Learning to Manage Combined Energy Supply Systems

Azalia Mirhoseini, Farinaz Koushanfar
Dept. of of Electrical and Computer Engineering, Rice University, Houston, Texas
azalia@rice.edu, farinaz@rice.edu

Abstract—**The operability of a portable embedded system is severely constrained by its supply's duration. We propose a novel energy management strategy for a combined (hybrid) supply consisting of a battery and a set of supercapacitors to extend the system's lifetime. Batteries are not sufficient for handling high load fluctuations and demands in modern complex systems. Supercapacitors hold promise for complementing battery supplies because they possess higher power density, a larger number of charge/recharge cycles, and less sensitivity to operational conditions. However, supercapacitors are not efficient as a standalone supply because of their comparatively higher leakage and lower energy density. Due to the nonlinearity of the hybrid supply elements, multiplicity of the possible supply states, and the stochastic nature of the workloads, deriving an optimal management policy is a challenge. We pose this problem as a stochastic Markov Decision Process (MDP) and develop a reinforcement learning method, called Q-learning, to derive an efficient approximation for the optimal management strategy. This method studies a diverse set of workload profiles for a mobile platform and learns the best policy in form of an adaptive approximation approach. Evaluations on measurements collected from mobile phone users show the effectiveness of our proposed method in maximizing the combined energy system's lifetime.**

I. Introduction

The mobile system's lifetime and functionality is limited by its constrained energy supply. The commonly utilized electronic energy supply (EES) unit for the mobile and embedded systems is an electrochemical battery. The battery technology has been improving at a very slow rate, setting back the otherwise fast growing processor functional capabilities. The wide-spread usage of batteries is because of their cost, rechargeability, and energy capacity advantages. Their drawbacks include nonlinear dependence of the lifetime on the drawn current, where a higher incident load depletes the battery's energy much faster.

An emerging form of EES with properties complementary to battery is a *supercapacitor (s-cap)*. The energy density of a s-cap is lower than a battery, but it is significantly higher than a capacitor. The s-caps also have higher energy leakage than batteries and their leakage increases with rising the voltage. On the other hand, when compared to batteries, the s-caps have better efficiencies in chargeing/discharging, higher number of cycles, where they are also more reliable and more robust to operational conditions. Recent work has suggested that a hybrid combination of batteries and s-caps can take advantage of their complementary energy properties [1]. Examples of combined solar batteries and s-caps were prototyped; The earlier evaluations were promising, especially for the sensor network load currents that have low duty cycles [2], [3], [4].

This paper aims to perform adaptive energy management optimization for a hybrid set of EES elements (consisting of s-caps and a battery) to extend the system's lifetime for much more complex scenarios than those available earlier. At each point of time, based on the present system state and the incident load, the management unit decides on a set of actions to best save the total energy. The problem is a combination of discrete decisions and continuous linear and nonlinear EES element properties. Our approach to address this problem is to map all the parameters and values into the discrete domain.

We start by quantizing the charge values in the EES elements, the workload values, and the actions. Next, we define formal notations for the states, actions, and outcomes. Using the discrete values and our introduced notations, we model the hybrid supply management problem as an instance of a discrete time stochastic Markov Decision Process (MDP): at each time step, the battery and the s-caps are at a certain charge state. Given the (present) load demand, the uncertainty in future workload, and the cost of taking actions, the system needs to decide among the presently available actions to maximize the overall objective of extending the lifetime.

The MDP solution should simultaneously consider all the dimensions for finding the optimal management policy. There are at least two sets of challenges. The first set of challenges is due to the three sources of curse of dimensionality: the state space, the action space, and the demand space. The second set of challenges is due to the workload uncertainty; Optimal management depends on the (uncertain) future load values that are often not available and hard to estimate in advance.

We develop a reinforcement learning method based on Q-learning, that studies a diverse set of collected workload traces to find an approximate optimum MDP solution. To address the first challenge set, our approach breaks down the computations into time steps and then iteratively traverses the states and variables to update the management policy. To address the second challenge set, our method iteratively updates the decision states. Assuming the workload fluctuations can be modeled as a wide sense stationary (WSS) random process, the Q-learning approach converges to the optimal policy [5], [6]. Our user studies on mobile platforms shows that the workload scenarios conform to the WSS assumption.

Our explicit contributions are as follows: (i) We develop a transformation of the hybrid EES management optimization problem to the discrete domain and formalize the parameters for the states, outcomes and actions. The problem is formulated as a stochastic dynamic MDP in the discrete domain. (ii) We introduce an approach for addressing our

978-1-61284-660-6/11/$26.00 © 2011 IEEE

MDP problem based on approximations by reinforcement learning that adaptively operates and learns the best policy. Our method overcomes the challenges associated with curse-of-dimensionality and inherent uncertainty that arise in the optimization problem. (iii) We validate our methods and models on extensive mobile phone measurements in our lab.

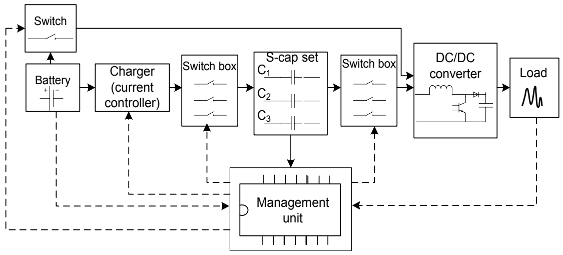

Fig. 1. Block diagram of the combined source.

II. Related work

Recent progress in the s-cap technology is paving the way for the widespread adoption of the s-caps as a major EES element. New generation of s-caps benefits from high energy densities in addition to their inherent high power densities [7]. It is now possible to utilize them in lower power applications such as portable systems' supplies. Larger s-caps have already been used in hybrid car applications.

A large body of work has focused on characterizing battery's behavior and exploiting its features for increasing the supply system's lifetime [8], [9], [10]. This paper incorporates the non-linear battery and s-cap models within the reward function of our MDP optimization framework.

The generic idea of combined sources consisting of fuel cells, batteries, and s-caps has been recently proposed [11], [1], [12], [13]. However, a detailed optimization framework considering different design parameters has not been provided. In our recent work [14], we proposed a hybrid management methodology, considering different optimization parameters, for scenarios where the system's tasks could be reordered (within a time budget) in order to maximize the supply's lifetime efficiency. In this work we target the more general scenarios, where optimizations are done on realtime workloads.

Hybrid systems integrating s-caps along with energy scavenging sources such as photovoltaic cells have recently been studied [3], [2], [4]. The optimization methods rely on both energy scavenging system and as s-cap properties. The earlier work has mainly focused on sensor network applications with notably low duty cycles. Those methods are often not directly applicable to more complex workloads. For embedded systems scenarios, such as mobile phones, there is an apparent difference in the usage patterns and reliability/usability requirements. Furthermore, battery properties are considerably different from other EES elements such as energy scavenging sources in that (i) they have a high energy density and can always provide a continuous source of energy to reliably serve the load, and (ii) they have a particular nonlinearity pattern that favors drawing a low current to increase their lifetime.

Learning the stochastic component(s) has been shown to be effective in finding near-optimal solutions to complex MDP problems [15], [16]. Q-learning is a reinforcement learning method mostly used in unsupervised settings [5]. To the best of our knowledge, this is the first work that models the combined energy supply management as an MDP problem and addresses the problem using reinforcement learning.

III. Combined system overview

To derive an optimal hybrid management strategy, the characteristics of each EES element should be considered. For example, battery's *rate capacity effect* governs the non-linear relationship between battery's lifetime and its charge/current demand; A high current load can exhaust a battery much faster. As another example, s-caps outperform batteries in power delivery because of their large power density that enables them to supply higher power loads. The Hybrid EES management should consider multiple aspects. First, it is desired to ensure that s-caps have a sufficient charge prior to serving the high-power loads. A challenge is that the workload peaks are usually not known a priori. Second, due to leakage, it is inefficient to fully precharge the s-caps when they are not needed for a long time duration. Therefore, the timing and rate of battery's charge recycling into the s-caps should be carefully selected. Third, to enhance efficiency, decisions for choosing the best EES element should be made in realtime based on the incident workload observations. Lastly, it is necessary to consider the cost (overhead) of each control decision.

Considering the above aspects, the management tasks, referred to as *actions*, are to adaptively assign the best EES element to supply the load. Also, there is a need to recycle the charge from the battery into the s-caps at optimal rates and at proper times. A key point is that the best EES management policy is a function of the uncertain workload. While the exact workload values are unpredictable, our in-lab measurements confirm their stochastic properties for a mobile platform.

To find the best management policy, we start by collecting a representative set of load currents from the comprehensive user studies. Next, we develop a reinforcement learning approach that studies the stochastic workload properties to provide the most energy efficient actions over the representative load scenarios. The reinforcement learning is performed offline. The resulting policy is then loaded to the portable system's hybrid management unit. The policy is applied online as the instantaneous load is observed at each state.

Figure 1 shows the block diagram of our proposed system. At each decision point, the management unit receives an instance of the online load and observes the EES state. Then, based on the learned policy, it determines the next action of the combined source. The charger and current controller unit sets the appropriate rate of charge transfer from the battery to the s-cap set as dictated by the management unit. The DC/DC converter is used to maintain the voltage value at the load level. The switches are incorporated to enable the appropriate charge flow (from the battery to the load or other s-caps or from the s-caps to the load) as needed according to the action.

IV. Problem definition

In this section, we outline the parameters, our hybrid management problem, constraints, and the design properties.

A. Notation and parameter definition

We adopt the common notation used in dynamic optimization to model the quantized discrete state/action problem. All the states and actions are made at pre-specified discrete time instants from a set $\mathcal{T}$, where $\mathcal{T} = \{0, 1, 2, ..., T-1\}$ and T is the total number of planning periods.

1) The source and load parameters: We assume that the charge values of the batteries and s-caps can only take discrete values; The average workload demand is discrete and constant during one time period. We also put a limit on the maximum charge that can be transferred from a battery to a s-cap in a single time interval. This condition is imposed to avoid battery exhaustion for charging the s-caps. Our notations are,

N_C Total number of s-caps.
$\mathcal{N}_c$ The index set of s-caps: $\mathcal{N}_c = \{1, 2, \ldots, N_C\}$.
r_{it}^{cap} Charge state of s-cap $i \in \mathcal{N}_c$ at time $t \in \mathcal{T}$.
C^{cap} S-cap charge values ($r_{it}^{cap} \in C^{cap}$), C^{cap}= $\{0, \Delta r_1, 2\Delta r_1, \ldots, R_{max}^{cap} - \Delta r_1, R_{max}^{cap}\}$.
r_t^{bat} Charge state of the battery at time $t \in \mathcal{T}$.
C^{bat} Battery charge values ($r_{it}^{bat} \in C^{bat}$), C^{bat}= $\{0, \Delta r_2, 2\Delta r_2, \ldots, R_{max}^{bat} - \Delta r_2, R_{max}^{bat}\}$.
w_t Load demand during $(t, t+1)$ for $t \in \mathcal{T}$.
R_{th} Maximum charge that battery transfers to a s-cap at a single period.

2) Action parameters: At each planning instant, a decision (action) for assigning the best source to the workload should be made. Besides, it should be determined whether the battery charges any of the s-caps or not. In our system model, a s-cap cannot be simultaneously charged by the battery and discharged by the load. The action-related parameters are,

$\mathcal{A}$ Set of all possible Actions.
a_t Action at time t: $a_t = \{a_{1t}^{cap}, a_{2t}^{cap}, ...a_{N_C t}^{cap}, a_t^{bat}\}$.
a_{it}^{cap} Action at time t for s-cap $i \in \mathcal{N}_C$, $a_{it}^{cap} = (a_{it}^{cap}(x), a_{it}^{cap}(y)) \in \{(0,0), (0,1), (1,0)\}$.
$a_{it}^{cap}(x)$ 1 if s-cap $i \in \mathcal{N}_C$ gets charged; 0 otherwise.
$a_{it}^{cap}(y)$ 1 if s-cap $i \in \mathcal{N}_C$ supplies the load; 0 otherwise.
a_t^{bat} 1 if battery supplies load in $(t, t+1)$; 0 otherwise.

3) State variables and state transition function: State variables are representative of the relevant history of the system. In our model, the state variable contains the EES's state of charge and the change in battery's charge in consecutive time instants. The transition function takes the current state, actions, and load as inputs. It outputs the next state. The model and notations are,

$\mathcal{S}$ Set of all possible hybrid system states.
s_t State variable at time $t \in \mathcal{T}$, $s_t \in \mathcal{S}$ $s_t = \{r_{t1}^{cap}, r_{t2}^{cap}, \ldots, r_{tN_C}^{cap}, r_t^{bat}, \Delta r_t^{bat}\}$.
Δr_t^{bat} $=r_{t-1}^{bat} - r_t^{bat}$ for $t \in \mathcal{T}$.
$S^M(.)$ Transition function, $s_{t+1} = S^M(s_t, a_t, w_t)$.

At time t, the transition function computes the next charge state of the s-cap indexed i denoted by $r_{(t+1)i}^{cap}$, as follows,

$$r_{(t+1)i}^{cap} = r_{ti}^{cap} - a_{it}^{cap}(y) w_t + (R_{th} - w_t a_t^{bat}) a_{it}(x).$$

The equation adjusts the s-cap's charge ($r_{(t+1)i}^{cap}$), by the amount of charge the s-cap delivers to the load ($a_{it}^{cap}(y) w_t$). It also adds the amount of charge the s-cap receives from the battery $(R_{th} - w_t a_t^{bat}) a_{it}(x)$ during $(t, t+1)$.

The transition function similarly adjusts the next charge state of the battery ($r_{(t+1)}^{bat}$) as follows,

$$r_{(t+1)}^{bat} = r_t^{bat} - w_t a_t^{bat} - (R_{th} - w_t a_t^{bat}) \sum_{i \in \mathcal{N}_c} a_{it}(x).$$

The equation above adjusts the battery's charge (r_{t+1}^{bat}) by the amount delivered to the load ($w_t a_t^{bat}$), and the amount sent to the s-caps ($(R_{th} - w_t a_t^{bat}) \sum_{i \in \mathcal{N}_c} a_{it}(x)$). Note that R_{th}, is an upper limit for the charge that the battery transfers to the s-caps at each single time period.

4) Reward function: Let $\mathcal{R}(s_t, a_t)$ denote the reward function for making a decision a_t at state s_t during the interval $(t, t+1)$. The reward should reflect the net lifetime increment/energy saving of the battery in the combined system; It is a function of the profit in terms of battery energy savings and the penalty in terms of the s-caps' leakage cost and the system's overhead cost. The formal definition is as follows,

$$\mathcal{R}(s_t, a_t) = -e^{(\Delta r_t^{bat})} - e^{(\Sigma_{i \in \mathcal{N}_c} \frac{r_{it}^{cap}}{R_{max}^{cap}})} - \lambda e^{(E[\Delta r_t^{bat}])}. \quad (1)$$

The first term presents the savings in battery charge consumption. Adopting the battery model from [9], we observed the exponential relationship between the battery's incident current and its lifetime. The second term reflects the leakage cost of the s-caps. The exponential modeling for the s-cap's leakage is derived by regression techniques performed on a real s-cap's leakage data. The approximate leakage power model (P_L) that we derived for the $200F$ s-caps used in our evaluations is $P_L(v_c) = \alpha e^{\beta v_c}(w)$, where v_c is the voltage of the s-cap, $\alpha = 1.14.10^{-9}(mW)$ and $\beta = 9.354(v^{-1})$. The third term presents the overhead cost of the system where λ is an estimation of the ratio of the overhead cost to the expected saving profit.

B. Formal problem and constraints definition

Our objective is to maximize the expected reward over the entire T. Assuming that the initial state s_0 is known, the objective function (OF) and constraints (C's) are written as,

$$OF: \quad \max_{a_0 \in \mathcal{A}} \mathcal{R}(s_0, a_0) + E[\sum_{t>0} \max_{s_t \in \mathcal{S}, a_t \in \mathcal{A}} \mathcal{R}(s_t, a_t)] \ (2)$$

$$\begin{aligned} C's: \quad & \forall t \in \mathcal{T}, \\ & 1: \quad a_t^{bat} + \sum_{i \in \mathcal{N}_c} a_{it}^{cap}(y) = 1, \\ & 2: \quad \sum_{i \in \mathcal{N}_c} a_{it}^{cap}(x) \leq 1, \\ & 3: \quad r_{ti}^{cap} \geq a_{it}(y)^{cap} w_t, \qquad i \in \mathcal{N}_C, \\ & 4: \quad r_{ti}^{cap} + (R_{th} - w_t a_t^{bat}) a_{it}(x) \leq R_{max}^{cap}, \\ & 5: \quad r_{(t+1)}^{bat} \geq 0. \end{aligned}$$

The expectation is applied to represent the problem uncertainty caused by the stochastic future workload. The first two constraints reflect our design policy requirements. The first equation states during each single interval $(t, t{+}1)$, exactly one

EES element, either the battery or one of the s-caps, supplies the load. The second constraint states that during each single interval $(t, t+1)$, at most one s-cap could be charged by the battery. We set these limits to avoid the voltage balancing problem. The last three constraints express the physical bounds of the problem. The third constraint set ensures the charge availability of the s-cap selected to supply the load. The forth one is to avoid overcharging s-caps. The last constraint set ensures charge availability of the battery.

V. Addressing hybrid management problem

In this section, we develop methods for addressing the optimization management problem in Equation 2 that results in an efficient dynamic energy management policy. To obtain the best policy for our discrete state, decision and workload space problem as defined in the previous section, we first model the problem as an MDP. At each planning instant, based on the given state of the hybrid source and the incident workload, a decision is made to improve the overall objective function. To solve the MDP problem, a method that considers all the variables of the system model such as the EES properties and the uncertainty of the workload should be incorporated.

Our approach for finding the best solution is based on assigning a value to a state $s_t \in \mathcal{S}$ and following an optimal policy from time t to the end of the planning period. This value is denoted by $V_t(s_t)$ and is called the *value function*. The value function is related to our OF (Equation 2) as follows,

$$V_t(s_t) = \max_{a_t \in \mathcal{A}} \mathcal{R}(s_t, a_t) + E[\sum_{\tau \geq t} \max_{s_\tau \in \mathcal{S}, a_\tau \in \mathcal{A}} \mathcal{R}(s_\tau, a_\tau)] \quad (3)$$

According to the OF in Equation 2, objective maximization is equivalent to obtaining $V_0(s_0)$. It was shown that an approximate MDP solution can be obtained by breaking the complex, multi-period OF in Equation 2 into easier steps at different points of time and expressing it as a *Bellman equation* that is the necessary condition to find an optimal solution of an MDP [16]. The Bellman representation of Equation 3 is,

$$V_t(s_t) = \max_{a_t \in \mathcal{A}} (\mathcal{R}(s_t, a_t) + E[V_{t+1}(s_{t+1}|s_t)]). \quad (4)$$

We first discuss the complexity of solving the above equation, and then provide a solution strategy.

1) Problem complexity: There are two major sources of complexity in Equation 4. The first source comes from the state and action variables which have multiple levels, resulting in a curse of dimensionality. As an example, assume that $N_C = 2$, $R^{cap}_{max} = 40$ and $R^{bat}_{max} = 100$. Then if the battery and s-caps could only take integer charge values, i.e, $\Delta r_1 = 1$ and $\Delta r_2 = 1$, the $\mathcal{S}$ space that contains all the possible states would consist of $41^2 \times 101^2 = 17147881$ states. For a more precise modeling of the continuous values in the discrete space, one would need to quantize to more levels, rendering the problem even more complex. The second source of complexity for our problem is the workload uncertainty. Computing the expectation in Equation 4 requires a priori workload information. Thus, this expectation has to be estimated.

2) Solution strategy: We develop a Q-learning approach to solve our MDP problem. Equation 4, in case the probabilistic characteristics of the load were known, could be solved by a dynamic program stepping backward in time. This requires iterating over all possible states to compute $V_t(s_t)$, which is computationally impractical. Instead, our Q-learning approximation approach studies the load by stepping forward in time and updates an approximate value function for making decisions. We begin with an initial approximation of value function for all points of time t and all possible states s_t. Then, we improve the approximation over a number of updating iterations (denoted by N_{ite}). Q-learning is particularly useful when one cannot predict the next state (at a future time) given a state and an action. We assign a value, which we call *Q-factor*, for being in a state s_t and imposing an action a_t. The value function of a state is the maximum Q-factor that can be obtained over all the applicable actions to that state,

$$\begin{aligned} Q(s_t, a_t) = & \quad \mathcal{R}(s_t, a_t) + V_{t+1}(s_{t+1}), \\ V_t(s_t) = & \quad \max_{a_t \in \mathcal{A}} Q(s_t, a_t). \end{aligned} \quad (5)$$

In this setting, at s_t, we choose an action and then observe the next state (s_{t+1}) based on the sample workload: $s_{t+1} = S^M(s_t, a_t, w_t)$, where, w_t is the workload at time t and $S^M(.)$ is the transition function defined in Section IV-A3. Beginning with an initial approximation of Q-factor values, denoted by $Q^0(s_t, a_t)$, for all possible points of time and states, one can iteratively update Q-factors over time and for the sample workloads. The update criteria is as follows. First, we record a sample estimate of the Q-factor for being at a state s_t and taking the action a_t at the nth ($1 \leq n \leq N_{ite}$) iteration using, $\hat{q}^n = \mathcal{R}(s_t, a_t) + V^{n-1}_{t+1}(s_{t+1})$. Then, the value function and the Q-factor are updated,

$$\begin{aligned} V^{n-1}_{t+1}(s_{t+1}) = & \quad \max_{a_{t+1} \in \mathcal{A}} Q^{n-1}(s_{t+1}, a_{t+1}), \\ Q^n(s_t, a_t) = & \quad (1 - \alpha_{n-1}) Q^{n-1}(s_t, a_t) + \alpha_{n-1} \hat{q}^n. \end{aligned} \quad (6)$$

The superscript indices n denote the iteration number; α_{n-1} is a scaling factor. A typical approach is to set α_{n-1} to $\frac{1}{n}$ to reduce the impact of later observations on the value function.

Our approach addresses the dimensionality challenge by beginning with an estimation for the Q-factors (and thus the value function) and then updating those values over a set of sample loads. After simulating the problem for enough number of iterations, an approximate value function converges to its exact value for WSS processes [16]. We now have a method that does not require computing the expectation in Equation 4. Instead, our method requires access to a set of representative loads. We formalize the solution in Algorithm 1.

Algorithm 1 works as follows: Step 0 initializes the parameters. Step 1 sets the general iteration criteria for updating Q-factors. Step 2 represents iterations over all sample loads. At Step 3, for all time instants, a decision is made first by solving the maximization problem (3a). Then, based on the decision and the sampled load, the next state is obtained (3b). The corresponding V value is then updated (3c). Lastly, a new sample observation is taken (3d) to update the Q-factor (3e).

```
Algorithm 1. Q-learning
0     Initialization
0a    Initialize $N_{ite}$, $S_0$;
0b    Initialize $m_{th}$ sample workload with $w_t^m$ for $t \in \mathcal{T}$,
      $m \in \{1, 2, ..., M\}$;
0c    Initialize $\mathcal{R}(s, a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$;
0d    $Q^0(s, a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$;
1     For n=1 to $N_{ite}$
2       For m=1 to M
2a        $s_0 = S_0$;
3         For t=0 to T-1
3a      Opt a decision to solve maximization problem;
          $a_t = argmax_{a \in \mathcal{A}} Q^{n-1}(s_t, a_t)$;
3b      Find next state using sample workload;
          $s_{t+1} = S^M(s_t, a_t, w_t^m)$;
3c      Update V;
          $V^{n-1}(s_{t+1}) = \max_{a \in \mathcal{A}} Q^{n-1}(s_{t+1}, a)$;
3d      Record a sample Q-factor;
          $q^n = R(s_t, a_t) + V^{n-1}(s_{t+1})$;
3e      Update the estimate of Q-factor;
          $Q^n(s_t, a_t) = (1 - \alpha_{n-1})Q^{n-1}(s_t, a_t) + \alpha_{n-1}q^n$;
```

	Profile Indices (P_i for $1 \le i \le 8$)	Duration (min)
U_1	4,6,1,3,8,2	15,18,40,5,20,5
U_2	1,3,4,7	40,28,25,35
U_3	3,1,4,1,5,1,3,8,1	20,4,6,45,10,5,18,2,18
U_4	1,7,3,1,4,6,1,2,6,8	20,18,15,10,7,13,15,4,16,10
U_5	8,1,7,6,2,8	40,30,8,12,18,20
U_6	4,3,7,1,3,6	25,15,38,14,36
U_7	1,2,8,4	33,60,29,6
U_8	5,7,3,1,4,6,1,2,6,8,5,1,2,1,3	2,10,8,10,7,13,15,5,6,18,4,20,10
U_9	1,8	110,18
U_{10}	5,7,3,1,4,6,1,2,6,8,5,1,2,1,3,5,2,8,6,5,4,1,2,6,3,7,4,5,1	2,4,8,18,7,5,4,2,7,5,6,5,4,6,3,1,2,4,3,2,1,3,2,5,4,1,6,2,6

Fig. 2. User profiles. The tasks are as follows: P1-airplane mode and display off (40mA); P2-default mode and display off (100mA); P3-browsing Internet over 3G network (310mA); P4-the revenge game tap-tap (220mA); P5-3D game GTI racing (400mA); P6-youtube video over WiFi (260mA); P7-youtube video (350mA); P8-voice phone (170mA).

After execution of Algorithm 1, the final derived Q-factors are loaded into the management unit of the combined system. The management unit, based on the observation of the system state s_t at each time t, makes a decision using Equation 7. By applying the optimal action a_t and observing the uncertain workload w_t, the system traverses to its next state s_{t+1}.

$$a_t = arg \max_{a_t \in \mathcal{A}} Q(s_t, a_t). \tag{7}$$

VI. Experimental Results

We adopted the workload scenarios from iPhone battery current measurements in our department [17]. The application current values are presented in the caption of Figure 2 (averaged over one minute with an accuracy of $10mA$). We created 10 user benchmarks. Figure 2 shows the task indices and the corresponding durations for each user benchmark.

A. Evaluation setup

We opted to use two 200F s-caps with capacities of $400C$. At each instant, the management unit was able to use one s-cap to supply the load and the battery to charge the other s-cap. A $4V$, $1000C$ battery was incorporated. Thus, supply system's initial energy capacity was $1800C$. We quantized and scaled the source and demand charge such that at each updating instant t, the load charge demand $\sigma_{load}(t)$ was $round(\frac{\sigma_{load}(t)}{10})$ referring to a general round function. The decisions were made at the beginning of one minute intervals. We assumed a fully charged initial state and set it to $S_0 = \{40, 40, 100, 0\}$ corresponding to the aforementioned workload scaling. R_{th} was set to $1C$ and λ in Equation 1 was set to 10% .

B. Convergence of the Q-learning algorithm

Considering the large state/decision space dimensionality of the problem, we verify the practicality of Algorithm 1 by evaluating its convergence rate. We observe the effect of incrementing N_{ite} on the Q-factors. We refer to the set of all Q-factors as the Q-table. Note that in an ideal case, if we do enough simulations to fill all possible elements of the Q-table, we would ensure that all the system states have been already studied and the Q-factors corresponding to those states are available. It can be seen in Figure 3 (left) that as the number of iterations increases, the number of elements of the Q-table that are filled for the first time decreases. Figure 3 (right), shows the normalized L_1 norm of $(|Q^n| - |Q^{n-1}|)$ where Q^n is the Q-table at iteration n and $Q^0 = 0$. This result shows that although increasing the number of iterations can increase the number of visited states, the newly visited states have much smaller Q-factors, meaning that they are less probable. After 100 iterations, the percentage of newly filled elements of Q-table (compared to the first iteration) and the normalized L_1 difference error is less than 12% and 3% respectively.

C. Q-learning efficiency evaluations

To verify the performance efficiency of our method for each source of complexity (i.e., the large decision state space and the stochastic properties), we create two sets of experiments. The first experiment is based on the assumption that the workload is known a priori. Thus, we train the hybrid system to optimally perform for the particularly known workload. The evaluation results for this experiment reflects the efficiency of our method in handling the large space dimension of the problem. As an example, to find the best policy for U_1, its workload is used to obtain the Q-factors in Algorithm 1.

The second experiment is performed on general realtime workloads. Here, the performance of our method for the stochastic workloads is evaluated. For this purpose, we generate Q-factors using multiple sample workloads; We run algorithm 1 for $M = 10$ sample paths that are the 10 user workloads. The result is a global table of Q-factors. For each user, we exploit the derived table to make decisions based on the incident state observation (s_t) and by using Equation 7.

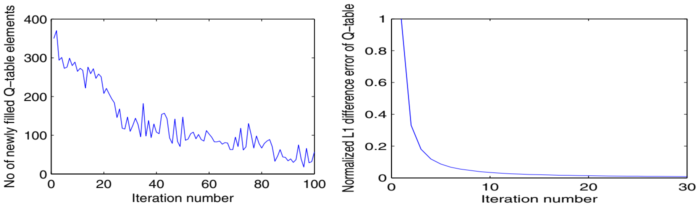

Fig. 3. Convergence of Q-learning.

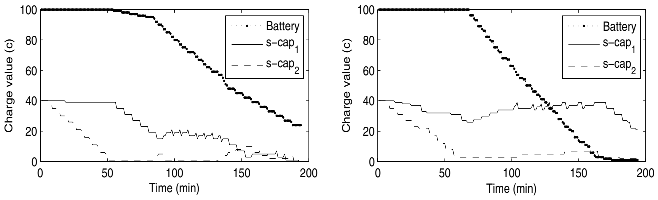

Fig. 4. Charge depletion of battery and s-caps for the same workload. Left: specifically trained policy, Right: global policy.

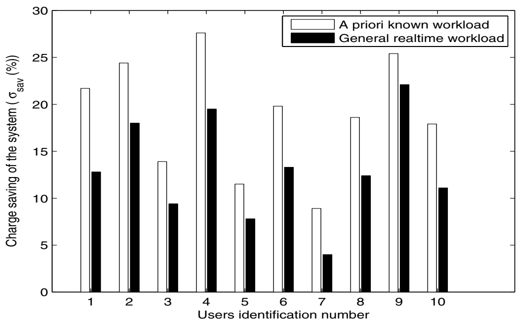

Fig. 5. Energy efficiency ($\sigma_{sav}\%$) of Q-learning for the two experiments.

Figure 4 shows the EES elements charge depletion patterns in the two aforementioned experiments for the same load. On the left, the system is specifically trained for the workload (first experiment) and on the right, a global policy is applied (second experiment). In both cases, the management unit continuously recycles the battery's charge into the s-caps and discharges the s-caps into the load. In the first experiment, the s-caps' are fully discharged into the load, while in the second experiment the battery is totally depleted and s-cap$_1$ has some charges left. The difference is due to the policy optimality when the load is known a priori compared to the case where a global policy is applied based on the incident load.

We define a metric denoted by σ_{sav} to quantitatively measure the energy efficiency of our method. For each workload, we measure the supply's charge depletion in our combined system (denoted by $\Delta\sigma_{bc}$). We also measure the charge depletion associated with those workloads for a battery-only supply system (denoted by $\Delta\sigma_b$), where the battery's capacity is equal to the total capacity of our combined system. The metric is defined as, $\sigma_{sav} = 1 - \frac{\Delta\sigma_{bc}}{\Delta\sigma_b}$. We adopt Lithium-Ion's battery model from [9] to extract the battery's charge consumption for each load considering the rate capacity effect.

Figure 5 shows the final results for the two sets of experiments. As expected, the efficiency is higher for a priori known workloads because the system is best trained for one particular workload. Even in this case, approximating the best solution requires the Q-learning to handle a large space of states and actions. The results verify an average $\sigma_{sav}(\%)$ of 19.0% for the 10 users. However, the results for the general realtime workloads show an average $\sigma_{sav}(\%)$ of 13.1% over the 10 tested loads.

VII. CONCLUSION

We developed a novel power management methodology for combined battery and supercapacitor energy supplies. To provide an optimal management strategy, we posed this problem as a discrete Markov Decision Process (MDP) based upon the EES element characteristics and the stochastic nature of the load. To solve our MDP formulation, we developed a Q-learning method that learns the dynamics of the load and iteratively updates the management policy based upon a specified reward function. Our framework can be easily modified and tuned to different design parameters including the reward function and the number of s-caps. Increasing the number of s-caps allows for a reduction in the overall leakage by lowering the stored energy per s-cap. However, the reasonably low leakage of our s-caps and the overhead cost did not necessitate incorporating more than two s-caps in our studied workload scenarios. We evaluated our approach on real iPhone's workload current measurements. Two scenarios for a priori known workload and a general unknown workload were evaluated. Using the Q-learning algorithm, the results showed an average energy savings of 19% and 13.1% for the two scenarios respectively.

VIII. ACKNOWLEDGEMENT

This research is in part supported by the Office of Naval Research (ONR) YIP award under grant No. R16480 and National Scient Foundation CAREER award under grant No. R3A530.

REFERENCES

[1] F. Koushanfar, "Hierarchical hybrid power supply networks," in *DAC*, 2010, pp. 629–630.

[2] G. Merrett, A. Weddell, A. Lewis, N. Harris, B. Al-Hashimi, and N. White, "An empirical energy model for supercapacitor powered wireless sensor nodes," in *ICCCN*, 2008, pp. 1–6.

[3] D. Brunelli, C. Moser, L. Thiele, and L. Benini, "Design of a solar-harvesting circuit for batteryless embedded systems," *TCAS-I: Regular Papers*, vol. 56, no. 11, pp. 2519–2528, 2009.

[4] T. Zhu, Z. Zhong, Y. Gu, T. He, and Z. Zhang, "Leakage-aware energy synchronization for wireless sensor networks," in *MobiSys*, 2009, pp. 319–332.

[5] C. Watkins, "Learning from delayed rewards," *PhD thesis, Cambridge University*, 1989.

[6] C. Watkins and P. Dayan, "Technical note: Q-learning," *ML Journal*, vol. 8, no. 3–4, pp. 279–292, 1992.

[7] M. Stoller, S. Park, Y. Zhu, J. An, and R. Ruoff, "Graphene-based ultracapacitors," *Nano Letters*, vol. 8, no. 10, pp. 3498–3502, 2008.

[8] L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino, and R. Scarsi, "Discrete-time battery models for system-level low-power design," *TVLSI*, vol. 9, no. 5, pp. 630–640, 2001.

[9] D. Rakhmatov and S. Vrudhula, "Energy management for battery-powered embedded systems," *TECS*, vol. 2, no. 3, pp. 277–324, 2003.

[10] C. Park, J. Liu, and P. Chou, "B#: A battery emulator and power profiling instrument," *IEEE D& T*, vol. 18, no. 2, pp. 150–159, 2005.

[11] T. Yalcinoza and M. Alam, "Improved dynamic performance of hybrid PEM fuel cells and ultracapacitors for portable applications," *IJHE*, vol. 33, no. 7, pp. 1932–1940, 2008.

[12] M. Pedram, N. Chang, Y. Kim, and Y. Wang, "Hybrid electrical energy storage systems," in *ISLPED*, 2010, pp. 363–368.

[13] F. Koushanfar and A. Mirhoseini, "Hybrid heterogeneous energy supply networks," in *ISCAS*, 2011.

[14] A. Mirhoseini and F. Koushanfar, "Hypoenergy: Hybrid supercapacitor-battery power-supply optimization for energy efficiency," in *DATE*, 2011.

[15] D. Bertsekas and J. Tsitsiklis, "Neuro-dynamic programming," 1996.

[16] W. Powell, *Approximate Dynamic Programming*. Wiley, 2007.

[17] C. Shepard, C. A. Rahmati, Tossel, L. Zhong, and P. Kortum, "Live-lab: measuring wireless networks and smartphone users in the field," *HotMetrics*, 2010.