

# An Energy Management Controller to Optimally Trade Off Fuel Economy and Drivability for Hybrid Vehicles

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 non-causal solutions are easy to determine if the drive cycle is known a priori. It is very challenging to design causal controllers that yield good fuel economy for a range of possible driver behavior. Additional challenges come in the form of constraints on powertrain activity, such as shifting and starting the engine, which are commonly called “drivability” metrics and can adversely affect fuel economy. In this paper, drivability restrictions are included in a Shortest Path Stochastic Dynamic Programming (SP-SDP) formulation of the real-time energy management problem for a prototype vehicle, where the drive cycle is modeled as a stationary, finite-state Markov chain. When the SP-SDP controllers are evaluated with a high-fidelity vehicle simulator over standard government drive cycles, and compared to a baseline industrial controller, they are shown to improve fuel economy more than 11% for equivalent levels of drivability. In addition, the explicit tradeoff between fuel economy and drivability is quantified for the SP-SDP controllers.

## 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].

The energy management problem has been studied extensively in academic circles. Various 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], as well as 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 [23], [24], [25], [26]. The majority of existing work focuses on controllers that seek to minimize fuel consumption, while ignoring other attributes that affect the smoothness and responsiveness of vehicle acceleration, which are commonly referred to as “drivability.” In practice, fuel-optimal controllers can lead to excessive gear shifting and engine starting/stopping [27], [28], [29], [30], and hence poor drivability. Previous research has addressed drivability in a suboptimal manner by incorporating penalties on engine starts in an ECMS formulation [18]. The reference [31] addressed engine starts indirectly by including a hysteresis term “to avoid a too frequent switch on - switch off of the [internal combustion] engine, which would cause an additional energy use and wearout.”

In this paper, drivability restrictions are directly incorporated in a causal, optimal controller design method for the energy management system of a Hybrid Electric Vehicle (HEV). The focus is on drivability with respect to engine start-stop and gear shifts; a host of other drivability issues, such as low-frequency longitudinal vibration and other attributes which are typically mitigated by hardware design or low-level control actions, are not considered. The main optimization tool is Shortest Path Stochastic Dynamic Programming (SP-SDP), which, as explained in [32], [33], [34], [26], is a specific formulation of Stochastic Dynamic Programming (SDP) that allows infinite horizon optimization problems to be addressed without the use of discounting (a discount factor in the cost function assures convergence by weighting future costs exponentially less than current costs). In the energy management problem, the power requested by the driver as a function of time is modeled as a stationary, finite-state Markov chain [23]. The state space of the Markov chain is constructed to include a terminal state corresponding to key-off [26]. The terminal state is designed to be absorbing (that is, it is reached in finite time with probability one, and there is zero probability of transitioning out of it). If zero cost is incurred in the absorbing terminal state, then the expected value of the cost function is finite, even without the discount factor often used in hybrid vehicle applications of SDP [33], [34].

The controllers generated through SP-SDP are causal state feedbacks and hence are directly implementable in a real-time control architecture. The controllers are provably optimal if the driving behavior matches the assumed Markov chain model. In this paper, the Markov chains representing driver behavior are modeled on standard government test cycles, as in [23], [26]. It is also possible to build the Markov chains on the basis of real-world driving data, as reported in [35].

In addition to generating a class of optimal controllers, the SP-SDP method allows direct study of the tradeoffs between different performance goals, here, drivability and

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].

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

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 the computed performance. Drivability is emphasized in this paper, but one could also study the fuel economy tradeoff with other attributes such as emissions, battery wear, or engine noise characteristics.

One place where SP-SDP 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. The SP-SDP 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.

The work reported here is a collaborative effort between the University of Michigan and Ford Motor Company. The vehicle studied is a modified Volvo S-80 prototype and 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 described in [2] and is termed hereafter the “baseline” controller. In addition, a high-fidelity vehicle simulation model calibrated for the prototype vehicle was provided; this is the same simulation model used by Ford to develop HEV control algorithms and to evaluate fuel economy and drivability for production vehicles [36].

The remainder of the paper is organized as follows. Section II presents the vehicle architecture and two dynamic models; one is a simplified vehicle model for controller design and the second is the high-fidelity model mentioned above. The drivability metrics used in the optimization problem are presented in Section III. The particular form of infinite-horizon stochastic optimal control used here, SP-SDP, is presented in Section IV, and a key result that greatly enhances off-line computational speed is presented in Section V. The procedure for sweeping out the Pareto tradeoff surface is presented in Section VI; this involves computing a large family of controllers based on the simplified control-oriented model and evaluating each controller’s performance with the high-fidelity model, which will more closely approximate the actual performance on the prototype vehicle. The main results of the work are presented in Sections VII and VIII. Concluding remarks are given in Section IX. The Appendix provides additional information on enhancing off-line computational speed for SP-SDP and points out a relation between SP-SDP and ECMS.

## II. VEHICLE

### A. Vehicle Architecture

The vehicle studied in this paper is a prototype series-parallel electric hybrid and is shown schematically in Fig. 1. 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

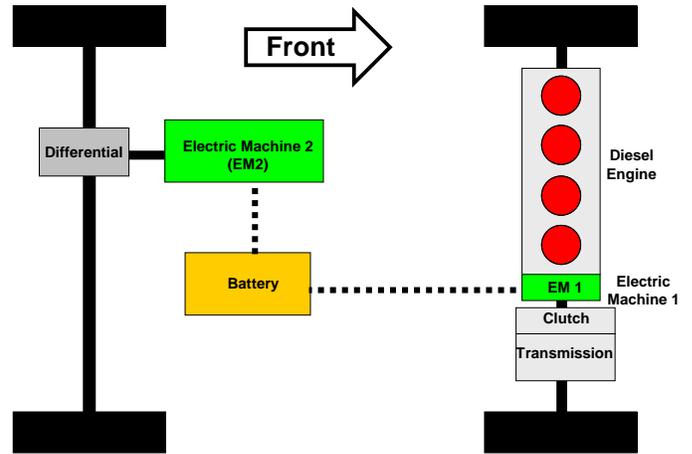


Fig. 1: Vehicle Configuration

proportional to vehicle speed. Energy is stored in a 6 Ah (1.9 kWh) battery pack with a recommended State of Charge (SOC) range of 0.35-0.65. 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	1.9 kWh (6 Ah)
Battery Power Limit	34 kW
Battery SOC Range	0.35-0.65
Vehicle Mass	1895 kg

### B. Vehicle Models

The work presented in this paper uses two separate 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 for controller design via dynamic programming, and is called the “control-oriented” model. The second model, provided by Ford Motor Company, is a complex, MATLAB/Simulink-based model with a large number of parameters and states [36]. Each subsystem in the vehicle is represented by an appropriate block with its own dynamics and low-level controllers. This model, which accurately represents the transient response of the engine, transmission and driveline, is referred to as the “high-fidelity” model in the remainder of the paper.

This combination of models allows the controller to be designed on the basis of a simple model for computational tractability, while providing performance assessment on the basis of a model that much more closely reflects the complicated dynamics of the prototype vehicle.

### 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,  $\omega_{ICE}$ , and torque,  $T_{ICE}$ . The fuel consumption  $\dot{m}_f$  is derived from a lookup table based on dynamometer testing,

$$\dot{m}_f = F(\omega_{ICE}, T_{ICE}).$$

The dual clutch transmission has discrete gears and no torque converter. The transmission is modeled with a constant mechanical efficiency  $\eta_{trans}$ . Gear shifts are allowed every time step and transmission dynamics are assumed negligible. While the physical configuration of the transmission allows arbitrary shifting, the low-level transmission controller enforces sequential up/down shifting and the model respects this assumption. This technique is advantageous in hardware because shifts execute by smoothly transitioning between the two clutches and continually transmitting torque. One transmission shaft holds the even gears and the other the odd gears. An arbitrary gear may be selected when the clutch is disengaged. 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 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  $\omega_{ICE}$ ,

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

In parallel mode, when providing power to the wheels, the torques  $T_{ICE}$  and  $T_{EM1}$  are proportional to wheel torque based on the current gear ratio  $R_g$  and  $\eta_{trans}$ ; during regenerative braking, for example, when absorbing power from the wheels<sup>1</sup>, the torques are proportional based on  $R_g$  and  $1/\eta_{trans}$ . Similarly, the rear electric machine torque  $T_{EM2}$  is proportional to the machine's gear ratio  $R_{EM2}$  and rear differential efficiency  $\eta_{diff}$  when providing power, and is

proportional to  $R_{EM2}$  and  $1/\eta_{diff}$  when absorbing power. The total wheel torque  $T_{wheel}$  from both axles is thus the sum of the ICE and  $EM1$  torques to the wheel and the rear electric machine  $EM2$  torque to the wheel, namely

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

where

$$\tau_{trans}(T) = \begin{cases} \eta_{trans} T & \text{if } T \geq 0 \\ \frac{T}{\eta_{trans}} & \text{otherwise,} \end{cases}$$

and similarly for  $\tau_{diff}$ .

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.

The battery system is similarly reduced to table lookup form. The electrical dynamics due to the motor, battery, and power electronics are assumed sufficiently fast to be ignored. The energy losses and efficiencies in these components can be grouped together such that the change in battery SOC is a function  $\bar{\kappa}$  of electric machine speeds  $\omega_{EM1}$  and  $\omega_{EM2}$ , torques  $T_{EM1}$  and  $T_{EM2}$ , and battery SOC at the current time step,

$$SOC_{k+1} = \bar{\kappa}(SOC_k, \omega_{EM1_k}, \omega_{EM2_k}, T_{EM1_k}, T_{EM2_k}). \quad (2)$$

In this simplest configuration, assuming a known vehicle speed, the only state variable required for the vehicle model is the battery SOC. Changes in battery performance due to temperature, age, and wear are ignored. Additional states are required to represent the stochastic drive cycle and to track drivability metrics.

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  $T_{demand}$ ,

$$T_{wheel} = T_{demand}.$$

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 *Gear*. In series mode, the electric machine command becomes *EM1 Speed*.

Optimization using the control-oriented model assumes a perfect driver during the design process; specifically, the desired road power is calculated as the exact power required to drive the cycle at that time. A PID controller based on velocity feedback is used to represent a causal driver during simulation of the high-fidelity model. Now, given vehicle speed, demanded road power and this choice of control inputs, the dynamics become an explicit function  $\kappa_k$  of the state *Battery SOC* and the three control choices shown in Fig. 2,

<sup>1</sup>Note that the model accounts for power loss in both directions.

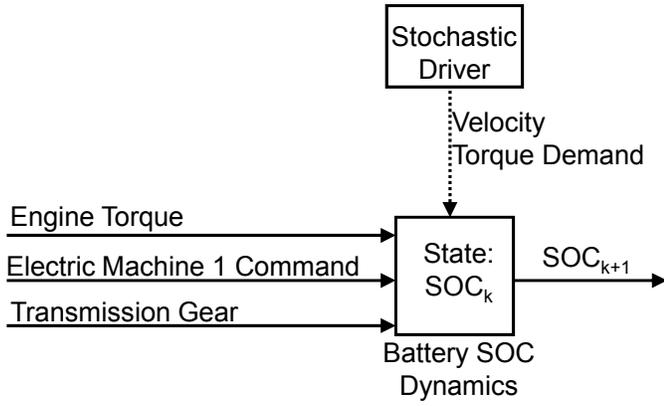


Fig. 2: Battery SOC Dynamics model. Parallel mode inputs are Engine Torque, Electric Machine-1 Torque, and Gear. The vehicle velocity and required torque are provided by the stochastic driver model. In Series Mode, the Electric Machine-1 command is speed rather than torque.

$$SOC_{k+1} = \kappa_k(SOC_k, T_{ICE_k}, T_{EM1_k}, Gear_k). \quad (3)$$

In series mode,  $T_{EM1}$  is replaced with  $\omega_{EM1}$ . 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) Regenerative braking is used as much as possible up to the actuator limits; friction brakes provide any remaining torque.
- 2) The clutch in the transmission allows the diesel engine to be decoupled from the wheels, permitting all-electric or series operation.
- 3) There is no ability to slip the clutch for vehicle launch.
- 4) There are no traction control restrictions on the amount of torque that can be applied to the wheels.

#### D. High-Fidelity Vehicle Simulation Model

The high-fidelity 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 high-fidelity model with the control algorithm developed here, the SP-SDP controller is implemented in Simulink by interfacing appropriate feedback and command signals: Battery State of Charge, Vehicle Speed, Engine State, Gear Command, etc. The high-fidelity model can then be driven by the SP-SDP controller along a given drive cycle using a causal driver model.

#### E. Baseline Industrial Controller

The baseline prototype energy management controller studied here is quite complex. Its key features are contained in three modules, as depicted in Fig. 3. Driver power demand is determined from pedal position. One module determines the battery power flow and adds it to the driver demand to determine the *Total Power*. A second module determines the

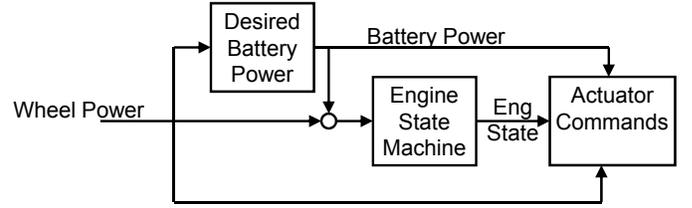


Fig. 3: High Level Baseline Controller Architecture.

engine state based on the *Total Power* using a state machine with hysteresis. A third rule-based module then determines individual actuator commands (e.g., power from the engine and the two electric machines) based on the *Total Power* and the desired engine state. The gear is selected independently by the transmission controller.

The primary tuning parameters are five scalar functions, two in the Battery Power module and three functions of vehicle speed in the Engine State Machine module. One advantage of the baseline architecture is that engine behavior and battery charge maintenance features are largely confined to their respective blocks, simplifying the tuning process considerably.

### III. DRIVABILITY CONSTRAINTS

#### A. Overview

Drivability is a term that covers many aspects of vehicle performance including acceleration, engine noise, braking, automated shifting activity, and shift quality [37], [38]. Meeting a customer’s expectations of drivability often involves a tradeoff with fuel economy. As an example, optimal fuel economy for gasoline engines typically dictates upshifting at the lowest possible speed, but this leaves the driver with little acceleration ability after the upshift. Consequently, upshifts are scheduled to occur at a speed that is higher than the value that is best for fuel economy.

Industry experts were consulted to assist in quantifying aspects of drivability that are strongly coupled to the energy-management controller. It was recommended that attention be focused on the frequency and timing of gear-shift events and engine-start/stop events. The mean time between events and the number of short-duration events were recommended as metrics, where a short-duration event means that dwell time in a particular state is less than a specified acceptable value. For the transmission, a particularly annoying short-duration event is “hunting,” that is, rapid shifting between the same two gears. Figure 4 shows 7 possible metrics based on mean and short-duration drivability metrics for the engine and transmission. For later use, these metrics are referred to as the “complex” drivability metrics.

#### B. Simplified Drivability Metrics

In order to incorporate these complex drivability metrics into the model, and then into the optimization problem, states would have to be added to keep track of the duration between shifts and between engine starts and stops, as well as the mean number of these events over a given time interval. While this is theoretically possible, the well-known “curse

	Engine Behavior	Transmission Behavior
Mean Dwell Times	Mean Engine On Time Mean Engine Off Time	Mean Time in Gear
Short Duration Events	Engine On Dwell Time < X seconds Engine Off Dwell Time < X seconds	Gear Dwell Time < X seconds Gear Hunting Events

Simplified Metrics	Engine Events	Gear Events
--------------------	---------------	-------------

Fig. 4: Drivability Metric Reduction. The seven complex engine and transmission metrics are divided into two categories, mean dwell times and short-duration dwell times. These metrics are then reduced to the two simplified metrics.

of dimensionality” would render the associated stochastic optimization problem computationally intractable. Even if the optimization problem could be solved, the designer would be faced with the difficult job of assigning relative weights to each of the metrics when performing a tradeoff analysis.

We chose therefore to simplify these complex metrics into two measures of drivability that can be more easily used. The first drivability metric is termed gear events, and is defined to be the total number of shift events on a given trip. The second drivability metric is termed engine events, and is defined to be 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 with the clutch engaged is counted as a gear event, whereas engaging or disengaging the clutch is not counted as a gear event, regardless of the gear before or after the event.

Figure 5 shows that the complex and simple metrics are strongly correlated; specifically, the figure shows that reducing the total number of engine on-off events over a drive cycle reduces the occurrence of events where the engine is on for less than 3, 5, 10 or 30 seconds. The data are shown along with a straight-line least-squares fit. The other complex metrics listed in Fig. 4 show similar correlations, being approximately monotone functions of the simple metrics. The data in Fig. 5 were obtained by simulating the SP-SDP controllers of the ensuing sections on the high-fidelity model. The results are presented here in order to motivate the use of the simplified metrics in the rest of the paper.

### C. Inclusion of Drivability Constraints in the Cost Function

The first step in the design of a controller with acceptable drivability properties is to pose a cost function that permits a compromise between fuel economy and drivability. This is achieved by the use of penalties. Specifically, the cost function over a particular drive cycle (suppressing the summing index) is

$$J = \sum_0^T \dot{m}_f + \alpha \sum_0^T \mathbf{I}_{GE}(x, u) + \beta \sum_0^T \mathbf{I}_{EE}(x, u),$$

where  $\mathbf{I}(x, u)$  are indicator functions and thus equal one when a state and control combination produces a gear event (GE) or engine event (EE) as defined in Section III-B, and are zero otherwise.  $T$  is the total trip time, from key-on to key-off. The drivability behavior is not incorporated as a direct constraint,

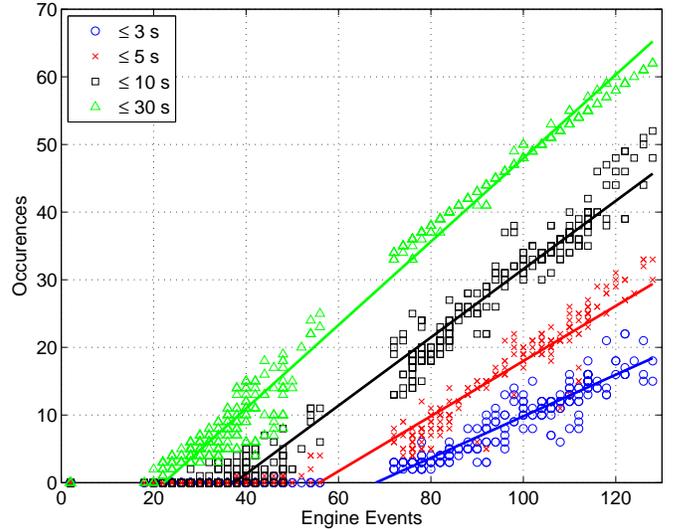


Fig. 5: Engine on durations less than some number of seconds compared to the simplified engine events metric. Data are shown for cutoffs of 3, 5, 10, and 30 seconds.

so the search for the weighting factors  $\alpha$  and  $\beta$  involves some trial and error because the mapping from penalty to outcome is not known a priori. Note that setting  $\alpha$  and  $\beta$  to zero corresponds to solving for optimal fuel economy without regard to drivability.

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 at the terminal state,  $x_T$  (that is, at the end of the trip at key-off), and is represented as a function  $\phi_{SOC}(x_T)$ . The performance index for a particular drive cycle is then

$$J = \sum_0^T \dot{m}_f + \alpha \sum_0^T \mathbf{I}_{GE}(x, u) + \beta \sum_0^T \mathbf{I}_{EE}(x, u) + \phi_{SOC}(x_T). \quad (4)$$

## IV. SHORTEST PATH STOCHASTIC DYNAMIC PROGRAMMING

### A. Problem Formulation

As the cycle is not known exactly in advance, this optimization is conducted in the stochastic sense by minimizing the expected sum of a running cost function  $c(x_k, u_k, w_k)$ , where  $x_k$  is the state,  $u_k$  is a particular control choice in the set of allowable controls  $U(x_k)$ , and  $w_k$  is a random variable arising from the unknown drive cycle. The expectation over the random process  $w$  is denoted  $E_w$ . The optimization problem is

$$\min E_w \sum_{k=0}^{\infty} c(x_k, u_k, w_k) \quad (5)$$

subject to the system dynamics,

$$x_{k+1} = f(x_k, u_k, w_k) \quad (6)$$

with  $u_k \in U(x_k)$ , where

$$U(x_k) = \{u_k \mid g_1(x_k, u_k) \leq 0, g_2(x_k, u_k) = 0\}.$$

Actuator limits, torque delivery requirements, and other system requirements are incorporated in the constraints  $g_1$  and  $g_2$ , which are enforced at each time step, in contrast to drivability goals, which involve performance over the whole cycle.

To implement the optimization goal (5), the running cost function is prescribed to represent (4),

$$c(x, u, w) = \dot{m}_f(x, u) + \alpha \mathbf{I}_{GE}(x, u) + \beta \mathbf{I}_{EE}(x, u) + \phi_{SOC}(x, w). \quad (7)$$

The SOC-based cost  $\phi_{SOC}(x, w)$  applies only at the end of the trip, when the key-off event occurs. As explained in Section IV-D, the transition to key-off is captured by the stochastic drive-cycle model in the random process  $w$ . The cost  $\phi_{SOC}(x, w)$  at the key-off event replaces the terminal-time cost  $\phi_{SOC}(x_T)$  in (4).

### B. Bellman Equation

To determine the optimal control strategy for this vehicle, the SP-SDP algorithm is used [25], [26], [33], [34]. This method directly generates a causal, time-invariant, state-feedback controller. Characteristics of future driving behavior are specified via a finite-state Markov chain rather than exact future knowledge. Given the system model (6), the optimal cost  $V^*(x)$  over an infinite horizon is a function of the state  $x$  and satisfies

$$V^*(x) = \min_{u \in U(x)} E_w[c(x, u, w) + V^*(f(x, u, w))], \quad (8)$$

where  $c(x, u, w)$  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  $c(x, u, w)$  and the expected future cost  $V(f(x, u, w))$ . The control  $u$  is selected based on the expectation over  $w$ , 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 any control that achieves the minimum cost  $V^*(x)$

$$u^*(x) = \operatorname{argmin}_{u \in U(x)} E_w[c(x, u, w) + V^*(f(x, u, w))]. \quad (9)$$

**Remark:** At each time step  $k$ , random variable  $w_k$  in (8) and (9) may be conditioned on the state and control input,

$$P(w_k | x_k, u_k). \quad (10)$$

### C. Stochastic Drive-Cycle Model

The drive cycle is modeled as a Markov chain. The drive cycle is assigned two states: current velocity  $v_k$  and current acceleration  $a_k$ , which are included in the full system state  $x_k$ . The random variable  $w_k$  in (8) is the acceleration at the next time step. Specifying the drive cycle is equivalent to assigning a probability distribution to  $w$ , that is, specifying

$$P(a_{k+1} | v_k, a_k) \quad (11)$$

for pairs  $v_k, a_k$ . Following [25], the transition probabilities (11) are estimated from known drive cycles that represent typical behavior, referred to as “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 the state as in (10); see [25], [26] for more detail. Specific design cycles include the standard cycles used to establish “window sticker” fuel economy such as the Federal Test Procedure (FTP). As mentioned previously, design cycles might also include measured driving behavior over “real world” vehicle use.

Bringing this all together, the full system state vector  $x$  contains five components: one state for the vehicle (*Battery SOC*), two states for the stochastic driver ( $v_k, a_k$ ), and two states to study drivability (*Gear* and *Engine State*). This formulation is termed the “SP-SDP-Drivability” controller. A summary of system states is shown in Table II.

TABLE II: Vehicle and Driver Model States, SP-SDP Drivability Controller

State	Units
Battery Charge (SOC)	unitless
Vehicle Speed	m/s
Vehicle Acceleration	m/s <sup>2</sup>
Gear	Integer 0-6
Engine State	On or Off

The inputs to the model are engine torque, gear number, and the powers or torques of the two electric machines. Looking ahead, Section V shows how an off-line optimization step can be used to replace the two electric machines by a single input representing total electric machine power, thereby reducing the inputs for the optimization problem to engine torque, gear number, and total electric machine power. The power balance to meet driver demand given in (1) then allows the elimination of one more input. The final control input  $u$  that will be used in the optimization problem consists therefore of *Engine Torque* and *Gear*.

**Remarks:** (a) The form of the Bellman equation (8) associated with any dynamic programming problem allows an analytical comparison with ECMS and is discussed in Section D of the Appendix. (b) 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 *Gear*. Two more states are required to study the stochastic version, and the simplified drivability model used here requires two additional states.

### D. 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,

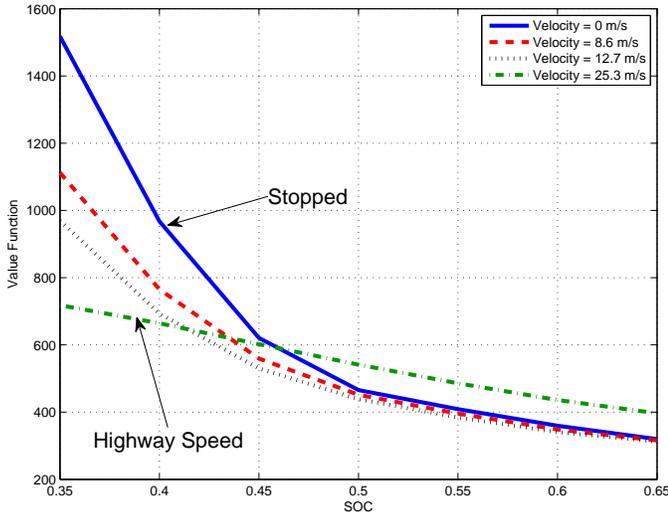


Fig. 6: Value Function  $V(x)$  for several velocities and fixed acceleration. The quadratic penalty on SOC strongly affects the value function at low speeds when the driver is more likely to turn the key off and end the trip.

shut down the vehicle, and removed the key. Once the key-off event occurs, there are no further 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. To include this in the SP-SDP 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. Fig. 6 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. Moreover, there will be a deceleration phase before reaching zero speed and thus an opportunity to recharge the battery.

## E. Implementable Constraints

Stochastic Dynamic Programming is inherently computationally intensive and can quickly become intractable. The computational 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 two additional states required to implement this cost function are the current gear and the engine state. Thus, 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 required grid points scales with the number of time steps used to specify the metric. For the 1 second update time studied here, penalizing engine events of 5 seconds duration or less (rather than the simple on/off used here) would require 5 grid points for the time history, increasing the size of the state-space by a corresponding factor of 5 over the on/off case.

## V. COMPUTATION REDUCTION

### A. Theory

#### Proposition: (Minimization Decomposition)

Consider a Bellman equation of the form

$$V^*(x) = \min_{\hat{u} \in \hat{U}(x), \bar{u} \in \bar{U}(x, \hat{u})} E_w[c(x, \hat{u}, \bar{u}, w) + V^*(f(x, \hat{u}, w))], \quad (12)$$

and define

$$\hat{c}(x, \hat{u}) = \min_{\bar{u} \in \bar{U}(x, \hat{u})} E_w[c(x, \hat{u}, \bar{u}, w)]. \quad (13)$$

Then  $V^*(x)$  satisfies (12) if, and only if, it satisfies

$$V^*(x) = \min_{\hat{u} \in \hat{U}(x)} E_w[\hat{c}(x, \hat{u}) + V^*(f(x, \hat{u}, w))]. \blacksquare \quad (14)$$

The proof and more detail are available in the Appendix. This result allows a significant reduction in computational complexity for problems that have the specific structure (12). The reduced Bellman equation (14) may be solved using only the reduced control space  $\hat{U}(x)$ . This structure appears quite often in energy management problems (see Appendix).

The above decomposition has been exploited in previous work without explicit theoretical justification [16], [39]. A typical example is the power-split HEV configuration which uses engine power and speed as inputs without an engine speed state [39]. The fuel-minimizing engine speed ( $\bar{u}$ ) for each engine power ( $\hat{u}$ ) is precomputed and stored as a table (see Appendix).

The following subsection details the physical explanation of the structure (12) 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 SP-SDP solution more complex by forcing the algorithm to

consider an additional dimension in the control space. If the additional control variable is discretized with say  $N = 10$  points, the size of the minimization operation in (8) over pairs  $(x, u)$  increases by a factor of ten. Exploiting the structure represented by (12) and using Minimization Decomposition reduces the computational cost to that of a vehicle with a single electric machine, i.e., a 90% reduction. The addition of the second electric machine is then approximately free in terms of computing an off-line solution to the SP-SDP problem.

Intuitively, Minimization Decomposition lumps the two electric machines into 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, engine torque, and gear to achieve the desired torque with minimal electric power, as shown in Fig. 7. The required minimization is static, and in an off-line setting such as SP-SDP, can be done once and reused. Once the static optimization is performed, the Super Electric Machine appears as a single power source for the SP-SDP optimization. Internally, however, the Super Electric Machine optimizes between the two (or possibly more) electric machines and issues appropriate commands.

A more technical justification follows. The torque balance (1) allows a tradeoff between the two electric machine torques. The system dynamics are only affected by the net change in SOC

$$\hat{u} : \delta SOC \quad (15)$$

and not by the split of the electric machine torques, which can be defined by one command

$$\bar{u} : \bar{T}_{EM2}. \quad (16)$$

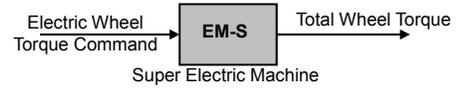
For a given  $\bar{T}_{EM2}$ , velocity, and gear,  $\bar{T}_{EM1}$  is exactly determined by  $\delta SOC$ . Since the power-split optimization is static (i.e., independent of the dynamic states of the model, including SOC), it takes the form (12) and can be computed a priori using (13) without loss 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 depend only on the current values of the EM torque commands, current gear, and velocity, and in particular, do not depend on their past values. For any control command under consideration, the knowledge of current gear, velocity,  $\bar{T}_{EM2}$ ,  $\delta SOC$ , and the required wheel torque uniquely determines all the terms in the torque balance (1). An optimal  $\bar{u}$  can then be selected as in (13). The required engine torque is determined from the torque balance, accounting for the direction-dependent efficiency losses in the transmission.

The physical control inputs to the system are engine torque, gear, EM1 torque and EM2 torque. The constraint to match driver demand torque removes one degree-of-freedom. By replacing the two electric machine commands with a single electric wheel torque command, the SP-SDP algorithm has only 2 control inputs.

## VI. SIMULATION PROCEDURE

SP-SDP-based controllers are compared to a baseline industrial controller. SP-SDP controllers are designed using the control-oriented model and evaluated using the high-fidelity

## Mechanical Analog:



## Internal Function:

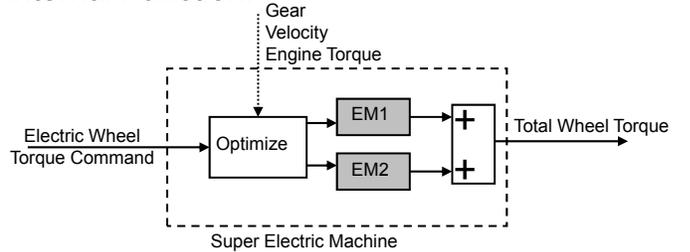


Fig. 7: Schematic diagram of a conceptual ‘‘Super Electric Machine’’ that optimizes the mix between the two electric machines. This allows one degree of freedom of the control optimization to be carried out off-line while maintaining the optimality of the solution.

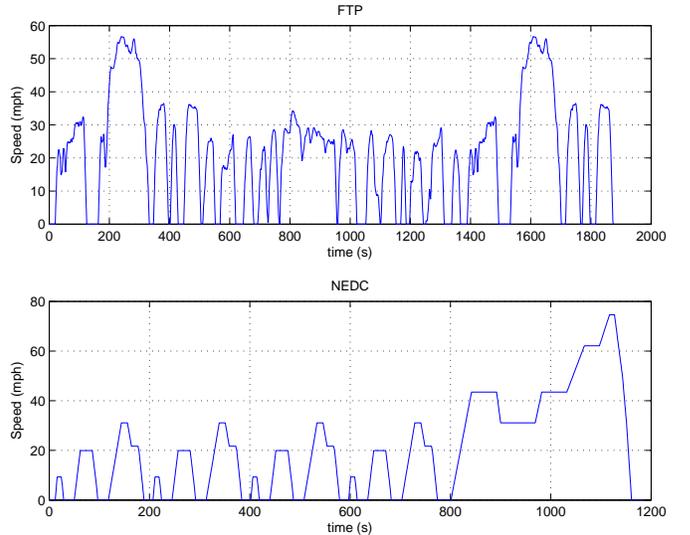


Fig. 8: Federal Test Procedure (FTP) and New European Drive Cycle (NEDC).

vehicle simulation model of Section II-D. This demonstrates some robustness by using two models of the same vehicle, differing in the level of detail in their dynamics. Strictly speaking, the optimality guarantees are no longer valid because the test model is different from the design model. For practical purposes, a strictly optimal model-based controller is unattainable in hardware because a model will always have some mismatch with a real vehicle. Demonstrating excellent performance on the (exact) design model is only marginally useful as it presents no model uncertainty. By designing the controller on a simple model and testing on a (not perfectly matched) complex model, we more closely approximate the process of designing on the basis of a model and testing on hardware.

Both SP-SDP and the baseline controllers are simulated on two government test cycles, the US Federal Test Procedure and the New European Drive Cycle (NEDC), which are shown in Fig. 8. Procedurally, this is conducted as follows:

- 1) A family of SP-SDP 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  $\alpha$  and  $\beta$  in (7).
- 2) Each controller in the family is simulated on the high-fidelity model using a causal driver, thus accounting for all the dynamics and real vehicle characteristics neglected in the optimization.
- 3) The fuel economy and drivability metrics are recorded. Fuel economy is computed in units of MPG (Miles driven Per Gallon of fuel consumed), and hence larger numbers mean better fuel economy.

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 economy characteristics because of the varying drivability penalties.

Because the simulations on the high-fidelity model use a causal driver model, the final SOC is not guaranteed to exactly match the starting SOC. This could yield false fuel economy results, so all fuel economy estimates 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

$$\Delta m_f = C_{Batt} \Delta SOC \frac{BSFC_{min}}{\eta_{max}^{Regen}} \quad (17)$$

where  $\Delta m_f$  is the adjustment to the fuel used,  $C_{Batt}$  is the battery capacity,  $\Delta SOC$  is the difference between the starting and ending SOC,  $BSFC_{min}$  is the best Brake Specific Fuel Consumption for the engine, and  $\eta_{max}^{Regen}$  is the best charging efficiency of the electric system. This correction is a reasonable approximation but not exact; the exact correction depends on the controller and the particular cycle. For the FTP cycle, the mean fuel economy correction for the SOC deviations presented in Fig. 9e is 1.6%, with a 1.3% standard deviation. Hence, using this simple correction does not change the conclusions of the presented results in any substantial way.

Fuel economy numbers in this paper always include the SOC correction. The fuel economy of the baseline controller running the FTP cycle is used as the nominal value for normalization. Therefore the normalized fuel economy of the baseline controller on FTP is one.

## VII. RESULTS: PERFORMANCE TRENDS

### A. Fuel Economy Results

The three metrics of interest in this paper are the numbers of gear and engine events, and the total fuel consumption corrected for SOC. The family of controllers generated as

described in Section VI yields the results shown in Fig. 9 for the FTP cycle and the NEDC.

Figs. 9a and 9b show 3-D scatter plots of fuel economy versus gear and engine events for the two cycles. Each point represents a single controller driven on the cycle in question. The total numbers of gear events and engine events are shown on the horizontal axes, while fuel economy is shown on the vertical axis as normalized MPG<sup>2</sup>. The combination of these points form a surface in 3-D space depicting the tradeoff surface of various operating conditions. Figure 9a shows a family of controllers designed using FTP statistics running the FTP cycle. Fuel economy data presented in this paper are normalized to the fuel economy of the baseline controller on FTP, shown as a large solid square. Hence, a fuel economy greater than one means more miles would be traveled using the same fuel as consumed by the baseline controller, or equivalently, less fuel would be consumed for the same distance traveled. A polynomial surface is fit to the raw data and used to generate isoclines of constant number of gear events, shown as solid and dashed lines.

Fig. 9c is a 2-D view of Fig. 9a 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. 9e. All simulations start at 0.5 SOC.

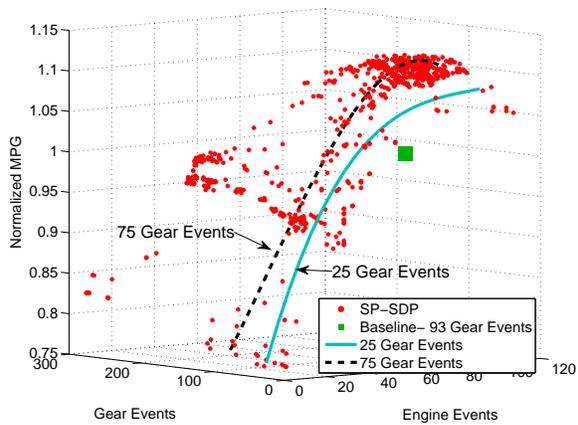
Similarly, a family of controllers is designed and simulated on the NEDC. Fuel economy results are again shown in 3-D and 2-D in Figs. 9b and 9d, while the final SOC is shown in Fig. 9f. Again, fuel economy is normalized to the baseline controller performance on FTP, so the baseline controller is slightly less fuel efficient on NEDC (0.99) than FTP (1.00).

### B. Discussion

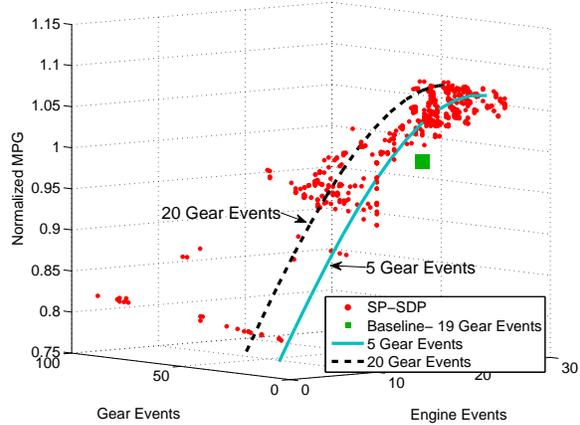
The frontiers of the 2-D and 3-D point clouds in Fig. 9 clearly demonstrate the tradeoff between fuel economy and drivability. The plot of final SOC for the FTP cycle (Fig. 9e) shows a distinct downward trend for large numbers of engine events. The target final SOC is 0.5, which the controllers come very close to achieving when engine events are unrestricted (low penalties). The final SOC penalty  $\phi_{SOC}(x, w)$  in (7) used in the control design process is only applied if the final SOC is below this target. For final SOC 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 SOC to avoid either additional engine starts or a final SOC that is too low.

An interesting phenomenon occurs when the engine events penalty is very high. In this case, to avoid engine shut-down, the only option is to disengage the clutch and enter series

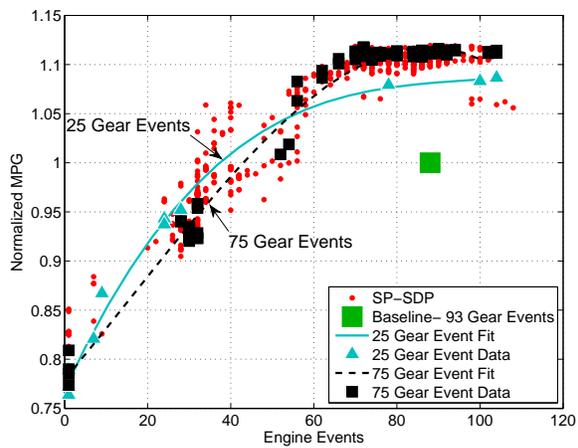
<sup>2</sup>Recall that more miles per gallon means better fuel economy, while the inverse would hold if units of liters per 100 kilometers were used.



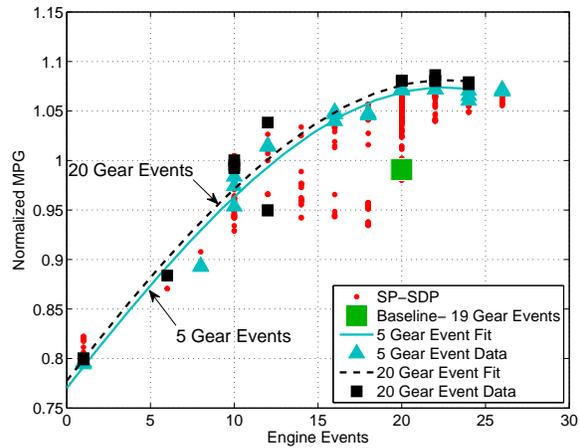
(a) FTP: Fuel Economy 3-D Scatterplot. Isoclines of constant gear events (GE) are shown as solid blue (25 GE) and dashed black (75 GE) lines.



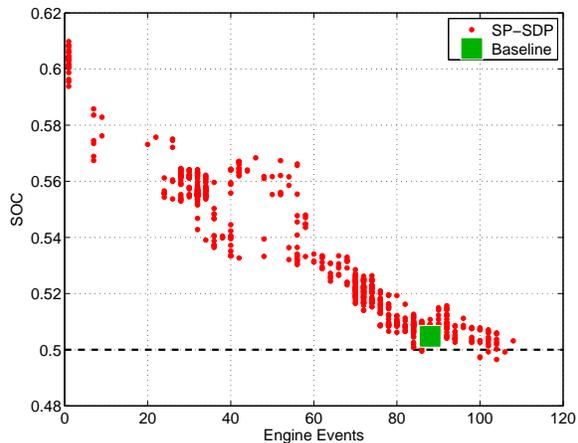
(b) NEDC: Fuel Economy 3-D Scatterplot. Isoclines of constant gear events (GE) are shown as solid blue (5 GE) and dashed black (20 GE) lines.



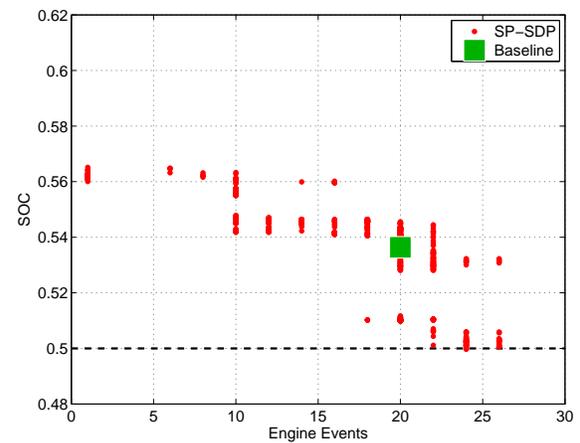
(c) FTP: 2-D view along the gear events axis of Fig. 9a. The data used to fit the constant gear events isoclines are shown as blue triangles (25 GE) black squares (75 GE).



(d) NEDC: 2-D view along the gear events axis of Fig. 9b. The data used to fit the constant gear event isoclines are shown as blue triangles (5 GE) black squares (20 GE).



(e) FTP: Final SOC for the cycle.



(f) NEDC: Final SOC for the cycle.

Fig. 9: Performance of SP-SDP 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  $\alpha$  and  $\beta$  in the cost function (7). Figs. 9a and 9b show the data as a 3-D scatterplot of fuel economy vs. drivability events. Fuel economy is presented in miles per gallon, so higher is better, and normalized so that the baseline controller has a fuel economy of 1.0 on FTP. SP-SDP controllers are shown as small dots (red). The baseline controller is shown as a large solid square (green). Figs. 9c and 9d show the view along the gear events axes of Figs. 9a and 9b respectively. The raw data points, isoclines, and baseline controller are still visible. Figs. 9e and 9f show the final SOC for these controllers. All controllers start with SOC=0.5.

mode. With this artifice, it is possible to have a cycle with no other engine events than the initial start and final stop.

The results show some unexpected trends. Figs. 9c and 9d show a slight decrease in fuel economy for large numbers of engine events. In some cases on FTP, decreasing the number of gear events actually increases fuel economy (9c). There are two issues here: The optimization is with respect to expected value and not a single sample path as in the plot, and the plot depicts controller performance on the high-fidelity model and not the control-oriented design model. To determine which of these two explanations is the correct one, simulations of controller performance for the FTP cycle were conducted on the simplified vehicle model (i.e., the control design model) in order to eliminate the issue of model mismatch. These simulations show the same trends discussed above, implying that model mismatch is not causing these phenomena. Large numbers of cycles were then simulated [35] to check the performance in the expected sense rather than for a single cycle. This second set of simulations show fuel economy is monotonic in both gear and engine events as one would expect.

The SP-SDP results show significant (11%) performance improvements over the baseline controller for the metrics considered here. Production controllers incorporate many additional attributes, such as noise, harshness, durability, safety, accessory loading, diagnostics, etc. These attributes may decrease the performance margin when fully incorporated. One obvious example is emissions, which was not considered in either the baseline or SP-SDP controllers. For previous work on NOx emissions, see [40], [23], [41].

## VIII. 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 SP-SDP controllers in Table III, all running the FTP cycle. The SP-SDP controllers are designed using FTP statistics and are selected from those shown in Figs. 9a, 9c, and 9e. SP-SDP #1 is the controller with the best corrected fuel economy without regard to drivability. The peak of the fuel economy surface (Fig. 9a) is very close to the baseline controller operating point in terms of drivability. SP-SDP #2 has the closest drivability metrics to the baseline controller, and is closely related to SP-SDP #1. SP-SDP #3 is selected by finding a controller with similar fuel economy to the baseline controller and about half the number of drivability events. Essentially, we are presenting two possible design choices: improved fuel economy with similar drivability (SP-SDP#2), or similar fuel economy with reduced drivetrain activity (SP-SDP#3). The designer may also select some compromise between the two.

Time histories of the baseline and SP-SDP #1 controllers are presented for the first 500 seconds of the FTP cycle in Fig. 10. The engine torque/speed operating points for these two controllers on the full FTP cycle are shown in Fig. 11.

Summary metrics are shown for the baseline and SP-SDP controllers in Table IV. The Forward Wheel Energy is the

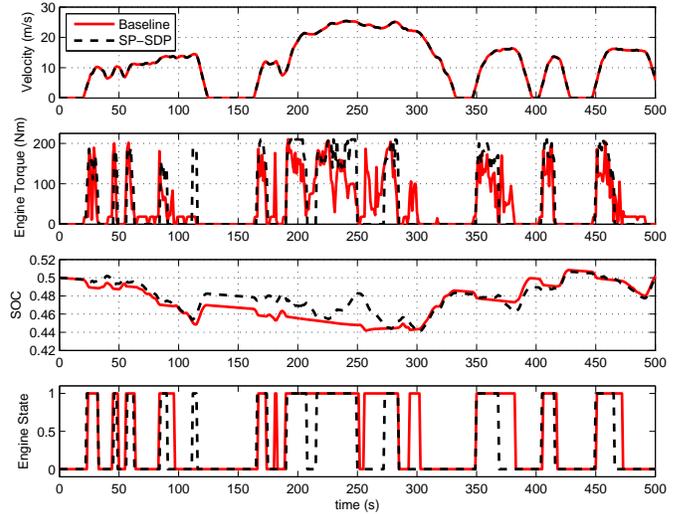


Fig. 10: Time traces of selected simulation parameters. The baseline controller is shown as a solid (red) line, and one particular SP-SDP controller that yields the best overall fuel economy is shown as a dashed (black) line; this is SP-SDP #1 in Table III.

integral of all motoring (output) wheel power, the Engine 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 Friction 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 Electro-Mechanical Losses are the difference between the two minus the change in battery energy (due to final SOC) and represent all losses in the electrical system including accessory loads. The Round-Trip Electrical Efficiency is the Discharge Energy plus any net change in SOC divided by the Charge Energy.

### B. Discussion

Figs. 10 and 11 and Table IV lend some insight into the performance differences between the SP-SDP and baseline controller. Table IV shows the SP-SDP controller is more efficient in its use of the diesel engine. The engine primarily operates near a high efficiency point or completely off, and yields a lower average BSFC as shown in Fig. 11. The high-torque operating points are also visible in Fig. 10. The electric machines are used more extensively by the SP-SDP controller, which allows more efficient ICE utilization and more efficient overall electrical propulsion. Friction braking is also minimized by the SP-SDP controller.

## IX. CONCLUSIONS

An energy management controller for a prototype parallel-series hybrid electric vehicle has been developed using Short-

TABLE III: Selected SP-SDP controller performance

Controller Description	Fuel Economy (Corrected)	Engine Events	Gear Events	Final SOC	Fuel Economy (Uncorr.)
Baseline Controller	1.000	88	93	0.505	0.997
SP-SDP #1-Best Fuel Economy	1.119	88	106	0.504	1.117
SP-SDP #2-Similar Drivetrain Activity	1.114	88	93	0.506	1.110
SP-SDP #3-Similar Fuel Economy	1.010	34	36	0.561	0.977

TABLE IV: Selected SP-SDP controller performance

	Baseline Controller	SP-SDP #1	SP-SDP #2	SP-SDP #3
Forward (Motoring) Wheel Energy (kJ)	8731	8580	8578	8579
Engine Output Energy (kJ)	11939	11917	11990	12856
Friction Braking Energy (kJ)	254.7	5.5	5.5	5.9
Electro-Mechanical Charge Energy (kJ)	5525	7059	7196	9616
Electro-Mechanical Discharge Energy (kJ)	2514	3831	3926	5302
Electro-Mechanical Losses (kJ)	2978	3199	3233	4042
Round-Trip Electrical Efficiency (%)	46.1	54.7	55.1	58.0
Engine BSFC (g/kWh)	264.8	237.0	236.9	251.6

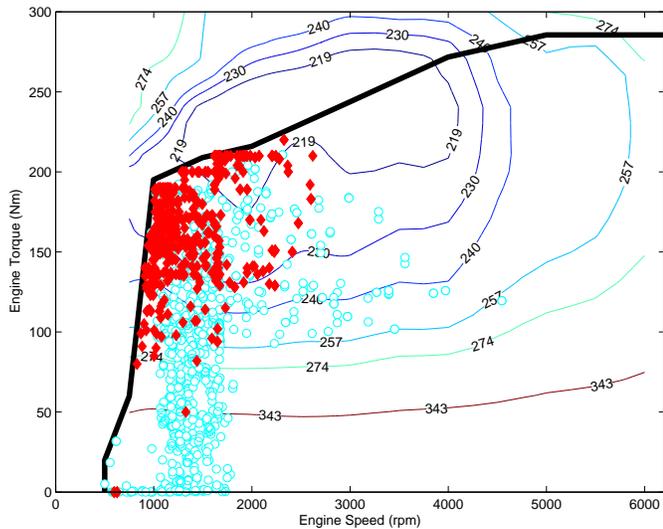


Fig. 11: Engine Torque-Speed operating points on the FTP cycle. The solid black line represents an operational restriction, which both the SP-SDP and baseline controllers respect. Baseline controller operating points are shown as circles (blue) and SP-SDP controller operating points are shown as diamonds (red). The SP-SDP controller is SP-SDP #1 in Table III and also shown in Fig. 10. The isoclines show constant brake specific fuel consumption in g/kWh.

est Path Stochastic Dynamic Programming to optimally perform the inherent tradeoff between fuel efficiency and drivability. The SP-SDP-based controllers minimize the expected value of a cost function, using a statistical description of expected driving behavior. Here, the cost was a weighted sum of consumed fuel and drivability penalties based on shift events and engine on-off events. By varying the weights, the Pareto tradeoff surface of fuel economy versus drivability for the SP-SDP-based controllers was evaluated on a high-fidelity vehicle simulation model.

The performance of the SP-SDP-based controllers was compared against an industrial baseline controller. For the

same level of drivability, the SP-SDP-based controllers were 11% more fuel efficient than the baseline controller on the FTP cycle and the New European Drive Cycle. The SP-SDP-based controllers were designed for the driving statistics of each of the two cycles.

In general, dynamic programming is well known to suffer from the “curse of dimensionality,” referring to the exponential explosion of problem size with the number of state and control variables. The system model addressed here had three power sources, namely, an internal combustion engine and two electric machines. A two-step off-line optimization strategy was presented that preserved optimality while presenting the SP-SDP algorithm with an equivalent system model that contained a single “super” electric machine. Ultimately, this allowed the SD-SDP algorithm to be run on a desktop PC. A similar two-step optimization strategy is applicable to other vehicle configurations that have multiple actuator degrees of freedom.

While the excellent fuel economy of the SP-SDP controllers is very interesting, we feel a more important observation is that the SP-SDP design method produces causal controllers that respect constraints and perform well on (and off [35]) standard test cycles. SP-SDP-based controllers can be directly implemented in a realistic control environment with little manual tuning, as demonstrated on an industrial vehicle model which includes detailed subsystem models, dynamics, delays, and limits.

## APPENDIX

### A. Proof of Minimization Decomposition

Equation (12) may be written as

$$V^*(x) = \min_{\hat{u} \in \hat{U}(x)} \min_{\bar{u} \in \bar{U}(x, \hat{u})} E_w[c(x, \hat{u}, \bar{u}, w) + V^*(f(x, \hat{u}, w))], \quad (18)$$

and by the linearity of expectation

$$\begin{aligned} & V^*(x) \\ &= \min_{\hat{u} \in \hat{U}(x)} \min_{\bar{u} \in \bar{U}(x, \hat{u})} (E_w[c(x, \hat{u}, \bar{u}, w)] + E_w[V^*(f(x, \hat{u}, w))]). \end{aligned} \quad (19)$$

The expectation of the value function is independent of  $\bar{u}$  yielding

$$V^*(x) = \min_{\hat{u} \in \hat{U}(x)} \left( \min_{\bar{u} \in \bar{U}(x, \hat{u})} E_w[c(x, \hat{u}, \bar{u}, w)] + E_w[V^*(f(x, \hat{u}, w))] \right). \quad (20)$$

Using the definition (13), (20) becomes (12). ■

### 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 (13),

$$\bar{u}^*(x, \hat{u}) = \operatorname{argmin}_{\bar{u} \in \bar{U}(x, \hat{u})} c(x, \hat{u}, \bar{u}), \quad (21)$$

and

$$\hat{c}(x, \hat{u}) = c(x, \hat{u}, \bar{u}^*(x, \hat{u})) = \min_{\bar{u} \in \bar{U}(x, \hat{u})} c(x, \hat{u}, \bar{u}). \quad (22)$$

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

$$\hat{f}(x, \hat{u}, w) = f(x, \hat{u}, \bar{u}, w) \quad (23)$$

with probability 1 (w.p. 1). Then the Bellman equation satisfies (12) and the Minimization Decomposition may be used.

While the property (23) 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

$$f(x, u, w) = \begin{bmatrix} f_u(x, u) \\ f_w(x, w) \end{bmatrix} \quad (24)$$

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 verification of condition (23). Whenever the number of actuators exceeds the dimension of  $f_u$ , (23) is likely to hold.

The main point 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 (e.g. 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 SP-SDP). 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. Suppose instead we choose engine speed  $\omega_{ICE}$  and electrical power  $P_{elec}$ , a slightly different definition. This allows a major decoupling of the system dynamics. The evolution of SOC 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  $c(x, \hat{u}, \bar{u}) = \hat{c}(x, P_{elec}, \omega_{ICE})$ .

### D. Comparison of SP-SDP to ECMS for Fuel-only Optimization

One of the most well known optimization methods for energy management in HEVs is the ‘‘Equivalent Consumption Minimization Strategy’’ (ECMS) [19], [42]. This method optimizes for fuel economy only, which is equivalent to taking the running cost in (8) as fuel flow rate, that is,  $c = \dot{m}_f$ . ECMS is popular among academics because it requires little computation and seems easy to implement. At each time step, the controller minimizes a function that trades off battery usage vs. fuel,

$$u_k^*(x) = \operatorname{argmin}_{u \in U} [\dot{m}_f(x, u) + \lambda_k \Delta SOC(x, u)]. \quad (25)$$

The design parameter is the weighting factor  $\lambda_k$ , which represents the relative value of battery charge in terms of fuel. In actual practice, a real difficulty arises because, unless  $\lambda_k$  is carefully chosen, the vehicle will not be charge sustaining. The required values for  $\lambda_k$  are highly cycle dependent and typically require on-line estimation.

It is now shown for the fuel only case that (8) and (9) of the SP-SDP algorithm yield a form very similar to (25) for the computation of the optimal control, with the added benefit that the SP-SDP algorithm automatically adjusts the weighting function. First, note that the state can be taken as  $x = [SOC, \bar{x}]'$ , where  $\bar{x}$  consists of vehicle velocity and acceleration.  $\bar{x}$  is independent of the control input  $u$  because vehicle acceleration is defined by the stochastic driver. The model (6) can thus be expressed in the form

$$\begin{bmatrix} SOC_{k+1} \\ \bar{x}_{k+1} \end{bmatrix} = \begin{bmatrix} SOC_k + \Delta SOC(SOC_k, \bar{x}_k, u_k) \\ \bar{f}(\bar{x}_k, w_k) \end{bmatrix}. \quad (26)$$

Next, let  $V^*$  be the optimal cost to go function in the Bellman equation (8), and define

$$Q(\sigma, \bar{x}) = E_w[V^*(\left[ \begin{array}{c} \sigma \\ \bar{f}(\bar{x}, w) \end{array} \right])]$$

for an arbitrary SOC  $\sigma$ . Substituting the SOC dynamics (26) for  $\sigma$ , the Bellman equation (8) becomes

$$V^*(SOC, \bar{x}) = \min_{u \in U} [\dot{m}_f(SOC, \bar{x}, u) + Q(SOC + \Delta SOC(SOC, \bar{x}, u), \bar{x})]. \quad (27)$$

The running cost  $c = \dot{m}_f$  in (8) is not a function of the random variable  $w$  and can be removed from the expectation. From this, the expression for the optimal control becomes

$$u^*(SOC, \bar{x}) = \operatorname{argmin}_{u \in U} [\dot{m}_f(SOC, \bar{x}, u) + Q(SOC + \Delta SOC(SOC, \bar{x}, u), \bar{x})]. \quad (28)$$

Doing a first-order Taylor expansion of  $Q$  then yields

$$u^*(SOC, \bar{x}) \approx \operatorname{argmin}_{u \in U} [\dot{m}_f(SOC, \bar{x}, u) + Q(SOC, \bar{x}) + \frac{\partial Q(SOC, \bar{x})}{\partial SOC} \Delta SOC(SOC, \bar{x}, u)]. \quad (29)$$

Recognizing that  $Q(SOC, \bar{x})$ , being independent of  $u$ , does not affect the minimization, and substituting  $x = [SOC, \bar{x}]'$  into (29), then yields

$$u^*(x) \approx \operatorname{argmin}_{u \in U} \left[ \dot{m}_f(x, u) + \frac{\partial Q(x)}{\partial SOC} \Delta SOC(x, u) \right]. \quad (30)$$

It follows that  $\frac{\partial Q(x)}{\partial SOC}$  is equivalent to the weighting factor  $\lambda$  in (25). The SP-SDP algorithm has the same structure as the ECMS method, but the weighting factor is a function of the state variables, and is automatically updated on-line. There is a variant of ECMS method called Adaptive ECMS (A-ECMS) in which the weighting factor is also allowed to change over time based on the current driving conditions [19]. A-ECMS is even more similar to the SP-SDP algorithm in that both methods have a weighting factor that is updated on-line as a function of the state.

This relationship is illustrated by again studying the value function  $V(x)$  as a function of SOC for fixed acceleration and velocity shown in Fig. 6. The local slope of  $V(x)$  in the figure is closely related to the weighting factor in (30), which, once again, is analogous to  $\lambda$  in (25). 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. It may appear linearly and explicitly as in ECMS, or nonlinearly and implicitly as in SP-SDP. All known information is incorporated in the weighting factor: current state, plant dynamics, and expected future driver demands. Once this weighting factor is determined, the control problem is a simple static optimization.

A basic difference of the algorithms lies in how they estimate the value of battery charge in terms of fuel: ECMS uses a value assigned by the designer; A-ECMS estimates a value based on battery charge and recent history; Deterministic

Dynamic Programming uses exact future knowledge; and SP-SDP uses estimates of cycle statistics. A benefit of dynamic programming methods, such as SP-SDP, is that they can optimally accommodate more complicated objectives, such as the fuel and drivability metrics studied here.

## ACKNOWLEDGMENT

The authors would like to thank the reviewers for their helpful suggestions to clarify the explanation of the Minimization Decomposition.

## 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 *Proc. IEEE Conference on Decision and Control*, 2008, pp. 4382–4389.
- [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 *Proc. American Control Conference*, 2009, pp. 4618–4625.
- [3] —, "Fundamental structural limitations of an industrial energy management controller architecture for hybrid vehicles," in *Proc. ASME Dynamic Systems and Control Conference*, 2009, pp. 213–221.
- [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 *Proc., SAE World Conference*, 2008, Paper no. 2008-01-1317.
- [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*, vol. 1, 2003, pp. 290–295.
- [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*, 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*, vol. 4, 2002, pp. 2076–2081.
- [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 *Proc. IEEE Conference on Decision and Control*, 2005, pp. 1816–1823.
- [20] C. Musardo, B. Staccia, S. Midlam-Mohler, Y. Guezennec, and G. Rizzoni, "Supervisory control for NO<sub>x</sub> reduction of an HEV with a mixed-mode HCCI/CIDI engine," in *Proc. American Control Conference*, vol. 6, 2005, pp. 3877–3881.
- [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*, vol. 4, 2002, pp. 2082–2088.
- [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 *Proc. of the American Control Conference*, 2004, pp. 4710 – 4715.
- [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. Vehicular Technology Conference*, vol. 3, 2002, pp. 1681–1685.
- [28] —, "Control strategies for hybrid vehicles: synthesis and evaluation," in *Proc. Vehicular Technology Conference*, vol. 5, 2003, pp. 3246–3250.
- [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," *Proc. 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*, 2008, pp. 1–6.
- [31] A. Kleimaier and D. Schroder, "An approach for the online optimized control of a hybrid powertrain," in *Proc. 7th Int Advanced Motion Control Workshop*, 2002, pp. 215–220.
- [32] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.
- [33] D. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2005, vol. 1.
- [34] —, *Dynamic Programming and Optimal Control*. Athena Scientific, 2005, vol. 2.
- [35] D. Opila, "Incorporating drivability metrics into optimal energy management strategies for hybrid vehicles," Ph.D. dissertation, University of Michigan, 2010.
- [36] 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 *Proc. SAE World Congress & Exhibition*, 2003, Paper no. 2003-01-0092.
- [37] X. Wei, "Dynamic modeling of a hybrid electric drivetrain for fuel economy, performance and driveability evaluations," *American Society of Mechanical Engineers, Dynamic Systems and Control Division (Publication) DSC*, vol. 72, no. 1, pp. 443–450, 2003.
- [38] P. Pisu, K. Koprubasi, and G. Rizzoni, "Energy management and drivability control problems for hybrid electric vehicles," in *Proc. IEEE Conference on Decision and Control*, 2005, pp. 1824–1830.
- [39] 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.
- [40] E. D. Tate, J. W. Grizzle, and H. Peng, "SP-SDP for fuel consumption and tailpipe emissions minimization in an EVT hybrid," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 3, pp. 673–687, 2010.
- [41] 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.
- [42] 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*, 2001, pp. 5064–5069.



**Daniel F. Opila** (M '08) received BS and MS degrees from the Massachusetts Institute of Technology, Cambridge, Massachusetts, in 2002 and 2003 respectively, and the Ph.D. degree from the University of Michigan, Ann Arbor, Michigan, in 2010.

He is a Senior Research and Development Engineer at Converteam Naval Systems in Pittsburgh, a division of GE Energy Management. Previously he was a Visiting Scholar at Ford Motor Company, a Senior Engineer at Orbital Sciences Corporation, and a Mechanical Engineer at Bose Corporation. He specializes in optimal control of energy systems, including hybrid vehicles, naval power systems, power converters, and renewables.

Dr. Opila is a member of ASME.



**Xiaoyong Wang** received the B.S. and M.E. degrees from Tsinghua University, Beijing, China, in 1998 and 2001, respectively, and the Ph.D. degree from the University of California at Los Angeles in 2008.

He is a research engineer with Ford Motor Company, Dearborn, Michigan. His main research interests include hybrid vehicle control, powertrain control and vehicle electrification technologies.

Dr. Wang is a member of ASME and SAE.



**Ryan McGee** (M '11) received the B.S.E.E. and M.Eng E.E. degrees from Cornell University, Ithaca, New York, in 1997 and 1998, respectively, and the M.B.A. degree from the University of Michigan, Ann Arbor, Michigan, in 2004.

He is a Technical Expert in Electrification Research and Advanced Engineering at Ford Motor Company. His research focus is on optimal and intelligent control systems applied to electrified powertrains.

Mr. McGee is a member of SAE.



**R. Brent Gillespie** received the B.S. degree in mechanical engineering from the University of California, Davis, in 1986, the M.M. degree in piano performance from the San Francisco Conservatory of Music in 1989, and the M.S. and Ph.D. degrees in mechanical engineering from Stanford University in 1992 and 1996, respectively. He is currently with the Department of Mechanical Engineering, University of Michigan, Ann Arbor. His current research interests include haptic interface and teleoperator control, human motor control, prosthetics, and robot-assisted

rehabilitation after neurological injury.



**Jeffrey A. Cook** (M'84–SM'98–F'02) received the B.S.M.E. degree from The Ohio State University, Columbus, Ohio, in 1973 and the M.S. degree in electronic and computer control systems from Wayne State University, Detroit, Michigan, in 1985.

He is an Adjunct Research Professor and Intermittent Lecturer in Electrical Engineering and Computer Science, and Intermittent Lecturer in Naval Architecture and Marine Engineering at the University of Michigan, Ann Arbor. From 1976 to 2007 he worked at the Ford Motor Company, Scientific Research Laboratory where he held technical and management positions in the United States and Europe. His research interests are in the area of embedded control systems and software.

Mr. Cook is a member of the ASME. In 2009 he led the IEEE Control Systems Society Committee on Outreach Curriculum Development Project.



**Jessy W. Grizzle** (S78M79SM90F97) received the B.S.E.E. degree from Oklahoma State University, Stillwater, in 1979 and the M.S.E. and Ph.D. degrees from the University of Texas at Austin, in 1980 and 1983, respectively.

He is currently the Jerry W. and Carol L. Levin Professor of Engineering and a Professor of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor. His research interests span theoretical nonlinear control, automotive systems, biomedical systems, and robotics.

Prof. Grizzle is currently a Senior Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL.