowledge in control systems and artificial intelligence. Knowledge learned in the AI class pertaining to fuzzy logic is what drove me to pursue the work presented in this paper. His knowledge in control systems and dedication to academia has helped me learn knowledge that I have applied in this work.

I would like to thank Dr. Brian Woerner for the role he played as the co-faculty advisor for the WVU EcoCAR team. During my time with the program, Dr. Woerner has actively been involved in troubleshooting issues experienced with sensors added in and on the vehicle. This contribution has helped the team progress tremendously since the summer leading into year 2 of the competition.

I would like to thank my fellow graduate assistants, both past and present in the WVU EcoCAR program. Benton Morris, the previous project manager, offered a variety of opinions when helping troubleshoot issues within the vehicle model code. He never shied away from a task and was always willing to assist in any way possible. Second, the previous engineering manager, Aaron Mull. Aaron and I spent countless

iii

nights in the lab troubleshooting issues with the vehicle. He exposed me to new methods of approaching a problem and how to find promising solutions. Aaron contributed a great deal of his personal time to the project, and I am confident that the vehicle would not be functioning as well without the insight that he provided during his time with and after leaving the team. Thirdly, I would like to thank Thomas Harris. Thomas was the reason I joined the EcoCAR team and provided mentorship to me during my early months on the team. His guidance helped lead me to the fulfillment of the lead role in my second year with the team. I will always be grateful for the opportunity that Thomas gave me to work on the EcoCAR team beside him during his time at WVU. I thank Clay Vincent and Zachary Flanigan for their involvement with the CAVs sub team. Advancements have been made for sensor fusion and vehicle communication that would not have been possible without their dedication to the project. I feel that I speak for everyone when I say they are some of the hardest working members on the EcoCAR team. Finally, I would like to thank our communications manager, Ryan Alexander. Ryan provides meaning for our work in multiple ways, including youth outreach events, where communication involving the STEM field is crucial both to our competition and the academic field. With his leadership, our team secured first place in the final competition communications presentation in year 3 of the competition, helping us to secure third place overall. His hard work and dedication are unmatched, and he has helped the team reach audiences throughout West Virginia. I would like to thank everyone on the WVU EcoCAR team for their continued work in the program, both past and present members.

Lastly, I would like to thank the WVU EcoCAR General Motors mentor Dr. William Cawthorne for his involvement with the team. Bill never hesitates to take time out of his schedule to organize meetings where he talks with different groups about areas of concern with the vehicle. He has spent countless evenings drawing diagrams, modeling systems, and helping to troubleshoot issues in the vehicle. The guidance and technical insight that Bill provides has been invaluable, and he helped the team to obtain first place in the energy consumption event at the year 3 final competition. Bill has expanded my

iv

knowledge of control systems and introduced me to different methods of approaching problems that I never thought were possible. I will always be grateful for his continued guidance throughout my academic career.

#### Dedication

This thesis is dedicated to my parents and family members. Everyone has provided unwavering support throughout my academic career, helping to keep me on the proper track to pursue further education. I would like to thank them for teaching me to have a strong work ethic and a desire to never stop learning. My parents taught me that failure is natural but does not define you as a person. Rather, failures are the steppingstones to continue working hard and overcome any obstacles I encounter in my academic career. I would not be where I am today without my parents continued guidance and support.

# Contents

Table of	Figur	res x
List of Ta	ables	xiv
List of A	crony	/msxvi
List of Sy	ymbo	lsxviii
1. Inti	roduc	tion1
1.1	Emi	issions2
1.2	Hyb	orid Electric Vehicles3
1.3	Hyb	orid Vehicle Classifications4
1.4	HE\	/ Architecture5
1.5	HE\	/ Parallel Architecture7
1.6	Eco	CAR Mobility Challenge
1.7	We	st Virginia University Vehicle Architecture11
1.7	.1	General Motors Donated Components12
1.7	.2	Third Party Donated Components12
1.7	.3	Vehicle Power Flow12
1.7	.4	Component Optimization14
2 Lite	eratur	re Review
2.1	Ove	erview of HEV Control Strategies19
2.2	Ene	ergy Management Control Problem21
2.3	Dyr	namic Programming22
2.4	Equ	ivalent Consumption Minimization Strategy26
2.5	Ada	aptive Optimal Supervisory Control29
2.6	Ado	ditional HEV Work
2.7	Fuz	zy Logic
3 Me	ethod	ology44

	3.1	Veh	icle Model Overview	45
	3.2	Driver Model		46
	3.3	Plar	nt Model	48
	3.3.	1	Engine Model	50
	3.3.	2	Battery Model	53
	3.3.	3	Electric Motor Model	56
	3.3.	4	Vehicle Drivetrain	58
	3.4	Con	troller Model	64
	3.4.	1	Accel Pedal and Regen Braking	66
	3.4.	2	Energy Management	67
	3.5	Fuzz	zy Logic Additional Control Block	73
4	Test	t Proc	cedure	80
	4.1	Opt	imal ECMS	82
	4.2	Ave	rage ECMS	84
	4.3	Fuzz	zy ECMS	86
5	Res	ults		87
	5.1	Brut	te Force Analysis	87
	5.2	Fue	l Economy Comparison	89
	5.3	SOC	Comparison	93
	5.3.	1	NYCC Cycle	95
	5.3.	2	US06 Cycle	100
	5.3.	3	EMC City Cycle	105
	5.3.	4	HWFET Cycle	110
	5.3.	5	EMC Highway Cycle	115
	5.3.	6	Remaining Cycles	122
	5.4	Ben	chmark Validation	123

6 Co	5 Conclusions and Recommendations124		
6.1	Analysis		
6.2	Additional Drive Cycles		
6.3	FLC Variation		
6.4	Model Modifications		
References			
Appendix A: Validation Drive Cycles137			
Appendix B: ECMS Brute Force Analysis MATLAB Code143			
Appe	Appendix C: F-ECMS Validation Results		

Table of Figures Figure 1: HEV Architecture [7]	6
Figure 2: Parallel HEV Configurations [8]	7
Figure 3: WVU Hybrid Power Flow	
Figure 4: BSFC Map for an IC Engine [15]	
Figure 5: Vertical Translation in BSFC	
Figure 6: Horizontal Translation in BSFC	
Figure 7: Electric Motor Efficiency Map	
Figure 8: EcoCAR Mobility Challenge City Drive Cycle	
Figure 9: DP Cost-to-go Matrix	23
Figure 10: Varying Penalty Function for HV SOC	
Figure 11: Overview of Fuzzy Logic Architecture	
Figure 12: Fuzzy Logic Trapezoidal Membership Function	
Figure 13: Clipped Trapezoidal Membership Function	
Figure 14: Hybrid Vehicle Model Overview	
Figure 15: Longitudinal Driver	47
Figure 16: Drive Cycle Multiport Switch	
Figure 17: Plant Model Overview	
Figure 18: Plant Model Application Layer	
Figure 19: Engine Subsystem	51
Figure 20: Engine Control Module/Mapped Engine Model	52
Figure 21: Powertrain Blockset Battery Model	53
Figure 22: Battery System Manager	54
Figure 23: Maximum Discharge Calculations	

Figure 24: Discharge Buffer Calculation	55
Figure 25: Maximum and Minimum Power Calculation	56
Figure 26: Mapped Motor Model	57
Figure 27: Motor Torque Limits	58
Figure 28: Vehicle Drivetrain Overview	58
Figure 29: Stateflow Shift Map	59
Figure 30: Transmission Up-Shift Threshold	59
Figure 31: Torque Converter Automatic Transmission Model	60
Figure 32: Brake Pressure Calculation	61
Figure 33: Differential and Compliance Subsystem	62
Figure 34: MathWorks Differential [49]	62
Figure 35: Electric Motor Differential	62
Figure 36: Wheels and Brakes	63
Figure 37: Vehicle Body Dynamics	64
Figure 38: Controller Overview	65
Figure 39: Controller Application Overview	65
Figure 40: Accel Pedal and Regen Braking System	67
Figure 41: Energy Management Overview	68
Figure 42: Motor Wheel Torque Determination Subsystem	68
Figure 43: Powertrain Constraints	69
Figure 44: ECMS Overview	71
Figure 45: Penalty Factor for a = 3	72
Figure 46: Fuzzy Logic Controller	74
Figure 47: Delta SOC Membership Functions	74

Figure 48: Wheel Torque Command Membership Functions	75
Figure 49: Delta EF Membership Functions	76
Figure 50: Fuzzy Logic RuleViewer Toolbox	77
Figure 51: Inference Rule Matrix Surface Plot	77
Figure 52: High-Level Control with Integrated Fuzzy Logic Controller	80
Figure 53: Calibratable Variable	83
Figure 54: Driver Style Comparison	86
Figure 55: NYCC Drive Cycle	95
Figure 56: NYCC 50% SOC Comparison	97
Figure 57: NYCC 50% EF Comparison	98
Figure 58: NYCC 32.5% SOC Comparison	99
Figure 59: NYCC 32.5% EF Comparison	
Figure 60: US06 Drive Cycle	
Figure 61: US06 50% SOC Comparison	
Figure 62: US06 50% EF Comparison	
Figure 63: US06 32.5% SOC Comparison	104
Figure 64: US06 32.5% EF Comparison	105
Figure 65: EMC City Drive Cycle	
Figure 66: EMC City 50% SOC Comparison	107
Figure 67: EMC City 50% EF Comparison	
Figure 68: EMC City 32.5% SOC Comparison	
Figure 69: EMC City 32.5% EF Comparison	110
Figure 70: HWFET Drive Cycle	111
Figure 71: HWFET 50% SOC Comparison	112

Figure 72: HWFET 50% EF Comparison	
Figure 73: HWFET 32.5% SOC Comparison	
Figure 74: HWFET 32.5% EF Comparison	
Figure 75: EMC Highway Drive Cycle	116
Figure 76: EMC Highway 50% SOC Comparison	
Figure 77: EMC Highway 50% EF Comparison	
Figure 78: EMC Highway 50% EF 1000-2000 Seconds	119
Figure 79: EMC Highway 32.5% SOC Comparison	
Figure 80: EMC Highway 32.5% EF Comparison	
Figure 81: EMC Highway 32.5% EF 1000-2000 Seconds	
Figure 82: NYCC Drive Cycle	137
Figure 83: SC03 Drive Cycle	137
Figure 84: US06 Drive Cycle	
Figure 85: EMC City Drive Cycle	
Figure 86: RTS95 Drive Cycle	139
Figure 87: HWFET Drive Cycle	139
Figure 88: HUDDS Drive Cycle	140
Figure 89: EMC Highway Drive Cycle	140
Figure 90: UDDS Drive Cycle	141
Figure 91: LA92 Drive Cycle	141
Figure 92: Artemis Rural Road Drive Cycle	142
Figure 93: SC03 50% SOC Comparison	147
Figure 94: SC03 50% EF Comparison	148
Figure 95: SC03 32.5% SOC Comparison	

Figure 96: SC03 32.5% EF Comparison	
Figure 97: RTS95 50% SOC Comparison	
Figure 98: RTS95 50% EF Comparison	
Figure 99: RTS95 32.5% SOC Comparison	
Figure 100: RTS95 32.5% EF Comparison	
Figure 101: HUDDS 50% SOC Comparison	
Figure 102: HUDDS 50% EF Comparison	
Figure 103: HUDDS 32.5% SOC Comparison	
Figure 104: HUDDS 32.5% EF Comparison	
Figure 105: UDDS 50% SOC Comparison	
Figure 106: UDDS 50% EF Comparison	
Figure 107: UDDS 32.5% SOC Comparison	
Figure 108: UDDS 32.5% EF Comparison	
Figure 109: LA92 50% SOC Comparison	
Figure 110: LA92 50% EF Comparison	
Figure 111: LA92 32.5% SOC Comparison	
Figure 112: LA92 32.5% EF Comparison	
Figure 113: Artemis Rural Road 50% SOC Comparison	
Figure 114: Artemis Rural Road 50% EF Comparison	
Figure 115: Artemis Rural Road 32.5% SOC Comparison	
Figure 116: Artemis Rural Road 32.5% EF Comparison	

# List of Tables

ble 1: Inference Rule Matrix
------------------------------

Table 2: Fuzzy Set Intersection	78
Table 3: Fuzzy & Crisp Set Intersection	78
Table 4: Experimental Factors and Respective Levels	81
Table 5: Validation Drive Cycles and Characteristics	82
Table 6: Optimal Equivalence Factors	85
Table 7: Optimal ECMS Results	89
Table 8: Fuel Economy Results in mpg for Average and Fuzzy EF for Aggressive and Mild Drivers at 5	60%
and 32.5% Initial SOC	90
Table 9: Total Fuel Economy % Comparison	92
Table 10: Average and Fuzzy ECMS Ending SOC Comparison	94
Table 11: Benchmark Fuel Economy Comparison	123

# List of Acronyms

ACC	Adaptive Cruise Control
A-ECMS	Adaptive energy consumption minimization strategy
A-OSC	Adaptive optimal supervisory control
ADAS	Advanced Driver Assistance System
AVTC	Advanced Vehicle Technology Competition
AWD	All-wheel drive
ANL	Argonne National Laboratory
ANN	Artificial neural network
AMT	Automatic manual transmission
BP	Back propagation
BEV	Battery electric vehicle
BSM	Battery system manager
BWS	Battery working state
BSFC	Brake specific fuel consumption
CNN	Chaining neural network
СР	Charge Depletion
CS	Charge Sustaining
CAVs	Connected and automated vehicles
DFCO	Deceleration fuel cut off
DOF	Degree-of-freedom
DSOC	Delta State of Charge
DOE	Department of Energy
DP	Dynamic Programming
eAWD	Electric all-wheel drive
EM	Electric motor
eRAD	Electric rear axle drive
ECMS	Energy consumption minimization strategy
EF	Equivalence factor
FWD	Front-wheel drive
F-ECMS	Fuzzy based ECMS
FLC	Fuzzy logic controller
GM	General Motors
HV	High voltage
HEV	Hybrid Electric Vehicle
ICE	Internal combustion engine
Km	Kilometer
kW	Kilowatt
kWh	Kilowatt hours
LN	Large negative

LP	Large positive
L	Liter
MP	Medium Positive
(MabX)	MicroAutoBox
mpg	Miles per gallon
MIL	Model-in-the-loop
NEDC	New European Driving Cycle
Nm	Newton-meters
OC	Opportunity charging
PHEV	Plug-in hybrid electric vehicle
PMP	Pontryagin's minimization principle
PI	Proportional-Integral
PCM	Propulsion controls and modeling
PSI	Propulsion systems integration
RBF	Radial basis function
SI	Spark ignition
SI	Spark Ignition
SOC	State of charge
S-DP	Stochastic dynamic programming
TSA	Torque-split algorithm
UDDS	Urban Dynamometer Driving Schedule
VIL	Vehicle-in-the-Loop
VMT	Vehicle miles traveled
VTI	Vehicle Technical Inspection
V2I	Vehicle-to-infrastructure
V2V	Vehicle-to-vehicle
VL	Very large
VS	Very small
WVU	West Virginia University
WTC	Wheel Torque Command

List of Symbols		
<u>Symbol</u>	Description	<u>Units</u>
	English	
Ah	Storage capacity of the ESS	Ah
$CO_2$	Carbon Dioxide	-
СО	Carbon Monoxide	-
$FE_{Avg}$	Fuel economy for the average ECMS algorithm	mpg
$FE_F$	Fuel economy for the fuzzy ECMS algorithm	mpg
J	Cost function	-
K <sub>i</sub>	Integrator gain	-
$K_p$	Proportional gain	-
kŴ	Kilowatt	kW
kWh	Kilowatt hours	kWh
L	Liter	L
mpg	Miles per gallon	mi/gal
$\dot{m}_{f.ena}$	Engine fuel flow rate	g/sec
$\dot{m}_{feav}$	Total fuel flow rate	g/sec
m <sub>ress</sub>	Equivalent fuel flow rate of the ESS	g/sec
Nm	Newton-meters	Nm
NOx	Nitrogen Oxide	-
p	Co-state vector	-
P <sub>hatt</sub>	HV battery power	А
$P_{ena}$	Engine Power	kW
$Q_{lhv}$	Lower heating value of fuel	Mj/kg
rpm	Revolutions per minute	rpm
S	Equivalence factor	-
sf c <sub>ea</sub>	Specific equivalent fuel consumption	g/sec
SOC	State of charge of the ESS	%
SOC <sub>max</sub>	Maximum SOC of the ESS	%
SOC <sub>min</sub>	Minimum SOC of the ESS	%
SOC <sub>taraet</sub>	Target SOC of the ESS	%
$t_f$	Ending time	sec
THC	Total hydrocarbon	-
$t_0$	Starting time	sec
Voc	Open circuit voltage	volts
$x_N$	Final state of the control variable x	-
$x_0$	Initial state of the control variable x	-
Ŷ	Cost-to-go	-
	Greek	
β	Positive weighting factor	-
δ	Penalty factor	-
$\eta_{em}$	Efficiency of the electric motor	%
$\eta_{eng}$	Engine efficiency	%
arphi	SOC deviation penalty factor	-
Ø	Penalty factor	-

### 1. Introduction

The objective of this research is to design and implement a fuzzy logic controller for implementation with an energy management strategy for a student designed hybrid electric vehicle (HEV). The main goal of the implemented control strategy is to improve from the baseline fuel economy for multiple unknown test cycles and maintain charge sustaining (CS) vehicle operations. The purpose of implementing an advanced control strategy in a hybrid electric vehicle is to maximize efficiency by splitting the torque between multiple components including the internal combustion engine (ICE) and electric motor. These advanced strategies are known as torque-split algorithms (TSAs) and exist in multiple ways; however, not all are created equally. Some strategies are rule-based which do not involve explicit optimization, but rather rely on different sets of rules that decide what value of control to apply. Optimization strategies involve calculating the optimal set point by minimizing a cost function over a known driving cycle which leads to a global solution [1].

Globally optimal strategies are the best performing; however, they are only implementable if all future driving conditions are known during the algorithm development period. Normal day to day driving prevents every driving condition from being available so adaptive strategies have been implemented to achieve close to optimal performance. These adaptive strategies are designed from the model-based optimization control methods where global optimums are used for baseline evaluation [1].

This work explains the implementation of a fuzzy logic-based controller (FLC) in tandem with an optimal control strategy. The FLC is used to tune the control strategy based on the driver style and several control parameters. The modeling and analysis described in this work is performed in the simulation, or model-in-the-loop (MIL), environment. The following pages describe the vehicle model used for the simulated engine, high-voltage (HV) battery and electric motor, vehicle dynamics, the control algorithm model, and the design and implementation of the FLC. Results from both a baseline control strategy and FLC strategy

are presented and analyzed at different initial setpoints. Lastly, conclusions are drawn, and recommendations and improvements are made for future work.

#### 1.1 Emissions

Each year the government continues to tighten emissions regulations, so the introduction of hybrid electric vehicles has become increasingly important. Global warming and the effect of combustion engine performance on the environment have led to developing all-electric means of transportation. In 2019, greenhouse gas emissions from transportation accounted for roughly 29% of the total greenhouse gas emissions in the U.S. [2]. In terms of the overall trend, from 1990 to 2019, the number of vehicle miles traveled (VMT) by light-duty vehicles increased by 48% due to contributing factors such as population and economic growth [2].

Several automotive manufactures, including General Motors (GM), are pushing to create an all-electric fleet to make a meaningful impact towards a zero-emissions future [3]. Companies like GM are spearheading the technological advancements needed to move the industry to becoming all-electric with no emissions from ICE or diesel engines. Before this change can happen, there are several factors that need to be addressed. Electric vehicles pose new problems, including additional trained personnel for maintenance, HV battery life, and charging stations. While electric vehicles do not require extensive maintenance work due to less moving parts when compared to an ICE engine, working with and servicing HV can be dangerous. Vehicles will also need to be able to drive for an extended range when compared to conventional vehicles that can obtain anywhere from 30-40 miles per gallon (mpg) for an average of 350-400 miles on a single tank. The last factor to consider is the amount of charging stations readily available for commercial use. Until these stations are as common and as efficient as gas stations, consumers may be hesitant to move from a conventional vehicle to an all-electric vehicle. This is where hybrid vehicle architecture will come into play.

#### **1.2 Hybrid Electric Vehicles**

HEVs can serve to bridge the gap for consumers who want to reduce their carbon footprints but are unsure of all-electric vehicle technologies. Hybrid vehicles can further help to reduce several categories of vehicle emissions, including both direct and cycle life. Direct emissions are emitted from a vehicle's tailpipe and include greenhouse gases. HEVs can produce less tailpipe related emissions when using gasoline because they can operate more efficiently [4]. Life cycle emissions include all emissions related to both fuel and vehicle production, use, and disposing. When determining life cycle emissions, all emissions are included for extracting petroleum from the ground, refining to gasoline, distribution, and vehicle consumption. Electric vehicles generally produce fewer life cycle emissions because most emissions are lower for electricity generation and can be further minimized by using electricity generated from renewable sources including solar or wind [4]. How can HEVs help to reduce emissions while improving fuel economy when they use a gasoline or diesel-powered engine to help provide propulsion for the vehicle?

Vehicle emissions can be reduced in several ways with the primary methods being engine optimization, energy capture through regenerative braking, and engine start/stop functionality. Conventional vehicle engines are less efficient than their electrical counterparts (30-40% efficient compared to 60-98% efficient); however, electric motors can be used to place the engine into a more efficient operating region. When the engine can operate at or close to the ideal operating torque region, fewer emissions will be produced, and fuel economy will be improved.

Electric motors also provide the additional advantage of regenerative braking. Regenerative braking occurs when an electric motor is used to provide generating, or negative, torque to an axle of the vehicle to capture energy while decelerating the vehicle. This energy can be stored in the HV battery for later usage without needing to use the engine to recharge the battery during operation which could result in lower fuel economy and increased emissions. Alongside regenerative braking is engine start/stop

functionality. At low vehicle or idle speeds, the engine can be shut off to prevent unnecessary fuel consumption and emission production. While this functionality is available in commercial ICE vehicles, using an electric motor provides the advantage of launching the vehicle from a stop or driving with the engine shut off for extended periods of time.

#### 1.3 Hybrid Vehicle Classifications

Hybrid vehicles can be broken down into two different classifications: Plug-in hybrid electric vehicles (PHEV) and non-plug in HEVs. PHEVs are characterized by large HV batteries that give a vehicle the advantage of functioning as a fully electric vehicle or a hybrid. These vehicles can be charged at home using automotive supplied cable that accepts power from a standard 120-volt outlet. When driving, the vehicle can follow charge depletion (CD) where the electric motor is used solely for propulsion until a predefined HV battery state of charge (SOC) is reached. The engine may also turn on during operation if the driver demanded torque exceeds the maximum available motor torque at that time. Once this threshold is crossed, the motor and engine can be used to provide propulsion for the vehicle in a CS mode where the SOC is maintained. Throughout the course of driving, if the SOC increases beyond an upper threshold, the engine may shut off to further increase fuel economy and reduce vehicle emissions.

HEVs can be characterized by their smaller battery packs. In standard driving conditions, the battery pack is not large enough to provide electric-only vehicle operation; however, the motor can be utilized in case of emergencies if the vehicle runs out of gas and needs to be moved off of the road. HEVs also cannot be plugged into a standard wall outlet to charge in between trips. Normal operating conditions for a HEV include using the motor to augment and optimize the operating modes of the engine. This makes the HEV appealing because although it does not provide electric-only driving, consumers are still able to reduce emissions while improving fuel economy. The PHEV and HEV both allow for engine optimization and regenerative braking; however, the HEV will be discussed in greater detail below due to the focus of this work. It should be noted that the results from this work can be applied to any hybrid architecture that utilizes an engine and electric motor.

#### 1.4 HEV Architecture

HEVs have three architectures: series, parallel and a combination of both (series-parallel) shown in Figure 1. A series hybrid (configuration B) is comparable to a battery electric vehicle (BEV) where the engine does not drive the wheels, rather, the engine is paired with an electric generator. The generator serves to charge the HV battery and power an electric motor that drives the vehicle. Series hybrids perform best in stop-and-go traffic situations where IC engines are more inefficient. When a large driver demanded torque is encountered, the motor can draw power from both the battery pack and generator for propulsion. However, the larger battery, motor and additional generator add to the overall cost of the vehicle making it more expensive than parallel hybrids [5].

In a parallel hybrid (configuration A), both the IC engine and electric motor work in tandem to propel the vehicle. One disadvantage when compared to other configurations is the battery size. Parallel hybrids tend to have smaller battery packs that rely on regenerative braking to keep the battery charged. However, the engine is connected directly to the wheels, which eliminates any inefficiencies of converting from mechanical power to electricity and back. Power from both components can be varied to ensure that each component is operating in its' most efficient region whenever possible. During a drive, if the generator is needed to recharge the HV battery, the motor can produce negative torque to generate current to charge the battery pack while the engine produces additional torque to satisfy the driver demand [5].

The series-parallel configuration (configuration C) consists of using the engine and motor to propel the vehicle separately as well as at the same time. At lower vehicle speeds, the vehicle can operate as a series and at higher vehicle speeds, when a series configuration is less efficient, the engine can take over and operate as a parallel hybrid. This configuration incurs higher costs than a parallel hybrid due to the larger

HV battery pack, generator, and more advanced computer system to control multiple systems when determining the best scenarios for using the engine. However, the added efficiencies mean that the series-parallel can outperform the two individual configurations by increasing fuel economy and reducing vehicle emissions [6].



Figure 1: HEV Architecture [7]

Each of these architectures can benefit from regenerative braking. The motor is used to provide negative or drag torque on an axle, the flow of power can be reversed from the motor through the inverter to the HV battery for recharging. If regenerative braking is not sufficient for recharging the battery, the system can make use of opportunity charging (OC).

Opportunity charging occurs when the engine is commanded to provide more torque than what the driver is requesting through the accelerator pedal. To maintain the initial driver demanded wheel torque, the motor provides negative torque equal to the difference between what the engine is currently providing and what the driver initially requested. The system compensates for the negative motor torque and recharges the HV battery while the driver demanded wheel torque is satisfied. OC can also be implemented to force the engine into a more efficient operating region at low torque/power demands if the HV battery has the capacity to store the captured energy.

#### 1.5 HEV Parallel Architecture

This work focuses purely on the parallel hybrid architecture, so the configurations for the parallel architecture will be highlighted and discussed. Six different architectures exist for a HEV that range from a P0 to a P5. These architectures are shown below in Figure 2 [8].



#### Figure 2: Parallel HEV Configurations [8]

In a P0 configuration, indicated by the green circle in the figure, the electric motor, or belt integrated starter generator, size is reduced to exist on the engine and is connected via a belt. This motor does not rely on a HV battery but rather is run on a 48V battery that replaces the vehicles low voltage battery for standard vehicle accessories. The P0 motor can supply up to 50 Nm of torque and provides several advantages including both idle start/stop functionality, engine load shift for increased efficiency operation regions, and energy recuperation [9].

The P1 motor is connected directly to the crankshaft of the IC engine (blue circle in above figure) and functions as a generator during deceleration events, as a starter in engine start/stop events, and as a

propulsive motor during varying acceleration events. The belt used from the PO motor is removed to reduce belt loses, however, due to the location of the motor on the engine side, it cannot be disconnected mechanically [9].

The P2, P3 and P4 architectures have better energy efficiency due to the positioning of the motor indicated in the above figure. The motor can be placed on the input shaft (P2) or output shaft (P3) of the transmission, or on a differential (P4). Depending on the motor being used and the placement in the vehicle, these motors can have an efficiency of up to 95% and can be used for engine start/stop, torque assist, energy recapture, and four/all-wheel drive if using a P4 architecture [9]. A P4 motor is not limited to the rear axle. If desired, the P4 motor could drive the front axle and the engine could be used to drive the rear axle. The P5 architecture resembles the P4, however, the motors are placed within the wheels to allow for increased energy recapture and efficiency.

#### 1.6 EcoCAR Mobility Challenge

Beginning in 1988, the U.S. Department of Energy (DOE) has sponsored Advanced Vehicle Technology Competitions (AVCTs) in partnership with leaders in the North American automotive industry. The AVTCs are managed by Argonne National Laboratory (ANL) and represent a coalition of industry, government, and academic partners to organize and execute the competition. These competitions challenge students beyond the traditional classroom, and feature opportunities for both graduate and undergraduate students in a variety of disciplines including project management, engineering, business, and communications. Students are tasked with re-engineering a donated production vehicle to improve efficiency while maintaining and expanding on consumer appeal items such as safety, cost, and performance. More than 20,000 students from 93 different institutions have participated, gaining realworld and hands-on experience when vehicle design and integration problems associated with building more advanced fuel-efficient vehicles [10]. There have been 12 competitions, starting with the Methanol

Marathon in 1988 to the Natural Gas vehicle Challenge in 1990 and in more recent years, the EcoCAR Mobility Challenge [11].

The EcoCAR Mobility Challenge is the latest vehicle competition in the AVTCs sponsored by the U.S. DOE. The challenge spans four years, starting in 2018, and challenges 11 universities to apply advanced propulsion systems and connected and automated systems to improve safety, consumer appeal, and overall vehicle efficiency [12]. All teams in the current competition have been donated, by GM, a new 2019 Chevrolet Blazer. The teams are tasked with redesigning and turning the Chevy Blazer into a HEV with level 2 SAE autonomy. Level 2 SAE autonomy is associated with advanced driving assistance systems (ADAS). ADAS can take over steering, acceleration, and deceleration in specific examples such as driving with adaptive cruise control on the highway. The car will maintain a set speed specified by the driver until a vehicle enters the safe distance space. At that time, the car will automatically decelerate to a safe following distance behind the lead vehicle. Level 2 autonomy still requires the driver to be fully alert of all vehicles surroundings and must be ready to take over vehicle control if prompted.

Each of the competing university's teams consist of three main technical swim lanes: propulsion systems integration (PSI), propulsion controls and modeling (PCM), and connected and automated vehicles (CAVs). Each of the swim lanes must work together over the 4-year competition to convert a stock ICE vehicle to a hybrid. The PSI sub-team consists of the mechanical and electrical teams. The mechanical team is responsible for any mechanical integration on or in the vehicle. The donated Chevy Blazer contained a 3.6-liter engine, so the mechanical team was responsible for performing an engine and transmission swap to install a smaller engine, increasing fuel economy. The mechanical team is also responsible for any other hardware installation, including the electric motor, HV battery pack, competition required toggle and emergency disconnect switches, etc.

The electrical team is responsible for any low or high voltage wiring that is implemented in the vehicle. When the electric motor and HV battery are integrated into the vehicle, the inverter/motor must be electrically connected to the HV battery pack and all components must be safely secured within the vehicle. Teams work with competition sponsors to ensure all HV wiring is done safely and meets industry standards. The teams also add several third-party controllers to the vehicle that require both power and communication wires. A low voltage power distribution system is added in the vehicle to allow the team to easily power any additional relays or controllers that are added in later years of the competition. VeSys is used to create wiring diagrams to document all team added wiring harnesses in the vehicle. Each year competition organizers inspect the vehicles in an event called the vehicle technical inspection (VTI). The PSI team is responsible for correcting any changes the competition organizers require so that the vehicle can be certified for on-road testing.

The PCM team is responsible for designing/programming and implementing various energy management strategies, power moding, component interfacing, and diagnostic systems in the vehicle. In the first year of the competition, the PCM team was responsible for different architecture modeling and the results drove the component selection for the remainder of the competition. Once strategies and algorithms are designed and tested, the team is responsible for flashing the created code onto a MicroAutoBox II (MabX) that interfaces with the stock controllers in the Blazer. Competition organizers require all teams to perform a minimum of 10 hours of closed course testing before the vehicle can be recommended for on road testing. This requires the PCM team to implement several layers of version control to ensure that all software passes individual testcases before being implemented in the car for closed course testing. Vigorous testing is carried out by the PCM swim lane on all team added controls to ensure full vehicle functionality before moving to on-road testing.

The CAVs team is responsible for both the team designed adaptive cruise control (ACC) system and sensor fusion. Team added sensors, including Bosch radars and a Mobileye camera, are used to move the vehicle

to level 2 autonomy. The CAVs team creates algorithms to combine dynamic sensor data from different locations on and in the vehicle to detect other vehicles on road and create a functioning ACC algorithm for the competition. Sensor fusion feeds relative vehicle speed and distance to the ACC controller which determines how fast or slow the vehicle should be moving. If no vehicles are present during operation, the ACC controller functions as a normal cruise controller.

Cohda wireless radios have also been introduced in the EcoCAR mobility Challenge and are utilized by the CAVs swim lanes. These radios support vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communications. Cohda wireless is one of the global leaders in mobile outdoor communication that will supply automotive industries with the software to optimize traffic efficiency through cooperative cruise control and infrastructure communication [13]. Teams are supplied with several radios for vehicle testing. The radios can be programmed to function as stop signs, traffic lights, and can be installed in several vehicles to create a network for vehicle communication. In Year 4 of the competition, teams are required to have basic ACC algorithms that will accelerate and decelerate the vehicle in unsmart intersections using wireless communication between 2 or more radios.

#### 1.7 West Virginia University Vehicle Architecture

Two distinct considerations drove the architecture selection for the West Virginia University (WVU) team: fuel economy results and integration considerations. Several vehicle architectures were analyzed in Year 1 of the Mobility Challenge by the PCM and PSI teams. Each architecture was simulated with a specific control strategy to determine which would pose the best fuel economy. This optimization period in Year 1 gave the WVU team a complete understanding of each architecture and the risks and rewards associated with them. Due to the nature of the competition requiring teams to have a fully functional hybrid in three years after the design phase in Year 1, the team selected a P4 architecture. The selected architecture may not be the most ideal, however, the team was confident in their ability to achieve full hybrid functionality and set themselves apart from the rest of the competition each year.

#### 1.7.1 General Motors Donated Components

The following components were donated by GM and reduced overall costs for the team in the Mobility Challenge. The selected engine is a 4-cylinder GM 2.5L LCV engine which is rated for a maximum torque of 259 Nm at 4400 revolutions per minute (rpm), maximum power of 151 kilowatts (kW) at 6300 rpm with a maximum engine speed of 7000 rpm [14]. The engine is paired with a GM 9-speed M3D 9T50 transmission. The transmission has an accumulator to enable engine start/stop functionality and has a wider 7.6:1 overall gear ratio that supports 'off-the-line acceleration' and low-rpm highway cruising [14]. The selected HV battery is the GM HEV4 battery pack. This pack has a nominal voltage of 300 volts with a total energy storage capacity of 1.5 kilowatt hours (kWh) and a peak power of 50 kW.

#### 1.7.2 Third Party Donated Components

The electric motor is an electric axle with a differential gearbox and integrated motor procured from Magna Powertrain. The electric axle is known as the electric all-wheel drive (eAWD) unit or electric rear axle drive unit (eRAD) and is used in the Volvo V60 hybrid. The eAWD has a peak power slightly greater than the battery pack at 60 kW. It has a maximum torque rating of 200 Nm with a continuous torque rating of 90 Nm. The unit also has a maximum speed of 12000 rpm. The gear ratio of the differential is roughly 9:1. Each eAWD system comes with an inverter that is uniquely paired with each motor.

#### 1.7.3 Vehicle Power Flow

The WVU P4 architecture has two primary modes of power flow: all-wheel drive (AWD) and front-wheel drive (FWD) with both regenerative braking and opportunity charging. These two modes are shown in Figure 3 below where the arrows indicate the power flow direction with blue arrows indicating mechanical power and orange arrows indicating electrical power. In AWD, the engine and motor are both used to

provide power to their respective axles, propelling the vehicle forwards. The left-hand illustration only highlights propulsive torque to the rear axle to demonstrate AWD capabilities. In both AWD and FWD, the motor can be used for regenerative braking shown in the right-hand side of Figure 3. During a coasting or deceleration event, the motor can be utilized to produce various amounts of negative torque to capture free energy from the vehicle's inertia. This torque production reverses the power flow and charges the HV battery. A slower deceleration will capture larger amounts of energy from regenerative braking.



Figure 3: WVU Hybrid Power Flow

In FWD with opportunity charging, the engine is commanded to supply an excess amount of torque greater than what the driver initially requested. This difference in this torque command from the driver versus the engine is made up by the motor. The motor is used to drag the rear axle by producing a negative torque equal to the difference of the driver and engine torques. This allows the HV battery to be recharged

if the SOC has dropped too low and can also be used to load the engine and place it into a more efficient operating region which will be discussed below.

#### 1.7.4 Component Optimization

From the above discussion, the engine can provide substantially more power when compared to the eAWD system. This difference in power, combined with a smaller HV battery pack led the team to the decision of not having a CD mode except for emergencies where the engine was not able to sustain propulsion for the vehicle. However, the eAWD system can be used to augment the engine efficiency operations. Now the question can be asked: How can a smaller motor be used to increase the efficiency of a larger engine?

First, engine characteristics need to be understood. What does engine efficiency mean? This entails moving the engine into an area where the brake specific fuel consumption (BSFC) is minimized. The BSFC is a function of both the torque being produced and the current speed of the engine. Engine speed can be determined by the speed of the vehicle wheels and the corresponding gear ratios. The gear ratios are defined by both the current transmission gear and a differential gear ratio. The engine torque is a function of the deflection of the accelerator pedal being manipulated by the driver. Reducing the BSFC will help to maximize the efficiency of the engine, but now the problem is creating an effective and efficient torque split algorithm that can recognize engine inefficiencies. If the most efficiency of the electric motor. If the vehicle was not equipped with a large energy storage system to handle the larger load on the motor, the HV battery would be drained quickly. The torque split algorithm selected must understand the operating regions for both propulsion components in order to calculate the ideal operating regions. This involves using the motor to reduce or increase the load on the engine to minimize the BSFC.

Consider an engine efficiency map shown in Figure 4 shown below [15]. The engine speed in rpm is on the x-axis and the available engine torque is located on the y-axis. The red line indicates the maximum engine torque available from the engine at any given operating point. The multicolored lines indicate the BSFC at different operating points. Due to confidentiality agreements with the EcoCAR competition and GM, this BSFC map is not for the 2.5-liter engine discussed above, but rather it is for a similar class engine.



Figure 4: BSFC Map for an IC Engine [15]

This engine is most efficient between the range of 1500-2900 rpm at a given load of 85-115 Nm of torque. At nominal cruise conditions on the highway (55-70 mph), a standard IC engine only produces the torque required to overcome friction and maintain the desired vehicle speed. However, to be more efficient, the engine may be asked to produce additional torque to reduce the BSFC. This is where the additional electric motor will come into play.

Now consider the following scenario shown in Figure 5. If the engine is producing roughly 60 Nm of torque at 2200 rpm (point A), it is less efficient than an operating point at 100 Nm and roughly 2200-2300 rpm (point B). To move the engine to the more efficient operating point B, the torque split algorithm can

increase the amount of torque requested from the engine. However, this will increase the overall torque production of the vehicle violating the initial torque request from the driver through the accelerator pedal causing the vehicle to accelerate. To mitigate this overshoot, the electric motor can be used to provide negative torque on the rear axle of the vehicle. This negative torque will be equal to the increased engine torque (roughly 40 Nm), which will allow the engine to be loaded into a more efficient region, reducing fuel consumption. Additionally, the negative torque production from the motor can be used to charge the HV battery in a situation called through the road or opportunity charging.



Figure 5: Vertical Translation in BSFC

Now consider the following situation (Figure 6) when the engine is producing a higher amount of torque at an increased engine speed, such as climbing up a grade. The engine can move into the more efficient BSFC region by producing less torque while the motor provides increased torque to maintain the original driver request. For this transition to happen, however, the engine speed must also be adjusted. Engine speed, as discussed above, is a direct reflection of the vehicle wheel speed and gear ratios. To translate to a lower engine speed, the transmission may need to shift to a lower gear, reducing the engine rpms. Programmable shift maps can be implemented in a control scheme to recognize the need for a reduced engine speed to translate horizontally while the engine torque is also reduced to move the engine from point A to point B.



Figure 6: Horizontal Translation in BSFC

Motor efficiencies are also a function of speed and torque; however, the motor operates with a single fixed gear ratio. An efficiency map of an electric motor is shown below in Figure 7. When a motor is producing positive, or propulsive torque in a forward direction, it is 'motoring'. Similarly, when the motor is producing negative, or brake torque in a backward direction, it is 'generating' or producing negative power to recharge the HV battery. In the figure below, motoring torque is on the positive y-axis and generating torque is on the negative y-axis.


Figure 7: Electric Motor Efficiency Map

The HV electrical system is composed of the motor, inverter and HV battery, so the losses from the battery and inverter need to be considered when designing an efficient torque split algorithm. Both the inverter and HV battery can be subjected to losses due to internal resistances when converting from DC to AC current and vice versa. All propulsion systems and respective losses need to be accounted for when designing a control strategy. This includes understanding component operating points and their respective efficient operating regions.

The goal of this work is to design an optimal control strategy using a FLC. The controller will be used to augment the optimal control strategy known as the energy consumption minimization strategy (ECMS). This selected control strategy is only optimal if all future driving conditions are known *a priori*. All future driving conditions cannot always be known in the real-world, so a FLC will be implemented to analyze current driving conditions to update and optimize the control parameter, or equivalence factor, of the ECMS.

# 2 Literature Review

The following section contains an outline of previous work completed in the development of hybrid control strategies for automotive and aerial vehicles which work to produce optimal torque strategies to increase fuel economy in different vehicles. Fuzzy logic control strategies are also examined in greater detail due to the nature of this paper.

### 2.1 Overview of HEV Control Strategies

When designing a HEV control strategy, there are two important factors to consider: Improving fuel economy and reducing vehicle emissions. This strategy is highly complex and must take component limitations and efficient operating regions into consideration. If the control strategy is not efficient or smart enough, all torque would be commanded from the electric motor, maximizing efficiency for the engine. However, this course of action would result in the HV battery being drained rather quickly. Instead, a strategy that allows the SOC to be maintained around a setpoint is introduced. This thesis will focus solely on maximizing fuel economy and charge sustainability through simulations with the design of a control strategy.

With the increase of HEVs in the world, there are also an increase in the control strategies used for maximizing efficiency. Automotive companies are constantly competing to design the best systems for consumer appeal and marketability, promising more fuel economy than their competitors. However, these control strategies can ultimately be broken down into 2 distinct categories: Rule-based and Optimal based strategies.

The main characteristic of heuristic control, or rules-based, strategies is their effectiveness to run in realtime. These rules do not involve any kind of explicit optimization, which reduces the computation power required to run in real-time. The rules used in this strategy are designed from intuition, heuristics, and/or the knowledge from other global optimum solutions [1].

In optimization strategies, the optimum actuator set points are calculated from the minimization of a cost function over some predefined drive cycle, known *a priori* which leads to the global optimal solution. These strategies find the minimum value of the cost function using the acquired knowledge of future driving conditions. Optimization strategies cannot be used directly for real-time implementation due to the computation power required and future driving conditions that are required. Instead, they can be used to design rules that can be used for online implementation [1].

Optimization methods can be further divided into two approaches: numerical and analytical. Numerical optimization methods include dynamic programming, genetic algorithms, and stochastic dynamic programming. When this method is used, the entire drive cycle must be taken into consideration to find the global solution numerically. Analytical methods use an analytical problem formulation to find the solution in a closed form that makes the numerical solution faster. Included in this method are Pontryagin's minimum principle (PMP) and the ECMS. These strategies can make use of a receding-horizontal approach for past, present, and future driving conditions [1].



Optimization strategies make use of drive cycles similar to the one shown below in Figure 8.

Figure 8: EcoCAR Mobility Challenge City Drive Cycle

Drive cycles are a function of vehicle speed specified at each instance that are used to test varying characteristics including fuel economy, performance, and emissions. The cycle shown above is used to test vehicle performance and fuel economy when driving in city conditions with several aggressive vehicle accelerations. Drive cycles, like the one shown above, can be used to determine the torque demands from the vehicle or powertrain being simulated or tested. If the total torque or power demands are found using *a priori* knowledge, optimization strategies can be used to determine an efficient cost-effective split between the IC engine and electric motor.

### 2.2 Energy Management Control Problem

Rizzoni et al. [1] discusses a formula to minimize the total mass of fuel consumed, as follows:

$$J = \int_{t_0}^{t_f} \dot{m}_f(u(t), t) dt$$
 1.1

Where  $\dot{m}_f$  is the mass flow rate of fuel used (g/sec), u(t) is the control variable leading to the minimization of fuel consumed for the cost function, J, over a drive cycle starting at time  $t_0$  and ending at time  $t_f$ . The cost function varies based on constraints in the system that can be broken down into two categories: global and local constraints. Global constraints include imposed constraints, such as state of charge (SOC) targets while local constraints focus more on component power, speed, and torque limits and/or predefined SOC boundaries. Other local constraints can also include drivability and driver comfort issues [1].

The energy management problem defined in equation 1.1 follows a global constraint to lower fuel consumption. This equation can be further broken down to impose additional constraints in the form of a penalty function as shown below in equation 1.2.

$$J = \emptyset\left(x(t_f)\right) + \int_{t_0}^{t_f} \dot{m}_f(u(t), t)dt$$
 1.2

Where  $\emptyset(x(t_f))$  is the new penalty function added to the performance index to obtain a chargesustaining performance index. Additional soft constraints can be applied in order to further constrict the energy management strategy including factors such as drivability, thermal dynamics, battery aging and life of product, ICE emissions, and varying torque production [1]. The following sections provide methods used to solve equation 1.1 with the additional penalties applied in equation 1.2.

### 2.3 Dynamic Programming

Dynamic Programming (DP) uses numerical methods to solve the global energy management problem for a given drive cycle by operating backwards over time. DP can provide the optimal solution to any variety of complex problems within computational limits, but it is only implementable using a simulation environment. Information must be known *a priori* for DP to be successful due to the nature of looking at the optimization horizon.

The algorithm used for DP is based on Bellman's principle of optimality, which starts from the final step and works backwards to generate an optimal cost-to-go solution as shown below [1]:

$$u_k = \mu^*(x_k, k) = \arg\min_{u \in U_k} \left( L_k(x_k, u) + Y_{k+1}(f_k(x_k, u_k), u_k) \right)$$
1.3

Where k = N-1..., 1.  $Y(x_1, 1)$  is found in the last iteration of the algorithm and is equal to the optimal cost  $J^*(x_0)$  discussed above.  $Y(x_N, N)$  is equal to the terminal cost and is dependent on the final state of  $x_N$ .  $Y(x_k, k)$  represents the optimal 'cost-to-go' from time k to the end of the optimization horizon.  $Y_k(x_k, u_k)$  is dependent on the control variable  $u_k$  which represents all values that the cost-to-go function can assume. The control sequence is performed backwards where values for the optimal choice at each time instance k and state value  $x_k$  are stored in the matrix  $u^*$  [1]. An example of a DP algorithm is shown below in Figure 9 [1]. The numbers on each segment represent the cost from the minimum cost-to-go function Y(x, k) at each instant. The algorithm solves from N to 1 on the time index axis.



Figure 9: DP Cost-to-go Matrix

Once the DP algorithm is finished, the minimum cost is selected for the implemented control algorithm. Wang et al. [16] applied DP to a series, parallel, and series-parallel powertrain for a Toyota Prius to find a potential for fuel economy improvement. The simulation results showed a 30% potential improvement in overall cost, which converts the electrical costs to fuel costs. From these results, an online real-time control algorithm was developed based on the DP results. In comparison to the simulated DP results, the online controller showed a 27% improvement in overall costs, which was very close to the original global optimized costs using DP.

Other methods of DP involve solving the problems analytically to reduce the overall computational loads. Work done by Larsson et al. [17] investigated the possibility of using an analytic solution for a continuous control signal. For this work, two different approximations of the cost-to-go function were considered. The first was a linear approximation and the second was a quadratic spline approximation. It was found that the computational time can be reduced by just below two orders of magnitude in exchange for a slight decrease in fuel economy simulation accuracy. In addition, when using the quadratic spline cost-togo function, the storage requirements for memory allocation can be significantly reduced (almost two orders of magnitude). This reduction in computational time and memory exhibits the option to implement this method in a vehicle controller unit. The authors state that, while the proposed method is technically an offline simulation, it is possible to perform the computations at a higher level on an external device which can transmit the solution to the vehicle via a cellular network. One downside to this method is that the specific analytical solution is not generic, so a solution must first be obtained for each individual powertrain configuration.

Similar work has been conducted by Yuan et al. [18] for a parallel HEV where DP was compared to PMP. The energy management problem was formulated as an optimal control problem between total fuel consumption and gear shifting frequency with admissible constraints. The algorithms are applied to a vehicle with an automatic manual transmission (AMT) as a potential replacement for a standard automatic transmission with a torque converter. The vehicle powertrain is used as the dynamic system with two applied state variables (battery SOC and AMT gear position). The independent control variables used are the engine throttle signal and gear shifting action. For the DP optimization, the cost function is as follows:

$$J(k) = \min_{u(k)} \{ J(k+1) + \dot{m}_f(k) + \beta | sh(k) | \}$$
 1.4

Where u(k) is the optimal control for the entire problem horizon,  $\beta |sh(k)|$  is introduced to limit excessive shifting, and  $\dot{m}_f$  is the flow rate of the equivalent fuel consumption rate.  $\beta$  is a positive weighting factor and can be tuned to reach an equilibrium point between the shifting frequency and fuel consumption. Similarly, an optimization function is created for the PMP comparison as follows:

$$[sh^{*}(t), th^{*}(t)] = \arg\min_{u(t)} H$$
 1.5

Where  $sh^*(t)$  and  $th^*(t)$  are the optimal control variables obtained at each instant to minimize the Hamiltonian defined in equation 1.6.

$$H(x(t), u(t), t, p(t)) = p^{T}(t) * f(x(t), u(t)) + \dot{m}_{f}(t) + \beta |sh(t)|$$
 1.6

Where p(t) is a vector of auxiliary variables called co-states. When compared side by side, the PMP produced near-optimal results that were close to the results from DP. While the PMP may not be exact, it can save approximately 77% of the computational time when compared to DP by approximating the co-states. With a reduced computational load, the PMP can be run on an online controller to achieve close to optimal results in a real-world environment.

A DP model of the 2019 Chevrolet Blazer used by the WVU in the EcoCAR Mobility Challenge was created by Aaron Mull of WVU [19]. Mull implemented a backwards operating model of the EcoCAR Blazer model in MATLAB by creating functions for each of the vehicle components, including the engine, battery, and motor. Cost functions were defined for each component to minimize fuel consumption and provide CS for multiple drive cycles. CS operation was imposed directly by selecting an initial state to be equal to the CS target SOC. Once constraints were imposed, the DP algorithm determined the effective cost-to-go matrix for each operating point from the start to end of each simulation. Simplified drive cycles were created to verify the functionality of the DP algorithm to ensure no constraints were violated during the optimization process. Once final cost functions were implemented, the DP algorithm was validated using a Simulink model of the 2019 EcoCAR Blazer. The DP algorithm was successful in calculating the best cost-to-go function for both drive cycles, and similar results were produced in the Simulink environment to validate all results. In the EMC City cycle, the maximum achievable fuel economy was found to be 30.18 mpg and in the EMC Highway cycle, the maximum fuel economy was found to be 37.24 mpg. In this paper, a similar Simulink model was implemented for the design of the F-ECMS algorithm. Due to multiple similarities between the models, the results from this DP algorithm will be used in the validation process of the F-ECMS algorithm designed and discussed in this report to further validate the effectiveness of this controller against the benchmark analysis performed by Mull.

Similar work has been done in the DP area including predictive control using Stochastic DP (S-DP) [20], real-time implementation based on DP [21] and other optimization methods using DP [22], however, due to the focus of this research paper, only a select few DP papers were highlighted.

### 2.4 Equivalent Consumption Minimization Strategy

The ECMS provides an effective solution to energy management problems discussed previously by using a heuristic method. ECMS is used in a charge-sustaining driving mode and works to use both the HV battery and electric motor as an auxiliary source of power. This algorithm works by applying a cost to the use of electrical, mechanical, and fuel energy when determining the optimum torque split ratio between both powertrains. The key idea, however, is that an equivalent fuel consumption value is associated with the use of all electrical energy in the system [1]. This relationship is shown below in the following equation:

$$\dot{m}_{f,eqv}(t) = \dot{m}_f(t) + \dot{m}_{ress}(t)$$
1.4

Where  $\dot{m}_f$  is the actual fuel consumption,  $\dot{m}_{ress}$  is the equivalent electrical fuel consumption, and  $\dot{m}_{f,eqv}$  is the total equivalent fuel consumption.

The actual fuel consumption of the engine is based on the lower heating value of the fuel used ( $Q_{lhv}$ , MJ/kg) along with the engine efficiency ( $\eta_{eng}$ ) and power produced by the engine ( $P_{eng}$ ) as shown below.

$$\dot{m}_f(t) = \frac{P_{eng}(t)}{\eta_{eng}(t)Q_{lhv}}$$
 1.5

The electrical energy is given an equivalent fuel consumption value, or virtual fuel, shown in equation 1.4 and can be found by assigning its' own virtual specific fuel consumption ( $sfc_{eq}$ , g/kWh) at the current time and multiplying by the battery power ( $P_{batt}$ ) [1]. It should be noted that proper unit conversions are needed in these equations when converting from MJ to kJ etc.

$$\dot{m}_{ress}(t) = sfc_{eq}(t) * P_{batt}(t)$$
 1.6

The virtual fuel consumption is determined by applying the equivalence factor (s(t)), which is a vector of values used for both charging and discharging the HV battery. This relationship can be seen in equation 1.7.

$$sfc_{eq}(t) = \frac{s(t)}{Q_{lhv}}$$
 1.7

Now the global problem of reducing the total cost can be condensed down to the local problem of minimizing the equivalent fuel consumption in equation 1.4. At each instance of time, the equivalent fuel consumption is determined for multiple candidates of the control variable  $P_{batt}$  which will provide the smallest equivalent fuel consumption value. This approach closely approximates the global solution and is less demanding on a computational scale when compared to other methods such as dynamic programming. The ECMS does not explicitly rely on information about the future driving conditions, however, the constant values for the equivalence factor must be selected beforehand which will affect both the power output and fuel consumption for the system [1].

During operation with the ECMS, boundaries must be determined to prevent the HV battery from violating any admissible limits. Due to these limitations, a multiplicative penalty function is generated to ensure all SOC limits are obeyed.

$$p(SOC) = 1 - \left(\frac{SOC(t) - SOC_{target}}{\frac{SOC_{max} - SOC_{min}}{2}}\right)^{a}$$
 1.8

This function takes deviations from the target SOC into account and will apply a penalty based on the direction of the deviation. Figure 10 shows 3 different penalty functions for varying values of the exponent a [1].



Figure 10: Varying Penalty Function for HV SOC

As seen in the above sigmoid function figure, smaller exponential values of *a* will result in steeper penalties being applied. As the SOC deviates to the left from the target towards the minimum specified SOC, the penalty being applied will increase which in turn will increase the cost to use the battery, making a discharge event less likely. If the SOC moves to the right from the target to the maximum specified SOC, the penalty will decrease, resulting in a lower cost to use the HV system for a discharging event.

This penalty function is added to equation 1.4 as follows:

$$\dot{m}_{f,eqv}(t) = \dot{m}_f(t) + \frac{s(t)}{Q_{lhv}} * P_{batt}(t) * p(SOC)$$
 1.9

Applying the ECMS through equation 1.9 provides results comparable to those achieved through DP. Different drive cycles require different values for the penalty function as well as different equivalence factors which must be obtained through numerical optimization beforehand. In ideal conditions through drive cycle simulation with prior knowledge, the ECMS results are very close to the global optimum. If the algorithm is used without *a priori* knowledge, the results are useable, but they are not as good as they could potentially be if the future driving conditions were known [1]. Adaptive methods have been introduced to provide optimal solutions without *a priori* knowledge of the driving conditions. A few

adaptive methods include fuzzy logic, neural networks, and proportional-integral-derivative (PID) controllers.

### 2.5 Adaptive Optimal Supervisory Control

Optimal fuel economy cannot be guaranteed if real-world information, such as the drive cycle or environment, is not available. To combat this problem, adaptive optimal supervisory control (A-OSC) algorithms are introduced [1]. The A-OSC can be broken down into numerous groups depending on how the control algorithm(s) is being optimized. For this paper, several methods will be discussed for an adaptive ECMS (A-ECMS) strategy, however, emphasis will be placed on using fuzzy logic with ECMS.

Adaptive drive cycle prediction focuses on adding 'on-the-fly' algorithms to the ECMS to estimate the equivalence factors. In the work performed by Sciarretta et al. [23], the method used for the A-ECMS was based on speed prediction for a parallel HEV. In this work, the equivalence factor was updated on an online controller using a look-ahead horizon to determine the most likely behavior of the battery in the near future for both charging and discharging situations. No predictions were needed for future conditions and only a few control parameters were required which vary from one vehicle to another. For one cycle discussed in the paper, the ECMS showed a potential for reducing fuel consumption by up to 30% when compared to the conventional value. For the urban driving cycle (UDC), the reduction was increased to 50% while the method imposed does not affect CS operations due to SOC excursions over each drive cycle being limited to 2%. The method was validated by comparing data using the steady-state fuel consumption which was not affected by SOC excursions, and the equivalent specific fuel consumption determined from the ECMS.

In similar work performed by Zhang et al. [24] for drive cycle prediction, a chaining neural network (CNN) was introduced for a velocity prediction approach. The developed CNN was used to predict the velocity over numerous temporal horizons using V2V and V2I communication. The proposed energy management

strategy was broken down into 3 parts. The first part included simulating vehicles in urban driving conditions to obtain real-time velocity and traffic information. The second step was to predict the velocity of a subject vehicle over various horizons using the CNN and given velocity of any lead vehicles. The final step was to adjust the EF by the defined adaptation law and apply the ECMS to split the commanded torque between the engine and motor. Several tests were conducted where the prediction time ranged from 5 to 20 seconds for three separate studies. This predicted velocity along with the SOC feedback was used to update the equivalence factor periodically. When evaluating the impact of the defined CNN parameters, a sensitivity analysis was conducted for different velocity prediction values and number of nodes in each hidden layer. The developed algorithm was compared to a base back propagation (BP) neural network to study improvements in the system over three different cases. The prediction error was found to increase as the prediction horizon expanded to 20 seconds and changed slightly from 20 to 30 seconds, however, the prediction error was reduced for all 3 cases when the velocity prediction and CNN parameter values were selected to be 2 and 20 respectively. ECMS when regulated by the proposed adaptation, showed a 0.2-5% increase in fuel economy for different prediction horizons in the three cases as opposed to ECMS with a traditional adaptation law.

Adaptive drive pattern recognition involves training an algorithm to recognize a previous driving pattern. The current algorithm parameters are then adjusted based on the information learned from previous drive cycle patterns. In work done by Jeon et al. [25], a multi-mode control algorithm using driving pattern recognition was developed and applied to a parallel HEV. Six driving patterns, including three urban, one expressway, and two suburban were selected for the training patterns. 24 parameters including stop/total time, average acceleration, and average cycle velocity were also selected to characterize the driving patterns. For each driving pattern selected, the control parameters were optimized using the Taguchi method through both fuel consumption and emissions simulations. These seven control parameters consisted of weighting factors for performance measures for deciding the ratio of engine to required

power from driving load. The additional control parameter was the charging/discharging method of the battery. A neural network decides, when driving, which representative driving pattern is closest to the current pattern by comparing the correlation related to the 24 characteristic parameters.

In work conducted by Harris et al. [26] of WVU, an on-board artificial neural network (ANN) was implemented to dynamically update the equivalence factor for the A-ECMS based on a sliding time window of past driving parameters. A radial basis function (RBF) ANN was used to implement the adaptive portion of the ECMS algorithm and was selected because the RBF can be trained very quickly by exposing it to the entire training data set all at once. The RBF consisted of one hidden layer and one output layer for the system where the weights between the two layers are updated during training. 30 drive cycles were used to train the RBF ANN and each cycle was characterized by 9 parameters. Multiple levels of variance in the RBF were examined along with different time windows (2, 3, and 4 minutes) that were used to update the equivalence factor. Of the 5 drive cycles used for validation of the RBF, 3 achieved a percent error within roughly 2.5% of the results from the optimal ECMS.

Kazemi et al. [27] of WVU proposed a predictive A-ECMS algorithm using intelligent transportation systems (ITS) for HEV powertrain control. 3 real-time control strategies were proposed for HEVs, each of which introduced an adjustment of the equivalence factor for the cost of electrical energy consumption. The first real-time strategy implemented a new definition for the EF, where the original EF was combined with a modification factor based on the energy requirements from a vehicle in the prediction horizon. This new formulation of the EF improved fuel economy by decreasing the number of engine on/off cycles during simulation. These cycles were prevented by decreasing the cost of using electrical energy when there was an upcoming regenerative braking event. The second strategy continued from the first but introduced a dynamic time horizon. The length of the time window was determined at each instance was defined by the time to the next minimum of the vehicle's drive cycle, indicated by the last moment of upcoming regenerative braking. In the third strategy, the EF was further modified to include the cost of

charging and discharging the battery in the near future. These costs were determined by running the vehicle model with the A-ECMS controller over a long drive cycle to determine the maximum and minimum EF values. Then, the best values of the adjustment EF variables were determined while the threshold of the EF was set to the maximum and minimum determined EF. The new strategy decreased the cost of electrical energy when there was a possible regenerative braking event on the prediction horizon, regardless of the SOC placement below the target value. Simulation results showed a fuel economy improvement of 1.8% for the UDDS drive cycle, a 4.1% improvement for the HWFET drive cycle and a 3.7% improvement in the US06 drive cycle.

## 2.6 Additional HEV Work

Additional work has been conducted which analyzes different aspects of a HEV control system outside of the global simulation-limited TSA area. The transmission shift map can be modified to allow the engine to operate in a more efficient region at different vehicle speeds. Many transmission shift maps are functions of the current vehicle speed and accelerator pedal position. However, work done by Connelly, et al. [28] at WVU, looked at developing SOC dependent and independent shift maps. This work was conducted in the EcoCAR3 challenge on a 2016 Chevy Camaro with a P3 architecture.

First, a sensitivity analysis was conducted in the MIL environment using the additional power available from the electric motor. This SOC independent shift map resulted in an increase to both the engine efficiency and fuel economy. A new three-parameter shift map was designed that added the current SOC as the third parameter in the map. With an SOC dependent shift map, additional solution space was analyzed to generate new shift lines in the map. When comparing the SOC independent to the dependent shift maps, the difference in fuel economy was negligible, however, both maps showed improvements when compared to the original stock shift map. In the same EcoCAR 3 competition with the 2016 Chevy Camaro, a power-loss minimization TSA was developed for the PHEV architecture by Derek George et al. [29] of WVU. A golden search algorithm was paired with a cost function to find the optimal engine torque available for the system. This developed function was compared to a base power-loss algorithm to determine the effectiveness of the implemented algorithm. The cost function served to minimize the total power-loss of the system while meeting the driver demanded wheel torque and CS functionality. Three cost functions were created for the analysis to determine the most effective power-loss strategy.

In the VIL environment, the compared cost function showed a 14.6% decrease in measured engine torque transients. In addition to the reduced torque transients, fuel economy was improved by 1.7%. An emissions analysis was also performed at the Center for Alternative Fuels, Engines, and Emissions research facility where carbon dioxide ( $CO_2$ ), carbon monoxide (CO), nitrogen oxide ( $NO_x$ ), and total hydrocarbon (THC) were measured. When compared to the base power-loss algorithm, there was a 10.4% decrease in CO, an 84.6% decrease in  $NO_x$ , a 15.6% increase in  $CO_2$ , and an 8.1% decrease in THC. In future work for this algorithm, it was recommended that a similar method of reducing the engine torque transients through A-ECMS be explored to further reduce emissions and improve fuel economy [29].

#### 2.7 Fuzzy Logic

Conventional control theory relies on an analytical model of a process to be controlled; however, this process can fall short if the model of the process is difficult to obtain or is highly nonlinear. How could a controller be designed to automate the process of driving a car? Automotive companies have begun designing control systems to produce autonomous vehicles, but humans have been driving for decades without needing mathematical models. This is where fuzzy logic can be introduced to help simplify the mathematical model. A fuzzy based control system is a real time expert level system that implements some experience and knowledge from a human operator. Fuzzy logic uses if-then rules and sets to define

the meaning of qualitative values for a controller input. This logic can capture the continuous nature of human decisions to improve on the standard binary control logic [30].

Take the example of a thermostat used to control a furnace. A simple control algorithm to control the temperature may state that if the temperature falls 5 degrees below the setpoint, turn on the furnace. Once the temperature is 5 degrees above the setpoint, shut the furnace off. Fuzzy if-then rules can be introduced to further control the temperature around a setpoint. If-then rules can be defined to state: If the temperature is *low* and the temperature delta is *small*, turn the furnace on *low*. While this is a basic example, the idea of fuzzy logic can be expanded to control more complex systems, such as the accelerator pedal commands in a vehicle for cruise control.

A fuzzy logic controller consists of three major areas or modules: the fuzzification input, inference engine, and defuzzification output, as shown below in Figure 11 [31].



Figure 11: Overview of Fuzzy Logic Architecture

Fuzzy properties called linguistic values (low, high) are introduced in the fuzzification input and define fuzzy sets with membership functions and values between 0 and 1. The fuzzification process represents the degree to which the crisp measurements belong to each of the fuzzy sets that are defined by the linguistic values. The inference engine if-then rules are used to create a fuzzy output or command which is converted back to a crisp output using the defuzzification engine.

Crisp inputs include physical values from the system such as a vehicle speed of 10 mph or an acceleration of 9.81  $\frac{m}{s^2}$ . When converting from the crisp inputs to the fuzzy inputs for the inference engine, linguistic variables and values are implemented. Linguistic variables are qualitative in nature and serve as a classification such as height, age, error levels, temperature, vehicle speed, position, etc. Linguistic values can vary for the linguistic variables. For example, a linguistic variable for acceleration can take on several values including, large negative, zero, small positive, large positive, etc. The linguistic values can take on a range of real numbers, so the large positive mentioned above could relate to  $[1, 2] \frac{m}{s^2}$  for acceleration [31].

Membership functions of a fuzzy set determines the degree to which a crisp value for a linguistic variable belongs to a certain linguistic value. Membership functions can take on various shapes, however, the more common shapes are trapezoidal, triangular, bell shaped, and sinusoidal. An example of the trapezoidal membership function is shown below in Figure 12 [31].



Figure 12: Fuzzy Logic Trapezoidal Membership Function

The final step in the fuzzy algorithm is the defuzzification process. Defuzzification methods have the purpose of converting the command from a fuzzy to a crisp output that is executed by the control system. For example, the fuzzy output could be in the large positive (LP) membership function region, which could correlate to a physical output of 3 for an actuator. The elements of the fuzzy command matrix are viewed as heights of 'clipped' fuzzy sets as shown below in Figure 13.



Figure 13: Clipped Trapezoidal Membership Function

There are several conversion algorithms that are based on the height of the fuzzy sets and/or the area of the sets. Some of the methods for conversion are:

- Center-of-Sum
- Height-at-Lower-Value
- Height-at-Peak-Value
- Last-of-Maxima

However, the trapezoidal and triangular membership functions, commonly used, contain discontinuities in their derivatives that propagate through the whole process and can result in discontinuities in the command rate. It is possible to have a negative effect on the performance of the control laws and to prevent this, the membership functions can be smoothed out at the edges or a low-pass filter can be executed on the command [31].

Zhang et al. [32] aimed to enhance fuel economy and impose SOC charge-sustainability by adapting the equivalence factor for the ECMS using a fuzzy logic proportional integral (PI) controller. The ECMS was

compared with and without fuzzy adaptation to compare fuel economy results. Constraints were placed so that the battery was forced into a charge sustainability mode, where the ending SOC had to equal the starting SOC. To impose the SOC constraints, the following cost function equation was implemented:

$$J(SoC(t)) = \int_{t_0}^{t} \dot{m}_f(u(t), t) dt + \eta \frac{Q_{max}}{H_{LHV}} \int_{SoC(t_0)}^{SoC(t)} V_{oc} d(1 - SoC) + \beta (SOC_r - SoC(t))^2$$
 1.10

Where  $SOC_r$  is the reference SOC value,  $SOC_r - SoC(t)$  is the delta SOC value during operation,  $\eta$  is the average component efficiency of the motor over the engine,  $\dot{m}_f$  is the equivalent engine fuel consumption,  $H_{LHV}$  is the lower heating value,  $V_{oc}$  is the open circuit voltage, and  $\beta$  is a tunable penalty coefficient. To further impose charge-sustainability, an additional equation can be introduced:

$$s(t) = s_0 + K_p (SoC_r - SoC(t)) + K_i \int_{t_0}^t (SoC_r - SoC(v)) dv$$
 1.11

Where  $s_0$  is the initial equivalence factor and  $K_p$  and  $K_i$  are the proportional and integral coefficients. The key problem in the research done was determining how to adjust the PI coefficients. To design the fuzzy logic controller, 5 linguistic values were used for both the inputs and outputs which ranged from negative to positive. Once the fuzzy controller was designed, simulations were conducted for ECMS with constant EF and the fuzzy PI ECMS. The results confirmed that the fuzzy PI adaptation algorithm was more robust than ECMS with a static equivalence factor in terms of fuel economy, with a 4.44% improvement for a China city bus cycle and 14.7% increase for the Europe (ECE) cycle.

In [33] Wang et al. designed an adaptive ECMS algorithm with fuzzy logic to adjust the equivalence factor based on the deviation of the actual SOC level. The EF was approximated with a tunable penalty factor in the following equation:

$$S(t) = \frac{\eta_{em}}{\eta_{eng}} + 2\delta \frac{H_{lhv}}{E_{bat}(t)} \Delta SOC(t)$$
 1.12

Where  $\eta_{em}$  and  $\eta_{eng}$  are the motor and engine efficiencies,  $\delta$  is the penalty factor and  $\Delta SOC$  is the difference between the current SOC and reference values. The fuzzy controller consists of two inputs and one output. The first input was the SOC deviation and had 7 membership functions, and the second input was the engine rotational speed characterized by 5 membership functions. A practical regenerative braking method was also included to balance the brake torque split to the electric motor and disk brakes. The fuzzy A-ECMS improved the fuel economy from 1.12% to 5.91% for the simulated drive cycles when the initial EFs were set to the optimal EF for ECMS. Under the real drive cycle testing, the fuzzy A-ECMS presented a 0.46% to 3.39% improvement in fuel economy when the initial EF was not adjusted based on prior cycle testing.

Denis et al. [34] proposed a control strategy based on fuzzy logic by feeding the proposed controller with driving condition information. Two inputs for the fuzzy controller were used: a moving average of the past speed and the global discharge rate. The moving average was used to locate the current speed among three speed distributions to use the past driving information to adapt the control logic. The global discharge rate is computed by dividing the difference between the current and targeted SOC with the remaining distance in the cycle left. During the trip, the controller will consider each update as a new driving cycle with a given initial SOC. Three membership functions were used for the moving average and nine functions were used to define the global discharge rate. The designed controller was compared to a DP simulation along with a rule-based strategy. The rule-based strategy showed a fuel consumption of 2.62 L/100 km, the fuzzy-based EMS showed a fuel consumption of 2.01 L/ 100 km and the DP algorithm showed a fuel consumption of 1.89 L/ 100 km. While the results may not be as good as the DP algorithm, the proposed fuzzy-EMS improved fuel economy over the rule-based algorithm and is able to operate online in a real-time environment.

In [35], by Sharkh et al., a fuzzy-rule-based electric-dominant energy management strategy was implemented for a PHEV. The fuzzy controller uses the vehicle power demand and the battery working

state (BWS) to determine the power split for the engine and electric motor. The BWS was introduced to overcome the disadvantage of relying only on the battery SOC for calculations because if over or underestimated, could result in severe damage to the battery pack from under or overcharging. The BWS uses both the SOC and terminal voltage where the SOC is defined with 5 membership functions (low, medium low, medium, medium high, and high) and the termina voltage is defined with 3 membership functions (low, medium, and high). The output for the BWS consists of 5 membership functions (very low, medium low, medium, medium high and very high). As the BWS approaches a defined setpoint for CS, the controller increases the power generated by the engine to keep the SOC at a setpoint. Vehicle power demand and the engine target power are characterized by seven membership functions. Offline simulations were performed to test the fuzzy controller for two different drive cycles: The Urban Dynamometer Driving Schedule (UDDS) and the New European Driving Cycle (NEDC). Results from both cycles showed that the BWS was effective in preventing the battery from over-discharging when the evaluated SOC level was erroneous. The fuzzy EMS also demonstrated the capability of commanding the engine to operate in its most fuel economic region during both drive cycles whenever possible.

In [36] by Zhao et al., a fuzzy ECMS was proposed as an intelligent energy management solution that could operate in real-time. Heavy duty HEVs involved in duty cycles are characterized by frequent start-stop events in off-road applications, and the dynamics were considered when designing the controller. The power split between the engine and motor for the fuzzy sets were characterized by the deviation of the SOC and component efficiencies. The cost function to be minimized is shown below in equation 1.13:

$$J_f(t_f, u, SOC) = \int_{t_0}^{t_f} \dot{m}_f(\tau, u, x) d\tau + \varphi(SOC(t_0), SOC(t_f))$$
 1.13

Where x denotes the engine, motor, and battery states, u represents the defined control variables,  $t_0$  is the initial time,  $t_f$  is the final time, and  $\varphi$  is the penalty function defined for the SOC deviation from the starting value.

The proposed fuzzy ECMS was evaluated under non-road transient cycle and hydraulic excavation transient cycle on a CAT C15 diesel engine model and showed a fuel economy improvement of 4.43 and 6.44% respectively. The fuzzy ECMS showed greater adaptability in tuning the cost factor under different drive cycles with faster dynamics.

Guo et al. [37] designed and implemented a fuzzy logic controller to detect various driving styles. Once drive cycles were classified, the driving styles were decoupled to design the fuzzy controller. The fuzzy logic controller uses the accelerator pedal opening and the accelerator pedal rate of change which reflects the power demand and its variation trend. Both inputs are characterized by 3 linguistic values which include values of small, medium, and big. The output of the controller is the driving style factor which is composed of 4 linguistic values which classify the drive cycle. The values for the output are economy, soft, normal, and aggressive and can take on values from 0 to 1 for the driving style factor. This output is used in tandem with the ECMS along with a hybrid particle swarm optimization-genetic algorithm to optimize the relationship of the driving style from the fuzzy controller to the EF in the ECMS to minimize fuel consumption. The proposed fuzzy logic driving style identification-based control strategy improved the energy economy by 3.69% in the NEDC. The work performed provided guidance for incorporating driving style into PHEV management strategies to help further improve fuel economy.

Wahsh et al. [38] proposed a fuzzy logic based control strategy for a parallel hybrid electric vehicle using the load torque and speed control for efficient vehicle driving. In this study, a fuzzy logic controller was designed and compared to a PID controller to determine which was more effective at controlling the speed of an induction motor. Inputs to the fuzzy logic controller included the speed error and change of

the speed error while the output to the system was the reference speed command. Both inputs and the output included 7 membership functions ranging from large negative to large positive. A city and highway drive cycle were used to validate the PID and fuzzy logic controllers, where the fuzzy controller outperformed the PID due to more robust characteristics when evaluating the control problem. For both drive cycles, the fuzzy controller produced a more stable speed command while obtaining a faster overall response in the motor speed for both drive cycles.

The potential for using a fuzzy logic controller for controlling and calibrating a HEV was explored by Anderson et al. [39]. Competitive control strategies were reviewed, and the fuzzy controller was identified as a strong candidate for reducing overall fuel consumption. A Toyota Prius was selected as the baseline vehicle for validation results due to a well-established system and vast amount of published data. The fuzzy controller was designed with 3 inputs: Vehicle speed, driver demanded torque, and SOC. The initial fuzzy rule base was broken down into 3 different operating modes: Regenerative braking, motor-only, and a combined engine and motor mode. From these operating modes, the initial fuzzy sets were designed using 4 different SOC ranges from very low (0-0.2) to high (0.8-1), where 0 corresponds to 0% and 1 corresponds to 100%. In the initial design, the membership functions did not overlap and were stated as unique sets for each defined SOC operating range. Validation of the fuzzy controller showed a fuel economy of 49 mpg for the UDDS cycle, while real-world reported results showed a fuel economy of 60 mpg. The initial results were promising and showed that the fuzzy controller could produce fuel efficiency similar to a conventional control strategy while maintaining SOC over a broad range of initial conditions. Further optimization yielded fuel economy results of 55 mpg for the same drive cycle, which fell 5 mpg short of the real-world results from the Prius. This was deemed to be acceptable by the authors when considering the simplicity and ease of implementation of the developed fuzzy algorithm when compared to the manpower required for the algorithm used in the Toyota Prius.

A fuzzy logic, rule based control strategy was proposed by Hannoun et al. in [40] for a PHEV. The strategy controlled the amount of energy flow among components to satisfy the driver demand while optimizing energy consumption and reducing polluting emissions. The optimization strategy was summarized to satisfy the conductor demand, reduce fuel consumption, and maintain a SOC between 70 and 95%. From this summary, the fuzzy controller was defined using 3 inputs: The requested power, SOC, and vehicle speed. Inputs were defined by 3 membership functions ranging from negative to high for the power and low to high for the SOC and vehicle speed. Three speed cycles were implemented for validation of the controller to illustrate various operating modes including single component operation (engine or motor) and a power assist mode where the motor is used to augment the engine. During operation, if the power demand is less than 10 kW, the engine is reduced to idle speed and the motor is solely used to provide power. If the power demand increases above 10 kW, the engine is used to provide additional power. Results showed that the conventional vehicle operates well below the optimal curve extracted from the engine efficiency curve while the hybridization model using the fuzzy controller operates at or near the optimal curve, improving fuel economy while maintaining SOC for the imposed constraints over multiple cycles. The fuzzy controller chose the best power split between the engine and motor to considerably improve the efficiency and reduce pollutant emissions from the engine.

Fuzzy logic often requires some form of expert level knowledge of the system to generate membership functions to solve the energy management problem. One disadvantage of the fuzzy approach is that the defined rules are independent of the membership functions and cannot guarantee that the developed system will produce viable results. In the work performed by Danhong et al. [41], a genetic algorithm was introduced to optimize a fuzzy controller implemented in a control algorithm for a PHEV. Inputs to the fuzzy controller consisted of the battery SOC and desired vehicle power while the output was the desired battery power. Three membership functions ranging from low to high characterized the inputs and 5 membership functions characterized the output ranging from negative big to positive big. Once the initial membership functions were defined, the genetic algorithm was implemented to adjust the membership functions as needed. First, an operating range for the GA was defined and an initial population with a predefined number of individuals was selected. The fitness for each chromosome was calculated and the reproduction, crossover, and mutation of the current population was performed to produce the next iteration of the population. These last steps were repeated until predefined criteria were met and both versions of the fuzzy controller were validated using the NEDC driving cycle. The equivalent fuel consumption of the fuzzy controller was 6.8 L/100km while fuel consumption of the genetic-fuzzy controller was 6.5 L/100km. The SOC for the genetic-fuzzy controller was kept higher when comparing results from both controllers and the output engine torque operated in a more stable region while the motor torque was allowed to fluctuate. The genetic-fuzzy controller was able to improve both fuel economy and CS, proving that the addition of the GA further improves the robustness of the controller without having an initial expert level knowledge of the system as a whole.

In addition to the work done in the automotive field for the ECMS strategy, work has also been performed for unmanned aerial vehicles. Xie et al. [42] proposed a method combined the ECMS strategy with fuzzy logic control to formulate the fuzzy-based ECMS (F-ECMS). Normal vehicle operations tend to keep the CS mode of a hybrid around 30% SOC. This CS strategy is not optimal for aerial vehicles because the residual battery capacity cannot guarantee a safe landing is failures occur involving the engine. The F-ECMS solved the deficiencies of ECMS and took into consideration the aircraft landing safely in the event of an engine failure. The final power requirement was used to consider inputs to the fuzzy logic controller where the maximum and optimal power of the engine could be substituted by their difference. This reduced the number of inputs to the controller to 3, also taking the SOC into account. They further reduced the number of membership functions for the engine down to 2 to reduce computational power required and consisted of positive and negative for the engine inputs. The SOC input consisted of 4 membership functions which included low, sustained, high and full. When compared to a combustion propulsion system, the hybrid

propulsion system, integrated with F-ECMS, showed an 11% fuel consumption reduction for designed flight missions. In the second test case, the F-ECMS was shown to solve the issues related to the conventional ECMS where flight missions need to be known *a priori* by operating within prescribed SOC limits. The F-ECMS was also superior to the A-ECMS because there is an associated lower computational cost involved.

The above works illustrate that fuzzy logic controllers appear to be a viable way to achieve a robust solution to complex problems of creating an algorithm to boost fuel economy while maintaining CS operations over different drive cycles or flight missions containing multiple characteristics. The implementation of a fuzzy logic controller requires some expert level knowledge of the system when implementing membership functions into the controller. However, with reduced manpower required to create these controllers, the use of fuzzy logic provides an adaptive way to simplify complex control problems in industry. The work discussed above is not an exhaustive list and extensive work has been done in this area even over just the past decade, however, to list every publication would be redundant. Work presented in this thesis will use two inputs to the fuzzy logic controller to reduce the computational demand with the goal of improving fuel economy, maintaining CS over numerous drive cycles, and extending the HV battery life.

# 3 Methodology

The objective of the current work is to present an adaptive ECMS strategy using a fuzzy logic controller to dynamically update the equivalence factor based on the deviation of the SOC from the target value and the drivers current wheel torque command to the controller. The proposed hypothesis is to improve fuel economy over the base ECMS while maintaining CS operations in a 20% SOC deviation window ranging from 40 to 60%. This section is dedicated to an overview of the MathWorks model used for simulation

and results generation. All results published in this report have been obtained strictly in the MIL environment.

### 3.1 Vehicle Model Overview

The vehicle model is composed of 5 key areas illustrated in Figure 14. The Driver subsystem contains a driver model along with the drive cycle selected for the current simulation. The Controller subsystem contains all student implemented code that is flashed onto real-time hardware for any vehicle testing that is performed. The Plant model contains simulated versions of the engine, HV battery, and EM. The three subsystems are connected through virtual interfaces, or busses, to mimic communication processes that would be observed in vehicle.

The two subsystems on the right-hand side of the figure are the Visualization and Logging subsystems. Scopes are implemented in the Visualization subsystem to observe signal communication and troubleshoot any errors in the model. Any signal that is contained on the *System Signals* bus can be viewed/monitored in the Visualization subsystem. The Logging subsystem follows a similar format to the Visualization subsystem. The Logging subsystem serves to record any important signals in the model including the HV battery SOC, wheel torque commands, component torque production, drive cycle error, etc. A high-level order for a simulation is as follows:

- 1.) The driver model provides an accelerator pedal input to follow the drive trace.
- 2.) The accelerator pedal is read into the Controller subsystem and torque commands are determined for the ICE and EM.
- 3.) The torque commands are sent to the Plant subsystem where the engine and EM models produce torque that is translated to the wheels to move the vehicle.
- 4.) The relative velocity is fed back to the Driver subsystem where the driver model determines whether the accelerator or decelerator pedal should be pressed to follow the drive trace.

5.) Results are recorded in the Logging subsystem for post processing and comparison between different simulations.

Steps 1-4 are repeated until the simulation is complete.



Figure 14: Hybrid Vehicle Model Overview

### 3.2 Driver Model

The Driver subsystem (Figure 15) is composed of 2 critical components: The Longitudinal Driver and the Drive Cycle Selection subsystem. The Longitudinal Driver implements a longitudinal speed-tracking controller and generates normalized acceleration and deceleration commands based on the reference drive cycle and vehicle feedback velocities. Additional parameters can be adjusted to tune the Driver controller, including the driver response time and preview distance. Other notable parameters can be tuned to closer mimic the vehicle being simulated including the vehicle mass, tractive force, and resistance coefficients [43].





The *Drive Cycle Selection* subsystem shown in Figure 16 contains various drive cycles that can be used during a simulation. Each drive cycle shown in the figure below belongs to the 'Drive Cycle Source' mask where any type of cycle can be input. The ones shown in the figure are preloaded, however, excel or text files can be loaded in if the data is available. A multiport switch is also implemented in this subsystem for automated testing. A script which will be discussed in a following section can be programmed to update the *Drive Cycle Selection* constant value shown at the top of the figure. This constant block determines which of the inputs to the multiport switch will be passed along to the Longitudinal Driver. For example, currently the constant block is set to 1, so the NYCC will be the drive cycle used for the simulation. If the constant block is changed to 2, the SC03 drive cycle would be used for the simulation and so on.



Figure 16: Drive Cycle Multiport Switch

# 3.3 Plant Model

The Plant model in Figure 17 consists of 3 main subsystems: The Input layer from the Driver and Controller, the Application layer and the Output layer which organizes the signals onto the 'Plant Signals' bus to be sent out to the system.



Figure 17: Plant Model Overview

The Application layer contains the engine, HV battery, electric motor, and drivetrain as shown in Figure 18. The green inputs to the system are the engine and motor torque commands which are generated from the Controller.



Figure 18: Plant Model Application Layer

# 3.3.1 Engine Model

Figure 19 provides a high-level overview of the engine model. The engine being used for simulations is the GM 2.5L LCV 4-cylinder engine with a maximum torque output of 259 Nm and maximum power output of 151 kW. The engine model is broken down into two separate areas: The spark ignition (SI) Engine Controller and Mapped SI Engine are shown in Figure 20.



### Figure 19: Engine Subsystem

The SI Controller uses the driver torque request in conjunction with the input engine speed to determine fuel, spark, open-loop air and actuator commands that are required to meet the initial torque command. The SI Controller also contains an output 'Info' bus that contains varying signals including the torque and load commands, cam phase angle commands, air flow and estimated torque commands. An additional output from this Powertrain Blockset is the fuel injector pulse-width (InjPw) that can be used along with the torque command and ignition switch to simulation start/stop logic [44]. The Mapped SI Engine uses power, air mass flow, emission performance, exhaust temperature, fuel flow, and efficiency look-up tables to simulate the engine for fuel economy and performance. Transfer functions are integrated into the model to closer mimic actual engine dynamics from smoothing and delay. The outputs of this block include the actual engine torque produced (EngTrq) and an info bus. The info bus contains varying signals, including the fuel flow, engine speed, exhaust manifold gas temperature, BSFC and several emission parameters [45].



Figure 20: Engine Control Module/Mapped Engine Model

The engine model table data for both the Controller and Mapped Engine was derived from steady-state operating conditions on a dyno at a GM facility. Even though the steady-state parameters have been implemented, the assumption can be made that the included parameters are adequate for modeling the dynamic operations of the engine.

#### 3.3.2 Battery Model

The battery modeled in Simulink is the GM HEV4 battery pack with a peak power of 50 kW and a nominal voltage of 300 volts. Figure 21 illustrates the battery model, which is the Datasheet Battery included in the Powertrain Blockset.

Several parameters can be specified for the Datasheet Battery, including the maximum battery charge, open circuit voltage table data (V), internal resistance (Ohms) number of cells in series and parallel, and the initial battery capacity (Ah). The data block uses the ambient temperature and current load as inputs to the system and uses internal resistance and open circuit look-up tables to determine the appropriate voltage, power, and SOC outputs to the system [46].



Figure 21: Powertrain Blockset Battery Model

Battery packs have different limits when discharging and charging the pack that are based on temperature, battery capacity, and current draw. In the vehicle, this is performed by the battery system manager (BSM) and the status of the battery is broadcast over CAN, however, in the model, the BSM must be modeled accurately to ensure that no incorrect loads are placed on the battery when implementing energy management strategies. The BSM model is shown below in Figure 22.


Figure 22: Battery System Manager

The limits for the Discharge and Charge Power subsystems were provided by GM and are given as functions of SOC for varying time intervals. The battery SOC and voltage are read into the Discharge and Charge Power subsystems and pushed through 1 dimensional look-up tables to determine power. The maximum power is determined for 0.1, 2, and 10 seconds for the available charge and discharge power. Figure 23 highlights an example of the Discharge Power subsystem.



Figure 23: Maximum Discharge Calculations

Once the maximum power for each time is determined, it is divided by the voltage to obtain the maximum discharge current available to the system. For simplicity with the student designed controls in the EcoCAR competition, the 2-second maximum is equivalent to the peak available power and/or current. The 10-second maximum power and/or current is considered equivalent to the continuous operating maximum.

The Charge Power subsystem follows a similar pattern; however, the signs are flipped to indicate negative power and current to simulate the battery being charged.

When using the 2-second discharge, a buffer is implemented to ensure the power is reduced from the 2second peak to the 10-second continuous after 2 seconds by implementing a windowed integrator. The logic around the Discharge Buffer subsystem is shown below. The windowed integrator integrates over a specified time (in this case, 2 seconds) window. When the maximum 2-second current is drawn, the buffer initializes with a value of 1 and will vary between 0 and 1. As the maximum current is drawn from 0-2 seconds, the buffer will decrease towards 0 from a value of 1 reducing the maximum available current to the 10-second, or maximum continuous current available. If the battery current is negative, the Buffer system will have a value of 1 for the maximum available 2-second current. The charge buffer is calculated in a similar fashion with the exception of battery current being negative to reflect the battery pack being charged.



Figure 24: Discharge Buffer Calculation

Once the buffer is calculated, the Limiter subsystem (right-hand side of Figure 22) is used to determine the maximum discharge and charge currents and powers, and the maximum battery voltage shown below in Figure 25. The 1-D tables output a gain value ranging from 0 to 1 based on the current value of the SOC. As the SOC approaches the upper or lower limit, the gain will decrease from 1 to 0. In parallel, the maximum allowable charge and discharge currents are multiplied by the buffers determined from Figure 24. These two signals are multiplied together to obtain the maximum allowable charge and discharge currents for the Controller model to use in constraints calculations, discussed later. The maximum charge and discharge currents are also multiplied by the current battery voltage to obtain a power in Watts to simulate actual BSM communication in the vehicle.



Figure 25: Maximum and Minimum Power Calculation

### 3.3.3 Electric Motor Model

Figure 26 highlights the Magna eRAD motor which is modeled in Simulink using the Mapped Motor from the Powertrain Blockset. The 'Mapped Motor Model' uses 2-D look-up tables tabulated with data from the Magna powertrain to mimic vehicle operations. Parameters for the model include rotational speeds, maximum torque values, torque control time constants and speed and torque losses. Tabulated loss data is used to calibrate the motor model due to efficiencies becoming ill defined for zero speed or zero torque. Inputs to the system include the current motor speed from the Driveline and Veh Dynamics subsystem, the torque command from the Controller, and the current battery voltage. The P4 motor current draw is output from the Mapped Motor Model and is determined by adding the mechanical power and power losses from the system and dividing the total by the battery voltage [47]. The motor torque produced is fed into the Torque Limit Calculations subsystem for additional limit determination and is also fed back to the Controller subsystem.



Figure 26: Mapped Motor Model

The motor has different limitations for continuous and peak torque production which is modeled in the *Torque Limit Calculations* subsystem shown below in Figure 27. The calculations follow a similar pattern to the BSM limits where the continuous and peak torque commands are determined using 1-D look-up tables based on the motor speed. These values are integrated using a windowed integrator block to determine a discharge and charge buffer for the system. The peak and continuous signals are fed into the *Max and Min Determination* subsystem where they are multiplied by the respective buffer to determine the maximum and minimum available torque for the system. The minimum available torque is used for regenerative braking events when the motor is used to supply negative torque on the rear axle to decelerate the vehicle. Also included in this subsystem is the *Motor Coupling Dynamics* transfer function that was implemented to further mimic actual torque production in the vehicle.



Figure 27: Motor Torque Limits

# 3.3.4 Vehicle Drivetrain

The Driveline and Veh Dynamics system (Figure 28) contains the Transmission Controller, Torque Converter, Brake Request, Differential, Wheels, Brakes and Vehicle subsystems. For these simulations, the grade, or slope of the road, is not considered along with the wind velocity (WindVel).



Figure 28: Vehicle Drivetrain Overview

The transmission is modeled using Stateflow in Simulink as seen in Figure 29 and accepts the vehicle speed and accelerator pedal command as inputs to the system. Simulink Functions are used to determine the threshold for shifting gears in the model.



Figure 29: Stateflow Shift Map

The Simulink functions use 2-D look-up tables that are tabulated with speed thresholds generated from GM data. The data contained in the look-up tables is confidential; however, an overview of the 'up\_th' function is shown below in Figure 30.



Figure 30: Transmission Up-Shift Threshold

The *Torque Converter Automatic Transmission* system, shown in Figure 31, contains a torque converter (left-hand side), 2 driveshaft compliance blocks, and a fixed gear transmission. All blocks are contained within the Powertrain Blockset in Simulink. The torque converter implements a three-part converter which consists of the impeller, turbine, and stator with an optional clutch lock-up system. The converter can simulate both driving and coasting capabilities. The lock-up type selected for these simulations is the Lock-up, which models automatic clutch engagement.

The *Ideal Fixed Gear Transmission* block [48] implements a transmission without a clutch or synchronization. The block is used to model the overall gear ratio and power loss for the system with minimal parameterization and computational cost. The transmission block was parameterized using data provided by GM for the M3D transmission. The efficiency is also included in the block and is determined from a 4-D look-up table also parameterized with data provided by GM.

The Driveshaft Compliance blocks represent a parallel spring-damper system to couple two rotating driveshafts. Parameters for the blocks include driveshaft angular velocity, torsional stiffness, and torsional damping to determine the torque. The "R" and "C" ports on the block are used for angular velocity in rad/s. The block also implements an output info bus containing torque, speed, and power information; however, for these simulations, the signals are the info bus will not be used.



Figure 31: Torque Converter Automatic Transmission Model

The *Brake Pedal to Total Braking Pressure Request* system determines how much force the brake pads should apply. This system is shown below in Figure 32. A 'Switch' block is implemented to determine how much braking force should be applied. If the input signal 'Park' is ever greater than 0, a constant 1 will be passed meaning that the maximum available braking force will be applied at the wheels to hold the vehicle stationary. In this model, the logic for a parking brake is not modeled, so a grounding block is used to signify that the vehicle does not have the parking brake set. Instead, the brake pedal (DecelPdl) is passed in the form of a decimal from 0 to 1 with a value of 1 signifying that the brake pedal has been applied fully. The decimal value from the Switch block is multiplied by the maximum brake pressure for 4 disk

brakes to determine how much pressure should be applied on the model of the wheels, which will be discussed below.



Figure 32: Brake Pressure Calculation

The *Differential and Compliance* subsystem, shown below in Figure 33, contains an Open Differential block along with Two-way Connection blocks and an additional Axle Compliance block. The Open Differential block implements a differential as a planetary bevel gear train. Parameters for this block can be adjusted, including the carrier-to-driveshaft ratio, crown wheel location, and coefficients for the axles and carries. For these simulations, the crown wheel is located to the left of centerline. The block can be used to couple post-transmission driveshafts to wheel axles and/or universal joints. It can also be used for modeling mechanical power splitting in a generic gearbox and drive line scenarios. The block uses a defined coordinate system that can produce tire and vehicle motion for engine, transmission, and differential configurations. For additional information, a picture of this differential from MathWorks is shown below in Figure 34 [49]. The Axle Compliance block is similar to the Driveshaft Compliance block discussed previously and implements a parallel spring-damper system to couple rotating driveshafts.



Figure 33: Differential and Compliance Subsystem



Figure 34: MathWorks Differential [49]

The *P4 Diff* system (Figure 35) contains another Open Differential block along with an Axle Compliance block. This system is similar to the *Differential and Compliance* system for the engine except the motor is being considered for the rear axle.



Figure 35: Electric Motor Differential

The *Wheels and Brakes* system (Figure 36) contains 2 longitudinal wheels from the Powertrain Blockset calibrated for disc brakes. The longitudinal force for this model is characterized by the Magic Formula constant value, meaning that the block implements the Magic Formula as a specific form of the tire

characteristic function that can be characterized by four dimensionless coefficients, which are the stiffness, shape, peak, and curvature. The coefficients are based on empirical data for the specific tires being modeled. The rolling resistance used is the pressure and velocity, which is a function of tire pressure, normal forces, and velocity. Additional parameters can be adjusted, including the wheel dynamic inertias, loaded and unloaded radii, tire pressure, brake friction coefficients, etc. The brake input signal (BrakeReq) is received from the *Brake Pedal to Total Braking Pressure Request* and split between the front and rear brakes with a 60/40% split. Axle torque inputs from the engine and motor are also received to produce axle torques for both the front and rear axle. These two torques are combined to produce the actual wheel torque produced at the wheels of the vehicle model [50].



Figure 36: Wheels and Brakes

The final system in the *Driveline* model is the *Vehicle* system, shown in Figure 37. This system is composed of a Vehicle Body 1DOF block along with 2 continuous low pass filters and several 2-way connection blocks. The vehicle body block implements a 1 degree-of-freedom (DOF) rigid vehicle body with a constant mass that undergoes longitudinal motion. The block accepts the total longitudinal force on the front and rear axles (FwF and FwR), the grade, and longitudinal wind speed (WindX). It should be noted that for these simulations, the grade and wind velocity are a constant 0. The block also determines and outputs the normal axle forces (FzF and FzR) which are sent out to the system after being pushed through low pass filters. The info bus contains various signals, however, the main signal used in this simulation will be the actual vehicle velocity (xdot), which is an output to the system in Figure 28 [51].



Figure 37: Vehicle Body Dynamics

# 3.4 Controller Model

The *Controller* system (Figure 38) consists of 3 main subsystems: The Input Layer, Hybrid Supervisory Controller, and Output Layer. The Input and Output layers are dedicated to organizing and distributing all relevant signals that the hybrid supervisory controller, or Application Layer, receives and sends out to the system. No calculations are performed in the input and output layers. Their purpose is to organize the signals being received and transmitted to other systems within the model.



Figure 38: Controller Overview

The Application Layer (Figure 39) consists of two main systems: The Accel Pedal and Regen Braking and Energy Management subsystems. The Brake Pedal to Total Braking Pressure Request contains the same logic that was highlighted in Figure 32 and the purpose of this subsystem will be discussed below.



Figure 39: Controller Application Overview

#### 3.4.1 Accel Pedal and Regen Braking

The purpose of the Accel Pedal and Regen Braking system (Figure 40) is to convert the driver's accelerator and decelerator pedal inputs to a wheel torque command that the Controller can use downstream for both a wheel torque and final brake command. This is done through in two separate subsystems. The Series Regen Braking subsystem uses the drivers brake pressure request to determine the total brake torque required to slow the vehicle down. This method of deceleration is known as series braking. The available motor torque is determined at the current vehicle speed and is compared to the total brake torque required. If the motor can provide the total brake torque that the driver is requesting, the friction brakes will not be used to ensure that the motor can recapture available energy through regenerative braking. However, if the brake request is too large, or the HV battery does not have the capacity to store the captured energy, the remaining brake torque will be sent to the friction brakes. In the Mobility Challenge, teams are not able to provide series braking requests as described above. The competition states that teams are only able to provide over-the-top braking in which the deceleration request from the driver must go entirely to the friction brakes. Teams may provide additional regenerative torque from the EM to further slow the vehicle down in a braking event. In this experiment, a series braking configuration is integrated to allow additional energy recapture through the EM during all braking requests to further mimic production-level vehicle control architectures.

The Accel Pedal to Traction Wheel Torque Request subsystem determines the final wheel torque command that is sent to the energy management system. One-dimensional look-up tables are implemented to determine the maximum amount of available torque that the engine and motor can provide at the current vehicle speed and transmission gear. This maximum wheel torque is multiplied by the accelerator pedal which has a range from 0 to 1 with 1 being 100% pedal. If the driver is not on the accelerator pedal, such as a coasting or braking event, the regen torque command will be sent as the final wheel torque command. If the wheel torque command is less than 0, the switch case for *Energy Management Off*, shown in Figure

66

39 is activated. In this subsystem, the motor torque command is equal to the regen torque command to ensure all possible energy is captured by the electric motor.



Figure 40: Accel Pedal and Regen Braking System

# 3.4.2 Energy Management

The *Energy Management* system, shown below in Figure 41, is composed of 3 subsystems: The *Control Domain, Powertrain Constraints,* and *ECMS*. The *ECMS* subsystem outlined with a dashed box contains the FLC designed and implemented for the work presented in this thesis.



Figure 41: Energy Management Overview

The *Control Domain* subsystem generates a predefined vector of possible wheel torque commands for the motor (Figure 42). Pure electric propulsion, no electric propulsion, and a combination of the difference between the two are all considered and pushed into a 'mux' block to combine the 3 inputs into a vector. The output of the mux block is fed into a 'reshape' block to change the dimensions of the vector to output a vector with a size of [1,1].



Figure 42: Motor Wheel Torque Determination Subsystem

The *Powertrain Constraints* subsystem in Figure 43 is used to determine component torques and infeasible operating conditions. At each timestep of the simulation, the limits for each powertrain component are determined. If the commanded torque exceeds a limit, an infeasible flag is set that will affect the torque output downstream. If one of the three infeasible flags are set, the entire torque command is flagged to ensure that no powertrain limits are exceeded during operation.



Figure 43: Powertrain Constraints

There are 3 different infeasible flags that can be set: An engine, motor, or battery infeasible flag. In the *Engine Constraint* subsystem, the motor wheel torque vector is subtracted from the wheel torque vector to determine the engine wheel torque vector possibilities. This vector is converted to a component torque and is compared to the maximum torque that the engine can produce at the current vehicle speed. If the command exceeds the calculated maximum, the infeasible flag will be set. Even if an infeasible flag is set, the complete engine torque vector is still sent to the ECMS algorithm downstream. This subsystem also determines the engine fuel flow rate by interpolating through a 2-dimensional look-up table using the engine speed and possible engine torque commands.

A similar process is followed in the *Motor Constraint* subsystem. The motor speed is read into the subsystem and pushed through a 1-D look-up table to determine the maximum motor torque available at the given speed. An assumption is made that the minimum available motor torque is equal to the maximum motor torque with a sign flip from positive to negative. The motor torque command vector is pushed through a saturation block to ensure that the signal stays within the maximum and minimum available motor torque. If the commanded motor torque is greater than the maximum or less than the minimum available torque, the infeasible motor condition will be set. The total motor power used for each portion of the torque vector is also calculated using an efficiency map supplied by Magna. This power vector is sent into the *Battery Constraint* subsystem.

The *Battery Constraint* subsystem determines the actual current draw and infeasible operating conditions on the HV battery. The actual battery current is calculated by dividing the motor power vector by the actual battery voltage to obtain the current draw on the HV battery. If this current exceeds the maximum or minimum current reported from the Plant model (HV battery in the vehicle), an infeasible flag will be sent downstream. Similarly, the battery SOC is fed into 1 dimensional look-up tables to determine the maximum and minimum available power. The purpose of these tables is to limit the HSC from overcharging/overdrawing power from the HV battery. The output of the tables is a gain value that ranges from 0-1. This final value is compared to the calculated motor power vector and if limits are exceeded, an infeasible flag will be set.

The *ECMS* subsystem, shown in Figure 44, is composed of the equivalence factor and associated SOC penalty calculations, battery power, fuel flow rate, the infeasible flag condition, tractive power error and the engine power delta. It is important to remember that each of the inputs to the summation block are vectors, not scalar values. Each input is composed of a vector determined upstream in the *Control Domain* subsystem. The summation is as follows in equation 3.1:

70

$$H = P_{fuel} + s * P_{batt} * p(SOC) + Con_P + Trac_{PE} + P_{eng,RL}$$
3.1

Where  $P_{fuel}$  is fuel power, s is the equivalence factor,  $P_{batt}$  is the associated battery power, p(SOC) is the multiplicative penalty factor,  $Con_P$  is the infeasible flag constraint factor,  $Trac_{PE}$  is the tractive power error, and  $P_{eng,RL}$  is the engine power delta.



Figure 44: ECMS Overview

The battery power is determined using the HV battery reported voltage and the current calculated in the *Battery Constraint* subsystem. This power is multiplied by the multiplicative penalty factor, which is a function of the battery SOC at each timestep. In this analysis, the penalty factor was adjusted so that from 40-60% SOC, the penalty would be approximately 1. As the SOC increases above 60%, the penalty decreases and as the SOC decreases below 40%, the penalty will increase. Figure 45 illustrates the penalty factor as a function of battery SOC. The maximum and minimum SOC values are 30 and 70% respectively.



Figure 45: Penalty Factor for a = 3

The equivalence factor, or 's', can be a constant value determined offline or a dynamic variable that updates during online operations and is used to relate the total amount of battery energy to power. This value can take on a multitude of ranges, but the range that applies to most drive cycles is between 2 and 4. The fuzzy adaptive controller will be discussed in-depth below in section 3.5.

The engine power delta calculation is performed by multiplying both the current engine torque vector and the previous commanded engine torque by the engine speed to determine power. The purpose of this calculation is to deter the management strategy from oscillating the commanded engine torque. As the commanded engine power differs from the previous power, the associated penalty will continue to increase, making the command more costly. The final engine power delta is the absolute value of the current engine power vector minus the previous engine power command.

The tractive power error penalty is calculated upstream and is implemented to prevent the minimization strategy from selecting a cost that will prevent the total wheel torque command from being obeyed. In a similar fashion, the infeasible flag is multiplied by a gain of 1E7. An extremely high gain is used to prevent the minimization strategy from selecting a cost that would cause damage to one of the powertrain components. The engine fuel flow rate is multiplied by the lower heating value to obtain the fuel power in kilowatts for each element in the commanded engine torque vector.

Once *H* is determined, the MATLAB function block is implemented to find the minimum allowable value of the cost function. This creates an index value 'idx' which is used downstream in the multiport switches to pass the torque command. For example, if the input vector has a length of 10 and the index is 2, then the multiport switches will pass the second vector value for the engine and motor torque commands. In the case where all torque splits are infeasible, the strategy will use the first index which is defined as the total driver demanded wheel torque. Before leaving the controller, a discrete transfer function is applied to the motor torque command to better match engine applied torque and the engine torque command is passed through a unit delay block. This ensures that the last value is used when determining the engine power delta penalty.

## 3.5 Fuzzy Logic Additional Control Block

The focus of this work was to implement a fuzzy logic controller in the management strategy to improve overall fuel economy, charge sustainability of the system, and extend the overall life of the HV battery. The fuzzy controller accepts two inputs: the difference in SOC between the actual and target, and the driver wheel torque command shown below in Figure 46. It should be noted that the wheel torque command and SOC inputs are single values, not vectors. Membership functions were defined for both inputs using the Fuzzy Logic Controller which belongs to the Fuzzy Logic Toolbox in Simulink. This toolbox provides a Simulink block for designing, analyzing, and simulating systems with fuzzy logic. Users are allowed the flexibility to model complex system behaviors using simple logic rules that can be implemented in a fuzzy inference system [52]. The wheel torque command was added to allow the fuzzy controller to make accurate updates to the equivalence factor based on the power demanded from the driver. During low power or torque demands, the controller may determine that the motor can be favored over the engine until the SOC decreases below the setpoint.



Figure 46: Fuzzy Logic Controller

The delta SOC variable shown in Figure 47 was designed to operate in a range of +-10% around the target SOC of 50% satisfying the defined operating window of 40-60% SOC. Trapezoidal membership functions were implemented using uniform distribution to ensure complete coverage of the defined SOC range. Functions include large negative (LN), medium negative (MN), small negative (SN), zero (Z), small positive (SP), medium positive (MP) and large positive (LP). Each membership function has a base coverage of 4% SOC. When the SOC increases during simulation, the membership functions will increase to the LP, indicating that the motor can be favored to reduce fuel consumption from the engine. Similarly, when the SOC is decreasing, the membership functions will decrease towards the LN membership function, indicating that the HV battery is losing charge and the engine should be favored over the motor.



Figure 47: Delta SOC Membership Functions

The wheel torque input (Figure 48) was created to operate in the range of 0-4100 Nm. Negative wheel torque was not considered due to the functionality of the management strategy. When the wheel torque

is not a positive value, the management strategy is disabled to favor regenerative torque from the motor. This torque input is considered the net input to the system and during operation, if the wheel torque command is positive but the HV battery SOC is low, the ECMS will favor an OC scenario to increase the load requested from the engine to provide regenerative torque from the motor. The range for the wheel torque membership functions were defined based on experience working in a VIL environment with the competition vehicle and 2.5L engine. The membership function shape is trapezoidal, similar to the delta SOC functions and can take on values that include very small (VS), small (S), medium small (MS), medium (M), medium large (ML), large (L), and very large (VL). If the torque command exceeds 4100 Nm, the input function will default to VL. The base of each function covers 900 Nm of wheel torque except for the VL function which covers from 3800 to 4100 Nm and above.



Figure 48: Wheel Torque Command Membership Functions

The output membership function shown in Figure 49 is a delta equivalence factor value. A delta was selected in order to help prevent the fuzzy controller from selecting an equivalence factor that would make the battery power too cheap, which could cause the battery to discharge down to a lower limit. The output of the controller is added to a predefined input to ensure that the equivalence factor will always take on an appropriate value, regardless of starting SOC or the selected drive cycle. The base of each membership function has a range of 1 with a total range of -1.5 to 1.5. Definitions of the membership functions are similar to the delta SOC input with LN meaning large negative and LP meaning large positive.



Figure 49: Delta EF Membership Functions

Once the membership functions for both inputs and the output were created, the inference rule matrix was designed. The inference rule matrix includes the 'if-then' rules that dictate how the controller functions. For example, one rule may state that if (Delta\_SOC is SP) and (WhITrqCmd is S) then (Delta\_EF is MN). This would allow the controller to decrease the equivalence factor making the battery and motor cheaper to use. The rules were defined using prior knowledge of possible wheel torque commands and fluctuations in the battery SOC. Varying weights can be assigned to rules to make some rules favored over others, however, for this design, all rules had an equal weight of 1. Once created, Rule Viewer, shown below in Figure 50, was used to double check all rules before implementing the controller in the model. Rule Viewer allows the user to select different input values to determine how the output will be affected. In this example, the delta SOC was defined at an arbitrary value of 7.45% below the setpoint with a small arbitrary torque input of 420 Nm. The resulting output is a small increase in the equivalence factor, meaning that the battery and motor will still be used to provide propulsion. Forty-nine rules in total were defined in the inference rule matrix.



Figure 50: Fuzzy Logic RuleViewer Toolbox

Figure 51 illustrates the surface plot generated from the inference rule matrix for both inputs and the output to the system. As the delta SOC decreases from the setpoint, the inference rule matrix begins to increase the equivalence factor. This increase occurs faster as the wheel torque command increases to ensure that the battery remains within the defined maximum and minimum limits. The complete inference rule matrix is shown below in Table 1. The wheel torque command (WTC) is illustrated in the far-left column, the delta SOC (DSOC) is shown in the top row, and the remaining information creates the output delta EF. For example, if the WTC is very small (VS) and the DSOC is large negative (LN) then the delta EF is medium positive (MP).



Figure 51: Inference Rule Matrix Surface Plot

#### Table 1: Inference Rule Matrix

WTC   DSOC	LN	MN	SN	Z	SP	MP	LP
VS	MP	SP	MN	MN	MN	LN	LN
S	MP	SP	MN	MN	MN	LN	LN
MS	MP	SP	SN	MN	MN	LN	LN
MS	LP	MP	SN	MN	MN	LN	LN
ML	LP	MP	SN	SN	MN	MN	LN
L	LP	LP	Z	Z	SN	MN	LN
VL	LP	LP	SP	Z	SN	MN	LN

Consider a scenario where the input crisp values exist in the area where overlapping occurs in the membership functions. For example, if the delta SOC is 4.25% and the wheel torque command is 350 Nm, the fuzzy set intersection of the input membership functions will be as follows taken from the above table:

Table 2: Fuzzy Set Intersection

WTC   DSOC	SP	MP
VS	MN	LN
S	MN	LN

The above fuzzy set intersection indicates that the fuzzy delta EF output could be a MN or a LN fuzzy value. Next, the implication 'Min' method is applied, meaning that the lowest value from both inputs will be used in the intersection. If the crisp values are applied to the table, the updated intersection can be seen in the table below, where the minimum value from the wheel torque command and delta SOC are selected.

WTC   DSOC	SP - 0.60	MP - 0.45
VS - 0.55	MN - 0.55	LN - 0.45
S - 0.40	MN - 0.40	LN - 0.45

Finally, the fuzzy output is converted to a crisp delta EF output that the ECMS algorithm can use downstream. The defuzzification method selected in this work is the centroid applied to the clipped fuzzy sets. This centroid function will return the center of the area under the aggregate fuzzy set. For this simple

example, the FLC outputs a crisp delta EF of -1.01, reducing the EF and enabling the ECMS to prioritize the motor over the engine.

Figure 52 provides a high-level overview of the closed-loop system in the MIL environment with the integrated FLC, combining subsystems previously discussed in section 3. The Controller is indicated by the blocks shaded in gray, green shaded boxes indicate the Plant model, and the Driver model is shaded in blue. The Longitudinal Driver provides pedal commands to the system using both the accelerator and decelerator (brake) pedal to reduce vehicle speed error from the selected drive trace. These pedal commands are sent to the Controller system, indicated with gray shaded blocks in the figure. A torque request is generated in the Pedal Map system and is sent downstream to the ECMS algorithm. The torque request and SOC are read into the Fuzzy Logic Controller to determine the change in EF that should be applied in the ECMS system. The ECMS minimizes the torque input from upstream in the Controller to optimize the engine and motor using signals discussed in section 3.4.2 along with the dynamic delta EF from the FLC. The torque outputs are sent to the engine and motor to determine actual torque production from both components. From the Engine & Motor system, the motor current draw is routed into the HV Battery system to determine the change in SOC. Simultaneously, the produced torque from the engine and motor is read into the Vehicle Dynamics system where the vehicle speed is adjusted. The updated vehicle speed is received by the Longitudinal Driver and the process is repeated until the drive cycle is complete.



Figure 52: High-Level Control with Integrated Fuzzy Logic Controller

# 4 Test Procedure

The testing methodology is broken down into 3 parts: Optimal, average, and fuzzy ECMS. The optimal ECMS is defined as the absolute best fuel economy that can be achieved while maintaining CS over each drive cycle. The optimal is found through a brute force analysis of the EF discussed below. The average ECMS uses the average EF (sum of all EF divided by the total number of drive cycles) found during the brute force analysis to determine how effective the ECMS can be over multiple drive cycles when the EF remains fixed. The fuzzy ECMS uses the same initial EF for each drive cycle and the additional FLC updates the EF based on the input SOC and wheel torque command from the driver.

Table 4 illustrates the experimental factors, levels applied on the factors, and the effects on the control strategy downstream to highlight the experimental design process. Each test consisted of changing one of three factors to analyze the effects on the control strategy downstream. For example, the first factor held constant in a test could be the NYCC drive cycle. The second factor held constant can be an initial SOC value of 50%. The driving style, or level, of the driver is selected as the modified factor to simulate both an aggressive and mild driver. Once the test is complete, the process is repeated with a different driving style while the drive cycle and initial SOC remain constant.

Each factor and their respective level have a cause-and-effect relationship with the control strategy downstream. If the driving style is modified, the wheel torque demand from the driver will fluctuate. Aggressive drivers have more oscillation in the demanded torque to follow a drive trace with minimal error while the mild driver will produce less oscillation and smooth the drive trace out with minor errors. Similarly, when the drive cycle is modified, the characteristics in the cycle will change. These characteristics are discussed below in Table 5 and include factors such as maximum vehicle speeds and accelerations. The initial SOC has a direct correlation with the cost of electrical energy in the ECMS. When a lower initial SOC is used, the cost of electrical energy will increase, causing the ECMS to prioritize the engine over the motor until the current SOC approaches the target SOC of 50%. It should be noted that the optimal results discussed below only used an initial SOC of 50% and an average driving style combining the aggressive and mild drivers.

Factor	Level	Effect	
Driver	Aggressive	Change in Wheel	
Driver	Mild	Torque Demand	
	NYCC		
	SC03		
	US06		
	EMC City	Change in driving	
	RTS95		
Drive	HWFET	change in unving	
Cycle	HUDDS	Style	
	EMC Highway	characteristics	
	UDDS		
	LA92		
	Artemis Rural		
	Road		
500	50 %	Change in cost of	
SUC	32.5 %	electrical energy	

Table 4: Experimental Factors and Respective Levels

Drive cycles were selected that covered an extensive range of vehicle dynamic possibilities, including moderate and hard accelerations, simulated city and highway speeds, and varying engine idle times. A total of 8 drive cycles were used in the optimal ECMS method to find the best performing EF, however, 3

additional drive cycles were used for further validation of the fuzzy ECMS strategy. The implemented drive cycles and their respective characteristics used for testing and validation are shown in Table 5. Drive cycles were repeated twice in each simulation to allow the fuel economy to balance out when starting at 0 mpg at the beginning of each simulation. SOC CS constraints were imposed to prevent the ECMS from deviating too far from the initial target SOC of 50%. When each cycle has finished running, the test was deemed CS if the ending SOC was within the range of +- 5% of the starting SOC. Additional loose constraints define an appropriate operating region of 40-60% SOC to ensure longer battery life and prevent the ECMS algorithm from using the motor for propulsion too heavily causing the HV battery to deplete to critical levels of less than 20% SOC.

	Simulation	Max Speed	Avg. Speed	Max Acceleration	Min Acceleration
Drive Cycle	Time (s)	(m/s)	(m/s)	(m/s²)	(m/s²)
NYCC	1196	12.38	3.17	2.68	-2.64
SC03	1200	24.50	9.60	2.28	-2.73
US06	1200	35.90	21.46	3.76	-3.08
EMC City	1478	31.75	11.70	3.63	-3.41
RTS95	1772	37.35	14.58	2.88	-2.71
HWFET	1530	26.78	21.56	1.43	-1.48
HUDDS	2120	25.93	8.43	1.96	-2.07
EMC Highway	5924	35.90	25.91	3.56	-2.89
UDDS	2738	25.35	8.76	1.48	-1.48
LA92	2870	30.04	11.00	3.08	-3.93
Artemis Rural Road	2164	30.97	15.96	2.36	-4.08

Table 5: Validation Drive Cycles and Characteristics

### 4.1 Optimal ECMS

An optimal equivalence factor needed to be determined for each drive cycle to obtain optimal fuel economy during a CS drive. This was done by performing a brute force analysis of the vehicle model. A brute force analysis was selected to ensure that the EF found was a global and not a local optimum. First,

the vehicle model was prepared for simulations that could be run from a script in MATLAB. Running the model from a MATLAB script provides several benefits including reduced simulation downtime to manually reset variables after each cycle along with reduced data files which improves the data reduction process. Preparations in the vehicle model included configuring the model to log certain signals along with replacing constant blocks with calibratable variables. For example, Figure 53 shows a calibratable variable (CurrDriveCycle) for the selected drive cycle. Before starting simulations, the active drive cycle passing through the multiport switch can be adjusted by updating a variable saved in the base workspace. Figures for each drive cycle used in the brute force discussed in the results section will be highlighted, and a figure of each cycle is also shown below in Appendix A. The MATLAB script used in the brute force analysis is shown below in Appendix B. The figure below illustrates the cyclic functionality of drive cycles in Simulink. Each drive cycle will repeat until the specified duration is complete to allow the SOC to stabilize over the duration of each test to allow better representation of average performance and reduce bias from the initial SOC.



Figure 53: Calibratable Variable

The brute force analysis included running each drive cycle for varying equivalence factors with a 'for' loop in MATLAB. The selected range for the tested equivalence factor values spanned from 2 to 4 in 0.05 increments. As discussed above, the optimum EF should be very low to ensure the motor is highly always favored, however, this would result in very poor charge sustainability. Constraints were put in place to prevent the ECMS from using an EF set too low, so charge sustainability was defined in a 5% window from the starting SOC. After each run, the battery SOC and fuel economy were loaded into the base workspace for data reduction. An 'if' loop was created to determine if the current EF provided adequate CS. If the ending SOC was less than 45% or greater than 55%, the EF was determined to be inadequate, and the ending fuel economy was set to 0 mpg. Once the brute force analysis was complete, the maximum fuel economy for each drive cycle with a constant EF was found. This fuel economy was determined using a SOC charge correction question to determine the equivalent fuel consumed or stored by the HV battery. By setting the non-CS drive cycles fuel economy variable to 0, the maximum fuel economy that provided CS results was found using a max function for each of the 8 drive cycles. The total fuel economy and corresponding EF for each of these drive cycles is shown below in section 5.

## 4.2 Average ECMS

Once the best EF was found for each drive cycle that maximized fuel economy and CS, the average ECMS could be run. From the first 8 drive cycles and EFs shown in Table 6, the average EF was determined to be 2.65. This average value was added to a new script which iterated over each drive cycle again, however, in post processing, the fuel economy was not reset if the ending SOC was outside of the defined CS range. In the average ECMS procedure, CS is still defined as +- 5% from the original initial SOC of 50%.

Drive Cycle	Equivalence Factor
NYCC	3.5
SC 03	2.45
US 06	3.05
EMC City	2.75
RTS 95	2.85
HWFET	2.75
HUDDS	2.95
EMC Highway	3.05
Average	2.92

Table 6: Optimal Equivalence Factors

The EF can vary greatly when changing between city and highway drive cycles so 3 additional drive cycles, the UDDS, LA92, and Artemis Rural Road were introduced with varying drive cycle characteristics to further validate the effectiveness of the average ECMS algorithm.

Each drive cycle was repeated 4 times to capture multiple iterations of data in varying scenarios. A starting SOC of 50% and 32.5% were defined as well as different driver reaction times of 0.1 and 0.5 seconds respectively. The difference in starting SOC provides a varying approach to determining how well the average EF can obtain CS results when the average EF is found for a starting SOC of 50%. Driving styles differ from person to person and terrain including elevation and starting/ending location. For example, in a real-world environment, the vehicle may be powered down after a large regenerative event down a hill or a hard acceleration up a hill, causing the ending SOC to differ greatly. A starting value 50% SOC was selected to better represent the average between different scenarios and the effect they have on the starting SOC when turning the vehicle on. A starting SOC value of 32.5% was chosen to represent a scenario where the starting SOC is closer to the defined lower operating limit of 30% SOC. This scenario is used to show how the control system will react when starting a simulation from a very low SOC. Adjusting the driver response time in the Longitudinal Driver block also provides additional flexibility of providing a

mild and aggressive driver. The Aggressive driver with a response time of 0.1 seconds will attempt to follow the drive cycle as close as possible, minimizing deviations while the mild driver with a response time of 0.5 seconds tends to smooth out the drive trace and allow for a margin of error when performing accelerations and decelerations. Figure 54 illustrates the difference between the aggressive and mild driving styles. The drive cycle is illustrated with a black line, the aggressive driver with a red line, and the mild driver with a green line. As seen in the figure, the aggressive driver follows the drive cycle almost perfectly while the mild driver deviates during both accelerations and decelerations to help mimic different and real-world driving styles.



Figure 54: Driver Style Comparison

### 4.3 Fuzzy ECMS

The fuzzy logic controller output membership functions were defined using data collected in the brute force EF testing. Results showed that the model performed best when operating in an EF range of 2-3, however, there were certain instances where the SOC would deviate from the defined operating range of 40-60% SOC. Due to these deviations, the output delta SOC function was defined to have a range of -1.5 to 1.5 to allow the EF to increase above 3 if the engine was needed to provide OC and decrease below 2 after large energy capturing instances. As discussed above, the EF is not based solely on the output of the fuzzy logic controller, but rather the output is added to the original EF defined before simulations begin. Creating a delta output rather than an absolute output provides the controller with added flexibility of adjusting the EF without losing resolution by expanding or reducing the total range of the output. A similar process was followed as discussed above in section 4.2 where the starting SOC and driver styles were varied to test the robustness of the F-ECMS strategy. A separate MATAB script was created to run the model to save computational time and record results from each drive cycle in separate structures in MATLAB for additional post processing.

# 5 Results

The results from the optimal, average, and F-ECMS will be discussed in this section. Results pertaining to the brute force analysis conducted to obtain the optimal EF for each of the first 8 drive cycles will be discussed in section 5.1. A comparison analysis will be discussed in section 5.2 involving the optimal, average, and F-ECMS algorithms to validate the effectiveness of the F-ECMS algorithm for both fuel economy and CS improvements. Section 5.3 will highlight results obtained when evaluating the F-ECMS algorithm for different driver reaction times along with 2 different starting values for the SOC. Section 5.4 will discuss results from comparing the F-ECMS to the DP algorithm performed by Aaron Mull of WVU.

### 5.1 Brute Force Analysis

Table 7 below shows the fuel economy and SOC results for the brute force analysis discussed in section 4.1. From these results, the absolute best fuel economy for each drive cycle using the imposed SOC CS constraints can be seen. These results provide us with the baseline for all future testing using the current vehicle model configuration. While these results may not accurately represent other work conducted with a similar EcoCAR Blazer model for a GM 2.5L LCV engine paired with a Magna eAWD motor, they do serve as an accurate benchmark for the work being conducted in this paper.

From the table below, drive cycles with mild accelerations and decelerations allow for more robust energy management from both powertrain components. This in turn results in lower costs to use the electric motor which correlates to a lower EF factor. Drive cycles characterized by harder accelerations and decelerations with lower top speeds such as the NYCC correlate to increased penalties associated with the electric motor due to poor energy recapture, resulting in a higher EF. However, during operation, the ECMS may decide that it is more cost effective to run the engine harder in order to generate electrical energy to charge the HV battery pack. These findings will be discussed below. It should be noted that the fuel economy results are calculated based on engine fuel flow, output from the SI Mapped Engine Blockset discussed in section 3.3.1 in kg/s. The fuel flow is converted from kilograms per second to the U.S. gallon equivalent. This fuel flow is divided from the total distance traveled in miles to obtain the engine mpg equivalent. Once a cycle is complete, a charge correction is applied to adjust the final fuel economy based on the ending SOC of the HV battery. In CS operations, the fuel economy may be adjusted if the ending SOC is different than the starting SOC, indicating that the drive may have been charge depleting (ending SOC less than starting). The difference in the SOC value is considered in the charge correction equation, which defines the correlation between equivalent electric gasoline used or stored and actual fuel flow from the engine model. If the SOC is less than the starting SOC, the electric equivalent fuel consumed will be a positive number. However, if the HV battery is charged over the course of the cycle, energy will be left over for the next test, improving the overall fuel economy for the drive. Charge correction has been applied to the fuel economy results shown in the table below.

Drive Cycle	Equivalence Factor	Ending SOC (%)	Fuel Economy (mpg)
NYCC	3.5	47.41	19.85
SC 03	2.45	52.27	36.40
US 06	3.05	54.95	30.55
EMC City	2.75	53.78	33.15
RTS 95	2.85	50.23	26.99
HWFET	2.75	52.22	41.19
HUDDS	2.95	54.92	35.66
EMC Highway	3.05	49.56	30.22

Table 7: Optimal ECMS Results

## 5.2 Fuel Economy Comparison

Table 8 provides the charge corrected fuel economy results from the average and F-ECMS for both driving styles with varying initial SOC values. The table is broken down into 4 main sections. The first row contains the different driving style and initial SOC value. The second column contains the drive cycle header and type of EF applied. The average EF is represented with an 'A-EF' and the fuzzy EF is represented with an 'F-EF'. Drive cycles used in the experiment are listed on the left-hand side of the table in the first column, and fuel economy data in units of mpg is displayed in the remainder of the table. The fuzzy EF competes with the average EF for each driver style and starting SOC value, improving fuel economy over the course of several drive cycles.
	Aggres	sive 50%	Mild	50%	Aggress	ive 32.5%	Mild	32.5%
Drive Cycle	A-EF	F-EF	A-EF	F-EF	A-EF	F-EF	A-EF	F-EF
NYCC	18.69	19.46	19.18	19.92	19.55	21.47	19.92	21.85
SC 03	36.12	34.14	36.25	34.33	38.26	35.98	38.19	36.03
US 06	27.72	27.94	28.60	28.83	28.27	28.12	29.22	28.92
EMC City	31.41	30.68	31.85	30.94	32.51	31.65	32.74	31.88
RTS 95	26.88	26.78	25.97	27.49	27.39	27.21	28.07	27.89
HWFET	39.46	38.81	40.52	39.63	40.22	39.49	41.01	40.93
HUDDS	32.08	31.55	33.58	32.76	33.05	32.58	34.65	33.81
EMC Highway	30.06	30.59	30.07	31.54	30.63	30.69	31.41	31.63
UDDS	33.65	33.67	34.00	34.33	34.60	34.72	34.65	34.94
LA 92	31.48	30.93	32.00	31.51	32.08	31.48	32.54	32.03
Artemis Rural Road	37.30	36.69	38.26	37.23	38.02	38.99	37.41	30.05

Table 8: Fuel Economy Results in mpg for Average and Fuzzy EF for Aggressive and Mild Drivers at 50% and 32.5% Initial SOC

Equation 5.1 is used to determine the percent difference between the average and fuzzy EF strategies.

$$\%_{Diff} = \frac{FE_F - FE_{Avg}}{FE_{Avg}} * 100$$
5.1

Where  $FE_{Avg}$  is the charge corrected fuel economy from the average ECMS and  $FE_F$  is the charge corrected fuel economy from the F-ECMS. Table 9 provides the total percentage difference for the fuel economy over each drive cycle for the average and optimal ECMS compared to the F-ECMS. Positive percentages represent an increase in fuel economy for the F-ECMS while negative percentages highlight a decrease in fuel economy over the average or optimal ECMS. The cells containing improvements in fuel economy are shaded with a light green background.

Please note, the optimal EF was only found for a starting SOC of 50%. For the tests comparing the optimal to the F-ECMS algorithms, the 32.5% SOC tests are used for comparing the hypothetic possibility of using an adaptive equivalence factor to highlight differences in fuel economy. For an accurate comparison, the brute force analysis would need to be repeated with an initial SOC of 32.5% and a target of 50%. As

mentioned earlier, the fuel economy for the NYCC cycle is substantially lower due to the aggressive accelerations and decelerations at low speeds, reducing the amount of energy recapture over the course of the drive. For every other cycle, the F-ECMS can maintain a percentage difference of an average 4% decrease, except for an 11.53% decrease in the HUDDS cycle when compared to the optimal ECMS. However, the F-ECMS was able to increase fuel economy substantially on multiple cycles, including the NYCC and EMC highway cycles when compared to the average ECMS. The drive cycles that showed improvement have similar characteristics. The ECMS algorithm can use the motor for launching events due to mild accelerations and is able to recapture energy through regenerative braking during longer, more mild decelerations. Mild decelerations also mean the engine can be pushed into fuel cut off (FCO) more often, saving additional fuel. Through the adjustment of the EF, the ECMS selects better operating regions for the engine to improve fuel economy and reduce emissions.

The F-ECMS does not perform as well when looking at the percent difference compared to the optimal ECMS. In the table shown below, the F-ECMS performs better than the optimal ECMS roughly 25% of the time. The largest error occurs during the HUDDS cycle and has a total error of 11.53%. The best improvement was 4.37% during the EMC Highway cycle when only considering the 50% starting SOC due to the brute force analysis only being completed for an initial SOC of 50%. The average decrease was 1.62% when looking at the drive cycles where the F-ECMS performed worse when compared to the average EF. In the case of the mild driver when starting at 50% SOC, the F-ECMS was able to improve fuel economy in 3 out of the 8 original cycles that the optimal ECMS was calibrated for. The lowest improvement was 0.35% while the highest was 4.37%. The mild driver provides a better representation of human drivers in a real-world environment due to the natural tendency of deviating away from the defined drive trace. The adjustment in the EF allowed the ECMS algorithm to better optimize the engine by utilizing the motor during launching events and the engine at higher speeds achieved in cycles such as the EMC Highway cycle. A better example of this can be seen from the mild driver with a starting SOC of

91

50% where the F-ECMS was able to improve fuel economy on the EMC Highway cycle by 4.37%. Reasons for the improvement in fuel economy come from setting the EF relatively high during early parts of the drive cycle. This penalized the motor which forced the ECMS to use the engine to provide OP and recharge the HV battery. During these events, the engine was optimized to operate in a more efficient BSFC region to provide better fuel economy while increasing the SOC of the HV battery pack to obtain CS results shown below in section 5.3

However, it is important to remember that the F-ECMS substantially reduces the computational effort required to use the ECMS in a real-world environment when obtaining an optimal or an average EF for one or multiple drive cycles or different road conditions. The F-ECMS allows for the design of a robust algorithm that can be competitive over numerous drive cycles with reduced computational effort required to tune the algorithm to one or a few drive cycles with varying characteristics.

	Average Comparison			Optimal Comparison				
Drive Cycle	Aggressive 50%	Mild 50%	Aggressive 32.5%	Mild 32.5%	Aggressive 50%	Mild 50%	Aggressive 32.5%	Mild 32.5%
NYCC	4.12	3.86	9.82	9.69	-1.96	0.35	8.16	10.08
SC 03	-5.48	-5.30	-5.96	-5.66	-6.21	-5.69	-1.15	-1.02
US 06	0.79	0.80	-0.53	-1.02	-8.54	-5.63	-7.95	-5.34
EMC City	-2.32	-2.86	-3.54	-2.62	-7.45	-6.67	-4.52	-3.83
RTS 95	-0.37	5.85	-0.65	-0.64	-0.78	1.85	0.82	3.33
HWFET	-1.64	-2.19	-1.82	-0.19	-5.78	-3.79	-4.13	-0.63
HUDDS	-1.65	-2.44	-1.42	-2.42	-11.53	-8.13	-8.64	-5.19
EMC Highway	1.76	4.89	0.20	0.70	1.22	4.37	1.56	4.67
UDDS	0.06	0.97	0.35	0.84				
LA 92	-1.75	-1.53	-1.87	-1.57				
Artemis Rural Road	-1.64	-2.69	2.55	1.71				

Table 9: Total Fuel Economy % Comparison

The above tables provide a substantial overview of the F-ECMS and how effective it can be at providing CS results and improving fuel economy by the end of each drive cycle; however, this is only a brief snapshot highlighting the results of each drive cycle. What happens throughout each simulation? Does

the SOC remain in a smaller CS window, or does it deviate outside of the defined operation region discussed above in section 4, reducing the overall life of the HV battery and related systems?

### 5.3 SOC Comparison

Table 10 provides the ending SOC results from the average ECMS strategy along with the F-ECMS for both driving styles with varying initial SOC. The first row of both tables groups the tests into 4 categories which pertain to the driver type (aggressive or mild) and starting SOC value (50% or 32.5%). In the second row, the average ECMS and F-ECMS are denoted with an A and F respectively.

In Table 10 we can see that the average EF keeps the control strategy operating in a CS mode roughly 73% of the time. For the selected drive cycles, an average EF is satisfactory for maintaining the SOC, prolonging HV battery life, and demonstrating that a constant EF performs well under certain circumstances. During simulations, the cycles that fail pertain more to the aggressive driver, correlating to more frequent oscillations in the torque command to keep the vehicle speed as close to the drive trace as possible. The drive cycles that fail also share common characteristics: Higher speeds, aggressive accelerations, and reduced idle times. In comparison, the F-ECMS shows a success rate of 95% when looking at the ending SOC in comparison to the defined ending range of +/- 5%. In the F-ECMS, the failures occurred during the UDDS cycle with an initial SOC of 32.5% and the mild 50% SOC NYCC test. However, the largest gap between the lower defined operating range and lowest ending SOC value is 0.78% which corresponds to a maximum error of roughly 1.5% when looking at every drive cycle for different driving styles.

	Aggress	ive 50%	0% Mild 50%		Aggressive 32.5%		Mild 32.5%	
Drive Cycle	A-EF	F-EF	A-EF	F-EF	A-EF	F-EF	A-EF	F-EF
NYCC	38.12	44.22	39.63	46.41	31.61	46.27	30.74	46.40
SC 03	61.61	46.15	60.61	45.75	61.60	46.13	56.57	45.74
US 06	52.10	49.74	58.93	50.27	52.25	49.71	58.33	50.27
EMC City	61.70	50.13	65.07	50.17	62.56	50.18	65.00	50.19
RTS 95	50.74	52.55	49.78	53.67	51.24	52.57	49.76	53.67
HWFET	62.82	49.77	65.30	50.24	62.65	49.79	65.15	50.24
HUDDS	51.95	47.55	59.01	47.83	50.92	47.54	60.40	47.85
EMC Highway	47.69	51.84	52.07	51.18	47.63	51.84	52.06	51.18
UDDS	43.23	45.73	40.64	45.20	44.73	45.71	38.60	44.92
LA 92	63.13	48.25	62.57	48.65	63.31	48.24	62.58	48.61
Artemis Rural Road	56.31	46.35	63.61	46.79	56.34	46.33	64.08	49.18

Table 10: Average and Fuzzy ECMS Ending SOC Comparison

Several drive cycles will be highlighted to showcase the results of the average and fuzzy ECMS methods. The drive cycles that were used for validation are presented in Appendix A and results not discussed in the following paragraphs are presented in Appendix C.

Why is the trend of SOC over time an indication of good EF management? The EF will affect how the ECMS optimizes torque between the engine and motor. An EF that is set very low may improve fuel economy by utilizing the motor more often, however, this may cause the SOC to deplete to critical levels. During operation, if the SOC drops to an extreme limit, the contactors in the battery pack may open under load, damaging the pack by pitting or welding the contactors shut, rendering the pack unusable and dangerous. Adjusting the EF allows the HV battery and motor to operate in constrained but efficient cases to keep the SOC within predefined limits, regardless of the driving style or cycle. A robust EF management strategy will allow the SOC to deviate away from the target SOC without violating predefined limits. Different driving styles also influence the fluctuations in SOC over different drive cycles with varying top speeds and accelerations. The trend in SOC over each drive cycle can help to pinpoint errors in a constant or average

EF rather than a dynamic EF. In the following sections, the EF for several drive cycles will be presented to observe how adjusting the EF prevents the SOC from approaching an extreme limit.

### 5.3.1 NYCC Cycle

The F-ECMS achieved a maximum fuel economy improvement of 0.35% when compared to the optimal ECMS with a starting SOC of 50% while simultaneously increasing fuel economy by a maximum of 9.82% when compared to the average ECMS. However, CS characteristics need to be considered when evaluating the effectiveness of the implemented FLC. Figure 55 displays the NYCC cycle for one iteration. It should be noted that drive cycles were ran 2 times in succession to validate fuel economy and CS SOC results and reduce bias from the initial SOC.





Figure 56 provides the SOC results in the NYCC cycle for the optimal, average, and F-ECMS strategies for a starting SOC value of 50%. The optimal results were obtained from the brute force analysis; however, the brute force analysis was designed to obtain the best fuel economy while maintaining an ending CS window of +- 5% when starting with an initial SOC of 50%. If the initial SOC is modified, the brute analysis will need to be rerun to determine the new EF for CS results. The NYCC cycle contains hard accelerations with reduced idle periods, and the EF was determined to be relatively high at 3.5. The average EF of 2.92 is substantially lower, meaning that the electrical to fuel conversion for the motor is cheaper over the course of the entire drive cycle. A lower equivalence value corresponds to lower energy management which resulted in non-charge sustaining operations for this cycle. However, other cycles in the MIL or VIL environment may show increased benefits from using the average EF determined in this thesis. The average ECMS also violated the 40-60% operating region constraints that were placed for simulations at the end of each cycle. Conversely, the F-ECMS method was able to maintain an average SOC value of 45% and was charge sustaining at the end of the cycle for both the aggressive and mild driver, indicating that the ending SOC fell within the prescribed bounds for the experiment. It should be noted that in the following SOC comparison figures, the 40-60% constraint operating window is indicated with a dashed horizontal blue line.



Figure 56: NYCC 50% SOC Comparison

Figure 57 illustrates the SOC and EF for both drivers along with the average EF for comparison. The SOC correlates to the left y-axis and the EF correlates to the right y-axis. It should be noted that for the EF comparison figures, the average EF of 2.92 is shown with a dashed horizontal line. In the figure, the EF begins relatively low allowing usage of the motor until the SOC crosses the 45% point. At that time during the drive cycle, the fuzzy logic controller begins to adjust the EF based on the deviation of the SOC and the drivers wheel torque command. When the SOC approaches 45% from a lower value, the EF is adjusted to favor the motor in the energy management strategy consistently throughout the cycle. The driver is more demanding for the aggressive style, which correlates to increased fluctuations in the EF. Due to aggressive accelerations in the cycle, the EF averages roughly 3.0 for the duration of the drive.



Figure 57: NYCC 50% EF Comparison

Figure 58 and Figure 59 illustrate the NYCC cycle when the starting SOC is at an extreme low point of 32.5%. During normal operations, an SOC value this low should be prevented to ensure proper functionality of the HV battery, however, an extreme case was considered to test the functionality of both the average and F-ECMS. In Figure 58, a similar process is followed where the F-ECMS can recharge the battery and obtain CS operations over the course of the drive cycle. The average ECMS, with a lower than optimal EF, is not able to maintain a balanced SOC which resulted in undesirable performance. It should be noted that the optimal EF ECMS SOC is also plotted in this figure. The optimal SOC provides a guideline for where the SOC should be operating during the simulation if the optimum EF is applied at a starting SOC of 50%. This guideline further shows the robustness of the F-ECMS algorithm to maintain proper SOC balancing, regardless of initial SOC. The F-ECMS adjusts the EF to increase the SOC over the drive cycle to maintain CS operation within the first 250 seconds.



Figure 58: NYCC 32.5% SOC Comparison

Similarly, in Figure 59, the EF factor adjusts at the beginning to roughly 3.8 for both driving styles and remains high until the SOC begins to approach 45%. At this point, the EF decreases slightly to favor the motor but remains higher than the average due to the characteristics of the drive cycle combined with a very low starting SOC. Regardless of the driver style, the F-ECMS can maintain roughly the same value for the SOC at each instance in the drive cycle.



Figure 59: NYCC 32.5% EF Comparison

# 5.3.2 US06 Cycle

The F-ECMS was not able to increase fuel economy when compared to the optimal ECMS, however, an improvement of 0.8% was found when comparing the average to F-ECMS with a mild driving style. The US06 drive cycle is presented below in Figure 60.



#### Figure 60: US06 Drive Cycle

Figure 61 demonstrates the improvement of the F-ECMS over both the average and optimal ECMS when looking at SOC for the US06 drive cycle. In the figure, while the optimal ECMS was CS, it produced poor SOC optimization during the drive cycle and led to the upper 60% constraint to be violated over the course of the simulation. A violation in the upper limit may not cause substantial differences in the overall life of the HV battery, however, goal of this work is to minimize the amount of time spent charging and discharging over a defined SOC limit to improve the overall HV battery life in the VIL environment. Consider the case where the SOC can deviate below the lower operating region of 40%. This may result in increased fuel economy by utilizing the motor, however, the SOC may continue to decrease to a critical level and may not recover fast enough. If the SOC reaches critical levels around 20%, the battery pack may open contactors, causing severe damage to the battery pack. Similar results can be seen for the average ECMS with the mild driver where the upper limit is crossed multiple times before the battery is discharged. The F-ECMS can keep the SOC balanced around 45-50% for the driver cycle regardless of the driving style used and is able to better optimize usage of the electric motor. It should be noted that the average ECMS in the mild driver simulation optimized the engine by providing more torque than the driver requested.

This optimization is due in part to the ECMS favoring the engine during higher vehicle speeds when the engine is more efficient. A lower torque demand paired with higher vehicle speeds results in a lower efficiency for the IC engine which consequently led to the ECMS commanding additional torque from the engine to reduce fuel flow. As a result, the ECMS produced larger amounts of opportunity charging throughout the simulation to maintain a higher-than-average SOC for the HV battery.



Figure 61: US06 50% SOC Comparison

Figure 62 provides the SOC and EF plots for the aggressive and mild driver with a starting SOC of 50% for the US06 cycle when the F-ECMS was implemented. During this drive cycle, the motor is able to recapture energy through regenerative braking to charge the HV battery, which resulted in an EF lower than the average in multiple instances of the drive cycle, including the times from 500-600 and 1050-1150 seconds respectively for the aggressive driver. From the figure, as the SOC decreases below 45%, the EF increases to compensative and make the motor more expensive to use. Similarly, as the HV battery is recharged,

102

the EF adjusts and lowers to gain more usage of the electric motor during vehicle launching operations where the engine is less efficient. For example, from roughly 500 to 600 seconds in the aggressive driver simulation, the EF is reduced to allow the motor to help launch the vehicle during the lower speed accelerations. Once the SOC drops below 45%, the EF increases to help raise the cost of using the motor, preventing the HV battery from being discharged too quickly.



Figure 62: US06 50% EF Comparison

Figure 63 and Figure 64 show the SOC and EF variation for the US06 drive cycle with a starting SOC of 32.5%. It should be noted that the optimal SOC curve for the US06 with an initial SOC of 50% is also shown in the figure to serve as a guideline for where the SOC would operate when using the optimum EF, similar to the case shown in the NYCC section. The SOC for the aggressive driver follows an ideal case where the SOC can fluctuate between 40 and 60% throughout the cycle. However, the average ECMS demonstrates a similar pattern to the results discussed above where the ECMS opted to run the engine harder for OC

due to lower torque demands from the mild driver. When not following the drive trace as closely as possible, the ECMS is able to expand the area of focus and favor more efficient regions of the engine to recharge the HV battery. The F-ECMS can adjust the EF as shown in Figure 64 to prevent the motor from operating at the extreme low SOC values from times at 200-450 and 800-1100 seconds. Similarly, as the SOC increases, the EF begins to oscillate to allow the cost of the motor to fluctuate during operation. The adjustment of the EF allows the SOC to operate around the lower charge sustaining setpoint of 45% without large deviations to help prolong the battery life while maintaining good fuel economy. In the initial phase of the cycle from 0 to roughly 100 seconds, the EF is kept relatively high for both drivers to ensure that the ECMS favors the engine for OC to increase the value of the SOC. Once a peak of roughly 50% is reached, the fuzzy controller adjusts the EF to allow more use of the electric motor when commanding propulsive torque.



Figure 63: US06 32.5% SOC Comparison



Figure 64: US06 32.5% EF Comparison

## 5.3.3 EMC City Cycle

The F-ECMS may not have improved fuel economy when compared to the optimal ECMS with the smallest decrease of 3.83%, however, it maintained an average decrease of only 2.7% when comparing the F-ECMS to the average ECMS. The F-ECMS may not have improved fuel economy during simulations, however, substantial improvements for charge sustainability and SOC stability were performed. The EMC City drive cycle is shown below in Figure 65.





Figure 66 and Figure 67 provide results with a starting SOC of 50% while Figure 68 and Figure 69 provide the 32.5% starting SOC results for the EMC City drive cycle. In Figure 66, both the average and F-ECMS start off with nearly identical SOC traces until roughly 200 seconds. From that point, the F-ECMS adjusts the EF to maintain a rough average of 45% SOC. While the optimal SOC curve operates within the 40-60% range, the average violates the upper limit multiple times in the case of the mild driver. The F-ECMS maintains CS at the end of the cycle while the average ECMS violates the upper 55% window during both the aggressive and mild driver for the ending SOC, providing OC for both driving styles. This OC improves fuel economy, however, the SOC violates imposed constraints that would ultimately not be considered for a CS hybrid in the brute force analysis. Documentation for the GM HEV4 battery pack states that a strict limit of 20-80% SOC should always be followed to avoid damaging the battery pack by opening the HV contactors under a load. For this instance, an upper value of roughly 65% SOC is considered to be operating closer to the extreme range and should be avoided for safety of the HV battery pack as well as long term battery life/health.



Figure 66: EMC City 50% SOC Comparison



Figure 67: EMC City 50% EF Comparison

In the case for the mild driver in Figure 68, the energy strategy determines that it is more efficient to use the engine to charge the motor resulting in a higher, but more stable SOC curve. The ending result is not CS because the SOC is outside of the 50 +/- 5% range defined in the problem statement. Due to the difference in driver torque commands from the aggressive to mild driver, the F-ECMS updates the EF more frequently in the aggressive simulations, regardless of starting SOC. It should be noted that in the low starting SOC case, the F-ECMS is able to adjust the EF to raise the SOC above 40% within the first 100 seconds of the simulation while the average ECMS takes roughly 200 seconds. In the case of the aggressive driver in Figure 68, the SOC deviates from 32.5% to 30% briefly while the F-ECMS begins to increase the SOC almost instantly with minor decreases. Differences in the driving style between the aggressive and mild driver tend to cause fluctuations in where the ECMS algorithm decides components should operate in the case of the average ECMS, however, for both driving styles, the F-ECMS is able to adjust the EF and

keep the SOC operating in a consistent smaller range. This adaptive EF maintains roughly the same SOC throughout the cycle regardless of the selected driver.



Figure 68: EMC City 32.5% SOC Comparison



Figure 69: EMC City 32.5% EF Comparison

# 5.3.4 HWFET Cycle

Despite the decrease in fuel economy when compared to the average and optimal ECMS, the F-ECMS was able to improve CS throughout the drive cycle, regardless of the applied driving style. The HWFET drive cycle is characterized with mild to sharp accelerations with very little idle periods and a top speed of roughly 60 mph maintained for longer periods of time, resulting in less energy recapture through regenerative braking. This cycle is shown in Figure 70.



#### Figure 70: HWFET Drive Cycle

Figure 71 shows the difference between the optimal, average, and F-ECMS for the aggressive and mild drivers. The optimal trace does deviate below the 40% SOC line and above the 60% line twice, however this is not a substantial amount. Conversely, the average ECMS tends to favor opportunity charging with the aggressive driver to recharge the HV battery during high-speed sections of the drive cycle. This is due to the efficiency regions of the engine when considering higher speeds with moderate levels of driver requested wheel torque. The ECMS optimizes the engine by placing it in a higher torque region while the motor is used to charge the HV battery pack. This stored energy is used at minimally in the cycle when the engine is not operating in its most efficient area. In the case of the mild driver, the average and F-ECMS SOC curves are very close in shape and value until approximately 600 seconds where the average ECMS increases torque commands to the engine to further charge the HV battery. In each case, the F-ECMS is able to adjust the EF, shown in Figure 72, to maintain SOC balance around 45% regardless of the vehicle speed or wheel torque commands from the driver.



Figure 71: HWFET 50% SOC Comparison

Due to the responsive difference in drivers, the EF in the aggressive driver tends to fluctuate more aggressively in the range of 2-3.4. The EF for the mild driver tends to center around 3.4 for the duration of the drive with minor exceptions when the SOC rises above 45%. Regardless of the driving style, the F-ECMS is able to adjust the EF appropriately to keep the SOC within the desired operating window.



Figure 72: HWFET 50% EF Comparison

Similar results were obtained for a starting SOC of 32.5% for the HWFET cycle as shown in Figure 73 and Figure 74. The SOC trace follows a similar trend where the HV battery is charged during the high-speed scenarios for the aggressive driver and the mild driver mimics the F-ECMS trace from roughly 300-600 seconds. The F-ECMS is able to increase the EF, shown in Figure 74, to increase the cost of the electric motor early on until the SOC reaches the lower operating region of 45%. The average ECMS takes longer to charge the battery, and in the case of the mild driver, the SOC does not increase above the lower bound of 40% until 150 seconds have passed in the simulation. The EF follows a similar trend where the set value oscillates more frequently for the aggressive driver but both driving styles produce similar maximum and minimum values. When observing the differences in SOC for the aggressive and mild drivers, it is clear that the adaptable EF is more robust when maintaining an average SOC regardless of the reaction time

for the driver or top speeds of the vehicle. For both driving styles, the F-ECMS is able to end with an SOC of roughly 50% while the average ECMS exceeds the upper limit of 60% for both cases.



Figure 73: HWFET 32.5% SOC Comparison



Figure 74: HWFET 32.5% EF Comparison

### 5.3.5 EMC Highway Cycle

The F-ECMS improved fuel economy for all 4 tests when compared to the optimal ECMS with a maximum increase of 4.37%. It should be noted that the F-ECMS improved fuel economy in both tests with an initial SOC of 32.5%, however, the optimal ECMS EF was not tuned using this initial SOC which could provide different fuel economy results. When compared to the average ECMS, the F-ECMS improved fuel economy by a maximum of 4.89%. This cycle is characterized by aggressive accelerations, high speeds of 80 mph, and reduced idle periods, as shown in Figure 75.





Figure 76 and Figure 77 provide simulation results for a starting SOC of 50% with Figure 78 providing a snapshot of a portion of the drive cycle to highlight differences in the EF when a different driving style is applied. Figure 79 and Figure 80 provide results for a starting SOC of 32.5% with Figure 81 providing a snapshot of a zoomed in portion of the drive cycle to highlight EF differences in the driving styles. The optimal CS EF was found for the best fuel economy and as seen in Figure 76, the SOC oscillates through the range of 40-50% SOC throughout the cycle. The average ECMS produces CS results for both driving styles, however, differences can be seen in the figure below. The aggressive driver SOC curve follows the optimal for most of the cycle, with minor discrepancies at both the start and roughly 3000 seconds into the cycle. Conversely, the average ECMS tends to charge the battery to a higher SOC when using a mild driving style. This serves as another indication that an average EF can produce CS results over the drive cycle, but tuning is required for different driving styles to achieve similar results. The F-ECMS is able to maintain 45% SOC for a large portion of the drive cycle and does not deviate below the upper or lower SOC bound. The F-ECMS produces consistent results are remain in an operating region around 45% for

the duration of the drive regardless of the applied driving style and improves fuel economy when compared to both the average and optimal EF.



Figure 76: EMC Highway 50% SOC Comparison

Due to the harder accelerations and fluctuations of the accelerator pedal to match the drive trace as closely as possible, the EF for the aggressive driver changes frequently and produces a higher maximum when compared to the average driver. The F-ECMS follows a similar trend when comparing to the drive cycles discussed above: As the SOC decreases below 45%, the EF will increase to prevent the energy strategy from using the motor too often and depleting the HV battery. It should be noted that the SOC in the figures below rises quickly when in a regenerative state. The drive cycle shown has a simulation time of almost 6000 seconds which causes the SOC to appear to increase over a few time steps when viewing the entire cycle from start to finish. Figure 78 provides a zoomed in portion of the simulation to highlight the change in SOC more clearly.



Figure 77: EMC Highway 50% EF Comparison

Figure 78 provides a zoomed in view of the EMC Highway cycle to further highlight the change in the EF for both driving styles. The EF for the aggressive driver changes more frequently based on both the more aggressive torque commands to follow the drive trace more accurately and the larger fluctuations in the SOC. This figure provides a better snapshot to further illustrate the correlations between a higher EF factor and an SOC value of less than 45%. As shown in the figure, the EF for the aggressive driver has an average of roughly 3.1 with deviations below the initial EF when the SOC is above 45%. The EF for the mild driver has an average slightly below the starting average of 2.92 and deviates above when the decreases below 45% to penalize the motor and maintain an average SOC above 40%.



Figure 78: EMC Highway 50% EF 1000-2000 Seconds

Figure 79 and Figure 80 provide the SOC and EF results for the EMC Highway drive cycle with a starting SOC of 32.5%. From the figures, the results are similar to the 50% starting SOC case with few exceptions. In the case of the aggressive driver, once the simulation begins, the SOC curve matches the optimal curve until roughly 1000 seconds, when the SOC begins to deviate below 40% slightly. This is not a large issue, and the operating region is far above the extreme limit of 20% discussed previously, however, the results still prove that the optimal and average ECMS fluctuate farther than the F-ECMS but get lower fuel economy. Similarly, the F-ECMS maintains the operating region of 45% regardless of the initial starting SOC value. The mild driver follows the same pattern where the average EF was able to maintain CS operations and increases the SOC quickly at the beginning of the cycle. The mild driver for both starting SOC points is nearly identical, which correlates to the ECMS finding the motor to be equal in cost once the SOC balanced above 45%. The mild driver shows a similar case to others previously discussed where the

SOC is charged well above the minimum requirement. This is due to the mild driver smoothing out the drive trace and producing a wheel torque request lower than the aggressive driver. This lower request results in the ECMS choosing to run the engine in a higher efficiency region to provide OC for the HV battery. As a result, the motor is favored more for launching events, depleting the SOC down from roughly 65 to 45% in most cases. Regardless of higher efficiency regions for OC charging, the F-ECMS is able to outperform the average EF and improve fuel economy throughout the cycle.



Figure 79: EMC Highway 32.5% SOC Comparison

Due to hard accelerations at the beginning of the drive cycle, the EF fluctuated rapidly for the aggressive driver. The EF for the aggressive and mild drivers peak with a value of 3.97 at the beginning of the cycle due to the higher torque demand to match the drive trace and provide OC for the HV battery. These aggressive fluctuations occur due to the nature of the drive cycle. At higher speeds, when the drive cycle velocity changes, the aggressive driver oscillates between the accelerator and decelerator pedals quickly,

providing both motoring and generating torque on the electric motor. This quick transition between the pedals correlates to the fuzzy controller setting the EF high when the driver is on the accelerator pedal, and low when the driver is on the decelerator pedal to better match the drive trace.



Figure 80: EMC Highway 32.5% EF Comparison

Figure 81 provides a zoomed in view of the EMC Highway cycle to further highlight the change in the EF for both driving styles. The EF for the aggressive driver changes more frequently based on both the more aggressive torque commands to follow the drive trace more accurately and the fluctuations in the SOC. This figure provides a better snapshot to further illustrate the correlations between a higher EF factor and an SOC value of less than 45%. The aggressive EF tends to use step changes when adjusting between 2 EFs while the mild EF tends to have linear slopes when adjusting. As discussed previously, in the case of the aggressive driver, the drive trace is followed as closely as possible. When the driver is on the accelerator pedal, the EF will be higher due to the motor having a more expensive cost at a lower SOC value. If the

decelerator pedal is pressed, the motor will recapture energy raising the SOC level. Once the driver tips back into the accelerator pedal to follow the drive trace, the F-ECMS favors the motor until the SOC decreases below 45% and the EF is set higher again. Regardless of the driving style selected for the lower starting SOC value, the F-ECMS dynamically adjusts the EF to keep the SOC operating in a strict CS window throughout the entire cycle while improving fuel economy.



Figure 81: EMC Highway 32.5% EF 1000-2000 Seconds

### 5.3.6 Remaining Cycles

Results for the remaining drive cycles are shown below in Appendix C. Each drive cycle contained varying characteristics including different accelerations and coast down times, maximum speeds, and idle periods. For each drive cycle, the F-ECMS was able to outperform the average ECMS in terms of CS throughout the entire drive cycle regardless of starting SOC or driver style selection. Similarly, as discussed above, the F-

ECMS can compete with both the optimal and average ECMS algorithms in terms of fuel economy while maintaining the imposed SOC CS constraints. These results further validate the overall functionality of the F-ECMS when improving fuel economy over multiple drive cycles with reduced computational power required when considering the amount of time needed to optimize the EF for individual drive cycles and for each time the control hierarchy is modified in the design and calibration phase of modeling.

### 5.4 Benchmark Validation

The benchmark fuel economy results from several cycles are shown below in Table 11. These DP results were provided by Aaron Mull of WVU [19] in his work to create a DP algorithm to determine the best fuel economy for the WVU EcoCAR team's Blazer Simulink vehicle model. The results from the DP algorithm serve as a benchmark for future design work using the Simulink Blazer model. Charge corrected fuel economy results shown in the table for the F-ECMS were obtained from results discussed above and include the best overall fuel economy for each drive cycle shown in the table. However, if modifications to a Simulink vehicle model are made in the Plant, Controller, or Drivetrain, this can result in skewed or abnormal fuel economy results. As seen in the table, 4 out of the 5 drive cycles showed a fuel economy improvement when comparing the F-ECMS to DP. The results from DP should serve as the absolute best fuel economy that the model is able to produce, however, multiple instances show that the F-ECMS was able to outperform the DP. Ideally, the fuel economy results from the F-ECMS should be within 1-2% range of the DP results to justify validation results from both experiments.

	DP Benchmark (mpg)	F-ECMS (mpg)	Error (%)	
EMC City	30.18	32.74	8.48	
EMC Highway	37.24	31.54	-15.31	
US06	24.64	28.92	17.37	
UDDS	30.74	34.94	13.66	
HWFET	32.95	40.93	24.22	

Table 11: Benchmark Fuel Economy Comparison

It should be noted that different Simulink models were used in both experiments. The Simulink model used to validate the DP algorithm was not the same model used to validate the F-ECMS controller. The DP model was created in MATLAB as a backwards looking model, where component blocks including the SI Mapped Engine and Datasheet Battery were reconstructed using 2-D lookup tables with inputs of torque and current respectively. This can account for some margin of error when considering the construction and computational interpolation of the 2-D lookup tables.

The Blazer vehicle model contains hundreds of parameters for defining the engine, motor, battery, and vehicle dynamics with characteristics ranging from fuel heating characteristics to friction coefficients of the simulated wheels and brakes. When these parameters differ, results such as fuel economy or charge sustainability can be affected due to differences in the power or torque required to propel the vehicle and follow a drive trace. Validation of the DP algorithm should be performed with the Simulink model used in this paper to verify proper functionality of the model. Validation includes using the component torque commands provided from the DP benchmark analysis to propel the vehicle and obtain new fuel economy and CS results for each drive cycle, using the appropriate starting SOC used in the DP analysis. Once a new benchmark is established for this model, the F-ECMS can be rerun over each drive cycle to verify and validate overall functionality as well as better or worse fuel economy and CS results.

# 6 Conclusions and Recommendations

In conclusion, the objective of this work was to design and integrate a fuzzy logic-based controller to implement with the ECMS. The fuzzy logic controller was selected due to the adaptability of the controller regardless of driver style or selected cycle. The goals of this work were to maintain or improve fuel economy when compared to the optimal baseline achieved by the ECMS through a brute force analysis along with improving the charge sustainability of the control algorithms and HV battery life. The overall performance of the ECMS is dependent on the adjustment of the equivalence factor that must be

determined offline with *a priori* knowledge of the current drive cycle to achieve globally optimal results. This global optimum is determined by performing a brute force analysis where the EF is incremented in small steps over a large window to determine the optimal EF to maximum fuel economy while providing CS results. The main drawback of the ECMS is that each drive cycle and driving style requires a different EF, and any changes made in the control architecture require the brute force analysis to be rerun to find the new global optimum. The implemented fuzzy logic controller updates the EF based on the current deviation of the SOC from a pre-determined setpoint as well as the current driver wheel torque demand to provide an adaptive robust ECMS algorithm, implementable over multiple drive cycles with various driver characteristics.

### 6.1 Analysis

A brute force analysis was performed on the base ECMS with an initial EF range of 2-4 for 8 drive cycles. Once complete, a fuzzy logic controller was designed based on prior knowledge of the control system along with possible wheel torque commands including minimum and maximum values for each drive cycle. The fuzzy logic controller was tested against the baseline optimal results of the ECMS for the original 8 drive cycles along with 3 additional cycles to further validate the designed controller. The average EF for the original 8 drive cycles was determined to be 2.92 and was tested against the fuzzy controller to further validate the designed control system. This average EF was used to illustrate how the ECMS may not perform well if a full EF analysis is not conducted prior to implementing the algorithm in a real-world environment. The selected drive cycles covered a broad range of characteristics including idle time, mild and aggressive acceleration and decelerations, and vehicle top speeds.

The optimal fuel economy was achieved within an average of +- 2.44% error when comparing the fuel economy results of the optimal ECMS to the F-ECMS. It was found that the fuel economy improvements occurred with both drivers at each initial SOC value. The best fuel economy improvement was during the

125
EMC Highway cycle with a total of 4.37% over the optimal ECMS. Fuel economy improvements from the 32.5% starting SOC were not included in this comparison due to a brute force analysis not being conducted on the different starting SOC. While an analysis was not performed, the results serve as a benchmark when comparing an adaptive ECMS to a constant ECMS strategy with varying initial SOC values. From the mild driver starting at 50% SOC, 3 out of the 8 drive cycles showed improvements in fuel economy. When comparing the F-ECMS to the average ECMS method, 17 out of a total of 44 drive cycles showed an improvement in fuel economy with the largest improvement being 9.69% in the NYCC drive cycle. Out of the 17 cycles that showed improvements, 9 were from the mild driver. The mild driver presents a more accurate representation of a human driver in a real-world environment where slight deviations from the defined drive trace occur due to human error.

Even though the fuel economy improvements may not have been significant in most cases, the F-ECMS was able to maintain CS operations for 42 out of the total 44 that were simulated. For comparison, out of the 44 total tests, the average ECMS operated in a CS region for 12 in total. The average ECMS results demonstrate the importance of updating the EF when charge sustainability is defined as a goal. Initial SOC values and driving styles play a large role in the SOC results for different drive cycles. If a constant EF is implemented, calibration of the EF offline is crucial for an ECMS controller. The integrated FLC proved that an average EF can be adjusted dynamically during random simulated drive cycles to achieve fuel economy improvements and charge sustainability with reduced computational time required to tune the ECMS each time logic is manipulated within the control hierarchy.

## 6.2 Additional Drive Cycles

Given the results of the fuel economy for the FLC, improvements could be considered which merits future work. For future work, it is recommended that additional drive cycles be used to determine an average EF. Out of the 8 drive cycles analyzed, the lowest EF was 2.45 and the highest was 3.5, however, only 3

drive cycles had an optimum EF of greater than 3. Drive cycles that incorporate less aggressive accelerations up to roughly 15-20 m/s along with sustained vehicle speeds (higher and lower) should be considered to obtain an average EF closer to 2.5 to better mimic real world driving in both city and highway environments. 8 drive cycles were selected due to the time constraint of designing and testing the FLC against both the optimal and average ECMS.

## 6.3 FLC Variation

The FLC is currently designed to operate based on a predefined EF. The output of the FLC is a delta that is added to or subtracted from the average EF to update the controller. Adjusting the output of the FLC to control the raw EF will help to reduce the amount of prior EF analysis that is conducted to further reduce the computation demand of the ECMS. Designing a sliding time window can help to further reduce the computational demand for embedded controllers that do not have a substantial amount of throughput power. The FLC in this work updates the EF at each time step if an adjustment needs to be made. This can result in very frequent and abrupt changes to the EF that can create unwanted step changes in component torque demands. The sliding time window could be designed to assume that the current driving conditions will remain the same for an extended period of time. This window could be adjusted to update the EF in 1, 2, or even 5-minute windows depending on the desired update rate to achieve less oscillation from the updated EF while maintaining CS operations.

## 6.4 Model Modifications

Simulations conducted for this experiment used a model that did not take the road grade into account. Further experimentation should include moderate grade changes during each cycle to better simulate real-world driving conditions. These perturbations in grade could be predetermined or could be randomized to better capture the efficiency of the F-ECMS algorithm. The drive cycles in this experiment

were repeated twice to reduce any bias associated with the initial SOC. Additional run time should be considered to further reduce and/or eliminate bias from the experiment.

Table 11 compares results from a DP experiment that was conducted by a fellow WVU graduate student. The results indicated that the F-ECMS was able to outperform the DP algorithm in terms of fuel economy improvements. The Simulink models used in both experiments differed slightly due to linear interpolation of the backwards looking 2-D lookup tables, which could have caused skewed results when evaluating the fuel economy for both algorithms. For further validation, the F-ECMS algorithm should be run in real-time using the PCM MabX controller for the drive cycles discussed in this research. Real-world results can be post processed to determine fuel economy results using the 2019 Chevy Blazer to validate both algorithms and determine the root cause of fuel economy errors discussed above.

In the last year of the EcoCAR Mobility Challenge, teams are expected to have fully integrated V2X communication with the Cohda wireless radios. The FLC could be designed to receive information from the radio as an input to the system to further optimize the EF. For example, if the HV battery SOC is at 46% and decreasing, the current controller will begin to increase the EF to increase the cost of using the electric motor. However, in 1000 meters, there could be a traffic light that the vehicle will need to stop at for a certain period of time. Rather than increasing the EF to penalize using the motor, the EF could remain the same or decrease slightly with the added knowledge that the vehicle will be slowing down soon and utilizing regenerative braking. This added logic would allow the SOC to decrease slightly below a loose constraint to improve fuel economy by reducing the amount of fuel required to provide propulsion. This knowledge could be extended to receiving *a* priori information about the drive cycle. If idle vehicle periods or speed and acceleration changes are known ahead of time in the VIL environment, the FLC could update the EF more appropriately ahead of time. Additional inputs for the FLC should also be considered. The current vehicle speed is not considered within the FLC. The vehicle speed can be treated as an input to

the FLC to help relate the cost to use the motor at lower speeds where the engine is considered less efficient.

Lastly, the Longitudinal Driver accepts a driver response time variable that can be modified to increase or decrease the driver's reaction time. Initial simulations were performed to determine the difference between an aggressive and mild driver with final values of 0.1 and 0.5 respectively. Real world testing should be performed to measure the reaction time of a group of individuals to gain a realistic response time when running simulations. When the reaction time is adjusted to further mimic human response, simulated results can closer mimic results that are achievable in real-world implementation.

## References

- S. Onori, L. Serrao, and G. Rizzoni, "Hybrid electric vehicles: Energy management strategies," *SpringerBriefs Control. Autom. Robot.*, no. 9781447167792, p. 121, 2016, doi: 10.1007/978-1-4471-6781-5.
- [2] "Sources of Greenhouse Gas Emissions," Jul. 27, 2021.
   https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions#:~:text=Carbon dioxide
   (CO2) makes, natural gas%2C to produce electricity. (accessed Oct. 09, 2021).
- [3] "GM Our Path to an All-Electric Future," 2021. https://www.gm.com/electric-vehicles.html (accessed Oct. 09, 2021).
- [4] U.S. Department of Energy, "Emssions from Hybrid and Plug-In Electric Vehicles."
   https://afdc.energy.gov/vehicles/electric\_emissions.html (accessed Aug. 16, 2021).
- [5] "HEV Types."

http://autocaat.org/Technologies/Hybrid\_and\_Battery\_Electric\_Vehicles/HEV\_Types/ (accessed May 09, 2021).

- [6] "All About Drivetrains | Union of Concerned Scientists," Mar. 14, 2018.
   https://www.ucsusa.org/resources/all-about-drivetrains (accessed May 09, 2021).
- J. Liu and H. Peng, "Modeling and control of a power-split hybrid vehicle," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 6, pp. 1242–1251, 2008, doi: 10.1109/TCST.2008.919447.
- [8] M. De Santis, S. Agnelli, F. Patanè, O. Giannini, and G. Bella, "Experimental Study for the Assessment of the Measurement Uncertainty Associated with Electric Powertrain Efficiency Using the Back-to-Back Direct Method," *Energies 2018, Vol. 11, Page 3536*, vol. 11, no. 12, p. 3536, Dec. 2018, doi: 10.3390/EN11123536.

- [9] "Mild Hybrid Electric Vehicle (MHEV) ." https://x-engineer.org/automotiveengineering/vehicle/hybrid/mild-hybrid-electric-vehicle-mhev-architectures/ (accessed May 09, 2021).
- [10] "» AVTC HistoryAVTC I Advanced Vehicle Technology Competitions." https://avtcseries.org/avtchistory/ (accessed May 09, 2021).
- [11] "» CompetitionsAVTC I Advanced Vehicle Technology Competitions." https://avtcseries.org/avtchistory/competitions/ (accessed May 09, 2021).
- [12] "» EcoCar Mobility ChallengeAVTC I Advanced Vehicle Technology Competitions." https://avtcseries.org/ecocar-mobility-challenge/ (accessed May 09, 2021).
- [13] "CAV Cohda Wireless." https://www.cohdawireless.com/sectors/cav/ (accessed May 09, 2021).
- [14] "GM 2.5 Liter I4 Ecotec LCV Engine." https://gmauthority.com/blog/gm/gm-engines/lcv/ (accessed May 09, 2021).
- [15] H. Bouvier, G. Colin, and Y. Chamaillard, "Determination and comparison of optimal eco-driving cycles for hybrid electric vehicles," in *2015 European Control Conference (ECC)*, Jul. 2015, pp. 142–147, doi: 10.1109/ECC.2015.7330536.
- [16] R. Wang and S. M. Lukic, "Dynamic programming technique in hybrid electric vehicle optimization," 2012 IEEE Int. Electr. Veh. Conf. IEVC 2012, 2012, doi: 10.1109/IEVC.2012.6183284.
- [17] V. Larsson, L. Johannesson, and B. Egardt, "Analytic solutions to the dynamic programming subproblem in hybrid vehicle energy management," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1458–1467, Apr. 2015, doi: 10.1109/TVT.2014.2329864.
- [18] Z. Yuan, L. Teng, S. Fengchun, and H. Peng, "Comparative Study of Dynamic Programming and

Pontryagin's Minimum Principle on Energy Management for a Parallel Hybrid Electric Vehicle," *Energies 2013, Vol. 6, Pages 2305-2318*, vol. 6, no. 4, pp. 2305–2318, Apr. 2013, doi: 10.3390/EN6042305.

- [19] A. Mull, A. Nix, D. Ph, S. Wayne, D. Ph, M. Perhinschi, D. Ph, "Powertrain Fuel Consumption Modeling and Benchmark Analysis of a Parallel P4 Hybrid Electric Vehicle Using Dynamic Programming," 2021.
- [20] L. Johannesson, M. Åsbogård, and B. Egardt, "Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2005, pp. 366–371, 2005, doi: 10.1109/ITSC.2005.1520076.
- [21] D. F. Opila, X. Wang, R. McGee, and J. W. Grizzle, "Real-Time Implementation and Hardware Testing of a Hybrid Vehicle Energy Management Controller Based on Stochastic Dynamic Programming," *J. Dyn. Syst. Meas. Control*, vol. 135, no. 2, Mar. 2013, doi: 10.1115/1.4007238.
- [22] L. V. Pérez, G. R. Bossio, D. Moitre, and G. O. García, "Optimization of power management in an hybrid electric vehicle using dynamic programming," *Math. Comput. Simul.*, vol. 73, no. 1-4 SPEC.
   ISS., pp. 244–254, Nov. 2006, doi: 10.1016/J.MATCOM.2006.06.016.
- [23] 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.
- [24] F. Zhang, J. Xi, and R. Langari, "Real-Time Energy Management Strategy Based on Velocity
   Forecasts Using V2V and V2I Communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 2, pp. 416–430, Feb. 2017, doi: 10.1109/TITS.2016.2580318.
- [25] S. Jeon, S. Jo, Y. Park, and J. Lee, "Multi-Mode Driving Control of a Parallel Hybrid Electric Vehicle

Using Driving Pattern Recognition," J. Dyn. Syst. Meas. Control, vol. 124, no. 1, pp. 141–149, Mar. 2002, doi: 10.1115/1.1434264.

- [26] T. P. Harris, A. C. Nix, M. G. Perhinschi, W. S. Wayne, J. A. Diethorn, and A. R. Mull,
   "Implementation of Radial Basis Function Artificial Neural Network into an Adaptive Equivalent
   Consumption Minimization Strategy for Optimized Control of a Hybrid Electric Vehicle," *J. Transp. Technol.*, vol. 11, no. 04, pp. 471–503, 2021, doi: 10.4236/jtts.2021.114031.
- [27] 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, 2017, doi: 10.1109/TIV.2017.2716839.
- [28] N. J. Connelly, D. I. George, A. C. Nix, and W. S. Wayne, "Generation and Analysis of Hybrid-Electric Vehicle Transmission Shift Schedules with a Torque Split Algorithm," *J. Transp. Technol.*, vol. 10, no. 01, pp. 21–49, 2020, doi: 10.4236/jtts.2020.101003.
- [29] D. George, "Hybrid Electric Vehicle Torque Split Algorithm for Reduction of Engine Torque Transients," Grad. Theses, Diss. Probl. Reports, Jan. 2018, doi: https://doi.org/10.33915/etd.7179.
- [30] R. Babuska and E. Mamdani, "Fuzzy control," *Scholarpedia*, vol. 3, no. 2, p. 2103, 2008, doi: 10.4249/SCHOLARPEDIA.2103.
- [31] M. Perhinschi, "Fuzzy Logic," Department of Mechanical and Aerospace Engineering, West Virginia University, Morgantown 2018.
- [32] F. Zhang, H. Liu, Y. Hu, and J. Xi, "A supervisory control algorithm of hybrid electric vehicle based on adaptive equivalent consumption minimization strategy with fuzzy PI," *Energies*, vol. 9, no. 11, Nov. 2016, doi: 10.3390/EN9110919.

- [33] S. Wang, X. Huang, J. M. Lopez, X. Xu, and P. Dong, "Fuzzy Adaptive-Equivalent Consumption Minimization Strategy for a Parallel Hybrid Electric Vehicle," *IEEE Access*, vol. 7, pp. 133290– 133303, 2019, doi: 10.1109/ACCESS.2019.2941399.
- [34] N. Denis, M. R. Dubois, and A. Desrochers, "Fuzzy-based blended control for the energy management of a parallel plug-in hybrid electric vehicle," *IET Intell. Transp. Syst.*, vol. 9, no. 1, pp. 30–37, 2015, doi: 10.1049/iet-its.2014.0075.
- [35] S. G. Li, S. M. Sharkh, F. C. Walsh, and C. N. Zhang, "Energy and battery management of a plug-in series hybrid electric vehicle using fuzzy logic," *IEEE Trans. Veh. Technol.*, vol. 60, no. 8, pp. 3571– 3585, 2011, doi: 10.1109/TVT.2011.2165571.
- [36] D. Zhao, R. Stobart, G. Dong, and E. Winward, "Real-Time Energy Management for Diesel Heavy Duty Hybrid Electric Vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 3, pp. 829–841, 2015, doi: 10.1109/TCST.2014.2343939.
- [37] Q. Guo, Z. Zhao, P. Shen, X. Zhan, and J. Li, "Adaptive optimal control based on driving style recognition for plug-in hybrid electric vehicle," *Energy*, vol. 186, p. 115824, 2019, doi: 10.1016/j.energy.2019.07.154.
- [38] S. Wahsh, H. G. Hamed, M. N. F. Nashed, and T. Dakrory, "Fuzzy logic based control strategy for parallel hybrid electric vehicle," *Proc. 11th Int. Middle East Power Syst. Conf. MEPCON'2006*, vol. 1, pp. 30–34, 2006.
- [39] T. A. Anderson, J. M. Barkman, and C. Mi, "Design and optimization of a fuzzy-rule based hybrid electric vehicle controller," 2008 IEEE Veh. Power Propuls. Conf. VPPC 2008, 2008, doi: 10.1109/VPPC.2008.4677545.
- [40] B. Belaidi, I. Ouachani, K. Gadonna, D. Van Rechem, and H. Billard, "Energy management strategy

for hybrid electric vehicle using fuzzy logic," *Lect. Notes Electr. Eng.*, vol. 681, pp. 555–564, 2021, doi: 10.1007/978-981-15-6259-4\_58.

- [41] Z. Danhong, Z. Yan, L. Kai-Pei, and C. Qing-Quan, "A study on fuzzy control of energy management system in hybrid electric vehicle," *Asia-Pacific Power Energy Eng. Conf. APPEEC*, pp. 0–3, 2009, doi: 10.1109/APPEEC.2009.4918119.
- [42] Y. Xie, A. Savvaris, and A. Tsourdos, "Fuzzy logic based equivalent consumption optimization of a hybrid electric propulsion system for unmanned aerial vehicles," *Aerosp. Sci. Technol.*, vol. 85, no. February, pp. 13–23, 2019, doi: 10.1016/j.ast.2018.12.001.
- [43] Inc. The MathWorks, "Powertrain Blockset Toolbox, Longitudinal Driver." https://www.mathworks.com/help/vdynblks/ref/longitudinaldriver.html.
- [44] Inc. The MathWorks, "Powertrain Blockset Toolbox, SI Controller." https://www.mathworks.com/help/autoblks/ref/sicontroller.html?searchHighlight=SI engine controller&s\_tid=srchtitle.
- [45] Inc. The MathWorks, "Powertrain Blockset Toolbox, Mapped SI Engine." https://www.mathworks.com/help/autoblks/ref/mappedsiengine.html?searchHighlight=mapped si engine&s\_tid=srchtitle.
- [46] Inc. The MathWorks, "Powertrain Blockset Toolbox, Datasheet Battery." https://www.mathworks.com/help/autoblks/ref/datasheetbattery.html?searchHighlight=datash eet battery&s\_tid=srchtitle.
- [47] Inc. The MathWorks, "Powertrain Blockset Toolbox, Mapped Motor Model." https://www.mathworks.com/help/autoblks/ref/mappedmotor.html?searchHighlight=mapped motor model&s\_tid=srchtitle.

- [48] Inc. The MathWorks, "Powertrain Blockset Toolbox, Ideal Fixed Gear Transmission." https://www.mathworks.com/help/autoblks/ref/idealfixedgeartransmission.html?s\_tid=doc\_ta.
- [49] Inc. The MathWorks, "Powertrain Blockset Toolbox, Open Differential." https://www.mathworks.com/help/vdynblks/ref/opendifferential.html?s\_tid=doc\_ta.
- [50] Inc. The MathWorks, "Powertrain Blockset Toolbox, Longitudinal Wheel." https://www.mathworks.com/help/autoblks/ref/longitudinalwheel.html?s\_tid=doc\_ta.
- [51] Inc. The MathWorks, "Powertrain Blockset Toolbox, Vehicle Body 1DOF Longitudinal." https://www.mathworks.com/help/autoblks/ref/vehiclebody1doflongitudinal.html?s\_tid=doc\_ta
- [52] Inc. The MathWorks, "Fuzzy Logic Toolbox." https://www.mathworks.com/help/fuzzy/index.html?searchHighlight=fuzzy logic toolbox&s\_tid=srchtitle.

.

Appendix A: Validation Drive Cycles



Figure 82: NYCC Drive Cycle



Figure 83: SC03 Drive Cycle



Figure 84: US06 Drive Cycle



Figure 85: EMC City Drive Cycle



Figure 86: RTS95 Drive Cycle



Figure 87: HWFET Drive Cycle



Figure 88: HUDDS Drive Cycle



Figure 89: EMC Highway Drive Cycle



Figure 90: UDDS Drive Cycle



Figure 91: LA92 Drive Cycle



Figure 92: Artemis Rural Road Drive Cycle

Appendix B: ECMS Brute Force Analysis MATLAB Code

```
clc
clear
tic
% Load in the parameters for the model to reference
load('Model_Parameters_V2.mat')
% Define the method for the ECMS multiport switch in the Energy Management
% Subsystem. A value of 1 corresponds to the constant ECMS. A value of 2
% corresponds to the fuzzy adaptive method.
ECMS_method = 1;
% Define the constant value for switching the multiport switch in the
% Driver subsystem to cycle through the various drive cycles.
DriveCycleSelection = 1:1:8;
CurrDriveCycle = 1;
ModelRunTime = 0;
s_equiv = [2:0.05:4];
for Z = 1:1:length(DriveCycleSelection)
    % This will allow the Drive Cycles to be iterated using the multiport
    % switch in the Driver Subsystem
    CurrDriveCycle = DriveCycleSelection(Z);
    % Update simulation time based on selected drive cycle
    if CurrDriveCycle == 1
        ModelRunTime = 598*2:
    elseif CurrDriveCycle == 2
        ModelRunTime = 600*2;
    elseif CurrDriveCycle == 3
        ModelRunTime = 600*2;
    elseif CurrDriveCycle == 4
        ModelRunTime = 739*2;
    elseif CurrDriveCycle == 5
        ModelRunTime = 886*2;
    elseif CurrDriveCycle == 6
        ModelRunTime = 765*2;
    elseif CurrDriveCycle == 7
        ModelRunTime = 1060*2;
    else
        ModelRunTime = 2962*2;
    end
    % Define the loop to adjust the EF and record reported results
    for i = 1:1:length(s_equiv)
        % Iterate through the EF
        S_equiv = s_equiv(i);
        % Display the current drive cycle and EF iteration
        disp('The Current Drive Cycle Number is:')
```

```
disp(Z)
    disp('The current equivalence factor is:')
    disp(S_equiv)
    disp('The current model run time is:')
    disp(ModelRunTime)
    % Once all updates are done, save and run the model
    save_system('Simplified_Model.slx')
    sim('Simplified_Model.slx')
   % Load in the struct from the 'To File' block once the simulation
    % is done running
    load('LoggedSignals.mat')
    % Extract the SOC timeseries to check to see if charge sustaining was
    % achieved during the cycle
    SOC_timeseries = loggedsignals.SOC;
    SOC_finalvalue = SOC_timeseries.data(end);
   % Extract the Fuel Economy timeseries to record the data for each
    % simulation
    FuelEconomy_timeseries = loggedsignals.MPG_Corr;
    FuelEconomy_finalvalue = FuelEconomy_timeseries.data(end);
   % Check to see if the absolute value of the delta SOC is less than
    % 5%. If not, zero out the fuel economy for that iteration
    if abs(SOC_finalvalue - SOCTrgt) < 5.1 && abs(SOC_finalvalue - SOCTrgt) > 0
        AverageFuelEconomy = FuelEconomy_finalvalue;
    else
        AverageFuelEconomy = 0;
    end
    % Create a vector to keep records of the average fuel economy and value
    % of s that corresponds to it.
    FuelEcon_sequiv(1,i) = s_equiv(i);
    FuelEcon_sequiv(2,i) = AverageFuelEconomy;
   % Create the combined struct to be analzyed after the simulations are
    % done
    FinalStruct(1,i) = loggedsignals;
    ActualValues(1,i) = s_equiv(i);
    ActualValues(2,i) = FinalStruct(i).SOC.data(end);
    ActualValues(3,i) = FinalStruct(i).MPG_Corr.data(end);
    ActualValues(4,i) = FinalStruct(i).MPG_NonCorr.data(end);
   % All logged variables sre deleted from the base workspace
    % before beginning the next simulation
    clear('loggedsignals','SOC_timeseries',...
        'SOC_finalvalue', 'FuelEconomy_timeseries',...
        'FuelEconomy_finalvalue', 'AverageFuelEconomy')
    delete('LoggedSignals.mat')
end
```

% Once an entire iteration of the EF is complete for each drive cycle,

```
% change directory and store the results in the appropriate folder
if CurrDriveCycle == 1
    cd ..
    cd ('Model Data for Constant S Equiv ECMS')
    cd ('NYCC Cycle')
    save('NYCC Data', 'FuelEcon_sequiv', 'ActualValues', 'FinalStruct')
    cd ..
    cd ..
    cd ('Simple Model V2')
elseif CurrDriveCycle == 2
    cd ..
    cd ('Model Data for Constant S Equiv ECMS')
    cd ('SC03')
    save('SC03 Data', 'FuelEcon_sequiv', 'ActualValues', 'FinalStruct')
    cd ..
    cd ..
    cd ('Simple Model V2')
elseif CurrDriveCycle == 3
    cd ..
    cd ('Model Data for Constant S Equiv ECMS')
    cd ('US06')
    save('US06 Data', 'FuelEcon_sequiv', 'ActualValues', 'FinalStruct')
    cd ..
    cd ..
    cd ('Simple Model V2')
elseif CurrDriveCycle == 4
    cd ..
    cd ('Model Data for Constant S Equiv ECMS')
    cd ('EMC City')
    save('EMC City Data','FuelEcon_sequiv','ActualValues','FinalStruct')
    cd ..
    cd ..
    cd ('Simple Model V2')
elseif CurrDriveCycle == 5
    cd ..
    cd ('Model Data for Constant S Equiv ECMS')
    cd ('RTS 95')
    save('RTS 95 Data', 'FuelEcon_sequiv', 'ActualValues', 'FinalStruct')
    cd ..
    cd ..
    cd ('Simple Model V2')
elseif CurrDriveCycle == 6
    cd ..
    cd ('Model Data for Constant S Equiv ECMS')
    cd ('HWFET')
    save('HWFET Data','FuelEcon_sequiv','ActualValues','FinalStruct')
    cd ..
    cd ..
    cd ('Simple Model V2')
elseif CurrDriveCycle == 7
    cd ..
    cd ('Model Data for Constant S Equiv ECMS')
    cd ('HUDDS')
    save('HUDDS Data','FuelEcon_sequiv','ActualValues','FinalStruct')
```

```
cd ..
        cd ..
        cd ('Simple Model V2')
   else
        cd ..
        cd ('Model Data for Constant S Equiv ECMS')
        cd ('EMC Highway')
        save('EMC Highway Data','FuelEcon_sequiv','ActualValues','FinalStruct')
        cd ..
        cd ..
        cd ('Simple Model V2')
   end
   \% Clear the final variables before beginning the next round of
   % simulations
   clear('FuelEcon_sequiv', 'ActualValues', 'FinalStruct')
end
toc
```

Published with MATLAB® R2020b





Figure 93: SC03 50% SOC Comparison



Figure 94: SC03 50% EF Comparison



Figure 95: SC03 32.5% SOC Comparison



Figure 96: SC03 32.5% EF Comparison



Figure 97: RTS95 50% SOC Comparison



Figure 98: RTS95 50% EF Comparison



Figure 99: RTS95 32.5% SOC Comparison



Figure 100: RTS95 32.5% EF Comparison



Figure 101: HUDDS 50% SOC Comparison



Figure 102: HUDDS 50% EF Comparison



Figure 103: HUDDS 32.5% SOC Comparison



Figure 104: HUDDS 32.5% EF Comparison



Figure 105: UDDS 50% SOC Comparison



Figure 106: UDDS 50% EF Comparison



Figure 107: UDDS 32.5% SOC Comparison


Figure 108: UDDS 32.5% EF Comparison



Figure 109: LA92 50% SOC Comparison



Figure 110: LA92 50% EF Comparison



Figure 111: LA92 32.5% SOC Comparison



Figure 112: LA92 32.5% EF Comparison



Figure 113: Artemis Rural Road 50% SOC Comparison



Figure 114: Artemis Rural Road 50% EF Comparison



Figure 115: Artemis Rural Road 32.5% SOC Comparison



Figure 116: Artemis Rural Road 32.5% EF Comparison