Incorporating Drivability Metrics into Optimal Energy Management Strategies for Hybrid Vehicles Part 2: Real-World Robustness and Constraint Implementation

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

Abstract—**Hybrid vehicle fuel economy and drive quality are coupled through the "Energy Management" controller that regulates power flow among the various energy sources and sinks. Most analytical studies have evaluated closed-loop performance on government test cycles, and there are few results that compare optimal control algorithms to the controllers employed in industry. This second of a two-part paper studies controllers designed using Shortest Path Stochastic Dynamic Programming (SPSDP), a stochastic optimal control design method which can respect constraints on drivetrain activity while minimizing fuel consumption. Part 1 described the problem formulation, models, and simulation results on government test cycles for a prototype vehicle. In Part 2, controllers are evaluated for robustness through simulation on large numbers of real-world drive cycles and compared to a baseline industrial controller developed by Ford. On real-world driving data, the SPSDP-based controllers yield 10% better fuel economy than the baseline controller, for the same engine and gear activity. SPSDP controllers are further evaluated for robustness to the drive cycle statistics used in their design. Simplified drivability metrics introduced in Part 1 are validated. Looking ahead, production implementations of the SPSDP method will likely require finer control over drivetrain behavior. Several options for achieving this are discussed, along with their relative benefits for performance versus computational tractability.**

I. Introduction

Designing an effective Energy Management controller for hybrid vehicles is a challenging, rich problem. The method used here is Shortest Path Stochastic Dynamic Programming (SPSDP), which generates optimal, causal controllers that can control complex vehicle behavior and respect constraints while minimizing fuel consumption. This second part of a two-part paper builds on the theoretical framework, methods, and test cycle simulations in Part 1 [1].

In this paper, we address practical considerations required to actually use SPSDP [1] in industrial development or production. Controllers are compared to a baseline industrial controller on real-world and government test cycles to evaluate performance, robustness, sensitivity, fuel economy vs. drivability tradeoffs, and real-world vs. test cycle performance. The drivability metrics of Part 1 [1] are validated, and techniques are discussed for incorporating additional vehicle behavior into SPSDP controllers.

SPSDP has the potential to be used directly in production vehicles with minimal manual tuning. To test controller performance and robustness under realistic conditions, about 500,000 simulations are conducted using large numbers of real-world drive cycles. With these data, controller performance can be evaluated and optimized with respect to various classes of driver behavior [2], [3], [4], [5], [6] in addition to government certification cycles [7], [8], [1]. These simulations are analyzed to determine not just mean performance, but standard deviations and worst-case performance. This real-world robustness testing addresses a common customer complaint that the fuel economy shown on the "window sticker" does not match the vehicle performance obtained in practice [9], [10], [11], [12], [13]. To provide a realistic benchmark, SPSDP energy management controllers are compared to an industrially-designed "baseline" controller.

As part of this real-world evaluation, controllers are constructed using statistics from multiple sets of design cycles, including both real-world and government test cycles. They are then evaluated on multiple sets of simulation cycles of various types. This allows evaluation of controller robustness to different drive cycles and the effects of the statistics assumed in the controller design.

A major enabling result in the present work was the development of the drivability metrics for powertrain activity presented in [1]; this process is discussed in detail here. The simplified drivability metrics are shown to be well correlated with more detailed metrics on both government test and real-world driving. This validates the simplified metrics as a useful approximation for controller design.

The main decision in the design of an SPSDP controller is the selection of a cost function which reflects desired vehicle behavior while maintaining feasible off-line computation. This paper focuses on two specific drivability attributes: the engine and transmission behavior. If the SPSDP method is to be used in practice, engineers will likely seek to incorporate additional criteria for vehicle behavior. Many other performance attributes can be studied using this theoretical framework, but care must be taken to avoid excessive computational burden. Several options for incorporating vehicle behaviors in the problem formulation are discussed.

The paper is organized as follows: After a statistical comparison of real-world and government test drive cycles, Section II examines how various controllers (each designed for a given drive cycle) perform on real-world drive cycles. Section III establishes the robustness of controller performance to

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.

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

different drive cycles, including the dependence of engine events (a drivability metric) and fuel economy. Section IV develops a reduced (simplified) set of drivability metrics and validates their use by comparison to a complex set of drivability metrics. Section V discusses further methods for controlling vehicle behavior using SPSDP. Finally, section VI is a general discussion of the SPSDP method used in both Parts 1 and 2 of this paper.

II. Robustness to Real-World Driving

A. Motivation

Controller performance is often reported on standard test cycles (FTP, NEDC, US06) for comparison and relevance to government certification. If real-world fuel economy is lower than on government test cycles, either the controllers are tuned primarily for the test cycles, or real-world driving is fundamentally less fuel-efficient than the test cycles. The real-world data used in this paper is aimed at evaluating controller robustness and performance in the "off-cycle" real world.

B. Drive Cycle Data

The drive cycle data used in this paper was collected by the University of Michigan Transportation Research Institute (UMTRI) [14]. The "source" data set supplied to us contains 2500 trips made by 87 drivers. Very short trips (less than 3 minutes or 0.5 km) are ignored. We randomly selected two sets of 100 drive cycles from the UMTRI data. They are called "Ensemble 1" and "Ensemble 2."

To gain some insight into the nature of the drive cycles, we briefly study the characteristics of their distributions. The cumulative distribution functions (CDF) of trip distance for the source data and both subsets are shown in Fig. 1a. The statistics for the two Ensembles are a reasonable match for the source data set. Each Ensemble represents about 1000 miles of driving, or 3 tanks of gas.

The CDFs of vehicle speed for Ensembles 1 and 2 are depicted in Fig. 1b, using vehicle velocity on a second-by-second basis. Two standard government test cycles are also shown, the Federal Test Procedure 75 (FTP) and the New European Drive Cycle (NEDC). This yields a total of five total in the figure: the Source Data, *Ensemble 1*, *Ensemble 2*, FTP, and NEDC.

There are three interesting things to notice in Figure 1b. The first is that the government test cycles are fundamentally different from the real-world data. The real-world cycles contain substantially higher velocities in general. The second detail is the step-like nature of the NEDC cycle, which arises because it is contrived. The cycle is composed of perfect ramps to constant speeds and is specified by hand. Lastly, *Ensemble 2* has lower velocities than *Ensemble 1*, a difference that will be reflected in the fuel economy results presented in Section II-D.

C. Simulation Procedure

To study the effectiveness of the SPSDP controller design methodology, controllers are first designed based on statistics from particular drive cycles [1]. Large numbers of controllers are then simulated on a set of real-world driving data using two different methods:

- **Concatenated Cycle** - The Ensemble of 100 cycles is assumed to represent one vehicle's driving history, about 1000 miles. The starting SOC of the first trip is 0.5, and the starting SOC for each subsequent trip is the ending SOC of the previous trip.
- **Individual Cycles** - Each cycle is studied individually, and the starting SOC for each trip is 0.5.

Each method has its advantages, so both are used here. For the **Concatenated Cycle**, fuel economy for the Ensemble is simply the total fuel consumption divided by the total distance. Drivability events are summed over all trips. The engine is off at the start and end of each trip. The total fuel consumption is corrected based on final SOC, but the battery energy is negligible compared to the fuel consumption on such a long trip. The SOC correction (see (22) in [1]) impacts fuel economy less than 0.1%. This is one major advantage of this method: there is no concern about SOC correction yielding false fuel economy results. This method is arguably more representative of typical driving as the SOC varies at the start of each trip. However, the Ensemble trips are selected from a group of drivers, so they may better represent a vehicle shared by a household rather than a single driver.

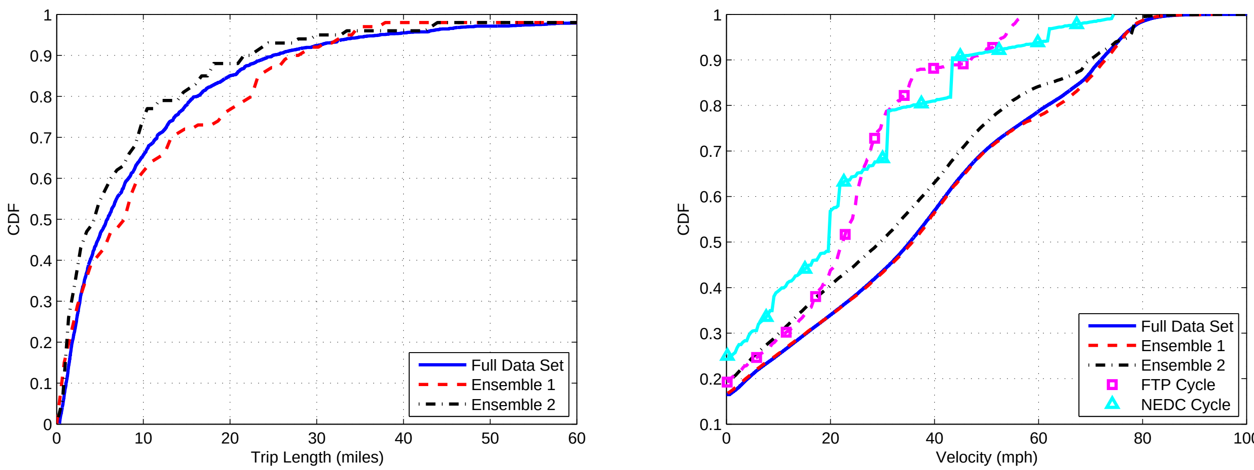
For the **Individual Cycles**, total fuel use is individually corrected for each cycle based on SOC. The fuel economy for the Ensemble is computed as the total of corrected fuel consumption divided by total driving distance. This yields a weighted average over all the trips. The drivability events are the sum of all the trips. This method is very useful for generating statistics. Each controller now has 100 sample points which can be used to calculate the mean, standard deviation, and $10^{th}/90^{th}$ percentile bands for fuel economy and final SOC. This allows deeper understanding than simply studying the weighted average of fuel consumption.

D. Results

The SPSDP controllers generate significantly better performance than the baseline controller on real-world cycles and are reasonably robust to variations in driving patters and the statistics used to design the controller.

The Ensemble is first treated as a Concatenated Cycle representing a single vehicle (Sec. II-C). Fuel economy and drivability results are shown in Fig. 2a. The figure shows five controllers sets evaluated on Ensemble 1: the baseline controller and 4 SPSDP controller families designed on statistics from FTP, NEDC, Ensemble 1 and Ensemble 2. Recall that a controller family is developed by fixing the driver statistics and sweeping the drivability penalties to generate a tradeoff curve (see [1]). All fuel economy numbers are normalized to the baseline controller running FTP, which has fuel economy 1 (Fig. 4a and Fig. 7c in [1]) just as in Part 1 [1].

The cycles in each Ensemble are also treated as Individual Cycles, where the SOC always starts at 0.5 (Section II-C). The weighted average of fuel economy compared to Engine Events is shown in Fig. 2b for Ensemble 1. The superiority of



(a) Cycle Distance CDF

(b) Velocity CDF

Fig. 1: Statistics of Real-World Driving. Fig. 1a is the Cumulative Distribution Function of trip distance for the source data and two subsets. Mean distances for the sets are: Full Data Set - 11.7 mi., *Ensemble 1* - 12.7 mi., *Ensemble 2* - 9.9 mi. Fig. 1b is the Cumulative Distribution Function of vehicle velocity for the source data, two subsets, and two government test cycles.

the SPSDP controllers with respect to the baseline controller is maintained under real-world driving conditions as shown in Figure 2. For the same drive quality index, the fuel economy is approximately 10% higher.

Using the individual simulation method, each controller now has 100 data points representing performance on individual drive cycles. The family of controllers designed with Ensemble 1, shown as a black curve in Fig. 2b, is selected for more detailed study. The (non-weighted) mean, standard deviation, and $10^{th}/90^{th}$ percentile bands are calculated for corrected fuel economy and final SOC for each controller running both Ensemble 1 and Ensemble 2. A response surface is fitted to these data as shown in Fig. 3. The same statistics are calculated for the baseline controller. The distributions are not Gaussian and differ with the number of engine events as well as the simulated Ensemble.

Results like Figure 2 are available [15] for controllers running Ensemble 2, and the results are very similar. Interested readers may study the sensitivity to driving cycles and design statistics on both Ensembles. Additional SOC results are also shown.

E. Discussion

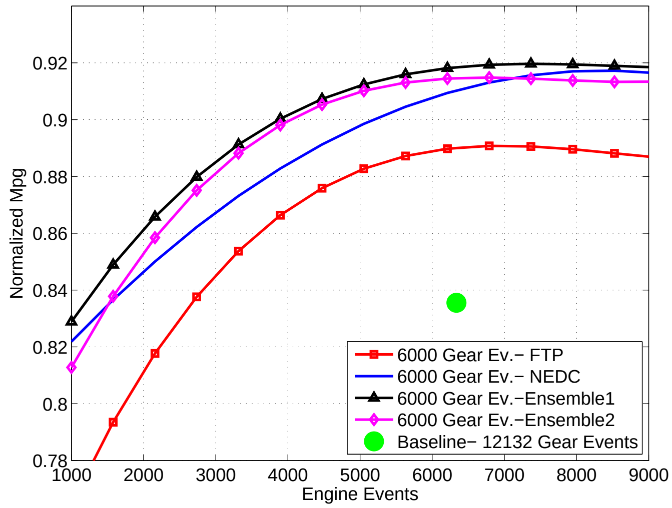
The performance of the SPSDP controllers on the concatenated Ensembles confirms that their superiority is legitimate and not an artifact of SOC correction; the energy in the SOC errors is minimal on such a long cycle. Figure 2 demonstrates that simulating the cycles individually and correcting for SOC yields results similar to the concatenated version. Specifically, the SOC correction method is valid and the exact starting SOC for each cycle is not crucial for overall performance. This is quite useful because the simulation of individual cycles is easily parallelized and avoids the difficulties of simulating such a long (16 hour) cycle.

Figure 2 shows that the SPSDP controllers and the baseline controller yield lower fuel economy in the real-world than on government test cycles, implying the difference is fundamental to the cycle itself and not a result of controller tuning. On FTP the baseline controller has fuel economy 1 and the SPSDP controllers do even better. This confirms a known weakness with the government test cycles: they are not representative of real-world driving. Real-world driving requires 15-20% more fuel consumption than the test cycles. Recall the differences among cycles illustrated in Fig. 1b. This causes a significant mismatch between the "window sticker" fuel economy from certification testing and the fuel economy consumers would measure in practice. Recent changes to the government testing procedure in the US have attempted to address this problem and bring the certification fuel economy predictions closer to real-world practice [16], [17]. Additional results on the government test cycles are presented in Section III-A (Fig. 4).

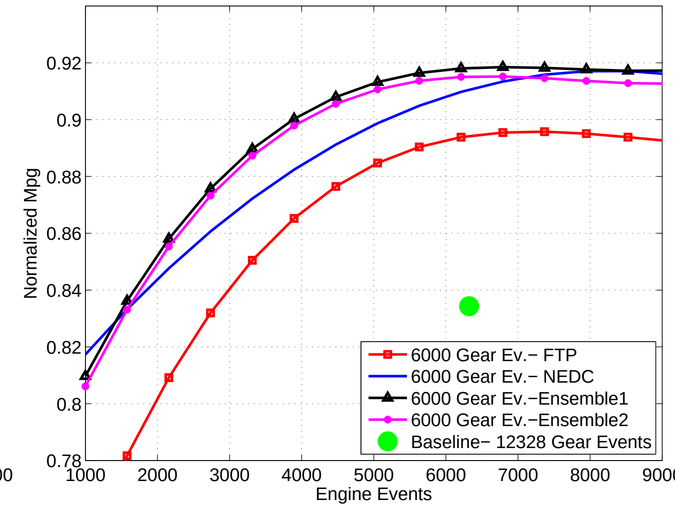
The fuel economy curves in Figs. 2 and 3 show a distinct "knee", where the tradeoff between fuel economy and engine activity becomes acute. In both figures, the number of engine events may be reduced from 9,000 to 5,000 with no significant loss of fuel economy. This illustrates the power of having this optimal Pareto curve available. Not only is the tradeoff quantified, but in some cases it may be possible to reduce engine activity without any sacrifice in fuel economy. Without such a curve, a fuel-optimal controller may be designed with 9,000 engine events without the designers knowing the same fuel economy can be attained with roughly half the engine activity.

The statistical analysis of the individual cycles in Fig. 3 shows very consistent performance. For the SPSDP controllers, performance one standard deviation below the mean still exceeds the mean of the baseline controller.

Battery charge maintenance for real-world driving is very



(a) Fuel Economy for Ensemble 1 - Concatenated Cycle



(b) Fuel Economy for Ensemble 1 - Individual Cycles

Fig. 2: Fuel Economy and Drivability Metrics on Ensemble 1 when simulated as Concatenated and Individual Cycles for the baseline controller and SPSDP controller *families* designed with statistics from FTP, NEDC, *Ensemble 1*, and *Ensemble 2*. Fuel Economy, Gear Events, and Engine Events are cumulative for the whole cycle set, representing approximately 1000 miles. Results are normalized to the baseline controller on FTP, as in [1].

consistent as shown in Figures 3c and 3d; there is little variation across the different drive cycles. The nominal target SOC is 0.5, which is nearly achieved in the high fuel economy operating region (above 5000 engine events). Both figures show a trend towards higher SOC with fewer engine events. As engine events become more costly, the controller tends to operate at a higher SOC, presumably to stay in electric mode and avoid engine starts whenever possible. The SPSDP controllers also maintain tighter control of final SOC compared to the baseline controller, as evidenced by the standard deviation and percentile bands.

The results in these two papers represent more than 4000 controllers and 500,000 simulated cycles (most in Figs. 2 and 3), requiring roughly 10.5 CPU-years of computing time on a cluster of desktop-class machines at the University of Michigan Center for Advanced Computing. Most problems can be solved with fewer controllers and cycles: the sensitivity, robustness, and real-world driving studies are not required in every application.

III. Robustness to Variable Driving Styles and Sensitivity to Design Cycle Statistics

This paper uses drive cycles for two purposes: to design a controller, and then to simulate its performance. As described in [1], SPSDP uses a Markov chain to model driver behavior, with the states and transition probabilities extracted from one or more design cycles provided during the controller development process. The statistical driver model clearly affects the final controller and the obtained closed-loop behavior, which opens a question about the relationship between the (possibly different) cycles used to design and simulate the controller. Here we investigate the sensitivity of closed-loop behavior to the assumed driver statistics.

In Section II-D, four different *families* of controllers were designed using statistics from FTP, NEDC, Ensemble 1 and Ensemble 2. These controllers were then simulated on the two Ensemble sets [15]. In this section, these four families of controllers are also simulated on the FTP and NEDC cycles.

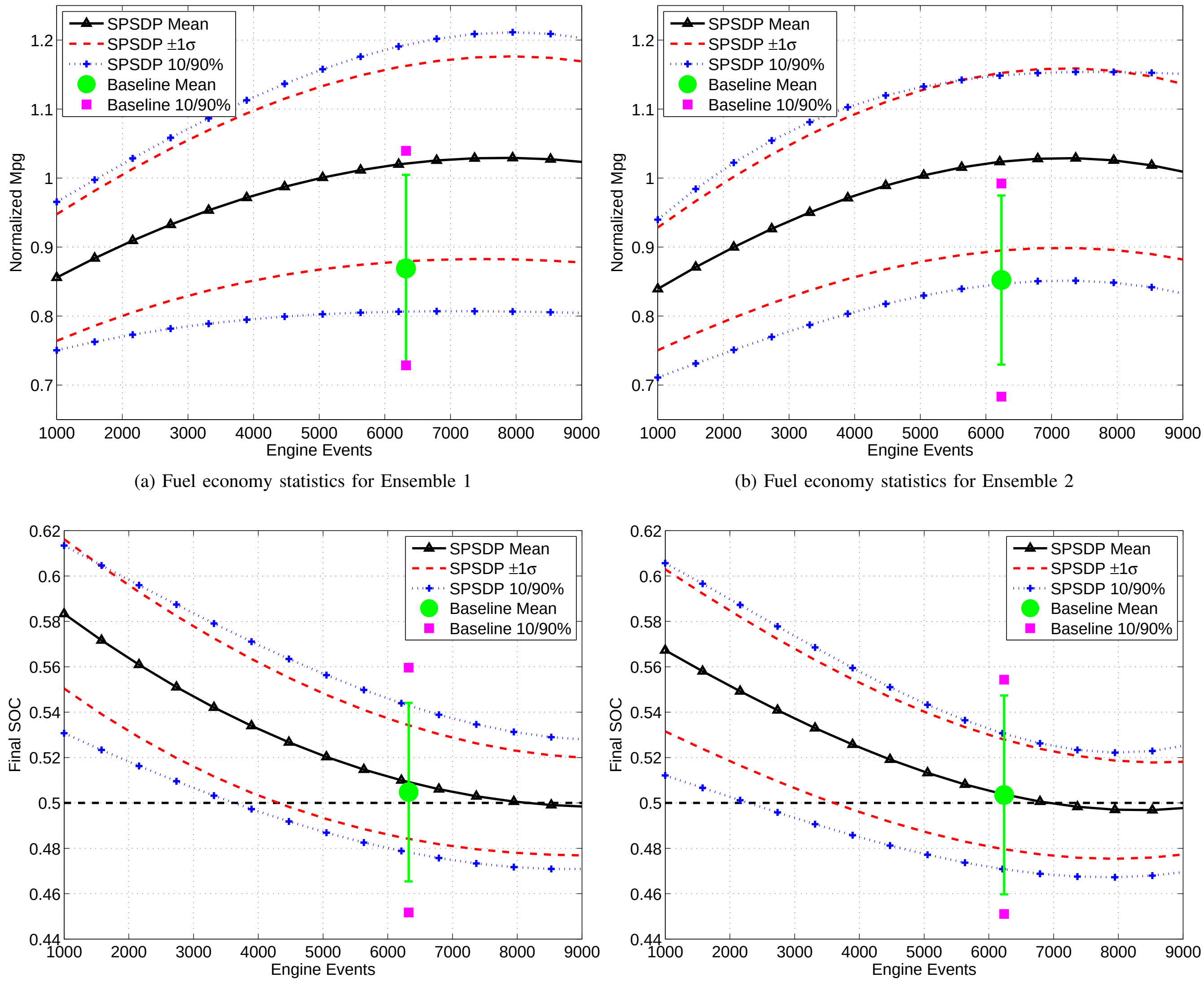
A. Results

In general, the SPSDP method is relatively insensitive to the choice of cycles used to design the controllers, although the controller that performs best on a given cycle is generally the controller designed for that cycle, as would be expected. The controllers from Section II-D are shown running the FTP and NEDC cycles in Figure 4, using the same markers as in Figure 2.

B. Discussion

This cross-combination of design and simulation cycles allows study of a controller's performance on both the design and arbitrary cycles. Excluding the NEDC, design cycle statistics cause less than 3.5% difference between controllers across all other cycles. The fuel economy difference when using Ensemble 1 or 2 statistics is small, which indicates that the sample sizes are large enough to reasonably represent typical driving.

This idea is reinforced in the surprising robustness of controllers when dealing with the FTP cycle. The enormous 20% fuel economy difference between FTP (fig. 4a) and Ensemble 1(Fig. 2) seems to indicate that the statistics of the two cycles are not representative of each other. Nevertheless, controllers designed on the Ensemble cycles do very well on the FTP cycle, and the FTP-based controllers sacrifice 3% performance on the Ensemble cycles. While the FTP and



(a) Fuel economy statistics for Ensemble 1

(b) Fuel economy statistics for Ensemble 2

(c) Final SOC statistics for Ensemble 1

(d) Final SOC statistics for Ensemble 2

Fig. 3: Statistical fuel economy. The SPSDP controller *family* designed on *Ensemble 1* is simulated on *Ensemble 1* and *Ensemble 2*, and each cycle is corrected for SOC. The mean, standard deviation, and 10^{th} and 90^{th} percentile are calculated. The mean, standard deviation (error bars), and 10^{th} and 90^{th} percentile are also calculated for the baseline controller.

Ensemble cycles are very different, the statistics from either cycle are sufficiently representative to generate controllers that perform reasonably well on both cycles.

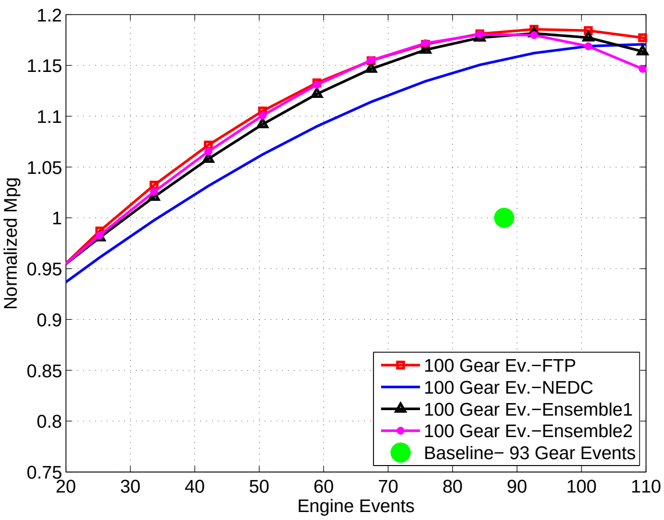
The NEDC cycle is somewhat of an outlier in this study. NEDC is used to address the European market in addition to US regulations. Although the fuel economy of the NEDC-based controllers is reasonable on the FTP and Ensemble cycles, the NEDC is generally avoided as a test cycle . As seen in Figure 1b, the NEDC cycle is vastly different from both real-world driving and the FTP. It is periodic and clearly specified by hand, which is not well suited to a Markov chain based expectation.

Figures 2 and 4 allow a comparison of performance on real-world and government test cycles, as mentioned in Section II-E. The controllers are the same in each figure and use consistent markers, they are just running different cycles. The fuel economy normalization is the same in all figures. The baseline controller drops from a fuel economy of 1 Mpg on FTP (Fig. 4a) to 0.835 Mpg on Ensemble 1 (Fig. 2a). The best SPSDP controllers achieve 1.18 Mpg on FTP, but only 0.92 Mpg on Ensemble 1.

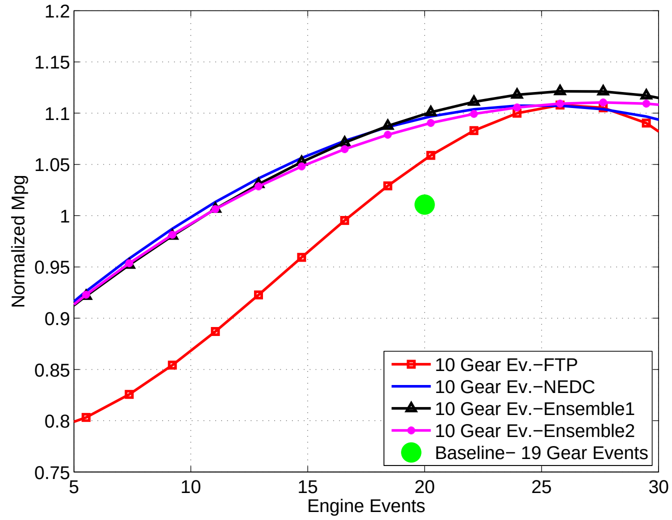
IV. Development and Validation of Drivability Metrics

A. Formulation

As mentioned in Part 1 of this paper [1], two significant characteristics that are noticeable to the driver are the basic behaviors of the transmission and engine. There are a large number of qualitative characteristics that describe powertrain behavior [18]. While metrics exist to quantify a large variety of behaviors, overall value judgements are largely qualitative. There are no high-level rules or metrics that exactly quantify



(a) FTP Cycle



(b) NEDC Cycle

Fig. 4: Fuel Economy and Drivability Metrics on the FTP and NEDC Cycle for 5 controller options. Controller *familes* are designed with statistics from FTP, NEDC, *Ensemble 1*, and *Ensemble 2*. All fuel economy figures are normalized to the baseline controller performance on FTP, shown as a large green dot in Fig. 4a. The controllers are the same as those shown in Fig. 2.

the overall drivability performance or describe the relative importance of various behaviors. An important contribution of the work reported here is the translation of these value judgements into quantitative metrics than can be used in the optimization formulation. The first step is to describe and quantify engine and transmission behaviors using performance metrics, and the second step is to reduce the complexity of these metrics so that they may be easily used in optimization.

A primary concern in drivetrain activity is the the frequency and timing of events, like gear shifts and engine start/stop. Two categories of metrics are used, the mean time between events and the number of short-duration events; the latter are especially bothersome to drivers. A short duration event occurs when the dwell time in a particular state is less than some specified value; the metric is the number of these occurrences. This type of metric is denoted "Dwell time less than X seconds," where X is the cutoff criteria. These "mean" and "short duration" categories of metrics applied to the engine and transmission generate 7 distinct metrics, termed the "complex" metrics. These 7 metrics represent a detailed description of vehicle behavior and are shown in the top table in Fig. 5. Many other metrics could obviously be used, but these are an important subset of the possibilities.

For the transmission, a particularly annoying short-duration event is "hunting," rapid shifting back and forth between the same two gears. We define a gear "hunting" event as a sequential upshift-downshift or downshift-upshift that occurs faster than some cutoff time X. The metric is the number of occurrences of a hunting event. This type of shifting often occurs in normal driving, but only becomes bothersome when the shifts are closely spaced. Shifting that is frequent or perceived to be unnecessary is often termed shift "busyness," and is reflected in both mean dwell time and short duration metric categories.

	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. 5: 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.

The most bothersome engine events are those of very short duration, and to some extent drivers ignore the long-horizon (10-20s) engine behavior as long as there are no short-duration events.

Although it is theoretically possible to incorporate these metrics in the optimization formulation, the computation burden of the required additional states makes the problem intractable. Another disadvantage is the large parameter space of penalty weights for the various metrics. Even if all these metrics were directly implemented and a controller computed, the control designer is left with a very complex design process.

Therefore, we choose to simplify these complex metrics into something that can be easily used. Ideally, the information contained in the seven metrics listed above could be distilled into a smaller number of simple metrics. Indeed, the behaviors measured by these metrics are well correlated with the two simple metrics proposed below, which allows effective control of complex behavior with a simple implementation.

B. Simplified Drivability Metrics

The seven complex metrics are reduced to two baseline metrics to quantify behavior for a particular trip. These are the two simplified metrics described in Part 1 [1]. 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. This reduction is depicted in Fig. 5.

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.

The utility of these simplified drivability metrics is validated by studying how they relate to the more detailed metrics. By finding simple metrics that are well correlated with the complex metrics, one can incorporate the simple metrics into the full SPSDP algorithm to maintain some control over complex behavior while keeping the problem feasible.

C. Drivability on Government Test Cycles

Simulations on the FTP cycle show strong correlations between the simple and complex metrics. For a *family* of controllers running the FTP cycle, drivability metrics are recorded for both the simple and complex metrics. The engine activity is studied in Figs. 6a-6c. The metric *Engine Events* is shown on the horizontal axis for all 3 figures. The mean engine on and off times are shown in Figure 6a. Short duration engine events are studied next, the engine on and off Dwell Time less than X seconds are shown in Figures 6b and 6c respectively for various cutoff criteria X.

The gear shifting activity of the vehicle is studied in Figures 6d-6f. The metric *Gear Events* is shown on the horizontal axis for all 3 figures. Figure 6d shows the mean dwell time in gear. This metric studies the time spent in one particular gear without shifting or disengaging the clutch. Short durations between shifts or clutch disengagements are indicated by gear dwell times less than some variable cutoff criteria as shown in Figure 6e. Transmission gear "hunting" is shown in Fig. 6f for varying time length criteria.

These figures (Figs. 6a-6f) show that the complex metrics are approximately monotone functions of the simple metrics. To highlight this property, the data in Figures 6c and 6e are shown with a straight line least squares fit to the data. Each cutoff time criteria is treated separately and each line matches the color shown for the underlying data.

The relationships between the metrics in these figures are very nearly piecewise linear. They generally are zero up to a certain value on the horizontal axis, then follow a straight line. These data are fit with functions of the form

$$\hat{y} = max(0, mx + b)$$

where m and b are the slope and intercept of the best straight-line fit to the nonzero data; coefficients are available in [15].

D. Drivability on Ensemble Cycles

The FTP results demonstrate that simplified drivability metrics of total engine events and total gear events can yield good vehicle behavior when evaluated in terms of more detailed metrics such as engine on-off dwell times and gear hunting. This is also true for real-world driving on the Ensemble cycles. This particular vehicle is relatively insensitive to gear activity, so here we focus on engine activity.

As discussed in Section II-C, there are two ways to study the Ensemble cycles. The first is to treat each Ensemble as a single trip, about 1000 mi. The second method is to treat each cycle individually as a unique data point. The drivability metrics are studied using these two methods. The three detailed engine activity metrics (Figs. 6a-6c) are first studied for the Concatenated Ensemble 1 in the top row of Fig. 7 (Figs. 7a-7c). Each data point represents one controller running a single simulation, the Concatenated Ensemble 1. Each data point represents the same cycle, and variation is tied to controller tuning (i.e., choice of penalties in the cost function).

The second method, treating each cycle individually, is shown in the bottom row of Figs. 7d-7f. Each data point represents a controller running one of the 100 cycles in Ensemble 1. In this case, variation between data points arises from different controller tunings *and* different cycles.

E. Discussion

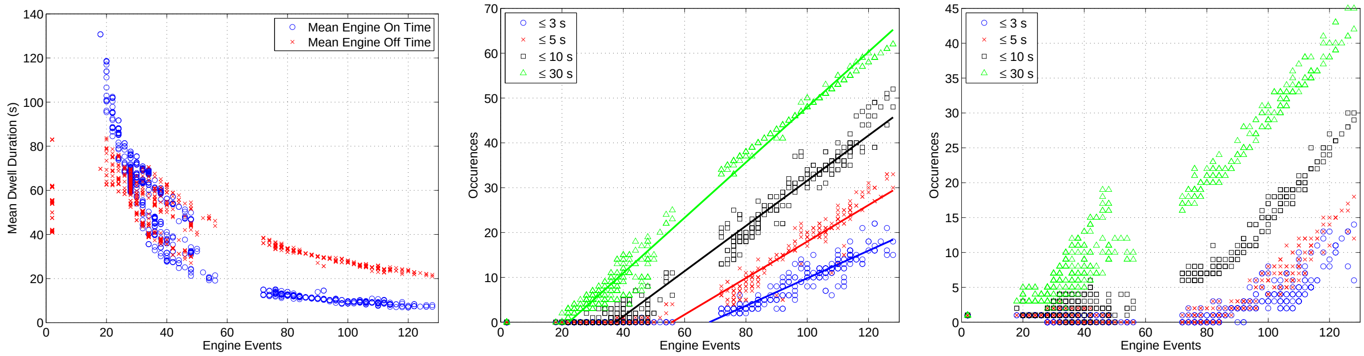
The strong correlations between the simple and complex metrics allow the drivability attributes to be easily quantified. A designer can be confident that the simple metrics are directly related to the complex versions, and that prescribing behavior with respect to the simple metrics will yield the desired results. The problem can be greatly simplified in that the designer is not required to specifically track and control each complex behavior of interest. The main algorithm will generate tunable performance that meets the criteria in a general sense.

For example, optimizing for fuel economy often leads to gear hunting behavior near a shift point. As the total number of shifts is penalized and reduced, hunting behavior is usually eliminated first as these frequent shifts do not significantly improve fuel economy.

Similarly, for fuel-optimal operation, the engine on/off decision can become very sensitive to driver demand, causing many short-duration engine events even when the driver applies a nearly constant pedal input. Reducing the total number of engine events tends to eliminate these short-duration events (Fig. 6b) and make the engine state less sensitive to the driver demand.

The detailed drivability metrics are still related to the simplified metrics in an approximately monotone fashion on real-world driving (Fig. 7). For the concatenated Ensembles (Figs. 7a-7c, the correlation is even more clear than on FTP. The nearly straight-line fits for short duration engine on/off events demonstrate that the simple and detailed metrics are related by a nearly constant ratio, which was unexpected.

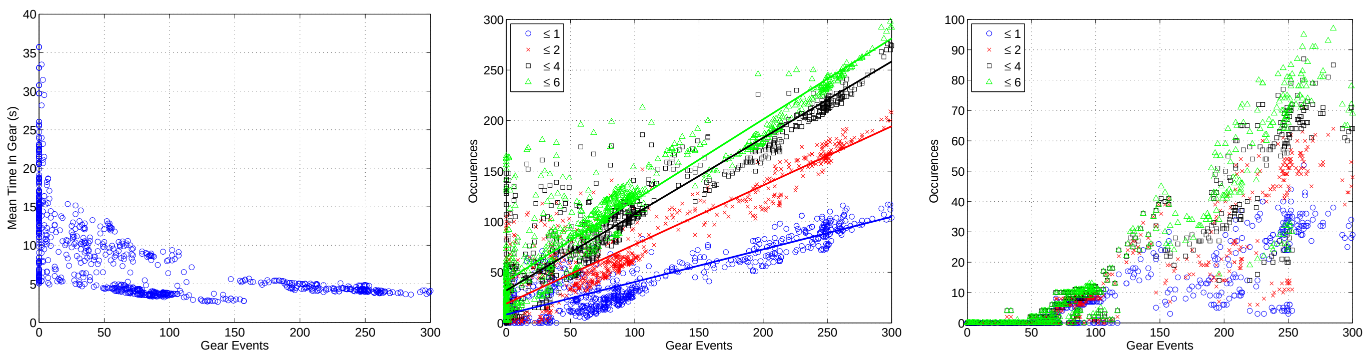
Perhaps most surprising are the results for the individual Ensemble cycles (Figs. 7d-7f). Each plot now depicts data



(a) Mean engine on and off durations compared to *Engine Events*.

(b) Engine on durations less than some number of seconds compared to *Engine Events*. Data are shown for cutoffs of 3, 5, 10, and 30 seconds.

(c) Engine off durations less than some number of seconds. Data are shown for cutoffs of 3, 5, 10, and 30 seconds compared to *Engine Events*.



(d) Mean dwell time in gear between shifts or clutch disengagements compared to *Gear Events*.

(e) Gear dwell times less than some number of seconds between shifts or clutch disengagements compared to *Gear Events*. Data are shown for cutoffs of 1, 2, 4, and 6 seconds.

(f) Gear hunting events that occur within some number of seconds. This is a sequential upshift/downshift or upshift/downshift that occurs within a set period of time. Data are shown for cutoffs of 1, 2, 4, and 6 seconds in comparison to *Gear Events*.

Fig. 6: Comparison of simple and complex drivability metrics on the FTP cycle. Complex engine activity metrics are compared to the simplified *Engine Events* metric in subfigures 6a-6c. Three gear activity metrics are compared to the simplified *Gear Events* metric in subfigures 6d-6f. Subfigures 6b and 6e have straight-line fits to the data.

points representing 100 different cycles with different controllers. The drivability behavior is no longer directly related to the controller but confounded by different cycles, yet the straight-line trends are still clearly visible. Even for the real-world drive cycles, it seems that the short duration engine event metrics are still related to the simple metrics by a nearly constant ratio.

These methods are quite effective and produce controllers that are very well behaved. This method is very useful for adjusting vehicle behavior in practice. Suppose a controller is selected based on simulation. Once implemented, the engine state is very sensitive to pedal, as in the previous example. The designer simply selects a new (still optimal) controller with higher penalties and thus engine events, and the engine state becomes less sensitive to the pedal. The designer has an easy way to control behavior that is simple to tune and still optimal. Contrast this to a rule-based controller, where the designer changes the rules that determine when the engine turns on. It is very difficult to directly tune those rules and maintain the best fuel economy.

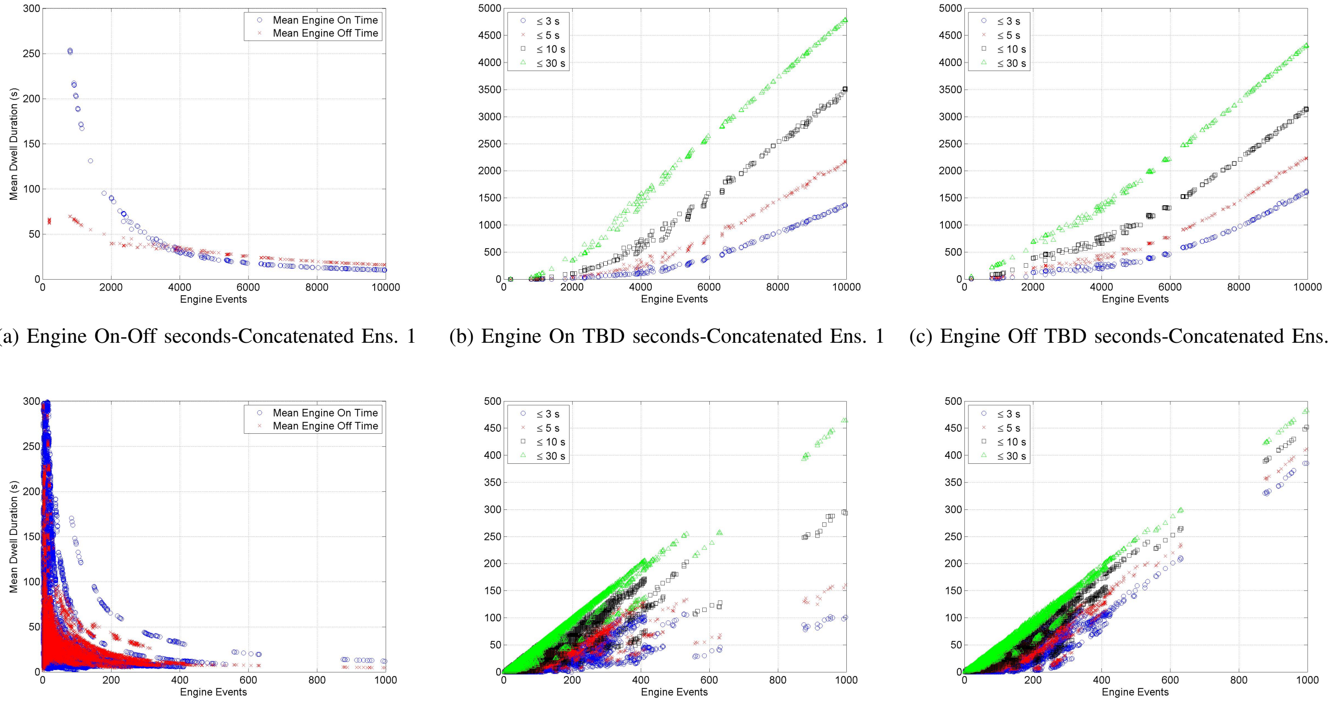
V. Controlling Additional Behavior using SPSDP

A. Motivation

The design of effective energy-management controllers for production vehicles is a challenging problem because designers must consider many different vehicle behaviors. In this paper we study drivability, but engineers must also consider other attributes like emissions, battery wear, durability, noise, etc. Even if an attribute is already considered, designers may want more detailed control. Regardless of the complexity of the current algorithm, designers are often seeking to control additional vehicle behaviors.

This section discusses methods for incorporating vehicle behavior into the SPSDP algorithm. The descriptions largely refer to a design process: a basic SPSDP controller exists, and a designer would like to adjust additional vehicle behavior. This approach covers several levels of complexity. If the basic SPSDP controller minimizes fuel, the designer may want to add drivability (as we did in this paper). A more advanced SPSDP controller may handle fuel and drivability but also require emissions control, for example.

These additional vehicle behaviors could also be addressed



(a) Engine On-Off seconds-Concatenated Ens. 1 (b) Engine On TBD seconds-Concatenated Ens. 1 (c) Engine Off TBD seconds-Concatenated Ens. 1

(d) Engine On-Off seconds-Individual Ens. 1 (e) Engine On TBD seconds-Individual Ens. 1 (f) Engine Off TBD seconds-Individual Ens. 1

Fig. 7: Comparison of simple and detailed drivability metrics for concatenated and individual Ensemble 1 cycles. Detailed engine activity metrics are compared to the simplified *Engine Events* metric in Figs. 7a-7c for the concatenated Ensemble, where the 100 cycles are treated as a single trip. Each marker represents the same drive cycle, the concatenated Ensemble 1. The same metrics are studied for the individual Ensemble 1 in Figs. 7d-7f, where each cycle is simulated individually. Each marker represents one controller running one of 100 cycles, so the markers no longer represent the same cycle.

with a rule-based controller that operates downstream of the SPSDP controller, but it is more satisfying to incorporate these behaviors in the problem formulation. If these vehicle behaviors require additional model states, solving the Bellman equation for SPSDP can become intractable, so here we discuss alternative methods with their benefits and drawbacks.

B. Formulation Methods

Fundamentally, controlling vehicle behavior requires the designer to specify a cost function for a model. Suppose we have a basic cost function $c(x, u)$ and model $f(x, u, w)$ that incorporates some level of functionality. The standard on-line SPSDP controller is

$$u^*(x) = \operatorname*{argmin}_{u \in U} E_w[c(x,u) + V^*(f(x,u,w))]. \quad (1)$$

We discuss three possible methods to incorporate additional behaviors which offer a tradeoff between computation and optimality:

1) Additional Penalties: It may be possible to control additional behavior by adding penalties to the cost function using only the states in the base model $f(x, u, w)$. If so, this yields a new cost function,

$$c_{pen}(x,u) = c(x,u) + \alpha \mathbf{I}_{Event1}(x,u) + \beta \mathbf{I}_{Event2}(x,u) \ldots \quad (2)$$

where $\mathbf{I}(x, u)$ is the indicator function of when a behavior occurs.

The solution complexity for the value function size remains the same since the base model is still used,

$$V^*_{pen}(x) = \min_{u \in U} E_w[c_{pen}(x,u) + V^*_{pen}(f(x,u,w))]. \quad (3)$$

This is termed the "Additional Penalties" method and yields an optimal control law. Even though the value function computation remains the same, the number of times $V_{pen}(x)$ must be evaluated to tune the controller is exponential in the number of penalties (α,β,...) in (2).

2) Extended Model: If the new behavior of interest requires additional states $\breve{x}$ or control inputs $\breve{u}$, the existing model is extended to include them

$$x_e = [x, \breve{x}] \quad (4)$$
$$u_e = [u, \breve{u}] \quad (5)$$
$$f_e(x_e, u_e, w). \quad (6)$$

The SPSDP controller design uses the extended cost $c_e(x_e, u_e)$ and dynamic model $f_e(x_e, u_e, w)$, and the new value function $V^*_e(x)$ is solved using the Bellman equation,

$$V^*_e(x_e) = \min_{u_e \in U_e} E_w[c_e(x_e,u_e) + V^*_e(f_e(x_e,u_e,w))]. \quad (7)$$

This is the standard, provably optimal method and is implemented using the on-line minimization (1). This is termed the "Extended Model" method. The complexity of computing the value function grows exponentially with system states, requiring more off-line computation. The number of value function solutions to tune the controller is still exponential in the number of penalties (α,β,...) in the cost function. This is the process used in this paper and its companion to include the simplified drivability metrics.

3) Instantaneous Cost: Additional states and penalties may also be incorporated solely in the on-line cost function [19], [20]. A basic SPSDP controller is designed first, and other behaviors are added later only in the on-line cost function (1). The value function $V(x)$ is first calculated via the Bellman equation using a simple cost $c(x,u)$ and model $f(x,u,w)$ that only includes minimal detail,

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

The real-time controller is then implemented using the simple value function and an extended cost function $c_e(x_e,u_e)$ that includes the additional vehicle behavior. The real-time controller must still track the extended set of states, but the value function is only solved for a reduced state space, requiring much less computation. The real-time controller is then

$$u^*(x_e) = \underset{u_e \in U_e}{\operatorname{argmin}} E_w[c_e(x_e,u_e) + V^*(f(x,u,w))]. \qquad (9)$$

The value function is approximated using the simple model $f(x,u,w)$, while the cost function is calculated using the extended model. In this case, the new behavioral restrictions are still implemented via optimization, but they are only instantaneous in the sense that there is no estimate of the future cost [20] because they are not included in the value function. This method is termed the "Instantaneous Cost" method. It can be implemented on-line with no additional off-line computation because it uses the original value function; it can modify vehicle behavior but the optimality guarantees are lost. The performance loss is highly dependent on the specific problem; drivability is studied in [19].

C. *Penalty Implementation*

Designers must carefully consider how events and penalties are defined; small changes in definition can save large amounts of computation. The selection of the drivability definitions and cost function for this paper was not easy, despite the simplicity of the final metrics.

Generating usable controllers requires three steps: selecting the cost function, calculating the value function off-line, and running a simulation. The mapping from penalty to behavior is not known a priori, so the simulation must be conducted to determine the behavior; tuning penalties accounts for most of the total computation. The relative benefits and drawbacks of the three methods are shown in Table I.

These three options are illustrated with examples from this work: Simplified drivability metrics were added to a fuel-only SPSDP controller using the Extended Model method by adding states to the value and cost functions. Once the current transmission gear state was included to track gear events, we added a penalty for non-sequential shifts (ie. $1^{st} - 3^{rd}$) without additional states using the Additional Penalties method. An Instantaneous Cost implementation can be used to limit the slew rate of engine power, which would otherwise require an additional state representing current engine power. This slew rate restriction was not added in the simulations in this paper, but was used in a hardware implementation.

Metrics and penalties for behaviors other than simplified drivability were also included or are planned, but have not been extensively studied. Their implementations are listed in Table II as examples. The Additional Penalties method is generally preferred because it maintains optimality, but not all penalties can be implemented this way. Several methods are often possible; Table II refers to the implementation we selected. Although in some cases additional penalties are computationally "free," from a practical perspective the number of possible parameter tunings still grows exponentially with additional behaviors.

TABLE II: Additional Event implementations

Behavior	Method
Series Mode Entry & Exit	Additional Penalties
Pedal Correlation with Engine Noise	Additional Penalties
Pedal Correlation with Engine Speed	Additional Penalties
Pedal Correlation with Engine Torque	Additional Penalties
Only single-step gear shifts	Additional Penalties
Only even-odd gear shifts	Additional Penalties
Emissions	Extended Model
Time delay between drivability events	Extended Model
Climate Control Loading	Extended Model
Slew rate of Engine Power	Instantaneous Cost
Slew rate of Engine Speed	Instantaneous Cost

D. *Computation*

As an example, this section quantifies the computation burden of adding the simplified drivability metrics using the Extended Model and Instantaneous Cost methods described in Section V-B.

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

Both methods require similar operations to calculate the current cost function, which is somewhat trivial. The main difference is in calculating the value function, which uses table interpolation both on-line and off-line. The number of points used for each table is shown in Table III for comparison.

TABLE III: Comparison of Computation Requirements

Method	Off-Line Table Size	On-Line Table Size
Extended Model	$1.2 \cdot 10^7$	$1.4 \cdot 10^4$
Instantaneous Cost	$1.3 \cdot 10^5$	$1 \cdot 10^3$

Each point in the on-line table represents a state, and each point in the off-line table represents a state, control, and disturbance combination. The scaling between off- and on-line table sizes is roughly the number of control and disturbance combinations available at each state, a factor of 126 for the Instantaneous Cost method and 882 for the Full Model

TABLE I: Comparison of Event Implementations

Option	Exponential growth in $V(x)$ solution complexity	Recompute $V(x)$ for each tuning	Optimal	Rerun simulation for each tuning	Usable for any behavior
Additional Penalties		X	X	X	
Extended Model	X	X	X	X	X
Instantaneous Cost				X	X

method. Off-line computations can be conducted on a desktop PC. The main point here is that the Instantaneous Cost method requires 2 orders of magnitude less off-line computation, but both methods have relatively simple on-line implementations that run in real-time.

VI. Summary Discussion

While we have emphasized so far the excellent fuel economy results of the SPSDP controllers in comparison to the baseline controller, we feel that the main contribution of this work is the demonstration of stochastic optimization as a viable *design method* for managing the fuel economy vs. drivability tradeoff. The controllers generated through SPSDP are directly implementable in real-time and are provably optimal. The primary engineering tasks are identifying the vehicle models (simplified and detailed), and performing the off-line computation of the stochastic dynamic programming algorithm. The off-line computations can be arduous[1], but they are straightforward and much more rapid than the traditional manual design process, where engineers spend most of their time writing rule-based controllers that require significant hand tuning with no optimality guarantees.

It is straightforward with SPSDP to generate Pareto tradeoff surfaces. SPSDP not only can identify the optimal tradeoff surface, but it directly generates controllers that operate on it. This greatly simplifies the design process by precisely quantifying tradeoffs among various attributes. Once the Pareto tradeoff surface is known, a decision can be made about where to operate the vehicle for a particular market.

The SPSDP controllers developed in this series of papers have been implemented on the prototype vehicle depicted in [1, Fig. 1], and will be the subject of a future publication. The controllers run in real time on a rapid-prototyping system, and the vehicle can be driven normally. To change vehicle behavior, it is straightforward to simply select an operating point from the Pareto surface, download the appropriate controller, and drive the vehicle again. The on-line computations are modest, consisting primarily of interpolation and minimization, and it is possible to trade storage memory for on-line computation. Changing controller designs simply involves replacing tables that are computed off-line. In addition, a designer can develop the full SPSDP controller and then reduce it to a simpler functional form. At its heart, SPSDP is a pure state feedback controller, so the feedback function can often be reduced to a simpler form with some performance loss [21], [22]. Manufacturers may use this technique not just to save on real-time computation, but to make controller behavior more transparent, directly tunable, or easier to maintain.

[1]A single instance of the controllers used here can be computed on a 2008-era laptop in a few hours.

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. When comparing architectures and components, the vehicle performance is highly dependent on the controller design, and it is difficult and time-consuming to manually tune a controller for each possible vehicle design. Effectively, the design must be conducted with limited ability to estimate the closed-loop performance of each candidate design. A method like Deterministic Dynamic Programming can automatically generate the best possible performance for a given vehicle, assuming known future drive-cycle information. Stochastic Dynamic Programming generates a causal controller that is a more reliable indicator of what is possible in practice.

A hardware test of this algorithm in the prototype vehicle is underway and will be the subject of a future publication.

VII. Conclusions

This second part of a two-part paper studies controllers generated using Shortest Path Stochastic Dynamic Programming (SPSDP) using the models and methods described in Part 1 [1]. Part 2 has focused on practical issues. Controller performance and robustness on real-world drive cycles were evaluated using a highly accurate simulation model and compared to a baseline industrial controller. The fuel economy, drivability, and battery SOC maintenance were studied on sets of 100 real-world cycles using both cumulative and statistical methods. Results show the SPSDP-based controllers yield 10% better performance than the baseline controller on real-world driving data.

The SPSDP-based controllers are robust to variations in drive cycle and the statistics used to design the controllers. This was shown by simulating about 1000 controllers designed using four different sets of drive cycle statistics on large numbers of real-world drive cycles.

Simplified metrics were developed to study the generally qualitative concept of drivability and shown to be useful even when considering more complex metrics. Future development of the SPSDP method will likely require additional control over vehicle behaviors, especially in an industrial setting. Selecting good metrics is crucial in generating the desired system behavior and implementing the behavior in SPSDP. There are several ways to incorporate a given metric in SPSDP, each with its own benefits and drawbacks.

The SPSDP design procedure is especially useful early in the design phase, during hardware selection or initial controller design. Controllers can be designed for arbitrary vehicle configurations and component sizing. The process can be highly automated.

This analysis shows that Shortest Path Stochastic Dynamic Programming is a viable method for designing real-world controllers. The controllers can be implemented directly with little manual adjustment, and generate performance comparable to the current industrial state of the art. These controllers are currently being tested in the prototype vehicle; this hardware testing will be the subject of a future publication.

References

[1] 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 1: Model, methods, and government drive cycles," *Submitted to IEEE Transactions on Control Systems Technology*, 2010.

[2] H. Larsson and E. Ericsson, "The effects of an acceleration advisory tool in vehicles for reduced fuel consumption and emissions," *Transportation Research, Part D: Transport and Environment*, no. 2, pp. 141–, 2009.

[3] S. Kamble, T. Mathew, and G. Sharma, "Development of real-world driving cycle: Case study of pune, india," *Transportation Research, Part D: Transport and Environment*, no. 2, pp. 132–, 2009.

[4] Y. Ding, "Trip-based explanatory variables for estimating vehicle fuel consumption and emission rates," *URBAN AIR QUALITY - RECENT ADVANCES, PROCEEDINGS*, pp. 61–77, 2002.

[5] E. Ericsson, "Independent driving pattern factors and their influence on fuel-use and exhaust emission factors," *Transportation Research Part D: Transport and Environment*, vol. 6, no. 5, pp. 325–345, 2001.

[6] J. Hansen, M. Winther, and S. Sorenson, "The influence of driving patterns on petrol passenger car emissions," *The Science of the Total Environment*, vol. 169, pp. 129–139, 1995.

[7] 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.

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

[9] B. Holmen and D. Niemeier, "Characterizing the effects of driver variability on real-world vehicle emissions," *Transportation Research Part D: Transport and Environment*, vol. 3, no. 2, pp. 117–128, 1998.

[10] K. S. Nesamani and K. P. Subramanian, "Impact of real-world driving characteristics on vehicular emissions," *JSME International Journal, Series B: Fluids & Thermal Engineering*, no. 1, pp. 19–, 2006.

[11] A. Esteves-Booth, T. Muneer, J. Kubie, and H. Kirby, "A review of vehicular emission models and driving cycles," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 216, no. 8, pp. 777–797, 2002.

[12] M. Ross, "Automobile fuel consumption and emissions. effects of vehicle and driving characteristics," *Annual Review of Energy and the Environment*, vol. 19, pp. –, 1994.

[13] A. Sjodin and M. Lenner, "On-road measurements of single vehicle pollutant emissions, speed and acceleration for large fleets of vehicles in different traffic environments," *The Science of the Total Environment. Vol. 169*, vol. 169, pp. –, 1995.

[14] D. LeBlanc, J. Sayer, C. Winkler, R. Ervin, S. Bogard, J. Devonshire, M. Mefford, M. Hagan, Z. Bareket, R. Goodsell, and T. Gordon, "Road departure crash warning system field operational test: Methodology and results." University of Michigan Transportation Research Institute, Tech. Rep. UMTRI-2006-9-1, June 2006, http://www-nrd.nhtsa.dot.gov/pdf/nrd-12/RDCW-Final-Report-Vol.1__JUNE.pdf.

[15] D. Opila, X. Wang, R. McGee, R. Gillespie, J. Cook, and J. Grizzle. (2010) Incorporating drivability metrics into optimal energy management strategies for hybrid vehicles - supplemental data. [Online]. Available: http://eecs.umich.edu/~grizzle/papers/auto.html

[16] E. P. Agency, "Fuel economy labeling of motor vehicles: Revisions to improve calculation of fuel economy estimates," *Federal Register*, vol. 71, no. 278, pp. 77 872–77 969, December 2006.

[17] ——. (2006, December) Epa regulatory announcement epa420-f-06-069. [Online]. Available: http://www.epa.gov/fueleconomy/420f06069.pdf

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.