Article

Data-Driven Adaptive Equivalent Consumption Minimization Strategy for Hybrid Electric and Connected Vehicles

Wilson Pérez * , Punit Tulpule , Shawn Midlam-Mohler and Giorgio Rizzoni

Center for Automotive Research, Department of Mechanical and Aerospace Engineering,
The Ohio State University, Columbus, OH 43212, USA; tulpule.3@osu.edu (P.T.);
midlam-mohler.1@osu.edu (S.M.-M.); rizzoni.1@osu.edu (G.R.)

* Correspondence: perez.430@osu.edu

Abstract: Advanced energy management strategies (EMS) are used to control the power flow through a vehicle's powertrain. However, the cost of high-power computational hardware and lack of a priori knowledge of future road conditions poses difficult challenges for engineers attempting to implement globally optimal frameworks. One solution is to use advanced driver assistance systems (ADAS) and connectivity to obtain a prediction of future road conditions. This paper presents a look-ahead predictive EMS which combines approximate dynamic programming (ADP) methods and an adaptive equivalent consumption minimization strategy (A-ECMS) to obtain a near-optimal solution for a future prediction horizon. ECMS is highly sensitive to the equivalence factor (EF), making it necessary to adapt during a trip to account for disturbances. A novel adaptation method is presented in this work which uses an artificial neural network to learn the nonlinear relationship between a speed and the state of charge (SOC) trajectory prediction obtained from ADP to estimate the corresponding EF. A traffic uncertainty analysis demonstrates an approximately 10% fuel economy (FE) improvement over traditional A-ECMS. Using a data-driven adaptation method for A-ECMS informed by a dynamic programming (DP) based prediction results in an EMS capable of online implementation.

Keywords: electrified powertrain energy management; connected and autonomous vehicles; eco-driving; optimal control; artificial neural networks

Citation: Pérez, W.; Tulpule, P.; Midlam-Mohler, S.; Rizzoni, G. Data-Driven Adaptive Equivalent Consumption Minimization Strategy for Hybrid Electric and Connected Vehicles. *Appl. Sci.* **2022**, *12*, 2705. <https://doi.org/10.3390/app12052705>

Academic Editors: Daniela Anna Misul and Ezio Spessa

Received: 3 February 2022

Accepted: 2 March 2022

Published: 5 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Government regulations worldwide are pushing toward clean, renewable energy to reduce pollution levels and create a sustainable future. In response, the automotive industry has invested heavily in innovative technology such as electrified powertrains. Electrification of powertrains provides attractive benefits over conventional powertrains, including engine downsizing, electric motor assist, and engine start-stop. Similarly, the automation of driving promises safer driving; hence, the automotive industry is embracing connected and automated vehicle (CAV) technology. A CAV uses advanced on-board sensing technology to obtain information about surroundings (commonly referred to as look-ahead data) and make informed decisions about present and future actions. Perception and control algorithms, such as sensor fusion and adaptive cruise control (ACC), leverage this information to make vehicles safer and more efficient.

The integration of electrification and driving automation has the potential of extending the benefits of both these technologies. However, the additional degrees of freedom result in a complex system which demands more advanced control methods. To achieve maximum powertrain efficiency, researchers and automotive manufacturers have dedicated significant effort to the development and advancement of energy management strategies (EMS) [1]. For hybrid electric vehicle (HEV) powertrain control, an EMS is a set of algorithms that decide the power split between the internal combustion engine and the EM. The goal of the EMS is

to improve fuel economy (FE) and optimize the performance of HEVs [2]. In [3], a discrete dynamic programming (DP) approach is used to optimize torque split, gear shifting, and velocity trajectory for a parallel HEV. A model predictive controller (MPC) is used in [4] to optimize the velocity profile and torque split with a gradient-based optimization. In [5], DP with an underlying adaptive equivalent consumption minimization strategy (ECMS) is used to optimize torque split and velocity trajectory. Although energy management of HEVs and CAVs has been the focus of many recent publications, the use of look-ahead data for optimal energy management remains an open research topic.

A key component of a look-ahead EMS is the use of vehicle-to-everything (V2X) communication. The authors in [6,7] showed how vehicle-to-infrastructure (V2I) and GPS signals can be used as inputs to a perception model to employ an artificial neural network (ANN) to characterize driving patterns in an optimal EMS for a plug-in HEV (PHEV). A traffic model was applied in [8] to predict future vehicle velocity with vehicle-to-vehicle (V2V) and V2I as inputs, and an optimal EMS was derived using Pontryagin's minimum principle (PMP). In [9], traffic light signal phase and timing (SPaT) was used to give dynamic speed advice to the driver and demonstrated a FE improvement of 12–14%.

Although the main focus of recent development in optimal energy management has been on look-ahead data-based optimal control, the key concern stems from the computational requirements of global optimization algorithms. The implementation of global optimization algorithms on microprocessors available on board a typical passenger car is challenging due to various reasons, including the high cost of hardware [1]. Research programs such as FutureTruck, EcoCAR, and NEXTCAR were designed to address these issues. A team of students and faculty at Ohio State University (OSU) implemented ECMS on a model-year 2000 Chevrolet Suburban SUV as part of the FutureTruck 2000 competition and achieved approximately 1.5 times the FE of the production Suburban [10]. More recently, the OSU NEXTCAR team implemented a model predictive control framework on a test vehicle using principles of approximate dynamic programming (ADP) with various look-ahead data sources. Real-world testing was conducted on a test track and resulted in more than 20% FE improvement over the base vehicle control strategy [11].

Many different methods to overcome the challenges of implementing global optimization algorithms are proposed in the literature. In [12], a DP-based EMS is used in a predictive control scheme to feed a conventional cruise controller with new set points on a heavy duty truck. The authors demonstrate real-world fuel consumption reduction for highway driving. In [13], DP optimization is done using cloud computing for optimizing the velocity reference that is tracked by a human driver. The authors of [14] reduced computational cost by separately formulating the car-following control and energy management strategy and then coupling the control laws in a MPC framework. Online implementation was demonstrated in hardware-in-the-loop experiments.

In recent studies, the authors in [15] proposed an intelligent ECMS, where a Bayesian regularization neural network was used to predict the near-optimal equivalence factor (EF), and a back propagation neural network was used to forecast the engine on/off state to improve the quality of the EF prediction. In [16], a two-layer strategy was used where the upper layer extracts trip and vehicle information using CAV technology and solves DP on a remote server. The lower level controller was based on Q-learning and was used for real-time implementation of the optimal result obtained from DP. In [17], the authors developed a hybrid deep Q-learning and policy gradient method for vehicles driving along multi-lane urban signalized corridors to learn well-established longitudinal fuel-saving strategies. Numerical experiments demonstrate up to a 46% fuel reduction compared to their baseline model. The authors of [18] proposed a look-ahead horizon-based co-optimization of vehicle dynamics and powertrain operation to minimize energy consumption with the constraint of driving safety to avoid traffic collisions. The optimal solution was derived by PMP and verified in a traffic-in-the-loop simulation environment.

Most of the past literature assumes no stochastic uncertainties in the prediction of future events. However, with existing hardware limitations, it is unrealistic to implement

the stochastic optimal control methods. The current work proposes a method of using CAV technology to improve the energy efficiency of HEVs with a clear path to on-board implementation. The novelty of the work lies in the approach that combines machine learning, optimal control and traditional ECMS and concurrently optimizes the vehicle speed profile as well as EMS. The proposed approach is computationally efficient and provides better FE gains than ECMS. The primary objective is to quantify the potential for FE improvements using a look-ahead predictive EMS. An approach is outlined for using look-ahead data to obtain a prediction of future vehicle state trajectories (e.g., vehicle speed and battery state of charge) for a single vehicle using a DP-based rollout algorithm. A novel EF adaptation technique for ECMS is introduced, and the effect of traffic uncertainty on FE is measured through statistical methods. In a simulation study, we demonstrate about 10% FE gains over a baseline consisting of a combination of A-ECMS and the intelligent driver model. Recently, an MPC-based approach was shown to provide 20% FE gains over a rule-based EMS [11] but required higher computation time. Considering that the baseline in [11] was inferior to A-ECMS and required more computational resources, the FE gains obtained in this study are significant.

The remainder of this paper first outlines the modeling framework and methodology used to formulate the proposed EMS. This is followed by a discussion of simulation results and key findings. Finally, a conclusion of the study is provided.

2. Materials and Methods

This section begins with the general architecture of the proposed EMS. The vehicle architecture and modelling used throughout this study are then described. Each block of the EMS is then explained in detail.

2.1. Proposed Architecture of EMS

The overall structure of the EMS presented in this paper, referred to as DP-ECMS, is shown in Figure 1. It relies on principles of optimal control to minimize fuel consumption considering look-ahead data. For this research, look-ahead data is a combination of traffic flow control devices (i.e., traffic light location and SPaT, stop signs) and road characteristics (i.e., speed limits, average traffic flow speed) obtained from sensors, including radar, camera, dedicated short range communication (DSRC), and GPS. All the look-ahead data is simulated and therefore assumed to be available at all times with no delays in data transmission. All modeling and simulation in this study is done using MATLAB and Simulink R2019b.

Look-ahead data is used to obtain a speed limit prediction for the ego vehicle based on upcoming road conditions. This is done by layering different look-ahead data sources and finding the maximum safe speed. The average traffic flow speed is first compared to the legal speed limit. The effective speed limit is taken to be the minimum of the two. If the vehicle is in range of a connected intersection, SPaT is used to determine if the vehicle can pass on green. A simple rule-based method is used to find the speed needed to pass during a green window, where the speed is calculated by dividing the distance to the traffic light by the time left in the current phase. If the speed to pass on green does not violate the bounds set by traffic or the legal speed limit, it is taken as the effective speed limit. If speed constraints are violated, the effective speed limit at the traffic light is set to zero and the vehicle will be forced to stop. If there is no traffic and no traffic lights, the effective speed limit is simply the legal speed limit.

Figure 1. Proposed EMS structure.

The EMS shown in Figure 1 uses a modular control structure, which can be broken down into three key phases: the rollout algorithm-based MPC, a supervisory control and an EF predictor. The MPC uses the future speed limits obtained from the road data, SPaT and traffic to optimize the vehicle speed and battery SOC. It generates an optimal feedback control law for each control input, namely the total wheel torque and the torque split between the EM and internal combustion (IC) engine. The nonlinear MPC that uses DP as a solver cannot be executed at the rate of the powertrain controller due to computation time limitations. Additionally, the feedback law is not robust to stochastic uncertainties due to traffic. Hence, a supervisory controller is designed that uses the optimal feedback law obtained from nonlinear MPC in combination with adaptive ECMS. The supervisory controller uses the optimal feedback law to compute total wheel torque, but uses ECMS to compute the torque split ratio. Since there is no optimization in the supervisory controller, it can be executed at the rate of a typical powertrain controller. The key issue is that the ECMS requires an optimal equivalency factor that depends on the speed profile. Hence, a data-driven EF predictor is used which is based on the optimal speed and SOC trajectories obtained from a closed-loop simulation over the prediction horizon. An ANN is trained to generate an optimal EF for the expected future speed profile. It should be noted that all three phases are executed at different time rates. The supervisory controller is executed at the rate of 100 Hz. The MPC is executed once every 10 s, and the EF prediction is executed once every 30 s.

2.2. Powertrain Model

The work presented in this paper is motivated by the EcoCAR Mobility Challenge (MC) competition objectives and is designed to serve as the primary energy management

system for the OSU competition vehicle. EcoCAR MC is a 4-year competition which challenges university students to apply advanced propulsion systems, electrification, SAE Level 2 automation, and vehicle connectivity to improve the energy efficiency, safety and consumer appeal of the competition-provided vehicle [19]. The vehicle in consideration is a 2019 Chevrolet Blazer with a through-the-road parallel hybrid architecture consisting of a General Motors (GM) 2.0 L turbocharged gasoline engine coupled with a DENSO 32 kW belted alternator-starter (BAS) for front powertrain propulsion and a 90 kW Parker Hannifin GVM210-100 electric machine mated to a BorgWarner eGearDrive rear-axle single speed gearbox to power the rear axle. A custom 3.5 kW-h Li-ion battery pack provided by Hybrid Design Services (HDS) serves as the electrical energy storage system (ESS). The vehicle powertrain architecture is highlighted in Figure 2.

Figure 2. Vehicle architecture.

In this paper we assume SI units for all model parameters. For powertrain energy analysis, it is sufficient to model the vehicle as a point mass and express longitudinal dynamics as the equilibrium of forces acting on the vehicle. The equation of motion for a point mass can be derived from the equilibrium of forces acting on a vehicle:

$$M_{veh} \frac{dv_{veh}}{dt} = \frac{T_{pwt} - T_{brake}}{R_{wh}} - c_{roll} M_{veh} g \cos \delta - \frac{1}{2} \rho_{air} A_f C_d v_{veh}^2 - M_{veh} g \sin \delta \quad (1)$$

where M_{veh} is the effective vehicle mass, v_{veh} is the longitudinal vehicle velocity, ρ_{air} is the air density, A_f is the vehicle frontal area, C_d is the aerodynamic coefficient, g is the gravity acceleration, δ is the road slope angle, c_{roll} is the rolling resistance coefficient, T_{pwt} is the torque generated at the wheel shaft by the powertrain, T_{brake} is the braking torque, and R_{wh} is the wheel radius. Wheel speed is calculated using:

$$\omega_{wh} = \frac{v_{veh}}{R_{wh}}. \quad (2)$$

A simple transmission model is used which assumes a constant efficiency term:

$$\omega_t = g_t \omega_{wh} \quad (3)$$

$$T_t = \begin{cases} \frac{\eta_t}{g_t} T_{wh} & \text{if } P_{wh} = T_{wh} \cdot \omega_{wh} \geq 0, \\ \frac{1}{\eta_t g_t} T_{wh} & \text{if } P_{wh} = T_{wh} \cdot \omega_{wh} < 0 \end{cases} \quad (4)$$

where ω_t is the transmission speed, g_t is the transmission gear ratio, T_t is the transmission torque, η_t is the transmission efficiency, and T_{wh} is the wheel torque, which is equal to the force at the wheels multiplied by the wheel radius.

A simple torque converter model is based on lookup tables created with experimental data for the pump and turbine torque ratio, capacity factor (K_{tc}), and speed ratio. The internal combustion engine model uses a static map-based approach which assumes that

the engine is a perfect actuator with no response delays. Fuel consumption is calculated using an engine fuel map which is a function of engine speed and torque:

$$\dot{m}_f = f_{FE}(\omega_{ice}, T_{ice}) \quad (5)$$

where $\dot{m}_f$ is the fuel flow rate, ω_{ice} is the engine speed, and T_{ice} is the engine torque.

The rear electric motor (REM) uses a static map model based on torque and efficiency maps. The REM is treated as a perfect actuator where transient effects are assumed to be captured by the map. The governing equations, in terms of power, for the REM model are as follows:

$$P_{elec} = \begin{cases} \frac{1}{\eta_{rem}(\omega_{rem}, P_{elec})} \omega_{rem} T_{rem} & \text{if } P_{elec} \geq 0 \text{ (motoring mode),} \\ \eta_{rem}(\omega_{rem}, P_{elec}) \cdot \omega_{rem} T_{rem} & \text{if } P_{elec} < 0 \text{ (generating mode)} \end{cases} \quad (6)$$

where η_{rem} , ω_{rem} , and T_{rem} are the REM efficiency, REM speed, and REM torque, respectively.

The battery pack is modeled by a 0th order equivalent circuit model (ECM). SOC dynamics are given by:

$$\dot{SOC}(t) = \begin{cases} \frac{-1}{\eta_{coul}} \frac{I(t)}{Q_{nom}} & \text{if } I(t) > 0 \\ -\eta_{coul} \frac{I(t)}{Q_{nom}} & \text{if } I(t) < 0 \end{cases} \quad (7)$$

where Q_{nom} is the nominal charge capacity, I is the battery current and η_{coul} is the Coulombic efficiency. The circuit equation, as a function of terminal power, is:

$$P_{batt} = V_L \cdot I = V_{OC}I - R_0I^2 \quad (8)$$

where P_{batt} is the battery power, V_L is the load voltage at the battery terminals, V_{OC} is the open circuit voltage, and R_0 is the internal resistance. Accessory and auxiliary loads are assumed to be constant and modeled as a constant current draw from the battery pack. An explicit expression of current as a function of power is obtained:

$$I = \frac{V_{OC}}{2R_0} - \sqrt{\left(\frac{V_{OC}}{2R_0}\right)^2 - \frac{P_{batt}}{R_0}}. \quad (9)$$

To validate model accuracy, the test vehicle was run over a team-developed drive cycle at the Transportation Research Center (TRC) in East Liberty, Ohio. Data were collected at a rate of 50 Hz using a dSPACE MicroAutoBox II with dSPACE experiment software, ControlDesk. Vehicle speed was obtained from the wheel speed sensor signal and fuel consumption from the fuel flow rate read from the engine control unit (ECU). Figure 3 shows the drive cycle and a comparison of the total fuel consumption. The final fuel consumption value obtained from the simulation is 1.38% higher than the experimental data, indicating that the model closely matches the test vehicle.

Figure 3. Comparing experimental and simulated data.

2.3. Nonlinear Model Predictive Control

The objective is to minimize fuel consumption as well as travel time. A rollout algorithm that is based on the concept of MPC is chosen due to the feedback inherently present in the controller architecture. Utilizing MPC allows the optimal policy to be updated at each time step. The feedback control structure and frequent updates add robustness to uncertainties in the predictions. DP is chosen as the optimization algorithm due to its ability to handle nonlinear constraints and its guaranteed optimal solution, given information about the road segment in the look-ahead horizon. The length of the look-ahead horizon is a design choice and a trade-off between computation time and optimality. A smaller look-ahead horizon has fewer steps, and hence is computationally more efficient, whereas a longer horizon uses more future information but requires more time to compute.

Assume that at time instance t the vehicle is at distance n from the origin. At this instance, using the future speed limits, DP provides the optimal feedback law U^* for the segment of horizon length N_h . The optimal feedback controller is re-computed using DP at the update rate of 10 s.

Most of the look-ahead data (speed limits, SPaT, stop signs, etc.) are static, i.e., the location of these data sources is fixed with respect to the distance along the route, and they do not vary with time. For this reason, the optimal control problem is expressed in terms of distance as opposed to time. The objective function for MPC is a convex combination of fuel consumption and travel time.

At any distance step n along the route, the objective function can be written as:

$$J = \min_{u_i \in U_i} G(x_{n+N_h}) + \sum_{i=n}^{n+N_h} \left(\beta \cdot \frac{\dot{m}_{f,i}(x_i, u_i)}{\dot{m}_{f,norm}} + (1 - \beta) \right) \cdot \frac{\Delta s_i}{V_{veh,i}} \quad (10)$$

where n is the first step (distance) for the current horizon at time j , N_h is the horizon length, $\beta \in [0, 1]$ represents the trade-off between fuel consumption and travel time, Δs_i is the distance traveled over one step, $\dot{m}_{f,i}$ is the fuel flow rate, $\dot{m}_{f,norm}$ is a normalizing

factor, $x_i = [V_{veh,i} \quad SOC_i]^T$ is the state vector, $u_i = [T_{whl,i} \quad \tau_i]^T$ is the control input vector, $V_{veh,i}$ is the vehicle speed, G is the terminal cost at distance $n + N_h$, and τ_i is the torque split ratio between REM torque and IC engine torque. Since the aim of this research is energy management, metrics such as driver comfort and tail pipe emissions are not explicitly considered in the objective function. The objective function is subject to the following constraints:

$$\begin{aligned} SOC_{min} &\leq SOC_i \leq SOC_{max} \\ (V_{veh,min,i})|_n &\leq V_{veh,i} \leq (V_{veh,max,i})|_n \\ T_{ice,min} &\leq T_{ice,i} \leq T_{ice,max} \\ T_{rem,min} &\leq T_{rem,i} \leq T_{rem,max} \\ x_{i+1} &= f(x_i, u_i, \Delta s_i) \end{aligned} \quad (11)$$

where SOC_{min} , SOC_{max} and SOC_i are the minimum, maximum, and current state of charge, respectively, $(V_{veh,min,i})|_n$ and $(V_{veh,max,i})|_n$ are the minimum and maximum allowable speeds determined from look-ahead data obtained when the vehicle is at distance n , $V_{veh,i}$ is current vehicle speed, $T_{ice,min}$, $T_{ice,max}$, $T_{ice,i}$, $T_{rem,min}$, $T_{rem,max}$, and $T_{rem,i}$ are the minimum, maximum, and current engine torque and REM torque, respectively, and f is the discrete powertrain model. The powertrain model is equivalently expressed such that the independent variable of the difference equation is distance and not time.

For the rollout algorithm, the terminal cost $G(x_{n+N_h})$ in Equation (10) can be determined using a base policy. The method chosen for this work is based on the principles of approximate DP where a base policy is used to approximate the terminal cost in Equation (10). The base policy is obtained by solving the optimal control problem for the entire route with the speed limits available at the beginning of the trip. The full route optimal control problem is the same as described by Equations (10) and (11) but with $n = 0$ and $N_h = N$, where N is the total number of steps for the full route. The terminal cost $G(x_N)$ is such that the SOC at the end of the trip is equal to the SOC at the start of the trip, i.e.,

$$G(x_N) = \begin{cases} 0 & SOC_N = SOC_0 \\ inf & otherwise \end{cases} \quad (12)$$

where inf could be a very large number as compared to the expected value function [20]. The full route optimal control problem is solved using DP to obtain the optimal cost-to-go or value function $G^*(x, i)$ for all $i = 0, 1, \dots, N$. The terminal cost for a horizon n to $n + N_h$ is then $G(x_{n+N_h}) = G^*(\cdot, n + N_h)$. In the rollout algorithm, at each step n , the optimal control problem given by Equations (10) and (11) is solved using DP. The key difference between the base policy and the rollout is that the latter receives updated speed limits at every step n . The rollout algorithm further provides many benefits including a closed-loop optimal policy that matches the full-route optimization in the case where no disturbances are experienced, and it eliminates the need to define terminal constraints on the SOC while ensuring approximately charge-sustaining behavior over the entirety of the trip [21]. If SOC constraints are instead defined for every horizon length, the result is a conservative strategy which attempts to drive the SOC back to the initial value for each segment. This limits the SOC range and could potentially decrease optimality since the vehicle will not be able to take advantage of the entire usable battery range.

2.4. A-ECMS

The main purpose of the supervisory controller is to perform instantaneous optimization informed by the predictions obtained from the rollout algorithm. Inside of the

supervisory controller is A-ECMS, where a novel adaptation method is accomplished by an ANN referred to as the EF predictor. A-ECMS is expressed as:

$$\min_{\tau_i^*} \dot{m}_{f,i}(x_i, \tau_i^*) + \frac{s}{Q_{lhv}} P_{batt}(x_i, \tau_i^*) \cdot p(SOC) \quad (13)$$

where τ_i^* is the optimal torque split at each instant, s is the EF which serves as a conversion factor from electrical power to equivalent fuel consumption, Q_{lhv} is the fuel lower heating value, P_{batt} is battery power, and $p(SOC)$ is a multiplicative penalty function that ensures SOC boundaries are not violated [10]. A-ECMS was chosen for its simplicity, ease of online implementation, and ability to adapt to different driving scenarios. Extracting features from the speed and SOC trajectories allows the EF predictor to estimate the s , which results in a near optimal torque split over N_h . Since the rollout algorithm recalculates the state predictions many times throughout the route, s can be consistently updated to ensure that it is as close as possible to the optimal value without needing a priori route information.

2.5. EF Predictor

The EF predictor uses a regression ANN to simultaneously identify the drive cycle type and estimate s based on future state trajectory predictions obtained using look-ahead data. The expected output of this ANN is the optimal EF for the upcoming driving segment, and Table 1 tabulates the features extracted from the data, which includes a selection used by the authors in [22] and additional features related to the SOC variation. These features are used as the input vector for the ANN.

Table 1. A List of 16 selected features for EF estimation.

List of Features	
Trip distance	Maximum, minimum and mean SOC
Maximum, minimum and mean vehicle speed	Standard deviation SOC
Maximum, minimum and mean acceleration	Maximum, minimum and mean SOC change
Speed standard deviation	Fuel consumed

The rationale behind using an ANN to estimate s originates from the concept that a well-tuned EF results in near-optimal behavior [23–25]. In its traditional formulation, ECMS relies on full route information which is impossible to know in the real world. However, using the look-ahead data available at the beginning of the trip, the full route speed limits can be predicted. Since the proposed approach uses DP to obtain the optimal speed, this optimal speed could be used to compute the optimal EF. The idea is that if s can be determined for every N_h such that the torque split obtained from ECMS matches that determined by DP, the fuel consumption should closely match the optimal result, since each recalculation is influenced by the full route solution. By doing so, A-ECMS can now be adapted based on look-ahead information as opposed to reacting to an SOC difference or relying solely on past information.

The ANN training data is generated using the optimal speed and SOC trajectories determined using DP over many trips. The speed limits and traffic light positions for the trips were generated using typical drive cycles that captured common driving types, such as highway, urban, and mixed driving. They are based off of five standard drive cycles with some modifications, namely the US06, HWFET, UDDS, NYCC, and LA92 cycles [26]. This ensures that the ANN training data is representative of realistic driving. Each drive cycle was first converted from a speed versus time plot to speed versus distance. The speed limit was approximated by matching the drive cycle with distance coordinates. Traffic lights were placed at locations where the drive cycle slows down to zero or closely approaches zero. Figure 4 shows an example of the speed limits constructed from drive cycles.

Figure 4. Speed limit based on US06.

The speed profile not only depends on the speed limits and traffic lights but also on surrounding traffic. For each of the drive cycles, five iterations are created where the speed limit is randomized within a certain range to model changes in average traffic flow speed. Traffic light positions remain fixed, but the SPaT is random for each iteration to simulate real-world traffic. The result is a large number of trips defined by speed limits as a function of distance and SPaT. These trips are further divided into road segments of length N_h , which are representative of what the vehicle will encounter on the road. For each segment of each trip, DP provides optimal speed and SOC trajectories. The result is a large data set of optimal speed and SOC trajectories over a horizon of length N_h . The actual training data set which is fed into the ANN consists of the calculated features shown in Table 1 for each speed and SOC trajectory found by DP.

To obtain the optimal s for each training data entry, a traditional ECMS with a constant s is used. The optimal s is found using the shooting method, where s is varied until the SOC trajectory terminates at the desired final SOC, SOC_f , starting from an initial SOC, SOC_i . Therefore, SOC_i and SOC_f are chosen to be the initial SOC and final SOC for each data entry in the training data. Since the base heuristic is available, matching both the speed and SOC trajectory determined by DP for each horizon will force ECMS to provide charge sustainability without explicitly accounting for the error between the desired final SOC and current SOC. This allows DP-ECMS to take advantage of the entire battery SOC range. Alternative methods which force ECMS to approach the target final SOC after every segment limit the range of SOC and result in more fuel use.

After finding the optimal s for each entry in the data set, the data was split as such: 70% used for training, 15% for validation, and 15% for testing. The ANN was trained using the MATLAB Deep Learning Toolbox [27]. The network contains 2 hidden layers with 25 neurons each and uses the Levenberg–Marquardt method as the training function and a hyperbolic tangent sigmoid transfer function.

2.6. Real-Time Implementation and Synchronization

It is important to note that the objective function for MPC is expressed in distance, while in the supervisory controller, it is in time. Careful choice of horizon length, prediction update rate (t_c), and EF update rate (t_s) is crucial for real-time implementation. A long horizon window is not practical due to the substantial computational load. Additionally, longer-term predictions are more uncertain due to uncertain future traffic. It was found that the nonlinear MPC with DP as solver can complete one optimization over 600 m in well under 10 s. Hence, the prediction horizon window is set to 600 m, and DP is recalculated every 10 s. For t_s , an update rate that is too rapid will react to every minor change and may cause oscillations in the torque demand. On the other hand, updating too slow can cause

the adaptation to become unstable and cause SOC to deviate far from the reference value. Based on this reasoning and studies found in the literature [28], $t_s = 30$ s is chosen. Figure 5 shows a timing diagram of the DP-ECMS energy management system. The horizon length is converted to time, t_h . This time is not fixed and depends on the horizon length (600 m) and the speed of the vehicle. The horizon length was chosen to be long enough so that t_c will always be much shorter than t_h regardless of the ego vehicle speed. This ensures that the vehicle will never travel further than the horizon length before updating the prediction.

Figure 5. DP-ECMS timing diagram.

2.7. Baseline FE

To quantify the improvement due to the DP-ECMS solution over a baseline, a human driver model is used based on the well-known intelligent driver model (IDM) [29,30] modified to make the ego vehicle stop at stop signs and traffic lights. This is called the enhanced driver model (EDM) [31] and is expressed by the following equations:

$$\frac{dv}{dt} = \begin{cases} a \left[1 - \left(\frac{v}{v_0 - \theta_{offset}} \right)^\delta \right] & \text{if } v < v_0 \text{ [Freeway Driving]} \\ -b \left[1 - \left(\frac{v_0 - \theta_{offset}}{v} \right)^\delta \right] & \text{if } v > v_0 \text{ [Freeway Driving]} \\ a \left[1 - \left(\frac{v}{v_{leader}} \right)^\delta \right] & \text{if } v < v_{leader} \text{ [Car Following]} \\ -b \left[1 - \left(\frac{v_{leader}}{v} \right)^\delta \right] & \text{if } v > v_{leader} \text{ [Car Following]} \\ -\frac{1}{b} \left(\frac{v^2}{2s} \right) & \text{[Stop]} \end{cases} \quad (14)$$

$$s = x_{leader} - x_{ego} - x_{safe} \quad (15)$$

$$s_{brake} = \left(1 + \frac{c_1}{\delta} \right) \frac{v^2}{2b} \quad (16)$$

where v , v_0 , and v_{leader} are the ego vehicle speed, speed limit, and lead vehicle speed, respectively, a and b are the maximum acceleration and deceleration, respectively, δ defines driver aggressiveness, θ_{offset} further models driver behavior where a relaxed driver would choose to drive slightly below the speed limit and an aggressive driver will get as close as possible to the speed limit, the term $\left(\frac{c_1}{\delta} \right)$ is a calibration term which captures the dependence of braking distance on aggressiveness, x_{leader} , x_{ego} , and x_{safe} are the lead, ego, and safe distance, respectively. The safe distance ensures a safe gap to the lead vehicle.

The EDM is used as a benchmark to compare the energy consumption over a trip driven by a human driver to a vehicle driven by the EMS developed in this research. The

driver is calibrated to match the aggressiveness of the optimal speed prediction to ensure that the comparison is fair. The term representing the trade-off between fuel consumption and travel time (β in Equation (10)) can be thought of as the driver aggressiveness since the vehicle will tend to either drive faster or slower, depending on the value chosen for β . For the results presented in this research, a value of $\beta = 0.65$ is chosen which represents a moderate driver.

The EDM calibration parameters are $[a, b, \delta, c_1, \theta_{offset}]$ and are determined using a genetic algorithm to solve a least squares optimization problem which minimizes the root mean square error:

$$RMS = \sqrt{\frac{1}{N} \sum_{j=1}^N (V_{EDM_j} - V_{data_j})^2} \quad (17)$$

where N is the number of samples, j is the current sample, V_{EDM_j} is the velocity calculated by the EDM and V_{data_j} is the experimental value. The experimental data used to calibrate the driver model is obtained by running DP, assuming that the vehicle stops at every traffic light along the route. This allows the calibration process to capture acceleration and deceleration to match the behavior of the DP solution. Table 2 shows the optimized parameter values.

Table 2. Calibrated EDM parameters.

Parameters	Optimized Values
a [m/s ²]	2.96
b [m/s ²]	1.54
c_1 [-]	3.75
δ [-]	1.44
θ_{offset} [m/s ²]	0.82

The EMS used for the baseline is A-ECMS, where the adaptation is done via PI control based on SOC feedback. The controller adjusts s to minimize the difference between the current SOC and the target SOC with the goal of maintaining charge sustainability. The formulation is as follows:

$$s(t) = s_0 + k_p(x_{ref} - x(t)) + k_I \int_0^t (x_{ref} - x(\tau)) d\tau, \quad (18)$$

where s_0 is the EF at $t = 0$, k_p and k_I are the proportional and integral gains, respectively, $x(t)$ is the SOC at time t , and x_{ref} is the reference SOC.

3. Results and Discussion

A test route chosen for a simulation study consists of both highway and city driving on local roads in Columbus, Ohio. Figure 6 shows the route on a map along with traffic light locations and speed limits. Table 3 shows relevant route characteristics.

Table 3. Route characteristics.

Trip Length	Max. Speed	Avg. Speed	Traffic Lights
8.4 km	30 m/s	23.3 m/s	9

Figure 6. Map of route.

Beginning with the fixed speed limit along the route, traffic patterns typical of daily driving are randomly generated by choosing a random location along the route, a random speed which is less than or equal to the original speed limit, and a random length of speed limit variation. Along the selected route, there are nine traffic lights. A similar approach is used to randomly set the traffic light phase and timing. Examples of randomly generated speed limits over the chosen route are shown in Figure 7, where colored (green or red) vertical lines indicate the traffic light location and the phase the light is in. The size of each vertical colored segment indicates how long the phase lasts.

Figure 7. Selected route with random traffic and SPaT.

The ability of the MPC to avoid red lights and the corresponding FE benefit from the EMS is shown first, followed by the FE distribution over several randomly generated traffic and SPaT patterns.

Figure 8 shows the speed profiles for both DP-ECMS and the baseline for one trip. When the vehicle is approaching an intersection during its red phase, the baseline driver initiates the stop sequence in anticipation of the red light. Since it has no knowledge of the signal timing, there is no way to determine if the light will turn green before the vehicle reaches the intersection. DP-ECMS is able to use available look-ahead data to calculate the appropriate speed to arrive at the light on green and adjusts accordingly. For all simulations, the DSRC range is assumed to be 500 m based on studies found in the literature [11,32,33]. Figure 9 shows the travel time versus the distance traveled. DP-ECMS slows the vehicle down just enough to pass through the green window when possible, while the baseline will always initiate a stop if the light is red. The travel time is approximately equal for both methods.

Figure 8. Speed profiles of DP-ECMS vs. baseline.

Figure 9. Time vs. distance for DP-ECMS and baseline.

Figure 10 compares the fuel consumption for the two systems. DP-ECMS achieves a 7.1% reduction in fuel consumed compared to the baseline. The use of DP over the full route, assuming prior knowledge of the vehicle instantaneous speed for the entire trip, provides an ideal fuel consumption. When compared to this DP solution (0.53 kg total fuel consumed), the baseline consumes 13% more fuel than DP (0.59 kg total fuel) and the proposed method uses 5% more fuel (0.56 kg total fuel) than DP. It is also shown in Figure 11

that both strategies are approximately charge-sustaining, and the corresponding adapted EFs are shown in Figure 12. Although tailpipe emissions are not directly considered in the objective function, reducing the number of traffic light stops provides the added benefit of emissions reduction [34]. Therefore, it is expected that the proposed EMS will reduce emissions along with fuel consumption. A direct way of addressing tailpipe emissions is to incorporate the emissions in the cost function and after-treatment system dynamics in the dynamic constraints [35].

Figure 10. Fuel consumption comparison for DP-ECMS and baseline.

Figure 11. SOC comparison for DP-ECMS and baseline.

A Monte Carlo analysis is conducted with a sample size of 85, where the random variables are SPaT and average traffic speed. Both the baseline and DP-ECMS are run using the same traffic and SPaT patterns to ensure that the results are directly comparable. Considering that the vehicle in question has an electrified powertrain, it is important to keep track of the battery SOC. A common way to account for differences in the final and initial SOC is to apply a correction factor to fuel consumption based on the magnitude of the deviation [36]. However, this process can be imprecise and is difficult to properly apply to a large number of samples. To simplify the process, each trip was repeated 10 times and the FE (MPG) was computed for the duration of the 10 cycles. By doing so, the small deviations in the SOC are negligible over the energy consumed by burning fuel over 10 trips. Figure 13 shows the Monte Carlo simulation results with overlaid histograms for both DP-ECMS and the baseline along with the mean and standard deviation.

Figure 12. EF comparison for DP-ECMS and baseline.

Figure 13. Fuel economy distribution of DP-ECMS and baseline.

The DP-ECMS method yields MPG values which range from about 18 to 28 MPG, while the baseline ranges from 16.1 to 25 MPG. The median bin for the baseline is 18.2 to 19.2 MPG, whereas for DP-ECMS, it is 20.2 to 21.3 MPG. DP-ECMS yields an average FE of 22.06 MPG, while the baseline is 19.96 MPG. The results show that, on average, the DP-ECMS provided a 10% FE increase over baseline. A histogram of the percent gain in FE is shown in Figure 14, which shows gains in FE for each trip.

By plotting the percent gain, it can be seen that the algorithm always performs either the same as or better than the baseline. It is expected that there will be cases where DP-ECMS will have similar FE to the baseline. Since the biggest advantage of DP-ECMS is the use of look-ahead data, if there are cases where there is no traffic and all green lights, there will most likely not be much benefit to having look-ahead data. Likewise, if the traffic is so heavy that the vehicle is not able to leverage the future knowledge, it will behave similar to the baseline.

Figure 14. Fuel economy percent gain of DP-ECMS over baseline.

4. Conclusions

This paper presented a novel approach for integrated EMS and speed control using A-ECMS considering look-ahead data. A data-driven approach using ANN was developed to predict the optimal EF for ECMS. A unique combination of a DP-based optimal control solution for speed profiling and ECMS with an optimal EF predictor was shown to be capable of online implementation on vehicle hardware. It was shown that a simple rule-based approach to handling the traffic light signal phase and timing avoids unnecessary stops at red lights and consequently reduces fuel consumption. A receding horizon optimal control framework was used to adapt to disturbances along the route and predict the future speed and SOC trajectories. A data-driven EF adaptation approach was shown to be beneficial, since the neural network is able to learn the nonlinear relationship between the input features extracted from optimal future trajectories and the corresponding optimal EF. This further allows ECMS to compute the optimal torque split ratio for each future horizon as opposed to simply reacting to the information available at each instant. A statistical analysis showed that DP-ECMS always performs equal to or better than the baseline, and an average 10% FE improvement was achieved while maintaining an approximately equal travel time. The steps towards implementing the proposed methods on vehicles should include various levels of x-in-the-loop testing, I/O considerations like CAN delay, errors and synchronization, and optimization of the code for the target hardware.

Author Contributions: Conceptualization, W.P., P.T., S.M.-M. and G.R.; methodology, W.P., P.T., S.M.-M. and G.R.; software, W.P.; validation, W.P., P.T., S.M.-M. and G.R.; formal analysis, W.P.; investigation, W.P.; resources, P.T., S.M.-M. and G.R.; data curation, W.P.; writing—original draft preparation, W.P.; writing—review and editing, P.T., S.M.-M. and G.R.; visualization, W.P.; supervision, P.T., S.M.-M. and G.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

Variable	Description	Value
Vehicle dynamics parameters		
M_{veh}	Effective vehicle mass	2450 [kg]
v_{veh}	Vehicle speed	[m/s]
T_{pwt}	Powertrain torque	[Nm]
T_{brake}	Brake torque	[Nm]
R_{wh}	Wheel radius	0.36 [m]
C_{roll}	Rolling resistance	
g	Gravitational acceleration	9.8 [m/s ²]
δ	Road grade angle	[deg]
ρ_{air}	Air density	1.25 [kg/m ³]
A_f	Vehicle frontal area	
C_d	Aerodynamic drag coefficient	
ω_{wh}	Wheel speed	[rad/s]
Transmission parameters		
ω_t	Transmission speed	[rad/s]
g_t	Transmission gear ratio	
T_t	Transmission torque	[Nm]
η_t	Transmission efficiency	
Engine parameters		
$\dot{m}_f$	Fuel flow rate	[kg/s]
f_{FE}	Engine FE map	
ω_{ice}	Engine speed	[rad/s]
T_{ice}	Engine torque	[Nm]
Electric motor parameters		
P_{elec}	Electric motor power	[kW]
η_{rem}	REM efficiency	
ω_{rem}	REM speed	[rad/s]
T_{rem}	REM torque	[Nm]
Battery parameters		
Q_{nom}	Nominal charge capacity	11.8 [Ah]
η_{coul}	Coulombic efficiency	
P_{batt}	Battery power	[kW]
V_L	Load voltage	[V]
V_{OC}	Open circuit voltage	[V]
R_0	Battery internal resistance	[Ω]
SOC	State of Charge	[%]
Optimal Control		
x	States	
u	Inputs	
$(\cdot)_i$	Quantity at time-instance i	
$(\cdot)_n$	Quantity at distance-step n	
$(\cdot)^*$	Optimal solution	
N_h	Look-ahead horizon (number of distance steps)	
$G(x)$	Terminal cost	
Δs_i	Distance discretization step size	[m]
β	Trade-off parameter between FE and travel time	
A-ECMS parameters		
s	Equivalence factor	
Q_{lhv}	Fuel lower heating value	43.2 [MJ/kg]
EDM parameters		
$a, b, c_1, \delta, \theta_{offset}$	EDM parameters	Table 2

References

1. Asher, Z.; Patil, A.; Wifvat, V.; Frank, A.; Samuelsen, S.; Bradley, T. Identification and Review of the Research Gaps Preventing a Realization of Optimal Energy Management Strategies in Vehicles. *Sae Int. J. Altern. Powertrains* **2019**, *8*, 133–150. [\[CrossRef\]](#)
2. Panday, A.; Bansal, H. A Review of Optimal Energy Management Strategies for Hybrid Electric Vehicle. *Int. J. Veh. Technol.* **2014**, *2014*, 160510. [\[CrossRef\]](#)
3. Heppeler, G.; Sonntag, M.; Sawodny, O. Fuel Efficiency Analysis for Simultaneous Optimization of the Velocity Trajectory and the Energy Management in Hybrid Electric Vehicles. *IFAC Proc. Vol.* **2014**, *47*, 6612–6617. [\[CrossRef\]](#)
4. Kim, T.; Manzie, C.; Sharma, R. Model Predictive Control of Velocity and Torque Split in a Parallel Hybrid Vehicle. In Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA, 11–14 October 2009; pp. 2014–2019.
5. Ngo, D.V.; Hofman, T.; Steinbuch, M.; Serrarens, A.F.A. An optimal control-based algorithm for Hybrid Electric Vehicle using preview route information. In Proceedings of the 2010 American Control Conference, Baltimore, MD, USA, 30 June–2 July 2010; pp. 5818–5823.
6. Gong, Q.; Li, Y.; Peng, Z.R. Trip-Based Optimal Power Management of Plug-in Hybrid Electric Vehicles. *IEEE Trans. Veh. Technol.* **2008**, *57*, 3393–3401. [\[CrossRef\]](#)
7. Gong, Q.; Li, Y.; Peng, Z. Power management of plug-in hybrid electric vehicles using neural network based trip modeling. In Proceedings of the American Control Conference, St. Louis, MO, USA, 10–12 June 2009; pp. 4601–4606.
8. Mohd Zulkefli, M.A.; Zheng, J.; Sun, Z.; Liu, H.X. Hybrid powertrain optimization with trajectory prediction based on inter-vehicle-communication and vehicle-infrastructure-integration. *Transp. Res. Part Emerg. Technol.* **2014**, *45*, 41–63. [\[CrossRef\]](#)
9. Mandava, S.; Boriboonsomsin, K.; Barth, M. Arterial velocity planning based on traffic signal information under light traffic conditions. In Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems, St. Louis, MO, USA, 4–7 October 2009.
10. Onori, S.; Serrao, L.; Rizzoni, G. *Hybrid Electric Vehicles Energy Management Strategies*; Springer: London, UK, 2016.
11. Deshpande, S.; Gupta, S.; Kibalama, D.; Pivarro, N.; Canova, M.; Rizzoni, G.; Aggoune, K.; Olin, P.; Kirwan, J. *In-Vehicle Test Results for Advanced Propulsion and Vehicle System Controls Using Connected and Automated Vehicle Information*; SAE Technical Paper; SAE: Warrendale, PA, USA, 2021.
12. Hellström, E.; Ivarsson, M.; Åslund, J.; Nielsen, L. Look-ahead control for heavy trucks to minimize trip time and fuel consumption. *Control. Eng. Pract.* **2009**, *17*, 245–254. [\[CrossRef\]](#)
13. Ozatay, E.; Onori, S.; Wollaeger, J.; Ozguner, U.; Rizzoni, G.; Filev, D.; Michelini, J.; Di Cairano, S. Cloud-Based Velocity Profile Optimization for Everyday Driving: A Dynamic-Programming-Based Solution. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2491–2505. [\[CrossRef\]](#)
14. Ruan, S.; Ma, Y.; Yang, N.; Xiang, C.; Li, X. Real-time energy-saving control for HEVs in car-following scenario with a double explicit MPC approach. *Energy* **2022**, *247*, 123265. [\[CrossRef\]](#)
15. Chen, Z.; Liu, Y.; Zhang, Y.; Lei, Z.; Chen, Z.; Li, G. A neural network-based ECMS for optimized energy management of plug-in hybrid electric vehicles. *Energy* **2022**, *243*, 122727. [\[CrossRef\]](#)
16. Rathore, A. Intelligent Energy Management Strategy for Eco-Driving in Connected and Autonomous Hybrid Electric Vehicles. Master's Thesis, McMaster University, Hamilton, ON, Canada, 2021.
17. Guo, Q.; Angah, O.; Liu, Z.; Ban, X.J. Hybrid deep reinforcement learning based eco-driving for low-level connected and automated vehicles along signalized corridors. *Transp. Res. Part Emerg. Technol.* **2021**, *124*, 102980. [\[CrossRef\]](#)
18. Xu, F.; Shen, T. Look-ahead Horizon based Energy Optimization for Connected Hybrid Electric Vehicles. In Proceedings of the 2020 IEEE 3rd Connected and Automated Vehicles Symposium (CAVS), Victoria, BC, Canada, 18 November–16 December 2020; pp. 1–5. [\[CrossRef\]](#)
19. Advanced Vehicle Technology Competitions: EcoCAR Mobility Challenge. Available online: <https://avtcservices.org/ecocar-mobility-challenge/> (accessed on 19 January 2022).
20. Elbert, P.; Ebbesen, S.; Guzzella, L. Implementation of Dynamic Programming for n-Dimensional Optimal Control Problems With Final State Constraints. *IEEE Trans. Control. Syst. Technol.* **2013**, *21*, 924–931. [\[CrossRef\]](#)
21. Deshpande, S.R.; Jung, D.; Canova, M. Integrated Approximate Dynamic Programming and Equivalent Consumption Minimization Strategy for Eco-Driving in a Connected and Automated Vehicle. *arXiv* **2020**, arXiv:2010.03620.
22. Park, J.; Chen, Z.; Kiliaris, L.; Kuang, M.L.; Masrur, M.A.; Phillips, A.M.; Murphey, Y.L. Intelligent Vehicle Power Control Based on Machine Learning of Optimal Control Parameters and Prediction of Road Type and Traffic Congestion. *IEEE Trans. Veh. Technol.* **2009**, *58*, 4741–4756. [\[CrossRef\]](#)
23. Sciarretta, A.; Back, M.; Guzzella, L. Optimal control of parallel hybrid electric vehicles. *IEEE Trans. Control. Syst. Technol.* **2004**, *12*, 352–363. [\[CrossRef\]](#)
24. Musardo, C.; Rizzoni, G.; Guezennec, Y.; Staccia, B. A-ECMS: An adaptive algorithm for hybrid electric vehicle energy management. *Eur. J. Control* **2005**, *11*, 509–544. [\[CrossRef\]](#)
25. Musardo, C.; Staccia, B. Energy Management Strategies for Hybrid Electric Vehicles. Master's Thesis, Politecnico di Milano, Milan, Italy, 2003.
26. Dynamometer Drive Schedules. Available online: <https://www.epa.gov/vehicle-and-fuel-emissions-testing/dynamometer-drive-schedules#main-content> (accessed on 19 January 2022).

27. MathWorks Deep Learning Toolbox. Available online: https://www.mathworks.com/help/deeplearning/index.html?s_tid=CRUX_lftnav (accessed on 19 January 2022).
28. Anil, V.S. Mission-based Design Space Exploration and Traffic-in-the-Loop Simulation for a Range-Extended Plug-in Hybrid Delivery Vehicle. Master's Thesis, The Ohio State University, Columbus, OH, USA, 2020.
29. Kesting, A.; Treiber, M.; Helbing, D. Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity. *Philos. Trans. R. Soc. Math. Phys. Eng. Sci.* **2010**, *368*, 4585–4605. [[CrossRef](#)]
30. Treiber, M.; Hennecke, A.; Helbing, D. Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E* **2000**, *62*, 1805–1824. [[CrossRef](#)]
31. Gupta, S.; Rajakumar Deshpande, S.; Tufano, D.; Canova, M.; Rizzoni, G.; Aggoune, K.; Olin, P.; Kirwan, J. *Estimation of Fuel Economy on Real-World Routes for Next-Generation Connected and Automated Hybrid Powertrains*; SAE Technical Paper; SAE: Warrendale, PA, USA, 2020.
32. Xu, Z.; Li, X.; Zhao, X.; Zhang, M.; Wang, Z. DSRC versus 4G-LTE for Connected Vehicle Applications: A Study on Field Experiments of Vehicular Communication Performance. *J. Adv. Transp.* **2017**, *435*, 1–10. [[CrossRef](#)]
33. Filippi, A.; Moerman, K.; Martinez, V.; Haran, O. IEEE802.11p ahead of LTE-V2V for safety applications. *Autotalks NXP* **2017**, *1*, 1–19.
34. Bloder, E.; Jäger, G. Is the Green Wave Really Green? The Risks of Rebound Effects When Implementing “Green” Policies. *Sustainability* **2021**, *13*, 5411. [[CrossRef](#)]
35. Jankord, G. Control of Criteria Emissions and Energy Management in Hybrid Electric Vehicles with Consideration of Three-Way Catalyst Dynamics. Ph.D. Thesis, The Ohio State University, Columbus, OH, USA, 2021.
36. Kong, Z.; Zou, Y.; Liu, T. Implementation of real-time energy management strategy based on reinforcement learning for hybrid electric vehicles and simulation validation. *PLoS ONE* **2017**, *12*, e0180491. [[CrossRef](#)] [[PubMed](#)]