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. This paper studies energy management controllers designed using Shortest Path Stochastic Dynamic Programming (SP-SDP), a stochastic optimal control design method which can respect constraints on drivetrain activity while minimizing fuel consumption for an assumed distribution of driver power demand. The performance of SP-SDP controllers is evaluated through simulation on large numbers of real-world drive cycles and compared to a baseline industrial controller provided by a major auto manufacturer. On real-world driving data, the SP-SDP-based controllers yield 10% better fuel economy than the baseline industrial controller, for the same engine and gear activity. The SP-SDP controllers are further evaluated for robustness to the drive cycle statistics used in their design. Simplified drivability metrics introduced in previous work are validated on large real-world data sets.

1 Introduction

The energy management controller for a hybrid vehicle regulates commands and operating modes for powertrain components including the engine, battery, transmission, electric motors, and brakes. While engine on-offs and gear changes are integral to saving fuel and meeting the driver’s power demand, their sequencing and frequency must be selected with great care to ensure a good driving experience. Various model-based methods for making these control decisions and tradeoffs have been introduced in the literature [1–5]. Among these methods, Shortest Path Stochastic Dynamic Programming (SP-SDP) is notable for its ability to generate causal controllers that can optimally tradeoff and regulate complex vehicle behavior and respect constraints,
while minimizing fuel consumption [6–11]. In this paper, we focus on practical considerations required to use SP-SDP in a production setting, with minimal manual tuning.

The SP-SDP technique is based on stochastic optimization and uses a cost function to indicate desired controller performance. Fuel consumption is typically included, but other attributes can be included as well, like emissions, battery wear, noise, vibration, and others. In this case, we focus the typical fuel economy and battery charge maintenance, but add the transmission shifting and engine on/off behavior, collectively termed “drivability.”

A common customer concern is that the fuel economy shown on the “window sticker” does not match the vehicle performance obtained in practice [12–16]. This strikes to the heart of the energy management controller’s sensitivity—or, alternatively, its robustness—to the assumptions made on expected vehicle driving statistics. It is common to report controller performance 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 have been tuned primarily for the test cycles, or real-world driving is fundamentally less fuel-efficient than the test cycles.

Real-world data are used in this paper to evaluate controller robustness and performance in the “off-cycle” real world. Approximately 500,000 simulations are conducted using a large number of real-world drive cycles [17]. With these data, controller performance is evaluated and optimized with respect to various classes of driver behavior [18–22] in addition to government certification cycles [6, 23–25]. The associated simulations are analyzed to determine not just mean performance, but standard deviations and 10th to 90th percentile performance. As a realistic benchmark, Ford Motor Company provided a “baseline” energy-management controller. SP-SDP and baseline controllers are compared on real-world and government test cycles in terms of fuel economy vs. drivability tradeoffs, and real-world vs. test cycle performance.

A major enabling result was the development of the drivability metrics for powertrain activity [6]. 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 useful approximations for controller design.

The methods employed here build on the theoretical framework, simulations, and hardware testing reported in previous work [6, 25]. Real-world drive data is used to extensively validate the performance of these methods in three ways: the variation of controller performance on real-world cycles, the robustness of controllers when designed for one cycle but operated on another, and the validity of the simplified drivability metrics as a useful approximation.

The paper is organized as follows: Section 2 describes the vehicle and two different simulation models, while Section 3 describes the SP-SDP and baseline industrial controllers. After a statistical comparison of real-world and government test drive cycles, Section 5 examines how various controllers (each designed for a given drive cycle) perform on real-world drive cycles. Section 6 establishes the robustness of controller performance to different drive cycles, including the dependence of engine events (a drivability metric) and fuel economy. Section 7 establishes the utility of a reduced (simplified) set of drivability metrics through comparison to a complex set of drivability metrics on a large data set.

2 Vehicle

2.1 Description

The vehicle studied in this paper is a prototype Volvo S-80 series-parallel electric hybrid and is shown schematically in Figure 1. A 2.4 L diesel internal combustion engine (ICE) is coupled to the front axle through a dual clutch 6-speed transmission. An electric machine, EM1, is directly coupled to the engine crankshaft and can generate power regardless of clutch state. A second electric machine, EM2, is directly coupled to the rear axle through a fixed gear ratio without a clutch and always rotates at a speed proportional to vehicle speed. Energy is stored in a 1.5 kWh battery pack, with a state of charge (SOC) range of [0.35, 0.65]. The system parameters are listed in Table 1.

<table>
<thead>
<tr>
<th>Table 1: Vehicle Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engine Displacement</td>
</tr>
<tr>
<td>Max Engine Power</td>
</tr>
<tr>
<td>Electric Machine Power EM1 (Front)</td>
</tr>
<tr>
<td>Electric Machine Power EM2 (Rear)</td>
</tr>
<tr>
<td>Battery Capacity</td>
</tr>
<tr>
<td>Battery Power Limit</td>
</tr>
<tr>
<td>Battery SOC Range</td>
</tr>
<tr>
<td>Vehicle Mass</td>
</tr>
</tbody>
</table>

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.

These mode definitions do not restrict the direction of power flow. The electric machines can be either motors or generators in all modes.
2.2 Operational Assumptions

Several operational assumptions were imposed based on the prototype vehicle used. Specifically, the clutch cannot slip to start the vehicle moving when stopped. Starting torque from a full stop is provided by EM not to slip to start the vehicle moving when stopped. The clutch can- not slip to start the vehicle moving when stopped. Starting torque from a full stop is provided by EM2. The clutch allows the diesel engine to be decoupled from the wheels. There are no traction control restrictions on the amount of torque that can be applied to the wheels. In terms of the controller, regenerative braking is used as much as possible up to the actuator limits, with the friction brakes providing any remaining torque.

2.3 Vehicle Models

The work presented in this paper uses two separate dynamic models to represent the same prototype hybrid vehicle; they are described below. The “control-oriented” model is quite simple and is used for controller design via dynamic programming. The “high-fidelity” model [26], provided by Ford Motor Company, is much more complex and is used for simulating drive cycles.

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.

2.3.1 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 with a 1 second sample time; very fast and very slow dynamics are ignored [6].

The battery system is represented in 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,

\[
SOC_{k+1} = \bar{\kappa}(SOC_k, \omega_{EM1_k}, \omega_{EM2_k}, T_{EM1_k}, T_{EM2_k}). \tag{1}
\]

In this simplest configuration, assuming a known vehicle speed, battery SOC is the only state variable\(^1\) required for the vehicle model.

During operation, the desired wheel torque is defined by the driver within the limits of vehicle capability. The vehicle should meet the torque demand, so the sum of the ICE and EM contributions to wheel torque must equal the demanded torque. This adds a constraint to the control optimization, reducing the 4 control inputs to 3. In parallel mode, the control inputs are Engine Torque \( T_{ICE} \), \( T_{EM1} \), and Gear.

Now, given vehicle speed, demanded road power and this choice of control inputs, the dynamics become an explicit function \( \kappa \) of the battery state SOC and the three control inputs,

\[
SOC_{k+1} = \kappa(SOC_k, T_{ICE_k}, T_{EM1_k}, Gear_k). \tag{2}
\]

In series mode, \( T_{EM1} \) is replaced with \( \omega_{EM1} \). The engine fuel consumption can be calculated from the control inputs.

2.3.2 High-Fidelity Vehicle Simulation Model

The high-fidelity model is based on MATLAB/Simulink and uses a large number of parameters and states [26]. Each subsystem in the vehicle is represented by an appropriate block with its own dynamics and low-level controllers. This model accurately represents the transient response of the engine, transmission and driveline. 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.

2.4 Driver Models

During the controller design process, the desired road torque is calculated as the exact torque required to drive the cycle at that time. When conducting simulations with the high-fidelity model, a PID controller based on velocity feedback is used to produce a desired power command and represents a causal driver.

3 Controller Design and Development

The controller design process is briefly summarized here. The interested reader should consult [6, 27] for further information.

\(^1\) As explained in [6], two additional states are required to represent the stochastic drive cycle and two more to track drivability metrics.
3.1 SP-SDP Controller

The controller is designed using Shortest Path Stochastic Dynamic Programming (SP-SDP), which, as explained in [7,9–11], is a specific formulation of Stochastic Dynamic Programming (SDP) that allows infinite horizon optimization problems to be addressed. Other algorithms to deal with infinite-horizon problems use a “discount factor” that exponentially decreases the weighting of future costs based on time. This algorithm does not use such a factor; instead, it assumes (with probability one) the trajectory will terminate in an absorbing state that incurs no future cost.

In the energy management problem, the acceleration requested by the driver, which is the equivalent of a drive cycle, is modeled as a stationary, finite-state Markov chain [28]. The controller minimizes the expected value of a cost function, which was chosen to reflect a tradeoff between fuel consumption and powertrain activity, with the latter measured by accumulated number of engine starts and gear shifts over a drive cycle.

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 and the vehicle model is accurate. In this paper, the Markov chains representing driver behavior are modeled on standard government test cycles, as in [27], and on the basis of real-world driving data, as in [27].

3.1.1 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

\[ c(x, u, w) = m_f(x, u) + \alpha I_{GE}(x, u) + \beta I_{EE}(x, u) + \phi_{SOC}(x, w), \]

where \( I(x, u) \) are indicator functions that equal one when a state and control combination produces a gear event (GE), that is a change in gear number, or engine event (EE), that is a change in engine on-off state, and are zero otherwise. The weighting factors \( \alpha \) and \( \beta \) are used to adjust drivability behavior. The SOC-based cost \( \phi_{SOC}(x, w) \) applies only at the end of the trip, when the “key-off” event occurs. The transition to key-off is captured by the stochastic drive-cycle model in the random process \( w \) [27]. The cost \( \phi_{SOC}(x, w) \) at the key-off event replaces the terminal-time cost in a finite horizon problem.

3.1.2 Control Algorithm and Implementation

The controller design process consists of two steps, one off-line and the other on-line. The majority of the computation occurs in the off-line step, which computes the solution of the optimal control problem and yields the value function \( V^*(x) \) and the optimal control \( u^*(x) \), both as a function of the state \( x \). The optimal control is a minimizer of the sum of the current cost \( c(x, u, w) \) and the expected future cost \( V^*(f(x, u, w)) \),

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

where \( w, E_w, f(x, u, w), \) and \( U \) are as defined in (3)–(5), and \( V^* \) satisfies the Bellman equation,

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

A standard iterative method of solving (7) and (8) is given in [10,11]. The state and control values are first quantized into finite grids. At each step of the iteration, the optimal control and value function are evaluated only at the grid points of the state variables, while the value function at the next time step of the dynamics, \( V^*(f(x, u, w)) \), is determined between the grid points through interpolation.

In the on-line step (the real-time controller), the output \( u^*(x) \) (7) is a nonlinear function of current state \( x \) which can be implemented using various methods that trade off computation for memory, the simplest being a look-up table. Both (7) and (8) are produced during the off-line step for all states \( x \), but explicitly storing this output is memory intensive. In our case, the pre-computed \( V^*(x) \) is stored as a look-up table for all \( x \), and the remainder of (7) is calculated on-line only for the current state.

Each set of weights used in the cost function \( c(x, u, w) \) and driving statistics \( w \) generate a different value function \( V^*(x) \), which in turn indirectly parameterizes the on-line calculation of \( u^* \) in (7). As a naming convention, a single “controller” is a function \( u^*(x) \) (7) generated by a particular set of parameters and probability distributions.
To study the effectiveness of the SP-SDP controller design methodology, two aspects of controller performance are considered. The first is the robustness of controller performance to variations in drive cycle compared to the expected probability distribution used to design the controller. The second aspect of interest is the ability to produce different drivability behavior using the tuning parameters.

To make this comparison, the controllers are grouped into “families.” A family is generated by fixing the model driving statistics and sweeping the 2 drivability penalties α and β in (6). Each family contains a few hundred individual controllers based on the same statistics, but each controller has different drivability and fuel economy characteristics because of the varying drivability penalties.

3.1.3 Relation to ECMS

Optimization-based energy management controllers for HEVs involve a tradeoff between fuel consumption and energy storage, regardless of the particular algorithm used. One of the most well known optimization methods for energy management in HEVs is the “Equivalent Consumption Minimization Strategy” (ECMS) [30].

At each time step, the controller performs an instantaneous minimization to trade off fuel consumption vs. battery usage, per

\[ u^*_k(x) = \arg\min_{u \in \Omega} [\dot{m}_f(x,u) + \lambda_k \Delta SOC(x,u)]. \]  

(9)

The design parameter is the weighting factor \( \lambda_k \), which represents the relative value of battery charge in terms of fuel. The optimal values for \( \lambda_k \) are highly cycle dependent and typically require on-line estimation; one such method is called Adaptive ECMS (A-ECMS) [31, 32].

SP-SDP and ECMS use a similar structure in (7) and (9), which is not surprising given that they are solving the same fuel minimization problem. One significant difference is the method used to estimate the relative value of battery charge at any given time. SP-SDP calculates this value based on the expected future behavior to produce \( V^*(x) \) in (8). ECMS uses the weighting factor explicitly in (9), and it may be dynamically calculated if using A-ECMS. The mathematics of this relationship are explicitly discussed in [6].

3.2 Baseline Industrial Controller

The baseline industrial energy management controller studied here is quite complex. Its key features are contained in three modules, as depicted in Fig. 2. Driver power demand is determined from pedal position to yield Wheel Power. One module determines the desired Battery Power. A second module determines the engine state based on the sum of the Wheel Power and Battery 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 Wheel Power, Battery 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.

As this is a commercial controller, the description above represents a compromise between intellectual property concerns and the desire to publish as much detail as possible.

4 Drive Cycle Data

The drive cycle data used in this paper was collected by the University of Michigan Transportation Research Institute (UMTRI) [17]. The “source” data set contains 2500 trips made by 87 drivers. Very short trips (less than 3 minutes or 0.5 km) are ignored. Two sets of 100 drive cycles are randomly selected from the UMTRI data. They are called “Ensemble 1” and “Ensemble 2.”

To gain some insight into the nature of the drive cycles, consider the characteristics of their distributions. The cumulative distribution functions (CDF) of trip distance for the source data and both subsets are shown in Fig. 3a. 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 fuel.

The cumulative distribution functions (CDFs) of vehicle speed for Ensembles 1 and 2 are shown in Fig. 3b, 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 CDFs in the figure: the Source Data, Ensemble 1, Ensemble 2, FTP, and NEDC.

There are three interesting things to notice in Figure 3b. 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 observation 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 5.1.
5 Fuel Economy Variation on an Ensemble of Drive Cycles

5.1 Method

The drive cycle data presented in Sec. 4 is used to evaluate the statistical performance of controllers across variations in drive cycle. The family of about 200 SP-SDP controllers designed with Ensemble 1 is simulated on both Ensemble 1 and Ensemble 2. Each controller now has two sets of 100 data points representing performance on individual drive cycles. The fuel economy is corrected based on the final SOC using the method discussed in the Appendix. The (non-weighted) mean, standard deviation, and 10th/90th percentile bands are calculated for corrected fuel economy and final SOC for each controller. A response surface is fitted to these data. 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.

The data are shown in Fig. 4. The horizontal axis shows the total number of engine events on the Ensemble set. The vertical axis shows fuel economy - the ratio of total distance driven to total fuel consumption, so higher numbers mean better efficiency. All fuel economy numbers are normalized to the baseline controller running FTP, which has fuel economy 1.

5.2 Discussion

The statistical analysis of the individual cycles in Fig. 4 shows very consistent performance. For the SP-SDP controllers, performance one standard deviation below the mean still exceeds the mean of the baseline controller. The SP-SDP controllers generate significantly better performance than the baseline controller on real-world cycles and are reasonably robust to variations in driving patterns.

The results across more than 40,000 simulated drive cycles demonstrate that the SP-SDP controller performance consistently exceeds that of the baseline controller despite wide variations in fuel economy on individual cycles for both controllers. In addition, the FTP and NEDC government test cycles have fuel economy about 1 standard deviation above the mean of the real world data.

The fuel economy curves in Fig. 4 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 on/off activity.

Battery charge maintenance for real-world driving is very consistent as shown in Figures 4c and 4d; 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 SP-SDP controllers also maintain tighter control of final SOC compared to the baseline controller, as evidenced by the standard deviation and percentile bands.

When considering large data sets like Ensembles 1 and 2, the cycle simulations can either be treated as individual trips or strung together as one long set of driving. Creating...
Fig. 4: Statistical fuel economy. The SP-SDP 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 10th and 90th percentile are calculated. The mean, standard deviation (error bars), and 10th and 90th percentile are also calculated for the baseline controller.

one long trip obviates the need for SOC correction because the change in battery energy is much smaller than the total fuel energy. These considerations are discussed in the appendix.

6 Robustness to Assumed Drive-Cycle Statistics: Designing with One Set, but Driving with Another

Drive cycles are used for two purposes in this paper: to design a controller, and then to evaluate its performance via simulation. SP-SDP 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 [6, 7, 28]. The statistical driver model clearly affects the final controller and the obtained closed-loop behavior, which raises a question about the relationship between the (possibly different) cycles used to design and simulate the controller.

6.1 Method

Fig. 5 shows the results of controller families that have been designed using statistics from FTP, NEDC, Ensemble 1, and Ensemble 2 and subsequently simulated when running the FTP and NEDC cycles.

The same controller families are also simulated for driving on Ensemble 1 and the results are presented in 6. Fuel economy results are corrected based on the final SOC of each individual cycle. The total fuel economy for the Ensemble is calculated using a weighted average, which is equivalent to the total distance divided by the total fuel burn. The SP-SDP data are presented for 6000 total gear events. The fuel economy is not particularly sensitive to the number of gear events; this value was selected to demonstrate that a reduction in gear
Fig. 5: Fuel Economy and Drivability Metrics on the FTP and NEDC Cycle for 5 controller options. Controller families 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. 5a.

Fig. 6: Fuel Economy on Ensemble 1 for controllers designed using statistics from FTP, NEDC, Ensemble 1, and Ensemble 2. events is possible compared to the baseline controller (12132 Gear Events) while maintaining superior fuel economy.

6.2 Discussion

This cross-evaluation of design and simulation cycles allows the study of a controller’s performance on both the cycles used to design the controller and “arbitrary” cycles. Excluding the NEDC, design cycle statistics cause less than 3.5% difference among controllers across all other cycles. The fuel economy difference when using Ensemble 1 or 2 statistics is small (less than 0.7% running Ensemble 1), which indicates that the sample sizes are large enough to represent typical driving. In general, the SP-SDP method is relatively insensitive to the choice of driver statistics 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.

This idea is reinforced in the surprising robustness of controllers when dealing with the FTP cycle. The enormous fuel economy difference between FTP (Fig. 5a) and Ensemble 1 (Fig. 6) seems to indicate that the statistics of either cycle does not represent the other. Nevertheless, controllers designed on the Ensemble cycles do very well on the FTP cycle, and the FTP-based controllers sacrifice only 3% performance on the Ensemble cycles. While the FTP and Ensemble cycles are very different, the statistics from either cycle are sufficiently representative to generate controllers that perform well on both cycles.

Figures 5 and 6 allow a comparison of performance on real-world and government test cycles. The controllers are the same in each figure and use consistent markers; they are being run on different cycles. The fuel economy normalization is the same in all figures. Both the SP-SDP 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. The baseline controller drops from a normalized fuel economy of 1 Mpg on FTP (Fig. 5a) to 0.834 Mpg on Ensemble 1 (Fig. 6). The best SP-SDP controllers achieve 1.18 Mpg on FTP, but only 0.918 Mpg on Ensemble 1. This confirms a known weakness with the government test cycles: they are not representative of real-world driving. Real-world driving yields 15-20% less fuel efficiency than the test cycles as evidenced by the drop from FTP to Ensemble 1. Recall the differences among cycles illustrated in Fig. 3b. This causes a significant mismatch between the “window sticker” fuel economy from certification testing and the fuel
Validation of Simplified Metrics for Drivability

7.1 Formulation

There are a large number of qualitative characteristics that describe powertrain behavior [2]. While metrics exist to quantify a large variety of behaviors, overall value judgements are nevertheless largely qualitative. Previous work [6, 25] introduced quantitative drivability metrics that are then simplified for use in optimization. This section provides additional validation for the simplified metrics on both government and real-world data 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 SP-SDP algorithm to maintain some control over complex behavior while keeping the problem feasible.

A primary concern in drivetrain activity is 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. 7. Many other metrics could obviously be used, but these are an important subset of the possibilities.

For the transmission, a gear “hunting” event is defined 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.

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 two simple metrics [6], which allows effective control of complex behavior with a simple implementation. 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. 7.

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 1st − 2nd shift is the same as a 1st − 3rd shift. Engaging or disengaging the clutch is not counted as a gear event, regardless of the gear before or after the event.

7.2 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. Short duration engine activity is studied in Fig. 8. The metric Engine Events is shown on the horizontal axis for both figures and compared to engine on and off Dwell Time less than X seconds for various cutoff criteria X.

The gear shifting activity of the vehicle is studied in Fig. 9. Short durations between shifts or clutch disengagements are indicated by gear dwell times less than some variable cutoff criteria and compared to the metric Gear Events.

The complex metrics are approximately monotone functions of the simple metrics as shown in Figs. 8 and 9. To highlight this property, the data are shown with a straight line least squares fit. Each cutoff time criteria is treated separately and each line matches the color shown for the underlying data.

7.3 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. This is also true for real-world driving on
the Ensemble cycles. This particular vehicle is relatively insensitive to gear activity, so engine activity is the primary concern.

The 100 cycles in Ensemble 1 are simulated sequentially and treated as a single trip of about 1600 km. The total drivability metrics are recorded for this very long trip. On and off Dwell Time less than X seconds are shown in Fig. 10. Each data point represents the same long cycle using a different controller, and variation is tied to controller tuning (i.e., choice of penalties in the cost function).

7.4 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. 10a) 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. 10). The correlation is even more clear than on FTP as expected due to the stochastic nature of the controller and larger number of sample points. 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.

8 Conclusions

Controllers generated using Shortest Path Stochastic Dynamic Programming (SP-SDP) were simulated using a highly accurate simulation model and compared to a baseline industrial controller on large numbers of real-world drive cycles. 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 SP-SDP-based controllers yield 10% better performance than the baseline controller on real-world driving data. The SP-SDP-based controllers are robust to variations in drive cycle and the statistics used to design the controllers, as demonstrated by testing about 1000 controllers designed using four different sets of drive cycle statistics on large numbers of real-world drive cycles.

Simplified metrics previously developed to study the generally qualitative concept of drivability are shown to be well-correlated with more complex metrics on real-world data. It is straightforward with SP-SDP to generate Pareto tradeoff surfaces. SP-SDP not only can identify the optimal tradeoff surface, but it directly generates controllers that op-
Fig. 10: Comparison of simple and detailed drivability metrics for the Ensemble 1 cycles. Detailed engine activity metrics are compared to the simplified Engine Events metric. The 100 cycles are treated as a single trip. Each marker represents the same drive cycle, the concatenated Ensemble 1.

These results provide statistical validation that controllers of this type function well across a wide range of real-world drive cycles and driving styles. The data presented represents more than 4000 controllers and 500,000 simulated cycles (most in Figs. 6 and 4), 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.

References


APPENDIX

Grouping Drive Cycles

When studying behavior on multiple drive cycles, there are at least two ways to study performance in the aggregate.

Individual Cycles - Each cycle is studied individually, and the starting SOC for each trip is 0.5.

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.

The data shown in the body of the paper uses Individual Cycles, where 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 (Section 5) and allows deeper understanding than simply studying the weighted average of fuel consumption.

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 below)
Fuel Economy Correction based on SOC

The final SOC is not guaranteed to exactly match the starting SOC on any given cycle. This could yield false fuel economy results, so all fuel economy estimates are corrected based on the final SOC of the drive cycle [6]. 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

$$\Delta m_f = C_{Batt} \Delta SOC \frac{BSFC_{min}}{\eta_{Regen}}$$ (10)

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 $\eta_{Regen}$ 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.

The performance of the SP-SDP 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. Figs. 6 and 11 demonstrate 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.

Fig. 6 showed Ensemble 1 calculated as Individual Cycles, and it is now treated as a Concatenated Cycle representing a single vehicle with fuel economy and drivability results shown in Fig. 11. The figure shows the same five controller sets evaluated on Ensemble 1: the baseline controller and 4 SP-SDP controller families designed on statistics from FTP, NEDC, Ensemble 1 and Ensemble 2. All fuel economy numbers are normalized to the baseline controller running FTP, which has fuel economy 1.

Fig. 11: Fuel Economy on Ensemble 1 treated as a concatenated cycle for controllers designed using statistics from FTP, NEDC, Ensemble 1, and Ensemble 2.