Design and Management of Satellite Power Systems

Jinkyu Lee, Eugene Kim, and Kang G. Shin
Department of Electrical Engineering and Computer Science
The University of Michigan, U.S.A.
{jinkyul, kimsun, kgshin}@umich.edu

***Abstract*—Satellites are indispensable for broadcast, weather forecast, navigation, and many other applications, but their design entails a number of stringent requirements, such as limited space and weight, impossible/costly online repairs, severe radiation, and a wide range of temperature they have to withstand. These requirements can only be met by an effective, robust co-design of physical and computing (control) parts of each satellite, making them prototypical cyber-physical systems (CPSes). Of the various CPS issues related to satellites, this paper focuses on offline design and online management of *satellite power systems*. Specifically, we analyze and model unique characteristics of power supply and demand of a satellite, which are dictated by the periodicity of power generation from solar panels and the nonlinear behavior of rechargeable battery cells. Based on the understanding of these characteristics, we first propose how to find the best configuration (e.g., the number, the arrangement, and the type) of solar panels and battery cells at design time, such that all tasks can be executed without power shortage throughout the satellite's mission lifetime. Second, we propose how to manage power online so as to execute the highest QoS versions of tasks (thus yielding the most power-effective performance) without compromising the power-sufficiency guarantee under a given configuration. As a case study, we study cubic-shaped nano-satellites, which have been launched multiple times since 2004. We borrow their architecture, configuration and parameters, and demonstrate the effectiveness of our design and management of satellite power systems.**

I. Introduction

Satellites have become essential for many applications, such as broadcast and communications, weather forecast, navigation, earth/space observation, and scientific experiments. Due to their significant impact on human lives, business and scientific exploration, satellites have been widely studied, mainly focusing on their physical and communication aspects, respectively from the aerospace and networking communities [1–3]. Unlike other systems operating on the earth, all sub-systems of a satellite should be accommodated within a limited budget of space and weight, resilient to severe radiation and a wide range of temperature, and ultra-reliable since it is impossible or very costly to replace (failed) physical parts after its launch. This calls for effective, robust design of physical parts and sophisticated controls thereof, addressing the issues of co-designing physical and computing (control) parts simultaneously. That is, satellites are a prototypical example of Cyber-Physical System (CPS), typified by tight integration and coordination between physical and computing parts [4–6]. As a result, the CPS community has recently paid considerable attention to satellites [7, 8].

Of the various CPS issues of satellites, this paper focuses on the design and management of satellite power systems. Since no physical fuel can be supplied to power satellites from earth, they are usually equipped with solar panels and rechargeable battery cells, which entail unique characteristics of power supply, as explained below. According to the revolution of a satellite around the earth, solar panels generate power and charge battery cells when it is exposed to the sun or during the "sun phase," while they cannot during the "eclipse phase" (without sunlight). The amount of power generated depends on the satellite's position, the efficiency of solar panels (which degrades with time), and temperature. On the other hand, battery cells supply power during the eclipse phase or the sun phase in case of a peak power demand. Their discharge behaviors are known to be non-linear [9–11], e.g., discharge efficiency—higher discharge rate, less deliverable power, and recovery efficiency—resting a battery cell will recover its voltage that was temporarily dropped. Battery cells' performance also varies with their efficiency and operating temperature.

As to power demand, each sub-system (e.g., on-board computer, receiver, transmitter, camera, magnetorquer, and magnetometer) consumes power, and is modeled with periodic power-consumption tasks with different versions,[1] e.g., a task that periodically takes and transmits pictures with different resolutions; a higher resolution consumes more power.

Considering the unique characteristics of satellite power supply and demand, we would like to achieve the following goals.

- G1. **Design**: we want to find the best configuration (e.g., the number, the arrangement, and the type) of solar panels and battery cells, which can guarantee offline (i.e., pre-launch) the execution of all tasks without power deficiency until the end of the satellite's mission lifetime.
- G2. **Management**: we want to manage power online (i.e., post-launch) so as to execute tasks with as high QoS versions as possible, without compromising the power sufficiency guaranteed by G1.

G1 is important in that we can not only reduce the production cost, but also allocate more weight and space budget for payloads. G2 is also significant because we can maximize mission performance for a given power systems design.

Existing studies on satellite systems have addressed G1 by over-designing power supply units [12–19]. As to G2, there

[1] Note that the notion of periodic power-consumption tasks has nothing to do with deadlines.

1052-8725/13 $26.00 © 2013 IEEE
DOI 10.1109/RTSS.2013.18

have also been a line of studies on systems with regenerative energy [20–22]. In particular, the authors of [20] provided a general problem formulation with an abstracted model of energy, deadlines and task rewards, where all tasks compete for the temporal resource of a computing unit, which is not the case for a satellite's sub-systems that share energy resource only. In contrast to these existing studies, we address both G1 and G2 by thoroughly investigating the characteristics of a satellite's power supply and demand.

To achieve G1, we analyze and model the characteristics of a satellite's power supply and demand, and derive lower-bounds of generated power from solar panels as well as the SoC (State of Charge) of the entire battery cells, and an upper-bound of power demand, e.g., the generated power during the sun phase is at least as much as $2.3W$ throughout the satellite's mission lifetime. We also derive a relationship between power sufficiency and the SoC level: a positive level of SoC implies that no sub-system suffers from power shortage. Applying these bounds and relationship to the execution of tasks with fixed versions, we find a fixed value X of SoC such that $X \leq X'$ holds where X and X' are SoC values at the beginning and end of a satellite's orbit period.[2] We prove that power sufficiency is guaranteed as long as there exists such a fixed value of SoC. Then, we calculate fixed points for different configurations of solar panels and battery cells, and choose the best configuration, thus addressing G1. Since there is a gap between the lower-bound of SoC and actual SoC (which is available online only), we can execute higher versions of tasks by exploiting this gap. To this end, we propose *dynamic-version execution*, which adaptively raises the execution versions, thus addressing G2. The underlying principle of the dynamic-version execution is to find the highest versions of tasks that yield SoC at the end of each orbit period at least as high as the fixed point that guarantees power sufficiency.

To demonstrate the effectiveness of the proposed approaches, we use cubic-shaped nano-satellites, called *CubeSats*, and borrow configurations from two actually-deployed CubeSats. For rechargeable batteries, we use a popular battery simulator, *Dualfoil* [23]. Our simulation results from actual/realistic configurations and Dualfoil show that we can find the Pareto optimal configurations of solar panels and battery cells, achieving G1. Also, the proposed dynamic-version execution effectively exploits online information, allowing execution of tasks with higher versions without compromising power-sufficiency guarantees (achieving G2).

In summary, this paper makes the following contributions:

- Thorough investigation of the unique characteristics of power supply and demand of a satellite, and derivation of their properties;
- Design and management of satellite power systems, and derivation of solutions to achieve the goals G1 and G2; and
- Case study of an actual nano-satellite, and demonstration of the effectiveness of our solutions.

[2]A satellite periodically revolves around the earth; each orbit period starts at the beginning of an eclipse phase, and ends at the end of the corresponding sun phase. See Fig. 5.

The rest of this paper organized as follows. Section II investigates characteristics of a satellite's power supply and demand. Section III proposes our design and management of power systems. Section IV focuses on a nano-satellite, and validates our solutions. Finally, Section V concludes the paper.

II. Characteristics of Satellites' Power Supply and Demand

Satellites are usually equipped with solar panels and rechargeable battery cells as their energy source and storage. Both are complementary to each other during two alternate phases. First, when a satellite faces the sun, photovolatics in solar panels convert sunlight into electrical energy; the generated power is supplied to the satellite's sub-systems, and then the remaining power is stored in battery cells. On the other hand, during the eclipse phase (or the sun phase with a peak power demand), the stored energy in battery cells is used to operate the satellite sub-systems.

To achieve G1 and G2, we need to know the amount of supplied and consumed power, thus requiring a good understanding of the unique characteristics of supplied power from solar panels and battery cells, and consumed power by the sub-systems. We first analyze the characteristics of power supplied by the solar panels, and then examine power demand from the sub-systems. Finally, we investigate the physical characteristics of a rechargeable battery cell, and express its SoC (State of Charge) using the supplied and demanded power.

A. Power Supply from Solar Panels

Since a satellite revolves around the earth, its solar panels supply power *periodically*. That is, the sun and eclipse phases alternate, and each panel can generate power only during the sun phase. The amount of power generated by a solar panel at t depends on the angle between sunlight and the solar panel, and the angle is determined by the satellite's position and attitude (denoted by $s(t)$). Then, the amount of power generated by all solar panels in a satellite at t is a function of $s(t)$ and n_{S}, where n_{S} is the configuration of solar panels at design time (e.g., the number, the arrangement, and the type of solar panels). Fig. 1 shows an example of the amount of generated power over time, which will be detailed in Section IV.

Another characteristic is the efficiency degradation of solar panels. Due to ultraviolet degradation, redication degradation, fatigue (thermal cycling) and micrometeoroid loss, the efficiency of solar panels decreases over time. Also, a rise in temperature decreases the amount of power generated by solar panels. For example, the amount of power generation by a solar panel in [17] is 1.2 W at $-20°C$, which drops to 1.1 W at $20°C$. Let $e_{\mathsf{S}}(t)$ and $\theta(t)$ denote the efficiency of solar panels and temperature at t.

Taken together, the amount of power generation at t (denoted by $\mathsf{Sup}(t)$) can be expressed as a function of n_{S}, $s(t)$, $e_{\mathsf{S}}(t)$ and $\theta(t)$ as:

$$\mathsf{Sup}(t) = F_{\mathsf{sup}}\Big(n_{\mathsf{S}}, s(t), e_{\mathsf{S}}(t), \theta(t)\Big). \tag{1}$$

As mentioned earlier, $\mathsf{Sup}(t)$ is an increasing (likewise, decreasing) function of $e_{\mathsf{S}}(t)$ (likewise, $\theta(t)$). We will present

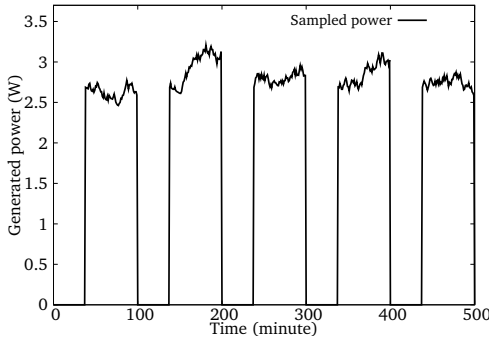


Fig. 1. Generated power from 10 solar panels; the eclipse and sun phases last 38 and 62 minutes, and the details are given in Section IV.

an example of F_{sup} in Section IV-B. Note that all parameters used in this paper are summarized in Table I.

B. Power Demand from Sub-Systems

Each sub-system's instantaneous power consumption is represented as a periodic task τ_i. Task $\tau_i \in \tau$ is specified by $\Big(p_i, c_i(t, v_i)\Big)$, where p_i is the period (Min) and $c_i(t, v_i)$ is the amount of instantaneous power consumption with version (or operating mode) v_i at t (W); so, if τ_i is idle at t, $c_i(t, v_i) = 0$. While some tasks have only one version, others can have multiple versions, e.g., taking pictures with different resolutions; a task with a higher version yields higher workload and higher power consumption than that with a lower version. Note that while p_i is determined at design time, $c_i(t, v_i)$ varies with time; v_i is now expressed as a fixed value, but it will later be generalized to $v_i(t)$ to account for dynamic version changes in Section III. For deterministic behaviors, we assume that p_i is a divisor of ℓ_{ORB} (the orbit period).

The periodic task model can abstract the power consumption of sub-systems in a satellite in that it not only directly expresses sub-systems with periodic operation, e.g., feedback loop control, but also indirectly expresses sub-systems with non-periodic operation. For example, a satellite to be presented in Section IV has four periodic and two non-periodic tasks. For the two non-periodic tasks τ_1 and τ_2, we can set $p_1 = p_2 = \ell_{\mathsf{orb}}$; this is also applicable when the minimum power is consumed during the inactive state of a sub-system.

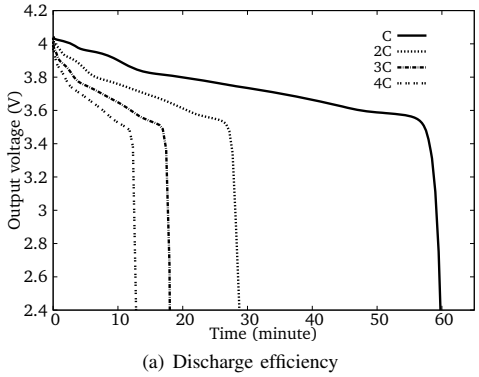
The amount of a satellite's instantaneous power consumption at t (denoted as $\mathsf{Dem}(t)$) is the sum of power consumption of all tasks (sub-systems) as:

$$\mathsf{Dem}(t) = \sum_{i=1}^{|\tau|} c_i(t, v_i), \tag{2}$$

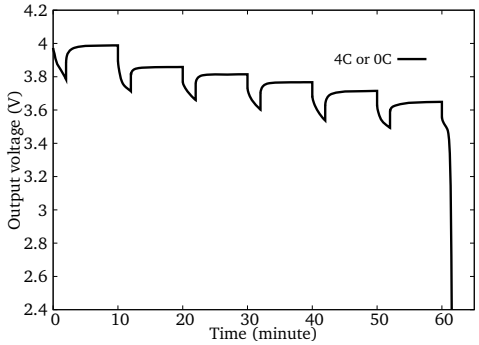
where $|\tau|$ is the number of tasks in τ.

C. Physical Characteristics of Rechargeable Battery Cells

Rechargeable battery cells are known to possess many unique, non-linear characteristics [9–11]. One of them is called *discharge efficiency*—the higher discharge rate, the less efficiency. Fig. 2(a) represents the output voltage drop of a battery cell at various constant discharge rates, simulated by Dualfoil [23]. We can observe that the operation time (during voltage larger than the cutoff voltage 2.4 V) is not inverse-proportional to the discharge rate. That is, while the operation time of a battery cell at the discharge rate of 1 C (C-rate) is 60 minutes, that at 2 C is 28.7 minutes, not 30 minutes. As the discharge rate gets higher, its deliverable power is decreasing; compared to the discharge rate of 1 C, there are 4.3%, 10.0% and 14.6% losses of deliverable power with the discharge rate of 2, 3 and 4 C, respectively. Such losses come from the process of converting chemical energy into useful electric energy for two main reasons. First, the internal resistance of battery cells proportionally increases with the discharge rate, which is called *IR loss*. The second reason is the polarization effect, which occurs because (i) a higher discharge rate yields more concentration difference between the reactants and products at the electrode surface, and/or (ii) a discharge rate is higher than the slowest rate among chemical reaction steps.



(a) Discharge efficiency



(b) Recovery efficiency: 2-minute discharge at rate 4 C and 8-minute rest are repeated.

Fig. 2. Physical characteristics of a rechargeable battery cell: we simulate a lithium-ion battery cell operating at 25°C, using Dualfoil [23].

Another interesting characteristic is called *recovery efficiency*. Fig. 2(b) represents the output voltage drop of a battery cell when it alternates to discharge for 2 minutes at 4 C and rest for 8 minutes. After a voltage drop during each discharge, the output voltage is recovered during each rest period. For example, after the first 2-minute discharge, the voltage is dropped to 3.78 V, but a rest for 8 minutes recovers voltage up to 3.99 V, as shown in the figure. Recovery efficiency occurs because a rest not only decreases the concentration difference

Parameter	Unit	Description
ℓ_{ORB}	Min	The orbit period
n_{S}		Configuration of solar panels determined at design time, e.g., the number, the position and the type of solar panels
n_{B}		Configuration of battery cells determined at design time, e.g., the number, the connectivity and the type of battery cells
p_i	Min	The period of τ_i
$c_i(t, v_i)$, $c_i^+(t\%\ell_{\mathsf{ORB}}, v_i)$	W	the amount of power consumption of τ_i with version v_i at t, and its upper bound
$e_{\mathsf{S}}(t)$, e_{S}^-	%	Solar panel's efficiency at t, and its lower bound
$e_{\mathsf{B}}(t)$, e_{B}^-	%	Battery cell's efficiency at t , and its lower bound
$s(t)$, s^-		Position/attitude of the satellite at t, and position/attitude that results in the minimum power supply
$\theta(t)$	$^\circ C$	Temperature at t
$t = 0$, $t = t^+$	Min	The times at the beginning and end of life (which are assumed to be the beginning of an orbit period)
$\mathsf{Dem}(t)$, $\mathsf{Dem}^+(t)$	W	The amount of power consumption at t (W), and its upper bound
$\mathsf{Sup}(t)$, $\mathsf{Sup}^-(t)$	W	The amount of power generation at t (W), and its lower bound
$\mathsf{SoC}(t)$, $\mathsf{SoC}^-(t)$	%	SoC at t, and its lower bound

TABLE I. PARAMETERS

between the reactants and products, but also takes time to finish the remaining chemical reactions that did not follow a high discharge rate, thus alleviating the polarization effect.

Similar to solar panels, battery cells are also affected by their efficiency and temperature. The efficiency of battery cells at t (denoted by $e_{\mathsf{B}}(t)$) degrades with time according to the *calendar fade effect*. A temperature change also influences deliverable capacity.

We represent the battery cells' charge status using *State of Charge* (SoC). The SoC at a given time is proportional to the amount of charge available at the time, and it is expressed in percent, from 0% when there is no charge, to 100% when it is fully charged. In this paper, we define SoC for the entire battery system, not for each cell. Then, we express SoC at $t+\Delta$ as a function of SoC at t, the amount of charged or discharged power during $[t, t+\Delta)$ (i.e., $\Delta \cdot (\mathsf{Dem}(t) - \mathsf{Sup}(t))$, n_{B}, e_{B}, and $\theta(t)$, as:

$$\mathsf{SoC}(t+\Delta) = F_{\mathsf{soc}}\Big(\mathsf{SoC}(t), \Delta \cdot \big(\mathsf{Dem}(t) - \mathsf{Sup}(t)\big), n_{\mathsf{B}}, e_{\mathsf{B}}(t), \theta(t)\Big), \quad (3)$$

where n_{B} denotes configuration of battery cells determined at design time (e.g., the number, the connectivity and the type of battery cells).

F_{soc} is a decreasing function of $\Delta \cdot (\mathsf{Dem}(t) - \mathsf{Sup}(t))$, and an increasing function of $e_{\mathsf{B}}(t)$. While the exact form of F_{soc} is complex due to the non-linear behaviors, it can be modeled and calculated numerically, e.g., see [9, 10, 23].

III. Design and Management of Satellite Power Systems

We now propose a new design and management scheme for satellite power systems that meets both G1 and G2. We first derive the properties of power supply and demand. Based on these properties, we propose a way of guaranteeing power sufficiency for a given design, and then meet G1 by applying it to different designs. We finally develop a strategy for execution of higher-version tasks without compromising the power sufficiency guarantee, thus meeting G2.

A. Properties of Power Supply and Demand

To achieve G1 and G2, we should know (i) the amount of power generated by solar panels and stored in battery cells, and (ii) the amount of power dissipated by sub-systems. Eqs. (1), (2) and (3) express what we need to know, but their exact values are available only at runtime, not at a pre-launch time. Therefore, we need to derive a lower-bound of (i) and an upper-bound of (ii), and use the bounds for achieving G1 and G2.

Based on an offline analysis, we determine the thresholds for relevant variables $e_{\mathsf{S}}(t)$, $e_{\mathsf{B}}(t)$ and $s(t)$ that hold for all $t \in [0, t^+)$, where 0 (t^+) is the time at the beginning (end) of the satellite life (and, 0 and t^+ are assumed to be the beginning of an orbit period). Here, we use the superscripts X^+ and X^- to represent upper and lower bounds of X, respectively, and define the thresholds such that $e_{\mathsf{S}}(t) \geq e_{\mathsf{S}}^-$, $e_{\mathsf{B}}(t) \geq e_{\mathsf{B}}^-$, and $c_i(t, v_i) \leq c_i^+(t\%\ell_{\mathsf{ORB}}, v_i)$ hold for all $t \in [0, t^+)$.[3] Recall that all parameters used in this paper are summarized in Table I. In Section IV, we will present specific threshold values for a satellite.

Using the thresholds, we can calculate lower bounds of $\mathsf{Sup}(t)$ and $\mathsf{SoC}(t)$, and an upper bound of $\mathsf{Dem}(t)$ as:

$$\mathsf{Sup}^-(t) = \mathsf{Sup}(t) \mid n_{\mathsf{S}}, s^-, e_{\mathsf{S}}^-, \theta(t), \quad (4)$$

$$\mathsf{Dem}^+(t) = \mathsf{Dem}(t) \mid \{c_i^+(t\%\ell_{\mathsf{ORB}}, v_i)\}_{i=1}^{|\tau|}, \quad (5)$$

$$\mathsf{SoC}^-(t) = \mathsf{SoC}(t) \mid \mathsf{Dem}^+(t), \mathsf{Sup}^-(t), n_{\mathsf{B}}, e_{\mathsf{B}}^-, \theta(t). \quad (6)$$

The functions $\mathsf{Sup}^-(t)$, $\mathsf{Dem}^+(t)$ and $\mathsf{SoC}^-(t)$ are interpreted as the minimum supply, the maximum demand, and the minimum SoC at the end of satellite life, and they are deterministic at design time since all parameters are constant or deterministic. Although $\theta(t)$ seems to be online information, we can calculate/predict the variable, because the orbit of a satellite is fixed and temperature depends on the position (meaning that $\theta(t)$ is periodic). We will give the detail of this in Section IV. The following lemma presents and proves two properties of $\mathsf{Sup}^-(t)$, $\mathsf{Dem}^+(t)$ and $\mathsf{SoC}^-(t)$: upper/lower bounds and periodicity.

[3] % means the modulo operation, so we define $c_i^+(t, v_i)$ only in $[0, \ell_{\mathsf{ORB}})$. We define s^- such that it yields the smallest power generation.

Lemma 1. *For all $t \in [0, t^+)$, the following (in)equalities holds:*

- $\mathsf{Sup}(t) \geq \mathsf{Sup}^-(t)$*;*
- $\mathsf{Dem}(t) \leq \mathsf{Dem}^+(t)$ *for given* $\{v_i\}_{i=1}^{|\tau|}$*;*
- $\mathsf{SoC}(t) \geq \mathsf{SoC}^-(t)$ *under the same initial value of the beginning of lifetime, i.e.,* $\mathsf{SoC}(0) = \mathsf{SoC}^-(0)$*;*
- $\mathsf{Sup}^-(t) = \mathsf{Sup}^-(t + \ell_{\mathsf{ORB}})$*;*
- $\mathsf{Dem}^+(t) = \mathsf{Dem}^+(t + \ell_{\mathsf{ORB}})$ *for given* $\{v_i\}_{i=1}^{|\tau|}$*; and*
- $\mathsf{SoC}^-(t) = \mathsf{SoC}^-(t + \ell_{\mathsf{ORB}})$ *under the same initial value of the beginning of each orbit period, i.e.,* $\mathsf{SoC}^-(t') = \mathsf{SoC}^-(t' + \ell_{\mathsf{ORB}})$*, where t' is the beginning of the orbit period which t belongs.*

Proof: As shown in Section II-A, $\mathsf{Sup}(t)$ is an increasing function of $e_{\mathsf{S}}(t)$. Therefore, by the definition of $e_{\mathsf{S}}(t)$ and s^-, $\mathsf{Sup}(t) \geq \mathsf{Sup}^-(t)$ holds for all $t \in [0, t^+)$. Similarly, by the definition of $c_i^+(t, v_i)$, $\mathsf{Dem}(t) \leq \mathsf{Dem}^+(t)$ for given $\{v_i\}_{i=1}^{|\tau|}$ also holds for all $t \in [0, t^+)$.

When it comes to $\mathsf{SoC}^-(t)$, we use the first two inequalities. Then, we guarantee that $\Delta \cdot \big(\mathsf{Dem}(t) - \mathsf{Sup}(t)\big)$ is no larger than $\Delta \cdot \big(\mathsf{Dem}^+(t) - \mathsf{Sup}^-(t)\big)$, which yields a smaller discharge rate and a longer rest period. Since $\mathsf{SoC}(t)$ is an increasing function of $e_{\mathsf{B}}(t)$, and $e_{\mathsf{B}}(t) \geq e_{\mathsf{B}}^-$ holds for all $t \in [0, t^+)$, the third inequality holds.

The periodicity for $\mathsf{Sup}^-(t)$ and $\mathsf{Dem}^+(t)$ (i.e., the fourth and fifth equalities) holds because all parameters in $\mathsf{Sup}^-(t)$ are constant or periodic, and $c_i^+(t, v_i)$ is periodic. Thus, all parameters in $\mathsf{SoC}^-(t)$ are either constant or periodic, and therefore the sixth equality follows. ∎

Figs. 6(b) and (c) in Section IV, and Fig. 3 in this section show examples of $\mathsf{Sup}^-(t)$, $\mathsf{Dem}^+(t)$ and $\mathsf{SoC}^-(t)$ for a satellite.

We need to guarantee that no sub-system will suffer from power shortage, and the following lemma formally expresses the guarantee only with $\mathsf{SoC}(t)$, which will be used to develop strategies for G1 and G2.

Lemma 2. *As long as $\mathsf{SoC}(t) > 0$ holds (i.e.,* strictly *larger than 0), no sub-system suffers from power deficiency.*

Proof: Sub-systems use the power generated from solar panels first, and draw the power from battery cells if the power generation is not available (during the eclipse phase) or the amount of generated power is not sufficient. Thus, no sub-system experiences power shortage, as long as battery cells can provide necessary power to sub-systems, which is expressed as $\mathsf{SoC}(t) > 0$. ∎

Lemma 2 provides a good abstraction of power-sufficiency guarantee. That is, while power supply and demand in a satellite are complex as shown in Section II, we can only find whether or not $\mathsf{SoC}(t)$ is dropped to zero. This eases the development of strategies for achieving G1 and G2.

Algorithm 1 Finding the smallest fixed point of SoC

1: $\mathsf{succ} \leftarrow 100\%$, $\mathsf{fail} \leftarrow 0\%$.
2: **for** Given number of iterations **do**
3: $\quad\mathsf{curr} \leftarrow (\mathsf{succ} + \mathsf{fail})/2$.
4: $\quad$Calculate $\mathsf{SoC}^-(t)$ for all $t \in [0, \ell_{\mathsf{ORB}})$, starting from $\mathsf{SoC}^-(0) \leftarrow \mathsf{curr}$.
5: $\quad$Check whether or not $\mathsf{SoC}^-(0) \leq \mathsf{SoC}^-(\ell_{\mathsf{ORB}})$ holds.
6: $\quad$Check whether or not $\mathsf{SoC}^-(t) > 0$ for all $t \in [0, \ell_{\mathsf{ORB}})$
7: $\quad$If Steps 5 and 6 holds, $\mathsf{succ} \leftarrow \mathsf{curr}$; otherwise, $\mathsf{fail} \leftarrow \mathsf{curr}$.
8: **end for**
9: $\mathsf{SoC}^* \leftarrow$ INFEASIBLE, if $\mathsf{succ} = 100\%$ holds; otherwise, $\mathsf{SoC}^* \leftarrow \mathsf{succ}$.
10: Return SoC^*.

B. Design-Time Guarantees on Execution of a Given Task Version (achieving G1)

At design time, we want to ensure all sub-systems do not experience power shortage until t^+ (the end of the satellite's life), for given n_{S} and n_{B} (configurations of solar panels and battery cells). Once we develop a strategy for this assurance, we test different configurations and find the best configuration that minimizes the cost, space and/or weight of power supply units, achieving G1.

To achieve this, we first consider *fixed-version execution* in which versions of tasks are determined at design time, and do not change over time. Note that this execution is not related to real-time scheduling in that tasks (sub-systems) of our interest share energy resource only, instead of competing for temporal resource in a computing unit.

We want to find the *smallest fixed point of SoC* (denoted by SoC^*), which satisfies the following two conditions for given $\mathsf{SoC}^-(t') \leftarrow \mathsf{SoC}^*$, where $t' \in [0, t^+)$ is the beginning of an orbit period: (i) $\mathsf{SoC}^-(t' + \ell_{\mathsf{ORB}}) \geq \mathsf{SoC}^*$, and (ii) $\mathsf{SoC}^-(t) > 0$ holds for all $t \in [t', t' + \ell_{\mathsf{ORB}})$. Then, if SoC at the beginning, i.e., $\mathsf{SoC}(0)$, is no smaller than SoC^*, we can guarantee the power sufficiency, which will be formally stated in Theorem 1. Algorithm 1 describes how to find a fixed point of SoC, using a binary search; note that we apply any t' (the beginning of an orbit period) since $\mathsf{SoC}^-(t)$ is a periodic function by Lemma 1, and hence, we apply $t' = 0$. Steps 5 and 6 address (i) and (ii), respectively. An example of the smallest fixed point is illustrated in Fig. 3. If SoC starts with 96.8%, 43.3% and 30.8% with Configurations D, E and F, each of SoC satisfies Steps 5 and 6. Since such values barely meet Step 6, they are the smallest SoC that satisfies Steps 5 and 6. Here, Configuration D, E and F mean 1, 2 and 3 battery cells connected in parallel, and one solar panel on each side, which will be detailed in Section IV.

Suppose that the SoC at t' is no smaller than SoC^* (i.e., $\mathsf{SoC}(t') \geq \mathsf{SoC}^*$) where $t' \in [0, t^+)$ is the beginning of an orbit period. Then, $\mathsf{SoC}(t) > 0$ holds for all $t \in [t', t' + \ell_{\mathsf{ORB}})$, by the definition of the fixed point and Lemma 1. This implies that by Lemma 2, no sub-system suffers from power deficiency in $[t', t' + \ell_{\mathsf{ORB}})$, which is formally stated in Lemma 3.

Lemma 3. *Suppose that SoC^* exists (i.e., not INFEASIBLE) in Algorithm 1. If $\mathsf{SoC}(t') \geq \mathsf{SoC}^*$ holds where $t' \in [0, t^+)$ is the beginning of an orbit period, $\mathsf{SoC}(t' + \ell_{\mathsf{ORB}}) \geq \mathsf{SoC}^*$ holds and*

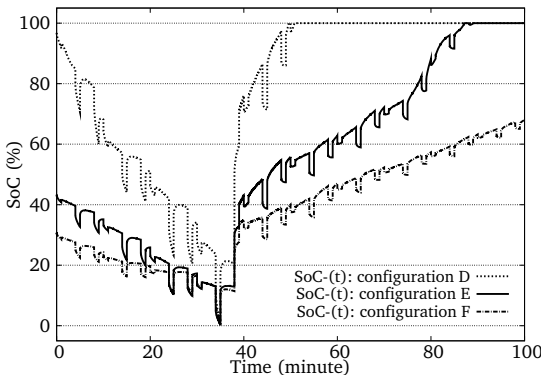


Fig. 3. An example of the minimum fixed point of $\mathsf{SoC}^-(t)$ (i.e., SoC^*) with three configurations (with $v_5 = v_6 = 1$), which are detailed in Section IV. Note that the three configurations satisfy $\mathsf{SoC}^-(t) > 0$ for $t \in [0, 100)$.

no sub-system suffers from power shortage in $[t', t' + \ell_{\mathsf{ORB}})$*, under fixed-version execution.*

Proof: SoC^* from Algorithm 1 satisfies (i) and (ii), by Steps 5 and 6, respectively. By Lemma 1, $\mathsf{SoC}(t) \geq \mathsf{SoC}^-(t)$ for all $t \in [0, t^+)$. Therefore, $\mathsf{SoC}(t' + \ell_{\mathsf{ORB}}) \geq \mathsf{SoC}^-(t' + \ell_{\mathsf{ORB}}) \geq \mathsf{SoC}^*$ holds, and $\mathsf{SoC}(t) \geq \mathsf{SoC}^-(t) > 0$ holds for all $t \in [t', t' + \ell_{\mathsf{ORB}})$. By Lemma 2, this proves the lemma. ■

Extending Lemma 3 to the entire interval of $[0, t^+)$, we finally guarantee power sufficiency as follows.

Theorem 1. *Suppose that* SoC^* *exists (i.e., not INFEASIBLE) in Algorithm 1. If* $\mathsf{SoC}(0) \geq \mathsf{SoC}^*$ *holds, no sub-system suffers from power deficiency at* $t \in [0, t^+)$ *under fixed-version execution.*

Proof: This is proved by mathematical induction. For each orbit period with its beginning time t', we prove the following proposition: if $\mathsf{SoC}(t') \geq \mathsf{SoC}^*$ is satisfied, $\mathsf{SoC}(t' + \ell_{\mathsf{ORB}}) \geq \mathsf{SoC}^*$ holds and $\mathsf{SoC}(t) > 0$ holds for all $t \in [t', t' + \ell_{\mathsf{ORB}})$.

(The basis) Since $\mathsf{SoC}(0) \geq \mathsf{SoC}^*$ holds, the proposition holds for $t' = 0$ by Lemma 3.

(The inductive step) We will prove that if the proposition holds for the k-the orbit period (whose beginning is t'), then the proposition holds for the $(k+1)$-th orbit period (whose beginning is $t' + \ell_{\mathsf{ORB}}$). Satisfying the proposition for the k-th orbit period implies $\mathsf{SoC}(t' + \ell_{\mathsf{ORB}}) \geq \mathsf{SoC}^*$. Therefore, by Lemma 3, the proposition holds for the $(k+1)$-th orbit period.

By the basis and inductive steps, the lemma holds. ■

Theorem 1 is useful because we make a power-sufficiency guarantee without investigating all intervals $[0, t^+)$; instead, what we need to do is to find SoC^* by looking at a single orbit period in Algorithm 1.

Using Theorem 1, we test all possible configurations of solar panels and battery cells, and choose one of the Pareto optimal configurations, which addresses G1. Section IV will provide an example.

Algorithm 2 Dynamic-version execution for each sub-period

1: $t' \leftarrow$ current time that is the beginning of the current sub-period, $t'' \leftarrow$ the end of the current orbit period
2: **for** All sets of versions $\{v_i\}_{i=1}^{|\tau|}$ in descending order of given task-level priorities for executing higher versions **do**
3: Calculate $\mathsf{SoC}'(t)$ for all $t \in [t', t'')$, starting from $\mathsf{SoC}'(t') \leftarrow \mathsf{SoC}(t')$.
4: Check whether or not $\mathsf{SoC}'(t'') \geq \mathsf{SoC}^*$ holds.
5: Check whether or not $\mathsf{SoC}'(t) > 0$ for all $t \in [t', t'')$.
6: If both Steps 4 and 5 hold, return the current set of versions.
7: **end for**

C. Online Power Management for Higher Version Execution (achieving G2)

We can achieve G1 with fixed-version execution by Theorem 1, which involves the worst case of all situations, abstracted by $\mathsf{SoC}^-(t)$. However, we can execute higher versions of tasks by exploiting the difference between the worst-case and the actual situations available online only. Here we develop *dynamic-version execution* that addresses G2. Note that the system is assumed to be able to measure the current SoC at any time, i.e., $\mathsf{SoC}(t)$ is measurable at any time t.

The underlying principle of dynamic-version execution is to execute tasks with the highest versions as long as power-sufficiency is guaranteed. This can be realized by making SoC at the end of each orbit period at least as much as SoC^*, because Lemma 3 ensures power-sufficiency for the next orbit period. The remaining issue is then how to determine the maximum versions of tasks to be executed in the current orbit period. For this, we divide an orbit period of ℓ_{ORB} into sub-periods, each of length δ; in Section IV-B, we will discuss how to select the sub-periods. At the beginning of each sub-period, we calculate the highest versions of tasks[4] in the current sub-period, such that SoC at the end of the current orbit period is no smaller than SoC^*, provided that fixed-version execution is applied to other sub-periods than the current sub-period in the current orbit period.

Let $\{v'_i\}_{i=1}^{|\tau|}$ denote the versions of tasks in the current sub-period (each of which satisfies $v'_i \geq v_i$), and t' and t'' denote the beginning of the current sub-period and the end of the current orbit period. Then, $\mathsf{Sup}(t)$, $\mathsf{Dem}(t)$ and $\mathsf{SoC}(t)$ are upper- or lower-bounded by $\mathsf{Sup}'(t)$, $\mathsf{Dem}'(t)$ and $\mathsf{SoC}'(t)$ for all $t \in [t', t'')$, respectively, where

$$\mathsf{Sup}'(t) = \mathsf{Sup}(t) \mid n_{\mathsf{S}}, s^-, e_{\mathsf{S}}(t), \theta(t), \tag{7}$$

$$\mathsf{Dem}'(t) = \begin{cases} \mathsf{Dem}(t) \mid \{c_i^+(t\%\ell_{\mathsf{ORB}}, v'_i)\}_{i=1}^{|\tau|}, & \text{if } t \in [t', t' + \delta), \\ \mathsf{Dem}(t) \mid \{c_i^+(t\%\ell_{\mathsf{ORB}}, v_i)\}_{i=1}^{|\tau|}, & \text{if } t \in [t' + \delta, t''), \end{cases} \tag{8}$$

$$\mathsf{SoC}'(t) = \mathsf{SoC}(t) \mid \mathsf{Dem}'(t), \mathsf{Sup}'(t), n_{\mathsf{B}}, e_{\mathsf{B}}(t), \theta(t). \tag{9}$$

[4]We assume that task-level priorities for executing higher versions are given.

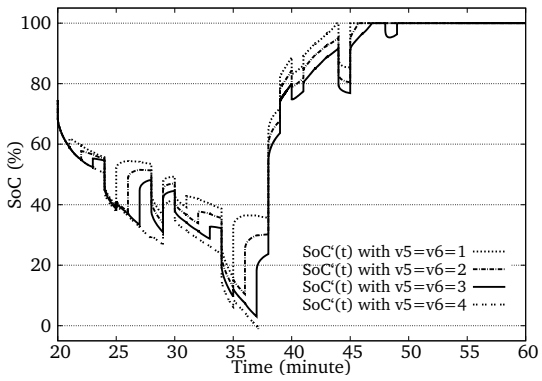


Fig. 4. An example of Algorithm 2: at $t' = 20$, the version in $[20, 40)$ is calculated, and only versions 1, 2 and 3 are feasible since version 4 does not satisfy Step 5.

Unlike Eqs. (4) and (6), $\mathsf{Sup}'(t)$ and $\mathsf{SoC}'(t)$ require non-deterministic online information, such as $e_{\mathsf{S}}(t)$ and $e_{\mathsf{B}}(t)$. This should be addressed because at t', we should know $\mathsf{Sup}'(t)$ and $\mathsf{SoC}'(t)$ for all $t \in [t', t'')$. However, we can regard $e_{\mathsf{S}}(t)$ and $e_{\mathsf{B}}(t)$ as constants in $t \in [t', t'')$ since their change during an orbit period is negligible. $e_{\mathsf{S}}(t')$ and $e_{\mathsf{B}}(t')$ are measured from sensors, or calculated by using the history of $\mathsf{Sup}(t)$ and $\mathsf{SoC}(t)$. Therefore, Eqs. (7), (8) and (9) for all $t \in [t', t'')$ are available at t' for given $\{v_i\}_{i=1}^{|\tau|}$ and $\{v'_i\}_{i=1}^{|\tau|}$.

Algorithm 2 summarizes what we have discussed so far, representing dynamic-version execution for each sub-period. The algorithm tests the highest version first, and in each loop, it investigates (i) SoC at the end of orbit period is no smaller than SoC^* in Step 4 and (ii) SoC is always positive during the current orbit period in Step 5. Fig. 4 illustrates an example of dynamic-version execution in Algorithm 2, in which $t' = 20$, $t'' = 100$ and $\delta = 20$. That is, the current time is 20, and we calculate the versions of tasks in $[20, 40)$, given the versions of tasks in $[40, 100)$ (which are the smallest versions). We focus on determining the versions of two tasks (v_5 and v_6), which will be detailed in Section IV. As shown in the figure, if the version is 1, 2 or 3, Steps 4 and 5 hold. However, there exists t such that $\mathsf{SoC}'(t) = 0$ with version 4. Therefore, the highest possible version in $[20, 40)$ is 3.

The following lemma presents a power-sufficiency guarantee for dynamic-version execution.

Theorem 2. *Suppose that SoC^* exists (i.e., not INFEASIBLE) in Algorithm 1. If $\mathsf{SoC}(0) \geq \mathsf{SoC}^*$ holds, no sub-system suffers from power deficiency at $t \in [0, t^+)$ under dynamic-version execution in Algorithm 2.*

Proof: The proof is similar to that of Theorem 1. For each orbit period starting at t', Algorithm 2 guarantees (i) $\mathsf{SoC}'(t' + \ell_{\mathsf{ORB}}) \geq \mathsf{SoC}^*$ and (ii) $\mathsf{SoC}'(t) > 0$ for all $t \in [t', t' + \ell_{\mathsf{ORB}})$. By definition, we can easily show that $\mathsf{SoC}'(t)$ is no smaller than $\mathsf{SoC}(t)$. Therefore, by Lemma 2, we prove the lemma for $t \in [t', t + \ell_{\mathsf{ORB}})$. Similar to Theorem 1, the guarantee with an orbit period is extended to the entire interval $[0, t^+)$, since $\mathsf{SoC}(t)$ at the beginning of each orbit period is no smaller than SoC^*. ■

By Theorem 2, we achieve G2 efficiently because Algorithm 2 investigates the current orbit period only. The efficiency of the algorithm is important because dynamic-version execution makes decisions online. In Section IV, we will evaluate the effectiveness of dynamic-version execution in terms of average executed versions of tasks. As expected, the sub-period length of δ affects performance; the smaller δ, the higher performance and the cost of more overhead.

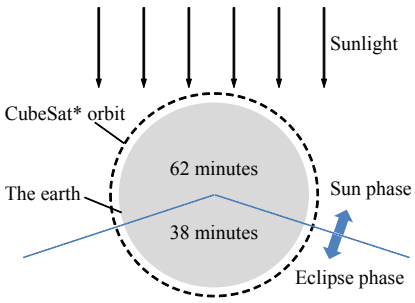


Fig. 5. CubeSat*'s orbit—100-minute orbit with 38-minute eclipse phase and 62-minute sun phase

IV. Case Study: CubeSat*

In this section, we study a cubic-shaped nano-satellite, called *CubeSat**, to exploit its simple, manipulable architecture. We first present its architecture, power supply units, and sub-systems that consume power, with specific configuration and parameters. Then, we evaluate the effectiveness of the design and management of power systems proposed in Section III.

A. Architecture, Power Supply and Demand

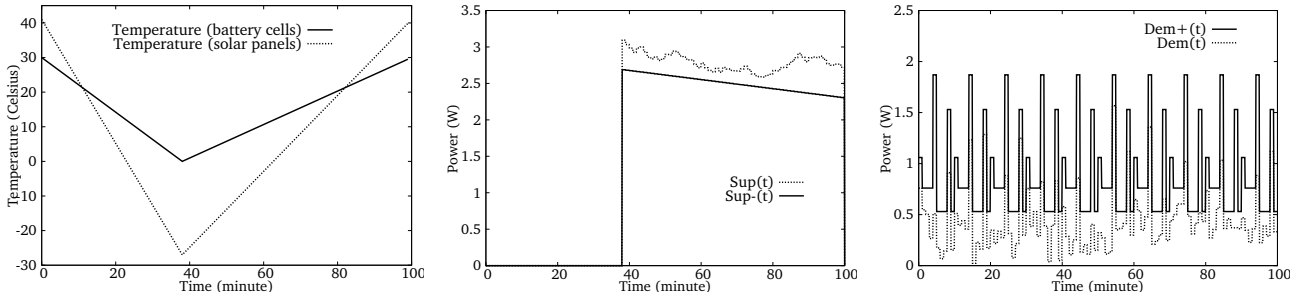
CubeSat is a nano-scale satellite, which has a cubic shape and weighs about 1kg. It was proposed by Robert Twiggs at Stanford University, and originated from an educational intent. Due to its low cost and short production time, many CubeSats were launched since 2004, and now they can perform commercial experiments (see a survey [24]). Examples of CubeSats include CADRE [18, 19], SwissCube [16, 17], and CanX [14, 15]. CubeSat is a proper target system for the design and management of power systems of satellites; its simple, manipulable architecture eases the design and management of power systems as well as the demonstration of their effectiveness, and it is a "real" system.

Of many CubeSats, we target SwissCube [16, 17] and CanX [14, 15], and borrow their system architecture, configurations and parameters. We call our target CubeSat, CubeSat*.

CubeSat* revolves around the earth every 100 minutes, i.e., $\ell_{\mathsf{ORB}} = 100$. As shown in Fig. 5, the eclipse and sun phases last 38 and 62 minutes, respectively. In CubeSat*, solar panels can be attached to five of six sides; one side is reserved for a channel for the camera. The inside of CubeSat* is occupied by battery cells, an onboard computer, a magnetorquer, a magnetometer, a receiver, a transmitter, and a camera. The onboard computer is responsible for controlling all sub-systems, and the magnetorquer and magnetometer stabilize CubeSat*, with attitude control. The receiver and the transmitter communicate

Task τ_i	Period p_i (Min)	Maximum power consumption $c_i^+(t, v_i)$ (W)
τ_1: Onboard computer	100.0	$\mathbf{0.38}$
τ_2: Receiver	100.0	$\mathbf{0.15}$
τ_3: Magnetorquer	10.0	$\mathbf{1.00}$, if $8.0 \leq t\%10.0 \leq 9.0$
τ_4: Magnetometer	10.0	$\mathbf{0.23}$, if $0.0 \leq t\%10.0 \leq 5.0$
τ_5: Transmitter	10.0	$\mathbf{1.11}$, if $4.0 \leq t\%10.0 \leq 4.0 + v$ ($v_4 = 1, 2, 3$ or 4)
τ_6: Camera	10.0	$\mathbf{0.3}$, if $0.0 \leq t\%10.0 \leq v$ ($v_5 = 1, 2, 3$ or 4)

TABLE II. CUBESAT*'S SUB-SYSTEMS MODELED AS PERIODIC TASKS



(a) Temperature ($\theta(t)$) for battery cells and solar panels

(b) $\mathsf{Sup}(t)$ and $\mathsf{Sup}^-(t)$ under $n_\mathsf{S} = 2$ and $e_\mathsf{S}(t) = e_\mathsf{S}^-$

(c) $\mathsf{Dem}^+(t)$ and $\mathsf{Dem}(t)$

Fig. 6. Profiles of temperature, power supply from solar panels and power demand from sub-systems

with the ground. The camera periodically takes pictures, which is the mission objective of CubeSat*.

We express each sub-system as a periodic task τ_i. Table II summarizes task parameters, which can be determined/measured offline (before launching the satellite). Note that the magnetorquer, magnetometer, transmitter and camera are actual periodic tasks, while the onboard computer and receiver can consume power at any time, thus assigning $p_1 = p_2 = \ell_{\mathsf{ORB}} = 100$.

Due to the limited budget of space and weight, we only consider two configurations of solar panels: one or two solar panels are attached on each side. We use n_S as the number of solar panels on each side, and therefore $n_\mathsf{S} = 1$ or 2. Each solar panel is triple-junction gallium-arsenide. Up to three battery cells can be accommodated, and the number of battery cells is denoted as n_B. Each battery cell is a polystor lithium-ion cell, and all battery cells are connected in parallel.

A mission lifetime of CubeSat* is 2 years, i.e., $t^+ = 1051200$ minutes.

B. Evaluation

To evaluate our design and management of CubeSat*, we need actual data and simulation tools for its operation environments, power supply from solar panels, power demand from sub-systems, and SoC of battery cells.

Temperature. To reflect a harsh thermal environment, we assume that the temperature of solar panels ranges from $-27°C$ (when each eclipse phase ends) to $41°C$ (when each eclipse phase starts). Since battery cells are shielded with insulators, we assume that their ambient temperature varies within $[0°C, 30°C)$. We assume that both temperatures vary linearly, and Fig. 6(a) shows temperature transition during an orbit period, where $t = 0$ and $t = 38$ are the beginning and end of the eclipse period.

Power supply from solar panels. According to [17], the worst-case generated power is between $w_{min} = 2.30W$ and $w_{max} = 3.26W$ under $n_\mathsf{S} = 2$. Based on the analysis of power generation with different temperature and efficiency in [17], we determine $\mathsf{Sup}(t)$ and $\mathsf{Sup}^-(t)$ during the sun phase as:

$$\mathsf{Sup}(t) = w(t) \cdot \frac{e_\mathsf{S}(t)}{e_\mathsf{S}^-} \cdot \frac{-0.00261 \cdot \theta(t) + 1.161}{-0.00261 \cdot 41 + 1.161} \cdot \frac{n_\mathsf{S}}{2},$$

$$\mathsf{Sup}^-(t) = w_{min} \cdot \frac{-0.00261 \cdot \theta(t) + 1.161}{-0.00261 \cdot 41 + 1.161} \cdot \frac{n_\mathsf{S}}{2},$$

where $w(t)$ is a random variable in $[w_{min}, w_{max})$, which varies with $s(t)$. Fig. 6(b) illustrates $\mathsf{Sup}(t)$ and $\mathsf{Sup}^-(t)$ under $n_\mathsf{S} = 2$ and $e_\mathsf{S}(t) = e_\mathsf{S}^-$ (which means that the time is near the end of lifetime). There is no power generation during $0 \leq t < 38$ (the eclipse phase), and after $t = 38$, $\mathsf{Sup}^-(t)$ decreases with time, due to temperature increase.

Power demand from sub-systems. Since $c_i^+(\cdot)$ is deterministic as shown in Table II, $\mathsf{Dem}^+(t)$ is simply calculated by using Eq. (5). We calculate the actual power consumption as: $\mathsf{Dem}(t) = \sum_{i=1}^{6} c_i^+(t\%100, v_i) * y(t)$, where $y(t)$ is a random variable in $[0, 1)$. Fig. 6(c) illustrates $\mathsf{Dem}^+(t)$ and $\mathsf{Dem}(t)$.

SoC of battery cells. To simulate SoC of battery cells, we use a popular battery simulator, Dualfoil [23], which simulates the behavior of rechargeable battery cells including a lithium-ion battery cell, based on an electro-chemical model that expresses its charge/discharge [9]. By setting the load profile as a sequence of constant current steps, we can obtain the physical and electrical states of battery cells until the voltage is dropped below the cutoff voltage. Dualfoil effectively addresses physical characteristics of rechargeable battery cells, including discharge efficiency, recovery efficiency and temperature's effect.

The input of Dualfoil (load profile) is current, so we calculate current from power and nominal voltage, i.e., power

Configuration name $(n_{\mathsf{S}}, n_{\mathsf{B}})$	Power-sufficiency guarantee with $v_5 = v_6 = 1, 2, 3, 4$
A (1,1)	X X X X
B (1,2)	X X X X
C (1,3)	X X X X
D (2,1)	O X X X
E (2,2)	O O O O
F (2,3)	O O O O

TABLE III. POWER-SUFFICIENCY GUARANTEE WITH DIFFERENT CONFIGURATIONS OF SOLAR PANELS AND BATTERY CELLS WITH DIFFERENT VERSIONS OF $\tau_5 = \tau_6$, UNDER FIXED-VERSION EXECUTION

= current $\times$ voltage. That is, using the profiles of demand and supply, we calculate the amount of power demanded or supplied, and translate it to current. We feed the current with temperature profile to Dualfoil, and then obtain the output voltage, which is converted to SoC.

Now, we present the evaluation results using the settings discussed so far.

Design of CubeSat*'s power system. We have two configurations of solar panels ($n_{\mathsf{S}} = 1$ and 2) and three configurations of battery cells ($n_{\mathsf{B}} = 1$, 2, and 3), implying a total of six configurations as shown in Table III. We test Algorithm 1 for all configurations A–F when the version of τ_5 and τ_6 is fixed at 1, 2, 3 and 4, in total 24 cases. Once there exists SoC^* derived by the algorithm, Theorem 1 proves the power-sufficiency guarantee. Note that the version of τ_5 depends on that of τ_6, because a higher resolution picture entails a large amount of data to be transmitted. Therefore, we only consider the same version of τ_5 and τ_6.

As shown in Table III, the power-sufficiency guarantee cannot be made even with the lowest version, if there is only one solar panel on each side (see configurations A, B and C), because there does not exist SoC^* with these configurations. As shown in Fig. 7, when $\mathsf{SoC}^-(t)$ at $t = 0$ (the beginning of an orbit period) is set to 100.0%, $\mathsf{SoC}^-(t)$ at $t = 100$ (the end of an orbit period) is only 84.3%, 92.5%, and 97.3% for configurations A, B and C; for any given $\mathsf{SoC}^-(0)$, there does not exist SoC^* with the configurations. On the other hand, when the number of solar panels on each side is two, (i.e., $n_{\mathsf{S}} = 2$), we can make power-sufficiency guarantees. If we focus on configuration D, SoC^* exists only with $v_5 = v_6 = 1$. Also, configurations E and F make power-sufficiency guarantees with any $v_5 = v_6$.

This can be interpreted as follows: (i) the case of $n_{\mathsf{S}} = 1$ cannot supply sufficient power even for $v_5 = v_6 = 1$ (the smallest demand); and (ii) the case of $n_{\mathsf{S}} = 2$ can for $v_5 = v_6 = 4$ (the largest demand). Further credibility to (i) is given by the fact that the initial SoC does not recover at the end of the orbit period with $n_{\mathsf{S}} = 1$, regardless n_{B} in Fig. 7. On the other hand, if $n_{\mathsf{S}} = 2$, the bottleneck is the performance of battery cells. That is, as shown in Fig. 3 in Section III, if $n_{\mathsf{B}} = 1$, SoC^* barely exists with $v_5 = v_6 = 1$; if we demand more power ($v_5 = v_6 \geq 2$), SoC^* does not exist. However, if we increase the number of battery cells (i.e., $n_{\mathsf{B}} = 2$ or 3), we can guarantee power-sufficiency with any version of $v_5 = v_6$.

Based on the results presented in Table III, the Pareto optimal points are configurations D and E. That is, at design time, we can guarantee the execution of the lowest version of τ_5 and τ_6 without experiencing power shortage, if we employ configuration D which incurs less cost. We can choose configuration E for the execution of the highest version of τ_5 and τ_6, at the expense of more design cost. We do not choose configuration F, since it incurs more cost than configuration E, while yielding the same guarantee on the executed version.

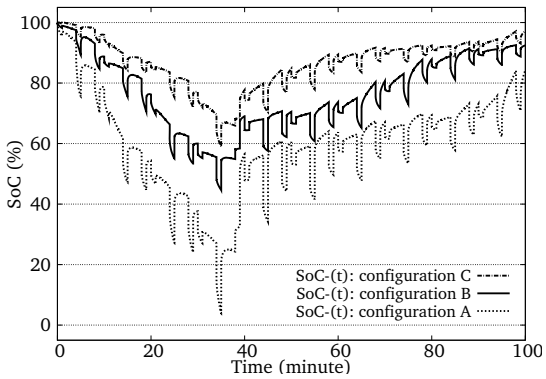


Fig. 7. $\mathsf{SoC}^-(t)$ with $\mathsf{SoC}^-(0) = 100$ under configurations A, B and C with $v_5 = v_6 = 1$

Online management of CubeSat*'s power system. While configuration D is design-time Pareto optimal, we can execute higher-version tasks using online information. Thus, we focus on configuration D, and evaluate the effectiveness of dynamic-version execution described in Algorithm 2, in terms of average executed versions of $v_5 = v_6$. As mentioned in Section III, the effectiveness depends on δ, the frequency of triggering the calculation of the highest possible version. Since the period of τ_5 and τ_6 is 10 and the orbit period is 100, we choose δ as 10, 20, 50, and 100. For $\mathsf{Sup}(t)$, we are interested in the time instant near the end of lifetime, thus setting $e_{\mathsf{S}}(t)$ to e_{S}^-.

Table IV presents the executed version of v_5 and v_6 in each interval during an orbit period. With $\delta = 100$, the decision on the executed version is made only at $t = 0$, resulting in $v_5 = v_6 = 1$ for the entire orbit period. Since the sun phase starts at 38, every decision after 38 is $v_5 = v_6 = 4$, e.g., at $t = 50$ with $\delta = 50$, at $t = 40$, 60 and 80 with $\delta = 20$, and $t = 40$, 50, $\cdots$, 90 with $\delta = 10$. During the eclipse phase, the cases of $\delta = 20$ and 10 can execute higher versions. As a result, while the average number of versions with $\delta = 100$ is 1.0, those with $\delta = 50$, 20, and 10 are 2.5, 3.2 and 3.7, respectively. In particular, we can execute the highest version most of the time with $\delta = 10$. Since the smaller δ, the higher computation overhead, there is a tradeoff between performance (average executed version) and overhead.

One may wonder if Algorithm 2 requires a simulation of Dualfoil, which is not suitable when the computing power of a satellite is low. In this case, we can choose a finite number of simulation inputs and store the results offline. During the operation, we calculate SoC by applying the nearest (but safely upper-bounded) simulation input.

Summary. Using actual/realistic profiles and simulator, we demonstrated that the proposed design can find the Pareto optimal configurations D and F. We also showed that the proposed

Interval	[0,10)	[10,20)	[20,30)	[30,40)	[40,50)	[50,60)	[60,70)	[70,80)	[80,90)	[90,100)	Average
δ =100	1	1	1	1	1	1	1	1	1	1	1.0
δ =50	1	1	1	1	1	4	4	4	4	4	2.5
δ =20	1	1	3	3	4	4	4	4	4	4	3.2
δ =10	1	4	4	4	4	4	4	4	4	4	3.7

TABLE IV. VERSION OF τ_5 AND τ_6 UNDER DYNAMIC-VERSION EXECUTION WITH VARIOUS δ

dynamic-version execution can exploit online information, and guarantee higher-version operations of tasks.

V. CONCLUSION

In this paper, we proposed the design and management of satellite power systems. Based on the analysis of characteristics of power supply and demand, we first proposed a solution to judge at design time whether or not power-sufficiency with a given configuration is guaranteed, and this solution is used to find the best configuration. Second, we developed an online dynamic-version execution policy, which executes the highest possible versions of tasks without compromising the power-sufficiency guarantee. Using a case study of CubeSat*, we demonstrated the effectiveness of our design and management of satellite power systems.

While our solution is dedicated to a small number of battery cells that are connected in parallel, large-scale satellites need a battery management system (BMS) to handle a large number of battery cells. In such a case, a reconfigurable architecture is promising in that it precludes a BMS from the failure caused by a single-cell failure, and enables the BMS to form an efficient configuration based on external conditions and electrical states of battery cells. The BMS should then address how to control charge, discharge, and rest of individual battery cells. In future, we would like to develop the BMS associated with solar panels, which achieves our design and management goals for large-scale satellites.

We have made some implicit assumptions for concise presentation; for example, task periods are no longer than the orbit period, and the orbit period is an integer multiple of each task period. In future we would like to relax these assumptions.

ACKNOWLEDGEMENT

We are grateful to Justin M. Bradley and Ella M. Atkins of University of Michigan Aerospace Engineering for helpful references and comments on CubeSat. The work reported in this paper was supported in part by the NSF under grants CNS-0930813, CNS-1138200, and CNS-1329702.

REFERENCES

[1] P. Fortescue, G. Swinerd, and J. Stark, *Spacecraft Systems Engineering*. Wiley, the 4th edition, 2011.

[2] A. C. Fu, E. Modiano, and J. N. Tsitsiklis, "Optimal energy allocation and admission control for communications satellites," *IEEE/ACM Transactions on Networking*, vol. 11, no. 3, pp. 488–500, 2003.

[3] F. Alagoz and G. Gur, "Energy efficiency and satellite networking: A holistic overview," *Proceedings of the IEEE*, vol. 99, no. 11, pp. 1954–1979, 2011.

[4] National Science Foundation (NSF), "Cyber-physical systems (CPS)," http://www.nsf.gov/pubs/2008/nsf08611/nsf08611.htm [Accessed 3 May, 2013].

[5] E. A. Lee, "Cyber-physical systems - are computing foundations adequate?" in *NSF Workshop on Cyber-Physical Systems*, 2006.

[6] ——, "Cyber physical systems: Design challenges," in *Proceedings of the 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, 2008, pp. 363–369.

[7] A. Kandhalu and R. Rajkumar, "Qos-based resource allocation for next-generation spacecraft networks," in *Proceedings of IEEE Real-Time Systems Symposium (RTSS)*, 2012, pp. 163–172.

[8] A. T. Klesh, J. W. Cutler, and E. M. Atkins, "Cyber-physical challenges for space systems," in *Proceedings of IEEE/ACM International Conference on Cyber-Physical Systems (ICCPS)*, 2012, pp. 45–52.

[9] T. F. Fuller, M. Doyle, and J. Newman, "Simulation and optimization of the dual lithium ion insertion cell," *Journal of the Electrochemical Society*, vol. 141, no. 1, pp. 1–10, 1994.

[10] D. Linden and T. Reddy, *Handbook of Batteries, the third edition*. McGraw-Hill Companies, 2001.

[11] H. Kim and K. G. Shin, "Scheduling of battery charge, discharge, and rest," in *Proceedings of IEEE Real-Time Systems Symposium (RTSS)*, 2009, pp. 13–22.

[12] J. C. Pemberton and F. Galiber, "A constraint-based approach to satellite scheduling," in *Proceedings of the DIMACS Workshop on Contraint Programming and Large Scale Discrete Optimization*, 2001, pp. 101–114.

[13] S. Waydo, D. Henry, and M. Campbell, "Cubesat design for leo-based earth science mission," in *Proceedings of the IEEE Aerospace Conference*, 2002, pp. 435–445.

[14] G. J. Wells, L. Stras, and T. Jeans, "Canada's smallest satellite: The canadian advanced nanospace experiment (CanX-1)," in *Proceedings of the 16th Annual AIAA/USU Conference on Small Satellites*, 2002.

[15] G. Bonin, D. Sinclair, and R. E. Zee, "Peak power tracking on a nanosatellite scale: The design and implementation of digital power electronics on the sfl generic nanosatellite bus," in *Proceedings of the 23th Annual AIAA/USU Conference on Small Satellites*, 2009.

[16] "SwissCube," http://swisscube.epfl.ch/ [Accessed 3 May, 2013].

[17] F. Jordan, "SwissCube: Electrical power system, final report," http://ctsgepc7.epfl.ch/03%20-%20Power%20system/S3-A-EPS-1-0-EPS.pdf [Accessed 3 May, 2013].

[18] "CADRE: Cubesat investigating atmospheric density response to extreme driving," *Aerospace Engineering 483 Final Report, University of Michigan*, pp. 1–126, April 15, 2011.

[19] J. W. Cutler, A. Ridley, and A. Nicholas, "Cubesat investigating atmospheric density response to extreme driving (CADRE)," in *Proceedings of the 25th Annual AIAA/USU Conference on Small Satellites*, 2011.

[20] C. Rusu, R. Melhem, and D. Mosse, "Multi-version scheduling in rechargeable energy-aware real-time systems," *Jounral of Embedded Computing*, pp. 2–11, 2004.

[21] C. Moser, D. Brunelli, L. Thiele, and L. Benini, "Real-time scheduling with regenerative energy," in *Proceedings of Euromicro Conference on Real-Time Systems (ECRTS)*, 2006, pp. 261–270.

[22] S. Liu, Q. Qui, and Q. Wu, "Energy aware dynamic voltage and frequency selection for real-time systems with energy harvesting," in *Proceedings of Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2008, pp. 236–241.

[23] J. Newman, "FORTRAN programs for the simulation of electrochecmical systems," http://www.cchem.berkeley.edu/jsngrp/fortran.html [Accessed 3 May, 2013].

[24] M. Polaschegg, "Study of a cube-sat mission," *The degree of Master of Sciences of the Post-Graduate University Course Space Sciences, Karl Franzens University of Graz*, pp. 1–103, 2005.