Incorporating Drivability Metrics into Optimal Energy Management Strategies for Hybrid Vehicles

Daniel F. Opila, Deepak Aswani, Ryan McGee, Jeffrey A. Cook, and J.W. Grizzle

***Abstract*— Hybrid Vehicle fuel economy performance is highly sensitive to the energy management strategy used to select among multiple energy sources. Optimal solutions are easy to specify if the drive cycle is known a priori. It is very challenging to compute controllers that yield good fuel economy for a class of drive cycles representative of typical driver behavior. Additional challenges come in the form of constraints on powertrain activity, like shifting and starting the engine, which are commonly called "drivability" metrics. These constraints can adversely affect fuel economy. The benefits of including drivability restrictions in a Shortest Path Dynamic Programming (SPDP) formulation of the energy management problem are investigated for the first time. It is shown that this method yields up to 10% fuel economy improvement on a representative parallel electric hybrid when compared to a simpler instantaneous optimization formulation. This result is obtained by comparing a SPDP controller designed for drivability to a second SPDP controller, designed for fuel economy only, that uses an additional instantaneous optimization step for the incorporation of drivability. The results also quantify the tradeoff between drivability and fuel economy.**

I. INTRODUCTION

Hybrid vehicles are becoming more and more common in the automotive marketplace today. The most common type is the electric hybrid, which consists of an internal combustion engine (ICE), a battery, and at least one electric machine (EM). Hybrids are built in several configurations including series, series-parallel, and the parallel configuration considered here. Since the initial debut of modern production hybrids in 1997, researchers have been working to improve the control algorithms for better fuel economy. Hybrid vehicles are characterized by multiple energy sources; the control strategy to select among these multiple energy sources is termed "Energy Management." An excellent overview of this area is available in [8].

The optimal solution to the energy management problem is readily computable if the drive cycle is exactly known in advance. While this is rarely the case for the general driving population, optimization over fixed drive cycles is useful for benchmarking purposes and initial component selection. When the drive cycle is not specified a priori, the design of the energy management system becomes much more challenging.

Initially, many industrial and academic solutions to this problem were rule-based. Engineers wrote a set of control laws or rules that tended to improve fuel economy. A common example of this is "load-leveling," which uses the battery and electric machine to keep the ICE operating near a point of high efficiency. Later, design methods built around the on-line minimization of an instantaneous (static) cost function were introduced; see [8]. For example, a method termed "Equivalent Consumption Minimization Strategy" (ECMS) poses an on-line instantaneous optimization of fuel economy to trade off battery usage vs. fuel consumption [6]. The controller is causal, but depends on a cycle-dependent weighting factor that must be selected ahead of time. To get around this problem, the method was later modified to include a weighting factor that is adjusted on-line based on driving conditions [5].

In [3], deterministic dynamic programming over a pre-specified drive cycle was used to develop energy management control strategies. Causal control laws were successfully extracted from the nominal non-causal dynamic programming solution, and were demonstrated to deliver very good performance on a hybrid electric delivery vehicle [4]. There was, however, no a priori guarantee that this would be possible in general, and the method to extract the causal controller was time consuming.

The technique used here is Shortest Path Stochastic Dynamic Programming [2], [9]. This method uses a Markov chain to represent the set of possible drive cycles. It is noticeably different from the instantaneous minimization used in ECMS in that the controller minimizes both a current cost and an expectation of the future cost, represented by a value function. The resulting optimal controllers are automatically causal, so no post-facto extraction process is necessary, as it was in [3].

In addition to fuel economy, the customer's perception of the drivertrain's performance is another key issue in designing the energy management system. In this context, customers are concerned with the vehicle's shifting, pedal response, etc. These vehicle characteristics are commonly termed "drivability." Most of the past work in hybrid energy management has focused primarily on fuel economy. In this paper we address the "basic" drivability issues of gear selection and engine on/off, rather than more detailed driveline dynamics as in [7].

Most previous attempts at addressing these basic drivability issues have been based on instantaneous, second-by-

D.F. Opila, J.A. Cook, and J.W. Grizzle were supported by a grant from Ford Motor Company.

Daniel Opila is with the Dept. of Mechanical Engineering, University of Michigan and is supported by a NDSEG fellowship. dopila@umich.edu

Deepak Aswani and Ryan McGee are with Ford Motor Company, Dearborn, MI.

Jeffrey Cook and Jessy Grizzle are with the Dept. of Electrical Engineering, University of Michigan. {jeffcook,grizzle}@umich.edu

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship.

second, optimization methods. This paper is unique in that it directly includes drivability and fuel economy over the entire drive cycle in the optimal controller design. Stochastic Dynamic Programming is used as a controller design method. Drivability costs are considered not only in the current time step (like on-line optimization methods), but also the future expectation. Simulations are conducted for a prototype production vehicle. Including drivability restrictions in the full controller design is shown to yield approximately 10% fuel economy improvement compared to a fuel-optimized controller with drivability restrictions implemented as an instantaneous optimization.

II. Vehicle Model

A. Vehicle Architecture

The vehicle model studied in this paper is a parallel electric hybrid. A 2.4 L diesel engine is coupled to the front axle through a clutched 6-speed automated manual transmission. An electric machine is directly coupled to the rear axle through a fixed gear ratio without a clutch, therefore the electric machine is always rotating at a speed proportional to vehicle speed. Energy is stored in a 1.5 kWh battery pack. The system parameters are listed in Table I.

TABLE I

Vehicle Parameters

Engine Displacement	2.4 L
Max Engine Power	120 kW
Electric Machine Power	35 kW
Battery Capacity	1.5 kWh
Battery Power Limit	34 kW
Vehicle Mass	1895 kg

B. Modeling Assumptions

For computational reasons, the vehicle model must be as simple as possible. The vehicle model used here contains the minimum functionality required to model the vehicle behavior of interest on a second-by-second basis. Dynamics much faster than the sample time of 1s are ignored. Long-term transients that only weakly affect performance are also ignored; coolant temperature is one example.

The dynamics of the internal combustion engine are ignored; it is assumed that the engine torque exactly matches valid commands and the fuel consumption is a function only of speed, ω_{ICE}, and torque, T_{ICE}. The fuel consumption F is derived from a lookup table based on dynamometer testing,

$$Fuel\,flow = F(\omega_{ICE}, T_{ICE}).$$

The automated manual transmission has discrete gears and no torque converter. Losses in this highly efficient transmission [1] are ignored. The engine speed is assumed directly proportional to wheel speed based on the current transmission gear ratio R_g,

$$\omega_{ICE} = R_g \omega_{wheel}.$$

The engine torque T_{ICE} transmitted to the wheel is similarly assumed proportional to wheel torque based on the current gear ratio R_g. The electric machine torque T_{EM} transmitted to the wheel is proportional to the constant EM gear ratio R_{EM}. The total wheel torque T_{wheel} is thus the sum of the ICE torque to the wheel $R_g T_{ICE}$ and the electric machine torque to the wheel $R_{EM} T_{EM}$,

$$R_g T_{ICE} + R_{EM} T_{EM} = T_{wheel}.$$

The engine can be turned off at any time, in which case the clutch is disengaged and engine speed is zero independent of wheel speed. Transmission gear shifts are allowed every time step (1s) and transmission dynamics are assumed negligible.

The battery system is similarly reduced to a table lookup form. The electrical dynamics due to the motor, battery, and power electronics are assumed sufficiently fast to be ignored. The energy losses in these components can be grouped together such that the change in battery State of Charge (SOC) is a function $\bar{\kappa}$ of Electric Machine speed ω_{EM}, torque T_{EM}, and battery SOC at the present time step,

$$SOC_{k+1} = \bar{\kappa}(SOC_k, \omega_{EM}, T_{EM}). \quad (1)$$

Assuming a known vehicle speed, the only state variable required for this vehicle model is the state of charge (SOC). Changes in battery performance due to temperature, age, and wear are ignored.

The control inputs for this vehicle are the IC engine torque, electric motor torque, and the gear. Given the control choices, ICE speed and EM speed are fixed given vehicle velocity. During operation, the desired wheel torque is defined by the driver. If we assume the vehicle must meet the torque demand perfectly, then the sum of the ICE and EM contributions to wheel torque must equal the demanded torque T_{demand},

$$R_g T_{ICE} + R_{EM} T_{EM} = T_{demand}.$$

With this constraint, the choice of ICE and EM torque are no longer independent. Their relationship can be expressed in several ways, including as a *Power Split Ratio* defined as the ratio of ICE power to the road power demand [3]. For computational convenience, the ICE crankshaft torque is chosen as the control input. This leaves the system control inputs as *Engine Torque* and *Transmission Gear*.

Simulation is conducted assuming a "perfect" driver. At each time step, the vehicle velocity is the desired cycle velocity. The desired road power is calculated as the exact power required to drive the cycle at that time. Now, given vehicle speed, demanded road power, and this choice of control inputs, the dynamics become an explicit function κ of the state *Battery SOC* and the two control choices as shown in Table II,

$$SOC_{k+1} = \kappa(SOC_k, T_{ICE}, Gear). \quad (2)$$

The engine fuel consumption can be calculated from the control inputs.

TABLE II
VEHICLE DYNAMIC MODEL

State	Control Inputs
Battery Charge (SOC)	Engine Torque
	Transmission Gear

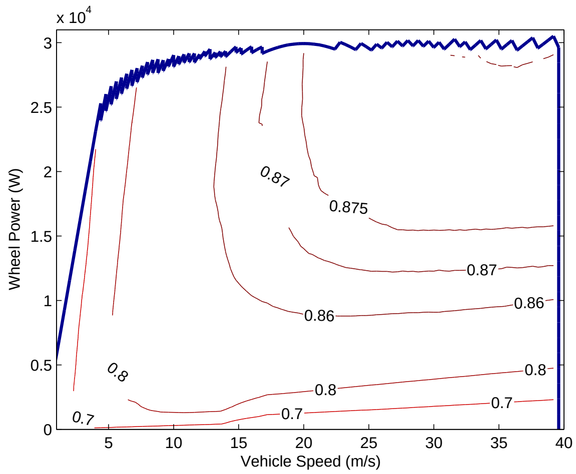

Fig. 1. Electrical System Discharge Efficiency at 0.5 SOC.

C. Component Models

The vehicle powertrain components are based on an early prototype of a production vehicle. The engine fuel consumption is calculated from a lookup table based on dynamometer tests.

The behavior of the electrical system is a function of the battery SOC, as shown in (1). Thus the system efficiency is a function of three variables. Typical electrical system discharge behavior is shown for a particular SOC of 0.5 in Figure 1; a similar table exists for the regeneration (battery charging) case. The system efficiency is a relatively weak function of SOC in the normal operating range.

The electrical efficiency for the battery discharge condition (Fig. 1) is defined as the ratio of wheel output power to the battery power,

$$\eta_{discharge} = \frac{P_{wheel}}{P_{Batt}}.$$

This efficiency definition is based on the change in the internal state of the battery and includes the losses to battery internal resistance, power electronics, and motor losses. The battery power limit for this system is roughly constant, so system losses translate to about 30% higher peak mechanical wheel power when charging compared to discharging.

D. Operational Assumptions

This model uses several assumptions about the allowed vehicle behavior.

1) The clutch in the automated manual transmission allows the diesel engine to be decoupled from the wheels. This allows the engine to shut off during forward motion.
2) There is no option to have the engine idle with the clutch disengaged; the engine shuts off.
3) There is no ability to slip the clutch for starts.
4) There are no traction control restrictions on the amount of torque that can be applied to the wheels.

III. DRIVABILITY CONSTRAINTS

A. Motivation

Customer perception is a crucial component in vehicle purchasing decisions. The driver's perception of overall vehicle response and behavior is termed "drivability." Manufacturers are very aware of this and exert significant development effort to satisfy drivability requirements. Generally speaking, drivability concerns affect designs as much as fuel economy goals.

Drivability is a rather vague term that covers many aspects of vehicle performance including acceleration, engine noise, braking, shifting activity, shift quality [7], and other behaviors. Improving drivability often comes at the expense of fuel economy. For example, optimal fuel economy for gasoline engines typically dictates upshifting at the lowest speed possible. This, however, leaves the driver little acceleration ability after the upshift. Thus upshifts are scheduled to occur at higher speeds than optimal for fuel economy. In this paper we address the "basic" drivability issues of gear selection and when to start or stop the internal combustion engine.

Current academic work in hybrid vehicle optimization primarily focuses on fuel economy. These tools are somewhat less useful to industry because of drivability restrictions in production vehicles, which fuel-optimal controllers usually violate. If these fuel-optimal controllers are used, drivability restrictions are typically imposed as a separate step.

In this paper we investigate the usefulness of optimizing for fuel economy and drivability simultaneously. By including these real-world concerns, one can generate controllers that improve performance and are one step closer to being directly implementable in production.

B. Chosen Penalties

In the context of the overall system, two significant characteristics that are noticeable to the driver are the basic behaviors of the transmission and engine. These are included in the simulation model presented in Section II. To effectively design controllers, qualitative drivability requirements must be transformed into quantitative restrictions or metrics. Drivability experts at Ford Motor Company were consulted to assist in developing numerical drivability criteria. A primary concern in engine behavior is the frequency of engine start/stop events.

For the transmission, bothersome behavior includes:

- Shift "busyness" - shifting too often or too much
- "Hunting" - rapid shifting back and forth between the same two gears
- Shift Timing-drivers have an innate expectation of shift timing and dislike unexpected deviations.

To address these issues, two baseline metrics are used to quantify behavior for a particular trip. The first is *Gear*

Events, the total number of shift events on a given trip. The second metric is *Engine Events*, the total number of engine start and stop events on a trip.

By definition, engine starts and stops are each counted as an event. Each shift is counted as a gear event, regardless of the change in gear number. A $1^{st}-2^{nd}$ shift is the same as a $1^{st}-3^{rd}$ shift. The clutch is disengaged when the engine stops, and engaged again when the engine has started. Engaging or disengaging the clutch for engine start/stop is not counted as a gear event, regardless of the gear before or after the event.

Despite the relative simplicity of these metrics, simulations have shown that they capture a wide range of vehicle behavior and are well correlated with more complicated metrics. For example, optimizing for fuel economy often leads to hunting behavior near a shift point. As the total number of shifts is reduced, hunting behavior is usually eliminated first as these frequent shifts do not significantly improve fuel economy.

IV. Shortest Path Stochastic Dynamic Programming

A. Cost Function

In order to design a controller with acceptable drivability characteristics, the optimization goal over a given trip of length T would ideally be defined as

$$\begin{gathered}\min \textstyle\sum_0^T Fuel\,flow \\ \text{such that} \\ \textstyle\sum_0^T GE \leq GE_{max}\,,\ \sum_0^T EE \leq EE_{max}\end{gathered} \tag{3}$$

where GE and EE are the number of Gear and Engine Events respectively, and GE_{max} and EE_{max} are the maximum allowable number of events on a cycle. Intuition suggests normalizing these constraints by some measure of cycle length, but engineers typically compare and design controllers on a given cycle and thus think of the problem as posed in (3).

This constrained optimization incorporates the two major areas of concern: fuel economy and drivability. Constraints of this type cannot be incorporated in the Stochastic Dynamic Programming algorithm used here because the stochastic nature of the optimization cannot directly predict performance on a given cycle. Instead, the drivability events are included as penalties, and those penalty weights are adjusted until the outcome is acceptable and meets the hard constraints. This new formulation has an optimization goal of

$$\min(\sum_0^T Fuel\,flow + \alpha \sum_0^T GE + \beta \sum_0^T EE). \tag{4}$$

The search for the weighting factors α and β involves some trial and error, as the mapping from penalty to outcome is not known a priori. Note that setting α and β to zero means solving for optimal fuel economy only.

These weighting factors allow the designer to trade off between fuel economy and the different drivability metrics. Controllers based on this cost function, however, completely drain the battery as they seek to minimize fuel. An additional cost is added to ensure that the vehicle is charge sustaining over the cycle, as described in Section IV-C. This SOC-based cost only occurs during the transition to key-off, so it is represented as a function $\phi_{SOC}(x)$ of the state x, which includes SOC. The performance index for a given drive cycle is

$$J = \sum_0^T Fuel\,flow + \alpha \sum_0^T GE + \beta \sum_0^T EE + \phi_{SOC}(x_T). \tag{5}$$

Now, to implement the optimization goal of minimizing (5), a running cost function is prescribed as a function only of the state x and control input u at the current time

$$c_{full}(x,u) = F(x,u) + \alpha \mathbf{I}_{GE}(x,u) + \beta \mathbf{I}_{EE}(x,u) + \phi_{SOC}(x) \tag{6}$$

where the function $\mathbf{I}(x,u)$ is the indicator function and shows when a state and control combination produces a Gear Event or Engine Event. Fuel use is calculated by $F(x,u)$. The SOC-based cost $\phi_{SOC}(x)$ still applies only at key-off, when the systems transitions to the key-off absorbing state as described in Section IV-C. Many other vehicle behaviors can be optimally controlled by adding appropriate functions of the form $\phi(x,u)$; a typical example is limiting SOC deviations during operation to reduce battery wear.

B. Problem Formulation

To determine the optimal control strategy for this vehicle, the Shortest Path Stochastic Dynamic Programming (SPDP) algorithm is used [2], [9]. This method directly generates a causal controller; characteristics of the future driving behavior are specified via a Markov chain rather than exact future knowledge. The system model is formulated as

$$x_{k+1} = f(x_k, u_k, w_k),$$

where $u(x_k)$ is a particular control choice in the set of allowable controls U, x_k is the state, and w_k is a random variable arising from the unknown drive cycle. Given this formulation, the optimal cost $V^*(x)$ over an infinite horizon is a function of the state x and satisfies

$$V^*(x) = \min_{u \in U} E_w[c(x,u) + V^*(f(x,u,w))], \tag{7}$$

where $c(x,u)$ is the instantaneous cost as a function of state and control; (6) is a typical example. This equation represents a compromise between minimizing the current cost $c(x,u)$ and the expected future cost $V(f(x,u,w))$. Note that the cost $V(x)$ is a function of the state only. This cost is finite for all x if every point in the state space has a positive probability of eventually transitioning to an absorbing state that incurs zero cost from that time onward.

The optimal control u^* is the control that achieves the minimum cost $V^*(x)$

$$u^*(x) = \operatorname*{argmin}_{u \in U} E_w[c(x,u) + V^*(f(x,u,w))]. \tag{8}$$

In order to use this method, the driver demand is modeled as a Markov chain. This "driver" is assigned two states:

current velocity v_k and current acceleration a_k, which are included in the full system state x. A probability distribution is then assigned to the set of accelerations at the next time step. This means estimating the function

$$P(a_{k+1}|v_k, a_k) \tag{9}$$

for all states v_k, a_k. This Markov chain captures the uncertainty in the problem, which is represented in (7) by the random variable w. The specific realization of w determines a_{k+1} in (9),

$$a_{k+1} = g(v_k, a_k, w_k) \tag{10}$$

$$P(a_{k+1}|v_k, a_k) = P(w : g(v_k, a_k, w_k) = a_{k+1}). \tag{11}$$

The transition probabilities (9) are estimated from known drive cycles that represent typical behavior, dubbed the "design cycles." The function g represents system dynamics. The variables v_k, a_k, and a_{k+1} are discretized to form a grid. For each discrete state $[v_k,\ a_k]$ there are a variety of outcomes a_{k+1}. The probability of each outcome a_{k+1} is estimated based on its frequency of occurrence during the design cycle. See [9] for more detail.

In addition to fuel economy, it is desirable to study the drivability characteristics of the vehicle. The metrics chosen are gear shifts and engine events as described in Section III. To track these metrics, two additional states are required: the *Current Gear* (1-6) and *Engine State* (on or off).

Bringing this all together, the full system state vector x contains five states: one state for the vehicle (*Battery SOC*), two states for the stochastic driver (v_k, a_k), and two states to study drivability (*Current Gear* and *Engine State*). This formulation is termed the "SPDP-Drivability" controller. A summary of system states is shown in Table III. The control u contains the two inputs *Engine Torque* and *Transmission Gear*, as described in Section II and Table II.

TABLE III

VEHICLE MODEL STATES

State	Units
Battery Charge (SOC)	[0-1]
Vehicle Speed	m/s
Current Vehicle Acceleration	m/s^2
Current Transmission Gear	Integer 1-6
Current Engine State	On or Off

Remark: As demands on controller functionality grow, so also must the complexity of the model. For example, to study fuel economy using deterministic dynamic programming, the only state required is the battery state of charge; the control inputs are *Engine Torque* and *Transmission Gear*. Two more states are required to study the stochastic version, and the drivability model requires two additional states.

C. Terminal State

As mentioned in Section IV-B, the dynamics of the system must contain an absorbing state. For this case, the absorbing state represents "key-off," when the driver has finished the trip, shut down the vehicle, and removed the key. Once the key-off event occurs, there are no furthur costs incurred, the trip is over, and the vehicle cannot be restarted. The probability of transitioning to this state is zero unless the vehicle is completely stopped ($v_k, a_k = 0$). The probability of a trip ending once the vehicle is stopped is calculated based on the design cycles. This probability is less than one because a stopped vehicle could represent a traffic light or other typical driving event that does not correspond to the end of a trip.

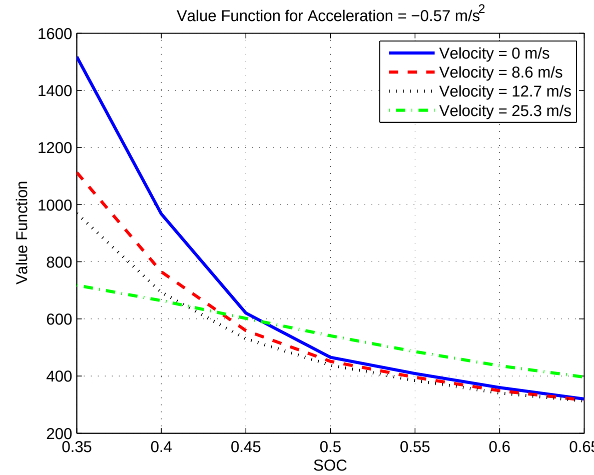

Fig. 2. Value Function $V(x)$ for several velocities and fixed acceleration.

For fuel economy certification, the battery final SOC must be close to the initial SOC or else the test is invalid. To include this in the SPDP formulation, a cost is imposed when the vehicle transitions into the key-off state and the SOC is less than the initial SOC. This penalty accrues only once, so the absorbing state has zero cost from then onwards. Here we add a quadratic penalty in SOC if the final SOC is less than the initial SOC. No penalty is assigned if the final SOC is higher than the initial SOC.

The effects of this key-off penalty are clearly visible in the value function $V(x)$. For the fuel-only case, the value function depends on the current acceleration, velocity, and SOC. Figure 2 shows $V(x)$ as a function of SOC for one particular acceleration and several velocities. This controller assumes the initial SOC is 0.5. Notice that at low velocities the value function has a pronounced quadratic shape for SOC under 0.5, but it flattens out at higher speeds. The SOC penalty only occurs at key-off, which can only occur at zero speed. Thus the SOC key-off penalty strongly affects the value function at low speeds, when there is a large probability of key-off in the near future. At higher speeds, there is little chance of key-off anytime soon, so the SOC penalty only weakly affects the value function.

D. Computation

The main difficulty with the method of Shortest Path Stochastic Dynamic Programming lies in actually computing the value function $V(x)$. Analytical results exist for some simple classes of problems, but in most cases, the only available solution method is numerical. While the off-line computational burden is large, the on-line computation is much less demanding and can be implemented in real time.

To design this controller, there are two basic steps: calculating the value function $V(x)$ off-line, and then implementing it in a controller to drive a cycle. The on-line implementation requires calculating the current cost $c(x,u)$ in (6), calculating the set of next states x_{k+1}, and interpolating into the precomputed $V(x)$.

To quantify the additional computation required, we compare the on-line and off-line requirements of three methods: Equivalent Consumption (ECMS), a simple "local" SPDP method discussed in Section V, and the SPDP-Drivability method proposed here. All three require similar operations to calculate the current cost function, which is somewhat trivial. The main difference is in calculating the value function. ECMS requires a single constant rather than a value function and so is quite simple. The value functions for SPDP-Local and SPDP-Drivability are calculated via table interpolation both on-line and off-line. The number of points used for each table is shown in Table IV for comparison. All off-line computations can be conducted on a desktop PC.

TABLE IV
COMPARISON OF COMPUTATION REQUIREMENTS

Method	Off-Line Table Size	On-Line Table Size
Local SPDP	$2.6 \cdot 10^5$	441
Drivability SPDP	$3.1 \cdot 10^6$	5292

E. Implementable Constraints

The cost function (6) must be carefully selected with consideration for computation. Stochastic Dynamic Programming is inherently computationally intensive and can quickly become intractable. The computation burden is exponential in the number of system states; thus the cost function should depend on a minimal number of states in order to limit the computation burden.

For optimization, at each time step a penalty is assigned if either a shift or engine event occurs. The only two states required to implement this cost function are the current gear and the engine state. This cost function definition captures the required behavior with minimal additional computation. Even so, including drivability in the optimization imposes roughly a factor of ten increase in computation over the fuel-only case.

In contrast, suppose the metric of interest were based on a moving window in time, e.g. the number of engine events in the previous 30 seconds. This would require an additional system state to store the number of these events. The additional computation burden for this case would be roughly a factor of fifty over the fuel-only case.

V. ALTERNATIVE PROBLEM FORMULATION

If drivability is an issue in controller design, there are two options. The first choice is to include the drivability issues in the SPDP-Drivability controller design by using additional states, as discussed in Section IV. A second choice is to design an SPDP controller for the fuel-only case and try to address drivability after the fact. In this case, the drivability restrictions are still implemented via optimization, but they are only local in the sense that there is no estimate of the future cost [9].

This local design method is implemented as follows. The value function $V_{local}(x)$ is calculated by optimizing for fuel only using (7) and a cost $c_{local}(x,u)$ that only includes fuel $F(x,u)$ and key-off SOC $\phi_{SOC}(x)$,

$$V^*_{local}(x) = \min_{u \in U} E_w[c_{local}(x,u) + V^*_{local}(f(x,u,w))],$$
$$c_{local}(x,u) = F(x,u) + \phi_{SOC}(x).$$

Recall that the only states required are velocity, acceleration, and SOC; this makes the computation much easier. The real-time controller is then implemented using the reduced dimension value function $V_{local}(x)$, but the full dimensional cost function (6) that includes drivability. The real time-controller must still track the full set of states, but it is much easier in real-time than when calculating $V(x)$. The real-time controller is then

$$u^*(x) = \operatorname*{argmin}_{u \in U} E_w[c_{full}(x,u) + V_{local}(f(x,u,w))].$$

This method is termed a "local" controller.

The main difference between the two controller design options is computation. Solving for the value function with the SPDP-Drivability controller requires about 10 times more computation than the local method. Using a local controller saves significant computation, but the result is sub-optimal and likely sacrifices some amount of performance. The main contribution of this paper is to determine if the performance benefit of using SPDP-Drivability controllers justifies the increased off-line computation.

VI. COMPARISON OF SPDP TO THE EQUIVALENT CONSUMPTION MINIMIZATION STRATEGY (ECMS)

One of the most well known optimization methods is known as the "Equivalent Consumption Minimization Strategy" (ECMS). This method optimizes for fuel economy only; it requires little computation and is easy to implement. At each time step, the controller minimizes a function that trades off battery usage vs. fuel,

$$u^*(x) = \operatorname*{argmin}_{u \in U}[Fuel(x,u) + \lambda \Delta SOC(x,u)]. \quad (12)$$

The design parameter is the weighting factor λ, which represents the relative value of battery charge in terms of fuel. The difficulty arises in calculating this weighting factor as it is highly cycle dependent.

Consider now the SPDP algorithm for the fuel only case. The cost function $c(x,u)$ in (7) is not a function of the random variable w and can be removed from the expectation. The value function $V(x)$ can be linearized about the operating point, transforming (8) into (13). This is a valid approximation because SOC only changes slightly at each time step,

$$u^*(x) = \underset{u \in U}{\operatorname{argmin}} [c(x,u) + \frac{\partial Q(x,u)}{\partial SOC} \Delta SOC)] \quad (13)$$

where

$$Q(x,u) = E_w[V(f(x,u,w)].$$

Notice that the local slope of the value function $\frac{\partial Q}{\partial SOC}$ in (13) is equivalent to the weighting factor λ in (12). The SPDP algorithm has the same structure as the ECMS method, but the weighting factor is a function of several variables. There is a variant of ECMS method called Adaptive ECMS (A-ECMS) in which the weighting factor is allowed to change over time based on the current driving conditions [5]. A-ECMS is even more similar to the SPDP algorithm in that both methods have a non-constant weighting factor.

This relationship is clearly illustrated by again studying the value function $V(x)$ as a function of SOC for fixed acceleration and velocity shown in Figure 2. The local slope of $V(x)$ in the figure is exactly the weighting factor $\frac{\partial Q}{\partial SOC}$ in (13) and analogous to λ in (12).

VII. Main Results

The main purpose of this paper is to quantify the benefits of including drivability in the full optimization. To that end, two sets of controllers are designed: a set of SPDP-Drivability controllers as in Section IV, and a set of local controllers as in Section V. Many different controllers are designed, each with different drivability penalties. In the end result, one can compare the effectiveness of optimization for drivability using the SPDP-Drivability vs. the Local method.

Both sets of controllers are designed and simulated on the Federal Test Procedure (FTP) cycle. These controllers are causal; the real-time implementation only has knowledge of the drive cycle statistics. Each individual controller is simulated and the metrics of interest are recorded.

There are two major ways to compare the results. The first method is to simply tabulate the total cost of a cycle based on the cost function (6). Then for each set of drivability penalties α and β, compare the total accrued cost of the local to the SPDP-Drivability controllers. This method answers the question: "Given the cost of drivability events, which method provides a better controller?"

The second method takes a different approach. Both local and SPDP-Drivability controllers are found that produce a given number of gear and engine drivability events. The fuel economy of the two is then compared. This method answers a different question: "Given a desired drivability performance on a certain cycle, which controller yields better fuel economy?" This second method is used here as it is more realistic: it more closely mimics the natural constrained optimization formulation (3). This method allows controllers to be selected by making informed judgements about drivability events. The number of events can be benchmarked against existing vehicles, and engineers can easily judge "too many" or "too few."

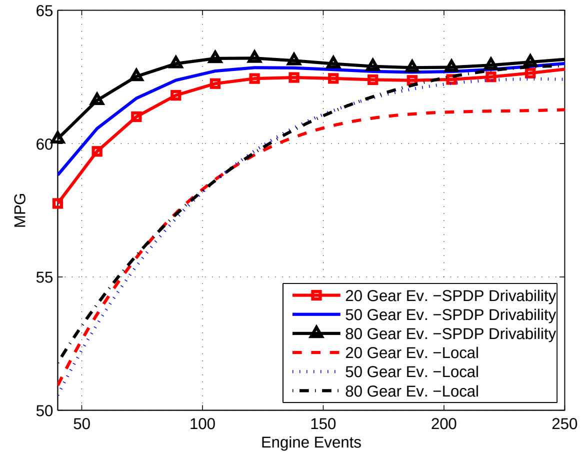

Fig. 3. Comparison of "Local" and "SPDP-Drivability" Controllers on FTP.

The results of this comparison are shown in Figure 3. This figure shows the fuel economy obtainable for a given number of gear and engine events using the two different control methods. The SPDP-Drivability controller yields performance improvement of up to 10%. This performance improvement occurs exactly in the region of interest for production vehicles: typical vehicles have 50-80 engine events and 60-80 gear events on FTP.

Figure 3 also clearly illustrates the tradeoffs between drivability and fuel economy. This allows an intelligent selection of the desired operating point. The simplest purpose of this figure is to select the best controller for given drivability criteria. However, the figure also shows the sensitivity to changes about that operating point. For example, the figure shows a high sensitivity to engine events. The designer may choose to increase fuel economy by allowing more engine activity. Similarly, suppose the initial operating point has 100 shifts. The designer knows that fuel economy is insensitive to gear activity in this vehicle and can choose to decrease shift activity with little loss of fuel economy. The results here depend on the hardware in question; other vehicle configurations may show different characteristics.

To generate Figure 3, a large number of controllers were generated and simulated. The results were then fit with a response surface to produce the curves shown. The raw results of these simulations are shown in Figure 4. Note that the fuel economy is a function of both gear and engine events, so this is naturally a 3-D table. In this case, fuel economy is relatively insensitive to gear events, so the data are shown as a function of engine events only.

Due to the stochastic nature of the optimal control problem, the final SOC is not guaranteed to end at any particular value. The final SOC for these simulations is always close to the initial SOC, but there is variation. Ignoring this variation could cause false fuel economy estimates. In this case, the SPDP-Drivability controllers not only got better fuel economy, but tended to have a higher final SOC. Thus the

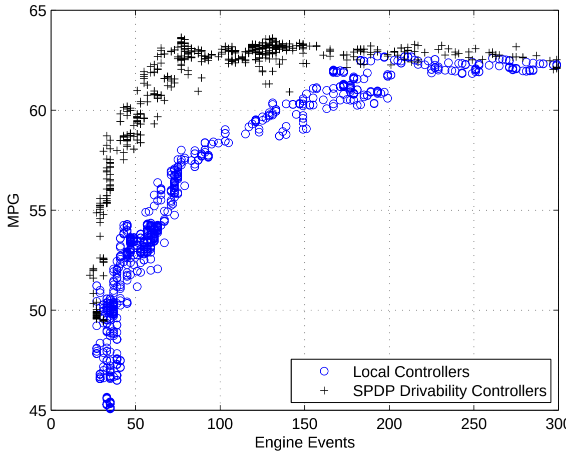

Fig. 4. Comparison of "Local" and "SPDP-Drivability" Controllers on FTP with Fuel Economy uncorrected for final SOC.

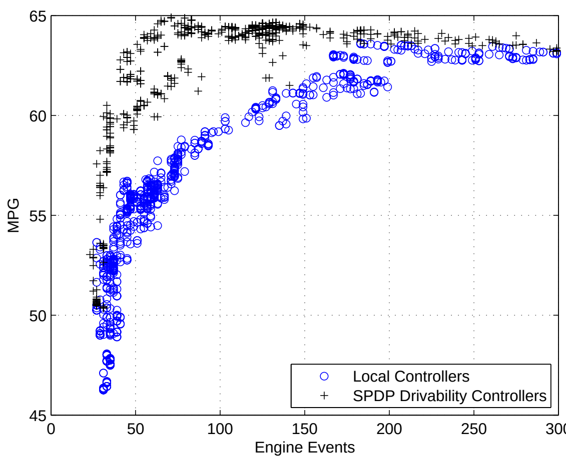

Fig. 5. Comparison of "Local" and "SPDP-Drivability" Controllers on FTP with Fuel Economy adjusted based on final SOC.

conclusions comparing the two controller types are valid. In order to make a more reliable comparison, one must estimate the relative value of SOC deviations at the end of a trip. For this vehicle and this particular cycle, a final SOC that is 1% higher than the inital SOC roughly corresponds to a 0.3 MPG decrease in fuel economy. The final fuel economy values are corrected based on this estimate and the final SOC to yield the data shown in Figure 5.

VIII. Conclusions

The optimal energy management strategy for a hybrid vehicle depends on the future drive cycle. This knowledge is not available in practice, leaving a challenging control design problem. One practical and successful option is an on-line optimization to minimize a cost function that depends on the current state and control. A second, more computationally intensive method is Stochastic Dynamic Programming (SPDP), which also includes a stochastic estimate of future costs.

Drivability is an important consideration in designing a deployable controller. When using SPDP, these restrictions could be implemented in two ways. The first and easiest option is to design a SPDP controller for fuel economy only, and add an additional on-line local optimization to include drivability. This method is very similar to how one would include drivability in an Equivalent Consumption Minimization Strategy (ECMS) type controller. The second option is to design a SPDP-Drivability controller with a full set of states that include drivability, but this method requires significant additional computation. The results presented here show that using this SPDP-Drivability controller with a representative vehicle simulation yields significant (up to 10%) fuel economy improvements that can justify the increased off-line computational complexity compared to the local on-line optimization. Quantifying this benefit is the major contribution of this paper.

When designing an energy management controller, the designer has a range of choices that trade off controller complexity for performance and functionality. Local optimization methods like ECMS are simple to design and implement, but other methods are more robust. Optimizing for fuel use with Shortest Path Dynamic Programming is useful to add robustness, but requires significantly more computation than ECMS. Including drivabiltiy in the SPDP controller imposes a factor of 10 increase in off-line complexity, but yields performance improvements on the order of 10% over the fuel-only SPDP local case under drivability constraints.

References

[1] M. A. Kluger and D. M. Long, "An overview of current automatic, manual, and continuously variable transmission efficiencies and their projected future improvements," in *Proceedings, SAE International Congress and Exposition*, 1999.

[2] C.-C. Lin, H. Peng, and J. Grizzle, "A stochastic control strategy for hybrid electric vehicles," in *Proceedings of the American Control Conference*, 2004.

[3] C.-C. Lin, H. Peng, J. Grizzle, and J.-M. Kang, "Power management strategy for a parallel hybrid electric truck," *IEEE Transactions on Control Systems Technology*, vol. 11, no. 6, pp. 839–849, 2003.

[4] C.-C. Lin, H. Peng, J. Grizzle, J. Liu, and M. Busdiecker, "Control system development for an advanced-technology medium-duty hybrid electric truck," in *Proceedings of the International Truck & Bus Meeting & Exhibition, Ft. Worth, TX, USA*, 2003.

[5] C. Musardo, G. Rizzoni, and B. Staccia, "A-ECMS: An adaptive algorithm for hybrid electric vehicle energy management," in *Proc. European Control Conference Decision and Control CDC-ECC '05.*, 2005.

[6] G. Paganelli, M. Tateno, A. Brahma, G. Rizzoni, and Y. Guezennec, "Control development for a hybrid-electric sport-utility vehicle: strategy, implementation and field test results," in *Proc. American Control Conference*, vol. 6, 2001.

[7] P. Pisu, K. Koprubasi, and G. Rizzoni, "Energy management and drivability control problems for hybrid electric vehicles," in *Proc. European Control Conference Decision and Control CDC-ECC '05.*, 2005.

[8] A. Sciarretta and L. Guzzella, "Control of hybrid electric vehicles," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 60–70, 2007.

[9] E. Tate, J. Grizzle, and H. Peng, "Shortest path stochastic control for hybrid electric vehicles," *International Journal of Robust and Nonlinear Control*, vol. 18, pp. 1409–1429, 2008.