

Optimal Energy Management and Storage Sizing for Electric Vehicles

A. DESHPANDE, J. TAYLOR
University of Toronto
Canada

SUMMARY

Combining storages with different characteristics can improve the performance and lifetime of electric vehicles. For example, a supercapacitor and a battery together can handle large power transfers from acceleration and regenerative braking while protecting the battery from degradation. In this paper, we use approximate dynamic programming to design a policy for power sharing between dual storage devices. We write the dynamic program as a linear program and use basis functions to approximate the optimal value function. Numerical results show that the resulting suboptimal policy can approximate the optimal policy with low error given a sufficient number of basis functions.

KEYWORDS

Electric vehicles, approximate dynamic programming, batteries, supercapacitors.

I. Introduction

Battery life is an important consideration in electric vehicle (EV) operation. An aging battery experiences a drop in energy capacity [1], which reduces the maximum distance that can be travelled on a single charge. In vehicle-to-grid applications, it also reduces the useful service that can be provided by the battery [2], as a battery of decreased capacity is able to store less energy. Given that batteries are expensive components, there is a significant financial cost of degradation which must be minimized.

Combining storages with different characteristics can improve the performance and lifetime of electric vehicles. The battery degrades if operated with large energy transfers, whereas a supercapacitor has a high power density but low energy capacity [3,4]. This means that a supercapacitor and a battery together can handle large power transfers from acceleration and regenerative braking while protecting the battery from degradation [4]. Such a system is called a hybrid energy storage system (HESS).

Many previous works have approached the problem of controlling the HESS — that is to say, determining energy discharges from the battery and supercapacitor to satisfy a sequence of vehicle energy demands while minimizing abovementioned costs. For example, [3,7] present heuristic rule-based approaches for limiting battery degradation that do not use demand forecasting to account for long-term costs, while [5,6] do forecast but solve the optimization problems using suboptimal greedy and MPC techniques. For optimality, dynamic programming (DP) has previously been applied in this context to determine a policy, i.e. a state feedback control function. References [8] and [9], for instance, have both solved the dual-storage problem via infinite horizon DP, but without addressing the curse of dimensionality inherent to DP.

Our approach differs from these papers because we allow for flexibly approximating the optimal policy and solve the problem offline, which resolves the issue of computation time. We first model the problem as inventory control with two storages and formulate it as an infinite horizon DP. We then solve it using linear programming (see, e.g., [10, pg. 51]) to make use of fast solution techniques for linear programs. Finally, we approximate the solution to the linear program using the basis function technique given in [11], after generating basis functions for the approximation using standard techniques such as in [12].

In this paper, we physically model the HESS in Section II and pose the infinite horizon problem in Section III. In Section IV, we describe the basis functions we use to approximate the value function. Finally, in Section V, we show numerical results for a vehicle with a battery and a supercapacitor.

II. Model

In this section, we model the two-storage device optimization problem. Let $E_h(t)$ and $U_h(t)$ denote the energy stored in and energy transferred from device h at time instant t , respectively. Here $h = 1$ refers to the battery and $h = 2$ refers to the supercapacitor. We define $U_h(t)$ to be positive when the device is discharged and negative when the device is charged. Furthermore, let $L(t)$ be the energy demand imposed on the system. We then define the state to be $x(t) = [E_1(t), E_2(t), L(t)]^T$.

In this problem, the HESS must be controlled to satisfy the known demand at all times. This imposes the constraint $U_1(t) + U_2(t) = L(t)$. We define the control to be $u(t) = U_1(t)$; $U_2(t)$ is not part of the control because it is a dependent variable through equality constraint.

Let $P_h(t)$ be the power transferred from device h when positive and Δt be the discrete time step. Then, allowing for regenerative braking, the state equation for the battery is:

$$E_1(t + 1) = \beta_1^{\text{loss}} E_1(t) - \alpha_1^C(U_1(t)) \min(0, U_1(t)) - \frac{1}{\alpha_1^D(U_1(t))} \max(U_1(t), 0) \quad (1)$$

Likewise, the state equation for the supercapacitor is:

$$E_2(t + 1) = \beta_2^{\text{loss}} E_2(t) - \alpha_2^C(U_2(t)) \min(0, U_2(t)) - \frac{1}{\alpha_2^D(U_2(t))} \max(U_2(t), 0) \quad (2)$$

Here, $\alpha_h^C(U_h(t))$ and $\alpha_h^D(U_h(t))$ respectively denote the charging and discharging efficiencies of device h . These depend on the energy transfer $U_h(t) = P_h(t) \div \Delta t$ as described in [21], since there is

more loss for high power transfer. β_h^{loss} is furthermore the leakage of device h . We write Eq. (1) and (2) more concisely as $x(t+1) = f(x(t), u(t), w(t))$, where $w(t)$ is the demand at the next period, a bounded random variable.

In this model, the energy in storage device h is bounded in the range $E_h^{\min} \leq E_h(t) \leq E_h^{\max}$. We discretize this range into N_h energy levels for the device h , and define $N = N_1 N_2$. The energy transfers are constrained to the ranges $-C_h^{\max} \leq U_h(t) \leq D_h^{\max}$, where C_h^{\max} and D_h^{\max} are the upper bounds on the charging and discharging rates, respectively. They are also discretized, as explained below after (2).

There are three relevant costs in this model. The first is the loss due to inefficient energy exchanges between devices, which is the sum of the energy exchanges weighted by their inefficiencies:

$$\sum_{h=1}^2 \left(\frac{1}{\alpha_h^D} - 1 \right) \frac{\max(U_h(t), 0)}{E_h^{\max}} - \sum_{h=1}^2 (1 - \alpha_h^C) \frac{\min(0, U_h(t))}{E_h^{\max}}$$

The second is that of high battery state of energy (SoE) [13, pg. 2615] and low supercapacitor SoE [14], where SoE is the ratio $E_h(t) \div E_h^{\max}$. It is given by

$$\nu_1 \left(\frac{E_1(t)}{E_1^{\max}} \right)^2 + \nu_2 \left(1 - \frac{E_2(t)}{E_2^{\max}} \right)^2$$

where ν_1 and ν_2 are positive weights for the battery SoE and supercapacitor SoE costs, respectively.

The final cost is that for large energy transfers from the battery, given by

$$\left(1 - \frac{E_1(t)}{E_1^{\max}} \right)^2 \left[K_1 \left(\frac{\max(U_1(t), 0)}{E_1^{\max}} \right)^2 + K_2 \left(\frac{\min(0, U_1(t))}{E_1^{\max}} \right)^2 \right]$$

Here K_1 and K_2 are, respectively, positive weights for the battery discharging and charging. This cost penalizes larger power transfers, which cause greater battery degradation [6], but only at low battery SoE.

We define the stage cost of the optimization problem, $g(x(t), u(t))$, to be the sum of the above three cost expressions. We then collect this cost function and the above constraints in the following optimization problem:

$$\min \mathbb{E} \left[\sum_{t=0}^{T-1} g(E_1(t), E_2(t), L(t), U_1(t), U_2(t)) \right] \quad (2)$$

subject to

$$E_h^{\min} \leq E_h(t) \leq E_h^{\max}, h = 1, 2 \quad (3)$$

$$-C_h^{\max} \leq U_h(t) \leq D_h^{\max}, h = 1, 2 \quad (4)$$

$$U_1(t) + U_2(t) = L(t) \quad (5)$$

$$E_h(t+1) = \beta_h^{\text{loss}} E_h(t) - \alpha_h^C \min(0, U_h(t)) - \frac{1}{\alpha_h^D} \max(U_h(t), 0), h = 1, 2 \quad (6)$$

Note that the energy balance equality constraint (5) can be substituted into the objective and the charging inequality constraints, which reduces the decision vector to $u(t) = U_1(t)$. The range of the energies that may be transferred by each device h at time t , $-C_h^{\max} \leq U_h(t) \leq D_h^{\max}$, is discretized into ρ_h levels. This means that the number of possible values of $u(t)$ is $\rho = \rho_1 \rho_2$.

Let Ω_t be the set of state-control pairs at time t that satisfy constraints (3) – (6). The optimal objective value of (2) is the optimal value function

$$J_0(x(0)) = \min \mathbb{E} \left[\sum_{t=0}^{T-1} g(x(t), u(t)) \right] \quad (7)$$

The optimization in (7) may then be written as the DP:

$$J_t(x(t)) = \min_{u(t)} g(x(t), u(t)) + \mathbb{E}[J_{t+1}(f(x(t), u(t), w(t)))]$$

starting from $J_T(x(T)) = 0$. This is subject to constraints (3) – (6), which can be expressed as $[x(t), u(t)]^T \in \Omega_t$. Here $U_2(t) = L(t) - U_1(t)$, using the energy balance constraint.

In this model, $L(t)$ is a random quantity which is observed at time t before the control $u(t)$ is applied because the controller must have full knowledge of the demand. It evolves according to $L(t+1) = w(t)$, where $w(t)$ is generated by a random process. Note that $w(t)$ depends on the energy in the subsequent state, $E_1(t+1)$ and $E_2(t+1)$. This is because the demand at any iteration cannot exceed the total stored energy at that iteration.

We incorporate regenerative braking in our model by allowing $L(t)$ to take on negative values. Let $L^{\min} = -\min(E_1^{\max} + E_2^{\max}, C_1^{\max} + C_2^{\max})$ be the minimum demand and $L^{\max} = \min(E_1^{\max} + E_2^{\max}, D_1^{\max} + D_2^{\max})$ be the maximum demand. Then the random demand is bounded to the range $L^{\min} \leq L(t) \leq L^{\max}$. It is also discretized into M levels.

To express the problem as a linear program, we define four sets. Let W be the set of random demands, indexed by $k \in \{1, \dots, M\}$. Let Y be the set of points $[E_1, E_2]^T$, indexed by $\tau \in \{1, \dots, N\}$. Let U be the set of controls, indexed by $p \in \{1, \dots, \rho\}$. Finally, let X be the set of values the discretized state, $x(t)$, can take. By the above definition, $X = Y \times W$, so we can index X by $i \in \{1, \dots, NM\}$. We use subscripts to index the elements each set, e.g., x_i is the i^{th} element in set X .

We hereon omit time indices because we will solve an infinite horizon approximation. Let x_i be the current state and $f(x_i, u_p, w_k)$ be the next state. As the state evolves with each iteration of the DP, the subsequent state may not lie in X . We use weighted interpolation to determine the value function at the state $f(x_i, u_p, w_k) \notin X$. If x_j is any state in X where $j \in \{1, \dots, NM\}$, the weight of $J_{t+1}(x_j)$ that contributes to $J_{t+1}(f(x_i, u_p, w_k))$ is defined as $q_{k,p,i,j}$. We note that no interpolation is needed for the demand, L , while solving the optimization problem offline because the values are sampled from the set W , though this is not the case when simulating online. Hence, we use bilinear interpolation to determine the weights $q_{k,p,i,j}$, as described in [15, pg. 319] and elsewhere. The

weights satisfy $\sum_{j=1}^{MN} q_{k,p,i,j} = 1$ and $0 \leq q_{k,p,i,j} \leq 1$.

We express our model as a Markov chain to enable a linear programming formulation. Let $P(f(x_i, u_p, w_k) | x_i, u_p)$ be the probability of moving from $x_i \in X$ to $f(x_i, u_p, w_k)$. The only random variable in $f(x_i, u_p, w_k)$ is $w_k \in W$, so we may express this in terms of the probability of w_k , which is $P(w_k | x_i, u_p)$. As shown in [16], we can consolidate these probabilities and the weights into a single transition probability matrix, \bar{P}_p , which depends on control u_p . The entry (i, j) of this matrix is:

$$\bar{p}_{i,j}(u_p) = \sum_k^M P(w_k | x_i, u_p) q_{k,p,i,j}$$

III. Infinite Horizon Approximation

The length of each time period is very small relative to the total driving time. For example, each period might represent a second. If the total driving time is an hour, this means 3600 periods. This motivates the use of an infinite horizon approximation.

The infinite horizon DP is:

$$J(x) = \min_u g(x, u) + \alpha \mathbb{E}_w [J(f(x, u, w))] \quad (8)$$

subject to $(x, u) \in \Omega$. Here α is a real-valued discount factor such that $0 < \alpha < 1$. Using this approximation, in the following sections we formulate an LP equivalent to this DP and use its solution to obtain the optimal policy. As a reminder, in the context of the hybrid storage problem, $x = [E_1, E_2, L]^T$ and $u = U_1$.

III.A. Linear Programming Formulation

We can re-formulate the DP in (8) as a linear program (LP) [10]. This allows us to make use of faster solution techniques for linear programs, such as the simplex method, and also to apply approximation.

Let $\lambda \in \mathbb{R}^{MN}$ be a vector of the value function at each value in X , and let g be a vector of the stage cost for each state-control pair. Then, the problem can be expressed as:

$$\max_{\lambda} 1^T \lambda \quad \text{s. t.} \quad (\iota - \alpha \bar{P}) \lambda \leq g \quad (9)$$

Here, $\iota = [I, \dots, I]^T$ is a matrix consisting of ρ copies of the identity matrix and $\bar{P} = [(\bar{P}_1)^T, \dots, (\bar{P}_\rho)^T]^T$.

III.B. Optimal Policy

The optimal policy may be constructed directly from the solution of the LP in (9) [10], which is the optimal value function. For each state $x_i \in X$, it is given by:

$$u^*(x_i) = \min_{u_p \in U} g(x_i, u_p) + \alpha \sum_{j=1}^{MN} \sum_{k=1}^M P(w_k | x_i, u_p) q_{k,p,i,j} \lambda_j$$

Here, we use the notation λ_j to refer to the element in the vector λ which corresponds to value function evaluated at the state $x_j \in X$.

For each state $x \notin X$, we use interpolation to implement the suboptimal policy, which is:

$$u(x) = \sum_{j=1}^{MN} q_j(x) u^*(x_j)$$

Here, we use the notation $q_j(x)$ to refer to the weight of $u^*(x_j)$ that contributes to $u(x)$, since the state $x \notin X$.

We sample the demand, L , from a continuous probability distribution when implementing the suboptimal policy online, as this allows us to simulate a driving cycle even with coarse discretization of demands when solving the LP. This means the above interpolation must be applied to determine $u(x)$ additionally in the case when $L \notin W$, not just the case when $[E_1, E_2]^T \notin Y$. As a result, in this case the weights $q_j(x)$ are obtained from trilinear interpolation, unlike $q_{k,p,i,j}$.

IV. Approximate Solution of LP

Unfortunately, the state space is too large to use a fine discretization. To improve tractability, we approximate the value function with basis functions using the methodology of [11].

The approximate value function has the form Φr , where $\Phi \in \mathbb{R}^{MN \times R}$ is termed the design matrix and $r \in \mathbb{R}^R$ is a vector of weights. By choosing R to be less than MN , we reduce the size of the linear program. The q^{th} column vector of Φ is a basis function denoted by ϕ_q . In this section, we discuss methods to generate basis functions for the two-storage problem.

We give all basis functions equal weighting as no procedure exists for optimally determining the weights [11]. This leads to the following approximation, which is also an LP:

$$\max_r 1^T \Phi r \quad \text{s. t.} \quad (\iota - \alpha \bar{P}) \Phi r \leq g$$

We use a combination of two types of basis functions: monomials, and vectors based on heuristic state aggregation. We describe these in the following two subsections.

IV. A. Monomials

We first use monomial basis functions [17]. Let i be the index of state x_i and j be the column index for entry (i, j) . Let $\tilde{\Phi}$ be the submatrix of Φ with columns corresponding to monomial basis vectors. $\tilde{\Phi}$ has the form:

$$\tilde{\Phi}_{i,j} = x_i^{j-1}$$

The exponent notation is used here to indicate a monomial of degree $j - 1$ in the state variables E_1, E_2 , and L .

IV. B. State Aggregation

We use the state aggregation technique of [18] to reduce the number of decision variables. This is to say that we approximate the optimal value function by a piecewise constant function, where each interval of the domain corresponds to an aggregation.

Appropriate state aggregations depend on the specific problem. However, in general, each aggregation is defined to have approximately the same value of $\phi_j(x_i)r_j$ for all states x_i in that aggregation, where j is the index of the aggregation (basis function). Let $\hat{\Phi}$ be the submatrix of Φ with columns corresponding to state aggregation basis vectors. $\hat{\Phi}$ has the form:

$$\hat{\Phi}_{i,j} = \begin{cases} 1 & \text{if } i \in H_j \\ 0 & \text{if } i \notin H_j \end{cases}$$

where H_j is the set of indices i in aggregation j . The aggregations are disjoint and the union contains all indices $i = 1, \dots, NM$. The aggregations H_j are determined by running value iteration a small number of times, and grouping similar entries of the value function.

V. Numerical Example

We tested our approach in simulation on a typical electric vehicle with a battery and a supercapacitor described in [5]. We used $R = 6\text{m}\Omega$ for the battery and $P_2^{\text{max}} = 274.2\text{kW}$ for the supercapacitor to calculate the charging and discharging efficiencies for each device, based on the datasheets of the devices in [5]. We also used a leakage factor of $\beta_h^{\text{loss}} = 1$ for both, based approximately on the same. In this paper, the approximate linear program was modelled in CVX [19] and solved using Gurobi [20] for various demand sequences. All other parameters were kept constant, which included the discount factor $\alpha = 0.99$, weighting factors $K_1 = K_2 = 1000$, and weights $v_1 = 100$ and $v_2 = 1$ for the SoE penalties on the battery and supercapacitor respectively. Finally, to perform the approximation, we used $R = 496$ basis functions composed of 210 state aggregations and monomials up to order 10.

We sized the energy storage system as per [5]. The energy capacities of the battery and the supercapacitor are approximately $E_1^{\text{max}} = 33.7\text{kWh}$ and $E_2^{\text{max}} = 160\text{Wh}$, respectively. The power densities of the Li-Ion batteries and supercapacitors in [5] are 0.43 kW/kg and 6.7 kW/kg , so for the given sizes, the maximum energy discharge ratings were approximately $D_1^{\text{max}} = 4.5\text{Wh}$ for the battery and $D_2^{\text{max}} = 7.5\text{Wh}$ for the supercapacitor. This rating assumes a time step of $\Delta t = 0.1\text{s}$, which we determined to be a reasonable time for energy transfers. Given that the input and output power ratings of the supercapacitor are symmetric [14], we chose to set $C_2^{\text{max}} = D_2^{\text{max}}$. However, the input power rating of a battery is generally lower than its output rating [21, pg. 14]. Based on the datasheet for the battery used in [5], the battery's rated charging power is 48 W/kg , so we set $C_1^{\text{max}} = 0.50\text{Wh}$.

We first solved the approximate linear program offline to determine the optimal policy, where the probabilities $p_{i,j}(u_p)$ of discrete demands in set W are distributed according to an artificial Beta distribution parameterized by β . We then applied demand sequences where the continuous random demands were sampled from the same distributions to determine how the battery and supercapacitor should be optimally controlled under different conditions. In [5], the storages were sized for a maximum power demand giving a travel time of 9600s. Accordingly, we simulated starting from initial energy state of $E_1 = E_1^{\text{max}}$ and $E_2 = E_2^{\text{max}}$ for the same duration. We tested the optimal policy online when the demands are generated by two different beta distributions: one parameterized by $\beta = 1$, and the other by $\beta = 10$. We also tested both without regenerative braking (i.e., $L^{\text{min}} = 0$) and with regenerative braking (i.e., $L^{\text{min}} = -\min(E_1^{\text{max}} + E_2^{\text{max}}, C_1^{\text{max}} + C_2^{\text{max}})$) to compare.

The simulation for the first case ($\beta = 1$ and no regenerative braking) is illustrated in Figure 1.

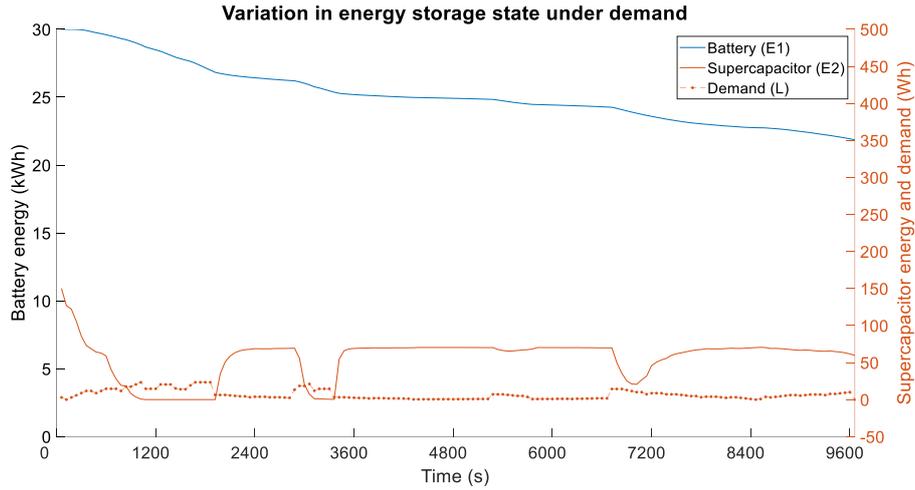


Figure 1: Testing optimal policy online with a sequence generated by a distribution with $\beta=1$ and no regenerative braking.

We see that the battery will tend to supply the majority of the energy both for very low demands and when the supercapacitor is depleted. We also observe that the supercapacitor often satisfies larger demands of short duration, such as during the interval of 130s to 140s in Figure 1. This result is expected because there is a relatively high cost of large discharges from the battery.

When there is no demand, the battery sometimes recharges the supercapacitor. This also confirms that the optimal policy is not myopic, since a trade-off is made between instantaneous transfer loss and satisfying fluctuating demands at a later time by pre-emptively charging the supercapacitor. Were it not for the benefits of the latter, there would be no reason for such energy exchanges. However, because the supercapacitor energy capacity is quite low, the demand must drop quite low until it is optimal for this to occur.

Additionally, we observed that by reducing K_1 and K_2 to 1, there is less cost for charging and discharging the battery. One can see this in Figure 2, which shows the response when the demand sequence is generated by a distribution with $\beta = 10$ and there is no regenerative braking.

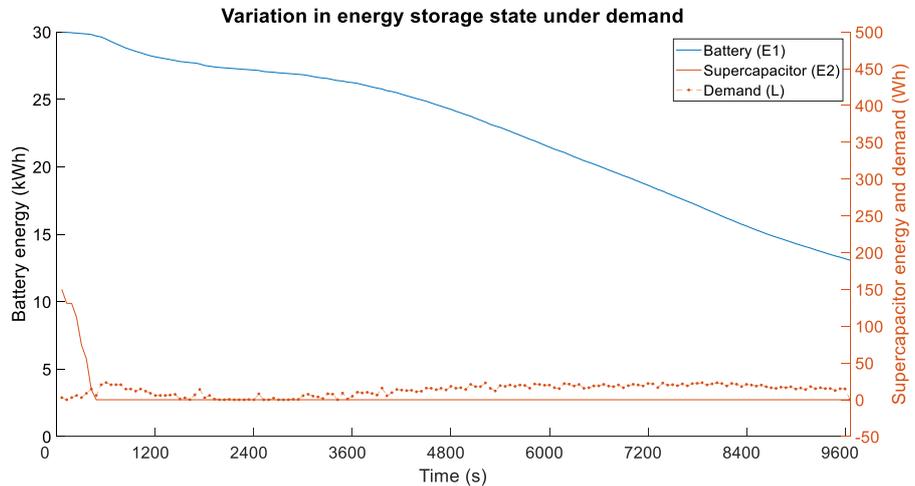


Figure 2: Testing optimal policy online with a sequence generated by a distribution with $\beta=10$. No regenerative braking, and lower battery charging and discharging cost ($K_1=K_2=1$).

We see that decreasing the cost of discharging from the battery results in the optimal policy being to not recharge the supercapacitor, even when there is no demand. This is due to the cost of energy transfer losses being on the order of the battery discharge cost, unlike in the previous test. The result confirms our expectations.

Finally, we tested the case with regenerative braking to compare to the above policies. Figure 3 shows the optimal response to a demand that is generated by a distribution with $\beta = 1$.

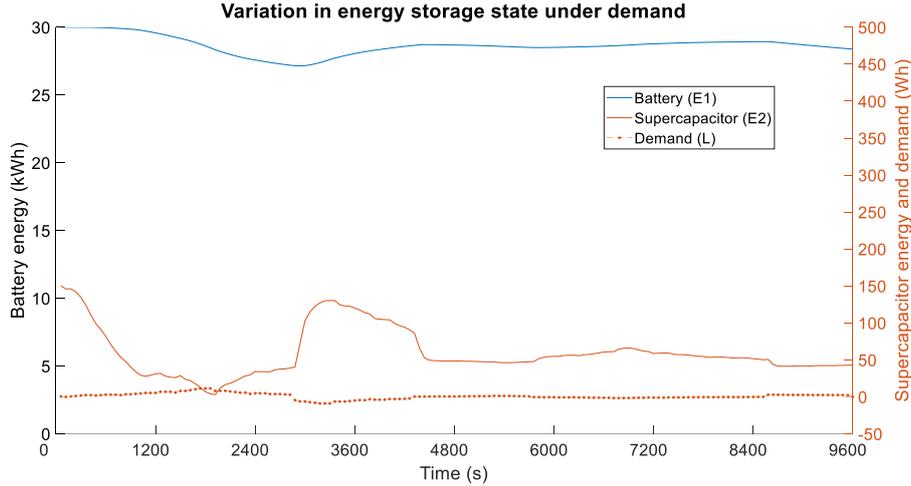


Figure 3: Testing optimal policy online with a sequence generated by a distribution with $\beta=1$ and regenerative braking present.

We see that the optimal policy in this case is to charge and discharge the supercapacitor sooner than the battery to satisfy the demand. This is comparable to the case when there is no regenerative braking, as in Figure 1. One difference is that the battery charges the supercapacitor less when there is low demand, unlike in Figure 1. This enables the supercapacitor to absorb future energy from regenerative braking.

In addition to testing the optimal policy, we also tested the value function approximation. We quantified the approximation error for problems of small size. Figure 4, for example, illustrates the trade-off made between the approximation error and the number of iterations in solving the approximate LP, where the latter depends on the number of basis functions.

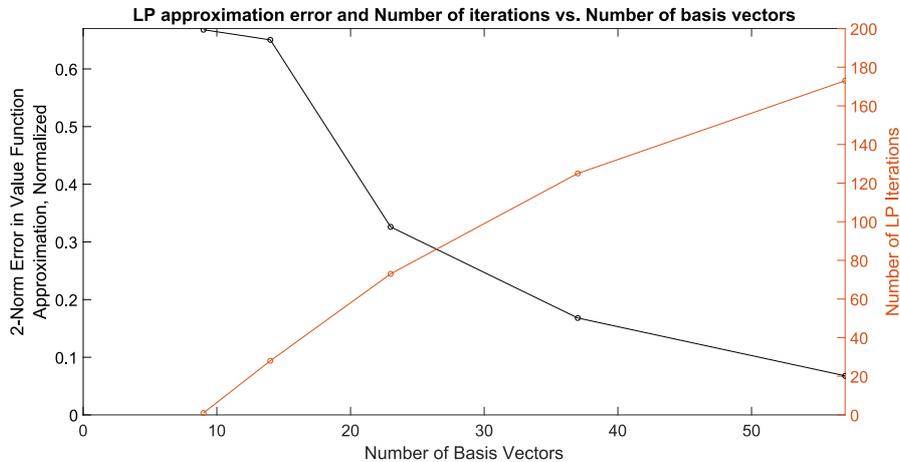


Figure 4: Variation in approximation error and solution time with number of basis functions added. Test size: $N_1=6$, $N_2=5$, $M=16$.

This shows that the approximation error can indeed be made arbitrarily small by flexibly adding more basis functions.

VI. Conclusion

This paper has developed and optimized a model for energy transfers from two storage devices to satisfy a random demand. The Markov decision process was formulated as a linear program and approximated using basis functions. The problem was then solved offline and tested online for the numerical example of an EV with hybrid energy storage and regenerative braking. The final results confirm that the approximate LP does allow for determining a suboptimal but high-performance policy. Further work remains to be done to size the storages in this model.

BIBLIOGRAPHY

- [1] Y. Shi et al. "Optimal Battery Control Under Cycle Aging Mechanisms in Pay for Performance Settings," IEEE Transactions on Automatic Control, 2018.
- [2] Wenten Zeng, John Gibeau, and Mo-Yuen Chow. "Economic benefits of plug-in electric vehicles using V2G for grid performance-based regulation service". IECON 41st Annual Conference of the IEEE Industrial Electronics Society, 2015, pp.: 004322 - 004327.
- [3] Phatiphat Thounthong, Stephane Rael, and Bernard Davat. "Energy management of fuel cell / battery / supercapacitor hybrid power source for vehicle applications," Journal of Power Sources, Volume 193, Issue 1, 2009, pp.: 376-385.
- [4] Riyanto Trilaksono Bambang et al. Energy management of fuel cell/battery/supercapacitor hybrid power sources using model predictive control," IEEE Transactions on Industrial Informatics, Volume 10, Issue 4, 2014, pp.: 1992-2002.
- [5] M. Moshirvaziri et al. "Power-mix optimization for a hybrid ultracapacitor/battery pack in an electric vehicle using real-time GPS data". Proceedings of the 39th Annual Conference of the IEEE Industrial Electronics Society, Nov. 2013, pp.: 4666-4671.
- [6] Riyanto Trilaksono Bambang et al. "Energy management of fuel cell/battery/supercapacitor hybrid power sources using model predictive control," IEEE Transactions on Industrial Informatics, Volume 10, Issue 4, 2014, pp.: 1992-2002.
- [7] MY Ayad, M Becherif, and A Henni. "Vehicle hybridization with fuel cell, supercapacitors and batteries by sliding mode control," Renewable Energy, Volume 36, Issue 10, 2011, pp. 2627-2634.
- [8] B. Liu et al. "Hybrid Electric Vehicle Downshifting Strategy Based on Stochastic Dynamic Programming During Regenerative Braking Process," IEEE Transactions on Vehicular Technology, Volume 67, Issue 6, June 2018, pp.: 4716-4727.
- [9] H. Marefat, M. Jalalmaab, and N. L. Azad. "Energy management of battery electric vehicles hybridized with supercapacitor using stochastic dynamic programming," Proceedings of the SICE International Symposium on Control Systems (SICE ISCS), Mar. 2018, pp.: 199-205.
- [10] D.P. Bertsekas. Dynamic Programming and Optimal Control, Vol. II. 3rd ed. Athena Scientific, 2007.
- [11] D. P. de Farias and B. Van Roy. "The Linear Programming Approach to Approximate Dynamic Programming," Operations Research, Volume 51, Issue 6, Nov. 2003.
- [12] R. E. Bellman. Dynamic Programming. Princeton University Press, 1957.
- [13] Ali Ahmadian et al. "Plug-in electric vehicle batteries degradation modeling for smart grid studies: Review, assessment and conceptual framework," Renewable and Sustainable Energy Reviews, Volume 81, 2018, pp.: 2609-2624.
- [14] Omar Laldin, Mazhar Moshirvaziri, and Olivier Trescases. "Optimal power Flow for hybrid ultracapacitor systems in light electric vehicles," IEEE Energy Conversion Congress and Exposition. 2011, pp.: 2916-2922.
- [15] G. Dahlquist and A. Bjorck. Numerical Methods. 1st ed. Prentice-Hall, 1974.
- [16] M. Reiter. "Introduction to dynamic programming". Lecture notes, Ch. 2. URL: <https://elaine.ihs.ac.at/~mreiter/dynprobgse.pdf>. 2018.
- [17] D. P. Bertsekas and J. N. Tsitsiklis. "Neuro-dynamic programming: an overview," Proceedings of the 34th IEEE Conference on Decision and Control, Vol. 1, Dec. 1995.
- [18] S. Darbha et al. "State aggregation based linear programming approach to approximate dynamic programming," Proceedings of the 49th IEEE Conference on Decision and Control (CDC), Dec. 2010.
- [19] Michael Grant and Stephen Boyd. CVX: Matlab Software for Disciplined Convex Programming, version 2.1. <http://cvxr.com/cvx>. Mar. 2014.
- [20] LLC Gurobi Optimization. Gurobi Optimizer Reference Manual. 2018. <http://www.gurobi.com>.
- [21] Mazhar Moshirvaziri. "Ultracapacitor/battery hybrid energy storage systems for electric vehicles," MA thesis