Scientific Research Publishing

Journal of Transportation Technologies, 2024, 14, 88-118
https://www.scirp.org/journal/jtts
ISSN Online: 2160-0481
ISSN Print: 2160-0473

Implementation of Fuzzy Logic Control into an Equivalent Minimization Strategy for Adaptive Energy Management of a Parallel Hybrid Electric Vehicle

Jared A. Diethorn, Andrew C. Nix, Mario G. Perhinschi, W. Scott Wayne

Department of Mechanical and Aerospace Engineering, West Virginia University, Morgantown, WV, USA
Email: jadiethorn@mix.wvu.edu, Andrew.Nix.mail.wvu.edu, Mario.Perhinschi@mail.wvu.edu, Scott.Wayne.mail.wvu.edu

How to cite this paper: Diethorn, J.A., Nix, A.C., Perhinschi, M.G. and Wayne, W.S. (2024) Implementation of Fuzzy Logic Control into an Equivalent Minimization Strategy for Adaptive Energy Management of a Parallel Hybrid Electric Vehicle. *Journal of Transportation Technologies*, **14**, 88-118.
https://doi.org/10.4236/jtts.2024.141007

Received: November 13, 2023
Accepted: January 27, 2024
Published: January 30, 2024

Abstract

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. Hybrid electric vehicles (HEVs) have been introduced to mitigate 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% of the desired SOC regardless of starting SOC.

Keywords

Hybrid Electric Vehicle, Fuzzy Logic, Adaptive Control, Charge Sustainability

1. Introduction

In 2019, greenhouse gas emissions from transportation accounted for roughly 29% of the total greenhouse gas emissions in the U.S. 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 [1]. HEVs can produce less tailpipe related emissions when using gasoline because they can operate more efficiently. 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 [2].

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. 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.

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.

HEVs have three architectures: series, parallel and a combination of both (series-parallel) shown in [3]. Each architecture serves a different purpose, with the price typically increasing due to the additional complexity of both the hardware and software of the system.

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. 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 [4]. Implementation of adaptive control strategies allow the optimization of an algorithm during use, which will lead to increased vehicle efficiency over the duration of a drive cycle.

2. Vehicle Testing Architecture

The vehicle modeled in this work is a P4 parallel HEV architecture that the West Virginia University EcoCAR team designed in the EcoCAR Mobility Challenge as part of the Advanced Vehicle Technology Competition (AVTC). The architecture is provided below in **Figure 1**.

The performance metrics for the hybrid powertrain are shown in **Table 1** [5].

Table 1. West virginia university EcoCAR team competition vehicle architecture powertrain performance metrics.

Component	Specifications
Engine	• General Motors (GM) 2.5 L Naturally Aspirated LCV • Peak Power: 148 kW • Peak Torque: 255 Nm
Transmission	• GM M3D (9T50) 9-Speed Automatic
Fuel	• E10 Regular
Energy Storage System	• GM HEV4 • Peak Power: 50 kW • Energy Capacity: 1.5 kWh
Motor	• Magna Powertrain Electrified Rear Axle Drive (eRAD) • Peak Power: 50 kW • Peak Torque: 200 Nm • Integrated Gear Ratio: 9.17
Inverter	• Magna Dual Inverter

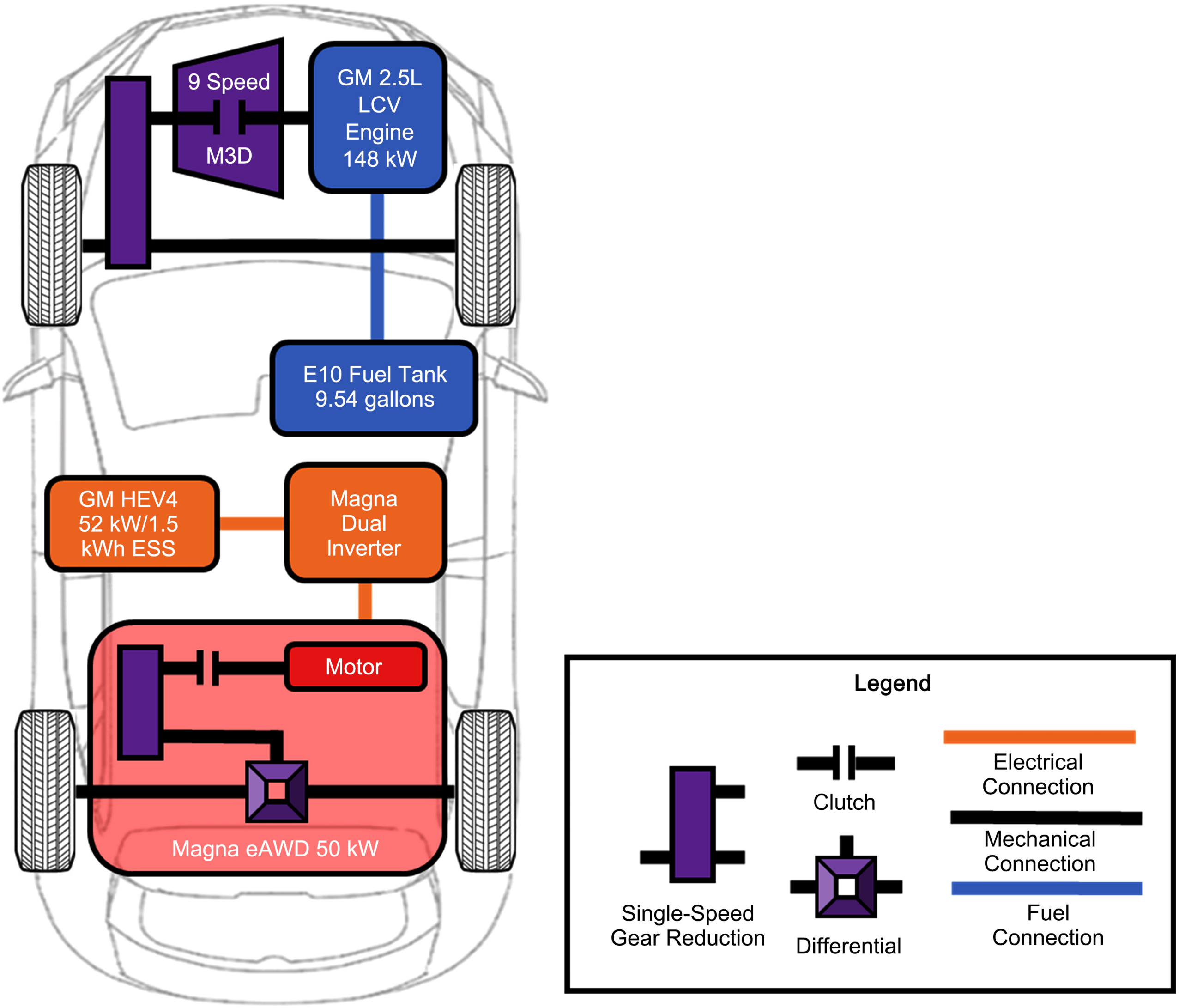

Figure 1. West virginia university EcoCAR team competition vehicle architecture [5].

Increasing efficiency entails moving the engine into an area where the brake specific fuel consumption (BSFC) is minimized. 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 efficient split is commanded between the engine and motor, the engine would never be used due to the higher efficiency of the electric motor. The torque split algorithm selected must understand the operating regions for both propulsion components in order to calculate the ideal operating regions.

Consider an engine efficiency map shown in Figure 2 [6]. The engine speed is on the x-axis and 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. Consider the following scenario shown in Figure 2 (Left). If the engine is producing roughly 60 Nm of torque at 2200 rpm (point A), it is less efficient than an

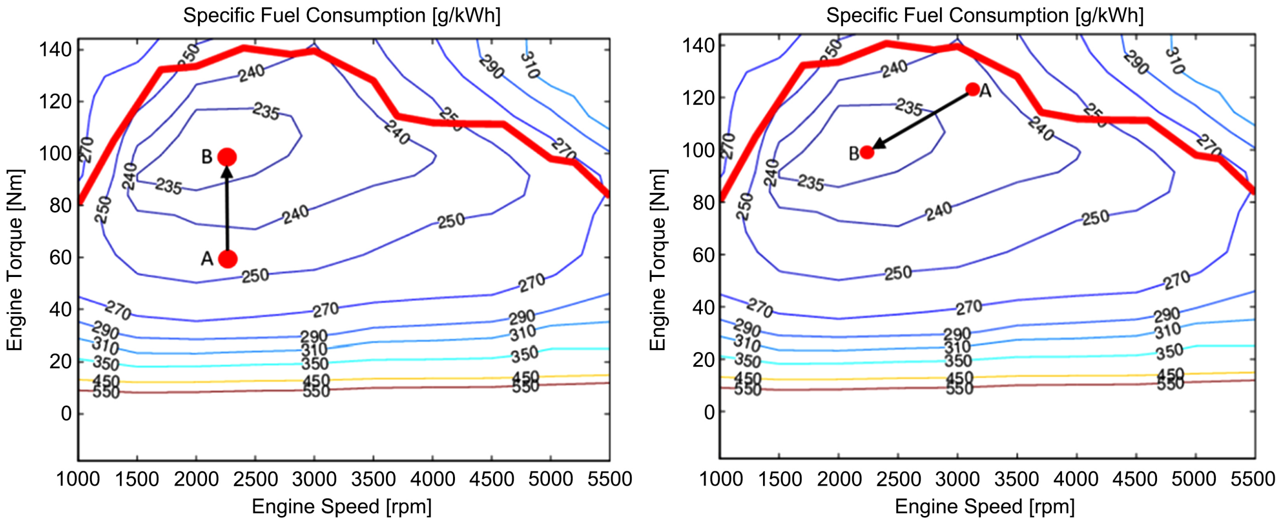

Figure 2. Engine optimization with an electric motor.

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. To mitigate a torque 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. Now consider the following situation (Figure 2, right) when the engine is producing a higher amount of torque at an increased engine speed. 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 by shifting to a lower gear.

3. Literature Review

3.1. The Basic Problem

When designing an 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. Strategies can be broken down into 2 categories: Rule-based and Optimal-based strategies. This paper will focus on Optimal-based strategies which can be further broken down into 2 categories: Numerical and analytical. Numerical optimization methods include dynamic programming, genetic algorithms, and stochastic dynamic programming. 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 [4].

Rizzoni *et al.* [4] introduces a formula to minimize the total mass of fuel consumed, as follows:

$$J = \int_{t_0}^{t_f} \dot{m}_f\left(u(t), t\right)\mathrm{d}t \tag{1}$$

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 [4].

3.2. 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 [4]:

$$u_k = \mu^*\left(x_k, k\right) = \arg\min_{u \in U_k}\left(L_k\left(x_k, u\right) + Y_{k+1}\left(f_k\left(x_k, u_k\right), u_k\right)\right) \tag{2}$$

where $k = N-1\cdots,1$. $\left(x_1, 1\right)$ is found in the last iteration of the algorithm and is equal to the optimal cost $J*\left(x_0\right)$ discussed above. $\left(x_N, N\right)$ is equal to the terminal cost and is dependent on the final state of x_N. $\left(x_k, k\right)$ represents the optimal "cost-to-go" from time k to the end of the optimization horizon. $Y\left(x_k, u_k\right)$ is dependent on the control variable uk 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 xk are stored in the matrix $u*$ [4]. The numbers on each segment represent the cost from the minimum cost-to-go function $\left(x, k\right)$ at each instant. Once the DP algorithm is finished, the minimum cost is selected for the implemented control algorithm. Work in DP has been conducted at West Virginia University on a similar vehicle model by Aaron mull *et al.* [5].

3.3. Equivalent Consumption Minimization Strategy

The ECMS provides an effective solution to energy management problems discussed previously by using a heuristic method. 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 [4]. This relationship is shown below in the following equation:

$$\dot{m}_{f,eqv}(t) = \dot{m}_f(t) + \dot{m}_{ress}(t) \tag{3}$$

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_{lh}, MJ/kg) along with the engine efficiency (η_{eng}) and power produced by the engine (P_{eng}) as shown below.

$$\dot{m}_f(t) = \frac{P_{eng}(t)}{\eta_{eng}(t)Q_{lhv}} \tag{4}$$

The electrical energy is given an equivalent fuel consumption value, or virtual fuel, shown in Equation (5) 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}) [4]. It should be noted that proper unit conversions are needed in these equations when converting from MJ to kJ etc.

$$\dot{m}_{ress}(t) = sfc_{eq}(t) * P_{batt}(t) \tag{5}$$

The virtual fuel consumption is determined by applying the equivalence factor ((t)), which is a vector of values used for both charging and discharging the HV battery. This relationship can be seen in Equation (6).

$$sfc_{eq}(t) = \frac{s(t)}{Q_{lhv}} \tag{6}$$

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 (3). 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. 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 [6]. During operation with the ECMS, boundaries must be determined to prevent the HV battery from violating any admissible limits that could harm the battery pack. Due to these limitations, a multiplicative penalty function is generated to ensure all SOC limits are obeyed.

$$p(\mathrm{SOC}) = 1 - \left(\frac{\mathrm{SOC}(t) - \mathrm{SOC}_{\mathrm{target}}}{\frac{\mathrm{SOC}_{\max} - \mathrm{SOC}_{\min}}{2}} \right)^a \tag{7}$$

This function takes deviations from the target SOC into account and will apply a penalty based on the direction of the deviation. **Figure 3** shows 3 different penalty functions for varying values of the exponent a [4].

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

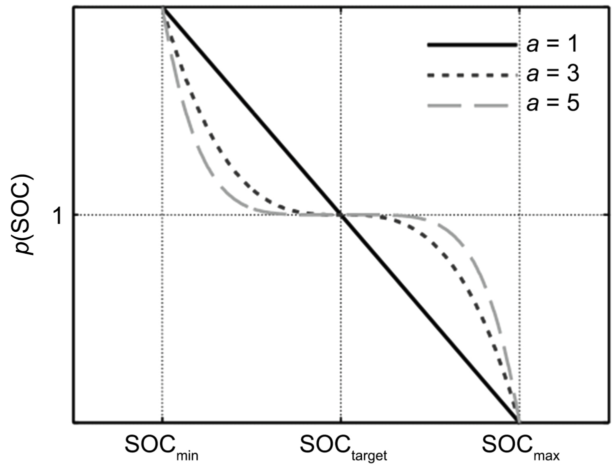

Figure 3. Varying penalty for HV SOC.

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 (3) as follows:

$$\dot{m}_{f,eqv}(t) = \dot{m}_f(t) + \frac{s(t)}{Q_{lhv}} * P_{batt}(t) * p(\text{SOC}) \tag{8}$$

Applying the ECMS through Equation (3) 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 [4].

3.4. 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 [4]. 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.

Several works have included using A-ECMS for speed predictions and chaining neural networks [7] [8] using the look-ahead horizon to determine the most likely behavior of the battery in the near future. This behavior helped to adjust the equivalence factor and apply the ECMS to split commanded torque between the engine and motor.

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 [9] [10], multi-mode control algorithms were designed using driving pattern recognition. Numerous driving patterns were identified to obtain different characteristics, including stop/total time, acceleration, and average cycle velocity. A neural network decides, when driving, which representative driving pattern is closest to the current pattern by comparing the correlation related to the characteristic parameters.

3.5. 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. 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. Fuzzy logic uses if-then rules and sets to define the meaning of qualitative values for a controller input [11]. A fuzzy logic controller consists of three major areas or modules: the fuzzification input, inference engine, and defuzzification output, shown in **Figure 4**.

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 [12]. Membership functions of a fuzzy set determine the degree to which a crisp value for a linguistic variable belongs to a certain linguistic value.

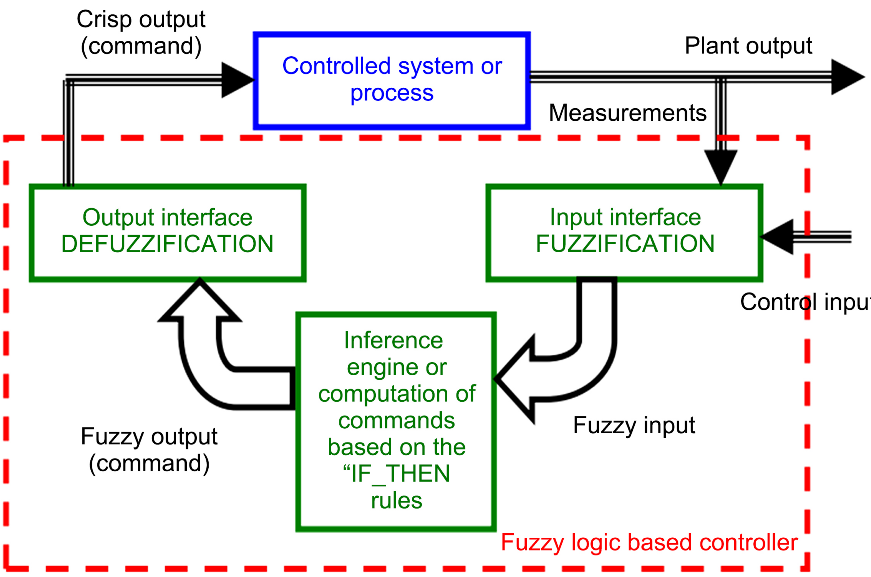

Figure 4. Overview of fuzzy logic architecture.

Membership functions can take on various shapes, however, the more common shapes are trapezoidal, triangular, bell shaped, and sinusoidal. 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. There are several conversion algorithms that are based on the height of the fuzzy sets and/or the area of the sets, including Center-of-Sum, Height-at-Lower-Value, Height-at-Peak-Value, and Last-of-Maxima [12].

In [13] [14], adaptations were made to the equivalence factor for the ECMS using fuzzy logic with proportional integral controllers, along with adapting based on the deviation of the actual SOC level. Constraints were placed on the model where the ending SOC had to equal the starting SOC. To impose the SOC constraints, the following cost function equation was implemented:

$$\begin{aligned} J\left(SoC(t)\right) = \int_{t_0}^{t} \dot{m}_f\left(u(t),t\right)\mathrm{d}t + \eta\frac{Q_{\max}}{H_{LHV}}\int_{SoC(t_0)}^{SoC(t)} V_{oc}\mathrm{d}\left(1-SoC\right) \\ + \beta\left(SOC_r - SoC(t)\right)^2 \end{aligned} \tag{9}$$

where SOC_r is the reference SOC value, $SOC_r - SoC(t)$ is the delta SOC value during operation, η 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 β is a tunable penalty coefficient. In both works, the adaptive ECMS improved fuel economy by 1% - 5%.

Denis *et al.* [15] 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. Three membership functions were used for the moving average and nine functions were used to define the global discharge rate.

4. 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 wheel torque command. The vehicle model is composed of 5 key areas (**Figure 5**). The Driver subsystem contains a driver model along with the drive cycle selected for the current simulation. The Controller subsystem contains all 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 motor. The three subsystems are connected through virtual interfaces to mimic communication processes that

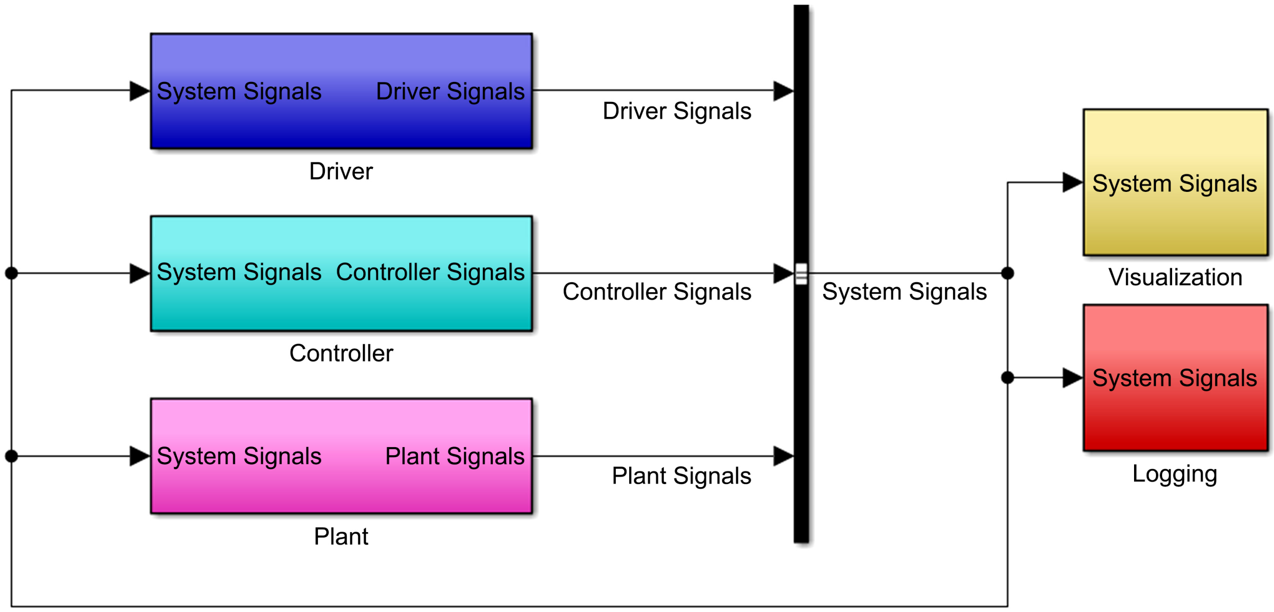

Figure 5. Hybrid vehicle model overview.

would be observed in a vehicle. The two remaining subsystems serve to visualize and log signals that are used in the post-processing part of the simulation. 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.

The Controller system (**Figure 6**) consists of 3 main subsystems: The Input Layer, Hybrid Supervisory Controller (HSC), 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.

The Application Layer (HSC) consists of two main systems: The Accel Pedal and Regen Braking and Energy Management subsystems. The following discussion will focus on modifications made to the Energy Management subsystem.

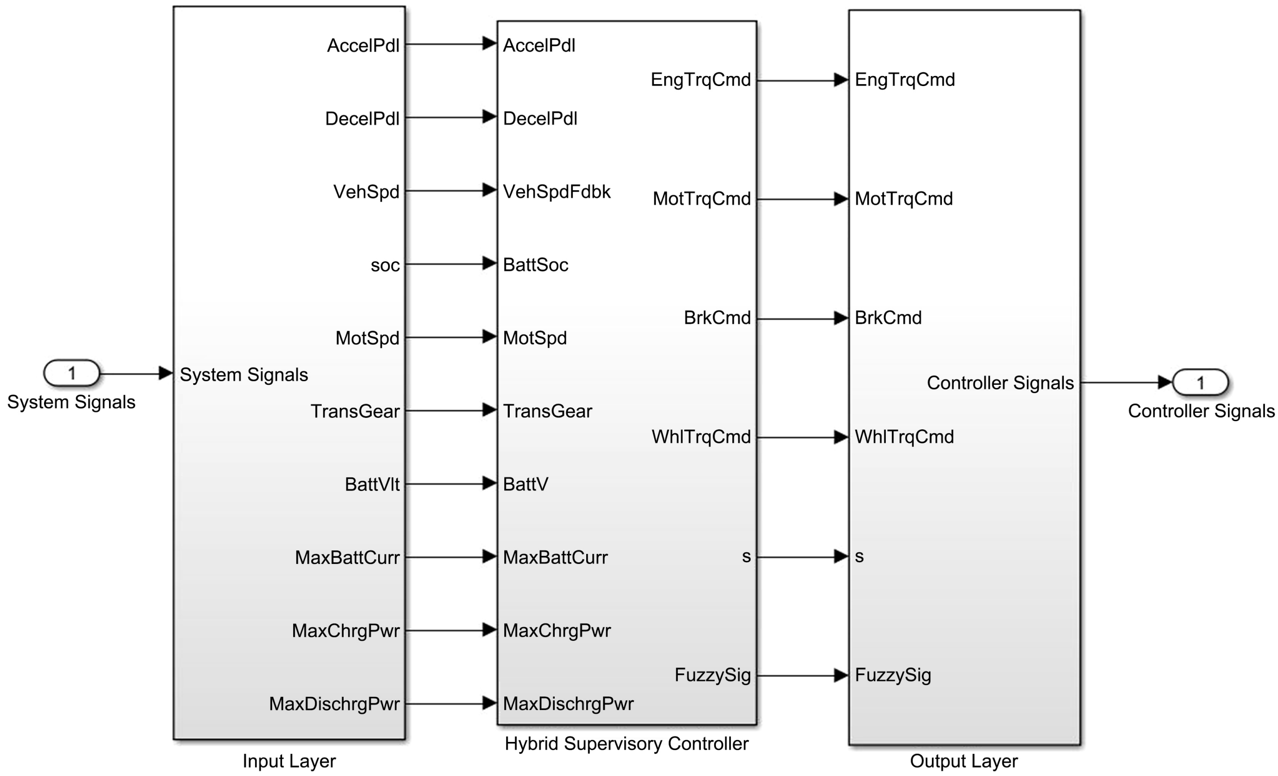

Figure 6. Controller overview.

The Energy Management system, shown below in **Figure 7**, 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 this work.

The Control Domain subsystem generates a predefined vector of possible wheel torque commands for the motor based on driver input. 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].

The Powertrain Constraints subsystem 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.

There are 3 different infeasible flags that can be set: An engine, motor, or battery 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

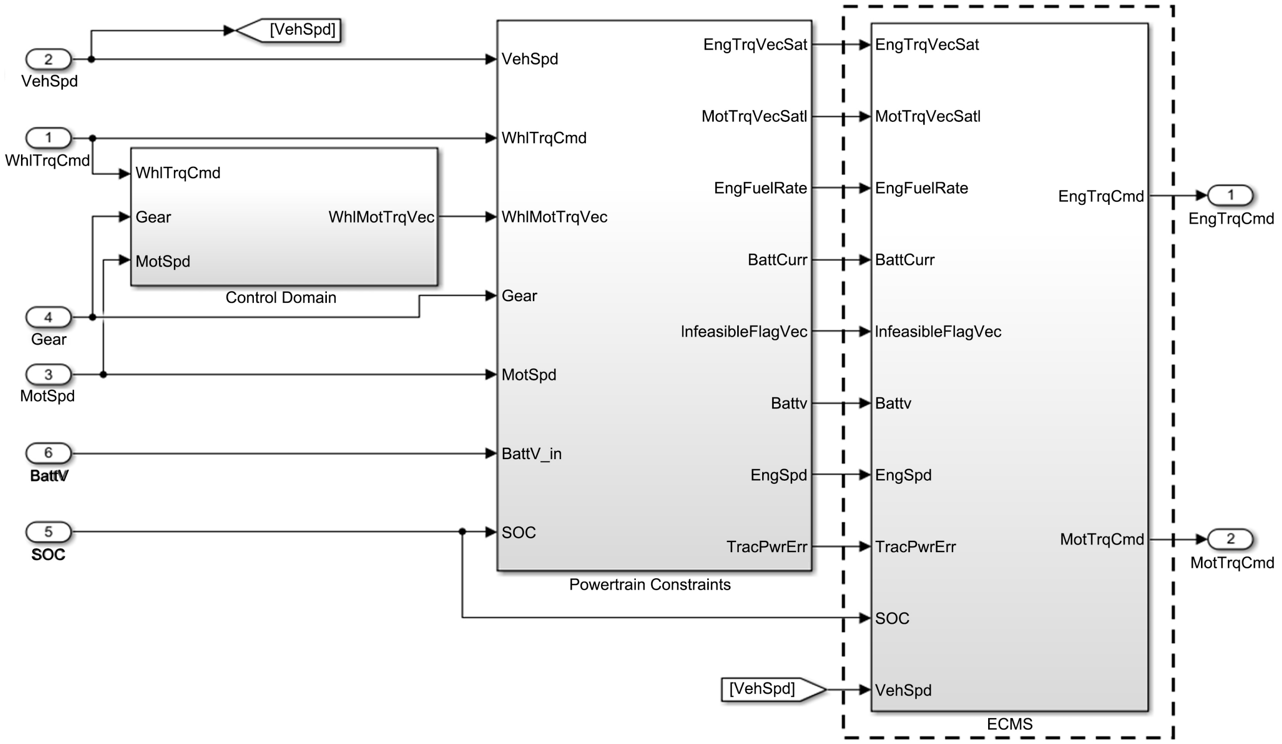

Figure 7. Energy management overview.

be set. 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. 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. 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 or 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 8**, 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. Each input is composed of a vector determined upstream in the Control Domain subsystem. The summation is as follows in Equation (10):

$$H = P_{fuel} + s * P_{batt} * p(\mathrm{SOC}) + Con_P + Trac_{PE} + P_{eng,RL} \quad (10)$$

where P_{fuel} is fuel power, s is the equivalence factor, P_{batt} is the associated battery power, $p(\mathrm{SOC})$ 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.

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 (a = 3 curve in **Figure 3**).

The equivalence factor, or "s", is typically a constant value determined offline but can be 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 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

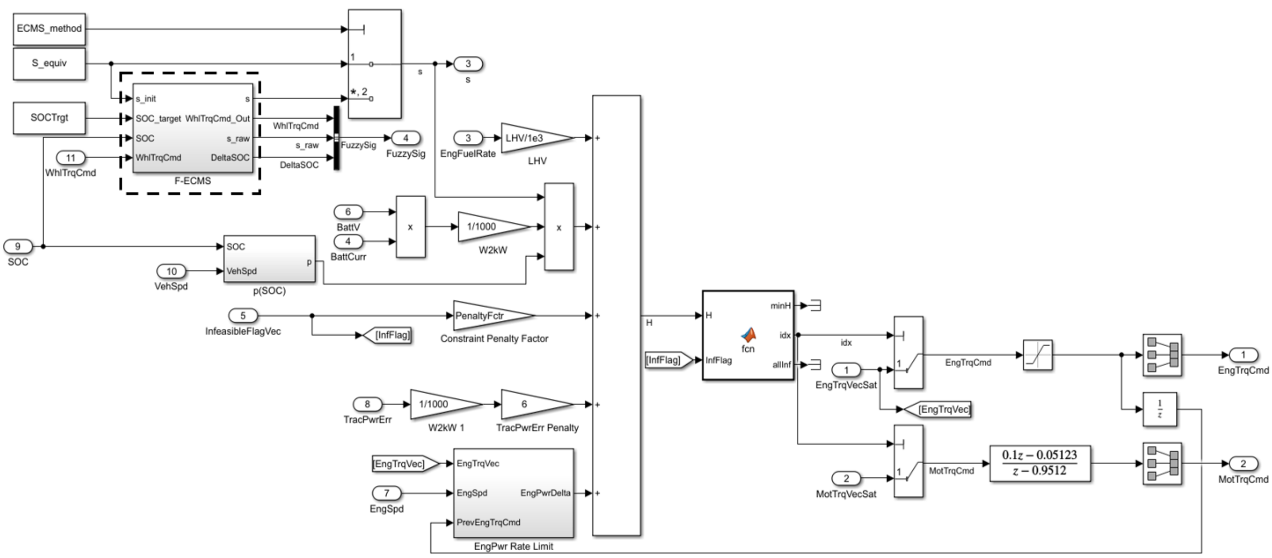

Figure 8. ECMS overview.

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 for the vector of the cost function. This creates an index value "idx" which is used downstream in the multiport switches to pass the desired torque command. 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 that is saturated between the minimum and maximum torque values.

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 and charge sustainability of the system. The fuzzy controller accepts two inputs: the difference in SOC between the actual and target (50%), and the driver wheel torque command shown below in Figure 9.

Membership functions were defined for both inputs using the Fuzzy Logic Controller which belongs to the Fuzzy Logic Toolbox in Simulink [16]. 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.

The delta SOC variable shown in Figure 10 was designed to operate in a range of +−10% around the target SOC of 50% satisfying the defined operating window of 40% - 60% SOC. Functions include large negative (LN), medium negative (MN), small negative (SN), zero (Z), small positive (SP), medium positive (MP) and large positive (LP).

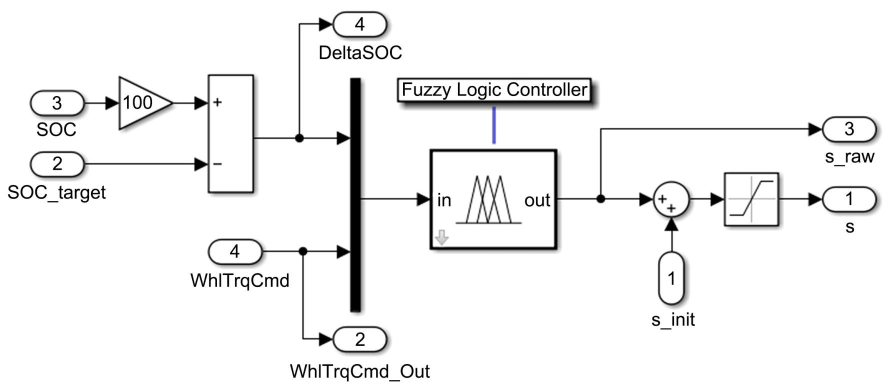

Figure 9. Fuzzy logic simulink controller.

The wheel torque input (Figure 11) 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. If the torque command exceeds 4100 Nm, the input function will default to VL.

The output membership function shown in Figure 12 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.

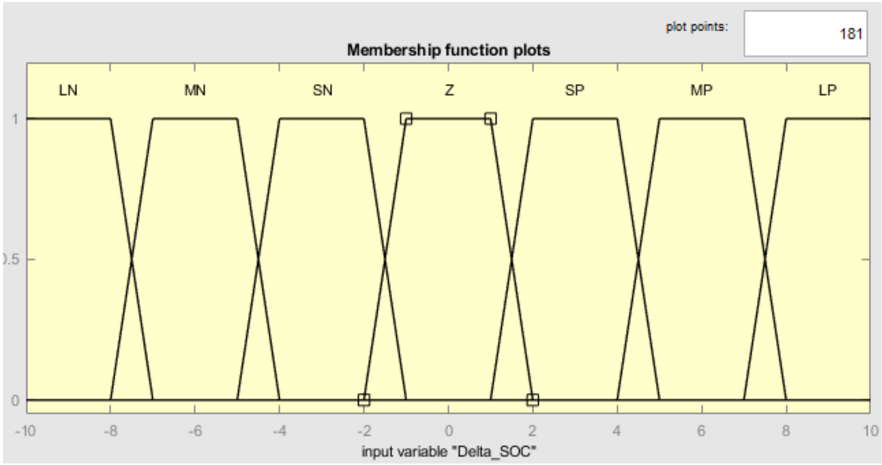

Figure 10. Delta SOC membership functions.

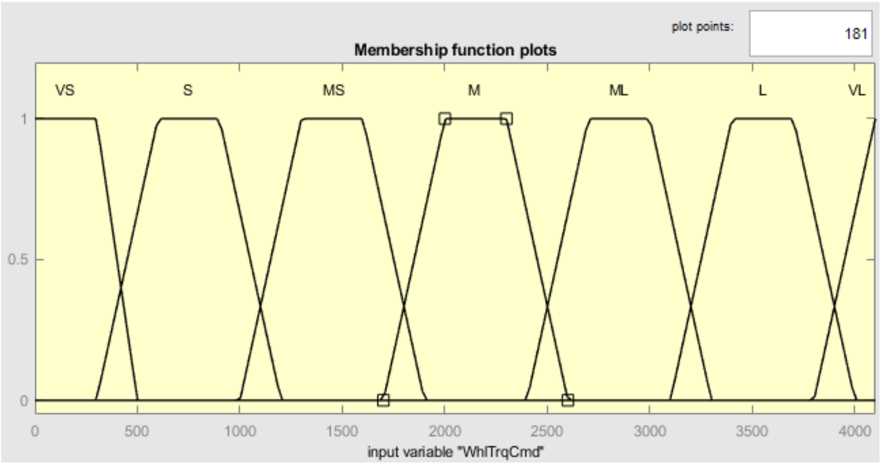

Figure 11. Wheel torque command membership functions.

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.

Figure 13 provides a high-level overview of the closed-loop system in the MIL environment with the integrated FLC. The flow of the diagram flows from left to right and top to bottom until each drive cycle is complete. Table 2 defines the inference rule matrix.

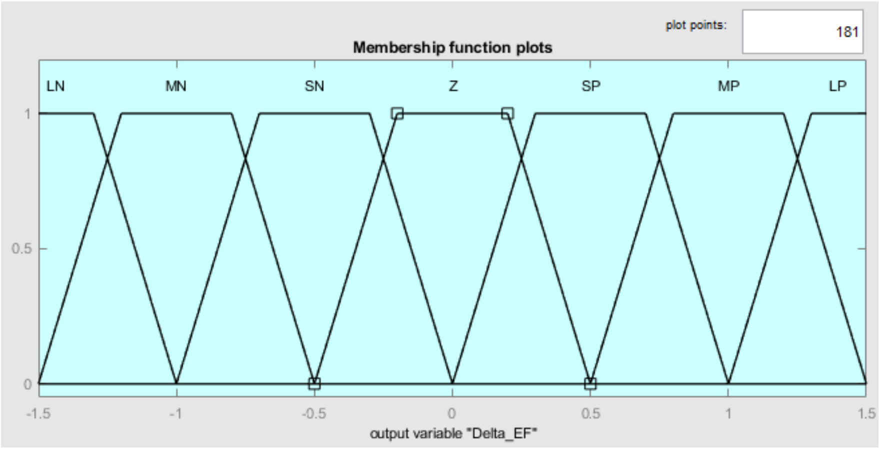

Figure 12. Delta EF membership functions.

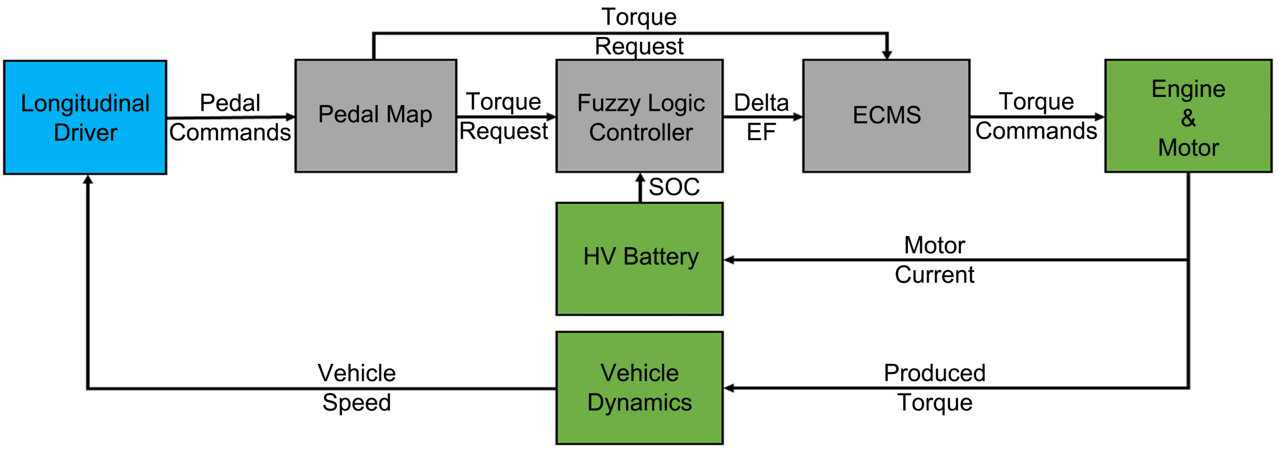

Figure 13. High-level control with integrated fuzzy logic controller.

Table 2. 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

5. Results

The results are 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 and is performed with a brute force approach. 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 is generalized. The fuzzy ECMS uses the same initial EF for each drive cycle and the additional FLC updates the EF. **Table 3** highlights the validation drive cycles and their respective characteristics.

Each drive cycle was cyclically run 4 times in Simulink to determine the optimum equivalence factor for each cycle at each equivalence factor. The selected range for the tested equivalence factor values spanned from 2 to 4 in 0.05 increments. 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 of 50%. **Table 4** provides the optimum EF for each drive cycle and the average over all cycles.

The average EF was used to determine charge sustainability over each drive cycle without tuning for the specific cycle. Each 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

Table 3. Validation drive cycles and characteristics.

Drive Cycle	Simulation Time (s)	Max Speed (m/s)	Avg. Speed (m/s)	Max Acceleration (m/s^2)	Min Acceleration (m/s^2)
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 4. Optimal equivalence factors.

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

EF is found for a starting SOC of 50%. 50 and 32.5% starting SOC values were selected to provide variability in the real-world when turning a vehicle on and driving, however, the 32.5% SOC scenario is used to show how the control system will react when starting a cycle from a very low SOC. Similarly, 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 to better mimic actual human driving response times.

Fuzzy ECMS

Results from the optimal and average ECMS 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. 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 where the starting SOC and driver styles were varied to test the robustness of the F-ECMS strategy.

6. Discussion

Table 5 below shows the fuel economy and SOC results for the brute force analysis discussed above. 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.

Table 5. Optimal ECMS results.

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

Drive cycles with mild accelerations and decelerations allow for more robust energy management from both powertrain components while 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. 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. 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. Charge correction has been applied to the fuel economy results shown in the table below.

Table 6 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.

Equation (11) is used to determine the percent difference between the average and fuzzy EF strategies.

$$\%_{Diff} = \frac{FE_F - FE_{Avg}}{FE_{Avg}} * 100 \quad (11)$$

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 7** 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.

Table 6. Fuel economy results in mpg for Avg. and fuzzy EF for aggressive and mild drivers at 50 and 32.5% initial SOC.

Drive Cycle	Aggressive 50%		Mild 50%		Aggressive 32.5%		Mild 32.5%	
	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 7. Total fuel economy % comparison.

Drive Cycle	Average Comparison				Optimal Comparison			
	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				

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 hypothetical possibility of using an adaptive equivalence factor to highlight differences in fuel economy.

The F-ECMS may not perform as well when looking at the percent difference compared to the optimal ECMS, however the best improvement was 4.37% during the EMC Highway cycle. 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 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.

Table 8 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. The average EF keeps the control strategy operating in a CS mode roughly 73% of the time. The drive cycles that fail 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%.

Table 8. Average and fuzzy ECMS ending SOC comparison.

Drive Cycle	Aggressive 50%		Mild 50%		Aggressive 32.5%		Mild 32.5%	
	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

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, 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. 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 also help to pinpoint errors in a constant or average EF.

6.1. 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%. 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 14**.

Figure 15 and **Figure 16** provide simulation results for a starting SOC of 50%. The optimal CS EF was found for the best fuel economy and as seen in **Figure 15**, 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

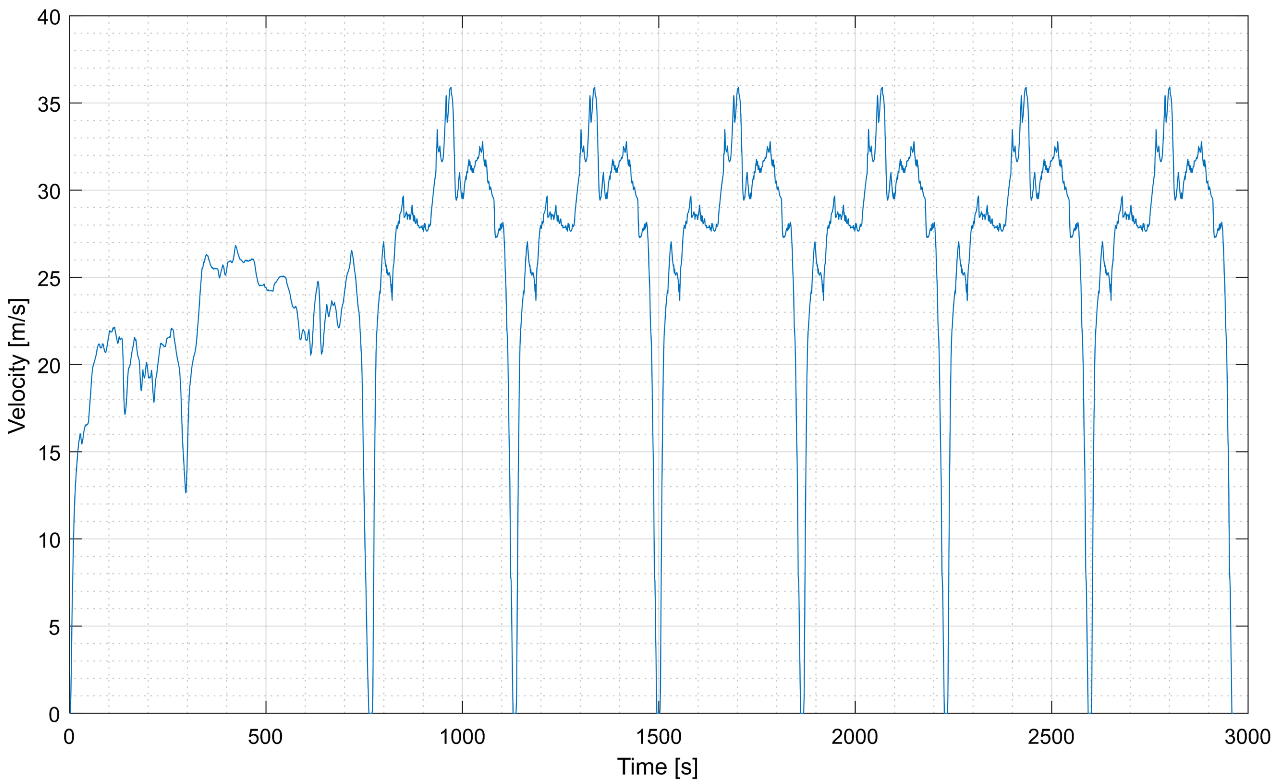

Figure 14. EMC highway drive cycle.

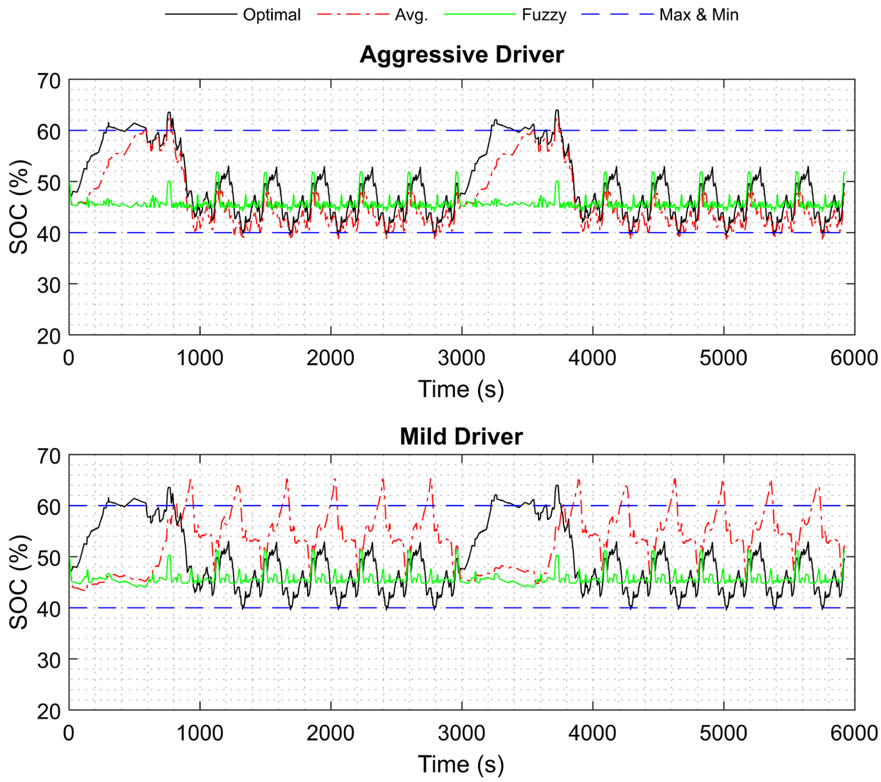

Figure 15. EMC highway 50% SOC comparison.

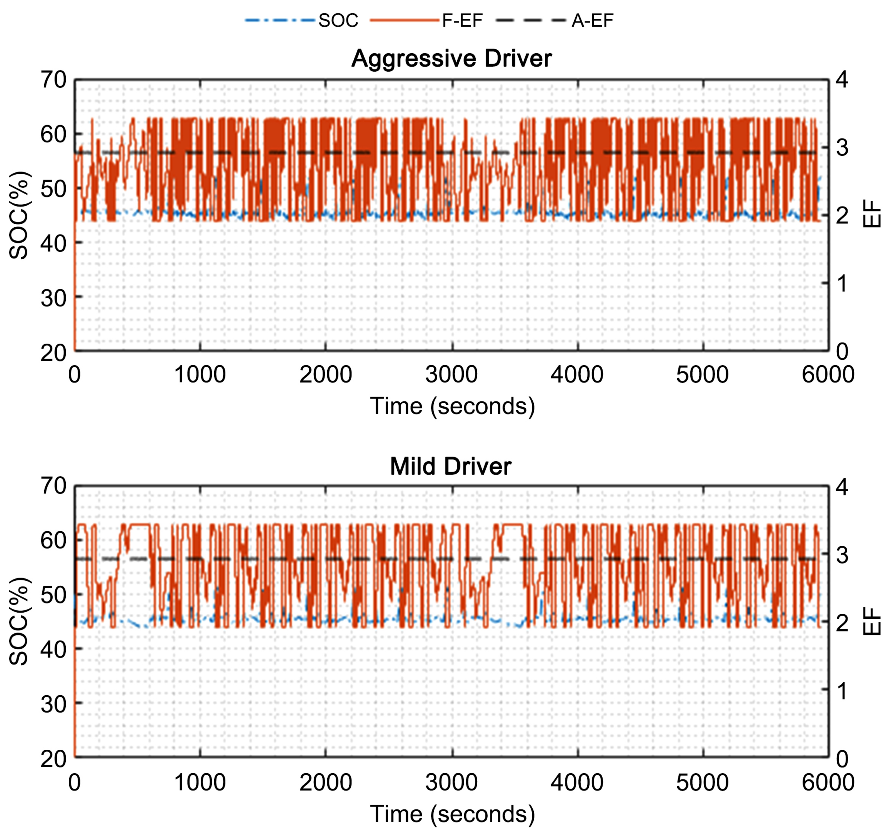

Figure 16. EMC highway 50% EF comparison.

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. 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. This is due in part to the nature of the drive cycle, with aggressive accelerations and decelerations that cause the SOC to fluctuate rapidly. When this happens, the algorithm must be tuned offline to combat the changes. However, with the F-ECMS, the fluctuations in SOC are not drastic due to the algorithm adapting to the driver torque demands on both the accelerations up to the maximum speed and back down to 0 mph. This can be viewed better for the case of the Mild driver below where the average ECMS consistently violates the upper SOC bound, but the F-ECMS does not. Fluctuations in the EF are due to the rapid changes in accelerator and decelerator pedal from the driver. During simulations, the driver will try to match the speed profile exactly, which can cause the algorithm to adjust to varying positive and negative torque requests at each timestep.

Figure 17 and Figure 18 provide the SOC and EF results for the EMC Highway drive cycle with a starting SOC of 32.5%. Results are similar to those shown above where the F-ECMS updates every timestep to match the response from the driver following the speed profile which contributes to a smoother SOC profile that does not violate the min/max threshold. The EF for the F-ECMS algorithm reaches a maximum at the beginning of the cycle due to the large acceleration request from the driver and low starting SOC. This causes the algorithm to raise

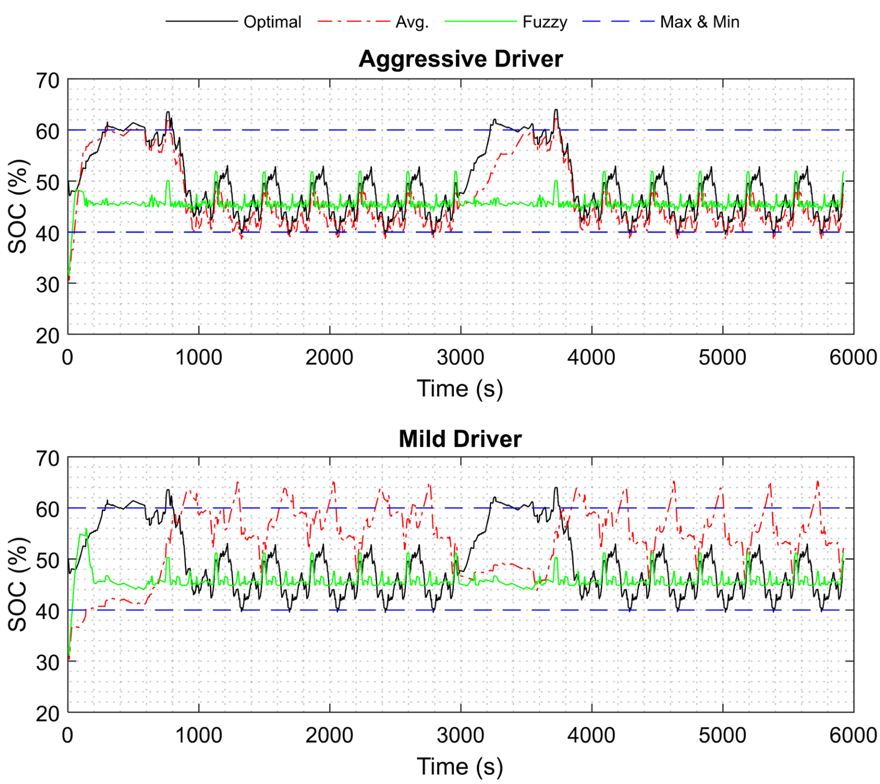

Figure 17. EMC highway 32.5% SOC comparison.

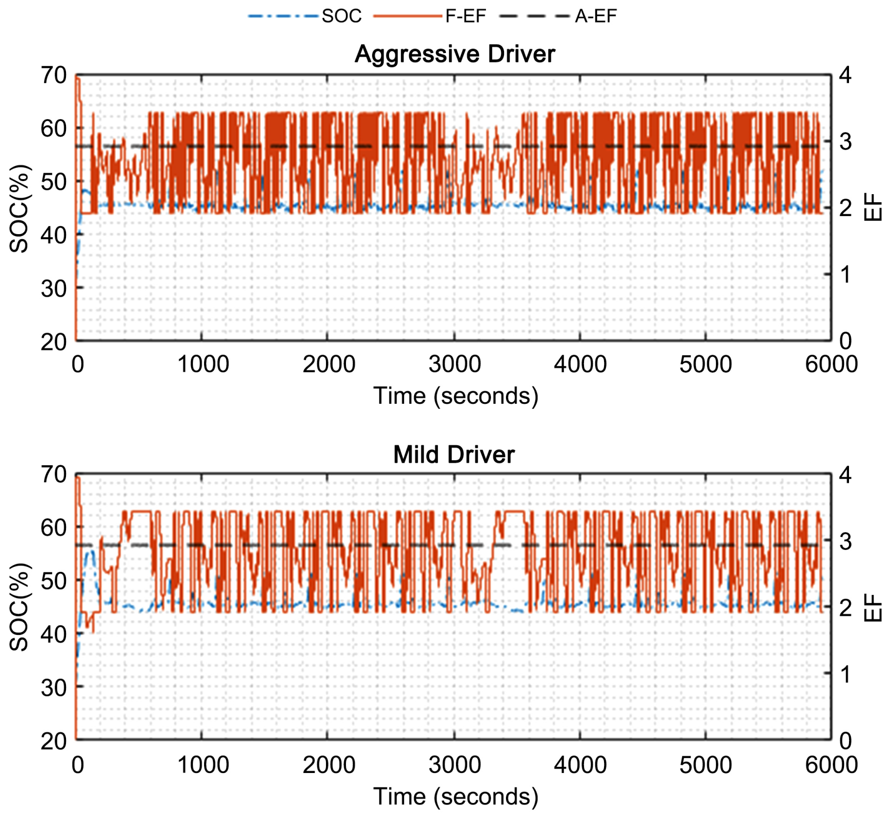

Figure 18. EMC highway 32.5% EF comparison.

the SOC above the minimum threshold before adjusting based on a lower acceleration request from the driver. In a similar fashion to the results shown above, the EF updates rapidly for the aggressive driver in response to following the speed profile with as little error as possible. While this does not represent real-world driving profiles, it does support the basis of charge sustainability without the need to tune the algorithm for individual driving characteristics.

6.2. EMC City Cycle

The EMC City drive cycle/speed profile is shown below in **Figure 19**.

The F-ECMS may not have improved fuel economy during simulations, however, substantial improvements for charge sustainability and SOC stability were performed. **Figure 21** and **Figure 20** provide results with a starting SOC of 50% while **Figure 22** and **Figure 23** provide the 32.5% starting SOC results for the EMC City drive cycle.

In **Figure 21**, the F-ECMS adjusts the EF to maintain a rough average of 45% SOC. 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. When the EF is set higher than needed for a drive cycle, the motor will not be favored in launching events. During decelerations, energy is captured, however, because the motor is not favored in certain instances, the SOC does not deplete, leading to the CS operations

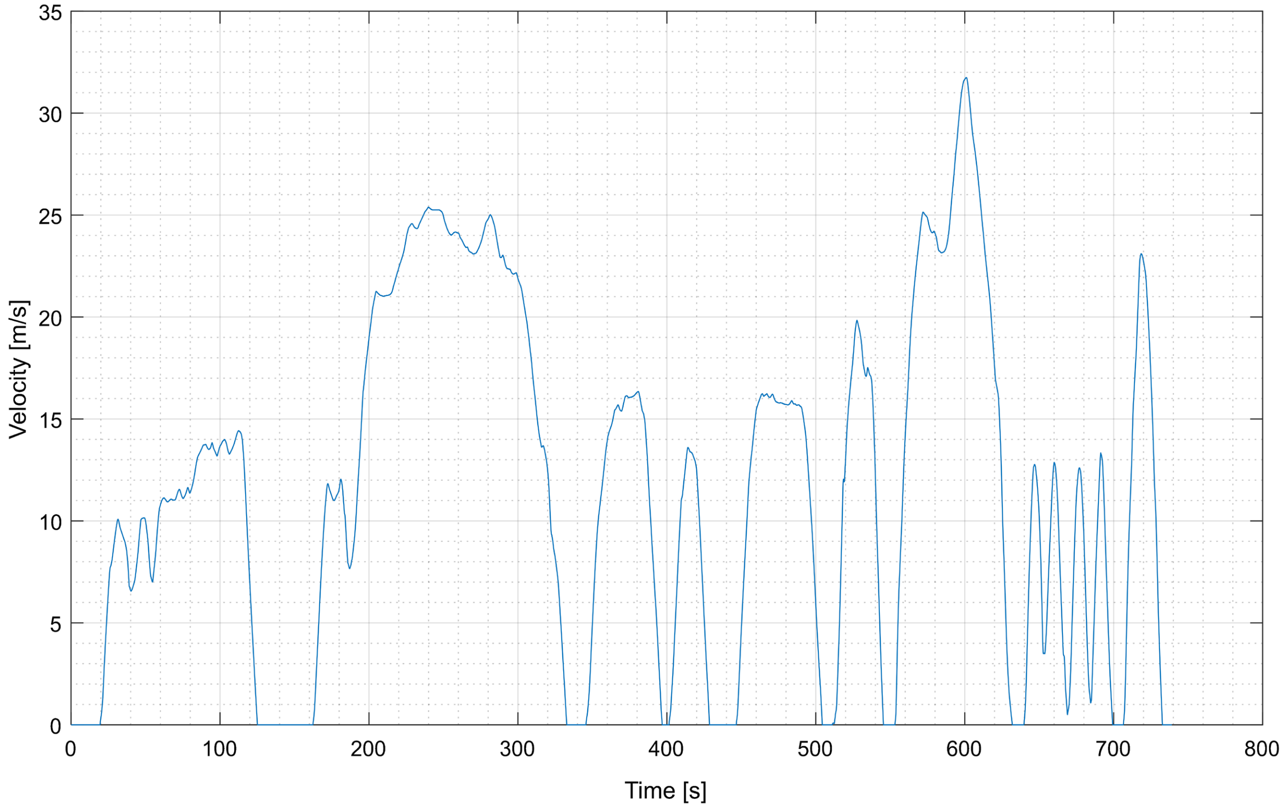

Figure 19. EMC city drive cycle.

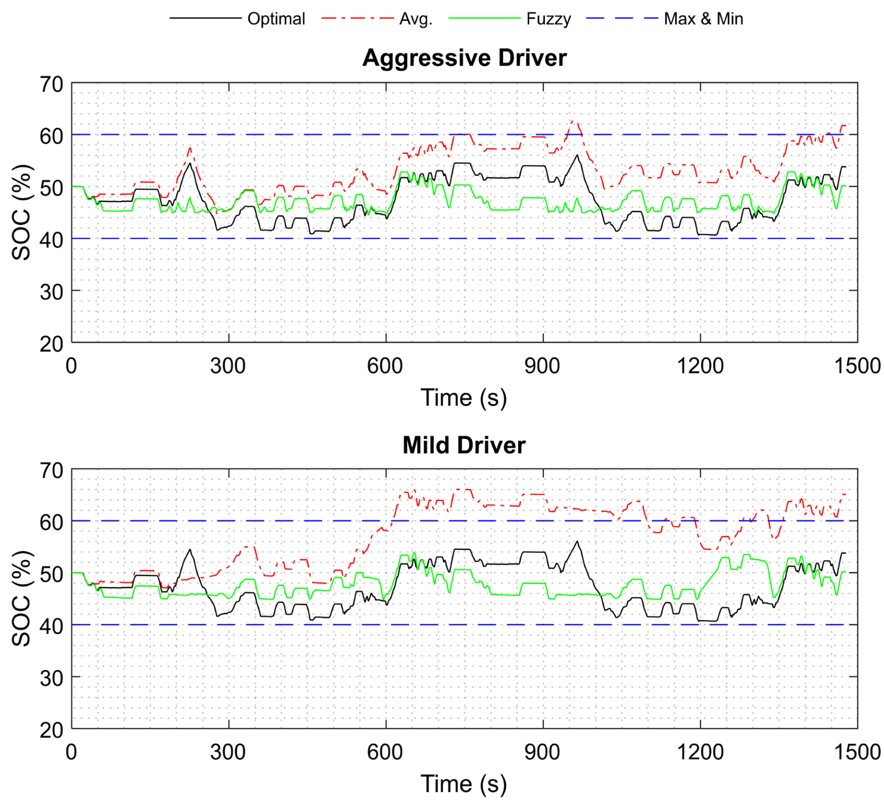

Figure 20. EMC city 50% EF comparison.

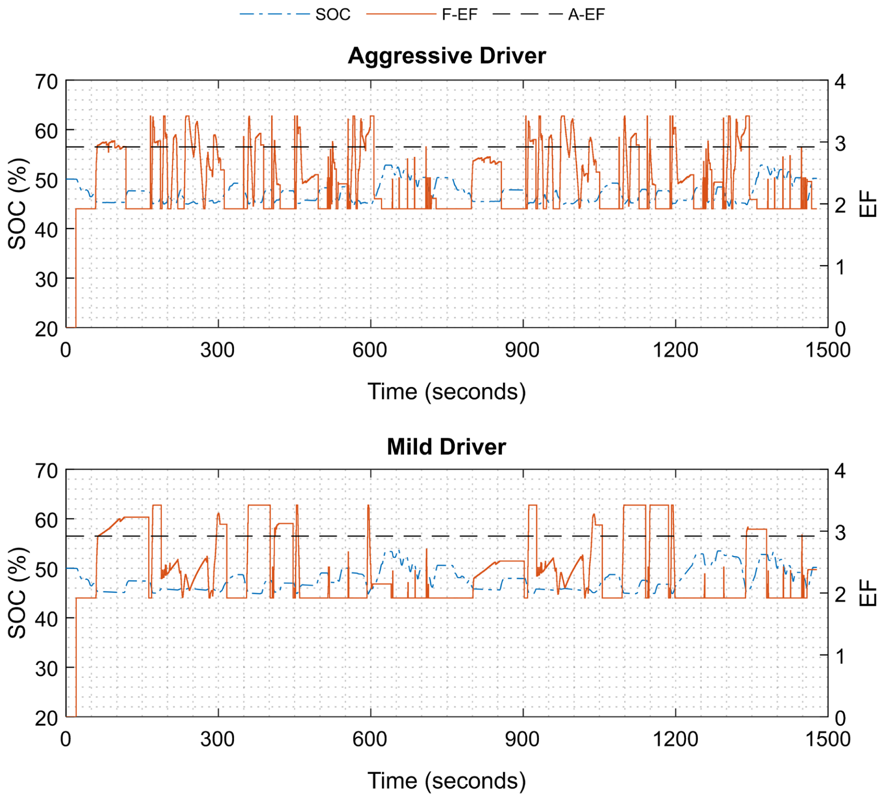

Figure 21. EMC city 50% SOC comparison.

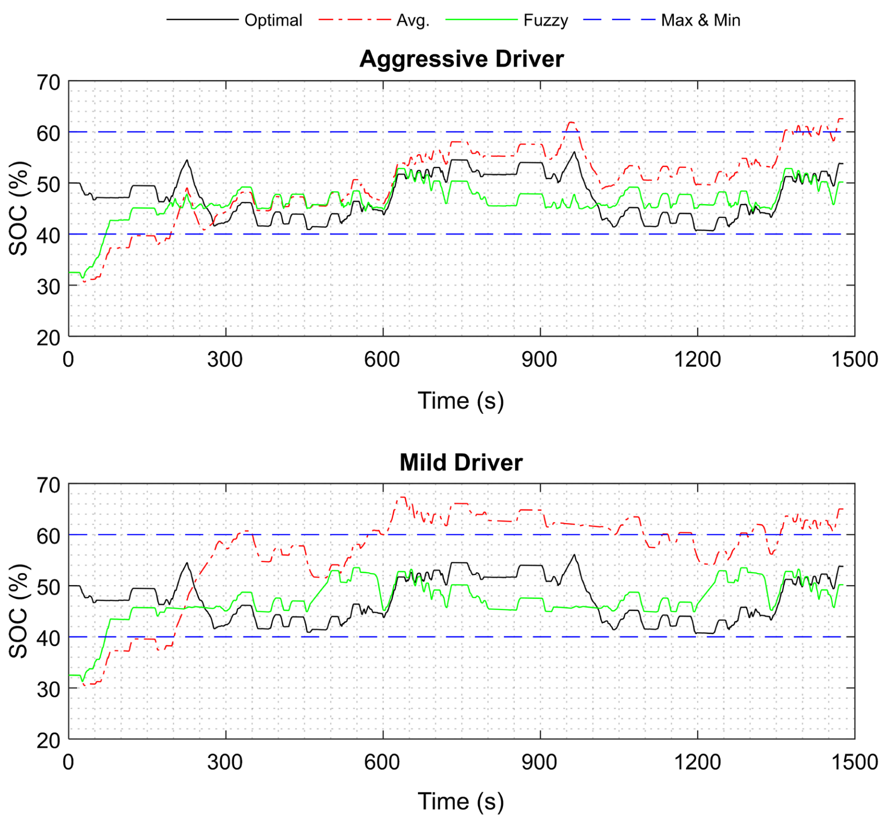

Figure 22. EMC city 32.5% SOC comparison.

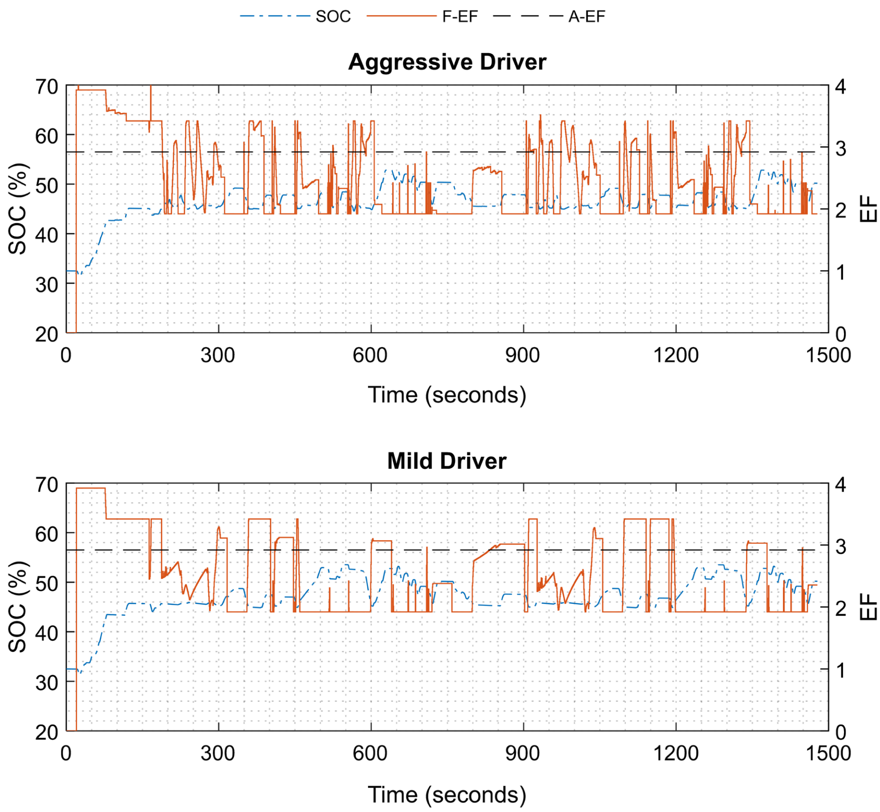

Figure 23. EMC city 32.5% EF comparison.

for both drivers. The F-ECMS adjusts the EF at the beginning of each cycle to compensate for acceleration requests from the driver. These responses help the SOC follow a trend similar to the optimal for the EMC City cycle, indicating that the F-ECMS achieves close to optimal results by updating the EF from driver inputs to the system.

Figure 20 highlights the effectiveness of the F-EF to maintain CS operations over the course of the drive cycle.

Figure 22 and Figure 23 showcase the ability of the F-ECMS to adapt to a lower starting SOC and retain CS operations within the first 100 seconds of the drive cycle. A large regenerative event at 200 seconds in the cycle causes the average ECMS to raise the SOC above the maximum threshold due to the untuned EF. Harder accelerations at the beginning also cause the delay of an SOC increase for the average ECMS while the F-ECMS is able to raise the SOC above the minimum threshold within the first 75 seconds. Once above this threshold, the F-ECMS adapts to the driver demands to follow an SOC curve similar to the one determined through the brute force analysis. Lower speeds in the cycle contribute to a lower EF average for the F-ECMS, with the average maintaining roughly 2.1 over the cycle.

7. Limitations

While the above results show promise that an adaptive strategy can be implemented to improve vehicle efficiency and charge sustainability, there can be drawbacks introduced into the system. Fuzzy logic shows promise for tuning a system to specifications or functions; however, some background knowledge is required in order to tune the system efficiently. The same results can be obtained if prior knowledge is not used, but this can make the end goal longer to reach, costing more time or money in the process. Similarly, the intended algorithm is very computationally expensive, which will require implementation on a stronger processor to run in real-time. For this reason, automotive companies may opt to use rules-based algorithms to reduce the hardware and essentially, vehicle costs for consumers.

8. 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. Throughout this work, multiple conclusions and recommendations were identified.

- Performance of the ECMS is dependent on adjustment of the EF.
- Each drive cycle and driving style require a different EF.
- The implemented fuzzy logic controller provides a robust method for updating the EF.
- Additional drive cycles should be analyzed that provide a wider range of optimal EFs.
- Adjusting the output of the EF to control the raw EF will help reduce the amount of prior EF calculations and modeling.
- The addition of a sliding time window to gradually adjust the EF rather than at each timestep will help with lessening oscillations in the EF.
- Modifying the FLC to prioritize regenerative braking will increase efficiency of the algorithm and reduce losses through the mechanical brakes.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] U.S. Environmental Protection Agency (2021) Sources of Greenhouse Gas Emissions. https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions#:~:text=Carbondioxide(CO2)makes,naturalgas%2Ctoproduceelectricity

[2] U.S. Department of Energy (2021) Emissions from Hybrid and Plug-In Electric Vehicles. https://afdc.energy.gov/vehicles/electric_emissions.html

[3] (2021) HEV Types. http://autocaat.org/Technologies/Hybrid_and_Battery_Electric_Vehicles/HEV_Types/

[4] Onori, S., Serrao, L. and Rizzoni, G. (2016) Hybrid Electric Vehicles: Energy Management Strategies. In: Gan, W.-S., Jay Kuo, C.-C., Zheng, T.F. and Barni, M., Eds., *SpringerBriefs in Electrical and Computer Engineering*, Springer, London, 121. https://doi.org/10.1007/978-1-4471-6781-5

[5] Mull, A., Nix, A., Perhinschi, M., Wayne, W., Diethorn, J. and Dunnuck, D. (2022) Powertrain Fuel Consumption Modeling and Benchmark Analysis of a Parallel P4 Hybrid Electric Vehicle Using Dynamic Programming. *Journal of Transportation Technologies*, **12**, 804-832. https://doi.org/10.4236/jtts.2022.124045

[6] Bouvier, H., Colin, G. and Chamaillard, Y. (2015) Determination and Comparison of Optimal Eco-Driving Cycles for Hybrid Electric Vehicles. 2015 *European Control Conference* (*ECC*), Linz, 15-17 July 2015, 142–147. https://doi.org/10.1109/ECC.2015.7330536

[7] Sciarretta, A., Back, M. and Guzzella, L. (2004) Optimal Control of Parallel Hybrid Electric Vehicles. *IEEE Transactions on Control Systems Technology*, **12**, 352-363. https://doi.org/10.1109/TCST.2004.824312

[8] Zhang, F., Xi, J. and Langari, R. (2017) Real-Time Energy Management Strategy Based on Velocity Forecasts Using V2V and V2I Communications. *EEE Transactions on Intelligent Transportation Systems*, **18**, 416-430. https://doi.org/10.1109/TITS.2016.2580318

[9] Jeon, S., Jo, S., Park, Y. and Lee, J. (2002) Multi-Mode Driving Control of a Parallel Hybrid Electric Vehicle Using Driving Pattern Recognition. *Journal of Dynamic Systems, Measurement and Control*, **124**, 141-149. https://doi.org/10.1115/1.1434264

[10] Harris, T.P., Nix, A.C., Perhinschi, M.G., Wayne, W.S., Diethorn, J.A. and Mull, A.R. (2021) Implementation of Radial Basis Function Artificial Neural Network into an Adaptive Equivalent Consumption Minimization Strategy for Optimized Control of a Hybrid Electric Vehicle. *Journal of Transportation Technologies*, **11**, 471-503. https://doi.org/10.4236/jtts.2021.114031

[11] Babuska, R. and Mamdani, E. (2008) Fuzzy Control. *Scholarpedia*, **3**, 2103. https://doi.org/10.4249/scholarpedia.2103

[12] Perhinschi, M. (2018) Fuzzy Logic. Master's Thesis, West Virginia University, Morgantown.

[13] Zhang, F., Liu, H., Hu, Y. and Xi, J. (2016) A Supervisory Control Algorithm of Hybrid Electric Vehicle Based on Adaptive Equivalent Consumption Minimization Strategy with Fuzzy PI. *Energies*, **9**, Article 919. https://doi.org/10.3390/en9110919

[14] Wang, S., Huang, X., Lopez, J.M., Xu, X. and Dong, P. (2019) Fuzzy Adaptive-Equivalent Consumption Minimization Strategy for a Parallel Hybrid Electric Vehicle. *IEEE Access*, **7**, 133290-133303. https://doi.org/10.1109/ACCESS.2019.2941399

[15] Denis, N., Dubois, M.R. and Desrochers, A. (2015) Fuzzy-Based Blended Control for the Energy Management of a Parallel Plug-in Hybrid Electric Vehicle. *IET Electrical Systems in Transportation*, **9**, 30-37. https://doi.org/10.1049/iet-its.2014.0075

[16] The Math Works (2023) Fuzzy Logic Toolbox. https://www.mathworks.com/help/fuzzy/index.html?searchHighlight=fuzzylogictoolbox&s_tid=srchtitle