FUNDAMENTAL STRUCTURAL LIMITATIONS OF AN INDUSTRIAL ENERGY MANAGEMENT CONTROLLER ARCHITECTURE FOR HYBRID VEHICLES

Daniel F. Opila[*]
Dept. of Mechanical Engineering
University of Michigan
Ann Arbor, Michigan 48109-2125
Email: dopila@umich.edu

Xiaoyong Wang
Ryan McGee
Ford Motor Company
Dearborn, Michigan 48120

Jeffrey A. Cook
J.W. Grizzle
Department of EECS
University of Michigan
Ann Arbor, Michigan 48109-2122

ABSTRACT

Energy management controllers for hybrid electric vehicles typically contain numerous parameters that must be tuned in order to arrive at a desired compromise among competing attributes, such as fuel economy and driving quality. This paper estimates the Pareto tradeoff curve of fuel economy versus driving quality for a baseline industrial controller, and compares it to the Pareto tradeoff curve of an energy management controller based on Shortest Path Stochastic Dynamic Programming (SPSDP). Previous work had shown important performance advantages of the SPSDP controller in comparison to the baseline industrial controller. Because the baseline industrial controller relies on manual tuning, there was always the possibility that better calibration of the algorithm could significantly improve its performance. To investigate this, a numerical search of possible controller calibrations is conducted to determine the best possible performance of the baseline industrial controller and estimate its Pareto tradeoff curve. Both the SPSDP and baseline controllers are causal (i.e., do not rely on future drive cycle information). The SPSDP controllers achieve better performance (i.e., better fuel economy with equal or better driving quality) over a wide range of driving cycles due to fundamental structural limitations of the baseline controller that can not be overcome by tuning. The message here is that any decisions that restrict controller structure may limit attainable performance, even when many tunable parameters are made available to calibration engineers. The structure of the baseline algorithm and possible sources of its limitations are discussed.

1 Introduction

Hybrid electric vehicle (HEV) energy management controllers have received a lot of interest in both academic and industrial circles [1]. While many design methods have been proposed, it is difficult to compare them. Most algorithms, even those from academia based on formal optimization methods, have at least some parameters that must be selected by the designer. This is even more true of industrial controllers, which tend to use extensive hand-tuning and in-vehicle calibration in order to trade off what are often very subjective driving quality attributes.

Any performance comparison of controller design methods is only as good as the engineers that tune the various algorithms, and thus the comparison always suffers from the refrain that “algorithm X could have been tuned better.” Comparisons are even more difficult when the designer is forced to compromise among competing performance attributes, such as the tradeoff between fuel economy and engine start-stop activity, which is investigated here. The relative value of one characteristic compared to another is highly subjective, meaning comparisons among different operating points necessitate a qualitative value judgement.

The goal of this paper is to study the performance of the industrial controller introduced in [2] as the baseline energy management system for a prototype HEV. Its performance is compared to a causal academic controller based on stochastic optimization, namely Shortest Path Stochastic Dynamic Programming (SPSDP). The baseline industrial controller is first evaluated to determine its Pareto tradeoff curve of best possible performance in terms of fuel economy versus engine activity. This is accomplished by sweeping the parameters of the controller over a wide range of values, thereby generating a point cloud of pos-

[*] Address all correspondence to this author.

sible fuel economy and engine start-stop operating points of the prototype HEV under this controller. The frontier of this point cloud is the Pareto tradeoff curve of maximum attainable performance; the HEV with this controller can be operated anywhere on that line, but not above it. A comparison is then made to the Pareto tradeoff curves of the SPSDP studied in [2, 3].

The method of SPSDP generates causal controllers that are directly implementable in a real-time setting [4]. In particular, the resulting controllers do not use *future* drive cycle information. This is in contrast to Deterministic Dynamic Programming [5], which is cycle dependent (it relies on *a priori* knowledge of the entire drive cycle). The causal nature of a SPSDP controller allows a fair comparison to the baseline controller.

The Pareto frontier for the SPSDP controllers are shown to lie above the Pareto frontier of the baseline controller, meaning that the SPSDP controllers achieve superior fuel economy performance for a given level of engine on-off activity, for *any* possible tuning of the baseline controller. This limitation is fundamental to the structure of the baseline algorithm: no amount of parameter tuning or calibration can generate performance that equals that of the SPSDP algorithm. An advantage of the SPSDP algorithm is that it directly generates controllers that lie on the tradeoff curve, and does so without requiring hand calibration. The role of expert judgement is then to decide where on the Pareto tradeoff curve to operate the vehicle for a given market.

Traditional vehicle software is produced through a process of continuous improvement. While each year's model has better vehicle control software than the last, in practice, control design engineers are hesitant to change the basic structure of the energy management algorithms, both because of their inherent complexity as well as their complex relationship to other vehicle systems. Instead, if a better controller is developed, its actions are analyzed in detail, and the existing software is tuned to mimic the actions of the new controller. This paper emphasizes that such an approach will not always work. While it is possible that a given controller architecture may be tuned for a particular vehicle to achieve the maximum performance, there are no guarantees. When manually tuning an algorithm, engineers may be unaware they are finding the maximum attainable performance for a *particular controller architecture* rather than the optimal causal controller. A more general benchmark that avoids specifying a controller architecture is required to correctly gauge performance. SPSDP is one such method for generating causal controllers.

The remainder of the paper is structured as follows. The vehicle model and drivability metrics are summarized in Sections 2 and 3, respectively; these are similar to previous work in [2, 3] and are included for the convenience of the reader. The architecture of the baseline industrial energy management controller and the tuning methods used are discussed in Section 4. The academic controller against which it is bench marked, SPSDP, is briefly described in Section 5, with details relegated to Appendix A. The main contribution of the paper, the careful comparison of an industrial state of the art controller to SPSDP through Pareto tradeoff curves, is presented in Sections 6 and 7.

Figure 1. The prototype hybrid: a modified Volvo S-80

2 Vehicle

2.1 Vehicle Architecture

The vehicle model studied in this paper is a prototype Volvo S-80 series-parallel electric hybrid and is shown in Figures 1 and 2. A 2.4 L diesel engine is coupled to the front axle through a clutched 6-speed automated manual 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, therefore the electric machine is always rotating at a speed proportional to vehicle speed. Energy is stored in a 1.5 kWh battery pack. The system parameters are listed in Table 1.

Table 1. Vehicle Parameters

Engine Displacement	2.4 L
Max Engine Power	120 kW
Electric Machine Power $EM1$ (Front)	15 kW
Electric Machine Power $EM2$ (Rear)	35 kW
Battery Capacity	1.5 kWh
Battery Power Limit	34 kW
Vehicle Mass	1895 kg

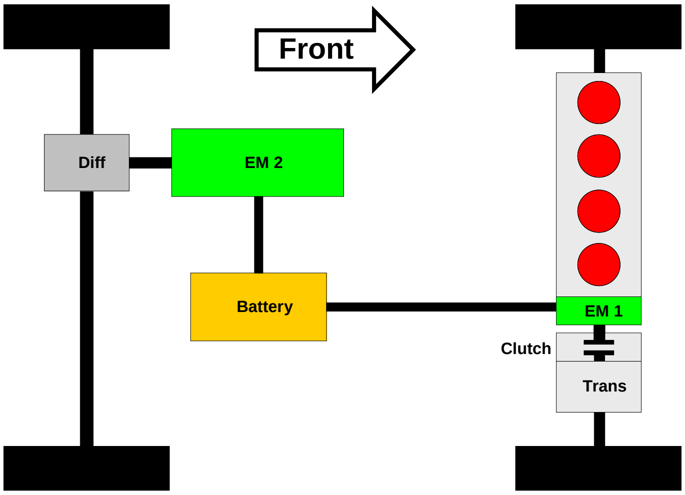

Figure 2. Vehicle Configuration

2.2 Vehicle Models

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

The second model comes from Ford Motor Company and uses its in-house modeling architecture. This sophisticated model is used to test fuel economy and controller behavior by simulating controllers on drive cycles. This model is referred to as the "vehicle simulation" model in this paper [6].

This combination of models allows the controller to be designed on a simple model and keeps the problem feasible, while providing accurate fuel economy results on a complex model.

2.3 Control Model

When using Shortest-Path Stochastic Dynamic Programming, the off-line computation cost is very sensitive to the number of system states. For this reason, the model used to develop the controller must be as simple as possible. The vehicle model used here contains the minimum functionality required to model the vehicle behavior of interest on a second-by-second basis. Dynamics much faster than the sample time of 1s are ignored. Long-term transients that only weakly affect performance are also ignored; coolant temperature is one example.

The vehicle hardware allows three main operating conditions:

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

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

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

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

The automated manual transmission has discrete gears and no torque converter. The transmission is modeled with a constant mechanical efficiency of 0.95. Transmission gear shifts are allowed every time step (1s) and transmission dynamics are assumed negligible. When the clutch is engaged, the vehicle is in **Parallel Mode** and the engine speed is assumed directly proportional to wheel speed based on the current transmission gear ratio R_g,

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

The electric machine $EM1$ is directly coupled to the crankshaft, and thus rotates at the engine speed ω_{ICE},

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

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

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

The clutch can be disengaged at any time, and power is delivered to the road through the rear electric machine $EM2$. This condition is treated as the "neutral" gear 0, which combines with the 6 standard gears for a total of 7 gear states. If the engine is on with the clutch disengaged, the vehicle is in **Series Mode**. The engine-$EM1$ combination acts as a generator and can operate at arbitrary torque and speed. The $EM1$ command is a speed rather than a torque in **Series Mode**. If the engine is off while the clutch is disengaged, the vehicle is in **Electric Mode**.

The battery system is similarly reduced to a table lookup form. The electrical dynamics due to the motor, battery, and power electronics are assumed sufficiently fast to be ignored. The energy losses in these components can be grouped together

Table 2. Vehicle Mode Definitions.

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

such that the change in battery State of Charge (SOC) is a function $\bar{\kappa}$ of Electric Machine speeds ω_{EM1} and ω_{EM2}, torque T_{EM1} and T_{EM2}, and battery SOC at the present time step,

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

Assuming a known vehicle speed, the only state variable required for this vehicle model is battery SOC. Changes in battery performance due to temperature, age, and wear are ignored. An additional constant power drain is used to represent accessory loads like radios and headlights, as well as additional losses.

During operation, the desired wheel torque is defined by the driver. If we assume the vehicle must meet the torque demand perfectly, then the sum of the ICE and EM contributions to wheel torque must equal the demanded torque T_{demand},

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

This adds a constraint to the control optimization, reducing the 4 control inputs to a 3 degree of freedom problem. In **Parallel Mode** the control inputs are *Engine Torque*, *EM*1 *Torque*, and *Transmission Gear*. In **Series Mode**, the electric machine command becomes *EM*1 *Speed*.

Simulation is conducted assuming a "perfect" driver. At each time step, the vehicle velocity is the desired cycle velocity. The desired road power is calculated as the exact power required to drive the cycle at that time, and is a function of the desired velocity profile. Now, given vehicle speed, demanded road power, and this choice of control inputs, the dynamics become an explicit function κ of the state *Battery SOC* and the three control choices as shown in Table 3,

Table 3. Vehicle Dynamic Model

State	Control Inputs
Battery Chg. (SOC)	Engine Torque
	*EM*1 Tq. (Parallel) or Speed (Series)
	Transmission Gear

$$SOC_{k+1} = \kappa(SOC_k, T_{ICE}, T_{EM1}, Gear). \tag{2}$$

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

Operational Assumptions:

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

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

2.4 Vehicle Simulation Model

As part of this project, Ford provided an in-house model used to simulate fuel economy. It is a complex, MATLAB/Simulink based model with a large number of parameters and states [6]. Every individual subsystem in the vehicle is represented by an appropriate block. For each new vehicle, subsystems are combined appropriately to yield a complete system.

The control-oriented model of Section 2.3 is very closely matched to the vehicle simulation model is nearly identical in the parameters: efficiencies, mass, drag, power limits, etc. The control-oriented model has a limited number of states and thus cannot fully match the simulation model.

The vehicle simulation model contains the baseline controller algorithm. To generate simulation results using this controller, the controller parameters are adjusted and a target drive cycle is provided to the model. The baseline controller does not use **Series Mode**, although the plant model allows it.

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

3 Drivability Constraints

3.1 Motivation

Drivability is a commonly used term that covers many aspects of vehicle performance including acceleration, engine noise, braking, shifting activity, shift quality [7], and other behaviors. All of these contribute to consumer perception of the vehicle, which is crucial in purchasing decisions. This research addresses only the hybrid vehicle drivability issues of gear selection and when to start or stop the internal combustion engine.

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

In this paper we investigate the usefulness of optimizing for fuel economy and drivability simultaneously. By including these real-world concerns, one can generate controllers that improve performance and are one step closer to being directly implementable in production. Specifically, these results validate the real-world performance of the SPSDP algorithm and compare it to the best possible performance of an industrial controller.

For this particular vehicle configuration, the fuel economy is much more sensitive to engine activity than transmission activity. For this reason, the results are shown only for engine activity, although the controller design and implementation also includes gear selection as in [2,3].

3.2 Chosen Penalties

In the context of the overall system, two significant characteristics that are noticeable to the driver are the basic behaviors of the transmission and engine. These are included in both vehicle models presented in Section 2. To effectively design controllers, qualitative drivability requirements must be transformed into quantitative restrictions or metrics. Drivability experts at Ford Motor Company were consulted to assist in developing numerical drivability criteria. Two baseline metrics are used to quantify behavior for a particular trip. The first is *Gear Events*, the total number of shift events on a given trip. The second metric is *Engine Events*, the total number of engine start and stop events on a trip.

By definition, engine starts and stops are each counted as an event. Each shift is counted as a gear event. In this paper, the transmission is constrained to one step shifts (i.e. $1^{st} - 2^{nd}$) to match the transmission restrictions of the baseline controller. Shifts that occur with the clutch disengaged are not counted. Engaging or disengaging the clutch is not counted as a gear event, regardless of the gear before or after the event.

Despite the relative simplicity of these metrics, simulations have shown that they capture a wide range of vehicle behavior and are well correlated with more complicated metrics.

4 Baseline Controller

4.1 Architecture

The "baseline" prototype controller architecture studied here is commonly used for energy management. It is obviously quite complex, but the critical energy management features are described here and shown in Figure 3. One module determines the optimal battery power flow and adds it to the driver demand to determine the *Total Power*. A second module determines the optimal engine state based on the *Total Power* using a state machine with hysteresis. A third module then determines individual actuator commands based on the *Total Power* and the desired engine state. The transmission gear is selected independently by the transmission.

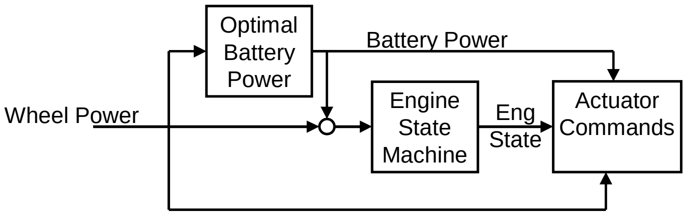

Figure 3. High Level Baseline Controller Architecture.

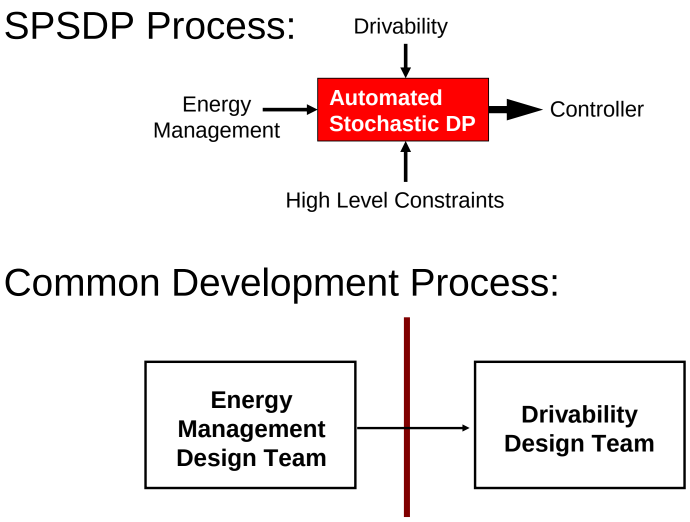

Figure 4. Two possible design processes. The SPSDP process conducts the optimization in one step, but adds complexity. The "Common" two stage optimization is often used in industry. It is easier to tune but may sacrifice some performance.

This architecture is fundamentally different from the SPSDP algorithm discussed in Section 5. These two choices are represented in Figure 4. The SPSDP controller is a single step optimization, while the baseline algorithm mimics the common two step design procedure. Structurally, the two-stage algorithm is similar to the "local" optimization discussed in [3,4].

The fundamental limitations of the baseline controller likely arise from three sources. The first is structural; the optimal battery charging power and engine state are determined sequentially and not simultaneously. Other major automakers use similar two-stage architectures that likely exhibit these limitations. The second possible source is that the engine state machine is inherently rule-based as a function of total power demand. While the total power demand is strongly correlated with optimal trajectories, a rule-based strategy is likely suboptimal for the metrics considered here because it imposes an assumed structure on the controller. The third possible source of limitations is the "actuator commands" block, which is somewhat rule-based and not a pure optimization.

4.2 Performance Capability

The flexibility of rule-based controllers with many calibration parameters is tempered by the fact that there is no guarantee of optimality. Furthermore, it is difficult to compare controller architectures since it is very hard to assess whether or not a given calibration might be improved by judicious tuning [2].

The goal of this work is to determine the best case performance of the baseline architecture and compare it to other available architectures. By simulating many possible tunings of the baseline controller, the best case performance can be estimated. The best case performance is useful information in and of itself, but also allows rigorous comparisons among multiple control methods. Performance can be defined not only as fuel economy, but to quantify additional performance tradeoffs. This paper studies engine activity, but other important tradeoffs could easily be considered.

The main tuning parameters available are six functions of speed, three in each of the "Optimal Battery Power" and "Engine State Machine" blocks. This is obviously a very large space to search, especially for an engineer tuning the algorithm by hand. One advantage of this architecture is that the engine behavior and battery charge maintenance are largely confined to their respective blocks with minor crosstalk, simplifying the tuning process considerably.

There are three tuning parameters in the Engine State Machine. These parameters are functions of speed, yielding three tuning functions that can be varied.

4.3 Search Methods

Initially, proposed controller tunings are simulated on the FTP cycle. The fuel economy numbers are corrected based on final SOC, and the number of *Engine Events* is recorded as discussed in Section 3. Controllers are evaluated based on both fuel economy and engine activity.

First, the three functions the in Engine State Machine are varied, using both small perturbations from the nominal tuning and a brute force sweep of a larger function space. This generates about 100,000 possible controllers. A few hundred of the best tunings are randomly selected for further study. For each of these tunings, the engine state machine parameters are fixed and the three functions in the battery power block are varied. This yields the base set of 180,000 controllers.

To evaluate robustness to real-world driving, 210 of these controllers are selected and extensively evaluated using real-world drive cycles.

5 Academic Benchmark

In order to evaluate the performance of the industrial algorithm, an academic algorithm is used to evaluate state-of-the-art performance for the same information conditions. Controllers must be causal, implementable, and use no future information. The method chosen is Shortest Path Stochastic Dynamic Programming (SPSDP), a well-established controller design method that has been previously used for hybrid vehicle energy management [4]. This method is not the focus of this paper, merely a benchmark for comparison. The details of the algorithm are included as Appendix A as well as in previously published work [2, 3].

SPSDP is an automated algorithm that generates a causal controller from a vehicle model, a cost function, and statistics about typical driving. The resulting controller can be directly implemented in real-time. The controller is provably optimal for the information given: statistics about general driving behavior without direct future knowledge. The designer can specify the cost function, in this case fuel and engine activity. The resulting controllers are optimal for a given cost function. The designer controls the assigned *cost*, and the algorithm produces *behaviors*. This is different from a manual algorithm where the designer specifies behaviors in an attempt to minimize cost, with no optimality guarantees. In practice, controllers are generated with a variety of penalty values in the cost function, and the resulting behaviors generate a continuous curve. The designer then picks the desired behavior.

The SPSDP algorithm has several advantages, but suffers from off-line complexity. While straightforward, it is not easy to set up the off-line optimization to generate the real-time controllers. The off-line step is also computationally intensive, requiring several hours to compute a controller. In effect, the designer is eliminating the need to decide on a controller architecture and tuning, but it comes at the cost of additional setup.

6 Simulation Procedure

Both baseline and SPSDP controllers are simulated on the same vehicle simulation model discussed in Section 2.4. These simulations are all causal, so the final SOC is not guaranteed to exactly match the starting SOC. This could yield false fuel economy results, so all fuel economy results are corrected based on the final SOC of the drive cycle. This is done by estimating the additional fuel required to charge the battery to its initial SOC, or the potential fuel savings shown by a final SOC that is higher than the starting level. This correction is applied according to

$$\Delta Fuel = C_{Batt}\Delta SOC\frac{BSFC_{min}}{\eta_{max}^{Regen}} \tag{3}$$

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

Controllers are initially evaluated on the FTP government test cycle, in which case there is only one simulation per controller. To study robustness to drive cycle variations, controllers are also simulated on a set of real-world driving data collected by the University of Michigan Transportation Research Institute

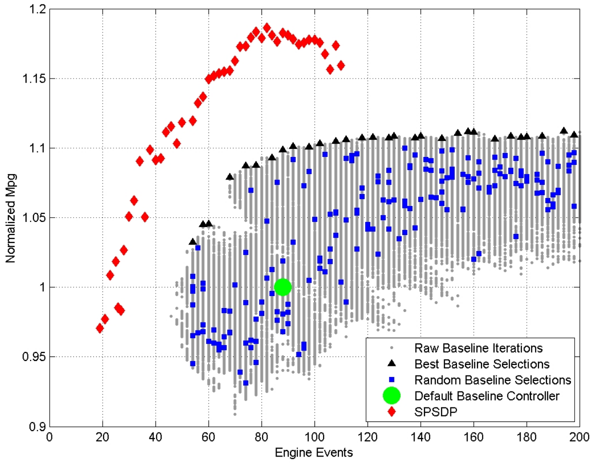

Figure 5. Best Case performance of the baseline controller running FTP compared to SPSDP controllers. The black triangles are the "best" available tunings, and the blue squares are selected randomly from the other reasonable controllers. SPSDP controllers are shown for comparison as red diamonds. Fuel economy is normalized to the default baseline controller running the FTP cycle. All markers represent the same controllers in Figures 5-7.

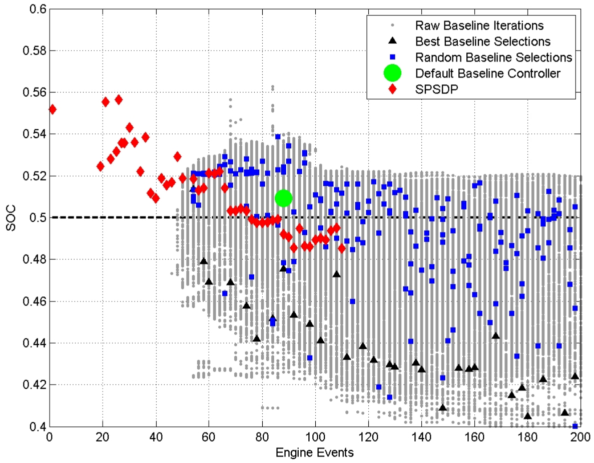

Figure 6. Final Battery SOC of the baseline and SPSDP controllers running FTP. All cycles start at SOC=0.5. The gray dots are all possible baseline controllers, the black triangles are the "best" available baseline controllers, and the blue squares are selected randomly from the other reasonable baseline controllers. SPSDP controllers are shown as red diamonds. All markers represent the same controllers in Figures 5-7.

(UMTRI) [8]. 100 cycles are randomly selected from these data to generate a set of *ensemble* cycles. Procedurally, this is conducted as follows:

1. Each controller is simulated on each of the 100 cycles in the ensemble using the vehicle simulation model.
2. The results for the ensemble set of 100 cycles are compiled to generate average or cumulative performance for that particular controller.

In the end result, each controller has average performance metrics (fuel economy and drivability) representing cumulative performance on the set of ensemble cycles. Note that studying 100 controllers on 100 cycles each means 10,000 simulations.

7 Results

As discussed in Section 4.3, the base set of 180,000 possible controller tunings is first simulated on the FTP cycle. The fuel economy numbers are corrected based on final SOC, and the number of *Engine Events* is recorded as discussed in Section 3. These results are shown in Figure 5 as small gray dots. Many of the parameter values yield unreasonable controllers with poor fuel economy and large numbers of engine events. They are outside the bounds of the figure.

The default tuning of the baseline controller (provided by Ford) is shown as a large green circle. The controllers designed using SPSDP are shown as red diamonds. One major benefit of SPSDP is clearly visible: controllers are always on the frontier of attainable performance without iterative searching. Varying the cost function merely moves the operating point along the Pareto curve of maximum performance. The SPSDP controllers achieve equal or better performance than the baseline under all conditions, as would be expected with the optimality guarantees.

These initial results are for a single drive cycle. Real-world performance is studied by simulating the controllers on a group of 100 drive cycles. It is impractical to simulate all the controllers on 100 cycles each, so the majority of the brute-force search is conducted on the FTP cycle. 210 controllers are then selected for further testing. 30 of those are selected from the frontier as the "best" available, and the remainder are selected randomly from the cloud of reasonable controllers. The selected controllers are shown as black triangles and blue squares respectively in Figure 5.

All results in this paper are normalized to the default baseline controller running the FTP cycle. All markers in Figures 5-7 represent the same controllers running various cycles. The fundamental tradeoff between vehicle fuel economy and the amount of engine activity is clearly visible in the results.

Varying the engine state machine parameters does change the battery SOC behavior, but the controllers are still reasonably charge-sustaining, as shown in Figure 6. These changes in battery SOC are used to correct the cycle fuel economy for all results shown. This correction generally only changes the results 1-2% and does not alter the relative comparison. The SPSDP controllers generally achieve better performance than the baseline both in uncorrected fuel economy and in final SOC. All markers represent the same controllers as shown in Figure 5.

Fuel economy on government test cycles differs from that of

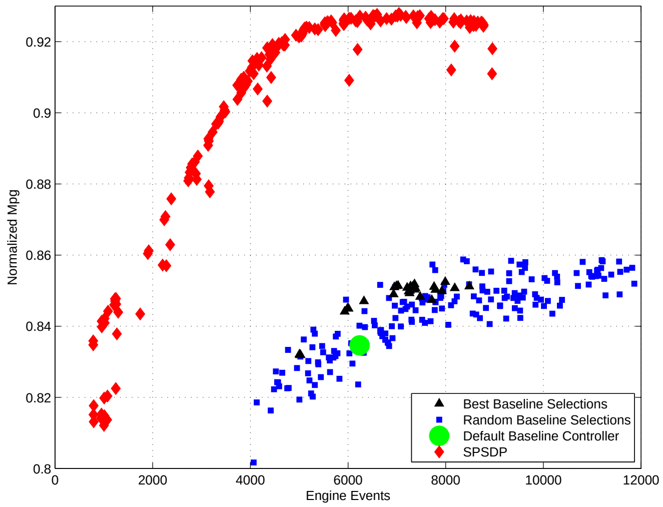

Figure 7. Best Case performance of the baseline controller and SPSDP running the ensemble set of 100 cycles. Fuel economy is normalized to the default baseline controller running the FTP cycle. All markers represent the same controllers in Figures 5-7.

real-world driving, so each controller is also evaluated on an ensemble set of 100 drive cycles. 210 of the baseline controllers are selected and compared to the SPSDP controllers in Figure 7. The results are normalized to the baseline controller running the FTP cycle (Figure 5), so both the SPSDP and baseline controller yield lower fuel economy in the real-world(0.86-0.93) than on government test cycles (1.1-1.18). The SPSDP controllers in Figures 5-7 are the same for all three figures, and are designed using probabilities from the set of 100 real-world drive cycles.

While it is not easy to specify why the SPSDP controllers perform better, in general they are more aggressive and efficient in their use of the ICE and the electric machines. The ICE operates largely in a "bang-bang" fashion, either at a high efficiency operating point or completely off. The electric machines are generally used closer to their maximum efficiency, or near maximum power to enable high ICE power outputs when little road load power is required.

These general operating principles seem intuitive, but the result is one of the major benefits of SPSDP: it automatically generates the optimal controller without a designer specifying control actions. Even given these principles, a designer would be hard-pressed to write control laws that generate optimal performance. These principles also do not necessarily hold in general and may change with different vehicles. Guessing the wrong "rules of thumb" in the design phase can impose performance limits, as demonstrated in this paper.

8 Conclusions

A baseline industrial energy management algorithm is extensively tuned to achieve its maximum performance, but it falls short of another causal controller design method, SPSDP. There is no possible tuning or calibration of the baseline algorithm that can match the SPSDP controller performance. This implies fundamental structural limitations of the baseline algorithm. These limitations likely arise for three reasons: the battery power flows and engine start-stops are determined sequentially and not simultaneously, the engine on-off control is constrained to be a function of total power demand, and some actuator selection is rule-based.

The SPSDP-based controllers do not exhibit similar limitations. In particular, a SPSDP-based controller uses full-state feedback, and thus power flows, engine on-off events and gear number can be general functions of vehicle speed, battery SOC, gear number, engine state and total power demand. While it is very possible that a simpler feedback structure may exist, that is, one that depends on fewer variables and hence is more easily calibrated in the field, the search for such a feedback is a separate problem. As part of that search, the control designer has to decide how much degradation in performance is acceptable for ease of tunability, maintenance, or other considerations.

The work presented here underlines the point that making an *a priori* choice of feedback structure or vehicle behavior can induce significant structural barriers to obtaining optimal vehicle performance, barriers that cannot be overcome at later stages in the design process, no matter how well the nominal controller is tuned. One way to avoid making these choices at an early stage is to adopt a more sophisticated controller design procedure in the prototyping phase, one that automatically searchers over all possible state feedback controllers. One such method is SPSDP.

Acknowledgements: This material is based upon work supported under a National Science Foundation Graduate Research Fellowship and a grant from Ford Motor Company. Daniel Opila is supported by NDSEG and NSF-GRFP fellowships. Special thanks to the University of Michigan Transportation Research Institute (UMTRI) for providing drive cycle data.

REFERENCES

[1] Sciarretta, A., and Guzzella, L., 2007. "Control of hybrid electric vehicles". *IEEE Control Systems Magazine,* **27**(2), pp. 60–70.

[2] Opila, D., Wang, X., McGee, R., Cook, J., and Grizzle, J., 2009. "Performance comparison of hybrid vehicle energy management controllers on real-world drive cycle data". In Proceedings of the American Control Conference. Pre-print available: http://www.eecs.umich.edu/ grizzle/papers/auto.html.

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

[4] Tate, E., Grizzle, J., and Peng, H., 2008. "Shortest path stochastic control for hybrid electric vehicles". *International*

Journal of Robust and Nonlinear Control, **18**, pp. 1409–1429.

[5] Lin, C.-C., Peng, H., Grizzle, J., and Kang, J.-M., 2003. "Power management strategy for a parallel hybrid electric truck". *IEEE Transactions on Control Systems Technology,* **11**(6), pp. 839–849.

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

[7] Pisu, P., Koprubasi, K., and Rizzoni, G., 2005. "Energy management and drivability control problems for hybrid electric vehicles". In Proceedings of the European Control Conference Decision and Control CDC-ECC.

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

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

APPENDIX

A Shortest Path Stochastic Dynamic Programming

A.1 Cost Function

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

$$\begin{gathered}\min \textstyle\sum_0^T Fuel\,flow \\ \text{such that} \\ \textstyle\sum_0^T GE \le GE_{max}, \sum_0^T EE \le EE_{max}\end{gathered} \tag{4}$$

where GE and EE are the number of Gear and Engine Events respectively as described in Section 3, and GE_{max} and EE_{max} are the maximum allowable number of events on a cycle.

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

Controllers based only on fuel economy and drivability completely drain the battery as they seek to minimize fuel. An additional cost is added to ensure that the vehicle is charge sustaining over the cycle. This SOC-based cost only occurs during the transition to key-off, so it is represented as a function $\phi_{SOC}(x)$ of the state x, which includes SOC [2–4]. The performance index for a given drive cycle is

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

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

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

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

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

A.2 Problem Formulation

To determine the optimal control strategy for this vehicle, the Shortest Path Stochastic Dynamic Programming (SPSDP) algorithm is used [4, 9]. This method directly generates a causal controller; characteristics of the future driving behavior are specified via a Markov chain rather than exact future knowledge. The system model is formulated as

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

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

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

where $c(x,u)$ is the instantaneous cost as a function of state and control; (6) is a typical example. The optimal control u^* is a control that achieves the minimum cost $V^*(x)$. This equation represents a compromise between minimizing the current cost

$c(x,u)$ and the expected future cost $V(f(x,u,w))$. Note that the cost $V(x)$ is a function of the state only. This cost is finite for all x if every point in the state space has a positive probability of eventually transitioning to an absorbing state that incurs zero cost from that time onward. Equation (7) is solved using modified policy iteration, which is one of several available solution methods.

In order to use this method, the driver demand is modeled as a Markov chain. This "driver" is assigned two states: current velocity v_k and current acceleration a_k, which are included in the full system state x. A probability distribution is then assigned to the set of accelerations at the next time step based on drive cycles that represent typical driving behavior [2–4]. This choice of typical drive cycles does change the controller that is generated, but the algorithm is robust to a wide range of probability distributions as shown in [2].

In addition to fuel economy, it is desirable to study the drivability characteristics of the vehicle. The metrics chosen are gear shifts and engine events as described in Section 3. To track these metrics, two additional states are required: the *Current Gear* (0-6) and *Engine State* (on or off).

Bringing this all together, the full system state vector x contains five states: one state for the vehicle (*Battery SOC*), two states for the stochastic driver (v_k, a_k), and two states to study drivability (*Current Gear* and *Engine State*). This formulation is termed the "SPSDP-Drivability" controller. A summary of system states is shown in Table 4. The control u contains the three inputs *Engine Torque*, *EM*1 *Torque/Speed*, and *Transmission Gear*, as described in Section 2 and Table 3.

Table 4. Vehicle Model States

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