Power Guarantee for Electric Systems Using Real-Time Scheduling

Eugene Kim, Youngmoon Lee, *Student Member, IEEE*, Liang He, *Senior Member, IEEE*, Kang G. Shin, *Life Fellow, IEEE*, and Jinkyu Lee, *Member, IEEE*

Abstract—Modern electric systems, such as electric vehicles, mobile robots, nano satellites, and drones, require to support various power-demand operations for user applications and system maintenance. This, in turn, calls for advanced power management that jointly considers power demand by the operations and power supply from various sources, such as batteries, solar panels, and supercapacitors. In this article, we develop a power scheduling framework for a reliable energy storage system with multiple power-supply sources and multiple power-demand operations. Specifically, we develop *offline power-supply guarantee analysis* and *online power management*. The former provides an offline power-supply guarantee such that every power-demand operation completes its execution in time while the sum of power required by individual operations does not exceed the total power supplied by the entire energy storage system at any time; to this end, we develop a plain power-supply analysis as well as its improved version using real-time scheduling techniques. On the other hand, the latter efficiently utilizes the surplus power available at runtime for improving system performance; we propose two approaches, depending on whether future scheduling information of power-demanding tasks is available or not. For evaluation, we perform simulations to evaluate both the plain and improved analyses for offline power guarantee under various synthetic power-demand operations. In addition, we have built a simulation model and demonstrated that the proposed framework with the offline analysis and online management not only guarantees the required power-supply, but also enhances system performance by up to 56.49 percent.

Index Terms—Offline power-supply guarantee, online power management, real-time scheduling, electric systems

1 Introduction

Recent advances on cyber-physical systems (CPS) require various mechanical/electrical operations for many physical platforms, such as electric vehicles, mobile robots, nano satellites, and drones. For example, a drone needs to (i) operate the motors to fly, (ii) power sensors, coolers, and communication modules, (iii) operate camera(s) and stepper motors to take pictures and deliver parcels, and (iv) perform the computation necessary to maintain its stability (see Fig. 13 in the supplement, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2020.2977041.). These *multiple power-demand operations* impose different power demands on the system and may be triggered at different times, requiring the system to effectively provide time-varying supply of power [1], [2], [3], [4], [5], [6]. Existing studies on the power scheduling problem focused on the reduction of peak power by scheduling multiple power-demand operations and analysis thereof [7], [8], [9], [10].

However, a complete solution to the power scheduling problem also needs to characterize energy storages/sources because it affects power capability for the power-demand operations. We address this need by targeting hybrid energy storage systems (HESSes) comprised of *multiple power-supply sources and storages*, such as batteries, supercapacitors, and renewable energy sources, whose concept and implementation have been proposed and explored in [4], [11], [12], [13], [14], [15], [16]. A combination of high-energy-density batteries and high-power-density supercapacitors enables the system to supply the required time-varying power for a longer time, improving system sustainability. Moreover, renewable power sources (e.g., solar, wind and geothermal energy) enhance the system's power capacity and reduce the load intensity on batteries and supercapacitors, thus prolonging their lifetime [17], [18], [19], [20], [21].

Researchers have studied the optimal design of a HESS to meet the power demands at the minimum cost [12], [22], [23], [24], [25]. They first explored possible storage configurations (i.e., size, type, connection) of multiple power-source supplies and storages, and then selected the best configuration by comparing the performance and the energy production cost. Search for an optimal design iterates to generate candidate configurations until one of the candidates satisfies the pre-defined performance and cost criteria [23]. However, these approaches do not guarantee power-sufficiency during operation, as the electric load and criteria are determined based on the load history or the designer's experience and intuition.

- *E. Kim and K. G. Shin are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109. E-mail: {kimsun, kgshin}@umich.edu.*
- *Y. Lee is with the Department of Robotics, Hanyang University, Ansan-si 15588, Korea. E-mail: youngmoonlee@hanyang.ac.kr.*
- *L. He is with the Department of Computer Science and Engineering, University of Colorado Denver, Denver, CO 80204. E-mail: liang.he@ucdenver.edu.*
- *J. Lee is with the Department of Computer Science and Engineering, Sungkyunkwan University (SKKU), Suwon-si 16419, Republic of Korea. E-mail: jinkyu.lee@skku.edu.*

Manuscript received 27 Mar. 2019; revised 17 Feb. 2020; accepted 23 Feb. 2020. Date of publication 28 Feb. 2020; date of current version 23 Mar. 2020. (Corresponding author: Jinkyu Lee.)
Recommended for acceptance by R. Ge.
Digital Object Identifier no. 10.1109/TPDS.2020.2977041

To ensure the power sufficiency, we develop a power scheduling and analysis framework for a reliable energy storage system with multiple power-supply sources and multiple power-demand operations, achieving the following goals.

G1. *Offline power-supply guarantee:* We provide an offline guarantee to complete every operation before its deadline (i.e., *operation-level power guarantee*) while keeping the amount of power supplied to the entire system no less than the sum of power required by individual operations at any time instant (i.e., *system-level power guarantee*).

G2. *Online power management:* We develop online power management that effectively utilizes the difference between the worst-case supplied/demanded power and the actual one for improving system performance.

This is a typical CPS problem in that we should address power scheduling and its analysis in the cyber space, based on comprehensive understanding of the physical characteristics of power supply and demand.

To achieve G1, we find similarities between the power-supply guarantee problem and a real-time scheduling problem that determines the execution order of real-time tasks; while the operation-level power guarantee corresponds to the task-level deadline satisfaction, the system-level power guarantee matches the computing platform's capacity constraint. Using the techniques of real-time scheduling, we solve the power-supply guarantee problem in two steps. First, we address the case of multiple power-demand operations with a single, uniform power-supply source (e.g., a battery pack), and develop a scheduling framework and offline power guarantee analysis. For the analysis, we develop a plain analysis that only considers each operation's executable interval (from its release to deadline). We then improve the plain analysis by iteratively reclaiming each operation's slack (the gap between the finishing time and the deadline). Second, based on the scheduling and analysis framework, we address the general problem—multiple power-demand operations with multiple power-supply sources. To this end, we develop two scheduling frameworks to utilize additional sporadic power-supply sources: one for sharing additional power-supply sources by all power-demand operations (called the uniform supply approach), and the other for assigning the sources to only some of power-demand operations (called the dedicated supply approach). Our solution for G1 not only demonstrates that the technique of real-time scheduling helps solve a CPS problem, but also addresses the design problem of a HESS by finding a combination of energy storages at minimum cost.

For the evaluation of G1, we synthesize various power-demand operations and compare the power-guarantee performance of the combination of (i) analysis version (i.e., the plain and improved analyses), and (ii) the existence of additional power-supply sources and the framework type for handling them (i.e., no additional supply, uniform and dedicated supply). We also consider two different sets of additional power-supply sources for the uniform and dedicated supply approaches. We have the following observations for the performance of power guarantee. First, the improved analysis dominates the plain one, meaning that the operation set power-guaranteed by the latter is always power-guaranteed by the former. Second, the performance difference between the former and the latter varies with (ii) and the sets of additional power-supply sources. Third, the dedicated supply outperforms the uniform supply for most cases, but not always.

In addition to making the offline power-supply guarantee (G1), we develop online power management (G2), which aims at increasing the utilization of the energy generated by renewable power-supply sources. That is, scheduling multiple power-supply sources and power-demand operations according to G1 necessarily yields surplus energy since G1 considers the worst case, i.e., the largest power demand and the smallest power supply. While we can utilize the surplus energy for improving various performance aspects of a HESS, we focus on reduction of the peak power supplied by the main energy storage (i.e., a battery pack) to improve its capacity and lifetime [15], [24], [26], [27], [28], [29]. We propose two online power management approaches: *static-demand-based* and *predicted-demand-based* approaches. The former operates effectively when the future scheduling information of power-demand operations is known, while the latter does for a more general case where only the past scheduling information is available.

We focus on a case study for a HESS-powered system to evaluate the proposed solutions for G1 and G2 together. The system for the case study consists of power-consuming components such as wheel motors, stepper motors, coolers, sensors and converters, whose operations are controlled by an application. The system is powered by a HESS consisting of Lithium-ion batteries, ultra-capacitors (UCs), and solar panels. Using the offline power-supply guarantee analysis, we can determine the minimum battery capacity without compromising power-supply guarantee. Our experimental results using a simulation model show that the system with our online power management schemes not only supplies a sufficient amount of power to the components during their operation, and but also reduces the battery's peak power by 56.49 percent.

In summary, this paper makes the following main contributions:

- Design of a power scheduling and analysis framework that consists of (i) offline power-supply guarantee, which is the first power guarantee analysis applicable to the design of a HESS, and (ii) online power management for a HESS to enhance the energy storage's performance,
- Demonstration of the effectiveness of the proposed framework via simulations with synthesized power-demand operation sets and a realistic simulation model for a case study, and
- Solution of an important CPS problem using real-time scheduling techniques.

The paper is organized as follows. Section 2 describes the characteristics of power-supply sources and power-demand operations, and then formally states our main problem. Sections 3 and 4 present the power scheduling and analysis framework for the offline power-supply guarantee, while Section 5 details online power management. Section 6 evaluates the offline power-supply guarantee using simulation. Section 7 focuses on a case study and evaluates the power sufficiency and peak power reduction achieved by our solutions. Finally, the paper concludes with Section 8.

2 System Model and Problem Statement

2.1 System Model

As the first step for offline power-supply guarantee and online power management, we need to investigate the characteristics of power-demand operations and power-supply sources. Since our target power-demand operations and power-supply sources are the same as our preliminary conference version [30], we describe them in the supplement, available online, and briefly summarize here.

A system has a set of mechanical/electrical operations $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_{n^{\mathrm{d}}}\}$, where n^{d} is the number of operations.[1] The power usage of λ_i is modeled by the minimum inter-arrival time T_i^{d}, its maximum power consumption P_i^{d}, and the maximum length of execution L_i^{d}.

We consider a battery-centric hybrid energy storage system (HESS) that is comprised of three parts: (i) an *energy-dense* lithium-ion battery pack $\Gamma_{\mathtt{bat}}$, (ii) a set of auxiliary renewable energy sources $\Gamma = \{\Gamma_1, \Gamma_2, \ldots \Gamma_{n^{\mathrm{s}}}\}$, and (iii) a *power-dense* energy buffer $\Gamma_{\mathtt{UC}}$. Each energy source Γ_i is modeled with the maximum inter-arrival time T_i^{s}, the minimum supplied power P_i^{s}, and the minimum length of supply duration L_i^{s}.

Throughout the paper, $P(t)$ implies the actual power demand/supply at t, and P represents the maximum power demand or minimum power supply. Also, we use the term of an "operation" for an "instance" of the operation, when no ambiguity arises.

2.2 Problem Statement

Using the characteristics of power-demand operations and power-supply sources described in Section 2.1, we formally state the problem to be solved as:

Given a set of mechanical/electrical operations λ, the battery pack $\Gamma_{\mathtt{bat}}$, the energy buffer $\Gamma_{\mathtt{UC}}$, a set of auxiliary renewable energy sources Γ, and the operation interval $[0, t_{\max})$,
Determine $P_{\mathtt{UC}}^{\mathrm{s}}(t)$ and $\{P_i^{\mathrm{d}}(t)\}_{i=1}^{n^{\mathrm{d}}}$ for all $t \in [0, t_{\max})$ to achieve the following objective function.

$$\text{Minimize(S0)} \quad \int_0^{t_{\max}} P_{\mathtt{bat}}^{\mathrm{s}}(t) + P_{\mathtt{bat-loss}}^{\mathrm{s}}(t)\, dt,$$

$$\text{Subjectto(S1)} \quad \sum_{\lambda_i \in \lambda} P_i^{\mathrm{d}}(t) - P_{\mathtt{UC}}^{\mathrm{s}}(t) - \sum_{\Gamma_i \in \Gamma} P_i^{\mathrm{s}}(t) = P_{\mathtt{bat}}^{\mathrm{s}}(t) \le P_{\mathtt{bat}}^{\mathrm{s}},$$

$$\text{(S2)} \quad P_{\mathtt{UC}}^{\mathrm{s}}(t) \le P_{\mathtt{UC}}^{\mathrm{s}},$$

$$\text{(S3)} \quad \text{Every instance of each operation } \lambda_i \in \lambda \text{ finishes its execution within } T_i^{\mathrm{d}} \text{ time units after its release.}$$

$\{P_i^{\mathrm{s}}(t)\}_{i=1}^{n^{\mathrm{s}}}$ are not control knobs in that their generated power is immediately stored in $\Gamma_{\mathtt{UC}}$ or served for power-demand operations. $P_{\mathtt{bat}}^{\mathrm{s}}(t)$ is also not, because it is determined once $P_{\mathtt{UC}}^{\mathrm{s}}(t)$ and $\{P_i^{\mathrm{d}}(t)\}_{i=1}^{n^{\mathrm{d}}}$ are determined as shown in S1. For each $P_i^{\mathrm{d}}(t)$, we determine the time instant at which each λ_i starts to execute. When it comes to $P_{\mathtt{UC}}^{\mathrm{s}}(t)$, we determine the amount of supplied power from $\Gamma_{\mathtt{UC}}$ to λ at each time instant t.

$P_{\mathtt{bat-loss}}^{\mathrm{s}}(t)$ in S0 denotes the amount of power dissipation at t caused by the power supply of the battery pack; in other words, we lose $P_{\mathtt{bat-loss}}^{\mathrm{s}}(t)$ of power due to supplying $P_{\mathtt{bat}}^{\mathrm{s}}(t)$ of power from the battery pack at t. Therefore, S0 implies the amount of energy used and dissipated by the battery pack; since the former (i.e., the amount of energy used) is not a control knob, we should reduce the latter (i.e., the amount of energy dissipated) using a simple circuit-based battery model [31]: $P_{\mathtt{bat-loss}}^{\mathrm{s}}(t) = I_{\mathtt{bat}}^2(t) \cdot R_{\mathtt{bat}}(t)$, where $R_{\mathtt{bat}}(t)$ is a battery's internal resistance, and $I_{\mathtt{bat}}(t)$ is the battery's discharge/charge current for supplying $P_{\mathtt{bat}}(t)$. Since power dissipation is quadratically proportional to discharge/charge battery current, the minimization of battery peak current may reduce energy dissipation of the battery pack, potentially extending the battery operation-time.

3 Scheduling of Multiple Power-Demand Operations With a Uniform Supply

In this section, we present how to schedule a set of power-demand operations (λ) when the battery pack ($\Gamma_{\mathtt{bat}}$) is the sole power-supply source. We first develop a scheduling framework that considers the characteristics of λ and $\Gamma_{\mathtt{bat}}$ described in Section 2.1. Under this scheduling framework, we then develop an offline power guarantee analysis that determines whether every operation in λ powered by the battery pack is performed before its deadline without suffering any power shortage. Finally, we derive an improved version of the offline power guarantee analysis, which accommodates significantly more (guaranteed) power-demand operations under the same scheduling framework.

3.1 Scheduling Framework

We would like to schedule a set of multiple operations (λ) so as to achieve a system-level and operation-level power guarantee under the uniform supply from the battery pack $\Gamma_{\mathtt{bat}}$. By "system-level power guarantee," we mean that the sum of power demand at any time instant should be no larger than maximum power capability of the battery pack, i.e., $\sum_{\lambda_i \in \lambda} P_i^{\mathrm{d}}(t) \le P_{\mathtt{bat}}^{\mathrm{s}}$, which is equivalent to S1.[2] To achieve the operation-level power guarantee, every operation should receive sufficient power within its period, which is S3.

Our scheduling framework employs the work-conserving policy based on the worst-case power demand and operation-level fixed-priority scheduling policy. The former implies that an operation λ_k can start to execute as long as its worst-case power demand P_k^{d} (as opposed to the actual power demand $P_k^{\mathrm{d}}(t)$) is no larger than the difference between the battery capability $P_{\mathtt{bat}}^{\mathrm{s}}$ and the sum of the worst-case power demand of currently-executing operations $\sum_{\lambda_i \in Q_{\mathtt{run}}} P_i^{\mathrm{d}}$. The latter implies that the scheduling framework prioritizes the operations that satisfy the above condition.

Since each operation exhibits the non-preemptive behavior as described in Section 2.1, an operation can start its execution only when it is released or the other operation is finished, and each operation, once started, continues its

1. Throughout the paper, we use the superscripts d and s for power demand and power supply, respectively.

2. Since this section considers the sole supply of $\Gamma_{\mathtt{bat}}$, we remove terms of the energy buffer and a set of renewable power-supply sources in S1.

execution until the completion. Therefore, the scheduling framework can be expressed (as in Algorithm 1) by describing actions for the situations. Lines 1–3 update the ready queue $Q_{\texttt{ready}}$ that contains operations ready to execute, and the running queue $Q_{\texttt{run}}$ for currently-executing operations. Lines 4–9 select operations to be started and move them from $Q_{\texttt{ready}}$ to $Q_{\texttt{run}}$. Although the scheduling framework prioritizes operations based on their priorities, a lower-priority operation λ_j in $Q_{\texttt{ready}}$ can start its execution earlier than a higher-priority operation λ_k in $Q_{\texttt{ready}}$, if the remaining power capability of the system (i.e., $P^{\texttt{s}}_{\texttt{bat}} - \sum_{\lambda_i \in Q_{\texttt{run}}} P^{\texttt{d}}_i$) is larger than $P^{\texttt{d}}_k$ but, no larger than $P^{\texttt{d}}_j$, as described in line 5.

Algorithm 1. Scheduling Framework

The following steps are performed whenever at least one mechanical/electrical operation is finished or released at t:

1: For each λ_k finished at t, $Q_{\texttt{run}} \leftarrow Q_{\texttt{run}} \setminus \{\lambda_k\}$
2: For each λ_k released at t, $Q_{\texttt{ready}} \leftarrow Q_{\texttt{ready}} \cup \{\lambda_k\}$
3: Sort $Q_{\texttt{ready}}$ by given operation-level fixed-priorities
4: **for** $\lambda_k \in Q_{\texttt{ready}}$ (a higher priority operation is chosen earlier) **do**
5:   **if** $P^{\texttt{d}}_k \le P^{\texttt{s}}_{\texttt{bat}} - \sum_{\lambda_i \in Q_{\texttt{run}}} P^{\texttt{d}}_i$ **then**
6:     $Q_{\texttt{run}} \leftarrow Q_{\texttt{run}} \cup \{\lambda_k\}$
7:     $Q_{\texttt{ready}} \leftarrow Q_{\texttt{ready}} \setminus \{\lambda_k\}$
8:   **end if**
9: **end for**

3.2 Offline Power Guarantee Analysis

Since each operation is non-preemptive, we need to check if each operation λ_k can start its execution no later than $(T^{\texttt{d}}_k - L^{\texttt{d}}_k)$ time units after its release; once it starts to execute, it completes execution within its period without any preemption. Therefore, we focus on the interval of length $(T^{\texttt{d}}_k - L^{\texttt{d}}_k + \epsilon_t)$, which begins at the time of λ_k's release, and check whether the sum of energy consumed by other operations within the interval is strictly less than $(P^{\texttt{s}}_{\texttt{bat}} - P^{\texttt{d}}_k + \epsilon_p) \cdot (T^{\texttt{d}}_k - L^{\texttt{d}}_k + \epsilon_t)$, where ϵ_t and ϵ_p denote the time and power quantum (the smallest unit). Since the smallest power that prevents λ_k from execution at each time instant is $(P^{\texttt{s}}_{\texttt{bat}} - P^{\texttt{d}}_k + \epsilon_p)$, the above condition guarantees the start of λ_k's execution to be no later than $(T^{\texttt{d}}_k - L^{\texttt{d}}_k)$ time units after its release.

The remaining step is then to calculate $I_{k\leftarrow i}(\ell)$, the amount of energy demanded by instances of λ_i within an interval of length ℓ that contributes to prevention of λ_k from starting its execution. Note that for calculation of $I_{k\leftarrow i}(\ell)$, we limit the power demand at each time instant to $(P^{\texttt{s}}_{\texttt{bat}} - P^{\texttt{d}}_k + \epsilon_p)$, since we need to know whether the sum of all power demands at each time instant is no smaller than $(P^{\texttt{s}}_{\texttt{bat}} - P^{\texttt{d}}_k + \epsilon_p)$. Now, we will describe how to calculate an upper bound of $I_{k\leftarrow i}(\ell)$.

First, if λ_i has a higher priority than λ_k, then the upper-bound of $I_{k\leftarrow i}(\ell)$ is the maximum energy demanded by λ_i in an interval of length ℓ, which is calculated as

$$W_{k\leftarrow i}(\ell) = \min\big(P^{\texttt{d}}_i, P^{\texttt{s}}_{\texttt{bat}} - P^{\texttt{d}}_k + \epsilon_p\big) \times \min\Big(\ell, N_i(\ell) \cdot L^{\texttt{d}}_i + \min\big(L^{\texttt{d}}_i, \ell + T^{\texttt{d}}_i - L^{\texttt{d}}_i - N_i(\ell) \cdot T^{\texttt{d}}_i\big)\Big), \tag{1}$$

where $N_i(\ell) = \lfloor \frac{\ell + T^{\texttt{d}}_i - L^{\texttt{d}}_i}{T^{\texttt{d}}_i} \rfloor$. This calculation is similar to the workload calculation of real-time scheduling [32]. Briefly, $N_i(\ell)$ implies the number of instances of λ_i, and each of their periods is completely included within the interval of length ℓ (including the first instance of λ_i), which contributes $\min(P^{\texttt{d}}_i, P^{\texttt{s}}_{\texttt{bat}} - P^{\texttt{d}}_k + \epsilon_p) \cdot N_i(\ell) \cdot L^{\texttt{d}}_i$ of energy to $W_{k\leftarrow i}(\ell)$. The second part of $W_{k\leftarrow i}(\ell)$ represents the contribution of the operation whose period is partially included in the interval of length ℓ. Fig. 16 in the supplement, available online, shows an example of $W_{k\leftarrow i}(\ell)$ with $N_i(\ell) = 2$, which contributes $\min(P^{\texttt{d}}_i, P^{\texttt{s}}_{\texttt{bat}} - P^{\texttt{d}}_k + \epsilon_p) \cdot 2 \cdot L^{\texttt{d}}_i$ of energy, and the third instance of λ_i contributes $\min(P^{\texttt{d}}_i, P^{\texttt{s}}_{\texttt{bat}} - P^{\texttt{d}}_k + \epsilon_p) \cdot \min(L^{\texttt{d}}_i, \ell + T^{\texttt{d}}_i - L^{\texttt{d}}_i - N_i(\ell) \cdot T^{\texttt{d}}_i)$ of energy to $W_{k\leftarrow i}(\ell)$.

Second, if λ_i has a lower priority than λ_k, then we consider two sub-cases. Since each operation is non-preemptive, λ_i can execute before the start of λ_k's execution, if λ_i starts its execution before the release of λ_k. In this case, the energy demand is upper-bounded by $\min(P^{\texttt{d}}_i, P^{\texttt{s}}_{\texttt{bat}} - P^{\texttt{d}}_k + \epsilon_p) \cdot \min(L^{\texttt{d}}_i - \epsilon_t, \ell)$, which is an upper-bound of $I_{k\leftarrow i}(\ell)$ for the first sub-case of $P^{\texttt{d}}_i \ge P^{\texttt{d}}_k$. If $P^{\texttt{d}}_i < P^{\texttt{d}}_k$, then an upper-bound of $I_{k\leftarrow i}(\ell)$ can be larger than the first sub-case. That is, due to our worst-case-based work-conserving policy, it is possible for λ_i to start its execution before the start of λ_k's execution, whenever λ_k does not satisfy Line 5 of Algorithm 1 but λ_i does. In this case, we use the general upper-bound $W_{k\leftarrow i}(\ell)$ as an upper-bound of the second sub-case.

Combining all the results discussed so far, we develop a power guarantee analysis as follows.

Lemma 1. *Suppose every $\lambda_k \in \lambda$ satisfies Eq. (2). Then, every instance of every operation $\lambda_k \in \lambda$ finishes its execution within its period of length $T^{\texttt{d}}_k$, while guaranteeing the sum of power demands at any time instant is no larger than the battery power capability (i.e.,* S1 *and* S3 *hold).*

$$\sum_{\lambda_i \in \lambda \setminus \{\lambda_k\}} I_{k\leftarrow i}(T^{\texttt{d}}_k - L^{\texttt{d}}_k + \epsilon_t) < (P^{\texttt{s}}_{\texttt{bat}} - P^{\texttt{d}}_k + \epsilon_p) \cdot (T^{\texttt{d}}_k - L^{\texttt{d}}_k + \epsilon_t), \tag{2}$$

where $I_{k\leftarrow i}(\ell) = W_{k\leftarrow i}(\ell)$, if λ_k has a lower-priority than λ_i or $P^{\texttt{d}}_i < P^{\texttt{d}}_k$; $I_{k\leftarrow i}(\ell) = \min(P^{\texttt{d}}_i, P^{\texttt{s}}_{\texttt{bat}} - P^{\texttt{d}}_k + \epsilon_p) \cdot \min(L^{\texttt{d}}_i - \epsilon_t, \ell)$ otherwise.

Proof. As discussed so far, $I_{k\leftarrow i}(T^{\texttt{d}}_k - L^{\texttt{d}}_k + \epsilon_t)$ in Eq. (2) is an upper-bound of the amount of energy demanded by instances of λ_i in an interval of length $(T^{\texttt{d}}_k - L^{\texttt{d}}_k + \epsilon_t)$. Therefore, if Eq. (2) holds, then there exists an instant t_1 within the interval, such that the sum of power demands is less than or equal to $(P^{\texttt{s}}_{\texttt{bat}} - P^{\texttt{d}}_k)$. This means that λ_k can start its execution no later than $(T^{\texttt{d}}_k - L^{\texttt{d}}_k)$ time units after its release, implying that λ_k finishes its execution within its period. □

The lemma works not only for an offline power guarantee for the case of the sole supply, but also for a basis to develop an offline power guarantee for the general case to be discussed in Section 4. Also, the lemma can be used for addressing a design problem: calculation of the minimum capability of the battery cell that can supply given λ by finding the minimum $P^{\texttt{s}}_{\texttt{bat}}$ that satisfies the lemma for given λ.

3.3 Improved Power-Guarantee Analysis

We now develop a new power-guarantee analysis, which is an improved version of the one presented above. We first

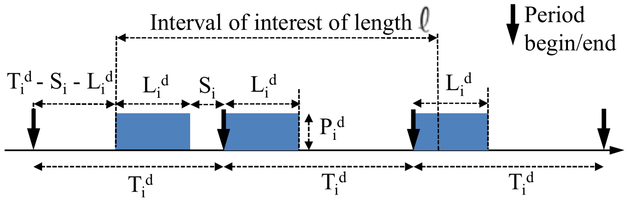


Fig. 1. An example of $W^*_{k\leftarrow i}(\ell, S_i)$ with $N^*_i(\ell, S_i) = 2$.

define a slack of $\lambda_i \in \lambda$ as the minimum time between the finishing time and the deadline of every instance of λ_i, and let S_i denote the slack of λ_i. In other words, if the absolute deadline of an instance of λ_i is t, its execution is completed no later than $(t - S_i)$. By definition, S_i is non-negative. If we use $S_i > 0$, the first instance in the interval of interest for $W_{k\leftarrow i}(\ell)$ cannot execute as late as possible (as shown in Fig. 16 in the supplement, available online); instead, the first instance should finish its execution S_i ahead of its deadline as shown in Fig. 1. We can reduce $W_{k\leftarrow i}(\ell)$ as $W^*_{k\leftarrow i}(\ell, S_i)$ as follows.

$$W^*_{k\leftarrow i}(\ell, S_i) = \min\big(P^{\mathtt{d}}_i, P^{\mathtt{s}}_{\mathtt{bat}} - P^{\mathtt{d}}_k + \epsilon_p\big) \times \min\Big(\ell, N^*_i(\ell, S_i) \cdot L^{\mathtt{d}}_i + \min\big(L^{\mathtt{d}}_i, \ell + T^{\mathtt{d}}_i - S_i - L^{\mathtt{d}}_i - N^*_i(\ell, S_i) \cdot T^{\mathtt{d}}_i\big)\Big), \quad (3)$$

where $N^*_i(\ell, S_i) = \lfloor \frac{\ell + T^{\mathtt{d}}_i - S_i - L^{\mathtt{d}}_i}{T^{\mathtt{d}}_i} \rfloor$. Similarly to $W_{k\leftarrow i}(\ell)$, the physical meaning of $W^*_{k\leftarrow i}(\ell, S_i)$ is maximum energy interference of λ_i on λ_k, which is upper-bounded by the maximum energy demanded by λ_i in an interval of length ℓ with given S_i. Except for S_i, the way to calculate $W^*_{k\leftarrow i}(\ell, S_i)$ is the same as $W_{k\leftarrow i}(\ell)$.

Replacing $W^*_i(\ell, S_i)$ into $W_i(\ell)$, we develop an improved version of the power guarantee analysis stated in the following lemma.

Lemma 2. *Suppose that every $\lambda_k \in \lambda$ satisfies Eq. (4), and S_i for every $\lambda_i \in \lambda$ is given. Then, every instance of every operation $\lambda_k \in \lambda$ finishes its execution within its period of length $T^{\mathtt{d}}_k$, while guaranteeing the sum of power demands at any time instant is no larger than the battery power capability (i.e.,* S1 *and* S3 *hold).*

$$\sum_{\lambda_i \in \lambda \setminus \{\lambda_k\}} I^*_{k\leftarrow i}(T^{\mathtt{d}}_k - L^{\mathtt{d}}_k + \epsilon_t, S_i) < (P^{\mathtt{s}}_{\mathtt{bat}} - P^{\mathtt{d}}_k + \epsilon_p) \cdot (T^{\mathtt{d}}_k - L^{\mathtt{d}}_k + \epsilon_t), \quad (4)$$

where $I^*_{k\leftarrow i}(\ell, S_i) = W^*_{k\leftarrow i}(\ell, S_i)$, if λ_k has a lower-priority than λ_i or $P^{\mathtt{d}}_i < P^{\mathtt{d}}_k$; $I^*_{k\leftarrow i}(\ell) = \min(P^{\mathtt{d}}_i, P^{\mathtt{s}}_{\mathtt{bat}} - P^{\mathtt{d}}_k + \epsilon_p) \cdot \min(L^{\mathtt{d}}_i - \epsilon_t, \ell)$ otherwise.

Proof. By the definition of $W^*_{k\leftarrow i}(\ell, S_i)$, $I^*_{k\leftarrow i}(T^{\mathtt{d}}_k - L^{\mathtt{d}}_k + \epsilon_t, S_i)$ in Eq. (4) is an upper-bound of the amount of energy demanded by instances of λ_i in an interval of length $(T^{\mathtt{d}}_k - L^{\mathtt{d}}_k + \epsilon_t)$ when the slack of λ_i is S_i. The remainder of the proof is the same as Lemma 1. □

For power guarantee, Lemma 2 misses one important issue—how to calculate S_i. Using Eq. (4), we can calculate S_i as follows.

Lemma 3. *The slack of $\lambda_k \in \lambda$ (denoted by S_k) can be lower-bounded as:*

$$S_k \geq T^{\mathtt{d}}_k - L^{\mathtt{d}}_k - \frac{\sum_{\lambda_i \in \lambda \setminus \{\lambda_k\}} I^*_{k\leftarrow i}(T^{\mathtt{d}}_k - L^{\mathtt{d}}_k + \epsilon_t, S_i)}{P^{\mathtt{s}}_{\mathtt{bat}} - P^{\mathtt{d}}_k + \epsilon_p}. \quad (5)$$

Proof. The lemma holds by Eq. (4). That is, Eq. (4) can be rephrased as:

$$\frac{\sum_{\lambda_i \in \lambda \setminus \{\lambda_k\}} I^*_{k\leftarrow i}(T^{\mathtt{d}}_k - L^{\mathtt{d}}_k + \epsilon_t, S_i)}{P^{\mathtt{s}}_{\mathtt{bat}} - P^{\mathtt{d}}_k + \epsilon_p} < T^{\mathtt{d}}_k - L^{\mathtt{d}}_k + \epsilon_t. \quad (6)$$

Then, the physical meaning of the left-hand side of Eq. (6) is the maximum interval length that other tasks than λ_k prevent λ_k from execution. This means that the interval length between λ_k's release and finishing times is at most as much as the sum of the left-hand side of Eq. (6) (i.e., interference from other operations than λ_k) and $L^{\mathtt{d}}_k$ (i.e., the execution time of λ_k itself). Therefore, the slack value of λ_k is at least as much as the right-hand side of Eq. (5). □

The overall process on how to incorporate Lemma 3 into Lemma 2 is the same as the slack reclamation of real-time scheduling [33]. That is, in the first iteration, we apply Lemma 2 with $S_k = 0$ for every $\lambda_k \in \lambda$, and update $S_k > 0$ for every $\lambda_k \in \lambda$ using Lemma 3. In the second iteration, we repeat the same process with the new slacks. This process is repeated until every operation satisfies Eq. (4) (achieving the power guarantee) or there is no more slack update (failing the power guarantee).

In terms of power guarantee, it is trivial to show that the improved version (Lemma 2 with Lemma 3) is always better performance than Lemma 1. That is because Lemma 2 with $S_k = 0$ for every $\lambda_k \in \lambda$ is equivalent to Lemma 1. On the other hand, the improved version incurs high time-complexity due to some iteration. Due to the iteration, the time-complexity of the slack reclamation is known to be pseudo-polynomial [33]. Since we are interested in "offline" power guarantees, the pseudo-polynomial time-complexity is acceptable for most cases. In Section 6, we will compare the performance of the vanilla (simple) and improved versions of the proposed offline power guarantee.

4 Scheduling of Multiple Power-Demand Operations With Multiple Power-Supply Sources

This section addresses a more general situation than Section 3, in which additional power is sporadically generated from multiple power sources such as an RBS and a solar panel, and immediately stored in the energy buffer or used for power-demand operations. We will first address a scheduling challenge due to the existence of sporadic additional power supply. Second, we present two approaches, depending on how to distribute the additional power supply to power-demand operations. Finally, we describe how to incorporate the two approaches in the proposed offline power-guarantee analysis.

4.1 A Scheduling Challenge

Unlike the situation where a battery pack is the only power-supply source discussed in Section 3, a straightforward approach cannot yield a system-level power guarantee, as shown in the following example.

Example 1. Suppose that additional power is supplied by Γ_1 in $[t_1, t_2)$, while the battery pack is the only supply in $[t_0, t_1)$ and $[t_2, t_3)$, where $t_0 < t_1 < t_2 < t_3$, as shown in Fig. 17a in the supplement, available online. Also, there are two operations λ_1 and λ_2 ready to execute at t_0, and $\lambda_1 \leq P^{\mathsf{s}}_{\mathtt{bat}}$ and $\lambda_1 + \lambda_2 > P^{\mathsf{s}}_{\mathtt{bat}}$, as shown in Fig. 17b in the supplement, available online. Suppose that we apply the worst-case-based work-conserving policy in Algorithm 1, implying we start execution of an operation in the ready queue as long as the system has enough remaining power supply to accommodate the worst-case power demand of the operation. Then, λ_2 can start its execution at t_1, but there is a problem at t_2, at which power supplied by Γ_1 ends. In $[t_2, t_3)$, the total amount of power demand is strictly larger than that of power supply, entailing either eviction of one of "non-preemptive" operations, or risking power shortage in executing operations, both of which are considered as a system failure.

Example 1 shows the need for a more fine-grained way to handle additional sporadic power-supply sources. To meet this need, we consider two policies depending on how to distribute the additional supply as follows.

- Calculate the additional "guaranteed" uniform supply $P^{\mathsf{s}}_{\mathtt{uni}}$, meaning that additional power-supply sources (and the energy buffer) can *always* provide power as much as $P^{\mathsf{s}}_{\mathtt{uni}}$ (as the battery pack provides up to $P^{\mathsf{s}}_{\mathtt{bat}}$). This entails the calculation of $P^{\mathsf{s}}_{\mathtt{uni}}$; once it is calculated, we can reuse the power scheduling and analysis framework presented in Section 3, by adding $P^{\mathsf{s}}_{\mathtt{uni}}$ to the existing uniform supply $P^{\mathsf{s}}_{\mathtt{bat}}$. In this case, all operations can share the power generated by additional power-supply sources.
- Assign additional power to a partial set of operations. Power generated by additional power-supply sources (and stored in the energy buffer) is used only when the operations in the partial set are executed. This entails the way to divide power generated by additional power-supply sources for individual power-demand operations.

In what follows, we will detail the above two approaches, including their scheduling frameworks.

4.2 Uniform Supply

In this approach, we calculate the additional guaranteed uniform supply $P^{\mathsf{s}}_{\mathtt{uni}}$ from additional power-supply sources. After calculating $P^{\mathsf{s}}_{\mathtt{uni}}$, we can reuse the scheduling framework in Algorithm 1. That is, we just change the $P^{\mathsf{s}}_{\mathtt{bat}}$ term in Line 5 to $P^{\mathsf{s}}_{\mathtt{bat}} + P^{\mathsf{s}}_{\mathtt{uni}}$. The main issue of this approach is to accurately calculate $P^{\mathsf{s}}_{\mathtt{uni}}$; the larger $P^{\mathsf{s}}_{\mathtt{uni}}$, the more operations to be accommodated.

The basic idea to obtain $P^{\mathsf{s}}_{\mathtt{uni}}$ is to calculate the amount of the minimum supplied energy in $[0, t)$ by considering the fact that the energy buffer has at least $P^{\mathsf{s}}_{\mathtt{UC}} \cdot L^{\mathsf{s}}_{\mathtt{UC}}$ of energy at 0 and each additional supply generates power at the end of its instances' periods. If we divide this amount by t and take the minimum, we guarantee to supply power as much as $P^{\mathsf{s}}_{\mathtt{uni}}$ in $[0, t)$, as stated in the following lemma.

Lemma 4. *We can calculate $P^{\mathsf{s}}_{\mathtt{uni}}$ using the following equation.*

$$P^{\mathsf{s}}_{\mathtt{uni}} = \min_{0 \leq t \leq \mathsf{LCM}} \frac{f(\Gamma_{\mathtt{UC}}, t) + \sum_{\Gamma_i \in \Gamma} f(\Gamma_i, t)}{t}, \tag{7}$$

where LCM *is the least common multiple of $\{T^{\mathsf{s}}_i\}_{\Gamma_i \in \Gamma}$,*

$$f(\Gamma_{\mathtt{UC}}, t) = P^{\mathsf{s}}_{\mathtt{UC}} \cdot \min(t, L^{\mathsf{s}}_{\mathtt{UC}}), \text{and} \tag{8}$$

$$f(\Gamma_i, t) = P^{\mathsf{s}}_i \cdot \left\lfloor \frac{t}{T^{\mathsf{s}}_i} \right\rfloor \cdot L^{\mathsf{s}}_i + P^{\mathsf{s}}_i \cdot \max\left(0, t - \left\lfloor \frac{t}{T^{\mathsf{s}}_i} \right\rfloor \cdot T^{\mathsf{s}}_i - (T^{\mathsf{s}}_i - L^{\mathsf{s}}_i)\right). \tag{9}$$

Proof. Since the amount of energy in the energy buffer at $t = 0$ is at least $P^{\mathsf{s}}_{\mathtt{UC}} \cdot L^{\mathsf{s}}_{\mathtt{UC}}$, the amount of the supplied energy from the energy buffer in $[0, t)$ is $P^{\mathsf{s}}_{\mathtt{UC}} \cdot t$ if $t \leq L^{\mathsf{s}}_{\mathtt{UC}}$, and at least $P^{\mathsf{s}}_{\mathtt{UC}} \cdot L^{\mathsf{s}}_{\mathtt{UC}}$ otherwise, which is recorded in $f(\Gamma_{\mathtt{UC}}, t)$ of Eq. (8). For given t, $\lfloor \frac{t}{T^{\mathsf{s}}_i} \rfloor$ means the number of instances of λ_i whose periods are completely included in $[0, t)$, and each instance generates energy no smaller than $P^{\mathsf{s}}_i \cdot L^{\mathsf{s}}_i$. The second term of Eq. (9) presents the minimum energy generated by the last instance whose period is partially included in $[0, t)$. Therefore, $\sum_{\Gamma_i \in \Gamma} f(\Gamma_i, t)$ represents the amount of generated energy by Γ in $[0, t)$. Since we assume that the capacity of the energy buffer is sufficiently large, we can always use power as much as the lower-bound of $\frac{f(\Gamma_{\mathtt{UC}}, t) + \sum_{\Gamma_i \in \Gamma} f(\Gamma_i, t)}{t}$ for $0 \leq t \leq \mathsf{LCM}$. □

Note that the additional power supply should be stored in the energy buffer $\Gamma_{\mathtt{UC}}$, and hence $P^{\mathsf{s}}_{\mathtt{uni}}$ cannot be larger than $P^{\mathsf{s}}_{\mathtt{UC}}$.

If LCM is very large or time-complexity is critically important, e.g., for online admission control for operations, we need a tractable way to calculate $P^{\mathsf{s}}_{\mathtt{uni}}$, which is covered in the following lemma.

Lemma 5. *$P^{\mathsf{s}}_{\mathtt{uni}}$ is at least as much as the minimum of $P^{\mathsf{s}}_{\mathtt{UC}}$ and $\sum_{\Gamma_i \in \Gamma'} \frac{P^{\mathsf{s}}_i \cdot L^{\mathsf{s}}_i}{T^{\mathsf{s}}_i}$, if Γ' satisfies Eq. (10).*

$$\sum_{\Gamma_i \in \Gamma'} (T^{\mathsf{s}}_i - L^{\mathsf{s}}_i) \cdot \frac{P^{\mathsf{s}}_i \cdot L^{\mathsf{s}}_i}{T^{\mathsf{s}}_i} \leq P^{\mathsf{s}}_{\mathtt{UC}} \cdot L^{\mathsf{s}}_{\mathtt{UC}}. \tag{10}$$

We will describe how to find Γ' after the lemma.

Proof. Since $\Gamma^{\mathsf{s}}_{\mathtt{UC}}$ can supply power at most $P^{\mathsf{s}}_{\mathtt{UC}}$, $P^{\mathsf{s}}_{\mathtt{uni}}$ should be no larger than $P^{\mathsf{s}}_{\mathtt{UC}}$. Otherwise, there should exist some Γ_i that always generates power, which is impossible because we cannot control if each Γ_i generates power.

While $\frac{P^{\mathsf{s}}_i \cdot L^{\mathsf{s}}_i}{T^{\mathsf{s}}_i}$ is the average rate of power generation by Γ_i, it cannot entirely contribute $P^{\mathsf{s}}_{\mathtt{uni}}$. This is, because there is no power generation in $[0, T^{\mathsf{s}}_i - L^{\mathsf{s}}_i)$. To supply $\frac{P^{\mathsf{s}}_i \cdot L^{\mathsf{s}}_i}{T^{\mathsf{s}}_i}$-rate power in $[0, T^{\mathsf{s}}_i - L^{\mathsf{s}}_i)$, we need to use $(T^{\mathsf{s}}_i - L^{\mathsf{s}}_i) \cdot \frac{P^{\mathsf{s}}_i \cdot L^{\mathsf{s}}_i}{T^{\mathsf{s}}_i}$ amount of power from $\Gamma_{\mathtt{UC}}$. Then, the generated energy from Γ_i in $[x \cdot T^{\mathsf{s}}_i - L^{\mathsf{s}}_i, x \cdot T^{\mathsf{s}}_i)$ (as much as $P^{\mathsf{s}}_i \cdot L^{\mathsf{s}}_i$) enables to supply $\frac{P^{\mathsf{s}}_i \cdot L^{\mathsf{s}}_i}{T^{\mathsf{s}}_i}$-rate power in $[x \cdot T^{\mathsf{s}}_i - L^{\mathsf{s}}_i, (x+1) \cdot T^{\mathsf{s}}_i - L^{\mathsf{s}}_i)$. Therefore, the LHS of Eq. (10) is an upper-bound of the amount of energy for every operation $\lambda_i \in \lambda'$ to contribute $\frac{P^{\mathsf{s}}_i \cdot L^{\mathsf{s}}_i}{T^{\mathsf{s}}_i}$ to $P^{\mathsf{s}}_{\mathtt{uni}}$.

Since the amount of power in $\Gamma_{\mathtt{UC}}$ at $t=0$ is $P^{\mathrm{s}}_{\mathtt{UC}} \cdot L^{\mathrm{s}}_{\mathtt{UC}}$, we enforce the constraint Eq. (10). □

To calculate $P^{\mathrm{s}}_{\mathtt{uni}}$ using Lemma 5, we introduce a heuristic to find Γ' that satisfies Eq. (10). Staring from $\Gamma' = \emptyset$, we repeat to add an operation in $\Gamma \setminus \Gamma'$ to Γ' until Eq. (10) is satisfied. Here, the order to be selected from $\Gamma \setminus \Gamma'$ is based on the following criteria (the larger value, the earlier selection):

$$\frac{\frac{P^{\mathrm{s}}_i \cdot L^{\mathrm{s}}_i}{T^{\mathrm{s}}_i}}{\frac{T^{\mathrm{s}}_i - L^{\mathrm{s}}_i}{T^{\mathrm{s}}_i} \cdot P^{\mathrm{s}}_i \cdot L^{\mathrm{s}}_i} = \frac{1}{T^{\mathrm{s}}_i - L^{\mathrm{s}}_i}. \tag{11}$$

We use the above criteria because a larger value of $\frac{1}{T^{\mathrm{s}}_i - L^{\mathrm{s}}_i}$ means more contribution of $P^{\mathrm{s}}_{\mathtt{uni}}$ (i.e., denominator) per usage of $P^{\mathrm{s}}_{\mathtt{UC}} \cdot L^{\mathrm{s}}_{\mathtt{UC}}$ (i.e., numerator).

We will describe how to use $P^{\mathrm{s}}_{\mathtt{uni}}$ in Section 4.4.

4.3 Dedicated Supply Approach

In this approach, we can determine λ^{ded}, a set of operations completely powered by a set of additional power-supply sources Γ and the energy buffer $\Gamma_{\mathtt{UC}}$. Once we determine λ^{ded}, operations in $\lambda \setminus \lambda^{\mathrm{ded}}$ can be executed according to Algorithm 1, and their power guarantee is judged by Lemma 1 with $\lambda \setminus \lambda^{\mathrm{ded}}$. On the other hand, each operation in λ^{ded} is fully supplied by Γ with $\Gamma_{\mathtt{UC}}$, and does not use power from $\Gamma_{\mathtt{bat}}$. Our policy is to execute each operation in λ^{ded} at the end of each period, which accommodates more operations in λ^{ded} (because this policy uses less initial energy from the energy buffer). Formally, $\lambda_k \in \lambda^{\mathrm{ded}}$ starts its execution at $r + T^{\mathrm{d}}_k - L^{\mathrm{d}}_k$, where r is the release time of an instance of λ_k, and $\Gamma_{\mathtt{UC}}$ supplies $P^{\mathrm{d}}_k(t)$ $(\le P^{\mathrm{d}}_k)$ amount of power to λ_k in $[r + T^{\mathrm{d}}_k - L^{\mathrm{d}}_k, r + T^{\mathrm{d}}_k)$.

Then, the remaining step is to determine λ^{ded}. The basic idea is to calculate the maximum energy demanded by λ^{ded} in $[0,t)$ and the minimum energy supplied by Γ and $\Gamma_{\mathtt{UC}}$ in $[0,t)$. We check whether the former is not larger than the latter at all times, which is stated in the following lemma.

Lemma 6. *Every instance of every operation $\lambda_k \in \lambda_{\mathtt{ded}}$ finishes its execution at the end of each period (e.g., $[r + T^{\mathrm{d}}_k - L^{\mathrm{d}}_k, r + T^{\mathrm{d}}_k)$ where r is the release time of an instance of λ_k), only with Γ and $\Gamma_{\mathtt{UC}}$, if the following inequality holds for all $t \in [0, \mathsf{LCM})$.*

$$\sum_{\lambda_i \in \lambda^{\mathrm{ded}}} f(\lambda_i, t) \le f(\Gamma_{\mathtt{UC}}, t) + \sum_{\Gamma_i \in \Gamma} f(\Gamma_i, t), \text{and} \tag{12}$$

$$f(\lambda_i, t) = P^{\mathrm{d}}_i \cdot \left\lfloor \frac{t}{T_i} \right\rfloor \cdot L^{\mathrm{d}}_i + P^{\mathrm{d}}_i \cdot \max\left(0, t - \left\lfloor \frac{t}{T^{\mathrm{d}}_i} \right\rfloor \cdot T^{\mathrm{d}}_i - (T^{\mathrm{d}}_i - L^{\mathrm{d}}_i)\right). \tag{13}$$

Proof. Since $f(\lambda_i, t)$ in Eq. (13) exhibits the same formula as Eq. (9), it calculates the maximum energy demanded by λ_i in $[0,t)$ when its instances are executed at the end of their periods. Therefore, the LHS (Left-Hand Side) of Eq. (12) is the maximum energy demanded by λ^{ded} in $[0,t)$. On the other hand, the RHS (Right-Hand Side) of the equation is the minimum energy supplied by Γ and $\Gamma_{\mathtt{UC}}$ in $[0,t)$ as explained in Lemma 4. Therefore, the lemma follows. □

If time-complexity is critical, we can use another necessary condition presented in the following lemma (likewise Lemma 5).

Lemma 7. *Every instance of every operation $\lambda_k \in \lambda_{\mathtt{ded}}$ finishes its power execution at the end of each period (i.e., $[r + T^{\mathrm{d}}_k - L^{\mathrm{d}}_k, r + T^{\mathrm{d}}_k)$ where r is the release time of an instance of λ_k), only with Γ and $\Gamma_{\mathtt{UC}}$ (without $\Gamma_{\mathtt{bat}}$), if the following inequality holds:*

$$\sum_{\lambda_i \in \lambda^{\mathrm{ded}}} \frac{P^{\mathrm{d}}_i \cdot L^{\mathrm{d}}_i}{T^{\mathrm{d}}_i} \le P^{\mathrm{s}}_{\mathtt{uni}} \text{ in Lemma 4 (or Lemma 5).} \tag{14}$$

Proof. By the meaning of $P^{\mathrm{s}}_{\mathtt{uni}}$, Γ and $\Gamma_{\mathtt{UC}}$ can *always* supply $P^{\mathrm{s}}_{\mathtt{uni}}$-rate power. If we enforce each operation $\lambda_k \in \lambda_{\mathtt{ded}}$ executes its instances at the end of their periods, their cumulative demand is always not larger than $P^{\mathrm{s}}_{\mathtt{uni}}$-rate power. Therefore, as long as their total demanded power rate (i.e., the LHS of Eq. (14)) is no larger than $P^{\mathrm{s}}_{\mathtt{uni}}$, every instance of every operation $\lambda_k \in \lambda_{\mathtt{ded}}$ finishes its execution within its period of length T^{d}_k, only with Γ and $\Gamma_{\mathtt{UC}}$. □

The remaining problem is then how to select λ^{ded} that satisfies Lemma 6 (or Lemma 7). Here we describe a simple, but effective heuristic. We sort $\lambda_i \in \lambda$, based on the ratio of the LHS to the RHS of Eq. (2). If the ratio of λ_i is larger than that of λ_k, we interpret that λ_i is more difficult to satisfy Eq. (2) than λ_j. Therefore, starting from $\lambda^{\mathrm{ded}} = \emptyset$, we repeat the following step until there is no operation to be moved: we select an operation λ_j with the largest ratio among operations in $\lambda \setminus \lambda^{\mathrm{ded}}$ such that $\lambda^{\mathrm{ded}} \cup \{\lambda_j\}$ satisfies Lemma 6 (or Lemma 7), and then add λ_j to λ^{ded}.

Another simple strategy for selecting λ^{ded} is the lower-priority-first strategy. That is, as we use fixed-priority scheduling, the lower-priority task tends to violate Eq. (2) (as well as Eq. (4)). Starting from $\lambda^{\mathrm{ded}} = \emptyset$, we scan all $\lambda_j \in \lambda$ from the lowest-priority to highest-priority operations, and check Lemma 6 (or Lemma 7) is satisfied if λ_j is included in λ^{ded}. If the lemma satisfied, we add λ_j to λ^{ded}; otherwise, we do not add.

4.4 Incorporating Additional Supply to the Proposed Analysis

Section 4.2 calculates $P^{\mathrm{s}}_{\mathtt{uni}}$, the additional guaranteed uniform supply by additional power-supply sources, and Section 4.3 calculates λ^{ded}, a set of operations completely powered by a set of additional power-supply sources. The uniform and dedicated approaches utilize $P^{\mathrm{s}}_{\mathtt{uni}}$ and λ^{ded}, respectively, and improve the offline power guarantee as follows.

First, when it comes to the uniform supply, we can check a power guarantee of this approach by applying Lemma 1 (or Lemmas 2 and 3) for all $\lambda_k \in \lambda$ and replacing $P^{\mathrm{s}}_{\mathtt{bat}}$ with $P^{\mathrm{s}}_{\mathtt{bat}} + P^{\mathrm{s}}_{\mathtt{uni}}$. Second, as to the dedicated supply, we can check the power guarantee by checking Lemma 1 (or Lemma 2) only with $\lambda_i \in \lambda \setminus \lambda^{\mathrm{ded}}$. Note that for $\lambda_i \in \lambda^{\mathrm{ded}}$, we automatically guarantee their power sufficiency in that λ^{ded} is constructed so as to supply all power demands in λ^{ded} by Γ and $\Gamma_{\mathtt{UC}}$ without $\Gamma_{\mathtt{bat}}$.

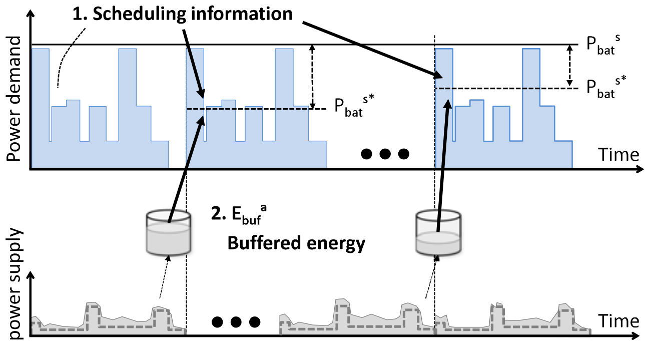


Fig. 2. Static-demand-based online power management: Additional buffered energy can be calculated based on buffer voltage. Future power demand scheduling information and the buffered energy amount are used to determine the buffer power.

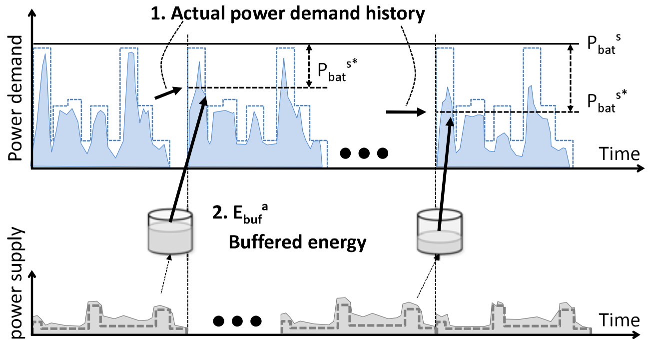


Fig. 3. Predicted-demand-based online power management: Power demand history and the buffered energy amount are used to determine the buffer power.

5 Online Power Management

Thus far, we have explored an offline analysis. The offline analysis provides the system-level power guarantee in our scheduling framework based on power-demand and supply task information. We have also described how the sporadic power supplies can improve the power-guarantee analysis. However, if one performs a set of power-demand operations with a set of power-supply sources that satisfy the offline power guarantee analysis in Section 4, there will be extra energy stored in the energy buffer as the analysis is based on the minimum (not actual) power supply. Thus, we propose an online power management framework, which adaptively controls the power of the energy buffer based on scheduling information. The goal of the online management is to reduce the peak power supplied by the battery while guaranteeing system-power capability that achieves the goal in Section 2.2; the online power management framework determines a desirable battery current bound to achieve the peak power reduction. Then, whenever battery current exceeds the current bound, the system controls the energy buffer to supply the stored energy to the system load so as to limit the battery current. The energy system must be able to supply the required power within its system configurations and energy buffer states.

In this section, we have explored two different approaches to determine the battery current bound. The first approach (called *static-demand-based online power management*) assumes that we can predict future scheduling information of power-demanding tasks and use the information to determine the limits as shown in Fig. 2. For the situation where future power-demand information is not available, the second approach (called *predicted-demand-based online power management*) uses past power requirement information only as seen in Fig. 3; it keeps calculating average and standard deviation of the past power requirement, and determines the battery current bound.

5.1 Static-Demand-Based Online Power Management

The static-demand-based online power management framework periodically controls power of the energy buffer, as shown in Fig. 2 and Algorithm 2. At t_0, the beginning of each period (for online power management) of length t_{p}, we calculate the amount of energy in Γ_{UC}, which is necessary for an offline power guarantee for the current period $[t_0, t_0+t_{\text{p}})$ (denoted by $E^{\text{m}}_{\text{buf}}(t_0)$) as follows. For the uniform power-supply approach in Section 4.2, $E^{\text{m}}_{\text{buf}}(t_0)$ is simply calculated by $t_{\text{p}} \cdot P^{\text{s}}_{\text{uni}}$ since the offline power guarantee exploits the property that $P^{\text{s}}_{\text{uni}}$ of power is *always* supplied by Γ. For the dedicated supply approach in Section 4.3, we calculate $E^{\text{m}}_{\text{buf}}(t_0)$ using the amount of energy consumed by λ^{ded} based on their maximum power demand parameters (i.e., P^{d}_i, not $P^{\text{d}}_i(t)$) during the interval.

Once $E^{\text{m}}_{\text{buf}}(t_0)$ is calculated, we can utilize the energy from Γ_{UC} up to the difference between the amount of total energy stored in Γ_{UC} at t_0 (denoted by $E_{\text{buf}}(t_0)$) and $E^{\text{m}}_{\text{buf}}(t_0)$. Note that we can measure the amount of energy in Γ_{UC} by monitoring the voltage level of Γ_{UC} at t_0 (denoted by $V_{\text{buf}}(t_0)$), using $E_{\text{buf}}(t_0) = \frac{1}{2} \cdot C_{\text{buf}} \cdot V^2_{\text{buf}}(t_0)$, where C_{buf} is a constant representing the capacitance of Γ_{UC} [15]. Lines 1 and 2 of Algorithm 2 represent the calculation of $E^{\text{m}}_{\text{buf}}(t_0)$ and $E_{\text{buf}}(t_0)$.

Then, we utilize the extra energy up to as much as $E_{\text{buf}}(t_0) - E^{\text{m}}_{\text{buf}}(t_0)$, for reducing the battery's peak power. We use energy from the energy buffer if the power usage of the battery pack is larger than a threshold $P^{\text{s*}}_{\text{bat}}(t_0)$. We determine $P^{\text{s*}}_{\text{bat}}(t_0)$ for the current period $[t_0, t_0+t_{\text{p}})$, using the scheduling results of the previous period $[t_0 - t_{\text{p}}, t_0)$, which is the worst-case energy consumption by λ during that period. During each scheduling period, we can figure out how much can be stored in a buffer by $P^{\text{s*}}_{\text{bat}}(t_0)$ as follows.

$$f^{t_i}_{\text{P2E}}(P_{\text{s}}) = \begin{cases} -t_i \cdot P_{\text{s}} + t_i \cdot P_i, & \text{if } P^{\text{s*}}_{\text{bat}} < P_i, \\ 0, & \text{otherwise,} \end{cases}$$

where P_i is a power consumption during a sub-period $t_i \in [t_0 - t_{\text{p}}, t_0)$ as seen in Fig. 4. Note that P_i and t_i are not typically the same as P and t of any task.

Then, sum of $f^{t_i}_{\text{P2E}}(P_{\text{s}})$ represents total stored energy in the buffer because of the power threshold ($P^{\text{s*}}_{\text{bat}}$) as shown in Fig. 4. Next, $f_{\text{E2P}}(E)$, the function which finds $P^{\text{s*}}_{\text{bat}}$ to store the amount of buffer energy (E), can be derived by applying the inverse function of $\Sigma_{i=1,\dots,N} f^{t_i}_{\text{P2E}}(P_{\text{s}})$.

Finally, for given $E^{\text{m}}_{\text{buf}}(t_0)$ and $P^{\text{s*}}_{\text{bat}}(t_0)$, the online power management framework controls the energy stored in the energy buffer within a period $[t_0, t_0+t_{\text{p}})$ as follows. Suppose that the battery pack should supply X amount of power if there is no supply from Γ_{UC} at t for the purpose of the peak power reduction.[3] Then, the energy buffer supplies $X - P^{\text{s*}}_{\text{bat}}(t_0)$ of power, only when $X \geq P^{\text{s*}}_{\text{bat}}(t_0)$ and $E_{\text{buf}}(t) \geq E^{\text{m}}_{\text{buf}}(t_0)$ hold.

3. The amount of energy Γ_{UC} (the energy buffer) should supply at t for an offline power guarantee already figured in X.

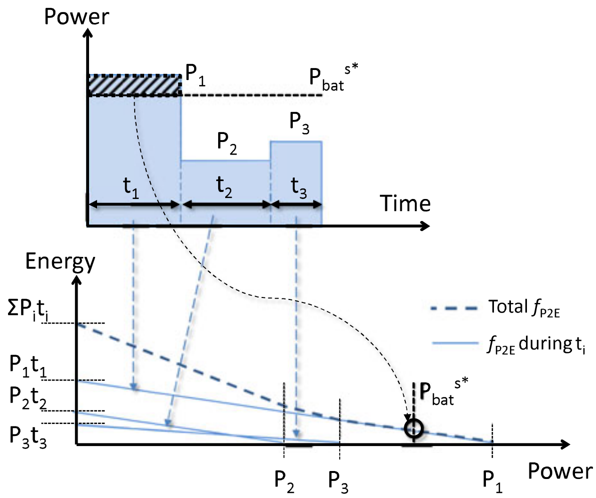


Fig. 4. Calculation of P_{bat}^{s*} using $f_{\text{E2P}}(E)$: This relation can be extracted based on power demand scheduling information.

Algorithm 2. Static-Demand-Based Online Power Management

The following steps are performed at t_0, the beginning of each period of length t_p,

1: Calculate $E_{\text{buf}}^{m}(t_0)$ depending on the uniform/dedicated supply approach.
2: Calculate $E_{\text{buf}}(t_0)$ by $\frac{1}{2} \cdot C_{\text{buf}} \cdot V_{\text{buf}}^2(t_0)$.
3: $P_{\text{bat}}^{s*}(t_0) \leftarrow f_{\text{E2P}}(E_{\text{buf}}(t_0) - E_{\text{buf}}^{m}(t_0))$

5.2 Predicted-Demand-Based Online Power Management

The goal of online power management is to minimize the battery (dis)charge stress by determining effective (dis)charge current limits in real time. The predicted-demand-based approach to be described in this subsection determines the limits based on the past power requirements and the online buffer energy state. The normal distribution of power requirements has been used to identify the past power requirements patterns. We adapt the formulations and ideas of the authors' recent conference paper [34] for this approach. We first formally state the problem as follows.

Given the power requirement $(I_{rq}(t))$,

Determine the (dis)charge currents from the energy buffer $I_L(t)$, such that

$$\begin{aligned} &\text{Minimize(S4)} \quad \text{Discharge/charge stress} = \frac{1}{T_{op}} \int a \cdot e^{b \cdot |I_b(t)|}\, dt, \\ &\text{Subjectto(S5)} \quad I_{rq}(t) = I_L(t) + I_b(t), \end{aligned} \tag{15}$$

where I_L is a buffer current, a and b are the coefficients in the battery stress model. The energy storage system must supply the required power with current (I_{rq}) to the electric load. As energy conserves, the sum of current supplies from energy storage components (I_L, I_b) must be the power requirement for all $k \in [0, k_e)$ (S5). The objective function in (S4) captures the (dis)charge stress during an operation period of T_{op}.

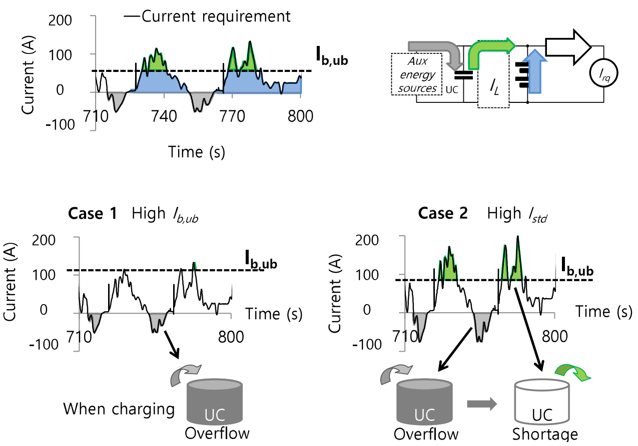


Fig. 5. Example of power buffering for reducing the peak power. Case 1 shows the UC state during charging and discharging if $I_{b,ub}$ and $I_{b,lb}$ are high. Large current bounds limit the discharging current from UC and increase the charging current to UC. Case 2 shows the UC state if actual peak currents are higher (and so I_{std} is higher). Higher (dis)charge currents cause larger (dis)charge current from UC, leading to more frequent UC overflow/shortages.

In this paper, the battery stress is an external condition and electrochemical reaction that affects battery degradation and lifetime [35], [36]. This work utilizes the current degradation model to evaluate the impact of discharge/charge current management on degradation and lifetime, because power buffer can control discharge/charge current and degradation rate. We assume the (dis)charge stress to be exponential with the (dis)charge current, because large (dis)charge current exponentially accelerates the battery degradation by exciting high over-potential between solid-phase and solution-phase voltages [36].

Algorithm 3. Target Current Decision (I_L)

1: **while** 1 **do**
2:   **if** $I_{b,ub} < I_{rq,n}$ **then** /* Peak power reduction */
3:     $I_L \leftarrow I_{rq} - I_{b,ub}$
4:   **else**
5:     $I_L \leftarrow 0$
6:   **end if**
7: **end while**

Algorithm 3 and Figs. 5 and 6 show the overall algorithm. Lines 2–6 of Algorithm 3 determine the discharge current from the power-dense battery (I_L) to meet the high (dis)charge current requirement (I_{rq}). When I_{rq} is greater than its upper bound $(I_{b,ub})$, the power-dense battery provides energy to reduce the discharge current of the energy-dense battery (Lines 2–3). For the effective power buffering, current upper bounds are critical because they dictate the amount of discharge current of the storage.

Now we describe how to determine the current bounds $(I_{b,ub})$. Algorithm 4 explains the main current bound searching algorithm and its preparation steps. The first preparation step is to calculate the available energy in the buffer (E_c). If we use an ultra-capacitor (UC) as the power-dense energy buffer, we can use the standard equation related to the capacitor energy $(E_c = \frac{1}{2} C_c V_{c,rated}^2)$, where $V_{c,rated}$ denotes the UC-rated voltage and C_c denotes the UC capacitance.

Algorithm 4. Determination of the Current Bounds for Peak Current Reduction

$k \leftarrow 1$
while 1 **do**
  $E_c \leftarrow \frac{1}{2} C_c V_{c,rated}^2$
  $T_c \leftarrow$ DisChargeCycleTimeUpdate()
  $I_{avg} \leftarrow \frac{k-1}{k} I_{avg} + \frac{1}{k} I_{rq}$
  $I_{std} \leftarrow \{\frac{k-1}{k} {I_{std}}^2 + \frac{1}{k}(I_{rq} - I_{avg})^2\}^{\frac{1}{2}}$
  $I_{ub} \leftarrow$ Find current bound$(E_c, I_{avg}, I_{std}, T_c)$
  $k \leftarrow \min(k+1, kValueMax)$
  Sleep()
end while

Next, it statistically figures out discharge/charge pattern including average charging/recharging cycle period (T_c), average current (I_{avg}) and its standard deviation (I_{std}). Then, the current bound is determined under the assumption that the required current would follow the normal distribution with average (I_{avg}) and standard deviation (I_{std}), over the next control period. When the average required current (I_{avg}) is high, the current bound I_{ub} should be increased to effectively buffer the recharging energy while reducing the charge stress. This buffered energy will be supplied to the load to mitigate the energy-dense battery discharge current as shown in Fig. 5. The standard deviation of the required current (I_{std}) also affects the determination of I_{ub}. A large standard deviation means the transfer of a large amount of energy with a high peak current. In such a case, I_{ub} should be increased to buffer energy effectively without incurring shortage of energy in the power-dense energy buffer as shown in Fig. 5.

Algorithm 5 and Fig. 6 describe how to determine I_{ub} based on the energy capacity of the buffer and the history of required power. To maximize the utilization of power-dense energy buffer, it should spend most energy in the buffer during the discharging period while minimizing peak discharge current (I_{ub}). This means I_{ub} should not only be set to as a small value as possible, but also make the supplied energy from the buffer (E_{bf}) equal to the stored energy in the buffer (E_c) under the load requirement pattern. In the algorithm, it searches proper I_{ub} meeting these conditions. For each case of I_{ub}, it calculates the supplied energy by integrating buffered power ($P_{bf} = V_b I_{bf} = V_b(i - I_{ub})$) during one charging/recharging cycle (T_c) as:

$$\begin{aligned}
E_{bf} &= \int_0^{T_c} P_{bf}(t)dt = T_c \int_{I_{ub}}^{\infty} P_{bf}(i)\mathrm{PDF}(i)di \\
&= T_c \int_{I_{ub}}^{\infty} V_b I_{bf}(i)\mathrm{PDF}(i)di \\
&= T_c \int_{I_{ub}}^{\infty} V_b(i - I_{ub})\mathrm{PDF}(i)di \\
&= T_c V_b \left[\int_{I_{ub}}^{\infty} i\mathrm{PDF}(i)di - I_{ub}\int_{I_{ub}}^{\infty}\mathrm{PDF}(i)di\right] \\
&= T_c V_b \left[\int_{I_{ub}}^{\infty} i\mathrm{PDF}(i)di - I_{ub}(1 - \mathrm{CDF}(I_{ub}))\right] \\
&= T_c V_b \left[I_{std}^2 \mathrm{PDF}(I_{ub}) + (I_{avg} - I_{ub})(1 - \mathrm{CDF}(I_{ub}))\right],
\end{aligned} \tag{16}$$

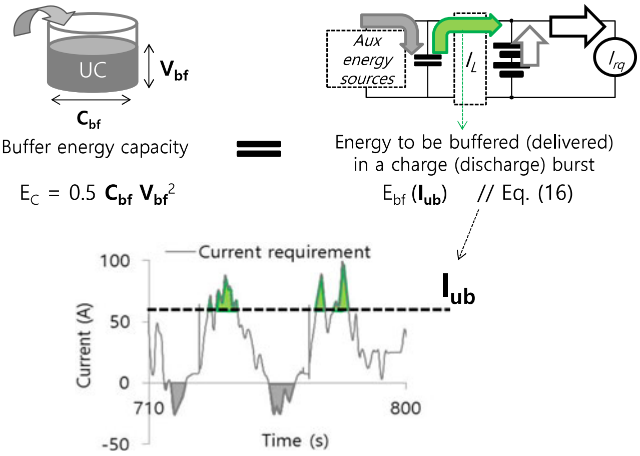


Fig. 6. Determination of current bounds for peak power reduction.

Algorithm 5. Find Current Bound$(E_c, I_{avg}, I_{std}, T_c)$

for $I_{ub} = I_{avg}; I_{ub} < I_{avg} + 3I_{std}; I_{ub} = I_{ub} + \frac{I_{std}}{20}$ **do**
  if $E_{bf}(I_{ub}, I_{avg}, T_c) \geq E_c$ **then**
    Break()
  end if
end for
return I_{ub}

where $\mathrm{PDF}(i)$ (*likewise* $\mathrm{CDF}(i)$) is a probability (*likewise* cumulative) density function of a current i, V_b is the battery output voltage, and P_{bf} (*likewise* I_{bf}) is the buffered power (*likewise* current). In the first line, we transform the buffered power over time, $P_{bf}(t)$, to the buffered power over current, $P_{bf}(i)$.[4] Then, we can statistically calculate the expected buffered energy based on a probability density function of a current, $\mathrm{PDF}(i)$, and buffered current (I_{bf}) as shown in the second line. We assumed battery voltage and system output voltage is the same and constant, so the buffered power can be calculated by multiplying the voltage (V_b) and current flowing through the energy buffer ($I_{bf} = i - I_{ub}$) as described in the third line. After rearranging the terms in the equation, we can derive the equation calculating the amount of the buffered energy which is function of I_{ub}. Note that for a normal distribution, we can calculate $\int_X^Y i\mathrm{PDF}(i)di$ in Eq. (16); see Eq. (17) in the supplement, available online.

6 Evaluation of Offline Power-Supply Guarantee Analysis

In this section, we demonstrate the performance of the offline power-supply guarantee analysis presented in Sections 3 and 4. First, we evaluate the analysis only with a uniform supply in Section 3, and then that with multiple power-supply sources in Section 4.

4. According to the mean ergodic theorem, $\lim_{T\to\infty}\frac{1}{T}\int_0^T X(t)dt \to \int_{-\infty}^{\infty} X(u)PDF(u)du$; i.e., time average of random process can be approximated by its average over the probability space.

TABLE 1
Parameters of the Power Supply: Renewable Energy Sources Γ_1–Γ_3, the Energy Buffer $\Gamma_{\mathtt{UC}}$, and the Battery Pack $\Gamma_{\mathtt{bat}}$

Tasks	$T^{\mathrm{s}}(s)$	$L^{\mathrm{s}}(s)$	$P^{\mathrm{s}}(W)$
Γ_1	6.0	2.0	3.0
Γ_2	6.0	1.5	3.0
Γ_3	1.5	0.2	0.2
$\Gamma_{\mathtt{UC}}$		3.0	3.0
$\Gamma_{\mathtt{bat}}$			40.0

TABLE 2
Parameters of the Power-Demand Operations: λ_1–λ_5

Tasks	$T^{\mathrm{d}}(s)$	$L^{\mathrm{d}}(s)$	$P^{\mathrm{d}}(W)$
λ_1	6.0	2.0	12.0
λ_2	6.0	2.0	9.6
λ_3	4.0	2.0	6.0
λ_4	4.0	1.0	6.0
λ_5	6.0	4.0	7.2

6.1 Results Without Additional Power-Supply Sources

To evaluate the offline power-supply guarantee analysis with a uniform supply, we need to determine parameters of the battery pack and a set of power-demand operations. For the power capability of the battery pack, we follow the parameters of the case study to be described in Section 7, setting $P^{\mathrm{s}}_{\mathtt{bat}}$ to 40W (see Table 1). For each power-demand operation λ_k, we determine its parameters T^{d}_k, L^{d}_k and P^{d}_k as follows. T^{d}_k is uniformly chosen in $[1.0, 10.0]s$; L^{d}_k is uniformly distributed in $[0.1, 0.5 \cdot T^{\mathrm{d}}_k]$s; and P^{d}_k is uniformly chosen in $[5.0, 15.0]$W. We also set ϵ_t and ϵ_p to 0.1s and 0.1W. Note that the scales of the above parameters are determined by considering the parameters of power-demand operations in the case study to be described in Section 7 (see Table 2).

We consider 10 choices of n (1, 2, $3, \ldots, 10$) for the number of operations in each set of power-demand operations. We randomly generate 10,000 operation sets for each n (a total of 100,000 operation sets), and check the ratio of operation sets power-guaranteed by:

- Plain−None: the plain analysis (i.e., Lemma 1), and
- Improved−None: the improved analysis (i.e., Lemmas 2 with Lemma 3).

We would like to mention that the guarantee ratio shown in the Y-axis in Fig. 7 means the ratio of operation sets, which are power-guaranteed by the corresponding analysis and framework. For example, in Fig. 7a, the dotted and solid lines for the X-axis of 4 (i.e., the number of operations) correspond to 78.30 and 88.16 percent, respectively. This means that out of 10,000 operation sets, 7,830 and 8,816 operation sets are

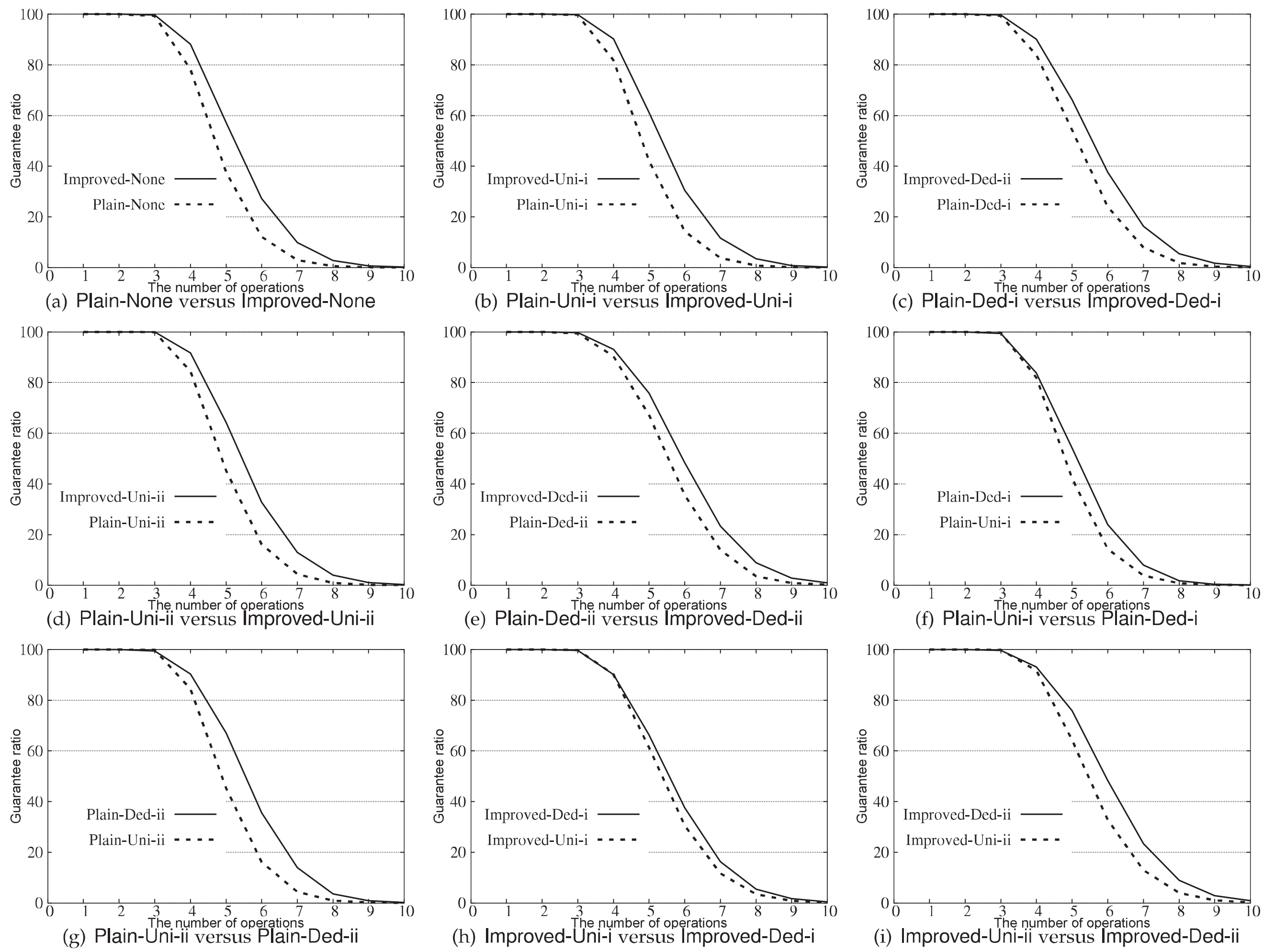


Fig. 7. The ratio of operation sets power-guaranteed under various situations.

power-guaranteed by Plain−None and Improved−None, respectively, when the X-axis is 4.

We have two observations on the results shown in Fig. 7a. First, both Plain−None and Improved−None guarantee the power supply of all operation sets when $n = 1$ or $n = 2$; the ratio of operation sets guaranteed decreases as n increases, and finally, the ratio becomes almost zero when $n = 10$. Second, Improved−None always results in no smaller ratio than Plain−None. The difference between the two ratios is maximized when $n = 5$ (57.22 percent versus 37.73 percent), which demonstrates the effectiveness of the improved analysis compared to the plain analysis.

6.2 Results With Multiple Power-Supply Sources

We now evaluate the offline power-supply guarantee analysis with multiple power-supply sources. Similar to the previous subsection, we need to determine parameters of not only power-supply sources, but also power-demand operations. For power-supply sources, we also follow the parameters of the case study to be described in Section 7. That is, we set $P^{\mathrm{s}}_{\mathtt{bat}}$, $P^{\mathrm{s}}_{\mathtt{UC}}$ and $L^{\mathrm{s}}_{\mathtt{UC}}$ to 40W, 3W and 3s, respectively, and λ_1, λ_2 and λ_3 have the parameters (6,2,3), (6,1.5,3) and (1.5,0.2,0.2), respectively, where (A, B, C) implies $T^{\mathrm{s}} = A$(s), $L^{\mathrm{s}} = B$(s) and $P^{\mathrm{s}} = C$(W), as shown in Table 1 for the case study to be described. We consider two situations for power-supply equipment in addition to $\Gamma_{\mathtt{bat}}$: (i) the system equipped with Γ_1, and (ii) the system equipped with Γ_1, Γ_2 and Γ_3. We will use the same operation sets as the previous section (i.e., 10,000 sets for each n), and check the ratio of operation sets guaranteed by the followings:[5]

- Plain−Uni−i: the uniform supply (Lemma 4) with the plain analysis (Lemma 1) when the additional power-supply is Γ_1,
- Improved−Uni−i: the uniform supply with the improved analysis (i.e., Lemmas 2 with Lemma 3) when the additional power-supply is Γ_1,
- Plain−Ded−i: the dedicated supply (Lemma 6) with the plain analysis when the additional power-supply is Γ_1,
- Improved−Ded−i: the dedicated supply with the improved analysis when the additional power-supply is Γ_1,
- Plain−Uni−ii: the uniform supply with the plain analysis when the additional power-supplies are Γ_1, Γ_2 and Γ_3,
- Improved−Uni−ii: the uniform supply with the improved analysis when the additional power-supplies are Γ_1, Γ_2 and Γ_3,
- Plain−Ded−ii: the dedicated supply with the plain analysis when the additional power-supplies are Γ_1, Γ_2 and Γ_3, and
- Improved−Ded−ii: the dedicated supply with the improved analysis when the additional power-supplies are Γ_1, Γ_2 and Γ_3.

We can summarize the results shown in Figs. 7b, 7c, 7d, 7e, 7f, 7g, 7h, and 7i as follows.

- R1. The ratio by both Improved−X−Y and Plain−X−Y decreases as n increases, from 100 percent with $n = 1$ to near-zero with $n = 10$ as shown in all the figures.
- R2. Improved−X−Y always results in no smaller ratio than Plain−X−Y, but the amount of improvement of Improved−X−Y over Plain−X−Y varies with the approach (uniform versus dedicated supply) and additional power-supply sources (Γ_1 versus Γ_1, Γ_2 and Γ_3); see Figs. 7b, 7c, 7d, and 7e.
- R3. The dedicated supply (i.e., X−Ded−Y) exhibits better performance than the uniform supply (i.e., X−Uni−Y) as shown in Figs. 7f, 7g, 7h, and 7i.
- R4. The gap between ratios by X−Ded−Y and X−Uni−Y when X=Improved is smaller than the gap when X=Plain; see Figs. 7f, 7g, 7h, and 7i.

R1 and R2 are straightforward. For R2, the maximum differences between the ratio by Improved−Y−Z and that by Plain−Y−Z for each pair of Y and Z are 19.21 percent ($= 61.3\% - 42.09\%$) with $n = 5$ when Y=Uni and Z=i, 18.92 percent ($= 64.23\% - 45.31\%$) with $n = 5$ when Y=Ded and Z=i, 13.73 percent ($= 37.64\% - 23.91\%$) with $n = 6$ when Y=Uni and Z=ii, and 12.74 percent ($= 48.33\% - 35.59\%$) with $n = 6$ when Y=Ded and Z=ii.

On the other hand, R3 and R4 are non-trivial. Although the ratio by X−Ded−Z seems higher than the ratio by X−Uni−Z in every situation, X−Ded−Z cannot dominate X−Uni−Z,[6] which is different from the dominant relation of Improved−Y−Z over Plain−Y−Z. That is, there exist many operation sets which are power-guaranteed by X−Uni−Y but not by X−Ded−Z; for example, there exist 1.12, 1.56, 0.53 and 0.79 percent such operation sets with $n = 5$ when the pair of (X and Z) is (Plain, i), (Improved, i), (Plain, ii) and (Improved, ii), respectively. Therefore, X−Uni−Z complements X−Ded−Z in terms of offline power guarantee. Also, we can observe R4. The gap between ratio by X−Ded−i and X−Uni−i with $n = 5$ is 12.15 percent ($= 54.24\% - 42.09\%$) when X=Plain, but the gap is reduced to 4.99 percent ($= 66.29\% - 61.3\%$) when X=Improved. Likewise, the same holds between the ratio by X−Ded−ii and by X−Uni−ii: 21.8 percent when X=Plain and 11.66 percent when X=Improved. This is because the improved analysis does not have much room for improvement since it already improved the power guarantee over the plain analysis.

7 Case Study for the Proposed Framework

We now conduct a case study and demonstrate how much real-world systems can benefit from the proposed framework. It consists of the offline power-supply guarantee analysis (in Sections 3 and 4) and online power management (in Section 5). We focus on whether or not they meet the goals stated in Section 2.2. To this end, we first build a simulation model on *Matlab Simulink* and run simulations with the proposed algorithms in the model, in order to demonstrate our framework guarantees power-supply requirement and reduces the energy dissipation. Hardware and parameters in the case study are derived from the prototype consisting of HESS developed in our earlier conference version [30]. We then present experimental results for the simulation

5. For the dedicated supply, we apply the lower-priority-first strategy for selecting λ^{ded}.

6. A is said to dominate B when all operation sets power-guaranteed by B are also power-guaranteed by A.

model, demonstrating the HESS's power-supply guarantee and reduction of energy dissipation.

7.1 Case Study With the Simulation Model

We target an actual system as a case study, which is equipped with wheels, wheel motors, stepper motors, coolers and a HESS including a pack of lithium-ion batteries, a pack of UCs, an RBS and solar panels; the actual system comes from the prototype developed in our preliminary conference version [30]. We can determine the parameters of power-demand operations and power-supply sources from their power demand/supply profiles and specifications. Specifications of energy sources/storage are presented in Table 1, and those of the power-demand operations are listed in Table 2. Based on the parameters, we determine the minimum battery capacity achieving a power-supply guarantee for the system via the power guarantee analysis in Sections 3 and 4. We have then executed various sequences of operations while recording battery states to evaluate the proposed system. The applications and tasks considered in the case study are detailed as follows.

Our driving application is programmed to operate wheel motors to achieve the driving profile. Our motor control (λ_1) depends on the PID controller to achieve the required speed, its control interval ($T_1^{\mathtt{d}}$), maximum acceleration time (time to achieve the target speed, $L_1^{\mathtt{d}}$), and the maximum power ($P_1^{\mathtt{s}}$) are 6s, 2s, and 12W, respectively. To reflect various user applications, we also ran applications that sporadically actuate motors (λ_4 and λ_5). Some applications actuate stepper motors to control position (λ_2), while others (λ_3) use thermal fins to regulate temperature for system thermal stability. The power-demand operations are shown in Table 2.

We deployed two types of solar panels. One solar panel can generate 10W, and the other 25W at sunny day. Its performance dropped largely at cloudy. So, we assumed the minimum power rate would be 3W during 1.5 seconds for the small panel and 3W during 2 seconds for the large one every 6 seconds at our application environment (Γ_1 and Γ_2). We also assumed 0.2W of power is regenerated for 0.2 seconds every 1.5 seconds during driving application (Γ_3). We used an ultra capacitor pack for the energy buffer which has 400F capacitance and 2.7V of rated-voltage. We designed a switched mode DC/DC converter to move energy between the UCs and battery cells. For this experiments, we optimized the converter configuration to maximize the converter efficiency at 3W power transfer for 3 seconds ($\Gamma_{\mathtt{UC}}$). We targeted lithium-ion batteries each of which has 1400mAh of energy capacity and 4.2W of the minimum power capability, under the assumption that each cell has 3.0V minimum voltage and 1C-rate of power rate. We have to determine the number of battery cells (which is equivalent to the battery capacity $\Gamma_{\mathtt{bat}}$) to guarantee power capability for the power-demand operations.

When it comes to the simulation model for the case study, we utilize *Matlab Simulink* standard with the *Simscape* model library to simulate the architecture as shown in Fig. 18 in the supplement, available online. The boost DC-DC converter model is used to connect UC and battery. We extracted two RC battery parameters from battery cells which is used in the case study. We used RC parameters of the UC that is described in its data sheet [37].

7.2 Evaluation for the Case Study

In this subsection, we evaluate our offline power-supply guarantee analysis and online power management for the case study.

7.2.1 *Offline Power-Supply Guarantee Analysis*

As mentioned in the description of the applications and tasks in the case study, we schedule the five operations in Table 2 supplied by the power sources in Table 1. We compare the three frameworks associated with the plain offline power guarantee analysis in Section 3.2 (i.e., Lemma 1, denoted by Plain) and its improved version in Section 3.3 (i.e., Lemma 2 with Lemma 3, denoted by Improved) as follows: (a) the framework without additional supply sources (using $\Gamma_{\mathtt{bat}}$ only), (b) the framework with additional supply sources (using all the supply sources in Table 1), which operate as a uniform supply, and (c) the framework with additional supply sources (using all the supply sources in Table 1), which operate as a dedicated supply. We use the same notations explained in Section 6 as follows:

- Plain−None: (a) associated with the plain analysis,
- Improved−None: (a) associated with the improved analysis,
- Plain−Uni−ii: (b) associated with the plain analysis,
- Improved−Uni−ii: (b) associated with the improved analysis,
- Plain−Ded−ii: (c) associated with the plain analysis, and
- Improved−Ded−ii: (c) associated with the improved analysis.

Note that we use the same notations as Section 6.

If we apply Plain−None, Lemma 1 cannot guarantee the timely execution of λ_5 (as well as λ_4) with power requirement. That is, during the interval of length $(T_5^{\mathtt{d}} - L_5^{\mathtt{d}} + \epsilon_t) = 6.0 - 4.0 + 0.1 = 2.1$, $\sum_{\lambda_i \in \lambda \setminus \{\lambda_5\}} I_{5 \leftarrow i}(T_5^{\mathtt{d}} - L_5^{\mathtt{d}} + \epsilon_t)$ is the sum of $2.1 \cdot 12.0 + 2.1 \cdot 9.6 + 2.1 \cdot 6.0 + 2.1 \cdot 6.0 = 2.1 \cdot 33.6 = 70.56$, which is no less than $(P_{\mathtt{bat}}^{\mathtt{s}} - P_5^{\mathtt{d}} + \epsilon_p) \cdot (T_5^{\mathtt{d}} - L_5^{\mathtt{d}} + \epsilon_t) = (40.0 - 7.2 + 0.1) * (3.0 - 1.0 + 0.1) = 32.9 \cdot 2.1 = 69.09$. However, if we apply Improved − None, Lemma 2 with Lemma 3 guarantees that every power demand operation in Table 2 can finish its execution within its deadline, without experiencing power deficiency. That is, after three iterations for finding slack values in Lemma 3, the slack values of λ_1, λ_2, λ_3, λ_4 and λ_5 are 1.6, 1.6, 0.1, 0.7, 0.0, respectively; then, during the interval of length $(T_5^{\mathtt{d}} - L_5^{\mathtt{d}} + \epsilon_t) = 6.0 - 4.0 + 0.1 = 2.1$, $\sum_{\lambda_i \in \lambda \setminus \{\lambda_5\}} I_{5 \leftarrow i}(2.1)$ is the sum of $2.0 \cdot 12.0 + 2.0 \cdot 9.6 + 2.0 \cdot 6.0 + 1.4 \cdot 6.0 = 63.6$, which is less than 69.09.

We now explain how Improved−None calculates proper power requirement, compared to MAX and AVG, which denote the sum of the maximum power demand (i.e., $\sum_{\lambda_i \in \lambda} P_i^{\mathtt{d}} = 40.8W$), and the average of the maximum power demand (i.e., $\sum_{\lambda_i \in \lambda} \frac{P_i^{\mathtt{d}} \cdot L_i^{\mathtt{d}}}{T_i^{\mathtt{d}}} = 16.5W$). First, we can find the lowest battery capacity $\Gamma_{\mathtt{bat}}$ that makes Improved−None give power supply guarantee, which is $\Gamma_{\mathtt{bat}} = 38.0W$.[7] Then, Improved−None (38.0W) reduces the required battery capability. Also, it is straightforward that, if the battery

7. We can find such battery capacity by applying binary search.

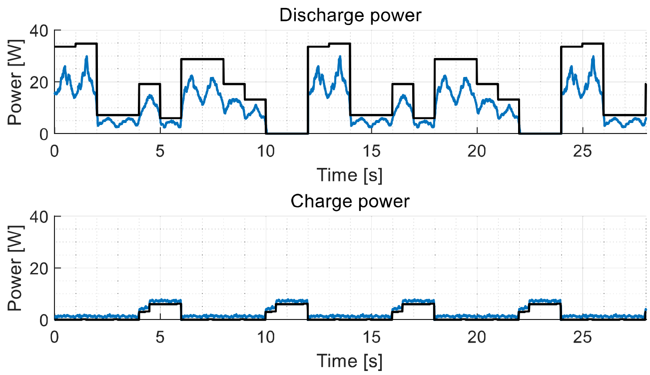


Fig. 8. Power demand and supply simulation with noise (*blue line*) and without noise (*black line*).

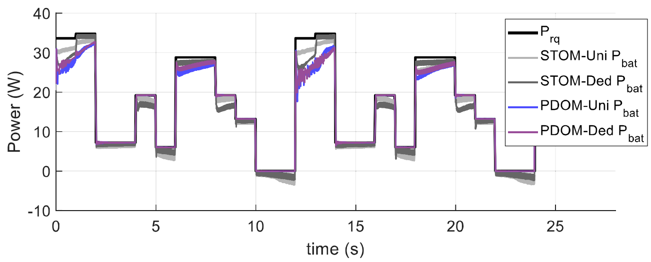


Fig. 9. Power scheduling simulation results with worst-case power demand and supply.

capacity is as much as AVG, we cannot supply the sufficient power in the worst case.

When it comes to Improved–Uni–ii and Improved–Ded – ii, they further reduce the required battery capability by consuming 36.3W and 34.8W, respectively.[8] Between the two, Improved–Ded–ii is the most effective framework for a power guarantee, in that it can reduce the required battery capability by 14.7 percent, compared to MAX, a naive approach without utilizing renewable energy sources.

7.2.2 Online Power Management

We target the case study explained in Section 7.1, and apply the simulation model to power demand and supply scheduling with our online management schemes. The worst-case power-demand operations are scheduled according to Table 2. The generated power is supplied into the energy buffer according to power-supply sources in Table 1. Fig. 8 shows the square-shaped worst-case power demand and supply results. We also generated more realistic total power demand and supply based on actual power supply/demand observations; we smoothed them and added uniform random noises. We then evaluated the schemes with ideal power supply and demand profiles and their smoothed profiles. That is, we consider two situations: (Sit1) the power demand and supply always exhibit their worst-case; and (Sit2) the power demand and supply include noises, which is more realistic. Under these situations, we have evaluated the following four online power management schemes:

8. Note that Plain–Ded–ii and Improved–Ded–ii choose λ_4 to which the additional power-supply sources are dedicated.

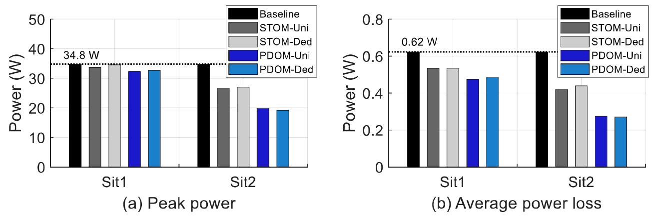


Fig. 10. Peak power (a) and average power loss (b) under Sit1 and Sit2.

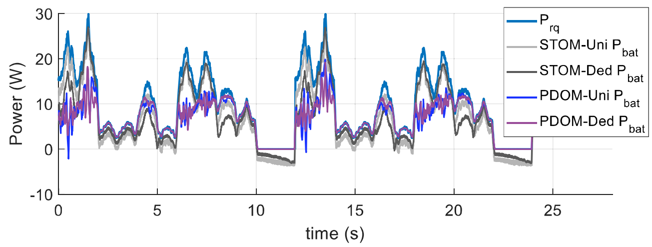


Fig. 11. Power scheduling simulation results with power supply and demand including noises.

- STOM–Uni: static-demand-based online management with the uniform supply,
- STOM–Ded: static-demand-based online management with the dedicated supply,
- PDOM–Uni: predicted-demand-based online management with the uniform supply, and
- PDOM–Ded: predicted-demand-based online management with dedicated supply.

We first evaluate the situation Sit1. Fig. 9 shows the result battery current under worst-case power supply and demand with the proposed online management methods, where $P_{\mathtt{rq}}$ and $P_{\mathtt{bat}}$ denote the total power demand and the amount of power supplied by the battery pack, respectively. STOM–Uni reduces battery power requirement evenly, while STOM–Ded decreases battery power requirement only when the dedicated tasks runs. PDOM–Uni and PDOM–Ded keep updating the current bound and limit battery current based on the current bound. STOM–Uni, STOM–Ded, PDOM–Uni, and PDOM–Ded reduce peak current to 3.44, 0.72, 7.30, and 6.01 percent, respectively, compared to $P_{\mathtt{rq}}$. Detailed values under Sit1 can be seen in Fig. 10a.

We also note the average power dissipation is calculated by $P_{\mathtt{loss}} = \frac{1}{t_{\max}} \int_0^{t_{\max}} P^{\mathtt{s}}_{\mathtt{bat-loss}}(t)\, dt = \frac{1}{t_{\max}} \int_0^{t_{\max}} \Sigma I^2_{\mathtt{bat,i}}(t) R_{\mathtt{bat,i}}(t)\, dt$. STOM–Uni, STOM–Ded, PDOM–Uni, and PDOM–Ded decrease the average power dissipation by 14.07, 14.23, 23.88 and 22.01 percent, respectively, compared to $P_{\mathtt{rq}}$. Detailed values under Sit1 can be seen in Fig. 10b.

Next, we evaluate the situation Sit2. Fig. 11 represents battery power of each approach with the more realistic power requirements. Fig. 12 shows the detailed peak power reductions. STOM–Uni, STOM–Ded, PDOM–Uni, and PDOM–Ded could decrease peak current respectively to 23.36, 22.44, 42.98 and 44.93 percent during the given operations, compared to $P_{\mathtt{rq}}$. The average power dissipation of STOM–Uni, STOM–Ded, PDOM–Uni, and PDOM–Ded are reduced to 32.60, 29.48, 55.79, and 56.49 percent, respectively, compared to $P_{\mathtt{rq}}$. Fig. 10a and (b) under Sit2 show the performance of the approaches under Sit2.

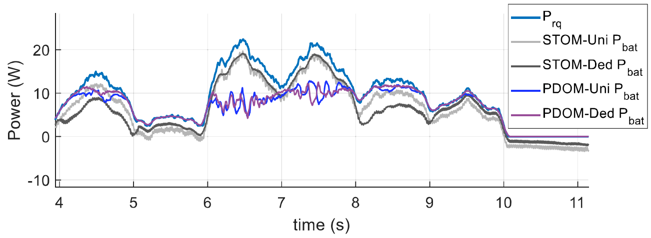


Fig. 12. Power scheduling simulation results with power supply and demand including noises (4s - 10s).

We can interpret the evaluation results as follows. First, if we compare the reduction of the peak current by the four online power management schemes in Sit1, with that in Sit2, that in Sit2 is more significant than that in Sit1. This is because Sit2 exhibits less power consumption of power-demand operations than their worst-case and more power generation by power-supply sources than their minimum; this yields more room to reduce the peak current, and our proposed online management schemes effectively exploit the room. Second, if we compare STOM$-^*$ and PDOM$-^*$, the latter yields better performance, since PDOM$-^*$ decides the current upper bound in real-time based on the average and standard deviation of recent current requirement instead of using worst-case task power requirement. This enables PDOM$-^*$ to reduce the peak current more aggressively while providing the buffer energy to the load without the buffer energy shortage. This improved peak power reduction leads to the lower average energy dissipation compared to STOM$-^*$. Third, while $^*-$Uni reduces the current uniformly, so peak current is always reduced by a certain amount, the peak current reduction of $^*-$Ded depends on which task is dedicated. Since the dedicated task (λ_4 in the case study) did not contribute the peak current in Sit1 and Sit2, $^*-$Uni would be more beneficial for peak current reduction. However, if dedicated task contributes the system peak current, $^*-$Ded could reduce peak power largely. Results in Figs. 9 and 11 show this trend clearly.

8 Conclusion

In this paper, we have proposed a power scheduling framework—as a guideline for the design of a HESS—that ensures power sufficiency for the worst-case power demand and supply using real-time scheduling schemes. To improve the runtime performance of the HESS further, we have also designed an online power management framework that utilizes the surplus energy from a real-time power supply, reducing the battery's peak power and hence extending its lifetime. We have evaluated the effectiveness of the offline power-supply guarantee using simulations, and validated the design with a simulation model based on the HESS-powered actual system running realistic applications, demonstrating power sufficiency with a lower-cost HESS and higher energy-efficiency.

In future, we would like to develop a design framework for general energy storage systems based on a power guarantee analysis. It will search for the optimal configurations of energy storage systems at their design or replacement time while considering power demand history and power supply's state-of-health. We also plan to build a power/energy management system that not only schedules power demand and supply, but also monitors and pro/diagnoses energy storages/sources.

Acknowledgments

An earlier (shorter) version of this article was presented at the IEEE RTSS 2016 [30]. The work reported in this article was supported in part by the NSF under Grants CNS-1446117 and CNS-1739577, in part by the ONR under Grant No. N00014-18-1-2141, and by LG Chem Ltd. This research was also supported in part by the National Research Foundation of Korea (NRF) funded in part by the Ministry of Science and ICT under Grant 2019R1A2B5B02001794 and 2017H1D8A2031628. This research was also supported by the Ministry of Science and ICT, Korea, under the Grand Information Technology Research Center support program (IITP-2020-2015-0-00742) supervised by the IITP.

References

[1] P. D. Blair, "Modeling energy and power requirements of electric vehicles," *Energy Convers.*, vol. 18, no. 3, pp. 127–134, 1978.

[2] G. Cai, L. Feng, B. M. Chen, and T. H. Lee, "Systematic design methodology and construction of UAV helicopters," *Mechatronics*, vol. 18, no. 10, pp. 545–558, 2008.

[3] E. Pastor, J. Lopez, and P. Royo, "UAV payload and mission control hardware/software architecture," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 22, no. 6, pp. 3–8, Jun. 2007.

[4] C. Chan and K. Chau, "An overview of power electronics in electric vehicles," *IEEE Trans. Ind. Electron.*, vol. 44, no. 1, pp. 3–13, Feb. 1997.

[5] T.-K. Lee, B. Adornato, and Z. Filipi, "Synthesis of real-world driving cycles and their use for estimating PHEV energy consumption and charging opportunities: Case study for midwest/U.S." *IEEE Trans. Veh. Technol.*, vol. 60, no. 9, pp. 4153–4163, Nov. 2011.

[6] E. Kim, J. Lee, and K. G. Shin, "Real-time prediction of battery power requirements for electric vehicles," in *Proc. ACM/IEEE 4th Int. Conf. Cyber-Physical Syst.*, Apr. 2013, pp. 11–20.

[7] Z. Li *et al.*, "On the feasibility of linear discrete-time systems of the green scheduling problem," in *Proc. IEEE 32nd Real-Time Syst. Symp.*, Nov. 2011, pp. 295–304.

[8] "Peak shaving through real-time scheduling of household appliances," *Energy Buildings*, vol. 75, pp. 133–148, 2014.

[9] T. Facchinetti and M. L. D. Vedova, "Real-time modeling for direct load control in cyber-physical power systems," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 689–698, Nov. 2011.

[10] G. Benetti, M. Delfanti, T. Facchinetti, D. Falabretti, and M. Merlo, "Real-time modeling and control of electric vehicles charging processes," *IEEE Trans. Smart Grid*, vol. 6, no. 3, pp. 1375–1385, May 2015.

[11] S. Pay and Y. Baghzouz, "Effectiveness of battery-supercapacitor combination in electric vehicles," in *Proc. IEEE Bologna Power Tech Conf.*, 2003, Art. no. 6.

[12] R. Kaiser, "Optimized battery-management system to improve storage lifetime in renewable energy systems," *J. Power Sources*, vol. 168, no. 1, pp. 58–65, 2007.

[13] S. Park, Y. Kim, and N. Chang, "Hybrid energy storage systems and battery management for electric vehicles," in *Proc. 50th ACM / EDAC / IEEE Des. Automat. Conf.*, 2013, pp. 1–6.

[14] J. Cao and A. Emadi, "A new battery/ultracapacitor hybrid energy storage system for electric, hybrid, and plug-in hybrid electric vehicles," *IEEE Trans. Power Electron.*, vol. 27, no. 1, pp. 122–132, Jan. 2012.

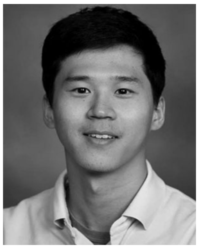
[15] E. Kim, K. Shin, and J. Lee, "Real-time discharge/charge rate management for hybrid energy storage in electric vehicles," in *Proc. IEEE Real-Time Syst. Symp.*, 2014, pp. 228–237.

[16] M. Ortuzar, J. Moreno, and J. Dixon, "Ultracapacitor-based auxiliary energy system for an electric vehicle: Implementation and evaluation," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 2147–2156, Aug. 2007.

[17] J. K. Shiau, D. M. Ma, P. Y. Yang, G. F. Wang, and J. H. Gong, "Design of a solar power management system for an experimental uav," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 45, no. 4, pp. 1350–1360, Oct. 2009.

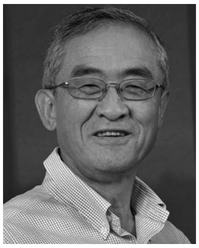
[18] N. Baldock and M. Mokhtarzadeh-Dehghan, "A study of solar-powered, high-altitude unmanned aerial vehicles," *Airc. Eng. Aerosp. Technol.*, vol. 78, no. 3, pp. 187–193, 2006.
[19] M. Z. Jacobson and M. A. Delucchi, "Providing all global energy with wind, water, and solar power, part I: Technologies, energy resources, quantities and areas of infrastructure, and materials," *Energy Policy*, vol. 39, no. 3, pp. 1154–1169, 2011.
[20] J. Paska, P. Biczel, and M. Kłos, "Hybrid power systems – an effective way of utilising primary energy sources," *Renewable Energy*, vol. 34, no. 11, pp. 2414–2421, 2009.
[21] S. Letendre, R. Perez, and C. Herig, "Battery-powered, electric-drive vehicles providing buffer storage for PV capacity value," in *Proc, Solar Conf.*, 2002, pp. 105–110.
[22] R. Baños, F. Manzano-Agugliaro, F. Montoya, C. Gil, A. Alcayde, and J. Gómez, "Optimization methods applied to renewable and sustainable energy: A review," *Renewable Sustain. Energy Rev.*, vol. 15, no. 4, pp. 1753–1766, 2011.
[23] O. Erdinc and M. Uzunoglu, "Optimum design of hybrid renewable energy systems: Overview of different approaches," *Renewable Sustain. Energy Rev.*, vol. 16, no. 3, pp. 1412–1425, 2012.
[24] E. Kim, J. Lee, and K. G. Shin, "Modeling and real-time scheduling of large-scale batteries for maximizing performance," in *Proc. IEEE Real-Time Syst. Symp.*, 2015, pp. 33–42.
[25] W. Zhou, C. Lou, Z. Li, L. Lu, and H. Yang, "Current status of research on optimum sizing of stand-alone hybrid solar–wind power generation systems," *Appl. Energy*, vol. 87, no. 2, pp. 380–389, 2010.
[26] M. Majima, S. Ujiie, E. Yagasaki, K. Koyama, and S. Inazawa, "Development of long life lithium ion battery for power storage," *J. Power Sources*, vol. 101, no. 1, pp. 53–59, 2001.
[27] S. S. Choi and H. S. Lim, "Factors that affect cycle-life and possible degradation mechanisms of a li-ion cell based on licoo2," *J. Power Sources*, vol. 111, no. 1, pp. 130–136, 2002.
[28] H. Kim and K. G. Shin, "Scheduling of battery charge, discharge, and rest," in *Proc. 30th IEEE Real-Time Syst. Symp.*, 2009, pp. 13–22.
[29] L. Benini, A. Macii, E. Macii, M. Poncino, and R. Scarsi, "Scheduling battery usage in mobile systems," *IEEE Trans. VLSI Syst.*, vol. 11, no. 6, pp. 1136–1143, Dec. 2003.
[30] E. Kim, Y. Lee, L. He, K. G. Shin, and J. Lee, "Offline guarantee and online management of power demand and supply in cyber-physical systems," in *Proc. IEEE Real-Time Syst. Symp.*, 2016, pp. 89–98.
[31] J. Zhang, S. Ci, H. Sharif, and M. Alahmad, "Modeling discharge behavior of multicell battery," *IEEE Trans. Energy Convers.*, vol. 25, no. 4, pp. 1133–1141, Dec. 2010.
[32] M. Bertogna and M. Cirinei, "Response-time analysis for globally scheduled symmetric multiprocessor platforms," in *Proc. IEEE Real-Time Syst. Symp.*, 2007, pp. 149–160.
[33] M. Bertogna, M. Cirinei, and G. Lipari, "Schedulability analysis of global scheduling algorithms on multiprocessor platforms," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 4, pp. 553–566, Apr. 2009.
[34] E. Kim and K. G. Shin, "Optimal design and management of a hybrid energy storage system," in *Proc. IEEE Conf. Decis. Control*, 2019, pp. 208–215.
[35] K. Uddin, S. Perera, W. D. Widanage, L. Somerville, and J. Marco, "Characterising lithium-ion battery degradation through the identification and tracking of electrochemical battery model parameters," *Batteries*, vol. 2, no. 2, 2016. [Online]. Available: http://www.mdpi.com/2313-0105/2/2/13
[36] X. Lin, J. Park, L. Liu, Y. Lee, A. M. Sastry, and W. Lu, "A comprehensive capacity fade model and analysis for li-ion batteries," *J. Electrochemical Society*, vol. 160, no. 10, pp. A1701–A1710, 2013.
[37] EATON, Snap-in cylindrical supercapacitors, Accessed: Feb. 17, 2020. [Online]. Available: https://www.eaton.com/us/en-us/catalog/electronic-components/xv-supercapacitor.html

Eugene Kim received the BS degree in electrical engineering from Seoul National University, Republic of Korea, in 2011, and the MS and PhD degrees in computer science and engineering from the University of Michigan, Ann Arbor, in 2017. He is currently a battery algorithm engineer at Faraday Future. His research interests include cyber-physical systems and battery management system for electric vehicles, spacecraft, and mobile systems.



Youngmoon Lee (Student Member, IEEE) received the BS degree in electrical and computer engineering from Seoul National University, Korea, in 2014, the MS degree in computer science and engineering from the University of Michigan, Ann Arbor, in 2016, and the PhD degree in computer science and engineering from the University of Michigan, Ann Arbor, in 2019. He is currently an assistant professor with the Department of Robotics, Hanyang University, Korea. His research interests include cyber-physical systems, embedded systems, and mobile computing.

Liang He (Senior Member, IEEE) is currently an assistant professor with the University of Colorado Denver. He worked as a research fellow with the University of Michigan, Ann Arbor, during 2015–2017, as a research scientist with the Singapore University of Technology and Design during 2012–2014, and as a research assistant with the University of Victoria during 2009–2011. His research interests include CPSes, cognitive battery management, and networking. He has been a recipient of the best paper/poster awards of MobiSys17, QShine14, WCSP11, and GLOBECOM11, and a Best Paper Candidate of GLOBECOM14.



Kang G. Shin (Life Fellow, IEEE) is the Kevin & Nancy O'Connor professor of Computer Science with the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor. His current research focuses on QoS-sensitive computing and networking as well as on embedded real-time and cyber-physical systems. He has supervised the completion of 85 PhDs, and authored/coauthored about 1,000 technical articles, a textbook and more than 40 patents or invention disclosures, and received numerous best paper awards, including 2019 Caspar Bowden Award for Outstanding Research in Privacy Enhancing Technologies, the Best Paper Awards from the 2011 ACM International Conference on Mobile Computing and Networking (MobiCom'11), the 2011 IEEE International Conference on Autonomic Computing, the 2010 and 2000 USENIX Annual Technical Conferences, as well as the 2003 IEEE Communications Society William R. Bennett Prize Paper Award and the 1987 Outstanding IEEE Transactions of Automatic Control Paper Award. He has also received several institutional awards, including the Research Excellence Award in 1989, Outstanding Achievement Award in 1999, Distinguished Faculty Achievement Award in 2001, and Stephen Attwood Award in 2004 from The University of Michigan; a Distinguished Alumni Award of the College of Engineering, Seoul National University in 2002; 2003 IEEE RTC Technical Achievement Award; and 2006 Ho-Am Prize in Engineering (the highest honor bestowed to Korean-origin engineers). He was a co-founder of a couple of startups and also licensed some of his technologies to industry.

Jinkyu Lee (Member, IEEE) received the BS, MS, and PhD degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea, in 2004, 2006, and 2011, respectively. He is an assistant professor with the Department of Computer Science and Engineering, Sungkyunkwan University (SKKU), Republic of Korea, where he joined in 2014. He has been a visiting scholar/research fellow with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI. in 2011–2014. His research interests include system design and analysis with timing guarantees, QoS support, and resource management in real-time embedded systems, mobile systems, and cyber-physical systems. He won the Best Student Paper Award from the 17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS) in 2011, and the Best Paper Award from the 33rd IEEE Real-Time Systems Symposium (RTSS) in 2012.