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

Bernhard Buchli, Pratyush Kumar, Lothar Thiele
ETH Zurich, Computer Engineering and Networks Laboratory, Zurich, Switzerland
FirstName.LastName@tik.ee.ethz.ch

***Abstract*—In this work we present the first formal study on optimizing the energy utilization of energy harvesting embedded systems while giving bounds on the minimum energy usage. Furthermore, to deal with the uncertainty inherent to solar energy harvesting we propose to use (i) a finite horizon scheme, and (ii) a non-uniformly scaled energy estimation based on an astronomical model. Under certain realistic assumptions, the finite horizon scheme can provide guarantees on minimum energy utilization, and therefore minimum utility. We show that a single non-uniform scaling function is applicable to solar energy traces from diverse locations. We further propose and evaluate a piece-wise linear approximation for efficient implementation as a small look-up table for 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 solution is highly effective at providing a guaranteed minimum utilization, and significantly out-performs four previously proposed solutions.**

I. Introduction

Motivation. Advances in Wireless Sensor Network (WSN) research have enabled many new application scenarios in diverse scientific fields, *e.g.*, geology [4], ecology [10], and zoology [23], [26]. These *WSNs* are comprised of sensor motes, which contain processing, communication, and various sensing capabilities. The power to operate the motes is generally supplied by batteries with finite capacity. However, many scenarios require autonomous system operation over time periods on the order of multiple years, which may not be achievable with finite batteries alone. To alleviate the shortcomings of battery powered systems, ambient energy harvesting, particularly in the form of solar energy harvesting, has been investigated as a feasible means to enable long-term operation. Many deployment reports have shown that energy harvesting can boost the system performance, and prolong the mission time.

Experience has shown that these improvements stem from a combination of adequate power subsystem planning, *e.g.*, [8], [17], and harvesting-aware dynamic power management, *e.g.*, [7], [22], [25]. Despite significant efforts, particularly in the area of runtime power management, the proposed schemes generally aim at maximizing short-term energy efficiency and operate on a best-effort basis. Achieving dependable long-term operation, *i.e.*, on the order of multiple years, has shown to be difficult despite leveraging a periodically recurring energy source [24]. The inability to provide minimum performance guarantees has so far prevented solar energy harvesting *WSNs* to enter important application domains, *e.g.*, safety-critical [1], and long-term monitoring and surveillance [4], where performance guarantees must be given.

To advance energy harvesting WSN technology to the domain of safety-critical systems, a minimum acceptable service must be guaranteed despite relying on an energy source that is highly variable both in time and amount of energy availability. Informally, a system that can sustain a pre-defined minimum performance level *without* interruption is said to provide a minimum acceptable service level. 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 the energy surplus that may likely be available during summer. 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.

Achieving the above objectives is complicated due to the highly variable, and only partially predictable energy profile. Greedily using the available energy in an attempt to maximize the utility instead of provisioning for times of deficit may cause system outages at a later time. Not only does this incur high penalties due to low-power re-connect hysteresis [8], but it also violates the minimum service objective. On the other hand, conservative short term usage may lead to low minimum service, and the risk of battery overflow, *i.e.*, the battery is full and the surplus energy exceeds the maximum system consumption. This significantly reduces the energy efficiency and system utility, as the excess energy is neither stored nor consumed. However, in reality it is practically impossible to always fully utilize all available energy without wastage.

Contemporary approaches to harvesting-aware power management can be classified as (i) predictive, and (ii) reactive approaches. Predictive approaches, *e.g.*, [17], [22], attempt to improve the system utility by *predicting* the harvestable 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, acceptable prediction accuracy with the limited computational resources available on contemporary WSN motes 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 [19], [25].

The approaches presented in literature are generally based on feedback-control, and heuristically attempt to trade-off the above objectives without formally defining the underlying optimization problem to be solved. Furthermore, optimality

and superiority of a particular approach very much depends on the application scenario under consideration, and the applied evaluation criteria. Most importantly, however, despite over a decade of research in this domain [16], none of the approaches published so far are able to provide minimum performance guarantees for solar energy harvesting systems.

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 used across all time intervals. 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 service level such that the minimum energy used, *i.e.*, use function, across *all* intervals is maximized. To identify an efficient algorithm to compute the optimal use function, we establish a relation between the 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 the use function 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.

A natural concern is whether an algorithm can be implemented on resource constrained low-power embedded devices typically used in WSN setups. We therefore present an approximated scheme where the use function is computed offline and represented in a look-up table. At runtime, only a few simple operations with the values in the look-up table are required to obtain the use function. The final point to consider is the large variability in the energy profile, which is not modeled in the algorithm, and must therefore be adequately represented in the estimate. To this end, we propose a non-uniform energy estimator based on an astronomical solar radiation model [7].

We experimentally validate the different proposed theoretical ideas on datasets available in a public database and raw data collected with our own deployment [4]. While the theoretical analysis is presented without taking conversion and storage inefficiencies into account, the experiments consider various efficiency factors to evaluate our approach in a realistic setting.

With the experiments we show the following key results. First, the clairvoyant algorithm computes optimal use functions that satisfy certain necessary conditions, as required by the connection to the Euclidean shortest path problem. Second, we show that, when relying on the energy estimation from [8], the non-clairvoyant, optimal algorithm leads to failure states, *i.e.*, the battery drains prematurely. However, with our proposed non-uniformly scaled estimator, the algorithm computes sustainable use functions for datasets from diverse locations. Third, the proposed FHC scheme improves the minimum energy usage, and greatly increases the total used energy when compared to baseline approaches. Fourth, we show that the efficient look-up table (LUT) approach results in only a marginal loss in performance with respect to the FHC scheme. In addition, the memory overhead of the LUT is found to be reasonably small for various data sets. Finally, we show that the proposed approach significantly out-performs all four algorithms that are used as baseline for performance comparison. The FHC scheme reaches the clairvoyant algorithm's performance to within 9.9% at best and 29.5% at worst.

II. 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* harvested energy. In contrast, we study the objective of maximizing the minimum used energy 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 known databases.

Dynamic Power Management. The first dynamic power management scheme for solar energy harvesting *WSNs* was proposed within a theoretical framework that defines Energy Neutral Operation (ENO) as the fundamental limit of energy harvesting systems [17]. ENO is achieved if the system consumes only as much energy as is harvested over a given time period δ. The day is discretized into slots of equal duration δ, 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 [22] 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 [25], a model-free approach to dynamic performance scaling is presented. Linear-Quadratic Tracking is used to dynamically adapt the system's duty-cycle based on the battery State-of-Charge (*SOC*) and so attempt to ensure 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, it 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]. This scheme 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.

III. 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 when applied to realistic scenarios are described in *Sec. VI* and *VII*.

A. 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*, 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 or store-consume inefficiencies. However, for the experimental validation we use a realistic model that considers various inefficiencies (see *Sec. VII*).

The *harvested* and *used energy* during the time interval $[t, t+1)$ are denoted as $p(t)$ and $u(t)$, respectively. While the harvested energy $p(t)$ is provided by the environment, the used energy $u(t)$ 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 any reasonable value $u(t)$. 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. Note that we are interested in providing a minimum service level *at all times*, hence the duty-cycle will not take arbitrarily low values and will therefore not introduce undue overhead. Moreover, we assume a properly provisioned power subsystem [8], which can safely provide the maximum power drawn by the system. A more refined model that takes upper- and lower-bounds on the energy consumption into account will be discussed in *Sec. VII*.

Given the battery's current state $b(t)$, the harvested $p(t)$ and used $u(t)$ energy, the system can be described by

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

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.

As described above, the controller can influence the energy consumption of the device by changing $u(t)$. Generally, a lower energy also leads to a lower service, *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

$$U(t_1, t_2) = \sum_{\tau=t_1}^{t_2-1} \mu(u(\tau)) \tag{2}$$

for some strictly concave[1] utility function $\mu : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$. 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 no additional utility beyond a certain consumed energy.

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

B. 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 harvest 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 $b^*(t) + p(t) - u^*(t) \geq 0,\ \forall 0 \leq t < T$, where $b^*(t)$ denotes the stored energy with harvest 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 provide a minimal service as high as possible .

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 the one that maximizes the total used energy, *i.e.*, maximizes $\sum_{t=\{0,1,\ldots,T-1\}} u(t)$.

It is by no means obvious how the use function $u^*(t)$ is to 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. Furthermore, it is not clear if there exists an efficient algorithm to determine the optimal use function $u^*(t)$, if it indeed exists.

IV. 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 harvested 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. V*, we will consider the problem where only an estimate of the harvested energy is known.

A. Necessary Conditions for Optimality

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

Theorem 1: Given a use function $u^*(t)$ such that the system never enters a failure state. If the relations in *(3)* hold

$$\begin{aligned} u^*(s-1) < u^*(s) &\Rightarrow b^*(s) = 0 \\ u^*(t-1) > u^*(t) &\Rightarrow b^*(t) = B \end{aligned} \tag{3}$$

then $u^*(t)$ is (i) optimal with respect to maximizing the minimal used energy among all use functions, and (ii) unique.

In other words, if we are able to construct a use function that satisfies (3) and never leads to a failure, then the minimal used energy is optimally maximized. We refer to this use function $u^*(t)$ as the optimal use function. This result defines a necessary and sufficient condition on the optimality to satisfy criteria **C1** and **C2**. Further, as only one use function satisfies this necessary condition, it also trivially satisfies criterion **C3**.

We will now show the sequence of lemmas that lead to the above fundamental result. All proofs are given in *Appendix* A.

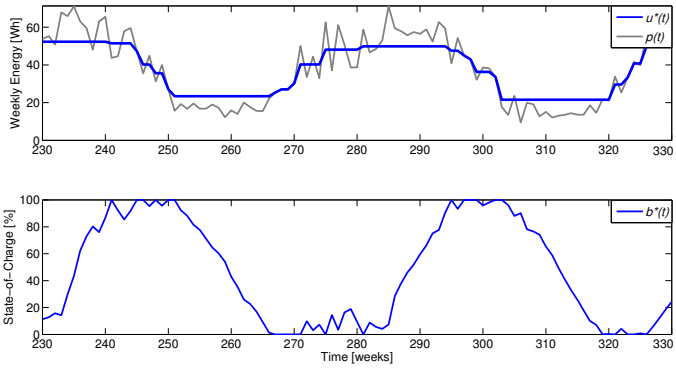

Fig. 1: (top) Optimal use function $u^*(t)$ for a given harvest function $p(t)$ and (bottom) corresponding stored energy $b^*(t)$.

Lemma 1 (Necessary Condition): Any optimal use function $u^*(t)$ satisfies *(4)* and (*5*).

$$\begin{aligned} &\forall s \leq \tau \leq t : 0 < b^*(\tau) < B \Rightarrow \\ &\forall s-1 \leq \tau \leq t : u^*(\tau) = u^*(t) \end{aligned} \tag{4}$$

$$\begin{aligned} u^*(s-1) < u^*(s) &\Rightarrow b^*(s) = 0 \\ u^*(t-1) > u^*(t) &\Rightarrow b^*(t) = B \end{aligned} \tag{5}$$

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 necessary condition applies to all possible harvesting functions $p(t)$. Next we show that $u^*(t)$ is a unique solution.

Lemma 2 (Uniqueness): If there exists a use function $u^*(t)$ that satisfies the necessary optimality conditions of Lemma 1 and that does not lead to a failure, 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 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 counterintuitive as one would expect that in case of a full battery it is reasonable to increase the energy usage, and in case of an empty battery one would decrease the energy usage. 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. 1*). Therefore, it makes sense to increase the energy usage upon entering such a period as now there is more harvested energy available. Conversely, a full battery marks the end of a surplus period and therefore, the usage of energy needs to be reduced.

Having shown the conditions for an optimal use function $u^*(t)$, it remains to be shown if there is an efficient algorithm to determine $u^*(t)$ for any given energy input $p(t)$.

B. Algorithm

In this section we show that there is an unexpected relation between the optimal power management problem for harvesting systems and a problem in computational geometry, *i.e.*, the shortest Euclidean path in simple polygons. The geometric problem 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.

Algorithm 1 Iterative algorithm for optimal control.

Require: $b(0)$; $b(T)$; B ; ϵ ; T; $(\forall\ 0 \leq t < T : p(t))$;
Ensure: $(\forall 0 \leq t < T : u(t))$;
  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**
      $f(t) \leftarrow (f(t-1) + f(t+1))/2$;
      $f(t) \leftarrow \mathrm{Max}(\mathrm{Min}(f(t), \sigma_u(t)), \sigma_l(t))$;
    end for
  until $\mathrm{Max}(|f' - f|) < \epsilon$
  for all $0 \leq t < T$ **do**
    $u(t) \leftarrow f(t+1) - f(t)$;
  end for

Similar to [11], we will 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

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

$$\sigma_u(t) = \sigma_l(t) + B \tag{7}$$

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

$$u(t) = f(t+1) - f(t) \quad \forall\ 0 \leq t < T. \tag{8}$$

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 run-time optimal algorithm from [14] is difficult to implement due to the use of advanced data structures and complex operations. In *Sec. V-D* 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. run-time 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)* and (*7*), as well as the initial and final points of a feasible path. Then, the initial path is chosen to

Algorithm 2 Iterative algorithm for periodic optimal control.

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

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. Results are described in *Sec. VII*.

V. 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. However, in reality only an estimate of the harvested energy in the future is available. In this section, we present a finite horizon control scheme that allows to formulate bounds on the achieved optimality despite of the fact that only an energy approximation is available.

To this end, we suppose that for all time instances $t \geq 0$ an energy estimation $\tilde{p}(t)$ is available that provides a lower bound on the actual harvested energy $p(t)$ in $[t, t+1)$:

$$p(t) \geq \tilde{p}(t) \qquad \forall\ t \geq 0$$

Since we consider solar energy harvesting systems in this paper, we can assume that $\tilde{p}(t)$ exhibits periodic behavior, *e.g.*, a yearly repeating summer/winter pattern. We suppose that the periodicity is P, and therefore we have

$$\tilde{p}(t) = \tilde{p}(t+P) \qquad \forall\ t \geq 0.$$

Moreover, the time interval we consider is expected to be an integral multiple of the period P, *i.e.*, $T = P, 2P, \ldots$.

The derivation of the finite horizon control scheme and its correctness proof involves three steps: At first, we will derive the optimal control for the above periodic energy approximation. In the second step, we will use this result to derive the adaptive finite horizon scheme. Finally, we will show that this scheme will never provide worse results than the optimal control for the estimated harvested energy $\tilde{p}(t)$. This way, (i) the system never enters a failure state, (ii) a lower bound of the use function can be provided, and (iii) a lower bound on the *utility* of the use function can be given. Note that the specific choice of utility function is very application specific (see also *Sec. VII*).

A. Optimal Periodic Control

As described above, we suppose a periodic estimated harvested energy $\tilde{p}(t)$. By assuming that the state of the battery at the end of the period $\tilde{b}(P)$ equals that at the beginning $\tilde{b}(0)$, we find that, in case of the optimal control, all essential quantities are periodic, *i.e.*, optimal use function, energy harvesting function and battery state: $\tilde{u}^*(t) = \tilde{u}^*(t+P)$ and $\tilde{b}^*(t) = \tilde{b}^*(t+P)$. Let us further suppose that the control horizon T equals the period of the harvested energy: $T = P$.

The optimized use function can then 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 finite horizon control scheme.

Algorithm 3 Finite horizon scheme for optimal control

Require: t ; b ; T ; $\tilde{b}(t+T)$; $(\forall 0 \leq \tau < T : \tilde{p}(t+\tau))$;
Ensure: 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 $(\forall 0 \leq t < T : u(t))$;
  $u \leftarrow u(0)$;

B. Finite Horizon Scheme

Following the ideas of receding horizon control, *e.g.*, [18], we replace the unknown harvested energy function with its estimate and optimize the future use function with the current battery state. From this optimized use function, we only execute the first time step. For the next time step, we again optimize the future use function, and so on.

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

The performance of the finite horizon scheme crucially depends on the estimate $\tilde{p}(t)$. 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 $p(t) \geq \tilde{p}(t)$. Next we will identify the guarantees that can be provided with such an estimator.

C. Guarantees

Here we make the realistic assumption that the estimate $\tilde{p}(t)$ is *static*. By this we mean that the estimate for the amount of harvested energy in any given time interval within the period is a constant across all corresponding periods. In other words, we assume that the estimate of the harvested energy in a certain week is the same across all years. As we will illustrate in *Sec. VII*, an estimate based on an astronomical radiation model [13] is appropriate for datasets from various locations.

With a static estimate $\tilde{p}(t)$, there are two options for designing a non-clairvoyant algorithm. First, with *Algorithm 1* we can use the estimate $\tilde{p}(t)$ to compute a static use function $\tilde{u}(t)$ for all time steps within the periodic window $[0, P)$. Second, with the finite horizon scheme of *Algorithm 3* we can recompute the use function $\tilde{u}(t)$ at each time step. The expectation is that the finite horizon scheme will perform better than the static scheme. Can we formally establish such a relation? In other words, can we prove that the minimum use

function with the finite horizon scheme is not less than that of the static scheme? Such a result will enable us to guarantee performance with respect to the energy estimate $\tilde{p}(t)$.

In *Algorithm 3*, we execute a clairvoyant optimal control at every time step t. Let us suppose that the initial battery at time $t = 0$ equals $\tilde{b}(0)$ as determined by *Algorithm 2*. Then the first control step is $u(0) = \tilde{u}(0)$. If the actual harvested energy is larger than, or equal to the estimated energy, the initial battery state for the next time step is larger than the estimated $\tilde{b}(1)$. Suppose that the optimal use function satisfies some monotonicity, then we will never have a smaller use function value than in the case of the estimated harvested energy. Such a monotonicity should state that the optimal use function can never increase with (i) increasing initial battery state and (ii) increasing energy harvesting function.

We now show the monotonicity property and based on this, the guarantees we can provide for the finite horizon control algorithm. The proofs for all theorems are in *Appendix* A.

Theorem 2: 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 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 result for the finite horizon control algorithm.

Theorem 3: 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 always larger than the optimal periodic one, *i.e.*, 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 from *Theorem 3*, we can provide the following guarantees: The system will never enter a failure state, and the minimal use function *and* the utility are larger than those obtained with *Algorithm 2*. Note that, as will be shown in *Sec. VI-B*, an estimator that meets the criteria is in fact feasible.

D. Implementation

A natural concern with the proposed finite horizon control scheme is the need to execute the clairvoyant algorithm for each time step. This can be a challenge on resource-constrained embedded processors of typical WSN motes. Therefore, in this section we study the possibility to offload the computational complexity to an offline phase and export a small look-up table which is then used at runtime to compute the use function.

Under the assumption of a static estimate of the harvested energy we can use the finite horizon scheme to identify the current use function $u(t)$ 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.

TABLE I: Non-volatile memory (in floats) for the LUT.

Dataset								Mean
TX2	TX1	CA	MD	MI	OR	ON	AK	515
454	494	478	555	542	580	507	509	

For a given time step, $F(t, \cdot)$ is a function of only the battery. We consider approximating $F(t, \cdot)$ by a piecewise-linear function $\overline{F_t}$, which can be computed offline for each possible battery state, discretized by a certain step (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 look-up table (LUT). As a result, the online algorithm is reduced to very few arithmetic operations. The major concerns with this approach are the loss in optimality due to the approximation, and storage requirements for the LUT. In *Sec. VII* we quantify the loss with real data-sets, and demonstrate that a LUT is an effective and efficient solution with very little runtime overhead and only marginal performance loss. The *non-volatile* storage requirements to store the LUT for each dataset used are listed in *Table I*. Note that these values represent an upper bound, as the step size of 1% can be considered a lower bound.

VI. Algorithm Parameterization

The control scheme introduced requires an estimation of the harvestable 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. Here we briefly review this model, and discuss parameter selection for the proposed control scheme.

A. Energy Estimation Model

To estimate the total solar energy available at a particular geographical location without the need for trace data, we leverage an astronomical model adapted for power subsystem planning of solar energy harvesting embedded systems [8]. It can accurately approximate the average harvestable energy for a given geographical location and time of year [7], [8], [24].

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. VII*.

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 and so obtain the energy estimate $\tilde{p}(t)$, we follow the approach in [7] and dynamically scale the design-time model by a factor α. This eliminates the need to re-compute the design-time model with varying Ω, therewith reducing the computational complexity.

As fully discussed in [7], the scaling factor α is computed as the ratio of the sums of true and estimated energy observed

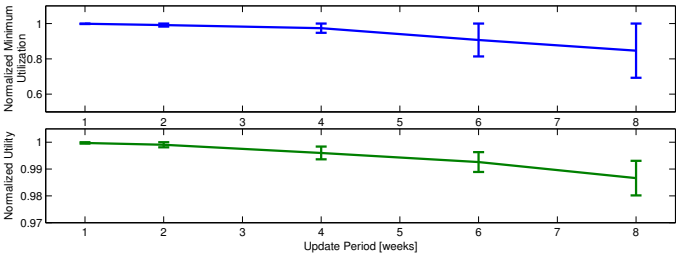

Fig. 2: (top) Average achieved minimum utilization and (bottom) utility for all datasets (normalized to maximum values). Errorbars represent the maximum deviation from the mean for all datasets. Note the different scales.

over a history window of length W weeks. The selection of W presents a trade-off between achievable utility and duty-cycle stability. However, due to our guarantee claims in *Sec. V-C*, we wish to obtain a static estimator that represents true long-term average conditions. Experiments have shown that a history size of $W = 52$ weeks presents a reasonable trade-off between non-volatile memory requirements and achievable performance.

B. Horizon and Frequency of Control

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. V-B* is given $\tilde{p}(t)\ \forall t \in [\tau, \tau + T)$ as estimation input, where $T = 1$ year.

The algorithm requires current and expected final battery fill levels as input, *i.e.*, $b(\tau)$ and $b(\tau + T)$, to compute the optimal energy allocation. We assume the battery to be initially charged to 50% of its capacity, *i.e.*, $b(0) = B/2$, and the expected final fill-level to be equal to the optimal fill-level, *i.e.*, $b(\tau + T) = \tilde{b}(\tau)$, which is found by executing Algorithm 2 with the energy model from *Sec. VI-A* as input.

Frequency of Control. With the energy estimation model and the control horizon established, we now investigate the appropriate duration of the *unit time interval* t for which the algorithm is to compute the supported energy utilization $u(t)$. It is generally desirable to keep the update intervals long so to reduce the algorithm's computational overhead and keep a stable service level. On the other hand, in the case of solar energy harvesting, meteorological conditions cause high variability in the energy profile, which may necessitate shorter intervals, so to allow the algorithm to react to variations.

To find the appropriate update interval, we simulate our algorithm (see *Sec. VII* for details) with $t \in [1, 2, 4, 6, 8]$ weeks and all datasets listed in *Table II*. The achieved average minimum utilization and utility are shown in *Fig. 2* (normalized to their respective maximum values). As is to be expected, shorter update periods allow the algorithm to make most efficient use of the energy, while increasing t adversely impacts both the utility and minimum due to somewhat conservative estimation.

With the goal of enabling competitive long-term minimum performance guarantees, we thus opt for a unit time interval of $t = 1$ week, which is considerably longer than the baseline algorithms that are used to benchmark our approach in *Sec. VII*. Note that shorter update periods work as well, and in fact may result in higher utilization, but also higher variability in the performance level. Moreover, increasing update periods cause a reduction of the variability of the energy profile, which causes it to approach the actual long-term average. This in turn makes the requirement for guaranteeing minimum utilization (see *Sec. V-C*), a realistically achievable condition.

TABLE II: Name, time-period, and location of datasets used for evaluation. P_{max} is the maximum possible consumption, while A_{pv} (in cm^2) is the used panel size. Finally, Ω is the environmental parameter used for the energy estimation.

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

VII. Experimental Evaluation

A. Experimental Setup

Input Data. Due to the limited availability of long-term energy traces from real-world deployments, we leverage the publicly available[2] National Solar Radiation Database (NSRD), from where we acquire 12 years worth of trace data for eight different locations. Additionally, we use two years of energy traces collected at two different locations with our own long-term WSN deployment [4]. See *Table II* for details. The simulation framework conditions the data with efficiency and other system parameters as described in [8].

Baseline Algorithms. We compare our approach against four state-of-the-art implementations of energy-predictive and battery-reactive power management approaches. Specifically, we implement a recently proposed technique that combines power subsystem capacity planning and dynamic power management in an attempt to provide uninterrupted operation with low service-level variability [7], and call it LT-DPM. We further implement the predictive duty-cycling scheme from [17] with two different energy predictors, *i.e.*, EWMA [17] and WCMA [22], and one reactive approach, *i.e.*, ENO-MAX [25]. For each of the baseline implementations we use the authors' recommended parameters, *i.e.*, $K = 3$, $D = 4$, $\alpha = 0.3$ for WCMA [22], and $\alpha = 0.5$ for EWMA [17]. For ENO-MAX [25], 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, but significantly reduces computation complexity. For the approach from [7], we assume daily control updates, *i.e.*, $N_w = 1$, while our approach operates with weekly control updates (see *Sec. VI-B*).

Methodology. To compare the proposed algorithm's performance against the baseline implementations, we leverage the simulation framework from [7], [8]. This framework has been shown to accurately simulate an energy harvesting system as described in [5], and incorporates various efficiency parameters, such as charging (η_{in}) and discharging (η_{out}) efficiencies. Here we consider, $\eta_{in} = 0.9$ and $\eta_{out} = 0.7$, but ignore all other battery inherent inefficiencies. As is further discussed in *Sec. VIII* battery degradation is not a problem: assuming an adequately provisioned power subsystem, *e.g.*, [8], our approach causes the battery to experience a single

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

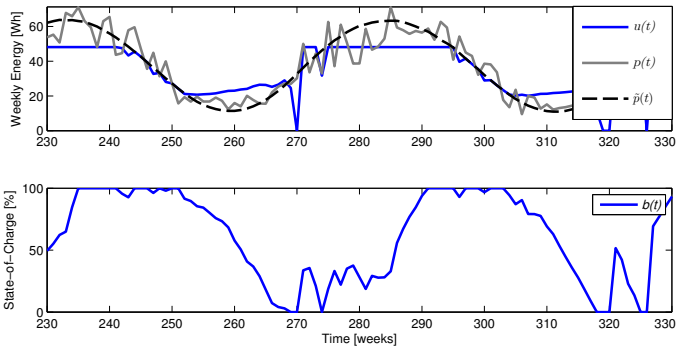

Fig. 3: (top) Use function $u(t)$ computed by OPT for a given harvest function $p(t)$ and the estimator $\tilde{p}(t)$ from *Sec. VI-A* and (bottom) the corresponding stored energy $b(t)$.

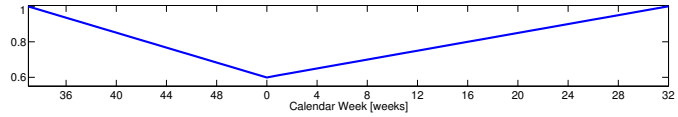

Fig. 4: Empirically obtained piecewise linear scaling function $g(t)$ for 1 year. Note the time-axis.

charge/discharge cycle per year. Considering that a battery is rated for a few hundred cycles, it is expected to outlast other system components, *e.g.*, electronics, mechanical parts, *etc.* We also enforce a low-power disconnect hysteresis of 60% [6], *i.e.*, once the battery is depleted, the load will only be reconnected once the battery has been recharged to 60% of its capacity.

In order to enable a fair comparison, we provide each of the algorithms the same power subsystem, *i.e.*, battery and solar panel, which is obtained according to the technique described in [8]. For this we assume a target duty-cycle $\rho = 40\%$, and a nominal battery capacity of $B_{nom} = 143$Wh to find the appropriate panel size from a set of four sizes, *i.e.*, 5cm^2, 10cm^2, 15cm^2, or 20cm^2, and power level P_{max} that maximizes the resource utilization. Note that the usable capacity $B = \eta_{out} \cdot B_{nom}$. Further note that P_{max} is the maximum power dissipation at full system performance. Thus, the system can consume a maximum of $E_{max} = P_{max} \cdot 24$ hours of energy. This means that the utilization computed by any of the algorithms will be capped at E_{max}. The values used for each of the datasets are listed in *Table II*. Also shown is the environmental parameter Ω that is obtained from profiling the first year of each dataset (see [8] for details).

Performance Metrics. Since we aim at maximizing the minimum service level, we report the *minimum energy utilization* for each of the approaches. Note that outages due to a depleted battery causes the minimum to be 0, and therefore also affects the application utility. As we also wish to maximize the total system utility, we use *(2)* with $\mu(u(t)) = \sqrt{u(t)}$.

B. Experimental Results

In the following we first illustrate that the energy estimator from *Sec. VI-A* is not sufficient, and show how a single non-uniform, piece-wise linear function can alleviate the problem for all datasets. Then we show that the highly resource efficient LUT approach only suffers a negligible performance penalty. Using our real-world trace data, we then compare our algorithm to the next best, *i.e.*, LT-DPM. Then we briefly discuss the characteristics of the baselines, and finally we present the performance results for all algorithms and datasets.

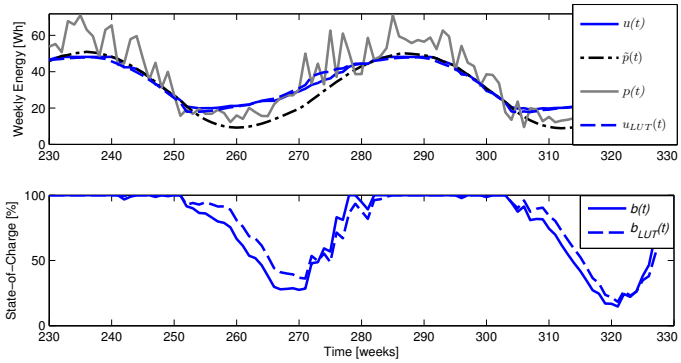

Fig. 5: (top) Use functions $u(t)$ and $u_{LUT}(t)$ computed by OPT with the scaled estimator $\tilde{p}(t)$ discussed in *Sec. VII-B1*, and with the LUT approach, respectively. (bottom) The corresponding stored energy $b(t)$ and $b_{LUT}(t)$.

1) OPT with Energy Estimation Model: Having shown the optimality of the algorithm with the clairvoyant approach both by example and analytically, here we investigate the performance of the proposed algorithm with the realistic energy estimator proposed in *Sec. VI-A*.

Fig. 3 shows the simulation results with years 6 and 7 of the MI dataset (see *Table II*), and the energy estimator discussed in *Sec. VI-A*. We first note the obvious clipping due to P_{max} as discussed in *Sec. VII-A*. As expected, when the battery is full, the OPT algorithm follows the input, *i.e.*, $\tilde{p}(t)$, until the net expected energy generation is less than the consumption. At that point OPT is appropriately pessimistic, but grows more optimistic in an attempt to fully leverage the battery as the winter progresses. The conservative increase in utilization is of course desirable behavior. However, as is visible in the figure, a small negative deviation from the estimation can be fatal, particularly at times when the battery reserves are low and therefore may not suffice to make up for the mis-prediction.

By closer inspection with different data sets, we find that the estimation must be more pessimistic to bridge critical periods, *i.e.*, when the battery is nearly empty. There are two intuitive approaches to achieve this: (i) force higher utilization in fall/early winter, or (ii) force lower utilization in late winter/early spring. Solution (ii) makes more sense for two reasons. First, it is hard to know how optimistic the system can be in the beginning of winter, and second, for maximizing the minimum utilization, it is less critical to throttle at a point where the utilization is significantly higher than the minimum observed (which happens at the beginning of winter). This reasoning is validated when we consider the perfect utilization achieved with the clairvoyant algorithm for the same trace in *Fig. 1*, which shows that the utilization could be higher in the beginning of the winter, but should be more conservative at the end of the winter and well into spring.

From the preceding discussion we conclude that the energy estimation from *Sec. VI-A* is too optimistic, and should be scaled by a non-uniform scaling function $g(t)$ across different weeks, as exemplified in *Fig. 4*. While likely not optimal for every conceivable situation, the general tendency has been empirically verified to fit very well for all 10 datasets we evaluated. The result of using this scaling function on the estimator with the same dataset is shown in *Fig. 5*. We note that the minimum utilization (ignoring the low-power outage) is reduced by only 4% from *Fig. 4* to *Fig. 5*, as the minimum occurs in the beginning of winter, where $g(t)$ is close to 1.

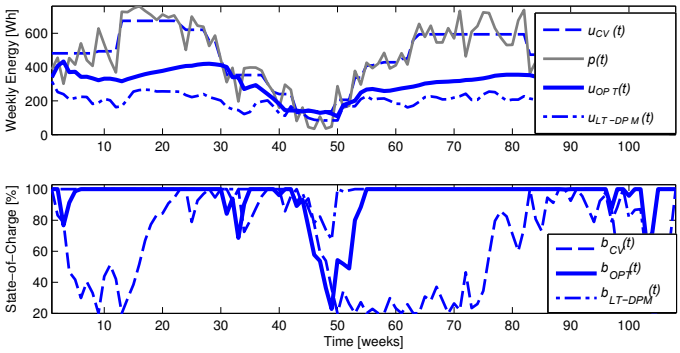

Fig. 6: (top) Use functions computed with clairvoyant ($u_{CV}(t)$), optimal ($u_{OPT}(t)$), and LT-DPM ($u_{LT-DPM}(t)$) approaches. (bottom) The corresponding stored energy $b(t)$.

In summary, an appropriate runtime energy estimation model consists of the energy estimator from [7] reviewed in *Sec. VI-A*, and scaled by a statically defined, non-uniform piece-wise linear function $g(t)$. With this approach we can satisfy the objectives of maximizing the long-term minimum achievable utilization, and maximized total utility.

2) OPT with Look-up Table: In this section we discuss the performance of the look-up-table (LUT) implementation discussed in *Sec. V-D*. Note that we pre-computed the LUT using the piecewise linearly scaled estimator from the previous subsection. At runtime, the system indexes the LUT with the current time t and battery fill-level $b(t)$ and performs a few arithmetic operations to compute the corresponding utilization. The utilization $u_{LUT}(t)$ for the MI dataset is shown along-side the computationally more complex explicit solution. We note that both implementations perform very similarly. However the LUT tends to suffer a small penalty in minimum utilization by being too conservative early in winter (see also *Sec. VII-B5*).

3) Comparison to LT-DPM: In this section we compare our algorithm to the algorithm from [7], to which we refer as LT-DPM. We discuss it separately from the other baseline algorithms, as it is the only algorithm we are aware of that is specifically designed for long-term uninterrupted operation of solar energy harvesting systems at a stable and deterministic minimum service level. *Fig. 6* shows the results for the real-world dataset: DH in *Table II*. See *Sec. VII-B5* for a complete discussion of the results for all datasets and algorithms.

Fig. 6 shows the use functions computed with LT-DPM $u_{LT-DPM}(t)$, the optimal explicit algorithm $u_{OPT}(t)$, and for reference the clairvoyant approach $u_{CV}(t)$, along with the input energy $p(t)$ and the corresponding battery states.

We first note that LT-DPM exhibits remarkably low utilization variance that is maintained throughout the year. This stems from the fact that this algorithm takes a long-term approach to compute the utilization 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 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. From the battery profile shown on the bottom axis it is obvious that the battery is full for most of the summer, yet the algorithm neither leverages the battery nor surplus energy by increasing the utilization. This stems from a fundamentally different approach, which attempts to ensure that the battery is at full capacity at the beginning of winter, at the cost of very conservative utilization during summer, and therefore reduced total system utility. In comparison, the OPT algorithm is able to leverage the surplus energy during summer, while ensuring that the battery is full at the appropriate time, and so ensure continuous operation.

Finally, we note that the minimum service-levels achieved by LT-DPM and our approach in this particular example are roughly equivalent during winter. In fact, on the period $t \in [250, 270]$ and $t \in [304, 320]$, OPT improves over LT-DPM's performance by 0.9% and 2.8% respectively. However, when considering the entire period shown, our algorithm achieves roughly 12.5% higher minimum utilization than LT-DPM. This improvement is mainly due to LT-DPM's pessimistic behavior in summer, where the use function tends to be lowest, despite the energy surplus that keeps the battery at full capacity.

In summary, the comparable utilization during winter can be mainly attributed to the fact that both algorithms use a very similar estimation model, and are able to leverage the available battery capacity. LT-DPM's very pessimistic approach that fails to leverage surplus energy during summer, significantly hurts its overall performance. The service-level achieved by the OPT algorithm, on the other hand, is constrained only by the approximation accuracy of the estimator $\tilde{p}(t)$, and can maximize both the minimum utilization and utility simultaneously.

4) Baseline Algorithms: Next we discuss the characteristics of the remaining three algorithms used for benchmarking our approach, *i.e.*, EWMA [17], WCMA [22], and ENO-MAX [25]. A performance comparison is discussed in *Sec. VII-B5*.

Recall that we parameterized each of the algorithms with the author's recommended, and arguably best parameters. Furthermore, they are all given the same adequately provisioned power subsystem for simulation. 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.

EWMA. This approach uses a simple prediction scheme based on an Exponentially Weighted Moving Average of past observations, and an energy allocation scheme that attempts to continually correct mis-predictions by increasing or decreasing the energy usage in future slots [16]. This algorithm is frequently used as baseline for comparison of energy prediction and allocation algorithms, and performs very well with little overhead. However, this algorithm is designed to predict energy on a horizon of a few minutes to hours, and hence will not be able to make full use of the allocated battery capacity. In fact, the algorithm may reduce the utilization as low as ρ_{min}, despite the battery being full. Here, ρ_{min} specifies the user defined minimum acceptable utilization.

WCMA. This approach has been shown to outperform the prediction accuracy of EWMA, *e.g.*, [2]. However, this does not translate into significant performance improvements with the energy allocation algorithm from [17]. This approach is significantly more computationally involved than EWMA, but behaves very similarly. Just like EWMA, it also requires a system that is capable of measuring (or somehow approximating) the energy generated by the panel.

TABLE III: Minimum daily energy utilization (Min) and total utility (μ) for the clairvoyant optimal algorithm (CV), the optimal algorithm with the energy estimator from *Sec. VII-B1* (OPT), the look-up table implementation (LUT), and the baselines LT-DPM, ENO-MAX, EWMA, and WCMA. Values in parentheses are for $\rho_{min} = 30\%$.

		Dataset									
		TX2	TX1	CA	MD	MI	OR	ON	AK	DH	GG
CV	Min	3.62	4.19	3.76	3.88	3.04	2.63	2.76	1.58	19.65	17.61
	μ	3235.2	3625.4	3439.4	3603.1	3453.6	3349.5	3612.3	3070.9	2174.9	2145.1
OPT	Min	2.86	3.76	3.42	3.38	2.73	2.36	2.32	1.22	15.47	14.86
	μ	3026.1	3399.9	3244.1	3407.5	3272.0	3139.6	3210.0	2336.4	1798.5	1530.7
LUT	Min	3.01	3.71	3.01	3.03	2.53	2.35	2.30	1.21	15.4	14.72
	μ	3023.7	3474.1	3177.0	3437.7	3278.2	3086.4	3201.7	2320.9	1790.1	1516.2
LT-DPM	Min	3.00	3.66	2.98	3.06	2.4	2.04	2.19	1.09	11.9	13.99
	μ	2854.5	3082.6	2857.8	2806.5	2549.1	2380.5	2385.8	1727.4	1502.3	1342.38
ENO-MAX	Min	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)	– –	– –
	μ	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	Min	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)	– –	– –
	μ	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	Min	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)	– –	– –
	μ	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)	– –	– –

ENO-MAX. ENO-MAX uses well-established control theory to determine the utilization that will result in maximum energy efficiency, while ensuring Energy Neutral Operation (ENO), *i.e.*, consumption is always less than generation. More specifically, this algorithm adjusts the utilization such that a predefined battery threshold B_{target} 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 B_{target} may not be maintained and therefore leads to high variability. Moreover, the aggressive downscaling when $b(t) < B_{target}$ results in minimum utilization that follows a user-specified minimum accepted level ρ_{min}. On the other hand, the ability to maximize the energy usage when $b(t) > B_{target}$, leads to high overall utility.

The ability to satisfy ρ_{min} depends on proper selection of B_{target} and ultimately on the proper battery and panel size. In fact, we believe that the main weakness of this approach is the use of a static threshold value. The algorithm could perform much better with a dynamically computed setpoint, possibly based on the same energy model that is used in this paper.

5) Performance Comparison: Here we discuss the performance results listed in *Table III*, as obtained with the proposed approach, and the previously introduced algorithms and all datasets listed in *Table II*. Note that for EWMA, WCMA, and ENO-MAX, we report the result for both a pessimistic $\rho_{min} = 1\%$ (as is done in *e.g.*, [17], [25]), and, listed in parentheses in *Table III*, a more realistic (for our scenario) value of $\rho_{min} = 30\%$. Recall that the power subsystem is designed with an expected duty-cycle $\rho = 40\%$).

First, as expected, EWMA and WCMA perform comparably. This is not surprising, as the same energy allocation scheme is used. Hence, the difference stems from differing prediction accuracies. In fact, WCMA achieves a better minimum utilization 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 minimum utilization 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 ENO-MAX algorithm has a similar limitation. Although achieving the highest total utility of all algorithms, the minimum utilization 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 minimum utilization obtained with these three algorithms to their battery agnostic nature.

Next, LT-DPM achieves significantly higher minimum utilization than the other baseline algorithms. In the best case, LT-DPM outperforms ENO-MAX (with $\rho_{min} = 1\%$) by 3660%. In the worst case, the minimum service-level achieved by LT-DPM is only approximately 0.7% better than that of the baselines with $\rho_{min} = 30\%$. While the achieved utility is generally higher than EWMA and WCMA, it lags behind ENO-MAX. At worst, the utility achieved by LT-DPM is 32% below ENO-MAX, and at best 64% higher than EWMA.

The explicitly computed OPT algorithm outperforms all approaches in terms of minimum service-level, except with the TX2 dataset, for which it performs only 4% worse than LT-DPM, but still better than all other baselines. In fact, at best, the proposed approach comes to within 9.9% of the clairvoyant approach (CV), and 29.5% at worst. The utility achieved by OPT is only surpassed by ENO-MAX, which exhibits significantly lower minimum utilization, and quite considerable variability.

Finally, the look-up table (LUT) implementation performs

negligibly worse than the explicit approach. This demonstrates that with our approach the computational effort can be offloaded to design-time, and achieve very good performance at the cost of a LUT look-up and a few arithmetic operations.

VIII. Limitations and Requirements

Sec. V-D detailed the computational overhead of our solution. Here we briefly discuss other practical considerations.

Energy Estimation. It is clear that the performance achievable by the algorithm is ultimately limited by the estimation. Finding an appropriate energy estimation model that is neither to 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]. Nevertheless, the energy model used in this paper allows the algorithm to achieve a minimum performance level to within 9.9% of the optimal (with clairvoyant estimator) at best, and 29.5% at worst. Similarly, the application utility achieved with our approach reaches the optimal utility to within 5.5% at best, and 31.4% at worst. While the estimation model, and particularly the scaling function may be fine tuned for the different datasets to improve these results, we opted to use one model for all datasets.

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 power output by 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 far from trivial [3].

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 [3]. In our simulation, *charging* and *discharging* efficiencies are incorporated through η_{in} and η_{out} respectively, specified by the system designer. *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 [3]. 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, *etc.*Solar *panel degradation* has been shown to generally be aesthetic in nature, and not significantly affect the panel's efficiency [12].

IX. Conclusion

We have shown that our proposed 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 utilization. Using eight synthetic, and two real-world datasets we show that our approach significantly outperforms four baseline implementations in terms of minimum utilization *and* total utility. Finally, our algorithm is applicable to any harvesting source, presuming an appropriate energy estimator is available.

Acknowledgment

This work was scientifically evaluated by the SNSF and financed by the Swiss Confederation and by Nano-Tera.ch.

References

[1] C. Arnhardt et al. 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*, 10:75–88, 2007.

[2] C. Bergonzini et al. Algorithms for harvested energy prediction in batteryless wireless sensor networks. In *Advances in sensors and Interfaces, 2009. IWASI 2009. 3rd International Workshop on*, pages 144–149. IEEE, 2009.

[3] H. J. Bergveld. *Battery management systems: design by modelling*. PhD thesis, Enschede, June 2001.

[4] J. Beutel et al. X-Sense: Sensing in extreme environments. In *Design, Automation & Test in Europe Conference & Exhibition, 2011*, pages 1–6. IEEE, 2011.

[5] B. Buchli et al. Demo abstract: Feature-rich platform for WSN design space exploration. In *Information Processing in Sensor Networks, 2011 10th International Conference on*, pages 115–116, April 2011.

[6] B. Buchli et al. Battery state-of-charge approximation for energy harvesting embedded systems. In *Proc. of 10th European Conference on Wireless Sensor Networks*, pages 179–196. Springer, 2013.

[7] B. Buchli et al. 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, 2014.

[8] B. Buchli et al. Towards Enabling Uninterrupted Long-Term Operation of Energy Harvesting Embedded Systems. In *Proc. of 11th European Conference on Wireless Sensor Networks*, Oxford, UK, Feb 2014. Springer Link, Lecture Notes on Computer Science.

[9] M. Buzzi. *Challenges in operational numerical weather prediction at high resolution in complex terrain*. PhD thesis, Diss., Eidgenoessische Technische Hochschule ETH Zuerich, Nr. 17714, 2008.

[10] M. Chang et al. Meeting ecologists' requirements with adaptive data acquisition. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 141–154. ACM, 2010.

[11] S. Chen et al. Finite-horizon energy allocation and routing scheme in rechargeable sensor networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 2273–2281. IEEE, 2011.

[12] P. Corke et al. Long-duration solar-powered wireless sensor networks. In *Proceedings of the 4th workshop on Embedded networked sensors*, pages 33–37, New York, NY, USA, 2007. ACM.

[13] J. Dave et al. Computation of Incident Solar Energy. *IBM Journal of Research and Development*, 19(6):539–549, 1975.

[14] L. Guibas et al. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2(1-4):209–233, 1987.

[15] D. Heinemann et al. Forecasting of solar radiation. *Solar energy resource management for electricity generation from local level to global scale. Nova Science Publishers, New York*, 2006.

[16] A. Kansal et al. An environmental energy harvesting framework for sensor networks. In *Low Power Electronics and Design, 2003. Proceedings of the 2003 International Symposium on*, pages 481–486.

[17] A. Kansal et al. Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems*, 6(4):32, 2007.

[18] W. H. Kwon et al. *Receding horizon control: model predictive control for state models*. Springer, 2006.

[19] T. N. Le et al. Power Manager with PID controller in Energy Harvesting Wireless Sensor Networks. In *Green Computing and Communications, 2012 IEEE International Conference on*, pages 668–670. IEEE, 2012.

[20] F. Li et al. Rubberband algorithms for solving various 2D or 3D shortest path problems. In *Computing: Theory and Applications, 2007. International Conference on*, pages 9–19. IEEE, 2007.

[21] F. Li et al. *Euclidean Shortest Paths*. Springer, 2011.

[22] J. R. Piorno et al. 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*, pages 6–10. IEEE, 2009.

[23] P. Sommer et al. The big night out: Experiences from tracking flying foxes with delay-tolerant wireless networking. In *Real-World Wireless Sensor Networks*, pages 15–27. Springer, 2014.

[24] J. Taneja et al. Design, Modeling, and Capacity Planning for Micro-solar Power Sensor Networks. In *Proc. of the 7th international conference on Information processing in sensor networks*, pages 407–418, Washington, DC, USA, 2008. IEEE Computer Society.

[25] C. M. Vigorito et al. 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*, pages 21–30. IEEE, 2007.

[26] P. Zhang et al. Hardware design experiences in ZebraNet. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 227–238, New York, NY, USA, 2004. ACM.

APPENDIX

A. Proofs of Lemmas

Proof of Lemma 1: We first show that

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

We show that a use function $u(\tau)$ that does not satisfy the above criterion can be replaced by a use function $u'(\tau)$ with a higher minimum used energy. The change will be local, *i.e.*, it will only influence two values of $u(\tau)$ in $s-1 \leq \tau \leq t$, *i.e.*, $u(\tau_1)$ and $u(\tau_2)$. Moreover, we will guarantee that $\sum_{\tau=s-1}^{t} u(\tau) = \sum_{\tau=s-1}^{t} u'(\tau)$ and therefore, the stored energy functions will satisfy $b'(\tau) = b(\tau)$ for all $0 \leq \tau \leq s-1$ and $t+1 \leq \tau \leq T$. This is achieved by setting $u(\tau_1) + u(\tau_2) = u'(\tau_1) + u'(\tau_2)$.

The change of some values of $u(\tau)$ must not violate the conditions on the stored energy. The maximum allowed variation on $u(\tau_1)$ and $u(\tau_2)$ is $\epsilon = \min_{s \leq \tau \leq t}\{|b(\tau)|, |B - b(\tau)|\} > 0$

As $u(\tau)$ is not constant in $s \leq \tau \leq t$, there exists a maximum and a minimum $u(\tau)$ in $s \leq \tau \leq t$, say $u(\tau_1)$ and $u(\tau_2)$, respectively with $\tau_1 \neq \tau_2$. We now choose

$$u'(\tau_1) = u(\tau_1) - \delta \qquad u'(\tau_2) = u(\tau_2) + \delta$$

with

$$\delta = \min\left\{\frac{u(\tau_1) - u(\tau_2)}{2}, \epsilon\right\} > 0$$

We finally have to show that this change leads to a higher minimum used energy, and hence $u(\tau)$ was not optimal.

Now we show that

$$u^*(s-1) < u^*(s) \Rightarrow b^*(s) = 0,$$
$$u^*(t-1) > u^*(t) \Rightarrow b^*(t) = B.$$

From the earlier condition, we know that $u^*(\tau - 1) \neq u^*(\tau) \Rightarrow b^*(\tau) = 0 \vee b^*(\tau) = B$. Suppose now that there is some use function with $u(s-1) < u(s)$ and $b(s) = B$. Then we can use the same method as in the proof of the first part of Lemma 1 to show that in this case, u was not optimal. To this end, we replace the value of u at $s-1$ by the larger value $u(s-1) + \delta$ and that at s by the smaller value $u(s) - \delta$ by choosing a suitably small positive value of $\delta \leq (u(s) - u(s-1))/2$ which does not lead to a failure or waste state. The stored energy at time s is now $B - \delta$ which is still feasible, but the overall utilization would be larger. Therefore, u was not optimal. The proof for the second relation is analogous. ■

Proof of Lemma 2: Let us suppose that there are two different feasible solutions u_1 and u_2 that satisfy the conditions of Lemma 1. At time $\tau = 0$, the stored energy is given as $b(0)$, both use functions result in the same given stored energy at the end of interval $b(T)$, and no use function leads to a failure or waste state.

As u_1 and u_2 are different, there is some first time instance $\tau = s$ where both functions are different, e.g. $u_1(s) > u_2(s)$. As the total harvested energy equals the used energy (no waste state) we have

$$\sum_{\tau=0}^{T-1} u_1(\tau) = \sum_{\tau=0}^{T-1} u_2(\tau).$$

Therefore, there must exists a time $\tau = t > s$ where $u_1(t) < u_2(t)$. We will show that such a time instance can not exist under the conditions of the Lemma.

Let us suppose that t is the first time instance with $u_1(t) < u_2(t)$, *i.e.*, $u_1(\tau) \geq u_2(\tau)$ for all $0 \leq \tau \leq t-1$. The stored energy under the use function u_i is denoted as b_i. From $b_i(\tau+1) = b_i(\tau) + p(\tau) - u_i(\tau)$ and the existence of time s we find that $b_1(t) < b_2(t)$.

In order to realize such a situation, we either have $u_1(t-1) > u_1(t)$ or $u_2(t-1) < u_2(t)$ or both conditions hold. We will show that none of the conditions is feasible. Suppose we have $u_1(t-1) > u_1(t)$. Because of Lemma 1 we find $b_1(t) = B$. But this is not possible as $b_1(t) < b_2(t)$ and $b_2(t) < B$. Suppose we have $u_2(t-1) < u_2(t)$, then we find $b_2(t) = 0$, again due to Lemma 1. But this is not possible as $b_1(t) < b_2(t)$ and $b_1(t) > 0$. ■

Proof of Theorem 2: We use a constructive proof principle. We are given the optimal use function $u^*(t)$ before applying any change to ($b(0)$, B, $p(t)$ for $0 \leq t < T$, $b(T)$). Then we can construct an adapted feasible use function for the new input parameters that has no smaller use function values than $u^*(t)$. The optimal use function for the changed input parameters can only be better.

If $b(0)$ increases by some value Δ, then we replace $u^*(0)$ by $u^*(0)+\Delta$. All other values $u^*(t)$, $t > 0$, remain unchanged. If $b(T)$ decreases by some value Δ, then we replace $u^*(T-1)$ by $u^*(T-1) + \Delta$. If B increases by some value, then $u^*(t)$ is still feasible, though not optimal in general. If $p(t)$ increases for some t by some value Δ, then we replace $u^*(t)$ by $u^*(t) + \Delta$ and the battery state at time $t+1$ remains the same as before. ■

Proof of Theorem 3: At every time step t when we execute Algorithm 3, we can compare the situation to the optimal periodic case with the estimated energy harvesting function. We use an inductive argument. Initially, the battery state is larger than that of the optimal periodic case as $b(0) \geq \tilde{b}(0)$. Suppose now we are at time t and the current battery state satisfies $b(t) \geq \tilde{b}(t)$. Then we will show that the determined use function is larger or equal than $\tilde{u}^*(t)$ and that $b(t+1) \geq \tilde{b}(t+1)$. The finite horizon control determines at time t a use function based on the horizon $[t, t+T]$ and the estimated harvested energy in this interval. As the final battery state at time $t+T$ equals that of the optimal periodic case and as the initial battery state is higher, the determined use function is larger than $\tilde{u}^*(t)$. The battery state at time $t+1$ is larger or equal due to two independent reasons. At first, the actual harvested energy in $[t, t+1)$ is larger or equal than the estimated one. Moreover, a larger initial battery at t leads to a larger battery at the next time step $t+1$ even in the case of equal harvested energy due to the triangle inequality in the equivalent shortest path problem. ■