A Comparison of Optimal Supervisory Control Strategies for Series Plug-in Hybrid Electric Vehicle Powertrains through Dynamic Programming

Rakesh Patil, Student Member, IEEE, Zoran Filipi, and Hosam Fathy

Abstract—This paper uses dynamic programming to compare the optimal fuel and electricity costs associated with two supervisory control strategies from the plug-in hybrid electric vehicle (PHEV) literature. One strategy blends fuel and electricity for propulsion throughout the useful range of battery state of charge (SOC), while the second strategy switches from all-electric to blended operation at a predefined SOC threshold. Both strategies are optimized for a series PHEV powertrain using deterministic dynamic programming (DDP) to ensure a fair comparison. The DDP algorithm is implemented using a backward-looking powertrain model instead of forward-looking models used in previous research. This approach helps evaluate the constraints at every time step 'on the go' in a computationally efficient manner with no requirement for storage. This makes our optimization feasible in terms of computational time and storage requirements. The paper's primary conclusion is that there is no significant difference in the performance of the two control strategies for the series PHEV except when gasoline is cheaper than electricity per mile. This result contrasts sharply with previous results for parallel and power-split PHEVs, and is examined for different relative fuel and electricity prices and trip lengths.

Index Terms—Dynamic Programming, plug-in hybrid electric vehicles (PHEV), Optimal Supervisory Control, Series Configuration.

I. INTRODUCTION

This paper examines the problem of optimizing the cost of the fuel and electricity consumed by a plug-in hybrid electric vehicle. The paper is motivated by the fact that PHEVs make it possible for two major infrastructures, the transportation infrastructure and the power grid, to exchange significant amounts of energy. In doing so, PHEVs make it possible to replace some of the petroleum currently being used for propulsion with other sources of energy [1]. To maximize this synergy, our overarching goal is to optimize the cost of fuel and electricity consumption for a PHEV.

In pursuit of the above goal, this paper specifically compares the optimal fuel and electricity consumption costs associated with two supervisory control strategies for PHEVs. The first strategy, blending, uses fuel and electricity together for propulsion throughout a given drive cycle [2]-[4]. The second strategy is dubbed EV/CS because it operates in "Electric Vehicle" mode first, then switches to "Charge Sustenance" at a predefined SOC threshold [4], [5]. We compare these strategies using deterministic dynamic programming to ensure fairness, and implement the DDP algorithm in a novel manner using a backward-looking powertrain model instead of forward-looking models used in previous research [2], [3], [6].

Previous research has shown that blending optimizes combined fuel and electricity costs for both parallel and power split PHEVs [2], [3], [6], [7]. Blending attains this optimality by employing the engine to slow down battery charge depletion, thereby minimizing the amount of time spent in charge sustenance. This is important because in the charge sustenance mode, the engine must (i) meet driver power demand, (ii) regulate battery SOC, while (iii) being mechanically coupled to the final drive. This combination of three requirements and constraints can tax engine efficiency significantly, making charge sustenance undesirable [3,6]. In a series powertrain, the engine is mechanically decoupled from the final drive. In this paper, we show that the difference in fuel and electricity cost between the blending and EV/CS strategies is not as pronounced for a series vehicle.

Research on optimal supervisory control strategies for hybrid vehicle powertrains has focused on optimization methods [2]-[8], the impact of different powertrain architectures [9], [10], and driving scenarios [6], [10]. The optimal control methods applied include fuzzy logic techniques [8], equivalent fuel consumption minimization strategy (ECMS) [2], [7], dynamic programming approaches [3], [6], [11], [12] and model predictive control [13]. A recent article by Pisu and Rizzoni compares some of these methods and the subsequent results obtained by using them to control a parallel hybrid powertrain [7]. Regardless of the optimization method used, the goal in hybrid power management is to

Manuscript received February 14, 2012. This work was supported in part by the U.S. NSF under Grant #0835995.

Rakesh Patil and Zoran Filipi are with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: rakeshmp@umich.edu; filipi@umich.edu).

Hosam Fathy is with the Department of Mechanical and Nuclear Engineering, The Pennsylvania State University, University Park, PA 16802 USA (e-mail: hkf2@psu.edu).
minimize vehicle-level objective(s), such as fuel consumption, by optimally allocating driver power demand among different propulsion devices. The dynamic programming approach explicitly uses Bellman’s principle for optimal control, and thus guarantees global optimality. The output of this process is a supervisory control trajectory that can be used to gain important physical insights and extract implementable rules for subsequent online power management [11], [14]. In [2], [4], authors compare the EV/CS and Blended control strategies for series, parallel and power-split architectures. They obtain implementable control rules and discuss the pros and cons of each architecture based on simulation results. They also assert the need for using globally optimal methods for a fair comparison. This motivates us to apply optimal control by using DDP for the series architecture.

Applications of dynamic programming in optimal powertrain control have followed the numerical methods outlined in [15]. These methods use a forward looking model of the system under consideration and interpolates to obtain the cost-to-go function. As exhaustive as this method is, it is, depending on the system under consideration, computational challenges arise. We did not experience the more popular problem of the curse of dimensionality [16], as our system was modeled with two states and two inputs. However, due to the large size of the battery in a series PHEV, its state-of-charge (SOC) dynamics are very slow in terms percentage change in SOC, requiring a very fine state space gridding. This resulted in a higher number of interpolations to obtain value function estimates and an exaggerated effect of the propagation of penalties used to handle constraints. These two effects required a higher amount of computational time and memory, rendering the problem computationally intractable.

There have been several attempts to overcome these particular computational challenges. In [17], authors present a novel procedure to consider constraints that bound the states of a system with only one state and one input. Several authors [12], [18], evaluate feasible sets of controls offline, before performing the optimization. The authors' approach in [12] is geared towards reducing computational time only. The issue of propagation of large penalty values associated with constraints is not considered. Further, the approaches outlined in [12], [18], are well suited for stochastic DP. To implement this methodology at every time step of deterministic DP would require significant offline calculations and storage. Our approach of using a backward looking powertrain model to implement DDP helps evaluate the constraints at every time step on the go in a computationally efficient manner with no requirement for storage. In [19], a preliminary version of this approach is presented. This paper explains the advantages of our approach in more detail and the approach is applied to a wider range of cases that allow us to examine the impact of driving distance on the optimal EV/CS and optimal blending strategies. We also show that the powertrain model under consideration is invertible, thereby allowing us to use a backward looking model.

The remainder of this paper is organized as follows. Section 2 describes the powertrain model and its equations. Section 3 presents the DP implementation. Section 4 defines the optimal control problem and outlines the cases studied. Finally, in section 5 we discuss the optimization results and compare the performance of the two optimal control strategies.

II. PROBLEM FORMULATION

The optimal control problem is formulated as follows: Minimize

\[ J = \sum_{k=0}^{T} (c_{r} m_{j}(k) + c_{e} P_{e}(k)) \]

subject to

\[ u(k) = g(x(k), x(k+1)), \quad x \in Z, \quad u \in U \]

(1. a-c)

where \( J \) represents the cost function to be minimized. It is given by the total dollar costs of the gasoline and electric power used during a chosen trip. The cost of gasoline is \$c_{r}/gallon and the cost of electricity is \$c_{e}/kWh. The engine fuel usage rate is given by \( (\dot{m}_{f}) \) and the battery power is given by \( P_{b} \). The total time for the driving trip under consideration is \( T \). The states and inputs of the system are \( x \) and \( u \) respectively.

The system equations \( \dot{x} = f(x, u) \) are implemented in a backward looking fashion \( u(k) = g(x(k), x(k+1)) \), rather than the more common forward looking model \( x(k+1) = f(x(k), u(k)) \). It should be interpreted as follows; given the current and future states \( (i.e. x(k) \text{ and } x(k+1)) \) respectively, the model determines the input \( u(k) \) required to make this transition. More details about this model are presented in section II.D and its use is explained in section III. The admissible ranges of \( x \) and \( u \) are \( Z \) and \( U \), which are governed by state and input constraints discussed in section II.C. In the remainder of this section, the series powertrain model is briefly explained and its equations and constraints are presented.

A. Model Overview

The series electric powertrain model considered for this study is schematically shown in Fig. 1 and the component specifications are listed in Table I. The arrows indicate the possible directions of power flow. The power electronics are a parallel bus which split the electric current between the generator, battery and the motor. Thus the engine power is converted to electrical power in the generator and is split between the driving motor and the battery in the power electronics depending on the wheel power demand.

![Fig. 1. Series hybrid powertrain model](image)

The Engine, Generator and the Motor are static map based models. The maps are obtained from Powertrain System Analysis Toolkit (PSAT) [20] through testing at Argonne National Laboratory (ANL) and they provide the operating
efficiencies of the components. The battery is an equivalent circuit model where its open circuit voltage and internal resistance are functions of the battery state of charge (SOC). The maps for voltage and internal resistance are based on Li-Ion battery chemistry and are also obtained from PSAT.

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engine</td>
<td>MY04 Prius 1.497 L gasoline, 57 kW, 110 Nm max torque at 4000 RPM.</td>
</tr>
<tr>
<td>Generator</td>
<td>Permanent magnet. 400 Nm max torque, 58kW peak power</td>
</tr>
<tr>
<td>Battery</td>
<td>Li-ion, 7.035Ah 75 cell SAFT model scaled by current capacity to 13.3 kWh total energy</td>
</tr>
<tr>
<td>Motor</td>
<td>Permanent magnet. 715 Nm max torque, 130 kW peak power</td>
</tr>
<tr>
<td>Final Drive</td>
<td>Ratio = 3.07</td>
</tr>
<tr>
<td>Resistance</td>
<td>$f_g = 88.6 \text{ N}, f_i = 0.14 \text{ N-s/m}, f_j = 0.36 \text{ N-s}^2/\text{m}^2$</td>
</tr>
</tbody>
</table>

### Table I
**Component Specifications of the Powertrain**

**B. Model Equations**

Equations governing the powertrain model dynamics are represented as nonlinear ODEs. This powertrain model has 3 state variables: vehicle velocity ($v$), engine speed ($\omega_e$) and battery state-of-charge (SOC). The three inputs used to control the model are the power demand by the driver at the wheels ($P_{\text{wh,dmd}}$), fuel flow rate to the engine ($\dot{m}_f$) and the torque demand from the generator ($\tau_g$).

We use a dynamic programming approach in which the driving cycle for which the operation costs have to be minimized is chosen *a priori*. Such an approach is called "Deterministic" Dynamic Programming (DDP) in the literature [3], [7]. For a chosen drive cycle, the vehicle velocity ($v_{\text{act}}$) and the wheel power demand ($P_{\text{wh,dmd}}$) can be calculated before the optimization. In the remainder of this section, we present the equations governing the remaining two state variables (SOC and $\omega_e$) and two inputs ($\dot{m}_f$ and $\tau_g$), which will be optimized.

The motor power demand at the wheels ($P_{\text{m,mech}}$) can be calculated by (1). The final drive is assumed to have a constant efficiency ($\eta_{fd}$) of 97%.

$$P_{\text{m,mech}} = \frac{P_{\text{wh,dmd}}}{\eta_{fd}}$$

The motor torque demand ($\tau_m$) is in turn related to the mechanical power demand through the motor speed ($\omega_m$)

$$\tau_m = \frac{P_{\text{m,mech}}}{\omega_m}$$  \hspace{1cm} (2)

$$\eta_m = f_m(\tau_m, \omega_m), \hspace{1cm} (3.\text{a-b})$$

$$P_{\text{m,elec}} = \begin{cases} P_{\text{m,mech}}, & P_{\text{m,elec}} \geq 0 \\ P_{\text{m,mech}}\eta_m, & \text{else} \end{cases}$$  \hspace{1cm} (3.a-b)

This torque is used to calculate the motor operating efficiency ($\eta_m$) and the electrical power that has to be supplied to the motor ($P_{\text{m,elec}}$). In (3) $f_m$ is the motor efficiency map.

The electrical power supplied to the motor by the power electronics comes from the battery and the generator. Power electronics acts as a parallel bus connecting the battery, motor and the generator and it is assumed that there are no losses in its operation. Thus we get the equations

$$P_b = P_{\text{m,elec}} + P_g \hspace{1cm} (4)$$

relating the electrical power in the two machines ($P_{\text{m,elec}}$ and $P_g$) and the battery ($P_b$). As the battery is an equivalent circuit model, the equations describing its current flow ($I_b$) and rate of change of SOC are given as

$$I_b = \frac{V_{oc} - \sqrt{V_{oc}^2 - 4R_iP_b}}{2R_i}$$ \hspace{1cm} (5. a-b)

$$\frac{dSOC}{dt} = -\frac{I_b}{Q_{max}}$$ \hspace{1cm} (5. a-b)

where $V_{oc}$ and $R_i$ are the open-circuit voltage and internal resistance of the battery, which are functions of SOC. $Q_{max}$ is the maximum current capacity of the battery. The equations describing the generator are similar to that of the motor with only one-way (mechanical to electrical) power conversion.

$$\eta_g = f_g(\tau_g, \omega_g)$$ \hspace{1cm} (6. a-b)

$$P_g = \tau_g \omega_g \eta_g$$  \hspace{1cm} (6. a-b)

where $\tau_g$ and $\omega_g$ are the generator torque and speed respectively. The generator torque is an input to the model while its rotational speed is proportional to the engine speed. The generator is connected to the engine via a step up gear ratio. The equations describing engine operation are

$$\tau_e = f_e(\dot{m}_f, \omega_e)$$ \hspace{1cm} (7. a-b)

$$\frac{d\omega_e}{dt} = \frac{\tau_e - \tau_g}{J_{\text{flywheel}}}$$ \hspace{1cm} (7. a-b)

where the engine fueling rate ($\dot{m}_f$) is an input to the model, $\omega_e$ is the engine speed, which is a state variable, $\tau_e$ is the engine torque and $J_{\text{flywheel}}$ is the inertia of the flywheel. The torque output for a given fueling rate is obtained through the fueling rate map $f_e$. Equations (1-7) along with the constraints described in the next section represent the vehicle model.

### C. Model Constraints

The powertrain model has constraints related to the ratings of the components and to the power flow between the components. For example, the battery has limits on its charging and discharging power available, depending on its SOC and its capacity. In addition, there are physical
constraints on the states and inputs denoted respectively by the sets \( Z \) and \( U \) below.

\[
Z = \left\{ \omega_e_{\min} \leq \omega_e \leq \omega_e_{\max}, \quad \text{SOC}_{\min} \leq \text{SOC} \leq \text{SOC}_{\max} \right\}
\]

\[
U = \left\{ 0 \leq m_f \leq m_{f,\max}(\omega), \quad 0 \leq \tau_g \leq \tau_{g,\max}(\omega) \right\}
\]

(8. a-b)

The chosen engine model has a maximum speed rating of 4500 RPM. The SOC is restricted to be within 0.9 and 0.3 at all times. The engine fueling rate \( (m_f) \) has a maximum value which varies with engine speed and the maximum generator torque output is also a function of its speed.

D. Backward Looking Powertrain Model

The system model represented by (1-7), is discretized for simulation using an Euler discretization. The model is then simulated as a backward looking model, i.e. the model is of the form

\[
u(k)=g(x(k),x(k+1))
\]

where, given the current and future states of the model (i.e. \( x(k) \) and \( x(k+1) \) respectively) the inputs required for that transition can be calculated. This is in contrast with the more popular approach of using a forward looking model which calculates the future states given the current states and inputs,

\[
x(k+1) = f(x(k),u(k))
\]

(10)

A backward looking model can be used because the powertrain components are appropriately sized ensuring that the vehicle model satisfies all the power demands at the wheels for the chosen drive cycles. Furthermore, we show that the forward looking model is invertible, or equivalently that the set of inputs maps to the set of future states with a one-to-one correspondence. The invertible nature of the model is discussed and proved in subsection A of the Appendix. In the next section, we describe the need for this backward looking powertrain model in our novel dynamic programming implementation.

III. DYNAMIC PROGRAMMING IMPLEMENTATION

The theory behind dynamic programming as a tool for calculating the optimal control is well understood. However, numerical problems arise when implementing the algorithm. Equation (11) shows the discretized form of the Hamilton-Bellman-Jacobi (HBJ) equation.

\[
V(k,x'(k)) = \min_U \{c(x'(k),u'(k)) + V(k+1,x'(k+1))\}
\]

(11)

This equation has previously been used to solve optimal powertrain control problems [3], [6], [7] and is introduced in more detail in [15]. Here, we briefly explain how (11) has been used by researchers previously in order to justify the need for our novel DP implementation.

The \( i^{th} \) point on the state grid is represented by \( x_i \). Given a current state \( x(k) \), applying an input \( u(k) \), results in a future state \( x'(k+1) \) and one-step cost \( c(x'(k),u'(k)) \). The optimal cost-to-go (or value function) from state \( x(k) \), at time step \( k \), is given by \( V(k,x(k)) \). It should be noted that to obtain \( V(k+1,x'(k+1)) \), an interpolation is required as \( x'(k+1) \) may not lie on the state grid. Another important computational aspect that is not represented in (11) is that the constraints are usually implemented through penalty functions.

These numerical problems are specific to the system under consideration. For the series PHEV powertrain model, the following are the two major issues that arise during the implementation of the DP algorithm given by (11). First, due to the large size of the battery in a series PHEV, its SOC dynamics are very slow in terms of percentage change in SOC. For example, drawing 32.2 kW of power (which is higher than the average power demand at the wheels) causes its SOC to drop by only 0.13 % over two seconds. To accurately capture the system dynamics the SOC grid has to be "fine" (4500 grid points in our case). This large number of grid points exacerbates the computational processor requirements due to the interpolations to calculate the value function. Furthermore, the large number of grid points also result in prohibitive computer memory requirements.

The second numerical issue arises due to the implementation of constraints that bound the state variables (i.e. the constraints given by (8a)). Penalty functions are used in previous approaches to characterize constraint violations [3] , [6], [11], [12]. Owing to the interpolations of the value function, there is a leaking effect, or propagation of the large penalties imposed. Due to the long time horizon (on the order of thousands of times steps), and a finer state grid for our control optimization, the leaking effect of the penalties is more pronounced. This results in a sub-optimal solution. Thus, even though imposing penalty functions does not increase computational time or memory requirements, it produces solutions that are not optimal.

To overcome these two major challenges we use the backward looking powertrain model and implement the DP algorithm as represented by (12).

\[
V(k,x'(k)) = \min_{x(k+1),x'(k+1)} \{c(x'(k),u'(k)) + V(k+1,x'(k+1))\}
\]

(12)

In (12), \( c(x'(k),x'(k+1)) \) is the cost of transitioning from state \( x' \) at time \( k \) to \( x' \) at time \( k+1 \) and \( V(k,x'(k)) \) is the optimal cost to go from state \( x' \) at time \( k \) to the final time. To obtain the cost of transitioning from state \( x' \) at time \( k \) to \( x' \) at time \( k+1 \), we have to simulate the backward looking model, given by (8). The set \( X(k+1), x' \) defines the set of all feasible reachable points in the discretized state space at time \( k+1 \), given the current state \( x' \). The major difference in our approach is the minimization of the costs over this set \( X(k+1), x' \) and not the set \( U(k) \) as is the case in (11), i.e. we consider the minimization of costs over all possible transitions to future states rather than the minimization of costs over all possible
inputs. Since (12) is an interpretation of the HBJ equation, this formulation still results in optimal solutions. However, in this approach the optimality of the solution is dependent on the discretization $X(k+1)$, rather than on $U(k)$, which affects the optimality of previous approaches.

There are three major advantages of this approach which overcome the numerical issues outlined above in this section. First, we can evaluate all the constraints a priori without any model simulation, knowing only the current and future states ($x'(k)$ and $x'(k+1)$ respectively). After evaluating the constraints, we obtain the set of future states that are feasible, $X(k+1, x')$. This set is a very small subset of the discretized state space. This results in a lower number of model simulations at every time step as compared to the previous approaches and a computational advantage. Second, since only transitions to states in $X(k+1, x')$ are evaluated, we do not require interpolations to obtain the optimal value function at $k+1$. This results in further reduction in computational time and memory. Finally, as the constraints are evaluated a priori, penalty functions are not necessary to describe constraint violations, thereby eliminating the propagation of penalties.

IV. OPTIMIZATION RESULTS

The overarching goal of this paper is to compare two PHEV power management strategies: blending and EV/CS. To ensure fairness in this comparison, we optimize both strategies for: (i) identical drive cycles, (ii) the same optimization constraints, and (iii) the same optimization objective (namely, the total cost of fuel and electricity). The only difference between the two strategies is the fact that we allow the blending strategy to tap into combustion engine power any time, while barring EV/CS from using the engine until battery SOC reaches a threshold of 0.4.

One important goal of this research is to quantify the impact of the relative cost of fuel and electricity on the optimality of blending vs. EV/CS. To do this, we repeat the optimization-based comparison of blending and EV/CS for fuel costs ranging from $1/gallon to $4/gallon, keeping the price of electricity constant at $0.1/kWh. A second important goal of this research is to quantify the impact of driving distance on the relative optimality of blending vs. EV/CS. Previous research shows that longer driving distances are statistically correlated with higher average vehicle speeds and propulsion energy needs per mile [10], [21]. We perform our optimization study using naturalistic driving cycles that capture this important correlation. Fig. 11 in the appendix plots five naturalistic driving cycles used in this study, showing both their speed profiles and the total distances traveled.

Tables II-IV in the appendix present the results of our optimization studies in full detail. Examining these results leads to the following six major observations:

Observation #1: For long driving cycles, in a scenario where gasoline is more expensive per mile than electricity, the blending and EV/CS strategies have almost identical total optimal energy costs. Fig. 2 illustrates this result for Cycle #5 and a fuel cost of $2.5/gallon. The blending strategy consumes fuel and electricity at a relatively uniform rate over the course of the entire cycle. EV/CS, in comparison, is unable to tap into engine power till the SOC reaches the threshold of 0.4, but compensates for that by using the engine more aggressively afterwards. The ultimate effect over the entire cycle is that both strategies consume the same amounts of fuel and electricity. This result stands in sharp contrast to similar studies for power split vehicles, where blending outperforms EV/CS [6], [7].

![Comparison of optimal solutions for minimizing total dollar costs](image)

Fig. 2. Comparison of optimal solutions for minimizing total dollar costs (Highway Cycle, $c_f = 2.5$/gallon, $c_k = 10$ c/kWh)

Observation #2: Observation #1 can be explained by the fact that the series powertrain mechanically decouples the internal combustion engine from the PHEV wheels, thereby allowing both blending and EV/CS to operate the PHEV’s combustion engine at minimum BSFC. Fig. 3 illustrates this by showing

![Engine Fuel usage shown on a BSFC map for highway cycle, $c_f = 1$/gallon, $c_k = 10$ c/kWh](image)

Fig. 3. Engine Fuel usage shown on a BSFC map for highway cycle, $c_f = 1$/gallon, $c_k = 10$ c/kWh (Blending above, EV/CS below)
the fuel consumption for both strategies superimposed on the engine BSFC map. Both strategies are clearly burn the most amount of fuel in the desirable low BSFC range.

**Observation #3:** The ability of EV/CS to match blending in terms of total fuel and electricity cost over long trips is maintained even when gasoline is relatively cheap. Fig. 4 illustrates this for Cycle #5 for an unrealistically cheap gasoline price of $1/gallon. In this case, the relative cost of fuel vs. electricity favors charge sustenance. The blending strategy operates in a charge sustenance mode. The EV/CS strategy is unable to do so initially because of its inability to tap into combustion power. Once the EV/CS strategy is, however, able to tap into combustion power, it recharges the battery and attains an overall fuel and electricity consumption cost almost identical to blending.

![Graph](image1)

**Fig. 4.** Comparison of optimal solutions for minimizing dollar costs (Highway Cycle, \(c_G = 1\) gallon, \(c_E = 0.1\) c/kWh)

![Graph](image2)

**Fig. 6.** Comparison of optimal solutions for minimizing dollar costs (Medium Length Cycle 1, \(c_G = 1\) gallon, \(c_E = 10\) c/kWh)

![Graph](image3)

**Fig. 7.** Fuel and Electricity consumption graphed vs. Distance (Gas = $1/gallon, Electricity = 10 c/kWh)

**Observation #4:** The ability of EV/CS to match blending in terms of total fuel and electricity cost is insensitive to total driving distance, as long as gasoline is more expensive per mile than electricity. Fig. 5 illustrates this by repeating the optimization study from Fig. 2 for drive Cycles 1-5. Please note that for the shorter drive cycles, neither blending nor EV/CS invoke combustion power, and therefore the cost of fuel consumption for both strategies is zero.
power, but for a time duration that is insufficient for recharging the PHEV battery. Under such a scenario, the EV/CS strategy continues to consume more electricity than blending, thereby costing more overall. This is illustrated in Figs. 6 and 7. This is the only scenario in this optimization study where there is a significant difference in overall trip energy cost between blending and EV/CS.

**Observation #6:** The blending strategy performs comparably to the EV/CS strategy in terms of overall fuel and electricity cost when gasoline is cheaper than electricity per mile for long trips. Fig. 4 illustrates this by showing that the total fuel and electricity costs for blending and EV/CS. These costs converge only for Cycle #5, whose length exceeds 60 miles, when gasoline is cheaper than electricity per mile.

V. CONCLUSIONS

A fair comparison of two supervisory control strategies for PHEVs, namely Blending and EV/CS was performed using a deterministic dynamic programming (DDP) framework. The unique contributions of this work are twofold - 1) Resolving the challenges in the implementation of DDP for a series PHEV powertrain model and 2) The results, showing that there is no significant difference in the performance of the two popular control strategies for a series PHEV with minor exceptions (listed below). In addition, relevance of the findings is enhanced through the use of naturalistic drive cycles as inputs instead of commonly used federal test cycles.

For a series PHEV powertrain, due to the large size of their batteries capturing the battery’s SOC dynamics through previously used DP implementations was computationally prohibitive. The factors for the computational intractability were the interpolations required to obtain the value function and the characterization of constraints through penalty functions. These issues were resolved through a DP implementation using a backward looking powertrain model and a finer state space gridding. The use of a backward looking powertrain model avoids the need for interpolations. The ability to evaluate constraints before simulating the powertrain model avoid the need for penalty functions as only feasible state transitions are considered. The new DP implementation does not lose any optimality compared to previous formulations.

The following results were obtained from the DDP studies for a series PHEV: The performance of the optimal EV/CS control strategy in terms of combined fuel and electricity dollar costs is comparable to that of the optimal Blending control strategy except for cases when gasoline prices are low (< $1/gallon) and the trip lengths are short. Thus, under the assumption that gasoline is more expensive per mile than electricity, the EV/CS solution performs comparably to the blended solution for all driving distances. This result is in contrast with previous comparisons in the literature for a power-split PHEV, where results indicate that the EV/CS strategy performs worse than the Blended strategy. Differences in the results for the series PHEV were shown to be mainly due to the mechanical decoupling of the engine from the wheels allowing for flexible engine operation.

A. Invertibility of Powertrain Model

The state space equations describing the vehicle powertrain dynamics (1-7) can be written as

\[ x = f(x,u) \]  \tag{14}

Applying a forward Euler discretization to the time variable, the forward looking vehicle powertrain model is written as

\[ x(k + I) = f(x(k), u(k)) \]  \tag{15}

For given current states \( x(k) \), an input \( u(k) \) is applied to obtain the future states \( x(k+I) \) of the model. Inverting this model with respect to the inputs and outputs, we get the backward looking powertrain model as

\[ u(k) = g(x(k), x(k + I)) \]  \tag{16}

In (16), for given current states \( x(k) \), the input is the future states \( x(k+I) \), and the equation outputs \( u(k) \), or equivalently the input required to transition from \( x(k) \) to \( x(k+I) \). This model is essential to implement our novel DP algorithm explained in Section III above.

Before utilizing the backward looking powertrain model we ensure that the forward looking powertrain model is invertible under all conditions. In order for a function to be invertible we have to show that the function representing the forward looking powertrain model \( f \) satisfies the following two conditions. First, \( f \) must be one-to-one (i.e. given a current state we have to show that each input \( u(k) \in U \), results in a unique future state \( x(k+1) \in Z \), where \( U \) and \( Z \) are defined in (8)). Secondly, we have to show that \( f \) is onto (i.e. that the inputs in \( U \) span the set \( Z \)). The second condition (the onto relation) can be guaranteed by redefining the set \( Z \) to be only the reachable set from the set of inputs \( U \). This means that only transitions to feasible future states will be considered in the optimization. To show the one-to-one mapping, the equations representing the function \( f \) are explained in detail below.

---

**APPENDIX**

**Fig. 8. Representation of Vehicle Model and interaction between components**

Fig. 8 is a diagram describing the relationship between the powertrain components. This is a forward looking representation that we will use to show the one-to-one dependence between the set of inputs \( \{m_r(k), \tau_e(k)\} \), and the set of outputs \( \{\omega_o(k+1), \text{SOC}(k+1)\} \).
The motor electrical power demand is an external input that the supervisory controller has to satisfy. This electrical power demand is calculated using (1-3). Assuming that the losses in the power electronics are negligible, the battery power demand \( P_b \) is related to motor electric power demand \( P_{m,\text{elec, dmd}} \) and the generator power output \( P_g \) as

\[
P_g(k) = P_{m,\text{elec, dmd}}(k) + P_g(k)
\]

(17)

At every time step \( k \), for a given \( P_{m,\text{elec, dmd}} \), \( P_b \) is dependent only on \( P_g \). The generator power is in turn dependent on generator torque \( \tau_g \), which is an input and the generator speed \( \omega_g \), which is proportional to the engine speed \( \omega_e \), which is a state. For a given, \( \omega_g(k) \) and \( \tau_g(k) \), we have a unique value of \( P_g(k) \) from (18) below. A unique value of \( P_g(k) \) for every \( \tau_g(k) \) is guaranteed because of the monotonic relationship between these variables due to the nature of the generator efficiency map. This monotonic relationship between \( P_g(k) \) and \( \tau_g(k) \) is graphically verified and shown in Fig. 9.

\[
\eta_g(k) = f_g(\tau_g(k), \omega_g(k))
\]

\[
P_g(k) = \tau_g(k)\eta_g(k)
\]

(18)

Since \( P_g \) is uniquely determined by \( \tau_g \) and \( P_b \) is uniquely determined by \( P_g \), every input \( \tau_g(k) \) results in a unique value of \( P_b(k) \).

The equations describing battery dynamics are given in (19).

\[
I_b(k) = \frac{V_m(k) - \sqrt{V_m^2(k)^2 - 4R(k)P_b(k)}}{2R(k)}
\]

\[
SOC(k+1) = SOC(k) - \frac{I_b(k)\Delta t}{Q_{\text{max}}}
\]

(19)

There is a monotonic relationship between \( P_b \) and \( I_b \), as well as between \( I_b \) and \( SOC(k+1) \). Thus every battery power input \( P_b \), results in a unique \( SOC(k+1) \), for a given \( SOC(k) \). From the above arguments, we conclude that every generator torque \( \tau_g(k) \) results in a unique future state of charge \( SOC(k+1) \) for a given pair of current states \( \{\omega_g(k), SOC(k)\} \).

To model the engine and flywheel dynamics and to show that each pair of inputs \( \{\dot{m}_f(k), \tau_e(k)\} \) results in a unique value of future engine speed state \( \omega_e(k+1) \), given a current engine speed \( \omega_e(k) \), we use (20)

\[
\tau_e(k) = f_e(\dot{m}_f(k), \omega_e(k))
\]

\[
\omega_e(k+1) = \omega_e(k) + \frac{\tau_e(k) - \tau_g(k)}{J_{\text{flyhl}}}
\]

(20)

Again the map \( f_e \) is graphically verified to be one-to-one in Fig. 10 and hence results in a unique engine torque \( \tau_e(k) \) for every fuel flow rate input \( \dot{m}_f(k) \), given \( \omega_e(k) \).

![Fig. 10. Monotonic relationship between engine fueling rate and engine torque output](image-url)

From our above discussion (i.e. the discussion before (19)), we have shown that every generator torque input \( \tau_g(k) \) results in a unique future state of charge \( SOC(k+1) \) for a given pair of current states \( \{\omega_g(k), SOC(k)\} \). For this \( \tau_g(k) \) and pair of current states \( \{\omega_g(k), SOC(k)\} \), every fuel flow rate input \( \dot{m}_f(k) \) results in a unique torque produced by the engine and hence a unique \( \omega_e(k+1) \). In other words it is possible to obtain the same \( \omega_e(k+1) \) for two different sets of inputs, but \( SOC(k+1) \) is uniquely decided by each \( \tau_g(k) \). This means that the pair of future states \( \{\omega_e(k+1), SOC(k+1)\} \) is uniquely determined by each input pair \( \{\dot{m}_f(k), \tau_e(k)\} \).

This ensures that when the forward looking powertrain model is inverted, for every future state considered in the backward looking model we can obtain the set of inputs required to transition to that future state, and the set of inputs will be uniquely determined. Thus ensuring the existence of the function \( g \), representing the backward looking model in (16).
B. Graphs and Tables

![Graphs and Table](image)

Fig. 11. Drive cycles used for optimization graphed as function of distance

### Table II

<table>
<thead>
<tr>
<th>Gas and Electricity Prices</th>
<th>Optimal Blending</th>
<th>Optimal EV/CS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gas (kg)</strong></td>
<td><strong>Electric (kWh)</strong></td>
<td><strong>Total Cost ($)</strong></td>
</tr>
<tr>
<td>1$/gal, 10 c/kWh</td>
<td>4.44</td>
<td>0</td>
</tr>
<tr>
<td>1.2$/gal, 10 c/kWh</td>
<td>3.89</td>
<td>2.23</td>
</tr>
<tr>
<td>1.5$/gal, 10 c/kWh</td>
<td>2.54</td>
<td>7.95</td>
</tr>
<tr>
<td>2.5$/gal, 10 c/kWh</td>
<td>2.54</td>
<td>7.95</td>
</tr>
<tr>
<td>4$/gal, 10 c/kWh</td>
<td>2.54</td>
<td>7.95</td>
</tr>
</tbody>
</table>

### Table III

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Distance (miles)</th>
<th>Optimal Blending</th>
<th>Optimal EV/CS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gas (kg)</td>
<td>Electric (kWh)</td>
<td>Total Cost ($)</td>
</tr>
<tr>
<td>short 1</td>
<td>7.3</td>
<td>0.37</td>
<td>0</td>
</tr>
<tr>
<td>short 2</td>
<td>24.03</td>
<td>1.43</td>
<td>0</td>
</tr>
<tr>
<td>medium 1</td>
<td>34.43</td>
<td>2.41</td>
<td>0</td>
</tr>
<tr>
<td>medium 2</td>
<td>44.2</td>
<td>2.86</td>
<td>0</td>
</tr>
<tr>
<td>long</td>
<td>62.49</td>
<td>4.44</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table IV

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Distance (miles)</th>
<th>Optimal Blending</th>
<th>Optimal EV/CS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gas (kg)</td>
<td>Electric (kWh)</td>
<td>Total Cost ($)</td>
</tr>
<tr>
<td>short 1</td>
<td>7.3</td>
<td>0</td>
<td>1.52</td>
</tr>
<tr>
<td>short 2</td>
<td>24.03</td>
<td>0</td>
<td>5.69</td>
</tr>
<tr>
<td>medium 1</td>
<td>34.43</td>
<td>0.52</td>
<td>7.73</td>
</tr>
<tr>
<td>medium 2</td>
<td>44.2</td>
<td>0.93</td>
<td>7.73</td>
</tr>
<tr>
<td>long</td>
<td>62.49</td>
<td>2.54</td>
<td>7.95</td>
</tr>
</tbody>
</table>

REFERENCES


[14] D. Kum, H. Peng and N. Bucknor, "Optimal Control of the Plug-In HEV for Fuel Economy under Various Travel Distances," 6th IFAC Symposium on Advances in Automotive Control, Munich, Germany


