


Optimal Power Management with Guaranteed Minimum Energy Utilization for Solar Energy Harvesting Systems

Journal Article**Author(s):**

Ahmed, Rehan; Buchli, Bernhard; Draskovic, Stefan; [Sigrist, Lukas](#) ; Kumar, Pratyush; Thiele, Lothar

Publication date:

2019-08

Permanent link:

<https://doi.org/10.3929/ethz-b-000352540>

Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

Originally published in:

ACM Transactions on Embedded Computing Systems 18(4), <https://doi.org/10.1145/3317679>

Optimal Power Management With Guaranteed Minimum Energy Utilization For Solar Energy Harvesting Systems

REHAN AHMED, BERNHARD BUCHLI, STEFAN DRASKOVIC, LUKAS SIGRIST, PRATYUSH KUMAR, and LOTHAR THIELE, Swiss Federal Institute of Technology (ETH), Switzerland

In this work we present a formal study on optimizing the energy consumption of energy harvesting embedded systems. To deal with the uncertainty inherent in solar energy harvesting systems, we propose the Stochastic Power Management (SPM) scheme, that builds statistical models of harvested energy based on historical data. The proposed stochastic scheme, maximizes the lowest energy consumption across all time intervals, while giving strict probabilistic guarantees on not encountering battery depletion. For situations where historical data is not available, we propose the use of (i) a Finite Horizon Control (FHC) scheme, and (ii) a non-uniformly scaled energy estimator based on an astronomical model, which is used by FHC. Under certain realistic assumptions, the FHC scheme can provide guarantees on minimum energy usage that can be supported over all times. We further propose and evaluate a piece-wise linear approximation of FHC for efficient implementation in resource constrained embedded systems. With extensive experimental evaluation for eight publicly available datasets and two datasets collected with our own deployments, we quantitatively establish that the proposed solutions are highly effective at providing a guaranteed minimum service level, and significantly out-perform existing solutions.

CCS Concepts: • **Theory of computation** → **Stochastic control and optimization**; *Scheduling algorithms*; • **Computer systems organization** → **Firmware**; *Embedded hardware*; • **Hardware** → **Renewable energy**;

ACM Reference Format:

Rehan Ahmed, Bernhard Buchli, Stefan Draskovic, Lukas Sigrist, Pratyush Kumar, and Lothar Thiele. 2019. Optimal Power Management With Guaranteed Minimum Energy Utilization For Solar Energy Harvesting Systems. *ACM Trans. Embedd. Comput. Syst.* 18, 4, Article 30 (July 2019), 25 pages. <https://doi.org/10.1145/3317679>

1 INTRODUCTION

Motivation. Advances in Wireless Sensor Network (WSN) research have enabled many new application scenarios in diverse scientific fields, e.g., geology [5], ecology [10], and zoology [31][27]. These WSNs are comprised of sensor motes, which contain processing, communication, and various sensing capabilities. In many scenarios, these sensors motes are in remote locations and/or are very large in number. Therefore, powering sensor motes through batteries alone is expensive or possibly intractable. The golden objective is to make sensor motes energy autonomous; such that they are able to operate autonomously over long periods of time. To this end, ambient energy harvesting has been shown to have significant potential in increasing the lifetime of sensor motes. However, despite significant research in this field (power subsystem planning, e.g., [16][8], hardware design optimization, [22][19][23], harvesting-aware dynamic power management, e.g., [29][24][7][25]), there are challenges that prevent wider adoption of these systems. Specifically, due to high variation in energy harvesting sources (such as solar energy), it is difficult to provide minimum performance guarantees. This inherent limitation has so far prevented solar energy harvesting WSNs to enter

Authors' address: Rehan Ahmed; Bernhard Buchli; Stefan Draskovic; Lukas Sigrist; Pratyush Kumar; Lothar Thiele, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, firstname.lastname@tik.ee.ethz.ch.

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *ACM Transactions on Embedded Computing Systems*, <https://doi.org/10.1145/3317679>.

important application domains, e.g., safety-critical [2], and long-term monitoring and surveillance [5], where performance guarantees must be given. In this work, we propose dynamic power management schemes that alleviate this issue.

As a use-case, we can think of an early warning system for natural disasters (e.g. [2]). Assume that this system comprises of several geographically distributed sensor motes. Each sensor mote is powered using a battery supplemented by a solar energy harvester. For this system to be effective, sensor motes have to be operational at all times. During normal sunny days, this is easy to guarantee; since the ample solar energy can be utilized as a power source. However, during bad weather conditions and during night time, powering the system is challenging due to absence/scarcity of harvesting energy. In these times of energy deficit, the battery should have sufficient charge to power the sensor mote. Furthermore, the amount of harvesting solar energy over long periods also varies. For example, solar irradiation in summer is typically much higher compared to winter. For this reason, at times of energy surplus, a sensor mote in our use-case has to *provision* for times of energy deficit by storing charge in the battery.

The first objective of all power management schemes proposed in this work is to maximize the minimum energy consumption that can be supported over all time intervals. This correlates to minimum service; enabling application of the proposed solution to safety critical application domains. This minimum service level has to be always guaranteed. If this can not be achieved, the system may not be responsive, which – for safety reasons – must be avoided at all cost. As a secondary objective we would like to maximize the system utility by leveraging energy surplus. This “free” energy can be used, e.g., to increase the sensing resolution in time and precision, perform local processing, communicate more frequently, and participate in packet forwarding to relieve the burden from energy starved or otherwise overloaded motes.

Contemporary approaches to harvesting-aware power management can be classified as (i) predictive, and (ii) reactive approaches. Predictive approaches, e.g., [16][24], attempt to improve the system utility by *predicting* the harvesting energy during a future time slot, and adapt the performance level accordingly. However, predicting future meteorological conditions is highly complex and may be computationally prohibitive [9]. Therefore, accurate prediction has so far only been possible for short prediction windows, i.e., minutes to hours [7]. *Reactive* approaches, on the other hand, schedule the service level in response to source variations, either by directly measuring the energy generation, or through monitoring the storage fill-level [29][18]. Both of these general approaches do not provide minimum service guarantees; which is their common limitation.

Approach and Summary of Results. In contrast to previous work, we formally study the energy harvesting problem with the objective of maximizing the minimum energy consumption across all time intervals. We call this parameter v . To this end, we first study a simplified clairvoyant setting wherein the harvested energy for a given time interval is exactly known in advance. With this assumption, we derive an optimal algorithm that computes the temporal energy profile called use function ($u(t)$) such that v is maximized. To identify an efficient algorithm to compute the optimal use function, we establish a relation between our harvesting problem and the shortest Euclidean path problem [21]. This relation allows the use of well-studied and efficient algorithms for the clairvoyant harvesting problem.

Then we study the more general and realistic problem, wherein the harvested energy is *not* known exactly; instead, only a conservative estimate is available. With this assumption we show that a Finite Horizon Control (FHC) scheme can be used to adaptively update $u(t)$ at runtime. We further prove that, under certain assumptions of the estimate, such a FHC scheme is guaranteed to provide a minimum energy usage that is better than a non-adaptive scheme.

Lastly, we present a Stochastic Power Management (SPM) approach which uses historical data of harvested energy to build statistical models. The motivation behind SPM is that in many settings, the conservative estimate of harvested energy is either not available or is overly pessimistic (e.g. ≈ 0). Therefore FHC is not expected to function well. SPM uses the statistical models to maximize v such that the probability of power outage (battery level reaching 0) is $\leq \lambda$. For a given system, λ is a design parameter. A low value of λ corresponds to a more conservative minimum power level and vice-versa.

A natural concern is whether the presented algorithms can be implemented on resource constrained low-power embedded devices typically used in WSNs. To demonstrate the applicability SPM, the scheme is implemented and its operation is validated on an energy harvesting testbed. SPM is shown to have negligible runtime computation overhead, since all computations are done offline, and at runtime, only one lookup and compare operation is required. For the FHC scheme, we present an approximate scheme where the use function is computed offline and represented in a look-up table. At runtime, only a few simple arithmetic operations with the values in the look-up table are required to obtain the use function $u(t)$.

We experimentally validate the different proposed theoretical ideas on datasets available in a public database and raw data collected with our own deployment [5]. While the theoretical analysis is presented without taking conversion and storage inefficiencies into account, the experiments consider battery charging efficiency in order to evaluate the proposed approaches in a realistic setting. With the experiments we show the following key results:

- (1) Clairvoyant (CV) algorithm computes optimal use functions that satisfy certain necessary conditions, as required by the connection to the Euclidean shortest path problem.
- (2) Finite Horizon Control (FHC) relying on harvested energy estimation scheme in [8] leads to failure states. To mitigate this, we propose a non-uniformly scaled estimator. Using the new estimator, the algorithm computes sustainable use functions for datasets from diverse locations.
- (3) FHC improves the minimum use function value (v) compared to baseline approaches. FHC v is on average 14.9% less than CV v .
- (4) Look-up Table (LUT) based scheme results in only a marginal loss in performance with respect to the FHC scheme. LUT v is on average 17.5% less than CV v .
- (5) Stochastic Power Management (SPM) outperforms FHC and LUT while providing strict guarantee on failure probability. SPM v is on average 6.3% less than CV v with probability of encountering failure state within a calendar year set to 0.01.
- (6) SPM is implemented on an energy harvesting test-bed and is shown to have negligible runtime computation and memory overhead. Measurements of energy consumption and battery level from the implementation closely match their corresponding expected values acquired from simulation; with Mean Square Error (MSE) of 0.317 J^2 in energy consumption and MSE of 0.071 J^2 in battery energy level.

2 RELATED WORK

We discuss related work in two areas: Optimal energy allocation, and dynamic power management. In this work we extend the former, and use the latter as baselines for evaluation.

Optimal Energy Utility. A closely related problem is studied in [11]. The authors consider throughput maximization of a network of energy harvesting sensor motes. They study the problem for a single mote, and then as a distributed algorithm for multiple motes. They establish a relation to the shortest path problem in a simply connected space for a single mote with known harvested energy. For multiple nodes they propose a heuristic that is optimal under the assumption that each

mote receives *homogeneous* harvesting energy. In contrast, we study the objective of maximizing the minimum used energy (v) over all time steps for a single mote. Interestingly, we also establish a relation between this problem and a Euclidean path minimization problem. Further, we study how to model and factor the variability in the harvested energy and, in contrast to [11], we extensively validate our approach with data from our own deployments, and from public databases.

Dynamic Power Management. The first dynamic power management scheme for solar energy harvesting Wireless Sensor Network (WSN) was proposed within a theoretical framework that defines Energy Neutral Operation (ENO) as the fundamental limit of energy harvesting systems [16]. ENO is achieved if the system consumes only as much energy as is harvested over a given time period. The day is discretized into equal duration slots, and the expected energy input for each slot is learned with an Exponentially Weighted Moving Average (EWMA) filter. The energy use for each slot is then computed by considering the mismatch between expected and actual energy input.

Weather Conditioned Moving Average (WCMA) [24] improves upon EWMA's prediction accuracy. This scheme not only considers the harvested energy in the same time slot during previous days, but also incorporates current weather conditions to obtain the expected energy input for a particular time slot. While achieving almost three-fold improvements in prediction accuracy over EWMA, it is not clear if and how this improvement translates into increased system performance.

In [29], a model-free approach to dynamic performance scaling is presented. We refer to this scheme as Energy Neutral Operation with Maximal energy efficiency (ENO-Max). Linear-Quadratic Tracking is used to dynamically adapt the system's duty-cycle based on the battery State of Charge (SoC), in order to attempt ensuring ENO while maximizing the energy efficiency. For the datasets evaluated, the authors report between 6% and 32% improvement in mean duty-cycle, and between 6% and 69% reduction in duty-cycle variance over EWMA. While presenting a low-complexity solution, ENO-Max suffers from relatively high duty-cycle variance, and relies on an accurate battery SoC approximation algorithm.

Finally, a new and promising approach that takes uninterrupted long-term operation as the ultimate goal is presented in [7]. We refer to this approach as Long-Term Dynamic Power Management (LT-DPM). LT-DPM relies on an astronomical model to provision the battery and panel, and leverages the same model at runtime to adjust the utilization such that long-term ENO may be maintained. The authors show that the approach can adjust to deviations from the design time model, while maintaining a higher minimum service-level than the above approaches.

3 PROBLEM DEFINITION

This section introduces a formal system model including the corresponding power management problem. We will also discuss the reasoning behind the model abstractions and show that they are feasible for a large class of energy harvesting systems. Adaptations that are necessary for application to realistic scenarios are described in *Sec. 5.5* and *7*.

3.1 System Model

A control algorithm that influences the energy consumption and service of an application will usually perform its function not continuously but at multiples of a fixed time interval. Therefore, we will use a *discrete time system model*, i.e., time $t \in \mathbb{N}$. The time difference of one is named a *unit time interval*, and it could represent a minute, a day, a week or even a month. The horizon of interest is some known interval $[0, T)$.

The *energy storage element* (in our case a battery) has stored energy $b(t)$ at time t , and a maximum capacity B . For the formal discussion we suppose a loss-free model, i.e., there are no leakage and store-consume inefficiencies. However, for the experimental validation we consider battery charging

efficiency (see Sec. 7). The *harvested* and *used energy* during the time interval $[t, t + 1)$ are denoted as $p(t)$ and $u(t)$, respectively. While $p(t)$ is provided by the environment, $u(t)$ is under the influence of the controller. We suppose that the controller can change parameters and modes of the running applications and the underlying hardware such that the used energy matches $u(t)$. As typical examples, the controller may change the duty-cycle of sensing, communication or computation activities, and/or switch certain components on or off to adjust the consumed energy. The battery level can be calculated by the following equation:

$$b(t + 1) = \max\{\min\{b(t) + p(t) - u(t), B\}, 0\}. \quad (1)$$

If $b(t) + p(t) - u(t) < 0$, the system is in a *failure state* at time t , due to battery depletion. We call a use function $u(t)$ *feasible* only if there are no failure states in $[0, T)$. In practical scenarios, a battery may not be allowed to drop to a zero energy level. This can be considered by assuming a virtually smaller battery than B as input for the proposed algorithm.

We assume that a lower value of $u(t)$ leads to a lower service in interval $[t, t + 1)$, e.g., a lower sampling rate or lower data resolution. This fact will be modeled by defining the *utility* $U(t_1, t_2)$ of the use function in time interval $[t_1, t_2)$ as follows

$$U(t_1, t_2) = \sum_{t_1 \leq \tau < t_2} \mu(u(\tau)) \quad (2)$$

for some strictly concave¹ utility function $\mu : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{> 0}$. The concavity constraint leads to a diminishing utility if the application consumes more energy, i.e., additional energy is more important in case of a low energy level than in case of an already high energy level. With μ one can also model that there is almost no additional utility beyond a certain consumed energy.

3.2 Optimality Criteria of the Controller

Based on the above system model, we can now define reasonable optimality criteria for the controller. Given the stored energy at time $t = 0$, the required stored energy at the end of the time-horizon T , and the harvesting function $p(t)$ for all $0 \leq t < T$, we are interested in an optimal use function $u^*(t)$ that satisfies the following conditions:

- **C1:** The system never enters a failure state, i.e., we have $b^*(t) + p(t) - u^*(t) \geq 0$ for all $0 \leq t < T$, where $b^*(t)$ denotes the stored energy with harvesting function $p(t)$ and use function $u^*(t)$.
- **C2:** There is no feasible use function $u(t)$ with a larger minimal energy, i.e., $\min_{0 \leq t < T} \{u(t)\} \leq \min_{0 \leq t < T} \{u^*(t)\}$ for all feasible $u(t)$. This condition is important as we aim to maximize the lowest service level.

There may be multiple use functions that satisfy the conditions defined above. In this case, we want to choose a use function $u^*(t)$ that maximizes a secondary objective function:

- **C3:** Amongst all use functions $u(t)$ that satisfy **C1** and **C2**, we choose one that maximizes the total utility, i.e., maximizes $U(0, T)$.

It is by no means obvious how $u^*(t)$ can be computed. At every unit-step, the controller needs to decide whether to use more energy, which would increase the use function for that unit time interval. On the other hand, it removes available energy from the battery which may cause battery depletion or a lower use function at a future time interval. Furthermore, it is not clear if there exists an efficient online algorithm to determine $u^*(t)$, if it indeed exists.

¹A strictly concave function μ satisfies $\mu(\alpha x + (1 - \alpha)y) > \alpha\mu(x) + (1 - \alpha)\mu(y)$ for any $0 \leq \alpha \leq 1$ and $x, y \in \mathbb{R}_{\geq 0}$.

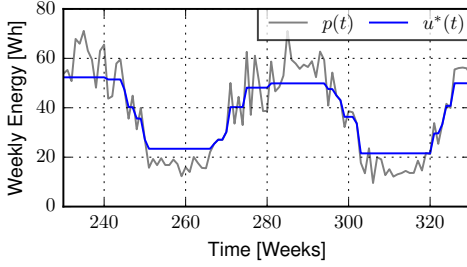


Fig. 1. Optimal use function $u^*(t)$ for a given harvesting function $p(t)$.

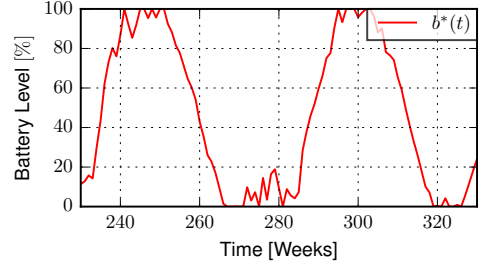


Fig. 2. Battery Level $b^*(t)$ corresponding to the optimal use function shown in Fig.1.

4 OPTIMAL CONTROL

Here we will study the problem of computing $u^*(t)$ when $p(t)$ is known in advance. In other words, we consider a clairvoyant algorithm which has perfect knowledge of the harvesting energy to be expected on the interval $[0, T)$. Though not practical, this simplified setting allows us to identify the main theoretical ideas that characterize this problem. In Sec. 5, we will consider the problem where only an estimate of the harvesting energy is known.

4.1 Necessary Conditions for Optimality

The main result of this section can be formulated by the following theorem.

THEOREM 4.1. *Given a use function $u^*(t)$ that satisfies the following two conditions: (1) The harvesting system never enters a failure or waste state (see condition C1), and (2) the following relations hold:*

$$u^*(s-1) < u^*(s) \Rightarrow b^*(s) = 0 \quad ; \quad u^*(t-1) > u^*(t) \Rightarrow b^*(t) = B \quad (3)$$

Then $u^(t)$ maximizes the minimal used energy (see condition C2), it maximizes the total utility $U(0, T)$ (see condition C3), and it is unique.*

In other words, if we are able to construct a use function that satisfies (3), never leads to a failure state, and never wastes energy, then the minimal used energy as well as the total utility is maximized. We refer to this use function $u^*(t)$ as the optimal use function. This result defines a necessary and sufficient condition on u^* to satisfy criteria C1, C2 and C3. We now show the sequence of two Lemmata that lead to the above fundamental result. All proofs are given in [1].

LEMMA 4.2. *Any optimal use function $u^*(t)$ satisfies (4) and (5).*

$$\forall s \leq \tau \leq t : 0 < b^*(\tau) < B \Rightarrow \forall s-1 \leq \tau \leq t : u^*(\tau) = u^*(t) \quad (4)$$

$$u^*(s-1) < u^*(s) \Rightarrow b^*(s) = 0 \quad ; \quad u^*(t-1) > u^*(t) \Rightarrow b^*(t) = B \quad (5)$$

This Lemma states that the used energy is constant over a given interval as long as the battery is neither full nor empty. In addition, if an optimal use function grows (shrinks), then the corresponding battery level must be empty (full). Interestingly the above condition applies to all possible harvesting functions $p(t)$. Next we show that $u^*(t)$ is unique.

LEMMA 4.3. *If there exists a use function $u^*(t)$ that satisfies the necessary optimality condition (5) of Lemma 4.2 and that does not lead to a failure or waste state, then it is unique.*

The above result is quite surprising but an example may help to interpret it. To this end, we use a data trace of harvested solar energy with a time granularity of one week. Fig. 1 shows 100 weeks of an optimal use function $u^*(t)$ and Fig. 2 shows the corresponding battery state $b^*(t)$. As is evident,

when the battery is neither full or empty, $u^*(t)$ is constant. Further, if $u^*(t)$ decreases, then the battery is full and if $u^*(t)$ increases, then the battery is empty. This seems highly counter intuitive as one would expect that in case of a full battery it is reasonable to increase the energy consumption, and in case of an empty battery one would decrease the energy consumption. However, considering that the algorithm seeks to fully leverage the available energy (including the energy that is stored in the battery), an empty battery marks the end of a period of deficit, i.e., the period during which the battery is being fully discharged (see Fig. 2). Therefore, it makes sense to increase the energy use upon entering such a period as now there is more harvesting energy available. Conversely, a full battery marks the end of a surplus period and therefore, the energy use needs to be reduced.

We now focus on the algorithm for determining $u^*(t)$ for any given $p(t)$.

ALGORITHM 1: Iterative algorithm for optimal control.

Input: $b(0); b(T); B; \epsilon; T; (\forall 0 \leq t < T : p(t));$

Output: $(\forall 0 \leq t < T : u(t));$

for all $0 < t \leq T$ **do**

$\sigma_l(t) \leftarrow \sum_{\tau=1}^t p(\tau - 1); \sigma_u(t) \leftarrow \sigma_l(t) + B;$

end for

$f(0) \leftarrow B - b(0); f(T) \leftarrow \sigma_u(T) - b(T);$

for all $0 < t < T$ **do**

$f(t) \leftarrow (\sigma_u(t) + \sigma_l(t))/2;$

end for

repeat

$f' \leftarrow f;$

for $t = 1$ **to** $T - 1$ **do**

$f(t) \leftarrow (f(t - 1) + f(t + 1))/2;$

$f(t) \leftarrow \text{Max}(\text{Min}(f(t), \sigma_u(t)), \sigma_l(t));$

end for

until $\text{Max}(|f' - f|) < \epsilon$

for all $0 \leq t < T$ **do**

$u(t) \leftarrow f(t + 1) - f(t);$

end for

4.2 Algorithm

In this section we show that there is an unexpected relation between the optimal power management problem for harvesting systems and the shortest Euclidean path in simple polygons problem in computational geometry. The later has been widely studied and efficient algorithms are available. In [14] an $O(n)$ algorithm is described where n denotes the number of polygon vertices. It can be expected that the optimal control problem can also be solved efficiently. Similar to [11], we first establish a connection between the two seemingly unrelated problems. To this end, we define a (*harvesting*) *polygon* by all points (t, σ) with integer time $t \in [0, T]$ and $\sigma_l(t) \leq \sigma \leq \sigma_u(t)$ where

$$\sigma_l(t) = \sum_{\tau=1}^t p(\tau - 1) \quad \forall 1 \leq t \leq T \quad ; \quad \sigma_u(t) = \sigma_l(t) + B \quad (6)$$

and $\sigma_l(0) = 0$. In other words, $\sigma_l(t)$ denotes the harvested energy in the time interval $[0, t]$. Therefore, the polygon contains all points (t, σ) where σ is between the energy harvested in $[0, t]$ and this value shifted up by the battery capacity B .

Now, a *feasible path* $f(t)$ in the (harvesting) polygon starts at $f(0) = B - b(0)$, ends at $f(T) = \sigma_u(T) - b(T)$, is monotonically increasing, and never leaves the polygon, i.e., $\sigma_l(t) \leq f(t) \leq \sigma_u(t)$ for all $t \in [0, T]$. Furthermore, as we consider a discrete time model, f is assumed to be piecewise-linear with discontinuities only at integral time steps.

As has been shown in [11], there is an one-to-one correspondence between any feasible path $f(t)$ in the harvesting polygon and a feasible use function $u(t)$ with no waste, where

$$u(t) = f(t+1) - f(t) \quad \forall 0 \leq t < T. \quad (7)$$

Moreover, the shortest feasible path $f^*(t)$ in the harvesting polygon yields the optimal use function $u^*(t)$ that maximizes the throughput in the network [11].

The runtime optimal algorithm from [14] is difficult to implement due to the use of advanced data structures and complex operations. In *Sec. 5.4* we will propose a strategy to determine explicit solutions to the optimal power management problem that can be easily stored and used on resource-constrained platforms. In *Algorithm 1* we describe a very efficient iterative algorithm that can be used on resource-rich platforms or server-controlled devices. This algorithm has guaranteed convergence in terms of precision vs. runtime and is based on the ideas presented in [20, 21]. *Algorithm 1* uses the parameter ϵ as a stopping criterion, i.e., the maximal change in any path coordinate from one iteration to the next.

The first lines in *Algorithm 1* define the harvesting polygon as described in (6), as well as the initial and final points of a feasible path. Then, the initial path is chosen to be in the middle between the upper and lower boundaries of the harvesting polygon. The main iteration attempts to make a path as straight as possible within the polygon, i.e., it puts a point in the middle between its neighbors unless it touches the upper or lower boundary. Experimental results are described in *Sec. 7*.

5 FINITE HORIZON CONTROL

The previous section provided conditions and an algorithm for optimal clairvoyant control: If the battery's initial state $b(0)$, its final state $b(T)$, and the harvested energy $p(t)$ for all time intervals in $[0, T)$ are given, then the optimal and unique use function $u^*(t)$ can be determined.

In this section we revert to a realistic setting where harvesting energy in the future is not known. Instead, we can estimate future harvesting energy. We present Finite Horizon Control (FHC) for power management that attempts to maximize utility based on energy estimates. We also formulate bounds on the achieved optimality of FHC. A Look-up Table (LUT) based scheme is also proposed in this section, for efficient deployment of FHC on a resource-limited hardware platform.

In the initial part of this section, we suppose that an estimate of harvesting energy is given for all $t \geq 0$, denoted by $\tilde{p}(t)$. We will discuss later in Section 5.5 how this estimate can be obtained. In addition, FHC/LUT will make use of the fact that in usual solar harvesting scenarios the harvested energy shows some pattern that repeats yearly, e.g., the yearly repeating summer/winter pattern.

In particular, we may assume that the unit time interval is one week. Therefore, it is assumed that $\tilde{p}(t)$ is periodic with period $P = T = 52$, the number of weeks in one year. Due to this periodicity we find $\tilde{p}(t) = \tilde{p}(t + k \times 52)$ for all $k \in \mathbb{N}$. We will make use of this fact by setting the finite control horizon in FHC scheme to one year. The derivation of FHC and its performance bounds involves four steps that will be described in the following subsections:

- (1) Derive an optimal periodic control scheme by assuming $p(t) = \tilde{p}(t)$ for all time instances t .
- (2) Use the scheme derived in step (1) to construct an adaptive finite horizon scheme.
- (3) Provide theoretical bounds on the performance of FHC scheme constructed in step (2).
- (4) Parameterize the above FHC scheme to improve its performance.

5.1 Optimal Periodic Control

As stated earlier, we suppose a periodic estimated harvesting energy $\tilde{p}(t)$. By assuming that the state of battery at the end of period $\tilde{b}(P)$ equals that at the beginning $\tilde{b}(0)$, we can conclude that all essential quantities are periodic, i.e., the optimal use function, the energy harvesting function

and the battery state: $\tilde{u}^*(t) = \tilde{u}^*(t + k \cdot P)$, $\tilde{p}(t) = \tilde{p}(t + k \cdot P)$, and $\tilde{b}^*(t) = \tilde{b}^*(t + k \cdot P)$ for all $k \in \mathbb{N}$. Furthermore, the time horizon is assumed to be equal to the period of harvesting energy, i.e., $T = P$.

ALGORITHM 2: Iterative algorithm for periodic optimal control.

Input: $P; B; \epsilon; (\forall 0 \leq t < P : \tilde{p}(t))$;
Output: $(\forall 0 \leq t < P : \tilde{u}(t)); (\forall 0 \leq t < P : \tilde{b}(t))$;
for all $0 \leq t \leq P$ **do**
 $\sigma_l(t) \leftarrow \sum_{\tau=1}^t \tilde{p}(\tau - 1)$; $\sigma_u(t) \leftarrow \sigma_l(t) + B$;
end for
 $f(0) \leftarrow B/2$; $f(P) \leftarrow \sigma_l(P) + B/2$
for all $0 < t < P$ **do**
 $f(t) \leftarrow (\sigma_u(t) + \sigma_l(t))/2$;
end for
repeat
 $f' \leftarrow f$;
 $f(0) \leftarrow (f(P - 1) - \sigma_l(P) + f(1))/2$;
 $f(0) \leftarrow \text{Max}(\text{Min}(f(0), B), 0)$;
 for $t = 1$ **to** $P - 1$ **do**
 $f(t) \leftarrow (f(t - 1) + f(t + 1))/2$;
 $f(t) \leftarrow \text{Max}(\text{Min}(f(t), \sigma_u(t)), \sigma_l(t))$;
 end for
 $f(P) \leftarrow f(0) + \sigma_l(P)$;
until $\text{Max}(|f' - f|) < \epsilon$
for all $0 \leq t < P$ **do**
 $\tilde{u}(t) \leftarrow f(t + 1) - f(t)$; $\tilde{b}(t) \leftarrow \sigma_u(t) - f(t)$;
end for

In this setting, the optimized use function can be determined using a periodic variant of *Algorithm 1* where we assume $\tilde{b}(0) = \tilde{b}(P)$ and we iterate the path update in a circular way with $f(t + P) = f(t) + \sigma_l(P)$, see *Algorithm 2*. In summary, *Algorithm 2* computes an optimal periodic battery state \tilde{b}^* and use function \tilde{u}^* given a periodic energy harvesting function \tilde{p} , its period P and the battery capacity B . Based on this result, we will next define the FHC scheme.

ALGORITHM 3: Finite horizon scheme for optimal control

Input: $t; b; T; \tilde{b}^*(t + T); (\forall 0 \leq \tau < T : \tilde{p}(t + \tau))$;
Output: u ;
 $b(0) \leftarrow b$; $b(T) \leftarrow \tilde{b}^*(t + T)$;
 for all $0 \leq \tau < T$ **do**
 $p(\tau) \leftarrow \tilde{p}(t + \tau)$;
 end for
 execute *Algorithm 1* with the above inputs and the result $(\forall 0 \leq t < T : u(t))$;
 $u \leftarrow u(0)$;

5.2 Finite Horizon Scheme

Following the ideas of receding horizon control, e.g., [17], we replace the unknown harvesting energy function with its estimate and optimize the future use function with the current battery state. For example, let us suppose that we are currently in week $t = 0$ and $b(0) = b$. In FHC, we will compute $\tilde{u}(t)$ using *Algorithm 1*. However, we will only execute the trace of $\tilde{u}(t)$ for the first time step. At $t = 1$, we will recompute $\tilde{u}(t)$ with the actually measured battery level at this point in time. This process is repeated indefinitely.

Algorithm 3 implements this strategy, i.e., at each time step t , it uses the current battery state b and provides the optimized use function value u . The algorithm requires the following static input data: the estimated energy function $\tilde{p}(t)$, the target battery value $\tilde{b}^*(t + T)$ at the end of the finite horizon from *Algorithm 2*, and the time horizon T . Note that \tilde{p} and \tilde{b} are periodic in P .

The performance of the finite horizon scheme crucially depends on $\tilde{p}(t)$. If the actual harvested energy is overestimated, then it is possible that *Algorithm 3* computes an aggressive use function which may lead to a failure state, i.e., the battery drains out. To avoid this a conservative estimate is required, where $p(t) \geq \tilde{p}(t) \forall t$, see also Section 5.5.

5.3 Guarantees

FHC scheme as described in the previous section replaced the unrealistic optimal clairvoyant control algorithm by an adaptive runtime algorithm. Following important guarantees that can be provided:

THEOREM 5.1. *Given an initial and final battery state $b(0)$ and $b(T)$, respectively, the battery capacity B , and the harvested energy function $p(t)$ for all $0 \leq t < T$. Then, according to Theorem 4.1, all values $u^*(t)$ for $0 \leq t < T$ of the optimal use function $u^*(t)$ are monotonically increasing with increasing $b(0)$, B , $p(t)$ for $0 \leq t < T$, and with decreasing $b(T)$.*

Based on the above property, we can now prove the main guarantee for the FHC scheme.

THEOREM 5.2. *Given a periodic estimated energy harvesting function $\tilde{p}(t)$ with period P , a battery with capacity B , the corresponding optimal periodic use function $\tilde{u}^*(t)$ and the corresponding periodic battery state $\tilde{b}^*(t)$. Further given an energy harvesting system with the same capacity B but with an initial battery $b(0) \geq \tilde{b}(0)$ and an actual energy harvesting function $p(t) \geq \tilde{p}(t)$ for all $t \geq 0$. Then the following holds: If we execute the finite horizon control according to *Algorithm 3* for each time step t , then the resulting use function satisfies $u(t) \geq \tilde{u}^*(t)$ for all $t \geq 0$.*

In other words, the use function when executing the finite horizon control algorithm is never smaller than the optimal periodic one as computed using *Algorithm 2*. Therefore, the minimal value of the use function as well as its utility are lower bounds for the online finite horizon control. Under the assumptions of *Theorem 5.2*, we can provide the following guarantees: The system will never enter a failure state, and the minimal use function and the utility are no smaller than those obtained with *Algorithm 2*. For a detailed explanation and proofs of the stated theorems, please refer to [1]

5.4 Efficient Implementation

A natural concern with the proposed Finite Horizon Control (FHC) scheme has high runtime overhead. In this section we mitigate this by using a simple algorithm based on a Look-up Table (LUT). Under the assumption of a static estimate of the harvested energy, we can use the finite horizon scheme to identify the current $u(t)$ as an explicit function F of the current time step t (with respect to the period) and the current battery state $b(t)$. With this, we can pre-compute the function F at design time, and obtain the use function from F with the current battery state.

For a given time step, $F(t, \cdot)$ is a function of only the battery state. We approximate $F(t, \cdot)$ by a piecewise linear function \bar{F}_t , which can be computed offline for a finite number of battery levels. In our experiments, we discretize the battery by steps of 1%. For each battery state, we explicitly use *Algorithm 1* to compute the optimal clairvoyant use function. These linear functions can then be represented concisely in a LUT. As a result, the online algorithm is reduced to very few arithmetic operations. Evaluations as shown in *Sec. 7* validate that the performance of LUT based algorithm is comparable to that of the original FHC scheme with a repeated execution of *Algorithm 1* in every time step.

5.5 Algorithm Parameterization and Adaptation

The control scheme introduced requires an estimation of the harvesting energy for each unit time interval to compute the optimal use function. With long-term operation in mind, we opted to use an astronomical model [13] as the basis for the energy estimator. We first briefly review this model, and discuss parameter selection for the proposed control scheme. Afterwards, other parameterization details of the finite horizon scheme are discussed shortly.

To estimate the total solar energy available at a particular geographical location without the need for historical harvesting data, we leverage an astronomical model adapted for power subsystem planning of solar energy harvesting embedded systems [8]. It can accurately approximate the average harvesting energy for a given geographical location and time of year [8][7][28].

Design-Time Energy Estimation. The total estimated solar energy incident on a flat, tilted surface depends on the following known quantities: latitude L of the deployment location, panel orientation and inclination angles ϕ_p , and θ_p respectively, and the solar panel surface area A_{pv} . It further depends on the unknown environmental parameter Ω , which incorporates weather and shading effects. To obtain reasonable values for Ω , we follow the guidelines in [8] and profile the first year for each of the datasets used for evaluation in Sec. 7.

Runtime Energy Estimation. The design-time energy estimation model relies on appropriate parameterization to be representative of the true conditions. It is assumed that A_{pv} , L , ϕ_p , and θ_p can be defined with sufficient accuracy, but Ω can only be approximated. This means that the design-time model may deviate from the true average energy input if the weather conditions and/or shading effects at the deployment location are poorly estimated. Therefore, in order to adapt the design-time model to reflect true conditions at runtime, we follow the approach in [7] and dynamically scale the design-time model by a factor α . The scaling factor α is computed as the ratio of the sums of true and estimated energy observed over a history window of length W weeks. To achieve duty-cycle stability, $W = 52$ was chosen.

Horizon of Control. With the objective of achieving long-term minimum performance guarantees, it is reasonable to consider the annual, rather than diurnal solar cycle for the control horizon. This means that at any time instance τ , the algorithm from Sec. 5.2 is given $\tilde{p}(t) \forall t \in [\tau, \tau + T)$ as estimation input, where $T = 1$ year.

Frequency of Control. For determining frequency of control, it is generally desirable to keep the update intervals long, in order to reduce the algorithm's computation overhead and keep a stable service level. In this work, frequency of control was chosen equal to unit time (1 week).

6 STOCHASTIC POWER MANAGEMENT

The Finite Horizon Control (FHC) scheme in section 5 assumes that a lower bound on the harvested energy is known. However, such a conservative lower bound may be difficult to determine or too conservative.

To address this limitation, we propose Stochastic Power Management (SPM) which utilizes a statistical model of harvested energy such that there is a *stochastic guarantee* of not encountering a failure state during operation. The stochastic guarantee we are interested in is the long term probability of encountering a failure state within the horizon of interest. Its lower bound λ is a parameterizable constant. Without loss of generality, we assume that the horizon of interest is one calendar year and unit time is one week, i.e., $T = 52$.

SPM uses historical data of harvested energy to construct its statistical model. Detailed statistical analysis is then performed to derive the use function while conforming to the stochastic guarantee. The statistical analysis performed in this section has the following sub-components: 1) Given historical data, determining a statistical model of the harvested energy, 2) determining a statistical

model of battery charge, 3) determining the probability of failure, 4) determining steady state statistical models, and 5) determining safe battery charges which can be used to dynamically exploit energy surplus. We will now explain these sub-components.

6.1 Statistical model of harvested energy

Given historical data, we use maximum likelihood estimation [30] to fit the data to a statistical distribution. It is assumed that data is Normally Distributed. We empirically observed that normal distribution provides a close match with the historical data set. However, other fitting methods/distributions may be employed; and the problem of fitting statistical models is orthogonal to the SPM scheme proposed in this section. The fitted distribution is discretized and normalized to determine a discrete statistical model: $h_t(x)$ denotes the probability mass function (p.m.f) of energy harvested in time interval $[t, t + 1)$.

6.2 Statistical model of battery charge

By using $h_t(x)$ and use function $u(t)$, we can determine a statistical model of the battery charge. Suppose that the battery has a fixed charge $b(t)$ at time t . Ignoring constraints on minimum and maximum battery level, the battery charge p.m.f at time $t + 1$ can be computed using following equation:

$$l_{t+1}^*(x) = h_t(x) * \delta(x - b(t)) * \delta(x + u(t))$$

where $*$ is the convolution operator and δ is the dirac delta function. Note that $l_{t+1}^*(x)$ does not take into account the minimum and maximum battery level constraints. We now construct the truncated battery charge p.m.f to take these constraints into account:

$$l_t(x) = \begin{cases} 0 & \text{if } x < 0 \text{ or } x > B \\ \sum_{x \leq 0} l_t^*(x) & \text{if } x = 0 \\ l_t^*(x) & \text{if } 0 < x < B \\ \sum_{x \geq B} l_t^*(x) & \text{if } x = B \end{cases} \quad (8)$$

In general, for computing the battery charge p.m.f at time t , we use the following equation to first find the non-truncated battery charge p.m.f.

$$l_t^*(x) = h_{t-1}(x) * l_{t-1}(x) * \delta(x + u(t - 1)) \quad (9)$$

l_t^* is then truncated using (8) to get l_t .

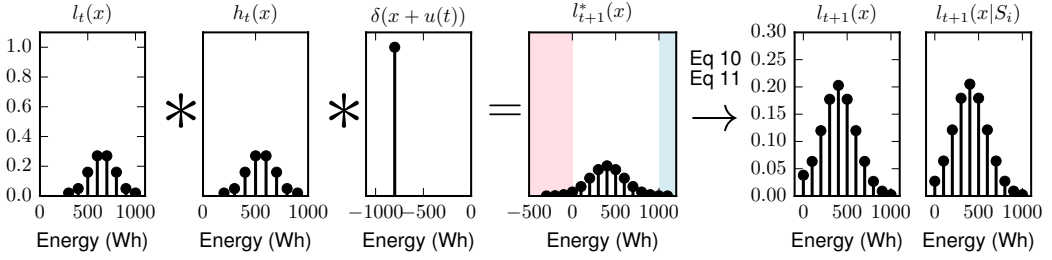
Another important statistical model is the p.m.f of battery charge assuming no failure in the previous interval. This function will be important for determining the probability of failure in the next subsection. We call this the success-conditional battery charge p.m.f and it is computed using the following equation:

$$l_t(x|S_{t-1}) = \begin{cases} 0 & \text{if } x < 0 \text{ or } x > B \\ \frac{l_t^*(x)}{P(S_{t-1})} & \text{if } 0 \leq x < B \\ \frac{\sum_{x > B} l_t^*(x)}{P(S_{t-1})} & \text{if } x = B \end{cases} \quad (10)$$

$P(S_t)$ is the probability of success (no failure) in interval $[t, t + 1)$ and can be computed as:

$$P(S_t) = 1 - \sum_{x < 0} l_{t+1}^*(x) \quad (11)$$

Computation of the non-truncated, truncated and success conditional battery charge p.m.f.s is illustrated using an example in Figure 3. The p.m.f.s of the battery charge and harvested energy are shown in the first two plots of Figure 3. It is assumed that energy consumption during interval

Fig. 3. computation of battery p.m.f at time $t + 2$

$[t, t + 1)$ is 800 Wh and battery capacity is 1000 Wh. First, the non-truncated battery charge p.m.f is determined by computing $l_t(x) * h_t(x) * \delta(x + 800)$. The resulting battery charge p.m.f has a non-zero probability at battery charge > 1000 (B) and at battery charge < 0 . Therefore, the battery is truncated using (8). The success conditional battery charge p.m.f is determined using (10).

6.3 Determining probability of failure within one year

In the previous subsection, we identified that the probability of failure within a given interval (one week) can be determined by means of (11). We will now explain how to determine the probability of failure within the horizon of interest. If failures in all intervals were independent, then we could compute the probability of failure as $\lambda^* = 1 - \prod_{i \in \{0, 1, \dots, 51\}} P(S_i)$. However, the failures are not independent. For instance, if a failure occurs in interval $[t, t + 1)$, then $b(t + 1) = 0$. Due to zero battery charge, the likelihood of failure in interval $[t + 1, t + 2)$ increases. To take this chained interdependence into account, we define $P(S_t|S)$ as the probability of success in interval $[t, t + 1)$ given that no failure has occurred in any previous time interval. Based on this definition, we can now determine the probability of failure using the following equation:

$$\lambda^* = 1 - \prod_{t \in \{0, 1, \dots, 51\}} P(S_t|S) \quad (12)$$

To compute $P(S_t|S)$, we make all battery charge p.m.f.s success-conditional. Specifically, we use following equation (13) to compute l_{t+1}^* .

$$l_{t+1}^*(x) = h_t(x) * l_t(x|S_{t-1}) * \delta(x + u(t)) \quad (13)$$

Note the use of the success-conditional battery charge p.m.f $l_t(x|S_{t-1})$. Subsequently, $l_{t+1}(x|S_t)$ is computed using $l_{t+1}^*(x)$ and (10). After making the battery charge p.m.f success conditional, $P(S_i|S)$ is computed as: $P(S_i|S) = 1 - \sum_{x < 0} l_{i+1}^*(x)$. Finally λ^* is computed using (12).

6.4 Steady State Probability

Due to the annual solar cycle, the statistical model for harvesting energy is periodic i.e., $h_{t+k \cdot T}(x) = h_t(x) \quad \forall k \in \mathbb{N}$. If we now assume that the use function is also periodic with period T i.e., $u(t + k \cdot T) = u(t) \quad \forall k \in \mathbb{N}$, we can construct a notion of steady-state battery charge p.m.f. Specifically, the battery charge p.m.f.s for all time intervals will evolve across years until they reach a steady state where $l_{t+k \cdot 52}(x) = l_{t+(k+1) \cdot 52}(x) \quad \forall x \in \mathbb{Z}, k \in \mathbb{N}, t \in \{0, 1, \dots, 51\}$. We use $l_\tau^\infty(x)$ and $l_\tau^\infty(x|S)$ to denote the steady state non-truncated and steady-state success-conditional battery charge p.m.f.s respectively, for week $\tau = t \bmod 52$. The steady state annual failure probability can be computed using the following equation:

$$\lambda^\infty = 1 - \prod_{\tau \in \{0, 1, \dots, 51\}} \left(1 - \sum_{x < 0} l_\tau^\infty(x)\right) \quad (14)$$

At this point, we can compute the long term probability of failure given a periodic use function, historical data of harvested energy and the battery capacity B . For long term operation of the system, it is required that $\lambda^\infty \leq \lambda$. From this point on, we will assume that all statistical models and use functions are periodic with period $T = 52$. τ will denote the week of the current year, i.e., $\tau = t \bmod 52$.

6.5 Safe Charges

An important aspect of the power management scheme is the capability to dynamically use/exploit surplus energy. To enable this, we will now define *safe charges*. Informally, a safe charge vector C contains battery levels for the 52 weeks, such that if battery charge in week τ is higher than C_τ , the additional battery charge can be dissipated in addition to the use function ($u(t \bmod 52)$) without increasing the steady state failure probability beyond λ .

We can account for the additional energy consumption by truncating the success conditional battery charge p.m.f at the start of week τ , at C_τ instead of the battery capacity B . This is because any battery charge beyond C_τ is dissipated in addition to $u(t \bmod 52)$. Specifically, we can use the following equation to compute success conditional battery charge p.m.f:

$$l_t(x|S_{t-1}, C_\tau) = \begin{cases} 0 & \text{if } x < 0 \text{ or } x > C_\tau \\ \frac{l_t^*(x)}{P(S_{t-1})} & \text{if } 0 \leq x < C_\tau \\ \frac{\sum_{x>B} l_t^*(x)}{P(S_{t-1})} & \text{if } x = C_\tau \end{cases} \quad (15)$$

where $\tau = t \bmod 52$. Steady state probability of failure can be computed as explained in subsection 6.3 and subsection 6.4 by changing (13) in the following manner:

$$l_{t+1}^*(x) = h_t(x) * l_t(x|S_{t-1}, C_\tau) * \delta(x + u(t)) \quad (16)$$

The safe charge vector C is not unique; and multiple safe charge vectors could satisfy the failure constraint. In general, elements of this vector should be minimized so that higher energy consumption can be supported without violating the failure constraint.

6.6 Stochastic Power Management

Based on the defined stochastic models, we can now explain the SPM scheme. SPM attempts to maximize the minimum value (v) of use function ($u(t)$) such that the steady state probability of encountering a failure state is $\leq \lambda$. As a secondary objective, SPM also tries to dynamically exploit periods of energy surplus without adversely affecting the failure probability.

ALGORITHM 4: SPM offline phase

Input: $h_\tau(x) \forall \tau \in \{0, 1, \dots, 51\}; \lambda; B;$

Output: $v; C_\tau \forall \tau \in \{0, 1, \dots, 51\};$

$C_\tau = B \quad \forall \tau \in \{0, 1, \dots, 51\}$

Assume $u(\tau) = v \forall \tau \in \{0, 1, \dots, 51\}$

Use binary search to find value of v such that $\lambda^\infty \leq \lambda$, where λ^∞ is calculated using (14)

for all $i \in \{0, 1, \dots, 51\}$ **do**

Use binary search to find value of C_i such that $\lambda^\infty \leq \lambda$, where λ^∞ is calculated using (14)

end for

The first goal of SPM is to maximize the minimum level of service. The minimum service level is computed offline and its evaluation steps are detailed in Algorithm 4. We assume that the use function $u(\tau) = v \forall \tau$. We use binary search to find the largest feasible value of v such that $\lambda^\infty \leq \lambda$. For this computation all elements of safe charge vector C are set of battery capacity B . The

ALGORITHM 5: SPM online phase**Input:** $b(t)$; v ; w ; C_τ , $S(t) \in \{\text{Nominal}, \text{Failure}\}$,**Output:** $u(t)$; $S(t+1)$; $\tau = t \bmod T$ surplus = $b(t) - C_\tau$ $u(t) = 0$ **if** surplus ≥ 0 **then** $u(t) = v + \text{surplus}/w$ $S(t+1) = \text{Nominal}$ **else if** $S(t) = \text{Nominal}$ **then** $u(t) = v$ **end if**

computed value of v is then used to compute safe charges. We use an iterative procedure where we first search for minimum safe charge at the start of week 0 (C_0) such that $\lambda^\infty \leq \lambda$. Next, this value of safe charge is fixed and the safe charge for remaining weeks is computed. The choice of an iterative procedure to compute safe charges is arbitrary, and more advanced search techniques could be employed.

The runtime algorithm is presented in Algorithm 5. This algorithm takes as an input the current battery level $b(t)$, v , a spread parameter w which specifies how energy surplus is utilized, and $S(t)$ which defines the system state when the online algorithm is called. *Failure* state indicates that due to battery depletion, the use function was smaller than v recently. *Nominal* state indicates that a failure has either never occurred, or has been recovered from.

Whenever $b(t)$ is greater than C_τ , we have energy surplus equal to $b(t) - C_\tau$. This surplus can be used either entirely in the current interval; or partially so that there are more chances of having surplus in future intervals. This *spread* of excess energy is determined by w . Note that in case of non-negative excess, the use function value is agnostic to the system state, and the system state in the next interval is set to *Nominal*; which corresponds to failure recovery if $S(t) = \text{Failure}$. If $b(t) < C_\tau$, SPM checks the system state. If it is *Nominal*, we set $u(t) = v$, otherwise, $u(t) = 0$. This is done to avoid consecutive failures. Once a failure is encountered, SPM waits for the battery to reach a safe charge before consuming any power. During runtime, if battery is fully depleted, the system state is set to *Failure*.

7 EXPERIMENTAL EVALUATION

In this section we compare the proposed algorithms with state-of-the-art power management algorithms.

Input Data. We leverage the publicly available² National Solar Radiation Database (NSRD), from where we acquire trace data for 8 different geographical locations. Each trace spans a time period of 12 years. Additionally, we use two years of energy traces collected at two different locations with our own long-term Wireless Sensor Network (WSN) deployment [5]. Details for the data sets and harvesting node parameters are given in *Table 1*.

Baseline Algorithms. We compare our approaches against four state-of-the-art implementations of energy-predictive and battery-reactive power management approaches. Specifically, we compare against Long-Term Dynamic Power Management (LT-DPM) [7] which is a system dimensioning and power management scheme geared towards providing uninterrupted operation with low service-level variability. We further implement the predictive duty-cycling scheme from [16] with two different energy predictors, Exponentially Weighted Moving Average (EWMA) [16] and Weather

²http://rredc.nrel.gov/solar/old_data/nsrdb/

Name	Time Period	Lat [°]	Long [°]	P_{max} [W]	A_{pv}	Ω
TX2	1/61 – 12/72	28.05	-97.39	0.4135	10	0.3431
TX1	1/61 – 12/72	31.77	-106.48	0.3915	15	0.2242
CA	1/98 – 12/09	34.05	-117.95	0.353	10	0.3282
MD	1/61 – 12/72	39.29	-76.61	0.3915	15	0.3351
MI	1/98 – 12/09	42.05	-86.05	0.286	15	0.4729
OR	1/61 – 12/72	45.52	-122.67	0.27	15	0.3955
ON	1/98 – 12/09	48.05	-87.65	0.248	20	0.4855
AK	1/61 – 12/72	61.21	-149.90	0.128	20	0.3448
DH	2/12 – 3/14	46.12	7.8212	6	172.5	0.6
GG	3/12 – 3/14	46.09	7.8133	6	172.5	0.6

Table 1. Dataset and harvesting node parameters

Conditioned Moving Average (WCMA) [24]. The duty cycling scheme in [16] attempts to continually correct mis-predictions by increasing or decreasing the energy usage in future slots. EWMA and WCMA are designed to predict energy on a horizon of a few minutes to hours. Therefore, it is expected that they will not be able to make full use of the allocated battery capacity. We also evaluate a battery reactive scheme Energy Neutral Operation with Maximal energy efficiency (ENO-Max) [29]. ENO-Max uses well-established control theory to determine the utilization that results in maximum energy efficiency, while ensuring Energy Neutral Operation (ENO). This algorithm adjusts the utilization such that a pre-defined battery threshold may be maintained. While it is very effective in achieving its maximization objective, the ENO objective causes the algorithm to aggressively adjust the utilization when target battery level cannot be maintained, which results in high variability in the energy consumption.

For each of the baseline implementations we use the author’s recommended parameters, i.e., $K = 3$, $D = 4$, $\alpha = 0.3$ for WCMA [24], and $\alpha = 0.5$ for EWMA [16]. For ENO-MAX [29], we use $\alpha = 1/24$, $\beta = 0.25$, and $B_{target} = 65\%$. Finally, due to the hourly values given by the National Solar Radiation Database, we use $N_w = 24$ instead of 48 daily update slots for EWMA, WCMA, and ENO-MAX. This results in a slight penalty in prediction accuracy. For LT-DPM [7], we assume daily control updates, i.e., $N_w = 1$, while our approach operates with weekly control updates (see Sec. 5.5).

Harvesting Node Model. To get solar panel size, we apply the system dimensioning approach proposed in [8] with a target duty cycle of 40% and nominal battery capacity of 100Wh. The solar panel sizes for different data sets are given in Table 1. It is assumed that battery has a charging efficiency of 0.9. All other efficiency parameters are ignored. Battery degradation is not considered and it is assumed that the battery would outlast other system components.

Performance Metrics. Since we aim at maximizing the minimum service level, we report the *minimum daily energy consumption* (v) for each of the approaches. Note that outages due to a depleted battery cause the minimum to be 0. We also report total system utility (U) and use (2) with $\mu(u(t)) = \sqrt{u(t)}$.

7.1 Preliminary Evaluation

In this section, we compare the performance of three proposed schemes (Clairvoyant (CV), Finite Horizon Control (FHC), and Stochastic Power Management (SPM)) for years 6 and 7 of the MI data set. Fig. 4 and Fig. 5 compare the use functions and battery levels of FHC with CV. We notice the FHC scheme encounters failures (battery fully drains) at several points in time. The reason behind

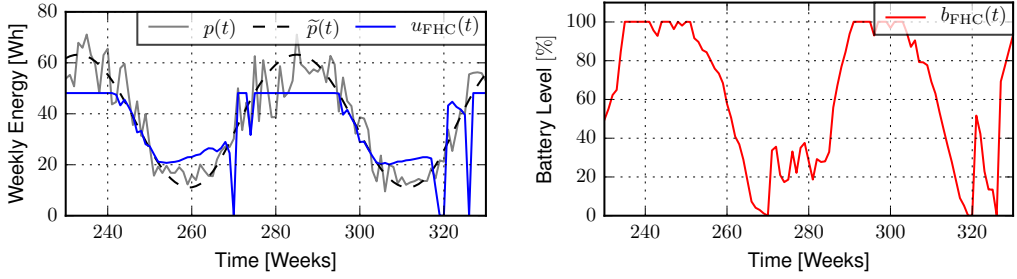


Fig. 4. Finite Horizon Scheme use function $u_{\text{FHC}}(t)$ for Fig. 5. Battery Level $b_{\text{FHC}}(t)$ corresponding to use function a given harvesting function $p(t)$ and its estimate $\tilde{p}(t)$ tion in Fig. 4

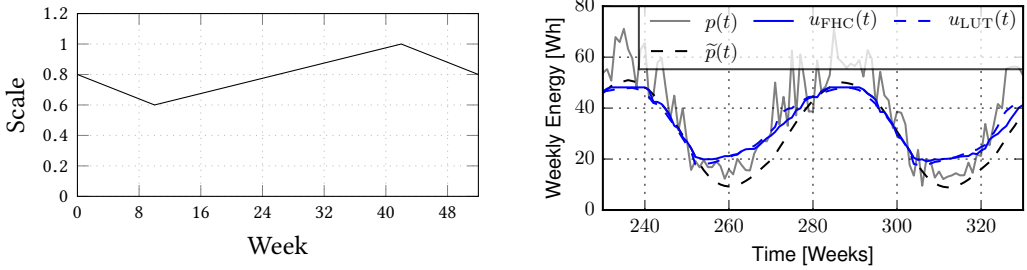


Fig. 6. Scaling function applied on the use function to avoid failure state in FHC scheme.

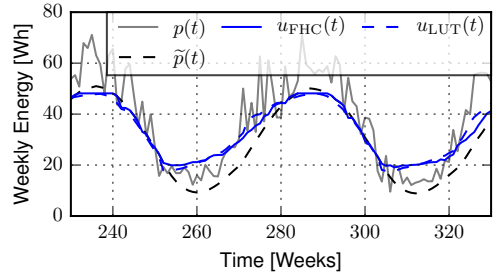


Fig. 7. Finite Horizon Scheme use function $u_{\text{FHC}}(t)$ for a given harvesting function $p(t)$ and its estimate $\tilde{p}(t)$

failures is that a small overestimation in energy can be fatal, particularly at times when the battery reserves are low. We notice that the battery is critically low towards the end of winter. Therefore, to avoid failures, the runtime estimator from [7] reviewed in *Sec. 5.5*, is scaled by the scaling function shown in *Fig. 6*. As shown in *Fig. 6*, the value of scaling function is dependent on the week of the year. This scaling ensures that the energy estimate is pessimistic in the winter months to avoid failure states. From this point on, all results of FHC will make use of this dynamically scaled energy estimation model. The scaling function in *Fig. 6* was carefully chosen so that FHC has no failures for data sets enumerated in *Table 1*. We use the same scaling function for all datasets. However, there is no guarantee that this scaling function works for all geographical locations. In fact, we will see later in this section that FHC may fail for other datasets/geographical locations.

Fig. 7 shows the use function of the FHC with the scaling function implemented. There are now no failures. Furthermore, *Fig. 7* also shows results for the Look-up Table (LUT) scheme. As shown in the figure, $u_{\text{FHC}}(t)$ and $u_{\text{LUT}}(t)$ are almost overlapping. However the LUT scheme tends to suffer a small penalty in v by being too conservative early in winter. Now, we compare all three schemes (CV, FHC, and SPM) for the same data. For SPM, weeks 230-330 were not used for building statistical models and $\lambda = 0.01$.

We see that SPM does not encounter a failure state. Furthermore, SPM does not require scaling function or other adaptations to avoid failure states. It inherently becomes more conservative in winter months due to the statistical models. The minimum value v of use function is also higher compared to FHC (20.0 vs 19.087). CV has the maximum minimum use function value of 21.31, but assumes full knowledge of the future harvesting energy. The battery levels for the three schemes are shown in *Fig. 9*. As shown in the figure, battery levels for SPM and FHC scheme never reach

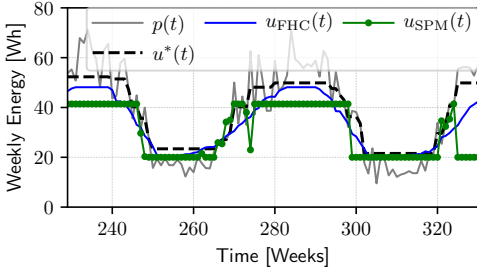


Fig. 8. Use functions of SPM, FHC and CV compared

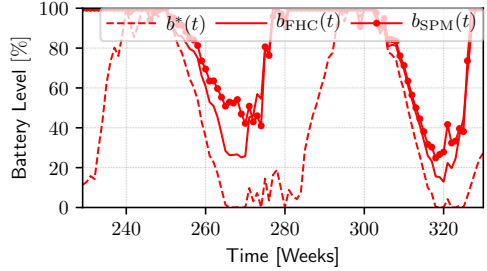


Fig. 9. Battery Levels corresponding to use functions given in figure 8

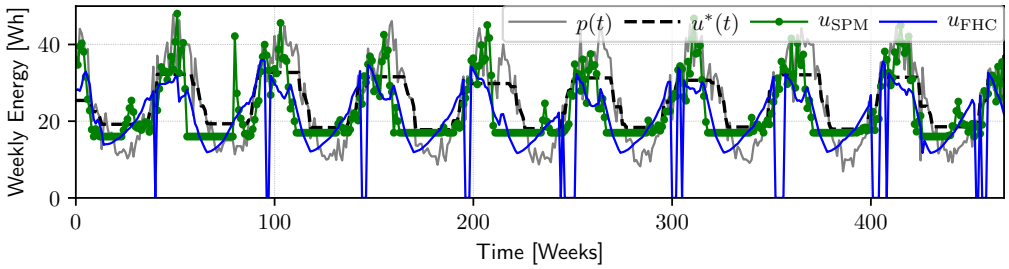


Fig. 10. Use functions of SPM, FHC and CV for Adelaide Australia

zero. Furthermore, SPM and FHC have similar minimum battery levels (14.39 and 14.67 for SPM and FHC, respectively).

These initial evaluations make a strong case for SPM. This is because SPM can give statistical guarantees, without requiring adaptations, while having better performance compared to FHC. However, the weak point of SPM is that it needs the availability of historical data to construct the statistical models while FHC does not have this requirement. We now show that SPM is also more robust. *Fig. 10* shows the use function of the three schemes for harvesting data of Adelaide, Australia, acquired from WRDC³ for years 1983-1992⁴. As *Fig. 10* shows, FHC encounters several failures (points in *Fig. 10* where $u_{FHC} = 0$) while SPM does not encounter any failure. The reason why FHC scheme encounters failures for this dataset is that the summer and winter months are shifted for the Adelaide dataset compared to other datasets in *Table 1*. Due to this season shift, the scaling function in *Fig. 6* no longer ensures that FHC is conservative in the winter months when the battery is low. Instead FHC ends up being unnecessarily conservative in the summer months when there is energy surplus. SPM accounts for this season inversion explicitly by creating appropriate statistical models. Furthermore, the minimum use function of SPM is close to the minimum use function of CV (16.0 vs 17.79).

7.2 Dynamic Adaptation

In this section, we present initial ideas on how SPM can be augmented to adapt to changes in the environment. To this point, the presented strategy is static in nature and assumes that the stochastic model of the energy harvesting source does not change. However, in practice the model can change

³World Radiation Data Centre: <http://wrdc-mgo.nrel.gov/>

⁴Year 1988 not considered due to incomplete data

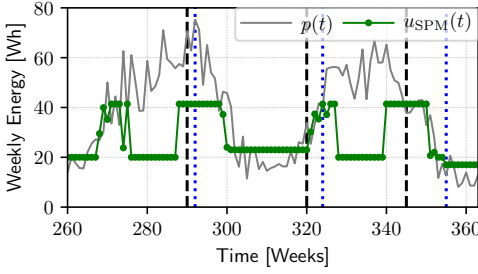


Fig. 11. SPM algorithm adapts to dynamic changes in the stochastic model.

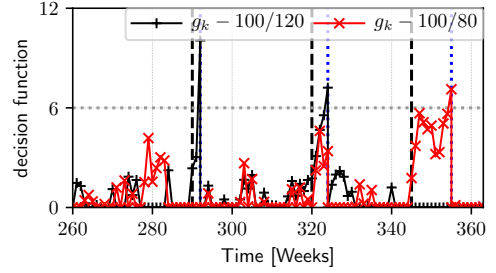


Fig. 12. The decision functions which are used to determine the stochastic models where the harvested energy is 80%, 100% or 120 %.

if the solar panel degrades or is occluded. We apply change detection theory to detect change in the model. Several approaches such as Shewhart Control Charts, CUSUM, and Bayes-type algorithms exist [3]. In this work, we evaluate the CUSUM algorithm due to its simplicity and low runtime overhead. In this section, we first provide a brief overview of this algorithm. We then explain how SPM is augmented to detect changes and adapt to them. For a more detailed description of the CUSUM algorithm, please refer to [3]. Finally, we demonstrate the functioning of the proposed adaptation scheme.

Let $p(t)$ be the harvested energy in week t . CUSUM algorithm uses two hypotheses. The null hypothesis (Θ_0) assumes no change in the harvesting model. The one hypothesis (Θ_1) signifies a change in a statistical model. A change is detected when the difference between the log-likelihood-ratio under the two hypotheses, and its current minimum value, exceeds a pre-set threshold. Specifically suppose that we observe energy y_i in week i . The log-likelihood-ratio under two hypotheses is given by: $S_k = \sum_{i=1}^k s_i$ where $s_i = \ln \frac{p_{\theta_1}(y_i)}{p_{\theta_0}(y_i)}$ and $P_{\theta_x}(y)$ is the probability of y under hypothesis X . Furthermore, we define the following decision function $g_k = S_k - \min_{1 \leq j \leq k} S_j$. A change is detected when the decision function exceeds a certain threshold h . Choice of the threshold h presents a tradeoff between the speed of change detection and the number of false detections. Following recursive formulation of the decision function also exists:

$$g_k = \begin{cases} g_{k-1} + s_k & \text{If } g_{k-1} + s_k > 0 \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

where $g_0 = 0$. After detection of the change, SPM can adapt its service to the precomputed optimal service under θ_1 stochastic models. To demonstrate this, we again take the MI dataset and artificially introduce three change points. The first change occurs at week 290 when the node starts getting 20% surplus energy. This surplus ends at week 320. At week 345, the energy harvesting node starts getting 20% less energy.

The SPM algorithm is adapted such that it decides between the three stochastic models where {80, 100, 120}% of the nominal energy is harvested. Two decision functions (17) are used for deciding between the neighbouring pairs of stochastic models, i.e., (100, 120) and (100, 80). Figure 11 shows that the scheme adapts to the changes. For this evaluation, the decision function threshold was empirically chosen to be 6. The two decision functions are plotted in Figure 12. As shown in Figure 11, SPM detects change in the stochastic models (indicated by the dotted blue line) after a variable delay. After this detection, SPM starts operating at the precomputed optimal setting for the detected stochastic model. This is evidenced by higher value of $v = 22$ Wh between weeks 300-320

and a lower $v = 18 \text{ Wh}$ after week 345. The nominal value of v is 20 Wh . In the presented example, battery depletion is not encountered. However, it should be noted that change in the stochastic model can increase probability of failure. However, after recovering from failure resulting from a change in stochastic model, the stochastic guarantee will again become valid. Making SPM adaptive to changes while maintaining its stochastic guarantee is part of future work.

8 REAL-WORLD IMPLEMENTATION AND EVALUATION

In this section, we present our sample implementation. This is done to demonstrate that SPM can be implemented with negligible runtime overhead. In addition, aspects of the real system which are not explicitly considered in our simplified model, such as maximum power point tracking and storage/power conversion inefficiencies, can also be taken into account.

8.1 System Implementation

Our sample system consists of photovoltaic energy harvester, a supercapacitor for energy storage, a power management unit, and a microcontroller unit consuming the energy. The photovoltaic panel used is a Panasonic AM-5412 of $50 \text{ mm} \times 33 \text{ mm}$ size, rated for 39 mW at 100 mW cm^{-2} . A 5 F supercapacitor was used as energy storage, and its charge ranges between 3.5 V and 4.2 V during operation. This small energy storage was chosen to allow for real-world energy storage dynamics during the sped up experiments, where one week corresponds to 120 s . In an actual deployment, the storage would accordingly be 5040 times larger, corresponding to a 18.9 Wh battery. We rely on a bq25570 harvesting management chip that performs passive Maximum Power Point Tracking (MPPT) for maximum energy transfer and supplies the consumer with 3.3 V . Finally, an MPS432P401R board is used as consumer, which is good representative microcontrollers used in embedded applications.

For the input energy trace, we consider a scaled version of the MI dataset. The scaling factor is chosen such that a maximum of 3.6 J are harvested in the 120 s interval by the solar panel. Using the scaled data, we compute the minimum use function v and safe charge vector C as explained in algorithm 4. These two values are stored on the node and determine the runtime power dissipation.

On the microcontroller, two tasks run on bare-metal: the adaptive consumer, and the power manager. The adaptive consumer varies the energy consumption by duty cycling a light emitting diode (LED). The maximum and minimum power dissipation (LED on and off) are measured to be 64.04 mW and 2.3 mW , respectively.

In our experimental evaluation, the power manager task runs periodically each 120 s ; corresponding to 1 week in a given real-world scenario. The power manager samples the voltage in the supercapacitor, and computes the stored energy. We call the usable energy in the capacitor at the start of the i^{th} interval E_i^{cap} . If the energy is lower than the safe charge C_i , the adaptive load is configured such that the energy consumption over the next period is v . If the energy is higher than the safe charge, the adaptive load is configured such that it consumes $v + E_i^{\text{cap}} - C_i$. The SPM algorithm has a minimal runtime overhead of less than 1 ms , resulting in a negligible energy overhead compared to the consumer energy consumption.

8.2 Setup and Results

For real world evaluation, we deploy the described system in a solar illuminates emulator. We use this to emulate the weekly average solar radiance of the MI dataset from year 1965 over a 2 year time interval; corresponding to the experiment length of 208 min . The deployment was instrumented using a RocketLogger [26] to monitor harvesting power, storage voltage, and system's power dissipation. Parallel monitoring of digital state pins of the implemented system allows tracking the state of the adaptive consumer and power manager tasks.

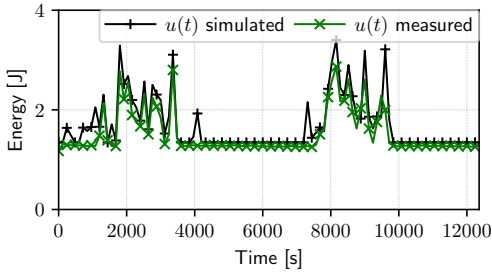


Fig. 13. Measured energy consumption for SPM matches its expected value with Mean Square Error (MSE) = 0.317

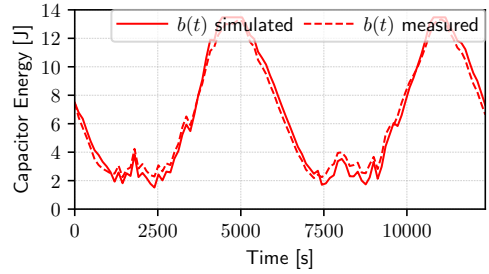


Fig. 14. Measured energy stored in the supercapacitor matches its expected value with MSE = 0.071

We processed the collected data on energy consumption and energy stored in the capacitor and compared it to simulation results. Figure 13 shows the energy consumption and Figure 14 shows the energy stored in the capacitor at the start of each 120 s interval. As shown in the figures, our measurements closely match the expected values which are acquired through simulating SPM and assuming the empirically observed efficiency of the power conversion circuit (92%). The error in the energy consumption can likely be decreased further by a variable, operating point based, efficiency characterization. The error in capacitor energy is less compared to the error in energy consumption because the safe charges provide an energy correction mechanism.

The presented implementation clearly demonstrates that non-linearities caused by power conversion circuit and MPPT can be accounted for in our model. Furthermore, SPM is shown to have negligible run time computation overhead (less than 1 ms). The memory overhead is also small (52 integer constants). Therefore, the implementation of SPM on a resource constrained embedded system is viable.

8.3 Experimental Results

We now compare the proposed schemes for all datasets given in *Table 1*. We also compare the proposed schemes with the baseline algorithms. For SPM, we always exclude the year which is being simulated from the statistical models. The minimum use function (v) and the total utility (U) values are reported for all datasets and all algorithms in *Table 1*. Note that the values of v reported in *Table 1* are daily minimum energy consumption values (as opposed to weekly values reported in earlier examples). The first year of all datasets is excluded from the analysis and metrics reported are for the remaining 11 years. *Fig. 15* and *Fig. 16* show box-whisker plot of the normalized v and U values respectively. All values are normalized with respect to corresponding CV values.

All of the baseline algorithms are parameterized with author's recommended, and arguably best parameters. Furthermore, all algorithms utilize the same adequately provisioned power subsystem, which is designed with an expected duty-cycle of 40%. However, due to limited prediction accuracy, EWMA and WCMA are required to operate at a relatively high control frequency to achieve acceptable performance. While ENO-Max is theoretically able to run at much lower control frequencies, the original parameterization was done for half hour update periods. Since the datasets used are given in hourly values, we use $t = 1$ hour for simulation of these baseline algorithms. For EWMA, WCMA, and ENO-Max, we report the result for both a pessimistic duty-cycle $\rho_{min} = 1\%$ (as is done in e.g., [29][16]), and, listed in parentheses in *Table 2*, a more realistic (for our scenario) value of duty-cycle $\rho_{min} = 30\%$.

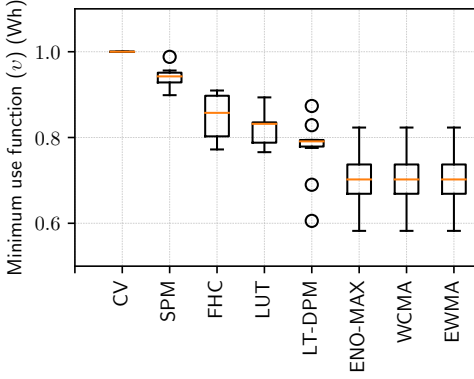


Fig. 15. normalized minimum use function

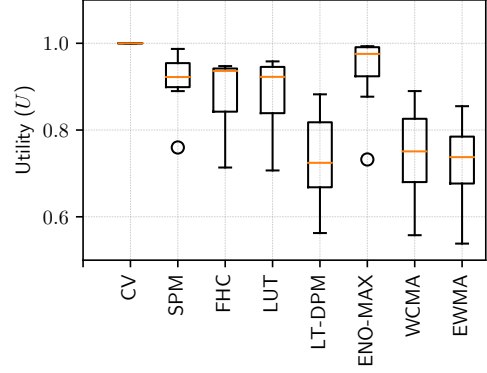


Fig. 16. Normalized utility

		Dataset									
		TX2	TX1	CA	MD	MI	OR	ON	AK	DH	GG
CV	v	3.62	4.19	3.76	3.88	3.04	2.63	2.76	1.58	19.65	17.61
	U	3235.2	3625.4	3439.4	3603.1	3453.6	3349.5	3612.3	3070.9	2174.9	2145.1
SPM	v	3.43	4.14	3.57	3.71	2.85	2.43	2.57	1.42	-	-
	U	3060.4	3550.2	3200.2	3556.3	3114.9	3061.4	3214.1	2333.25	-	-
FHC	v	2.86	3.76	3.42	3.38	2.73	2.36	2.32	1.22	15.47	14.86
	U	3026.1	3399.9	3244.1	3407.5	3272.0	3139.6	3210.0	2336.4	1798.5	1530.7
LUT	v	3.01	3.71	3.01	3.03	2.53	2.35	2.30	1.21	15.4	14.72
	U	3023.7	3474.1	3177.0	3437.7	3278.2	3086.4	3201.7	2320.9	1790.1	1516.2
LT-DPM	v	3.00	3.66	2.98	3.06	2.4	2.04	2.19	1.09	11.9	13.99
	U	2854.5	3082.6	2857.8	2806.5	2549.1	2380.5	2385.8	1727.4	1502.3	1342.38
ENO-MAX	v	0.10 (2.98)	0.10 (3.13)	0.54 (2.54)	0.09 (2.82)	0.52 (2.06)	0.06 (1.93)	0.66 (1.79)	0.03 (0.92)	-	-
	U	3198.4 (3205.4)	3600.3 (3601.2)	3409.1 (3410.3)	3566.7 (3568.5)	3317.9 (3317.7)	3143.5 (3147.1)	3170.1 (3167.6)	2236.5 (2248.4)	-	-
WCMA	v	0.18 (2.98)	0.19 (3.13)	0.15 (2.54)	0.17 (2.82)	0.12 (2.06)	0.12 (1.93)	0.11 (1.79)	0.06 (0.92)	-	-
	U	1989.0 (2878.7)	2266.8 (3073.2)	2104.1 (2815.9)	2101.5 (2796.6)	1858.0 (2505.2)	1789.4 (2322.1)	1706.2 (2312.6)	1356.4 (1712.2)	-	-
EWMA	v	0.18 (2.98)	0.19 (3.13)	0.15 (2.54)	0.17 (2.82)	0.12 (2.06)	0.12 (1.93)	0.06 (1.79)	0.03 (0.92)	-	-
	U	1735.2 (2766.0)	1877.0 (2887.0)	1859.1 (2686.2)	1907.4 (2716.2)	1775.8 (2490.9)	1616.7 (2313.8)	1549.2 (2291.8)	1124.9 (1652.7)	-	-

Table 2. Minimum daily energy use function (v) and total utility (U) for CV, FHC, LUT, and the baselines LT-DPM, ENO-Max, EWMA, and WCMA. Values in parentheses are for $\rho_{min} = 30\%$

Comparison of the proposed schemes (Clairvoyant (CV), Stochastic Power Management (SPM), Finite Horizon Control (FHC), and Look-up Table (LUT)). It is very promising to note that the minimum use function v values for SPM are close to the optimal v values returned by CV. We observe a minimum, average and maximum percentage difference of 1.2%, 6.3%, and 11.3%, respectively. The utility values show a similar behavior. FHC also performs fairly well for these datasets. However, the observed value of v is lower for all datasets compared to the corresponding values given by the SPM scheme. The utility U does not degrade as much though; and for MI, OR,

ON, AK, the utility values for FHC are actually higher compared the corresponding SPM values. This is because SPM may, on occasion, waste energy by not utilizing available energy even when the battery is full. This leads to an overall loss in utility. A promising result is that LUT performs very close to FHC. Therefore, implementation of the scheme on a sensor mote is feasible by employing LUT without a significant loss in performance. As stated before, the FHC scheme does not require long-term historical data which is a fundamental requirement of SPM. Consequently, if long-term data is not available, FHC/LUT would be a choice for implementation.

Comparison of the baseline schemes (LT-DPM, ENO-MAX, WCMA, EWMA). In terms of the minimum use function v , Long-Term Dynamic Power Management (LT-DPM) performs fairly well; with performance comparable to FHC and LUT. However, LT-DPM demonstrates a considerable loss in utility. The reason for this is that LT-DPM takes a long-term approach to compute energy consumption by considering both the period of deficit, i.e., when the generation is below consumption, and the period of surplus, i.e., generation exceeds consumption. This approach tries to fully leverage the battery in order to bridge periods of deficit, while simultaneously ensuring that the employed panel can actually recharge the battery during periods of surplus. This conservative approach has a detrimental effect on the total achievable system utility since surplus energy is not used to aggressively increase system energy consumption in periods of energy surplus. In comparison, the FHC and SPM algorithms are able to leverage the surplus energy during summer, while ensuring that the battery is full/adequate at the appropriate time to ensure continuous operation.

Moving on to other baseline schemes, Exponentially Weighted Moving Average (EWMA) and Weather Conditioned Moving Average (WCMA) perform comparably, as expected. This is not surprising, as the same energy allocation scheme is used. Hence, the difference stems from their prediction accuracies. In fact, WCMA achieves a better v only for two datasets (ON, AK), but improves over EWMA by up to 20.5% in system utility for $\rho_{min} = 1\%$ and 6.5% for $\rho_{min} = 30\%$. Both of these algorithms are unable to leverage the battery, causing the v to follow the user expected ρ_{min} , even when the battery is full. As has been shown in [7], if the power subsystem is not adequately provisioned, achieving even ρ_{min} may not be possible.

The Energy Neutral Operation with Maximal energy efficiency (ENO-Max) algorithm has a similar limitation. Although achieving the highest U of all non-clairvoyant algorithms, v varies greatly, and tends to be the lowest of all evaluated algorithms when using $\rho_{min} = 1\%$. For $\rho_{min} = 30\%$ the minimum performance is equivalent to that achieved of EWMA and WCMA. This is to be expected, as the range of acceptable utilization to chose from is significantly reduced. In fact, with $\rho_{min} = 30\%$, the utilization computed by these algorithms is overridden by ρ_{min} most of the time. We attribute the pessimistic v obtained with these three algorithms to their battery agnostic nature.

9 LIMITATIONS AND REQUIREMENTS

Historical Data. The Stochastic Power Management (SPM) scheme requires availability of long-term historical data. Emperically, SPM performs well if data is available for 10 years. However, in many scenarios, this data may not be available leading to low performance due to imprecise statistical models.

Energy Estimation. Finite Horizon Control (FHC) and Look-up Table (LUT) schemes rely on accurate energy estimates. Finding an appropriate energy estimation model that is neither too conservative nor too optimistic has been shown to be difficult, as the weather patterns that affect the solar energy incident are hard to model [9] and difficult to predict [15]. Even though the energy model used in this work functions well for all datasets in *Table 1*, its performance is not always guaranteed (e.g. *Fig. 10*).

Measurement Support. The proposed scheme requires that the system can measure or approximate the total harvested energy over a given time period t . This can be accomplished by measuring the output power of the panel, or inferring the harvested energy through battery State-of-Charge information. The former is the preferred choice, but incurs additional overhead in terms of measurement circuitry and continual processing. The advantage of the latter, e.g., [6] is that it may not require special purpose hardware. However, obtaining an accurate SoC indication is non-trivial [4].

Battery Inefficiencies. Batteries are non-ideal storage elements, which suffer from a variety of inefficiencies that are dependent on the specific battery chemistry and load behavior [4]. In our simulation, *charging* efficiency is taken into account and *discharging* efficiency is ignored. However, it is straightforward to account for discharging efficiency by scaling the use function values. *Leakage power* is ignored in this discussion. Considering the periodically recurring recharging opportunities, accounting for leakage is not as crucial as it is for purely battery operated devices. *Temperature* may impact the battery's apparent capacity [4]. Thus, for deployments that are exposed to high temperature variations, it may be necessary to account for the temporary change in apparent battery capacity imposed by temperature effects. Finally, *battery aging* is not likely to be a problem, since our approach causes the battery to experience only one deep discharge cycle per year, and is therefore expected to outlast the lifetime of other system components, e.g., electronics, mechanical parts, e.t.c., *Solar panel degradation* has been shown to generally be aesthetic in nature, and not significantly affect the panel's efficiency [12].

10 CONCLUSION

We have shown that the proposed Stochastic Power Management (SPM) approach successfully satisfies the objective of providing a guaranteed minimum energy utilization, and therefore minimum service level. To the best of our knowledge, we are the first to provide an analytic solution for solar energy harvesting embedded systems, which ensures uninterrupted operation at a maximized minimum energy consumption. We propose two different power management approaches. The first approach relies on energy estimates and is applicable to any energy harvesting source, presuming an appropriate energy estimator is available. The second approach exploits stochastic harvesting energy models and can give strict probabilistic guarantees on not encountering failure state. We extensively evaluate these schemes using eight synthetic, and two real-world datasets. Our approaches significantly outperform four baseline implementations in terms of minimum energy consumption *and* total utility. Finally, the stochastic scheme performs very well and has performance which is comparable to that of an optimal clairvoyant algorithm. The implementation of the stochastic scheme validates the presented model and shows that the scheme has minimal runtime overhead.

REFERENCES

- [1] Rehan Ahmed and others. *Optimal Power Management With Guaranteed Minimum Energy Utilization For Solar Energy Harvesting Systems*. Technical Report. Computer Engineering and Networks Laboratory ETHZ.
- [2] C. Arnhardt and others. 2007. Sensor based Landslide Early Warning System-SLEWS. Development of a geoservice infrastructure as basis for early warning systems for landslides by integration of real-time sensors. *Geotechnologien science report* 10 (2007), 75–88.
- [3] Michèle Basseville, Igor V Nikiforov, and others. 1993. *Detection of abrupt changes: theory and application*. Vol. 104. Prentice Hall Englewood Cliffs.
- [4] Hendrik Johannes Bergveld. 2001. *Battery management systems: design by modelling*. Ph.D. Dissertation. Enschede. <http://doc.utwente.nl/41435>
- [5] Jan Beutel and others. 2011. X-Sense: Sensing in extreme environments. In *Design, Automation & Test in Europe Conference & Exhibition, 2011*. IEEE, 1–6.
- [6] Bernhard Buchli and others. 2013. Battery state-of-charge approximation for energy harvesting embedded systems. In *Proc. of 10th European Conference on Wireless Sensor Networks*. Springer, 179–196.

- [7] Bernhard Buchli and others. 2014a. Dynamic Power Management for Long-Term Energy Neutral Operation of Solar Energy Harvesting Systems. In *Proc. 12th ACM Conference on Embedded Networked Sensor Systems*. Memphis, TN, USA.
- [8] Bernhard Buchli and others. 2014b. Towards Enabling Uninterrupted Long-Term Operation of Energy Harvesting Embedded Systems. In *Proc. of 11th European Conference on Wireless Sensor Networks*. Springer Link, Lecture Notes on Computer Science, Oxford, UK.
- [9] Matteo Buzzi. 2008. *Challenges in operational numerical weather prediction at high resolution in complex terrain*. Ph.D. Dissertation. Diss., Eidgenössische Technische Hochschule ETH Zuerich, Nr. 17714.
- [10] Marcus Chang and others. 2010. Meeting ecologists' requirements with adaptive data acquisition. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 141–154.
- [11] Shengbo Chen and others. 2011. Finite-horizon energy allocation and routing scheme in rechargeable sensor networks. In *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2273–2281.
- [12] Peter Corke and others. 2007. Long-duration solar-powered wireless sensor networks. In *Proceedings of the 4th workshop on Embedded networked sensors*. ACM, New York, NY, USA, 33–37.
- [13] J.V. Dave and others. 1975. Computation of Incident Solar Energy. *IBM Journal of Research and Development* 19, 6 (1975), 539–549. DOI : <http://dx.doi.org/10.1147/rd.196.0539>
- [14] L. Guibas and others. 1987. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica* 2, 1-4 (1987), 209–233.
- [15] Detlev Heinemann and others. 2006. Forecasting of solar radiation. *Solar energy resource management for electricity generation from local level to global scale*. Nova Science Publishers, New York (2006).
- [16] Aman Kansal and others. 2007. Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems* 6, 4 (2007), 32.
- [17] Wook Hyun Kwon and others. 2006. *Receding horizon control: model predictive control for state models*. Springer.
- [18] Trong Nhan Le and others. 2012. Power Manager with PID controller in Energy Harvesting Wireless Sensor Networks. In *Green Computing and Communications, 2012 IEEE International Conference on*. IEEE, 668–670.
- [19] Dexin Li and Pai H Chou. 2004. Maximizing efficiency of solar-powered systems by load matching. In *Low Power Electronics and Design, 2004. ISLPED'04. Proceedings of the 2004 International Symposium on*. IEEE, 162–167.
- [20] Fajie Li and others. 2007. Rubberband algorithms for solving various 2D or 3D shortest path problems. In *Computing: Theory and Applications, 2007. International Conference on*. IEEE, 9–19.
- [21] Fajie Li and others. 2011. *Euclidean Shortest Paths*. Springer.
- [22] Chao Lu, Sang Phill Park, Vijay Raghunathan, and Kaushik Roy. 2012. Low-overhead maximum power point tracking for micro-scale solar energy harvesting systems. In *VLSI Design (VLSID), 2012 25th International Conference on*. IEEE, 215–220.
- [23] Chao Lu, Vijay Raghunathan, and Kaushik Roy. 2011. Efficient design of micro-scale energy harvesting systems. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 1, 3 (2011), 254–266.
- [24] J Recas Piorno and others. 2009. Prediction and management in energy harvested wireless sensor nodes. In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. 1st International Conference on*. IEEE, 6–10.
- [25] Vijay Raghunathan and Pai H Chou. 2006. Design and power management of energy harvesting embedded systems. In *Proceedings of the 2006 international symposium on Low power electronics and design*. ACM, 369–374.
- [26] Lukas Sigrüst, Andres Gomez, Roman Lim, Stefan Lippuner, Matthias Leubin, and Lothar Thiele. 2017. Measurement and Validation of Energy Harvesting IoT Devices. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*. EDAA, Lausanne, Switzerland, 1159–1164. DOI : <http://dx.doi.org/10.23919/DATE.2017.7927164>
- [27] Philipp Sommer and others. 2014. The big night out: Experiences from tracking flying foxes with delay-tolerant wireless networking. In *Real-World Wireless Sensor Networks*. Springer, 15–27.
- [28] Jay Taneja and others. 2008. Design, Modeling, and Capacity Planning for Micro-solar Power Sensor Networks. In *Proc. of the 7th international conference on Information processing in sensor networks*. IEEE Computer Society, Washington, DC, USA, 407–418.
- [29] Christopher M. Vigorito and others. 2007. Adaptive control of duty cycling in energy-harvesting wireless sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. 4th Annual IEEE Communications Society Conference on*. IEEE, 21–30.
- [30] Eric W Weisstein. 2003. Maximum likelihood. (2003).
- [31] Pei Zhang and others. 2004. Hardware design experiences in ZebraNet. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, New York, NY, USA, 227–238.