Incorporating Drivability Metrics into Optimal Energy Management Strategies for Hybrid Vehicles Part 1: Model, Methods, and Government Test Cycles

Daniel F. Opila, Xiaoyong Wang, Ryan McGee, R. Brent Gillespie, Jeffrey A. Cook, and J.W. Grizzle

Abstract—**Hybrid Vehicle fuel economy performance is highly sensitive to the "Energy Management" strategy used to regulate power flow among the various energy sources and sinks. 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. In this paper, drivability restrictions are included in a Shortest Path Stochastic Dynamic Programming (SPSDP) formulation of the energy management problem to directly address this tradeoff and generate optimal, causal controllers. The controllers are evaluated on Ford Motor Company's highly accurate proprietary vehicle model on the FTP and NEDC government drive cycles, and compared to a controller developed by Ford for a prototype vehicle. The SPSDP-based controllers improve fuel economy more than 15% compared to the industrial controller on government test cycles. In addition, the SPSDP-based controllers can directly quantify tradeoffs between fuel economy and drivability. The theoretical basis of the SPSDP method is related to the popular Equivalent Consumption Minimization Strategy (ECMS). This paper is the first of two parts and focuses on methods and results on government test cycles, while the second part studies this method in a broader and more practical sense, including simulation on large numbers of real-world drive cycles.**

I. Introduction

Hybrid vehicles have become increasingly popular in the automotive marketplace in the past decade. 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, parallel, and the series-parallel configuration considered here. Hybrid vehicles are characterized by multiple energy sources; the strategy to control the energy flow among these multiple sources is termed "Energy Management" and is crucial for good fuel economy. An excellent overview of this area is available in [4].

This energy management problem has been studied extensively in academic circles. Varying control design methods are used, including rule-based [5], [6], [7], [8], neural networks [9], game theory [10], and fuzzy logic [11]. There are many proposed methods available for both the non-causal (cycle known in advance) and causal (cycle unknown in advance) cases [12], [13], [14], and those with partial future information [15], [16].

The most commonly used optimization strategies are the Equivalent Consumption Minimization Strategy (ECMS) [17], [18], [19], [20], [21], [22] and Stochastic or Deterministic Dynamic Programming (SDP) methods [23], [24], [25], [26]. The majority of existing work focuses on optimal controllers that seek to minimize fuel consumption. These controllers can be undesirable in practice because of excessive powertrain activity like shifting and staring the engine [27], [28], [29], [30]. These powertrain behaviors are known as "drivability." Previous scholars have recognized this problem and incorporated penalties on engine starts in an ECMS formulation [18], which is suboptimal.

In this paper, drivability restrictions are included in a Shortest Path Stochastic Dynamic Programming (SPSDP) formulation of the energy management problem to directly address this tradeoff and generate optimal, causal controllers. SPSDP is a specific formulation of SDP. This is the first time restrictions on powertrain activity have been incorporated into a provably optimal controller.

Most of the existing literature focuses on theoretical methods with a few test cases, typically simulations of a representative vehicle on the certification test cycles. There are

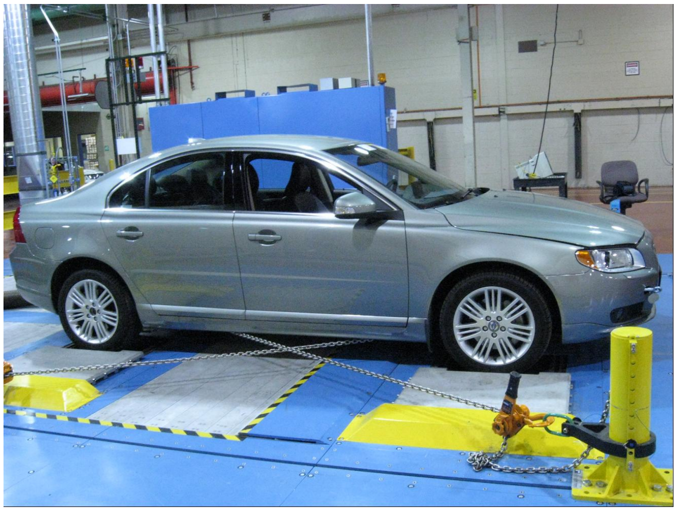


Fig. 1: The Prototype Hybrid: A Modified Volvo S-80.

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship. D.F. Opila is supported by NDSEG and NSF-GRFP fellowships. D.F. Opila, J.A. Cook, and J.W. Grizzle were supported by a grant from Ford Motor Company. Portions of this work have appeared in [1], [2], [3], [3].

Daniel Opila and Brent Gillespie are with the Dept. of Mechanical Engineering, University of Michigan.{dopila,brentg}@umich.edu Xiaoyong Wang and Ryan McGee are with Ford Motor Company, Dearborn, MI. Jeffrey Cook and Jessy Grizzle are with the Dept. of Electrical Engineering and Computer Science, University of Michigan. {jeffcook,grizzle}@umich.edu

relatively few results showing how these algorithms perform in practice [31], [32], [33], [34], [35], [14] and how they compare to the existing industrial state of the art. It is unclear how much of the existing literature is used by industry in actual production vehicles. The controllers developed in this paper and its companion [36] are extensively evaluated for real-world performance and robustness. They are compared to an industrial controller on both government test cycles and real-world driving using approximately 500,000 simulated drive cycles.

The theoretical basis of the SPSDP method is related to the more straightforward ECMS method. Despite vast differences in problem setup, both methods address the fundamental question in the energy management problem: the future value of battery charge in terms of fuel consumption. SPSDP has advantages in its ability to incorporate other behaviors along with fuel economy, which can be done suboptimally with ECMS.

The SPSDP controller calculations were conducted using a two-step optimization strategy that preserves optimality while decreasing computation by at least a factor of 10, thereby allowing the design to be conducted on a desktop PC. This method is applicable to other vehicle configurations that have multiple actuator degrees of freedom.

In addition to generating a class of optimal controllers, the SPSDP method allows direct study of the tradeoffs between different performance goals, specifically drivability and fuel economy. The ability to easily generate Pareto tradeoff curves is perhaps just as interesting as a specific fuel economy benefit. The designer can generate both the maximum attainable performance curve and causal controllers that generate that performance. Drivability is studied in this paper, but one could also study the fuel economy tradeoff with other attributes like emissions, battery wear, or engine noise characteristics.

The controllers generated through SPSDP are directly implementable in real-time and are provably optimal. One place where SPSDP can have a major impact is in controller design for new vehicles. Significant effort is required to develop a controller for a new drivetrain, especially with a completely new architecture (e.g., Series-Parallel vs. Power Split). The SPSDP method can automatically generate a provably optimal controller for a given vehicle architecture and component sizing much faster than a person could do it manually. This is especially valuable early in a program during the hardware design phase.

The SPSDP design method produces causal controllers that respect constraints, perform well on both government test cycles and real-world driving, and can be directly implemented with little manual tuning. It is hoped that these results can verify the usefulness of this algorithm and take these methods from academic research papers into industrial labs and onto the road. While this work demonstrates excellent fuel economy results for the SPSDP controllers, we feel that its main contribution is the demonstration of stochastic optimization as a viable *energy management design method.*

This research is a collaborative effort between the University of Michigan and Ford Motor Company. This work uses Ford's high-fidelity vehicle simulation model [37], which is used to develop HEV control algorithms and evaluate fuel economy for production vehicles. The vehicle studied here is a modified Volvo S-80 prototype (Fig. 1) that does not match any vehicle currently on the market. As a benchmark, Ford provided a controller developed for this prototype vehicle. This industrial controller was introduced in [2] and is termed the "baseline" controller.

The first of this two-part paper includes the problem setup and methods with initial performance evaluation: the vehicle model, development and validation of drivability metrics and constraints, controller design methods and implementation details, and results on the FTP and NEDC government test cycles. The second paper [36] studies performance in a broader and more practical sense: performance on large numbers of real-world cycles, quantification of off-line computational cost vs. performance improvement, and the cost of disallowing particular vehicle operating modes.

II. Vehicle

A. *Vehicle Architecture*

The vehicle studied in this paper is a prototype Volvo S-80 series-parallel electric hybrid and is shown schematically in Figure 2. A 2.4 L diesel engine is coupled to the front axle through a dual clutch 6-speed transmission. An electric machine, $EM1$, is directly coupled to the engine crankshaft and can generate power regardless of clutch state. A second electric machine, $EM2$, is directly coupled to the rear axle through a fixed gear ratio without a clutch and always rotates 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 $EM1$ (Front)	15 kW
Electric Machine Power $EM2$ (Rear)	35 kW
Battery Capacity	1.5 kWh
Battery Power Limit	34 kW
Vehicle Mass	1895 kg

B. *Vehicle Models*

The work presented in this paper uses two different dynamic models to represent the same prototype hybrid vehicle. The first model is quite simple; it has a sample time of 1s, uses lookup tables, and has very few states. It is used primarily to design the controller and do the optimization, and is called the "control-oriented" model.

The second model comes from Ford Motor Company and uses its in-house modeling architecture. This sophisticated model is used to evaluate fuel economy and controller behavior by simulating controllers on drive cycles. This model is referred to as the "vehicle simulation" model in this paper [37].

This combination of models allows the controller to be designed on a simple model that keeps the problem feasible,

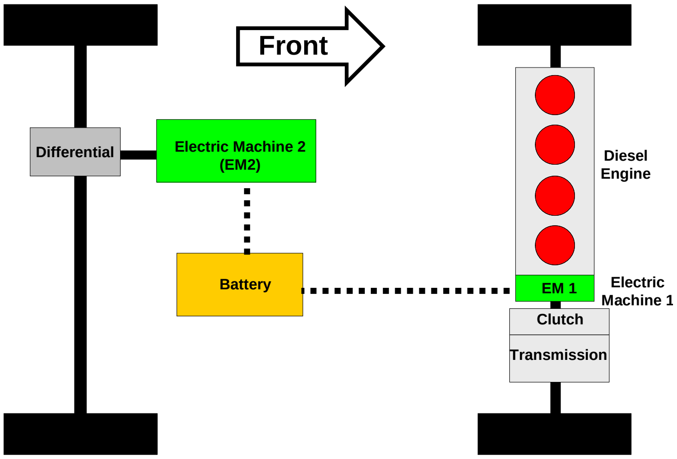


Fig. 2: Vehicle Configuration

while providing accurate fuel economy results on a highly representative model that is different from the design model.

C. Control-Oriented Model

When using Shortest-Path Stochastic Dynamic Programming, the off-line computation cost is very sensitive to the number of system states. For this reason, the model used to develop the controller 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 vehicle hardware allows three main operating conditions:

1) **Parallel Mode**-The engine is on and the clutch is engaged.
2) **Series Mode**-The engine is on and the clutch is disengaged. The only torque to the wheels is through $EM2$.
3) **Electric Mode**-The engine is off and the clutch is disengaged; again the only torque to the wheels is through $EM2$.

The model does not restrict the direction of power flow. The electric machines can be either motors or generators in all modes.

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 dual clutch transmission has discrete gears and no torque converter. The transmission is modeled with a constant mechanical efficiency of 0.95. Transmission gear shifts are allowed every time step (1s) and transmission dynamics are assumed negligible. When the clutch is engaged, the vehicle is in **Parallel Mode** and 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 electric machine $EM1$ is directly coupled to the crankshaft, and thus rotates at the engine speed ω_{ICE},

$$\omega_{EM1} = \omega_{ICE}.$$

In **Parallel Mode**, the engine torque T_{ICE} and $EM1$ torque T_{EM1} transmitted to the wheel are assumed proportional to wheel torque based on the current gear ratio R_g and the transmission efficiency η_{trans}. The rear electric machine torque T_{EM2} transmitted to the wheel is proportional to the machine's gear ratio R_{EM2} and rear differential efficiency η_{diff}. The total wheel torque T_{wheel} from both wheels is thus the sum of the ICE and $EM1$ torques to the wheel $\eta_{Trans} R_g(T_{ICE}+T_{EM1})$ and the rear electric machine $EM2$ torque to the wheel $\eta_{diff} R_{EM2} T_{EM2}$,

$$\eta_{trans} R_g(T_{ICE}+T_{EM1}) + \eta_{diff} R_{EM2} T_{EM2} = T_{wheel}. \quad (1)$$

The clutch can be disengaged at any time, and power is delivered to the road through the rear electric machine $EM2$. This condition is treated as the "neutral" gear 0, which combines with the 6 standard gears for a total of 7 gear states. If the engine is on with the clutch disengaged, the vehicle is in **Series Mode**. The engine-$EM1$ combination acts as a generator and can operate at an arbitrary torque and speed. If the engine is off while the clutch is disengaged, the vehicle is in **Electric Mode**. The clutch is never engaged with the engine off, so this mode is undefined and not used.

TABLE II: Vehicle Mode Definitions.

Mode	Clutch State	Engine State	Gear State
Electric	Disengaged	Off	0
Series	Disengaged	On	0
Parallel	Engaged	On	1-6
Undefined/not used	Engaged	Off	1-6

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 speeds ω_{EM1} and ω_{EM2}, torques T_{EM1} and T_{EM2}, and battery SOC at the present time step,

$$SOC_{k+1} = \bar{\kappa}(SOC_k, \omega_{EM1}, \omega_{EM2}, T_{EM1}, T_{EM2}). \quad (2)$$

Assuming a known vehicle speed, the only state variable required for this vehicle model is the battery SOC. Changes in battery performance due to temperature, age, and wear are ignored. 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 (1) must equal the demanded torque T_{demand},

$$T_{wheel} = T_{demand}. \quad (3)$$

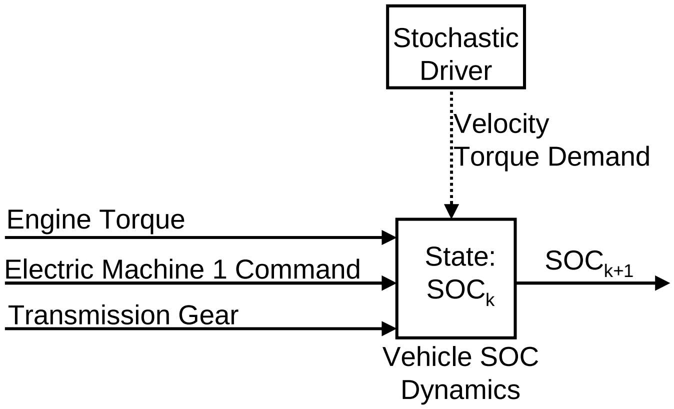


Fig. 3: Vehicle SOC Dynamics model. The three inputs are Engine Torque, Electric Machine 1 Torque, and Transmission Gear. The vehicle velocity and required torque are provided by the stochastic driver model. In Series Mode, the Electric Machine 1 command is speed rather than torque.

This adds a constraint to the control optimization, reducing the 4 control inputs to a 3 degree of freedom problem. In **Parallel Mode** the control inputs are *Engine Torque*, $EM1$ *Torque*, and *Transmission Gear*. In **Series Mode**, the electric machine command becomes $EM1$ *Speed*.

Optimization using the control-oriented model will assume a "perfect" driver. 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 three control choices as shown in Fig. 3,

$$SOC_{k+1} = \kappa(SOC_k, T_{ICE}, T_{EM1}, Gear). \quad (4)$$

In **Series Mode**, T_{EM1} is replaced with ω_{EM1}. The engine fuel consumption can be calculated from the control inputs.

Operational Assumptions:

This control-oriented model uses several assumptions about the allowed vehicle behavior.

1) The clutch in the transmission allows the diesel engine to be decoupled from the wheels.
2) There is no ability to slip the clutch for starts.
3) There are no traction control restrictions on the amount of torque that can be applied to the wheels.

D. Vehicle Simulation Model

As part of this project, Ford provided an in-house model used to simulate fuel economy. It is a complex, MATLAB/Simulink based model with a large number of parameters and states [37]. Every individual subsystem in the vehicle is represented by an appropriate block. For each new vehicle, subsystems are combined appropriately to yield a complete system.

This vehicle simulation model contains the baseline controller algorithm. To generate simulation results using this controller, an automated driver follows the target cycle using the baseline controller.

To use the vehicle simulation model with the control algorithm developed here, the SPSDP controller is implemented in Simulink by interfacing appropriate feedback and command signals: Battery State of Charge, Vehicle Speed, Engine State, Gear Command, etc. The vehicle simulation model can then be "driven" by the SPSDP controller along a given drive cycle.

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, automated shifting activity, shift quality [38], 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, but this leaves the driver little acceleration ability after the upshift. Thus upshifts are scheduled to occur at higher speeds than what is 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. Such results are somewhat less useful to industry because of drivability restrictions in production vehicles. If these fuel-optimal controllers are used, drivability restrictions are typically imposed as a separate step [1].

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. Simplified Drivability Metrics

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. Developing simple, quantitative drivability metrics was a major enabling contribution of this work. This process is discussed in detail in Part 2 of this paper [36].

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. Engaging or disengaging the clutch is not counted as a gear event, regardless of the gear before or after the event.

These simplified metrics are useful because they can be implemented in the optimization process with few additional states, generate acceptable vehicle behavior, and are well correlated with more sophisticated metrics as shown in Part 2 of this paper [36].

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

$$\begin{gathered}\min \textstyle\sum_0^T Fuel\,flow \\ \text{such that} \\ \textstyle\sum_0^T GE \leq GE_{max}\,,\, \sum_0^T EE \leq EE_{max}\end{gathered} \tag{5}$$

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. T is the time duration from “key-on,” the start of the trip, to “key-off,” the end of the trip.

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.

Controllers based only on fuel economy and drivability 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. 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 then

$$J = \sum_0^T Fuel\,flow + \alpha \sum_0^T GE + \beta \sum_0^T EE + \phi_{SOC}(x_T). \tag{6}$$

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.

Now, to implement the optimization goal of minimizing (6), a running cost function is prescribed as a function only of the state x and control input u at the current time

$$c_{full}(x,u) = F(x,u) + \alpha \mathbf{I}_{GE}(x,u) + \beta \mathbf{I}_{EE}(x,u) + \phi_{SOC}(x) \tag{7}$$

where the functions $\mathbf{I}(x,u)$ are the indicator function and show 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 $\phi_{SOC}(x)$ applies only at the end of the trip, when the system transitions to the key-off absorbing state [1].

B. Problem Formulation

To determine the optimal control strategy for this vehicle, the Shortest Path Stochastic Dynamic Programming (SPDP) algorithm is used [25], [26], [39]. This method directly generates a causal controller; characteristics of future driving behavior are specified via a finite-state Markov chain rather than exact future knowledge. The system model is formulated as

$$x_{k+1} = f(x_k, u_k, w_k), \tag{8}$$

where u_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

$$V^*(x) = \min_{u \in U} E_w[c(x,u) + V^*(f(x,u,w))], \tag{9}$$

where $c(x,u)$ is the instantaneous cost as a function of state and control; (7) is a typical example. This equation represents a compromise between minimizing the current cost $c(x,u)$ and the expected future cost $V(f(x,u,w))$. The control u is selected based on the expectation over the random variable w, rather than a deterministic cost, because the future can only be estimated based on the probability distribution of 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. Here, the absorbing state is key-off, the end of the drive cycle.

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

$$u^*(x) = \operatorname*{argmin}_{u \in U} E_w[c(x,u) + V^*(f(x,u,w))]. \tag{10}$$

Note that the disturbance w in (9) and (10) may be conditioned on the state and control input,

$$P(w|x_k, u_k). \tag{11}$$

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. The unknown disturbance w in (9) is the acceleration at the next time step, which is assigned a probability distribution. This means estimating the function

$$P(a_{k+1}|v_k, a_k) \tag{12}$$

for all states v_k, a_k. The transition probabilities (12) are estimated from known drive cycles that represent typical

[1] 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.

behavior, dubbed the "design cycles." 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, and is clearly a function of state as in (11). See [25], [26] for more detail.

To track the gear event and engine event metrics (Section III), 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-C and Table III.

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 design 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.

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.

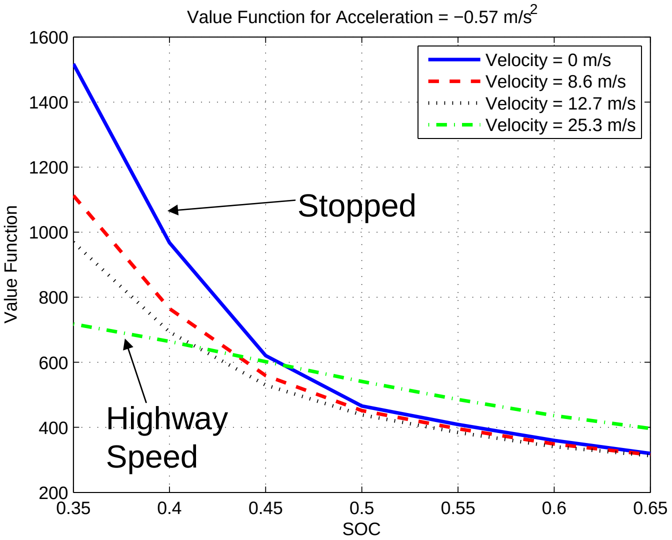


Fig. 4: Value Function $V(x)$ for several velocities and fixed deceleration at -0.57 m/s^2. The quadratic penalty on SOC strongly affects the value function at low speeds when the driver is more likely to turn the key off and end the trip.

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. Fig. 4 shows $V(x)$ as a function of SOC for one particular acceleration and several velocities, with target final SOC equal to 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 higher 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. Implementable Constraints

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 (7) should depend on a minimal number of states.

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. 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. The number of grid points required scales with the number of time steps used to specify the metric. For the 1 s update time studied here, penalizing engine events 5 seconds or less (rather than the simple on/off) would require 5 grid points for the time history, increasing the size of the state-space by the same factor of 5 over the on/off case.

Part II of this paper [36] addresses the selection and implementation of the cost function in greater detail.

V. Computation Reduction

A. Theory

Proposition: (Minimization Decomposition)

Consider a Bellman equation of the form

$$V^*(x) = \min_{\hat{u}\in\hat{U}(x),\bar{u}\in\bar{U}(x,\hat{u})} E_w[c(x,\hat{u},\bar{u}) + V^*(f(x,\hat{u},w))], \tag{13}$$

and define

$$\hat{c}(x,\hat{u}) = \min_{\bar{u}\in\bar{U}(x,\hat{u})} c(x,\hat{u},\bar{u}). \tag{14}$$

Then $V^*(x)$ satisfies (13) if and only if it satisfies

$$V^*(x) = \min_{\hat{u}\in\hat{U}(x)} E_w[\hat{c}(x,\hat{u}) + V^*(f(x,\hat{u},w))].\blacksquare \tag{15}$$

The proof and more detail are available in Appendix A. This result allows a significant reduction in computation complexity for problems that have the specific structure (13). The reduced Bellman equation (15) may be solved using only the reduced control space $\hat{U}(x)$. This structure appears quite often in energy management problems (see Appendix A).

The decomposition is often exploited, usually without explicit theoretical justification [16], [40], [41]. A typical example is the power-split configuration which uses engine power and speed as inputs without an engine speed state [40]. The fuel-minimizing engine speed ($\bar{u}$) for each engine power ($\hat{u}$) is precomputed and stored as a table (see Appendix A).

The subsection below details the physical explanation of the structure (13) for the vehicle considered in this work and how the decomposition is implemented.

B. Super Electric Machine

In comparison to previous work in [1], the addition of a second electric machine makes the computation of a SPSDP solution potentially more complex. Exploiting structure (13) and using Minimization Decomposition reduced the computational cost to that of a vehicle with a single electric machine, a 90% reduction. The addition of the second electric machine is approximately "free" in terms of computation.

The system inputs require a tradeoff between the the two electric machine torques. Define $\bar{T}_{(\cdot)}$ as the wheel torque delivered by a particular actuator. The system dynamics are only affected by the total amount of torque delivered to the wheels $\bar{T}_{SEM}$

$$\hat{u} : \bar{T}_{SEM} = \bar{T}_{EM1} + \bar{T}_{EM2} \tag{16}$$

and not by the split between the two machines T_{DEM}

$$\bar{u} : \bar{T}_{DEM} = \bar{T}_{EM1} - \bar{T}_{EM2}. \tag{17}$$

This torque splitting may be considered $\bar{u}$. Since this one degree of freedom optimization is static (i.e., independent of the dynamic states of the model including SOC), it takes the form (13) and can be computed *a priori* using (14) without loss of optimality. This reduces the dimension of the control space by one. The fundamental assumption that allows this to work is that the electric machine behaviors are dependent only on the current gear, engine state, and velocity, and not the past values of those states.

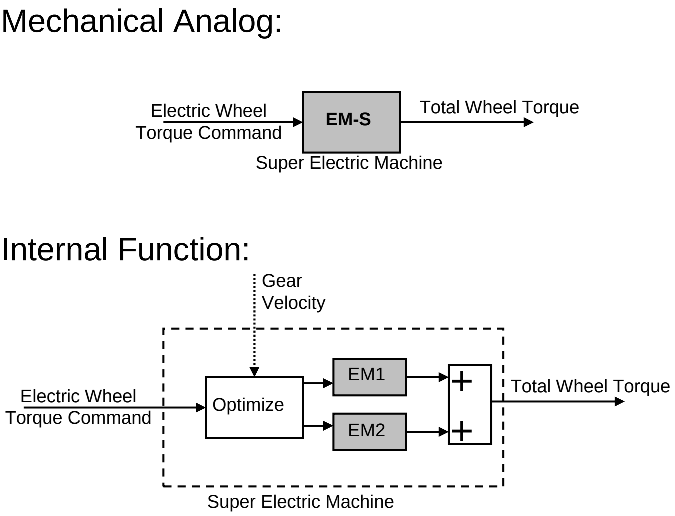


Fig. 5: Schematic diagram of a conceptual "Super Electric Machine" that optimizes the mix between the two electric machines. This allows one degree of freedom of the control optimization to be carried out off-line while maintaining the optimality of the solution.

The physical control inputs to the system are engine torque, transmission gear, EM1 torque and EM2 torque. By replacing the two electric machine commands with a single Electric wheel torque command, the SPSDP algorithm has only 3 control inputs.

A more intuitive explanation is to treat the system as a single "Super Electric Machine." This device is a black box that takes a desired wheel torque command as an input and uses the vehicle velocity and transmission gear to match the command torque with minimal electric power as shown in Fig. 5. Once the optimization is complete, this device acts just like a normal electric machine for the SPSDP optimization. Internally, the device optimizes between the two (or possibly more) electric machines and issues appropriate commands.

VI. Comparison of SPDP to the Equivalent Consumption Minimization Strategy (ECMS)

One of the most well known optimization methods for energy management in HEVs is known as the "Equivalent Consumption Minimization Strategy" (ECMS) [19], [31]. 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,

$$u_k^*(x) = \operatorname*{argmin}_{u\in U}[Fuel(x,u) + \lambda_k \Delta SOC(x,u)]. \tag{18}$$

The design parameter is the weighting factor λ_k, 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 (9) 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 (10) into (19). This is a valid approximation because SOC only changes slightly at each time step,

$$u^*(x) = \operatorname*{argmin}_{u \in U}[c(x,u) + \frac{\partial Q(x,u)}{\partial SOC}\Delta SOC)] \tag{19}$$

where

$$Q(x,u) = E_w[V(f(x,u,w))]. \tag{20}$$

Notice that the local slope of the value function $\frac{\partial Q}{\partial SOC}$ in (19) is equivalent to the weighting factor λ in (18). 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 [19]. 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 Fig. 4. The local slope of $V(x)$ in the figure is exactly the weighting factor $\frac{\partial Q}{\partial SOC}$ in (19) and analogous to λ in (18).

Fundamentally, all fuel-minimizing control algorithms must estimate the value of battery charge in terms of fuel and thus have some equivalent to the weighting factor λ. It may appear explicitly as in ECMS, or implicitly as in SPSDP. Once this weighting factor is determined, the control problem is a simple static optimization. All known information is incorporated in this weighting factor, including plant dynamics, states, and expected future driving patterns.

The basic difference between algorithms is in how they estimate this weighting factor: ECMS uses a value assigned by the designer; A-ECMS estimates the value based on battery charge and recent history; Deterministic Dynamic Programming uses exact future knowledge; and SPSDP uses estimates of cycle statistics. The dynamic programming methods like SPSDP can also optimally incorporate constraints on behavior, like the drivability metrics studied here.

VII. Simulation Procedure

SPSDP-based controllers are compared to a baseline industrial controller. SPSDP controllers are designed using the control-oriented model and evaluated using the Simulink-based vehicle simulation model. This demonstrates some robustness by using two models of the same vehicle, differing in the level of detail in their dynamics.

Both SPSDP and the baseline controllers are simulated on two government test cycles, the US Federal Test Procedure (FTP) and the New European Drive Cycle (NEDC), which are shown in Fig. 6. Procedurally, this is conducted as follows:

1) A “family” of SPSDP controllers is designed according to the methods of Section IV. A *family* is generated by fixing the model driving statistics and sweeping the 2 drivability penalties α and β in (7).
2) Each controller in the *family* is simulated on the vehicle simulation model.
3) The fuel consumption and drivability metrics are recorded.

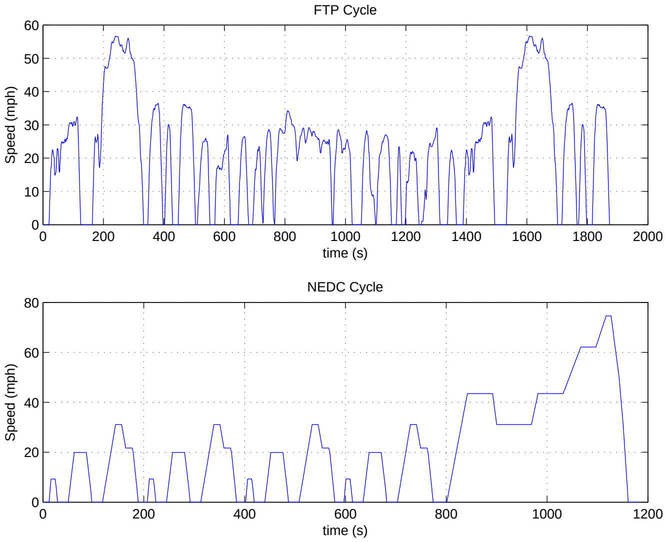


Fig. 6: FTP and NEDC cycles.

In the end, each *family* contains a few hundred individual controllers which have each been simulated on the cycle in question. Each simulation yields a data point with associated fuel economy and drivability metrics. Each controller in the *family* has different drivability and fuel consumption characteristics because of the varying drivability penalties.

Each controller is simulated on the vehicle simulation model discussed in Section II-D. The simulations are all causal, so the final SOC is not guaranteed to exactly match the starting SOC. This could yield false fuel economy results, so all fuel economy results are corrected based on the final SOC of the drive cycle. This is done by estimating the additional fuel required to charge the battery to its initial SOC, or the potential fuel savings shown by a final SOC that is higher than the starting level. This correction is applied according to

$$\Delta Fuel = C_{Batt}\Delta SOC\frac{BSFC_{min}}{\eta_{max}^{Regen}} \tag{21}$$

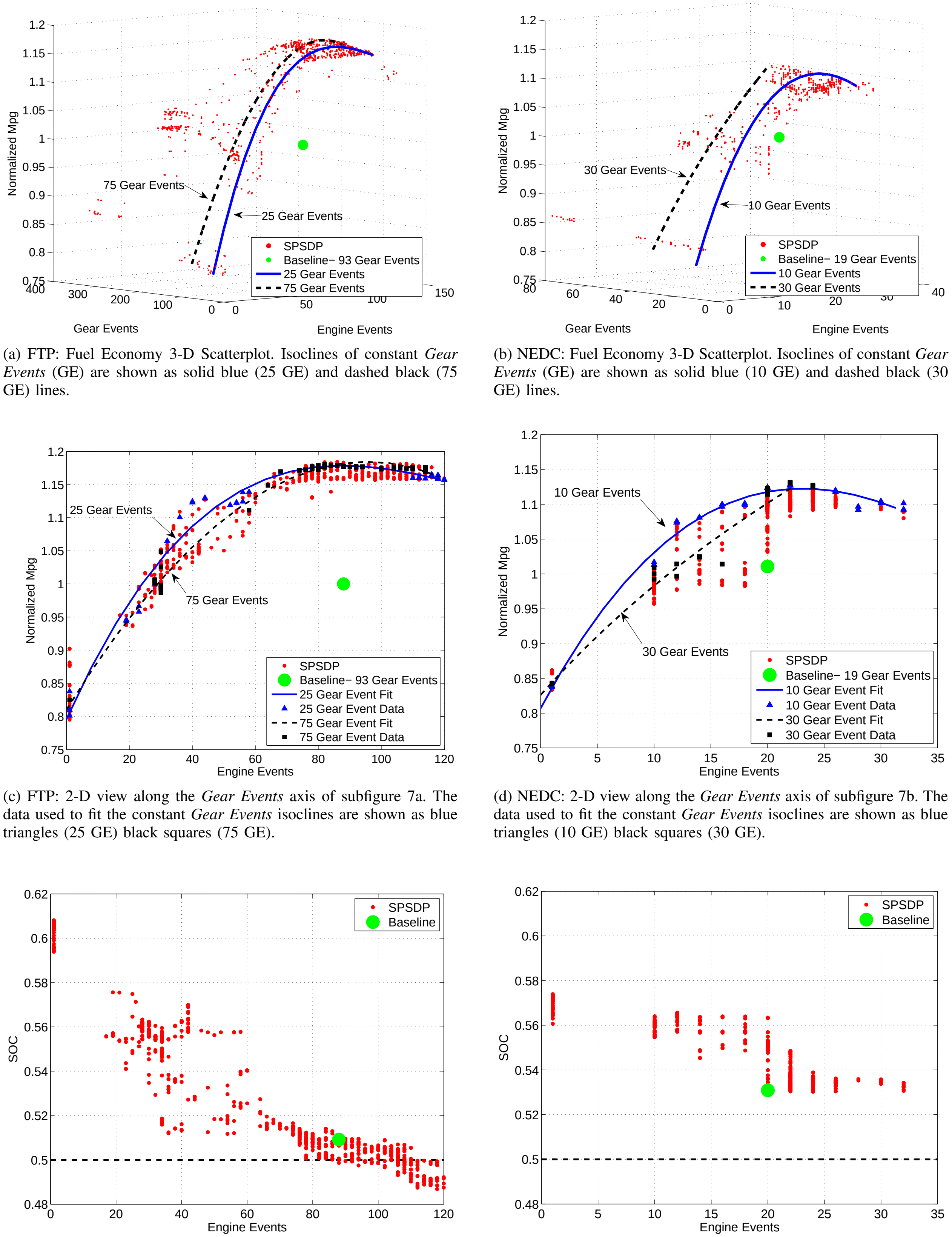
where $\Delta Fuel$ is the adjustment to the fuel used, C_{Batt} is the battery capacity, ΔSOC is the difference between the starting and ending SOC, $BSFC_{min}$ is the best Brake Specific Fuel Consumption for the engine, and η_{max}^{Regen} is the best charging efficiency of the electric system.

Fuel economy numbers in this paper always include the SOC correction, and are normalized so that the baseline controller running the FTP cycle has a fuel economy of one.

VIII. Results: Performance Trends

A. Fuel Economy Results

The main goal of this research is to tradeoff fuel economy and drivability requirements by using a class of optimal



(a) FTP: Fuel Economy 3-D Scatterplot. Isoclines of constant *Gear Events* (GE) are shown as solid blue (25 GE) and dashed black (75 GE) lines.

(b) NEDC: Fuel Economy 3-D Scatterplot. Isoclines of constant *Gear Events* (GE) are shown as solid blue (10 GE) and dashed black (30 GE) lines.

(c) FTP: 2-D view along the *Gear Events* axis of subfigure 7a. The data used to fit the constant *Gear Events* isoclines are shown as blue triangles (25 GE) black squares (75 GE).

(d) NEDC: 2-D view along the *Gear Events* axis of subfigure 7b. The data used to fit the constant *Gear Events* isoclines are shown as blue triangles (10 GE) black squares (30 GE).

(e) FTP: Final SOC for the cycle.

(f) NEDC: Final SOC for the cycle.

Fig. 7: Performance of SPSDP controllers on FTP and NEDC. A separate *family* of controllers is designed for FTP and NEDC using each cycle's statistics. The family is designed by sweeping the parameters α and β in the cost function (7). Figs. 7a and 7b show the data as a 3-D scatterplot of fuel economy vs. drivablity events. SPSDP controllers are shown as red dots. The baseline controller is shown as a large green circle. Figs. 7c and 7d show the view along the *Gear Events* axes of Figs. 7a and 7b respectively. The raw data points, isoclines, and baseline controller are still visible. Figs. 7e and 7f show the final SOC for these controllers. All controllers start with SOC=0.5.

controllers, and validate the result against industrial design methods. The three metrics of interest during vehicle driving are the number of *Gear Events*, *Engine Events*, and the total fuel consumption corrected for SOC. These metrics yield conflicting goals and there is a distinct tradeoff among them. To study this tradeoff, several hundred controllers are designed with varying penalties assigned to each *Gear Event* and *Engine Event*. This creates one *family* of controllers as described in Section VII. The results are shown for FTP and NEDC cycles in Fig. 7.

After simulation, the resulting data show the tradeoff between fuel economy and drivability. The typical result is a 3-D scatterplot of one *family* of controllers as shown in Fig. 7a. Each point represents a single controller driven on the cycle in question. The controllers are all driven on the same test cycle. The total number of *Gear Events*, and *Engine Events* are shown on the horizontal axes, while fuel economy is shown on the vertical axis as normalized MPG. The combination of these points form a surface in 3-D space depicting the tradeoff surface of various operating conditions. This particular figure shows a *family* of controllers designed using FTP statistics running the FTP cycle. Fuel economy for all results in this paper is normalized to the baseline controller performance on FTP, shown as a large green circle. A response surface is fit to the raw data and used to generate isoclines of constant gear, shown as solid (blue) and black (dashed) lines.

Fig. 7c is a 2-D view of Fig. 7a looking along the *Gear Events* axis. Each line in the plot represents a constant number of Gear Events, while the horizontal and vertical axes show the number of *Engine Events* and normalized fuel economy respectively. This particular vehicle is relatively insensitive to the number of Gear Events, so most of the results concentrate on the tradeoff between engine activity and fuel economy. The final SOC for these simulations is shown in Fig. 7e. All simulations start at 0.5 SOC.

Similarly, a *family* of controllers is designed and simulated on the NEDC cycle. Fuel economy results are again shown in 3-D and 2-D in Figs. 7b and 7d, while the final SOC is shown in Fig. 7f. Again, fuel economy is normalized to the baseline controller performance on FTP, so the baseline controller performance is slightly better on NEDC (1.01) than FTP (1.00).

B. Discussion

The frontiers of the 2-D and 3-D point clouds in Fig. 7 clearly demonstrate the tradeoff between fuel economy and drivability. Assuming the same *a priori* information (causal controllers) and a given Markov chain model, no controller's average performance can exceed the SPSDP frontier[2].

The plot of final SOC for the FTP cycle (Fig. 7e) shows a distinct downward trend for large number of engine events. The target final SOC is 0.5, which the controllers come very close to achieving when engine events are unrestricted (low penalties). The final SOC penalty $\phi_{SOC}(x)$ in (6) is only applied if the final SOC is below this target. For final SOCs above the target, the only cost is the fuel spent charging the battery. With smaller numbers of engine events, the controller has less freedom to turn the engine on and charge the battery. In effect, the controllers become more conservative and maintain higher SOCs in general to avoid either additional engine starts or a final SOC that is too low. This characteristic behavior appears in other cycles as shown in Part II [36].

IX. Results: Detailed Performance

A. Results

Several controllers are studied in greater detail on the FTP cycle, which generally yields more interesting behavior than the NEDC. The performance of the baseline controller is compared to 3 SPSDP controllers in Table IV, all running the FTP cycle. The SPSDP controllers are designed using FTP statistics and are selected from those shown in Figs. 7a-7e. SPSDP #1 is the controller with the best corrected fuel economy without regard to drivability. The peak of the fuel economy surface (Fig. 7a) is very close to the baseline controller operating point in terms of drivability. SPSDP #2 has the closest drivability metrics to the baseline controller, and is closely related to SPSDP #1. SPSDP #3 is selected by finding a controller with similar fuel economy to the baseline controller and a minimal number of drivability events.

Time histories of the baseline and SPSDP #1 controllers are presented for the first 500 seconds of the FTP cycle, which is also known as "Bag 1," in Fig. 8. The transmission gear is not shown if the engine is off or the clutch disengaged. The engine torque/speed operating points for these two controllers on the full FTP75 cycle are shown in Fig. 9.

While it is not easy to pinpoint the performance differences between the controllers, some summary metrics are shown for the baseline and SPSDP #1 controllers in Table V.

The Forward Wheel Energy is the integral of all motoring (output) wheel power, the Total Engine Mechanical Output Energy is the total energy delivered at the engine output shaft, Engine Brake Specific Fuel Consumption (BSFC) (g/kwH) is the total fuel consumed divided by the total engine output energy, and Braking Energy is the energy dissipated by the friction brakes.

For the electrical propulsion system, the Electro-Mechanical Charge Energy is the total mechanical energy absorbed by the electric machines, and the Electro-Mechanical Discharge Energy is the forward mechanical energy provided by the electric machines. The losses are the difference between the two, and represent all losses in the electrical system including accessory loads. The Round-Trip Electrical Efficiency is the Discharge Energy scaled by the Charge Energy.

For each metric, the Net Change is shown as the difference between the baseline value and the SPSDP value. The Percent Change is the Net Change scaled by the baseline value. The final column only applies for energy metrics and is the Net Change scaled by the Forward Wheel Energy, a measure of the total energy required for the cycle.

[2] The optimality guarantee for SPSDP is based on an expectation of driver behavior (9). A different controller could do better on a *particular* realization of the statistics (a particular cycle).

TABLE IV: Selected SPSDP controller performance

Controller Description	**Fuel Economy (Corrected)**	**Engine Events**	**Gear Events**	**Final SOC**	**Fuel Economy (Uncorr.)**
Baseline Controller	1.000	88	86	0.509	0.995
SPSDP #1-Best Fuel Economy	1.184	86	96	0.500	1.184
SPSDP #2-Similar Drivetrain Activity	1.183	88	97	0.500	1.183
SPSDP #3-Minimal Drivetrain Activity	1.007	28	0	0.560	0.976

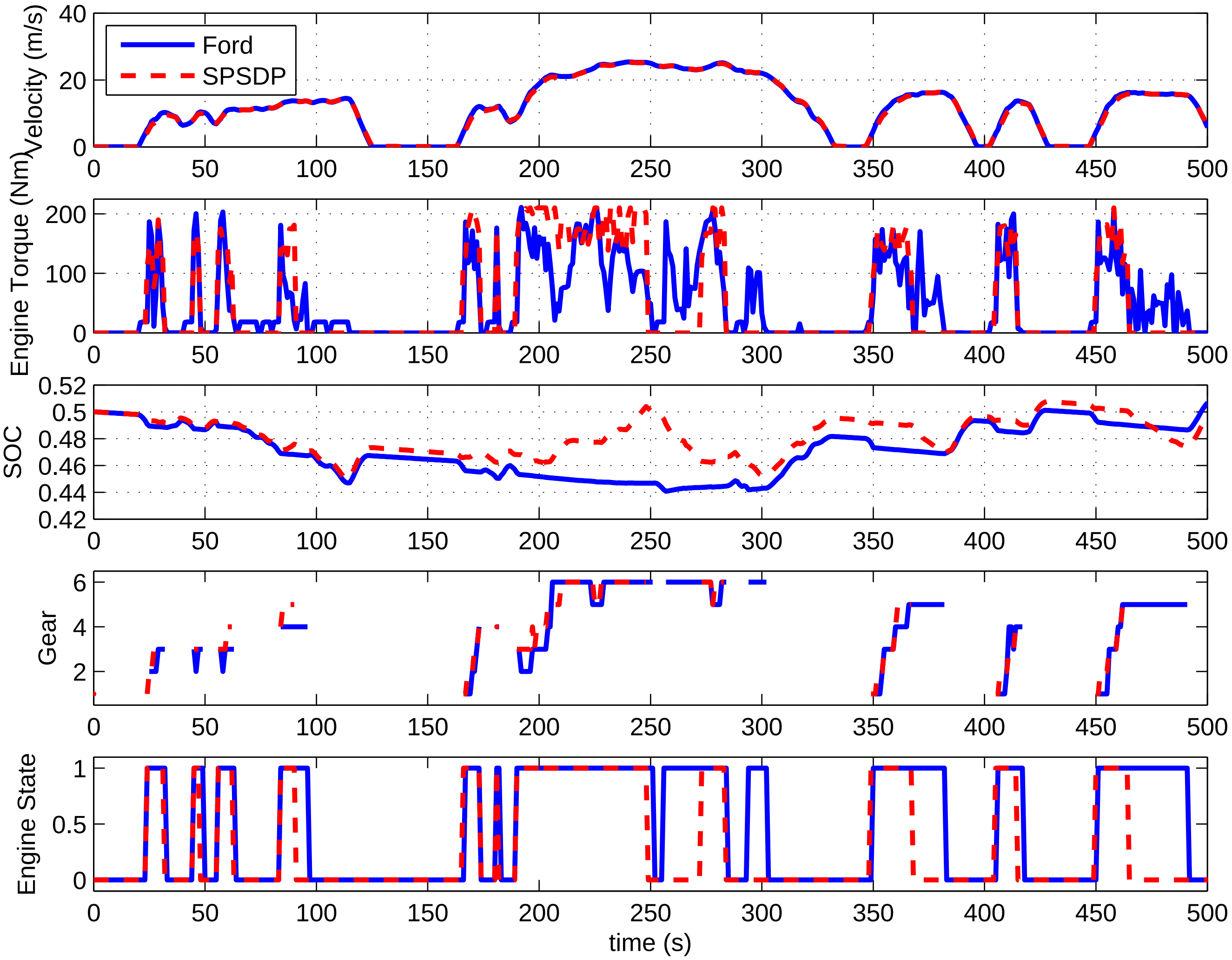


Fig. 8: Time traces of selected simulation parameters. The baseline controller is shown as a solid (blue) line, and one particular SPSDP controller that yields the best overall fuel economy is shown as a dashed (red) line; this is SPSDP #1 in Table IV. The transmission gear is shown only when the engine is on and the clutch is engaged.

TABLE V: Selected SPSDP controller performance

	Baseline Controller	**SPSDP #1**	Net Change	Percent Change	$\frac{NetChange}{TotalForwardWheelEnergy}$
Forward (Motoring) Wheel Energy (kJ)	8736	8295	-441	-5.05%	-5.05%
Total Engine Mechanical Output Energy (kJ)	12298	-894	11404	-7.27%	-10.2%
Braking Energy	673	0	-673	-100%	-5.47
Electro-Mechanical Charge Energy (kJ)	5275	6873	1598	30.3%	18.3%
Electro-Mechanical Discharge Energy (kJ)	2310	3565	1255	54.3%	14.4%
Electro-Mechanical Losses (kJ)	2965	3308	343	11.5%	3.93%
Round-Trip Electrical Efficiency (%)	45.0	51.1	6.1	-	-
Engine BSFC (g/kwH)	266.9	238.6	-28.3	-10.6%	-

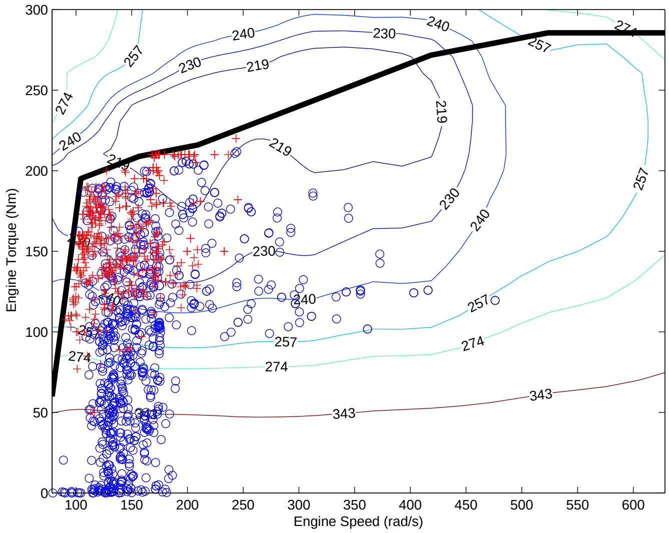


Fig. 9: Engine Torque-Speed operating points on the FTP cycle. The solid black line represents an operational restriction, which both the SPSDP and baseline controllers respect. Baseline controller operating points are shown as circles (blue) and SPSDP controller operating points are shown as plus signs (red). The SPSDP controller is SPSDP #1 in Table IV and also shown in Fig. 8. The isoclines show constant brake specific fuel consumption in g/kWh, an inverse measure of efficiency.

B. Discussion

In Table IV, SPSDP #3 has zero gear events, which arises because of the way gear events are defined. A gear event is counted only when the transmission shifts with the clutch engaged. No penalty is incurred if the clutch disengages, the transmission shifts, and then reengages. With sufficiently high penalties, the algorithm will always disengage the clutch or shut down the engine when a shift is required. The chosen definition works well for most reasonable controllers, but can generate unexpected behavior in the extreme cases. This is an excellent example of a practical consideration: the algorithm will automatically minimize cost based on the *defined* cost function. If one wanted to eliminate this behavior, the designer has several options. Gear Events could be redefined to count a gear event when the clutch disengages, which would generate different behavior. Alternatively, one could add an additional event definition that counts clutch engage/disengage events.

A similar situation occurs when the Engine Events penalty is very high. The clutch cannot slip at low speeds, so the only option is to shut down the engine, or disengage the clutch to enter series mode. Series mode is generally not used, but with sufficiently high penalties on engine activity, the controller will always enter series mode at low speeds to avoid engine shutdown. This yields a cycle with zero engine events, other than the initial start and final stop. The use of Series Mode is discussed further in Part II.

Figs. 8 and 9 and Table V lend some insight into the performance differences between the SPSDP and baseline controller. Table V shows the SPSDP controller is more efficient in its use of the diesel engine. The engine operates largely in a "bang-bang" fashion, either at a high efficiency operating point or completely off, and yields a lower average BSFC as shown in Fig. 9. The high-torque operating points are also visible in Fig. 8. This allows the engine to remain off for slightly longer periods of time (Fig. 8). Note that in general, the engine start/stop activity is almost identical with the exception of a few cases. The electric machines are used more extensively by the SPSDP controller, which allows more efficient ICE utilization and more efficient overall electrical propulsion.

Some of the fuel economy difference between the SPSDP and baseline controllers can be attributed to subtle differences in the driver algorithm that generates commands. The baseline controller uses a PID driver with feedforward that updates quickly at 10ms. The SPSDP controller as formulated here has a discrete 1s update time and the driver torque command is only updated at that rate. Using a proportional controller with feedforward allows the automated driver to accurately drive the cycle at the slow update rate. However, this slower update effectively decreases the variance in the driver torque demand when compared to the baseline controller. This explains the difference in the total positive wheel energy. The update rate explains about 3-5% of the fuel economy difference between the SPSDP and baseline, as confirmed by other simulations with both algorithms using the same update rate. For hardware testing, a slightly different implementation is used which allows a faster update based on driver demand and will be described in a future publication.

X. Conclusions

The energy management controller for a hybrid vehicle is a major factor in the vehicle's overall performance. This paper studies controllers generated using Shortest Path Stochastic Dynamic Programming (SPSDP) and evaluates their performance and robustness on government test cycles using a highly accurate simulation model. The SPSDP-based controllers use a statistical description of expected driving behavior to minimize a cost function that is a weighted sum of consumed fuel and drivability penalties, such as shift events and engine on-off events. By varying the weights, a control designer can systematically trade off fuel economy and drivability. These tradeoffs are optimal for given driving statistics, yielding the Pareto tradeoff curve of fuel economy versus engine on-off activity. The performance of the SPSDP-based controllers was compared against an industrial controller provided by Ford Motor Company.

The SPSDP-based controllers deliver greater than 15% performance improvement over the industrial controller on two government test cycles. Part II [36] of this paper extends these results to a real world setting and addresses more practical issues: robustness to drive cycle variation, performance on large numbers of cycles, sensitivity to design parameters, and practical implementation issues.

References

[1] D. Opila, D. Aswani, R. McGee, J. Cook, and J. Grizzle, "Incorporating drivability metrics into optimal energy management strategies for hybrid vehicles," in *Proceedings of 2008 IEEE Conference on Decision and Control*, 2008.

[2] D. Opila, X. Wang, R. McGee, J. Cook, and J. Grizzle, “Performance comparison of hybrid vehicle energy management controllers on real-world drive cycle data,” in *Proceedings of the American Control Conference*, 2009.

[3] ——, “Fundamental structural limitations of an industrial energy management controller architecture for hybrid vehicles,” in *ASME Dynamic Systems and Control Conference*, 2009.

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

[5] S. Barsali, M. Ceraolo, and A. Possenti, “Techniques to control the electricity generation in a series hybrid electrical vehicle,” *IEEE Trans. Energy Convers.*, vol. 17, no. 2, pp. 260–266, 2002.

[6] S. Barsali, C. Miulli, and A. Possenti, “A control strategy to minimize fuel consumption of series hybrid electric vehicles,” *IEEE Trans. Energy Convers.*, vol. 19, no. 1, pp. 187–195, 2004.

[7] J.-S. Won, R. Langari, and M. Ehsani, “An energy management and charge sustaining strategy for a parallel hybrid vehicle with cvt,” *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 2, pp. 313–320, 2005.

[8] M. Gokasan, S. Bogosyan, and D. J. Goering, “Sliding mode based powertrain control for efficiency improvement in series hybrid-electric vehicles,” *IEEE Trans. Power Electron.*, vol. 21, no. 3, pp. 779–790, 2006.

[9] D. Prokhorov, “Training recurrent neurocontrollers for real-time applications,” *IEEE Trans. Neural Netw.*, vol. 18, no. 4, pp. 1003–1015, July 2007.

[10] C. Dextreit, F. Assadian, I. Kolmanovsky, J. Mahtani, and K. Burnham, “Hybrid vehicle energy management using game theory,” in *Proceedings, SAE World Conference April 2008*, 2008.

[11] R. Langari and J.-S. Won, “Integrated drive cycle analysis for fuzzy logic based energy management in hybrid vehicles,” in *Proc. 12th IEEE International Conference on Fuzzy Systems FUZZ ’03*, vol. 1, 2003, pp. 290–295 vol.1.

[12] L. Perez, G. Bossio, D. Moitre, and G. Garcia, “Supervisory control of an hev using an inventory control approach,” *Latin American Applied Research*, vol. 36, no. 2, pp. 93–100, 2006.

[13] L. Perez and E. Pilotta, “Optimal power split in a hybrid electric vehicle using direct transcription of an optimal control problem,” *Mathematics and Computers in Simulation*, vol. 79, no. 6, pp. 1959–1970, 2009.

[14] S. Kermani, S. Delprat, T. M. Guerra, and R. Trigui, “Predictive control for hev energy management: experimental results,” in *Proc. IEEE Vehicle Power and Propulsion Conference VPPC ’09*, 2009, pp. 364–369.

[15] Q. Gong, Y. Li, and Z.-R. Peng, “Trip-based optimal power management of plug-in hybrid electric vehicles,” *IEEE Trans. Veh. Technol.*, vol. 57, no. 6, pp. 3393–3401, 2008.

[16] L. Johannesson, M. Asbogard, and B. Egardt, “Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming,” *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 1, pp. 71–83, 2007.

[17] G. Paganelli, S. Delprat, T. Guerra, J. Rimaux, and J. Santin, “Equivalent consumption minimization strategy for parallel hybrid powertrains,” in *Proc. IEEE 55th Vehicular Technology Conference VTC Spring 2002*, S. Delprat, Ed., vol. 4, 2002, pp. 2076–2081 vol.4.

[18] A. Sciarretta, M. Back, and L. Guzzella, “Optimal control of parallel hybrid electric vehicles,” *IEEE Trans. Control Syst. Technol.*, vol. 12, no. 3, pp. 352–363, 2004.

[19] C. Musardo, G. Rizzoni, and B. Staccia, “A-ECMS: An adaptive algorithm for hybrid electric vehicle energy management,” in *Proceedings of the European Control Conference Decision and Control CDC-ECC.*, 2005.

[20] C. Musardo, B. Staccia, S. Midlam-Mohler, Y. Guezennec, and G. Rizzoni, “Supervisory control for no/sub x/ reduction of an hev with a mixed-mode hcci/cidi engine,” in *Proc. American Control Conference the 2005*, B. Staccia, Ed., 2005, pp. 3877–3881 vol. 6.

[21] S. Delprat, T. M. Guerra, and J. Rimaux, “Optimal control of a parallel powertrain: from global optimization to real time control strategy,” in *Proc. IEEE 55th Vehicular Technology Conference VTC Spring 2002*, vol. 4, 2002, pp. 2082–2088 vol.4.

[22] S. Delprat, J. Lauber, T. M. Guerra, and J. Rimaux, “Control of a parallel hybrid powertrain: optimal control,” *IEEE Trans. Veh. Technol.*, vol. 53, no. 3, pp. 872–881, 2004.

[23] 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.

[24] B. Wu, C.-C. Lin, Z. Filipi, H. Peng, and D. Assanis, “Optimal power management for a hydraulic hybrid delivery truck,” *Vehicle System Dynamics*, vol. 42, no. 1-2, pp. 23–40, 2004.

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

[26] 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.

[27] S. Delprat, T. M. Guerra, and J. Rimaux, “Control strategies for hybrid vehicles: optimal control,” in *Proc. VTC 2002-Fall Vehicular Technology Conference 2002 IEEE 56th*, vol. 3, 2002, pp. 1681–1685 vol.3.

[28] ——, “Control strategies for hybrid vehicles: synthesis and evaluation,” in *Proc. VTC 2003-Fall Vehicular Technology Conference 2003 IEEE 58th*, vol. 5, 2003, pp. 3246–3250 Vol.5.

[29] G. Paganelli, T. M. Guerra, S. Delprat, J.-J. Santin, M. Delhom, and E. Combes, “Simulation and assessment of power control strategies for a parallel hybrid car,” *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 214, pp. 705–717, 2000.

[30] S. Kermani, S. Delprat, R. Trigui, and T. M. Guerra, “Predictive energy management of hybrid vehicle,” in *Proc. IEEE Vehicle Power and Propulsion Conference VPPC ’08*, 2008, pp. 1–6.

[31] 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.

[32] G. Paganelli, G. Ercole, A. Brahma, Y. Guezennec, and G. Rizzoni, “General supervisory control policy for the energy optimization of charge-sustaining hybrid electric vehicles,” *JSAE Review*, vol. 22, no. 4, pp. 511–518, Oct. 2001.

[33] A. Boyali, M. Demirci, T. Acarman, L. Guvenc, O. Tur, H. Ucarol, B. Kiray, and E. Ozatay, “Modeling and control of a four wheel drive parallel hybrid electric vehicle,” in *Proc. IEEE International Conference on Control Applications*, Oct. 2006, pp. 155–162.

[34] L. Graham, M. Christenson, and D. Karman, “Light duty hybrid vehicles - influence of driving cycle and operating temperature on fuel economy and ghg emissions,” in *Proceedings of the IEEE EIC Climate Change Technology*, M. Christenson, Ed., 2006, pp. 1–6.

[35] 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.

[36] D. Opila, X. Wang, R. McGee, R. Gillespie, J. Cook, and J. Grizzle, “Incorporating drivability metrics into optimal energy management strategies for hybrid vehicles Part 2: Validation and real-world robustness,” *Submitted to IEEE Transactions on Control Systems Technology*, 2010.

[37] C. Belton, P. Bennett, P. Burchill, D. Copp, N. Darnton, K. Butts, J. Che, B. Hieb, M. Jennings, and T. Mortimer, “A vehicle model architecture for vehicle system control design,” in *Proceedings of SAE 2003 World Congress & Exhibition.*, 2003.

[38] P. Pisu, K. Koprubasi, and G. Rizzoni, “Energy management and drivability control problems for hybrid electric vehicles,” in *Proceedings of the European Control Conference Decision and Control CDC-ECC.*, 2005.

[39] D. Bertsekas, *Dynamic Programming and Stochastic Control*. Academic Press, 1976.

[40] J. Liu and H. Peng, “Modeling and control of a power-split hybrid vehicle,” *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 6, pp. 1242–1251, 2008.

[41] L. Johannesson, M. Asbogard, and B. Egardt, “Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming,” in *Proc. IEEE Intelligent Transportation Systems*, 2005, pp. 366–371.

Appendix

A. Proof of Minimization Decomposition

Equation (13) may be written as

$$V^*(x) = \min_{\hat{u}\in\hat{U}(x)} \min_{\bar{u}\in\bar{U}(x,\hat{u})} E_w[c(x,\hat{u},\bar{u}) + V^*(f(x,\hat{u},w))], \tag{22}$$

and by the linearity of expectation

$$V^*(x) = \min_{\hat{u}\in\hat{U}(x)} \min_{\bar{u}\in\bar{U}(x,\hat{u})} (E_w[c(x,\hat{u},\bar{u})] + E_w[V^*(f(x,\hat{u},w))]). \tag{23}$$

The cost function $c(x, \hat{u}, \bar{u})$ is independent of w and the expectation may be dropped. The expectation of the value function is independent of $\bar{u}$ yielding

$$V^*(x) = \min_{\hat{u}\in\hat{U}(x)} (\min_{\bar{u}\in\bar{U}(x,\hat{u})} c(x, \hat{u}, \bar{u}) + E_w[V^*(f(x, \hat{u}, w))]). \tag{24}$$

Using the definition (14), (24) becomes (13). ■

B. Related Comments on Minimization Decomposition

To implement the controller developed using Minimization Decomposition, $\bar{u}$ must still be determined. It may be precomputed and stored when calculating (14),

$$\bar{u}^*(x, \hat{u}) = \underset{\bar{u}\in\bar{U}(x,\hat{u})}{\operatorname{argmin}}\, c(x, \hat{u}, \bar{u}), \tag{25}$$

and

$$\hat{c}(x, \hat{u}) = c(x, \hat{u}, \bar{u}^*(x, \hat{u})) = \min_{\bar{u}\in\bar{U}(x,\hat{u})} c(x, \hat{u}, \bar{u}). \tag{26}$$

This process reduces the space of control actions by $\bar{U}$. The computation scales linearly with the number of possible control actions, and can be significantly reduced depending on the problem structure and the size of $\bar{u}$.

Minimization Decomposition may also be used when solving for non-stationary value functions by appropriately replacing $V(x)$ with a time-dependent $V_k(x)$, for either deterministic or stochastic cases [16].

Remark: (Functional Form to use Minimization Decomposition) Suppose a system has dynamics $f(x, \hat{u}, \bar{u}, w)$ that are independent of some control component $\bar{u}$ and can be reformulated into a function $\hat{f}$, such that

$$\hat{f}(x, \hat{u}, w) = f(x, \hat{u}, \bar{u}, w) \tag{27}$$

with probability 1 (w.p. 1). Then the Bellman equation satisfies (13) and the Minimization Decomposition may be used. ■

While the property (27) seems quite restrictive, it occurs surprisingly often in the energy management problem. It is likely to occur if the number of control inputs M exceeds the dimension of the state space N, leaving a null control direction as used in [38].

Remark: (State Decomposition) In this energy management problem (as in most formulations) the dynamics may clearly be broken down into two parts

$$f(x, u, w) = \begin{bmatrix} f_u(x, u) \\ f_w(x, w) \end{bmatrix} \tag{28}$$

where the deterministic states are the known vehicle dynamics and the stochastic driver dynamics are independent of the control input. ■

This allows the control inputs to be studied without the effect of w, simplifying the condition (27). Define P as the dimension of f_u, the state space that depends on u in (28). If $M > P$, (27) likely holds.

The main point of this section is this: if the number of control inputs exceeds the number of states, the required computation can often be drastically reduced. Even with discrete states (ie. gear number) the same techniques may often be used.

C. Power-Split Example

Consider for example the "Power-Split" architecture of the Toyota Prius and Ford Escape, with a cost function that penalizes fuel use and SOC deviations from nominal to attain charge sustenance. If one assumes that the dynamics of engine speed changes are negligible at the timescales for energy management, the only vehicle state is SOC, as velocity and acceleration are assigned by the driver (stochastically when using SPSDP). Assuming the vehicle matches driver demand torque, the system is defined by two inputs. By using specific definitions of the system variables, the optimization reduces to two one degree of freedom problems. A common method is to treat the two control inputs as engine speed and engine power. In this example we choose engine speed ω_{ICE} and electrical power P_{elec}, a slightly different definition. This allows a major decoupling of the system dynamics. The evolution of the the SOC state is now only dependent on $P_{elec} = \hat{u}$ and completely independent of $\omega_{ICE} = \bar{u}$. The engine speed that results in minimum fuel use for a given P_{elec} can be calculated off-line because it is independent of SOC. This results in engine fuel consumption as a 1-D function of power $\hat{c}(x, \hat{u}) = \hat{c}(x, P_{elec})$, rather than the standard 2-D functions of power and speed $c(x, \hat{u}, \bar{u}) = \hat{c}(x, P_{elec}, \omega_{ICE})$.