Received: 8 February 2021 | Revised: 12 July 2021 | Accepted: 15 July 2021

IET Computers & Digital Techniques

DOI: 10.1049/cdt2.12033

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

EmRep: Energy management relying on state-of-charge extrema prediction

Lars Hanschke[1] | **Christian Renner**[2]

[1]Research Group smartPORT, Hamburg University of Technology, Hamburg, Germany

[2]University Koblenz - Landau, Koblenz, Germany

Correspondence

Lars Hanschke, Research Group smartPORT, Hamburg University of Technology (TUHH), Am Irrgarten 3-9, Building Q, Room 1.035, D-21073 Hamburg, Germany.
Email: larshanschke@googlemail.com

Funding information

Open access funding enabled and organized by Projekt DEAL

Abstract

The persistent rise of Energy Harvesting Wireless Sensor Networks entails increasing demands on the efficiency and configurability of energy management. New applications often profit from or even require user-defined time-varying utilities, for example, the health assessment of bridges is only possible at rushhour. However, monitoring times do not necessarily overlap with energy harvest periods. This misalignment is often corrected by over-provisioning the energy storage. Favourable small-footprint and cheap energy storage, however, fill up quickly and waste surplus energy. Hence, EmRep is presented, which decouples the energy management of high-intake from low-intake harvest periods. Based on the State-of-Charge extrema prediction, the authors enhance energy management and reduce saturation of energy storage by design. Considering multiple user-defined utility profiles, the benefits of EmRep in combination with a variety of prediction algorithms, time resolutions, and energy storage sizes are showcased. EmRep is tailored to platforms with small energy storage, in which it is found that it doubles effective utility, and also increases performance by 10% with large-sized storage.

KEYWORDS

embedded system, energy consumption, hardware-software codesign

1 | INTRODUCTION

Over the past years, energy harvesting (EH) has been a vital competitor for powering Wireless Sensor Networks (WSNs) in many applications. Devices powered by ambient energy—for example, solar EH [1], vibration harvesting [2] or temperature difference harvesting [3]—allow uninterrupted operation without human intervention. This is especially important in critical places, for example, bridges [2, 4] and the Matterhorn [5], where access is time consuming and costly. However, the varying nature of ambient energy entails careful management of activity, for example, as presented in [6–8] to use energy efficiently. The evolving field of Intermittent Computing (IC), that is, devices that greedily use available energy until shutdown, poses new challenges on rethinking hardware [9], program design [10, 11], and energy management [12, 13]. Still, their operation is tied to sufficient harvest, so the usage of IC is limited to applications where harvest and times of interest overlap.

Hence, Energy-Neutral Operation (ENO) devices—which bridge phases of low energy intake—are indispensable in applications where uninterrupted operation (depletion safety) is mandatory, such as bridge health monitoring [2]. Here, Vibro-acoustic Modulation (VAM) [4] and others require external vibrations by trains or cars; hence, measurements are only possible or meaningful at specific times of the day. This leads to time-varying utility which may not correlate with energy harvest and has to be incorporated into energy management. Literature [8] knows methods to model year-around utility but the performance on a day-basis is sub-optimal (c.f. Section 5). Here, small storage and uncertain harvest estimation may lead to unexpected shutdowns. Traditional energy management, however, on day-basis [7, 14], does not yet model time-varying utility.

As stated in Ref. [15], energy-neutral computing devices can provide high reliability and utility, since they balance energy intake and expenses over a certain time window, for example, a

year or a day. In phases of energy surplus, the storage device is charged up to the limit, and the stored energy is used in deficit phases to provide a stable level of activity throughout the window. In contrast to energy management techniques such as AsTAR [6], which uses energy until depletion, ENO devices allow high utility when harvest is low. If the energy surplus is greater than the storage capacity, however, the storage device saturates and is cut-off to prevent over-charging. Literature [16–18] has shown that—even potentially erroneous—adaption to future local harvest conditions is indispensable to optimise activity. Estimation-agnostic approaches boil down to power-neutral computing [15], which spends energy when it is available. Harvest estimation bears additional risks: if harvest is over-estimated, activity needs to be reduced, which possibly introduces fluctuating duty-cycles. If the future harvest is under-estimated, the problem of storage saturation is exaggerated—to save energy for the night, the energy surplus is wasted since the presence is better than expected. Even if the application utility could have been improved, current ENO devices lose the opportunities of high-intake phases because depletion in low-intake phases is feared, c.f. Figure 1. Here, a depletion-safe consumption for the next day is calculated. A consumption slightly higher leads an SoC below the minimum at the end of a low-intake phase; hence, the average consumption for the next day cannot be increased. This, however, misses the opportunities of higher activity in the high-intake phase since maximum harvest is well above the average consumption. The energy storage, hence, saturates since the node unnecessarily limits its activity.

We argue that the requirements of modern applications—also including industrial context—demand a new way of managing energy for ENO devices in EH-WSNs. Hence, we present EmRep,[1] which integrates utility into energy management. EmRep uses SoC estimation to identify low-intake and high-intake phases, and then decouples energy management in these two phases. This allows to utilise the surplus energy in high-intake phases—if it increases the application-defined utility—without sacrificing depletion safety in low-intake phases. In particular, we

- Integrate time-varying utility into energy management for ENO systems;
- Exploit energy surplus, which lowers storage saturation by up to 32% without increasing the risk of depletion;
- Increase performance across exemplaric application-specific utility profiles;
- Enable downsizing of storage size even with solar harvest estimation; and
- Show a case study for VAM in which EmRep increases the performance by 80%.

Next, we showcase the challenges faced in EH-WSNs, highlight related work and present a detailed problem analysis.

[1]EmRep is an (**e**nergy)-analogy to the workout type **AMRAP**, where the goal is to execute **A**s **M**any **R**epetitions **A**s **P**ossible.

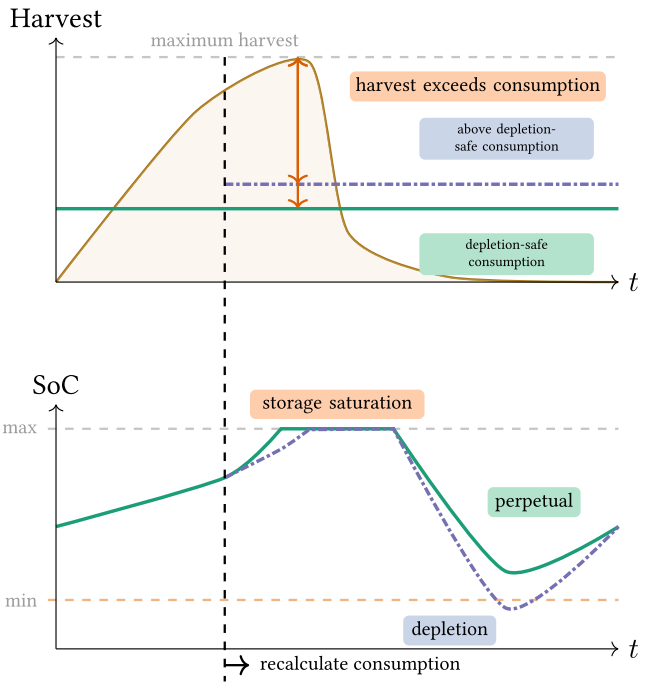
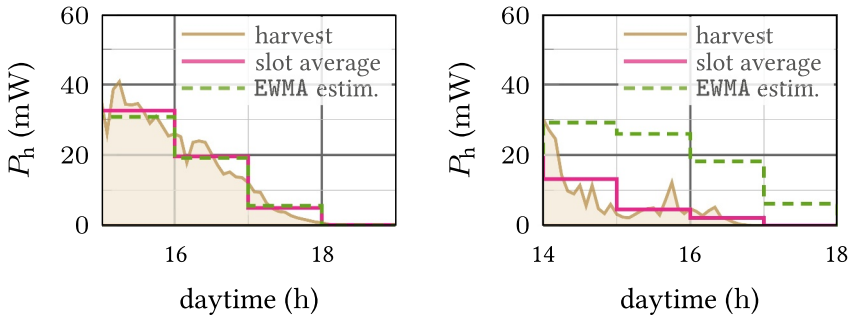


FIGURE 1 Illustration of saturation problem for small-sized energy storage; when consumption is recalculated, the desired average consumption is significantly lower than the maximum harvest; however, the average consumption cannot be increased since depletion in low-intake phase is imminent; performance in high-intake phase is sacrificed for depletion-safety

2 | BACKGROUND & RELATED WORK

2.1 | Challenges of energy harvesting

The varying nature of ambient energy requires careful adaption to the currently harvested energy. Sources such as the solar source allows to learn prevalent patterns, for example, diurnal patterns influenced by trees or buildings. For solar harvesting, a plethora of algorithms evolved, for example, EWMA, WCMA [17], Pro-Energy [16] and Delta-T [19]. Although all models only *estimate* the future, and hence not being totally accurate, they improve the performance of ENO drastically. As also identified by the authors of Ref. [20], intake estimation is of utmost importance to increase the efficiency for EH-WSN. Harvest estimation models typically divide the day into slots with average harvest at a fixed resolution, that is, time slots that typically last between 30 min and 240 min. This entails problems, since average values are not able to replicate the trace exactly, and over- and under-estimation lead to performance degradation. We highlight the average replication problem in Figure 2, where we compare the real harvest to slot average and EWMA estimation. On days which are very similar to the learnt pattern, the estimation error is low—c.f. Figure 2a—whilst the error can be very high, when harvest conditions change rapidly—c.f. Figure 2b. Even correct slot averages can be misleading, since they do not take harvest imbalance within the slot into account. This is especially severe in early mornings



(a) Low estimation error. **(b)** High estimation error.

FIGURE 2 Although being inevitable for activity planning, harvest estimation may be inaccurate; here, we highlight the effects of slot averaging on harvest estimation; the slot duration is 60 min; even the real slot average cannot perfectly represent the harvest trace; the harvest trace is at 5 min resolution

where the SoC in solar-powered systems is near the minimum operating level. If the harvest lies well above the average on the start of the timeslot, the storage can already fill up, and the risk of depletion subsides. The opposite holds if harvest arrives late in the time slot; although this distribution has the *same* average, energy is taken from the storage which has not been harvested yet; hence, the risk of depletion increases. Nevertheless, energy management without harvest estimation and anticipatory buffering boils down to simply reacting on present harvest without optimising the complete horizon. While this approach suffices if activity is bound to high harvest, it fails in phases of low-intake since previously harvested energy was already spent. Hence, times of no to little harvest have to be bridged by preserving energy. Over the past, batteries [21] and supercapacitors [22] have become established energy storage. However, the choice of storage size is non-trivial: large storage allows to survive long low-harvest phases while small storage allows to recharge quickly and keep outages short [23]. Additionally, times of interest are not necessarily aligned with times of harvest; hence, careful energy management is needed to efficiently use present energy. This need increases if time-varying utility is combined with favourable short energy storage.

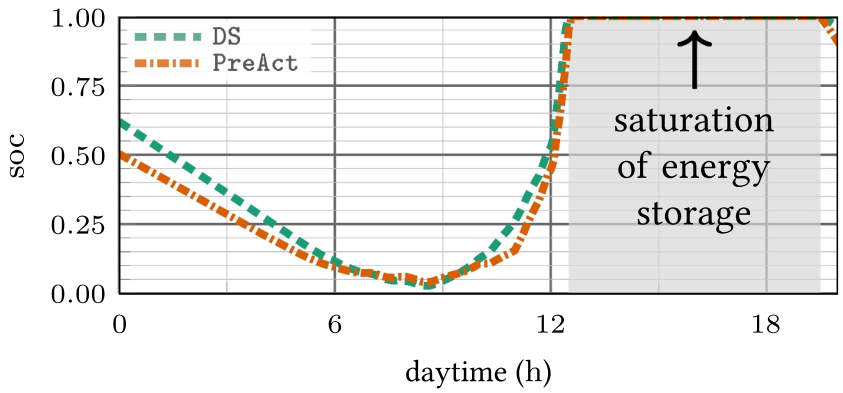
2.2 | Energy management

Since the introduction of the concept of ENO in Ref. [18], several energy management algorithms have been developed. The authors of Ref. [18] aim at finding one duty cycle to ensure the consumed energy does not exceed the harvested energy for a finite horizon, for example, 24 h. This calculation is adapted when an energy surplus or deficit is detected. However, a linear relationship between duty cycle and utility is assumed and no support for varying utility is provided. A linear-quadratic tracking approach is presented in Ref. [24] in which the authors aim to maintain a stable SoC of the battery. Whilst the linear-quadratic tracking outperforms [18] in an average duty cycle and duty cycle variance, it also lacks support for time-varying utility. The task of finding the optimal sensing rate is transformed into a control problem by the authors of Ref. [14]. While the controller generation is performed offline, sensing rate adaption is performed online on the node. However, Ref. [14] neither supports time-varying utility nor does it give implications on practical performance. Even the new approach such as AsTAR [6], which aims to offer lightweight dynamic task rate adaption without the need for prior modelling, limits the use of energy to high-intake phases. AsTAR bounds task execution rates to voltage levels, which allows the node to execute tasks at high rates during high-intake phases, low-intake phases are spent in shut-off states to prevent the depletion of energy storage. This inhibits utility-based scheduling, where high execution rates and high harvest mismatches, and boils down to best-effort scheduling. In contrary, authors of [7] present a lightweight algorithm based on a binary search. The goal is to find the maximum average node power consumption restricted by a given policy. Although the depletion safe policy (DS)—which keeps the SoC above depletion limit—ensures uninterrupted operation and stable duty cycle, the approach lacks support for time-varying utility.

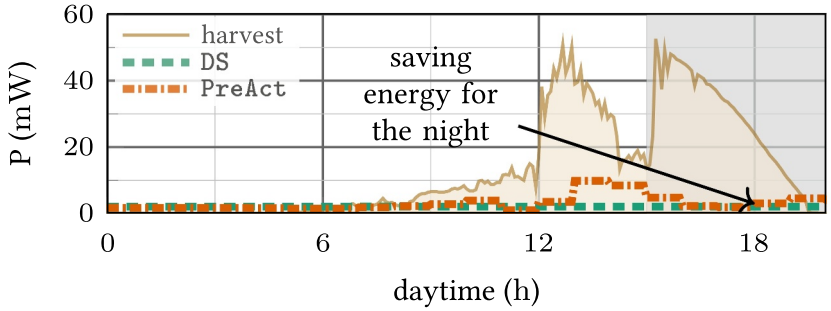
Targetted at long-term sensing (usually year-around), the authors of Ref. [25] present LT-ENO. It incorporates an astronomical model, which maps changes of solar irradiation during the year. However, it assumes uniform utility and aims to maintain a stable duty cycle throughout the year, which is adapted if harvested energy and predicted energy differentiate. The authors of Ref. [8] present PreAct, a long-term energy management algorithm designed for time-varying utility. PreAct outperforms the algorithms presented in Refs. [24, 25], and can be considered as the state-of-the art approach for ENO systems. It uses an underlying PID controller to follow an optimal state-of-charge curve and dynamically correct harvest mismatch. However, PreAct performs sub-optimal with small energy storage, (short-term) control horizon on day-basis and prediction algorithms, c.f. Section 5.

2.3 | Problem analysis

As stated before, up until now, no energy management algorithm exists, which incorporates time-varying utility, and is targeted at small-sized storage and short-termed horizon. Small storage is assumed to charge once per day; hence, the capacity limits the usable energy during times of low-intake. However, even with small solar cells, for example, a 35×35 mm, the harvest during the day often exceeds the node consumption by a magnitude, c.f. Figure 3b. The energy surplus leads to saturation of the energy storage, typically long before the low-intake phases begin. Current energy



(a) Exemplarily: SoC of two common energy management algorithms with 50 F supercapacitor.



(b) Corresponding power consumption shows saving for the low-intake phase; although SoC and harvest are high; `PreAct` tries to compensate; `DS` keeps stable power consumption.

FIGURE 3 Study of saturated energy storage; surplus energy is wasted instead of increasing utility for application; data gathered by the simulation framework in Ref. [27] and based on real-world harvest data collected in Ref. [33]

management schemes, however, aim to maintain the fully charged storage to safely live throughout the low-intake phase. The targetted low-intake node consumption is much smaller than high-intake harvest; hence, the storage is kept saturated. The incoming energy is wasted, if an application profits from a higher node consumption. This deficiency is pinpointed in Figure 1.

Figure 3a showcases one day with different energy management algorithms. The used energy storage is a 50 F supercapacitor with 35×35 mm solar panel. The node uses Pro-Energy [16], a state-of-the-art solar harvesting forecasting algorithm. However, both supplied management schemes, that is, DS [7] and PreAct [8], fail to utilise all the energy in the afternoon: nearly seven hours of high energy availability is wasted. Both schemes do not allow for higher node consumption to prevent depletion during the low-intake phase.

This behaviour adds up, as we highlight in Figure 4. For small-sized storage, for example, a 25 F supercapacitor, up to 19% of the time in our simulations surplus energy is wasted. In total, this adds up to 903 h or an average of 4.65 h/d.

Hence, we design EmRep, which is targeted to mitigate storage saturation and efficiently use harvested energy under time-varying utility.

3 | DESIGN OF EmRep

As highlighted in Section 2.3, we see one dominating problem with the existing EH management schemes: bridging low-intake phases limits the management flexibility in high-intake phases. This entails the saturation of energy storage and hence wastes surplus energy.

3.1 | System model

We employ a system model similar to Ref. [8], which we depict in Figure 5. In this model, we assume the *node* to include microcontroller, radio and other attached hardware such as sensors and actuators. The EH circuit side consists of two essential entities: the harvester and the energy storage. External devices such as Analogue to Digital Converters or coulomb counters gather information about the harvest H throughout a time horizon T_H, and the SoC of the energy storage.

Although EmRep is designed with solar EH in mind, the energy source is exchangeable in general. However, using another source requires that harvest estimation is possible either from local learning or external knowledge (like weather

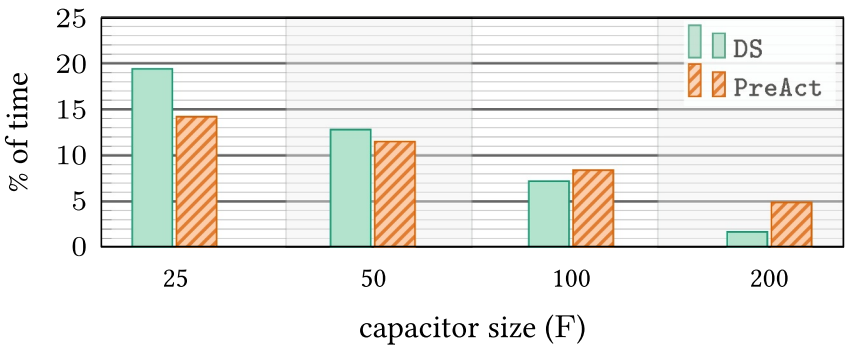


FIGURE 4 Percentage of time spent with full energy storage (saturation); simulation of 194 d; storage is full upto 19.4% of time for 25 F capacitor; incoming energy at the harvester is wasted

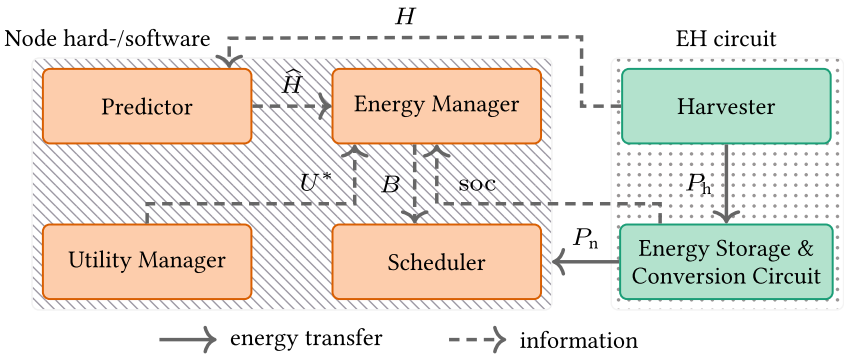


FIGURE 5 Modelling of the energy harvesting system; harvested energy is stored and voltage is converted before used by the node; energy manager joins harvest estimation $\widehat{H}$ and utility profile U^* to calculate energy-aware budget B; budget determines how much energy is used, the scheduler determines when to use this energy; time-dependencies are omitted for readability

forecast). For solar harvesting, this pattern may be influenced by buildings, trees and the angle to the sun. The energy storage entity may be one or multiple supercapacitors [23] or a battery [21]. The generated power at the harvester P_h translates to an energy transfer into the storage. The node is only attached to the storage and consumes P_n from the storage. Both P_h and P_n are influenced by charging and discharging efficiency, which are highly platform-dependent. However, correct modelling has to account for the inefficiencies that have to be obtained prior to simulations.

The transition between the harvest source, energy storage and node is highly platform dependent. For maximum efficiency, the storage can be connected directly to the node and, hence, serves as supply. However, recent literature has shown that a fluctuating supply voltage affects the clock frequency of microcontrollers and hence code execution duration [26]. If conversion losses can be accepted, a boost converter can be used to stabilise supply voltage. Furthermore, more energy from the capacitor can be used since it can be discharged to lower voltage levels. For common supply voltage levels, for example, V_n=3.3 V, the conversion efficiency is usually above 80% [27] so that benefits outweigh. As a consequence, the power demand of a sensor node $P_\mathrm{n}(t)$ simplifies to

$$P_\mathrm{n}(t) = V_\mathrm{n} \cdot I_\mathrm{n}(t), \tag{1}$$

with $I_n(t)$ being the current drawn by the node. This current varies with used peripherals, for example, radio or MCU, and has to be obtained by in-lab measurements [27, 28] or on-line [29]. Over a time interval Δt, $I_n(t)$ usually fluctuates but the literature for energy management [7] has shown that the time average over Δt can be used without sacrificing accuracy.

The software side of the model consists of four entities: predictor, utility manager, scheduler and energy manager. As discussed in Section 1, applications may demand the time-varying use of resources, for example, prefer high sampling rates in the morning. These demands are fed into the energy manager with a utility profile $U^*(s)$. Since ENO requires spent energy to be not greater than harvest energy for a finite time horizon T_H, it is crucial to estimate the harvest intake. For solar energy, numerous algorithms exist, for example, Pro-Energy, WCMA and Delta-T. Such algorithms try to learn prevalent patterns and estimate future harvest H with as low error as possible. This information is used by the energy manager to decide on an appropriate duty cycle $dc(s)$, that is, the ratio between active times and overall time. Please note that EH usually divide the horizon T_H into slots of fixed time, with time slots $s \in 1, \ldots, S$ usually lasting 30–240 min for a diurnal cycle. Given a known active node power consumption P_n, the duty cycle $dc(s)$ can be used to calculate the target power consumption

$$P^*_\mathrm{n}(s) = V_\mathrm{n} \cdot I_n \cdot dc(s) = V_\mathrm{n} \cdot B. \tag{2}$$

In literature, the product of active power consumption of the node and duty cycle is often referred to as budget B. How budget B is spent is determined by the scheduler. Literature knows various techniques, for example, by using program flows [28] or a Lazy Scheduling Algorithm (LSA) [30]. In contrast to schedulers, which decide *when* to spend energy, energy managers decide *how much* energy to spend. Hence,

this work has a different scope than previously published literature on energy-aware scheduling, for example, [27]. EmRep implements the energy management entity and thus merges information about future harvest, current SoC and instructions for the scheduler.

3.2 | Utility

Literature suggests different definitions of the utility; either as a relative metric at which a portion of the overall available energy is distributed. For for example, PreAct implicitly defines the utility as a portion of the mean expected energy intake. A more application-specific approach, for example, as presented in Ref. [31], directly relates the utility to a sampling rate. Since applications often require a specific amount of activity, for example, structural health monitoring with the desired number of measurements per hour, we see the latter approach as more promising. Hence, the achieved utility $U(s)$ is given by

$$U(s) = \frac{B(s)}{\mathrm{B}_{\max}}, \quad s \in 1, \ldots, S \tag{3}$$

with $\mathrm{B}_{\max}$ denoting the maximum achievable budget and the corresponding sampling rate and $B(s)$ the current average node consumption. $\mathrm{B}_{\max}$ is the slot average of the node consumption, which directly translates to the sampling rate if the energy consumption of the sampling activity is known. Correspondingly, we define a utility profile $U^*(s)$, which declares a desired utility throughout the horizon T_{H}. Next, we introduce the main characteristics of EmRep.

3.3 | Principle

As briefly introduced in Section 1, EmRep uses SoC estimation to identify high-intake and low-intake phases, manages energy separately in these phases together aiming at utility maximisation.

3.3.1 | SoC curve

The SoC of harvesters with the supercapacitor depends on the usable energy in the capacitor. Ideally, the energy in the capacitor with nominal capacity C can be calculated by $E^*_{\mathrm{cap}} = \frac{1}{2} \cdot C \cdot {V_{\mathrm{cap}}}^2$. In practice, however, energy from the capacitor below $V_{\min}$ is not feasible, c.f. [32], because the employed boost converter needs a minimum input voltage to provide a stable output voltage at a high load. Hence, the usable energy in the capacitor is

$$E_{\mathrm{cap}} = E^*_{\mathrm{cap}} - E^{\min}_{\mathrm{cap}} = \frac{1}{2} \cdot C \cdot \left({V_{\mathrm{cap}}}^2 - {V_{\min}}^2\right). \tag{4}$$

For the used system model, the SoC is the ratio between E_{cap} and $E^{\max}_{\mathrm{cap}}$, hence is calculated by

$$\mathrm{soc}(V_{\mathrm{cap}}) = \frac{E_{\mathrm{cap}}}{E^{\max}_{\mathrm{cap}}} = \frac{{V_{\mathrm{cap}}}^2 - {V_{\min}}^2}{{V_{\max}}^2 - {V_{\min}}^2}. \tag{5}$$

EmRep uses the SoC estimation of Ref. [7], which solves the underlying differential equation with the Newton's method.

The simplified equivalent circuit, found in Figure 6, allows the authors to simulate the SoC course of the energy storage. The coherence between model, simulation, and real-world performance has already been shown in Refs. [7, 27]. We also use this model in the simulation framework detailed in Section 4.1 for fair and reproducible testing.

When the Sun shines on the solar panel, the harvesting current is $I_h > 0$. The consumption part consists of a boost converter with efficiency $\eta(I_n, V_{\mathrm{cap}})$, which provides the output voltage V_{n}. The sensor node draws a current I_n from the boost converter output; the boost converter draws the current I_{r} at its input. If I_h exceeds I_{r}, the capacitor is charged with current I_{c}. To prevent overcharging, the switch S_{h} disconnects the solar panel whenever $V_{\mathrm{cap}} > V_{\max}$ is detected.

Since the current flowing into the reserve is the difference between harvest current and boost converter current,

$$I_{\mathrm{c}} = I_h - I_{\mathrm{r}} \tag{6}$$

holds. Together with the converter efficiency equation (equal power at in- and output side)

$$I_{\mathrm{r}} \cdot V_{\mathrm{cap}} \cdot \eta(I_n, V_{\mathrm{cap}}) = I_n \cdot V_{\mathrm{n}} \tag{7}$$

and the capacitor model from Ref. [33] giving $I_{\mathrm{c}} = C \cdot \dot{V}_{\mathrm{cap}}$, the final formula for the system model is as follows:

$$C \cdot \dot{V}_{\mathrm{cap}} + I_h = \frac{I_n \cdot V_{\mathrm{n}}}{V_{\mathrm{cap}} \cdot \eta(I_n, V_{\mathrm{cap}})}. \tag{8}$$

Simulating the time-dependent capacitor voltage course with this formula, however, is not feasible due to the additional hardware characteristics. The boost converter is only able to provide a stable output voltage if a capacitor voltage $V_{\mathrm{cap}} > V_{\min}$ is achieved; additionally, $V_{\mathrm{cap}} \leq V_{\max}$ due to the overcharging protection. The solution, however, is possible for discrete time intervals (timeslots) since I_h and I_n can be assumed to be piecewise constant within the timeslot. This results in a first-order differential equation whose solution is approximated with the Newton's method and a binary search algorithm.

The load maximisation algorithm finds the maximum power consumption $P^*_{\mathrm{n}}(s) = I^*_{\mathrm{n}} \cdot V_{\mathrm{n}} = P_{\mathrm{ds}}$ of the node throughout the complete horizon without depleting the energy storage. This behaviour is also shown in Figure 3, where DS maintains a stable power consumption neglecting the energy surplus. EmRep uses this power consumption as the baseline in the DS region. Based on P_{ds}, the SoC curve for the next prediction horizon is calculated. The SoC curve of devices equipped with solar harvesters often follow the trend depicted in Figure 7: a minimum—usually at the end of the low-energy intake phase—and a maximum—during or at the end of the high-energy intake

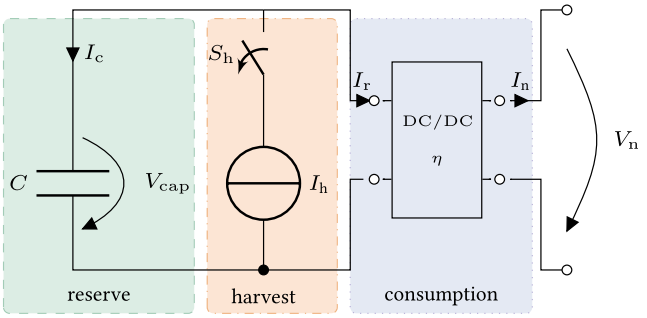


FIGURE 6 Simplified equivalent circuit of the resembled harvester; a supercapacitor serves as the reserve, which is charged by the solar panel (current source); sensor node is supplied via boost converter; model obtained from Ref. [7]

phase. When energy storage charges once per day, two extrema naturally divide the prediction horizon T_H into two regions. In the low-energy region, little or no harvest can be expected; thus, preventing depletion of the energy storage is the key. In the second region, the harvesting intake often exceeds node consumption, which either saturates energy storage or gives more freedom to spend energy. In EmRep, we call these two regions the DS region and the Free region; cf. Figure 7.

Directly after passing t_n^{max}, EmRep estimates the SoC for the next prediction horizon T_H. The employed SoC estimation strategy highly depends on the used platform; for example, if a supercapacitor or regular battery is used. Two two-tuples for the next minimum $(t_{n+1}^{min}, soc_{n+1}^{min})$ and maximum $(t_{n+1}^{max}, soc_{n+1}^{max})$ are stored.

3.3.2 | Regions

In the DS region, the node consumes the previously calculated P_{ds}. Up until t_{n+1}^{min}, EmRep only checks if the target soc_{n+1}^{min} will be reached and slightly lowers/increases P_{ds} if necessary, that is, the SoC estimation reruns until t_{n+1}^{min} if estimation and reality differentiate.

After t_{n+1}^{min} passed, the Free region starts with a new iteration of the algorithm of Ref. [7]; still, not for T_H but for

$$T_{free} = t_{n+1}^{max} - t_{n+1}^{min}. \quad (9)$$

The algorithm decrements the power consumption for the SoC calculation until a consumption is found that (a) maintains an SoC above soc^{min} within T_{free} and (b) reaches soc_{n+1}^{max} at t_{n+1}^{max}. Here, to maintain depletion safety for the whole horizon T_H, it is crucial that soc_{n+1}^{max} is met at t_{n+1}^{max}. We allow for a certain SoC deviation soc_Δ to increase consumption flexibility. Large values of soc_Δ allow for a higher power consumption during the Free region but increase the risk of downtime, and lower the power consumption in the following DS region. Our parameter evaluation shows that soc_Δ should range between 1% and 19%; above and below these limits, performance suffers significantly. We found that $soc_\Delta = 5\%$ offers a good balance across the storage size and prediction methods.

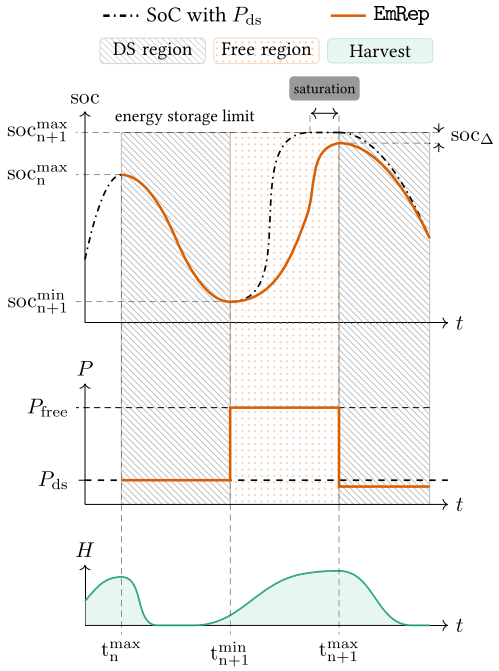


FIGURE 7 Behaviour of EmRep; management regions are determined using SoC estimation; in DS region power consumption P_{ds} ensures uninterrupted operation leads storage saturation; in Free region the energy surplus is exploited by increasing node consumption to P_{free} with a goal to provide higher utility

As stated in Section 1, EmRep is tailored to small size storage, which is assumed to recharge significantly during the horizon. If T_{free} is small, or the margin between SoC extrema is small, the flexibility for EmRep is small and hence P_{free} is only slightly higher than P_{ds}. For harvesting patterns, which may lead to two SoC peaks, EmRep still works and the DS region and the Free region occur multiple times per horizon. However, in such situations T_{free} and the difference between SoC extrema is small and so is the surplus energy; hence, the benefit of EmRep towards DS decreases.

3.3.3 | Utility optimisation

As stated in Section 3.2, utility is defined as a ratio between a maximum budget and achieved budget. This linear correlation allows for optimisation of the budget; hence, a maximum budget also maximises the utility. Additionally, it is not beneficial for the application to provide a budget higher than B_{max}. In the SoC curve estimation, EmRep takes this into account and uses the utility profile $U^*(s)$ together with B_{max} as upper bound for the budget in slot s.

3.3.4 | Schedule generation

As detailed in Figure 5, EmRep implements the energy manager entity. To develop a schedule, which can be executed by the application; however, a scheduler is needed, similarly to the

one presented in Ref. [27]. EmRep is compatible with schedulers that keep a target average consumption B, equivalent to a duty cycle as stated in Equation (2). For evaluation, we use a scheduler that transforms the average consumption into a uniformly distributed set of activities within the timeslot. The simple scheduler, which we call *classical* scheduler, assumes a program consisting of only one part with duration d_{task} and an average power consumption P_{task}. Both parameters have to be obtained by in-lab measurements of the real hardware and fed into the scheduler. First, the classical scheduler determines the overall energy available in the slot and calculates the maximum number of repetitions $N \in \mathbb{N}^+$ of the program which can be executed so that the assumption

$$B \cdot V_{\text{n}} \cdot T_{\text{S}} \geq N \cdot d_{\text{task}} \cdot P_{\text{task}} + (T_{\text{S}} - N \cdot d_{\text{task}}) \cdot P_{\text{q}} \tag{10}$$

holds. Here, T_{S} denotes the length of the timeslot and P_{q} the power drawn in the used sleep mode. Afterwards, starting times for N tasks are generated by distributing the times uniformly throughout the slot. Prior simulations show that the influence of the task distribution within the timeslot is limited as long as the average consumption is met; hence, the classical scheduler is used in favour of the computational-expensive scheduler presented in Ref. [27].

4 | EVALUATION SETUP & METRICS

In this section, we briefly elaborate on energy managers used for comparison, introduce the simulation tools and hardware, outline the used metrics and motivate relevancy of utility profiles. For comparison, the authors of PreAct [8] have kindly provided us with an implementation of their state-of-the-art energy manager for long-term ENO systems. Since depletion-safety is key for EmRep, a natural competitor is the DS algorithm presented in Ref. [7]. Please note that DS is utility-agnostic whilst both PreAct and EmRep include time-varying utility into their management by design.

4.1 | Hardware & simulation toolkit

We use our prototype harvester which is presented in detail in Ref. [27]. The sensor node is powered by a solar cell generating 35 mA at 4 V. The power generated at the solar panel is directly fed into a supercapacitor; hence, the voltage level of the capacitor influences the power point of the solar panel. To provide a stable supply voltage of $V_{\text{n}} = 3.3$ V for the sensor node, the harvester is supplied with a TI TPS61021 A boost converter. We have shown [27] that the conversion circuit offers an efficiency between 78% and 93% across output currents between 1 and 330 mA.

Real-world experiments for EH-WSN are time consuming and equal conditions for all nodes are hard to obtain. Hardware-specific variances, shadowing conditions and wireless medium access make fair comparison of energy management techniques nearly impossible. Various testbeds [34–36] exist which emulate harvesting conditions. Still, wireless transmission conditions may change due to interference and influence the power consumption of a sensor node, for example, connecting time to a WiFi access point increases with the surrounding traffic [37]. Furthermore, the variety of parameters, for example, energy manager, scheduler, capacitor size, and harvest estimation, increases the number of experiments and hence the time needed for the whole experiment is increased drastically. Hence, we use a thoroughly designed simulation toolkit, which ensures repeatable conditions and a fair comparison. To demonstrate that the used simulation toolkit is on par with the real world, we have conducted an experiment with the testbed as presented in Ref. [34]. The testbed uses controlled LEDs to replay previously recorded or artificial traces to investigate corner cases or the general behaviour of energy-aware algorithms. We replicate attached sensors of the node by switched resistors that lead to similar power consumption. The same artificial light trace is used for the testbed and the simulation toolkit and the sensor node activity and SoC are compared. We found that simulation and real-world test lead to the same outcome and hence conclude coherence between simulation and real-world performance.

In Ref. [27], the simulator has been validated by comparing the simulations to real-world tests.

The data set is 194-day real-world solar trace recorded with a similar miniature harvester at a time resolution of 5 min. These values are fed into the simulation toolkit and are used for slot averaging and harvest estimation. In addition to the provided EWMA harvest estimation, we implement the WCMA estimation as well as the state-of-the art estimation algorithm Pro-Energy. Furthermore, we extend the simulation framework to a utility manager, which gives a control about the utility profiles. Please note that the framework also simulates node shutdown, that is, a voltage dropping below V_{min}; the utility is consequently set to zero. It is also assumed that a shutdown leads to loss of harvest estimation; hence, prevalent patterns have to be learnt again. Table 1 gives an overview of the used simulation parameters. The parameters for WCMA and Pro-Energy are chosen based on suggestions in the corresponding papers. However, our goal is not to compare prediction methods but to investigate the influence of differently designed prediction methods on energy managers. For brevity, we only show a subset of the simulation results.

4.2 | Metrics

To evaluate the performance of the different energy managers, we use utility as the primary metric as presented in Section 3.2. The utility is used to define the specific amount of activity for the sensor node; hence, we can assume that a utility in a timeslot s greater than the specified utility $U^*(s)$ is not beneficial. Hence, we define the *effective* utility

TABLE 1 Summary of simulation parameters

Parameter	Value(s)
Simulation duration	194 d
Harvest resolution	5 min
Solar cell	Up to 35 mA (and 4 V)
EWMA parameters	$\alpha = 0.8$
WCMA parameters	$\alpha = 0.8$, $D = 10$, $K = 5$
Pro-energy parameters	$\alpha = 0.5$, $D = 14$, $K = 3$, $G = 5$
Number of time slots	6, 12, 24
Supercapacitor size	25, 50, 100, 200, 400 F
Supercapacitor voltage V_{cap}	Up to 2.7 V
Supercapacitor leakage current	16 to 261 μA
Shutdown voltage V_{min}	1.3 V
Turn-on voltage	1.6 V
Conversion efficiency	$\eta = 0.78$ to 0.93
Maximum budget B_{max}	2.73 mA

$$U_{\text{eff}}(s) = \frac{\min(U(s), U^*(s))}{U^*(s)}. \tag{11}$$

Enabled by this metric, we can judge how well the desired utility profile is met. As stated in Section 3.2, we define the utility based on a maximum budget B_{max} which can be achieved by the sensor node. Defining B_{max} is non-trivial: since the sensor has to fulfil the ENO condition, it cannot spend more than it harvested. Hence, we set it to the average current fed into the capacitor resulting in $B_{max} = 2.73$ mA. However, this value can also be changed to an application-specific value, for example, representing a budget needed for the desired sampling rate. To reveal differences between configurations, we also define a relative utility

$$U_{\text{rel}}(s) = \frac{U_{\text{ref}(s)} - U_{\text{eff}}(s)}{U_{\text{ref}}(s)}. \tag{12}$$

The reference effective utility $U_{\text{ref}}(s)$ is specific to a configuration, for example, for a specific capacitor size.

4.3 | Utility profiles

To highlight the influence of time-varying utility on energy-management, we select four different utility profiles, which we depict in Figure 8, and motivate their relevancy in the following. Please note that we keep the average throughout T_H equal for all profiles since it directly influences the results of the effective utility. All utility profiles are defined with hourly resolution, which we found to be most intuitive. If higher resolution is required, it can easily be changed in the simulation.

The *Const* profile serves as the baseline for management, as it replicates utility-agnostic behaviour. Each slot is equally important and hence no saving of energy is performed. However, it is totally different from greedy algorithms or IC devices as the goal is to provide a constant duty cycle—also in night time. An example application is a weather surveillance station which constantly gathers temperature and humidity in the air.

The *Workday* profile represents a classic case at which harvest is aligned with the interest of the application. Between 8 and 20 h, the interest of the application is very high, while evening, night and very early mornings are out of interest. This should be the profile where EmRep performs best since it profits most from the application interest in the Free region. An example application is a gas sensor at outside work plants, for example, harbours. Here, during normal work hours, the interest is high to ensure safe working conditions for employees while interest outside the working hours is of limited interest.

The *Worknight* profile is the most demanding for energy managers working on solar harvesting system since application interest is misaligned with harvest. An example application is a surveillance camera system. During the day, workers are present so risk by burglars is limited, while night times are lonely, hence gathering images is important.

The *Rushhour* profile is used by our VAM case study. Here, early mornings and evenings are of most interest. This applies also to traffic surveillance systems since traffic is highest during these times. For energy managers, providing high utility at the end of low-intake phases is usually most challenging since the energy storage is at very low levels.

Please note that the shown utility profiles are examples to highlight the abilities and opportunities of utility-based management for EH-WSN. The utility profiles can be freely defined based on applications' needs at hourly resolution and for values $U^*(s) \in [0, 1]$.

5 | RESULTS

In this section, we showcase the following:

- The influence on the SoC curve and saturation times (Section 5.1)
- The interplay of energy storage capacity, absolute and relative utility (Section 5.2)
- The effect of application utility profiles on different energy managers (Section 5.3)
- The impact of non-ideal prediction harvest forecast (Section 5.4)
- A case study for structural health monitoring based on VAM on bridges (Section 5.5)
- The limitations of EmRep (Section 5.6)

5.1 | Benefits of decoupling

To highlight the benefits of decoupling low- and high-energy intake periods for energy management, we analyse SoC

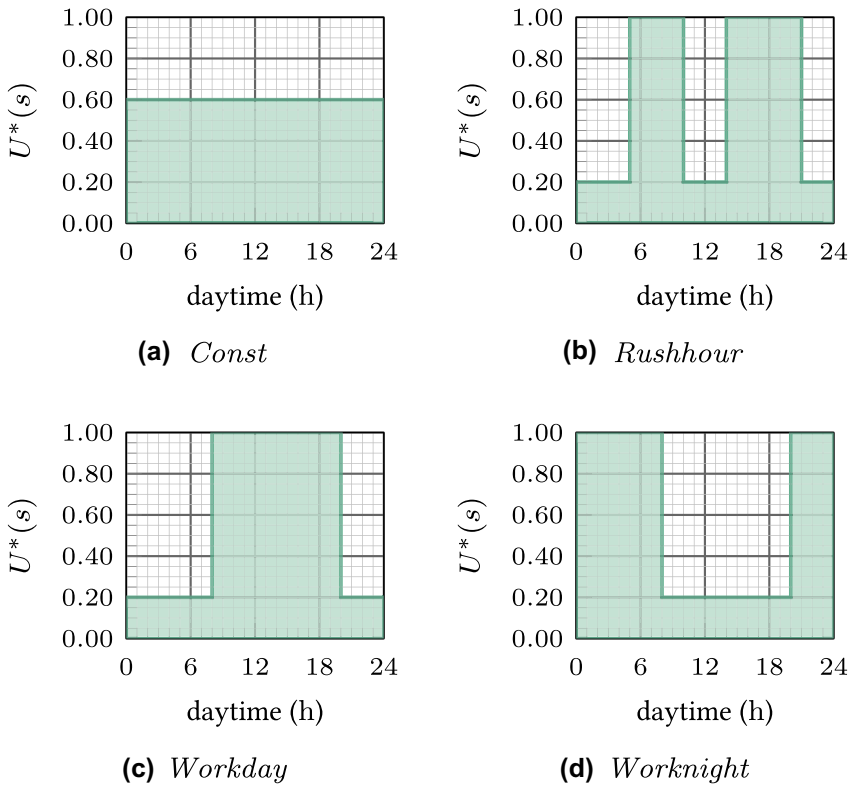


FIGURE 8 Examples for potential utility profiles used for simulation; average requested utility is equal for all profiles; profiles can be freely defined on hourly resolution

curves and resulting utilities in Figure 9. We compare all energy managers with the same EWMA filtered harvest prediction for our data set. The day is divided into $S = 24$ slots; hence, the accuracy of forecast and utility calculation is 60 min. For completeness, we also show the harvest at the solar panel of the four-day excerpt. The size of the capacitor is 50 F.

At the beginning of day 35, all energy managers share a nearly equal utility of around $U = 0.15$, which is limited by the amount of energy stored in the capacitor. The latter was fully charged before and since no harvest is expected during night time, it equals an average consumption of I_n=0.45 mA. Since the predicted harvest is slightly lower than expected, all managers have to correct their average consumption slightly immediately following the minimum SoC.

At position (a), EmRep evaluates the choice of B again to reach the maximum at 17 h, position (b). Since SoC is still low at 0.1, B is even lower than before to the minimum. This, however, changes with the increasing harvest at which B can be adjusted so that an average utility of $U = 0.32$ between 11 and 17 h is reached. Still, after 17 h at Day 35, position (b), the utility during the night is on par with the other energy managers since the SoC is similar during late in the afternoon.

The same trend follows for day $36 - 38$: the time with full energy storage roughly decreases by 1 h per day. This translates to a reduction of saturation time by 20% compared with the five hours of DS. More importantly, the ability to optimise B only towards the end of the high harvest phase at 18 h allows EmRep to increase utility to $U = 0.63$ between 11 and 18 h at Day 37. At the same time, PreAct averages $U = 0.2$ and DS achieves $U = 0.17$.

This advantage also manifests when comparing the saturation times with different energy managers. In Figure 10, we depict the number of hours in the 194 d lasting data set, which is spent at full storage. The effect of reducing saturation by design is clearly visible: EmRep decreases time in saturation by up to 37.3% at 25 F compared with PreAct. This yields an absolute reduction of 391 h or an average of 2.01 h/d. As the storage size increases, the problem of saturation lessens, for example, at 200 F EmRep reduces saturation time by 19.1% compared with DS. Next, we highlight the influence of the storage size on utility.

5.2 | Influence of storage capacity

5.2.1 | Absolute results

We highlight the influence of the supercapacitor capacity on the effectiveness of the different energy managers in Figure 11. The ordinate shows the average of the slot-wise computed effective utility. The different subplots show an increasing number of timeslots, which affects utility and prediction accuracy as well as management reactivity. Figure 11 showcases the situation with ideal slot-wise prediction. This means that energy managers work on the correct slot average; however, the misalignment of harvest throughout the timeslot can affect the results, c.f. Figure 2. For example, the slot average can be equal if high harvest arrives late in the slot compared with

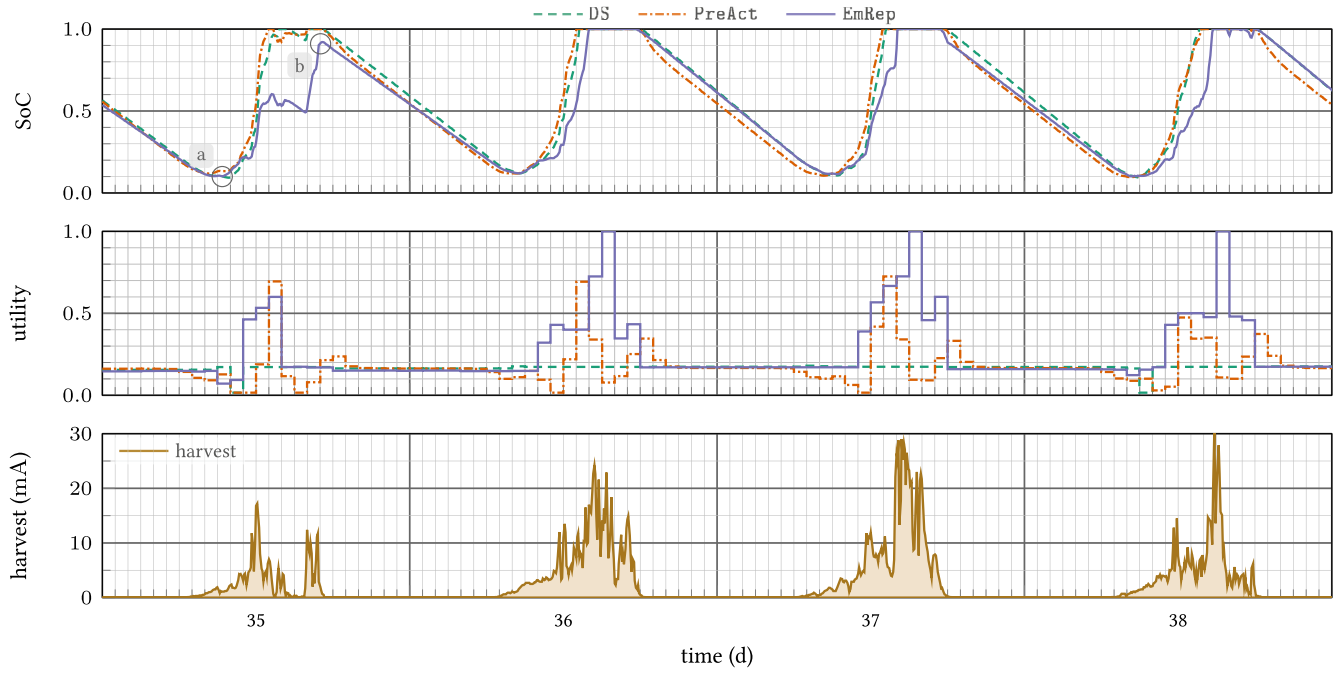


FIGURE 9 Comparison of SoC curve and utility for different energy managers; simulation uses EWMA prediction with $S = 24$ and C= 50 F; DS shows nearly constant utility while PreAct pushes towards the benefits of surplus energy; still EmRep is the only energy manager that achieves full utility together with higher average utility

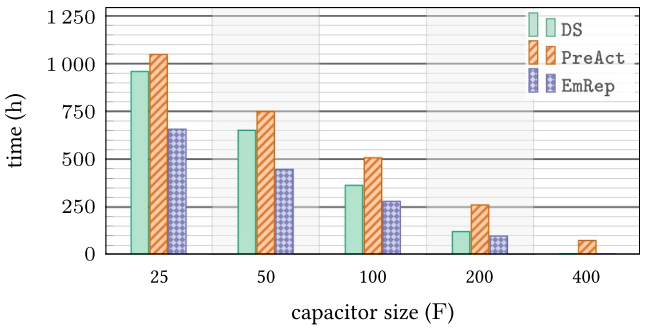


FIGURE 10 Average saturation duration of energy storage for small capacitors is significantly reduced; the results include prediction methods and slot sizes as shown in Table 1 and *Const* profile; with a small 25 F capacitor, EmRep decreases time with a saturated storage at minimum by 31.5%

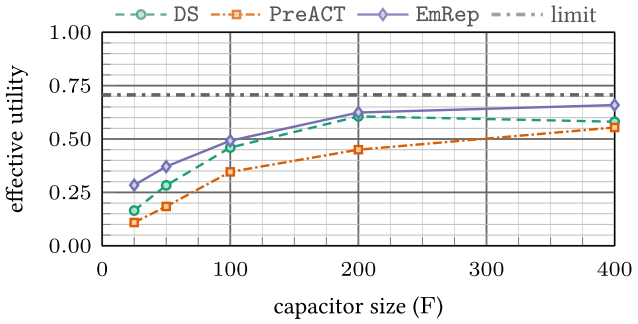
stable (lower) harvest throughout the slot—the effect on the sensor node, however, can be totally different since it could have been shutdown before the end of the slot. This is especially critical at early mornings as SoC is at a very low level. We also depict a realistic limit in our simulations: as stated in Section 4, the capacitor voltage influences the power point of the solar panel. Our limit shows the average power which could have been generated at the solar panel if the SoC was always soc = 0.5. The closer an energy manager is able to get to this line, the better the performance is.

Clearly, only increasing the energy storage does not linearly increase the effective utility. Increasing the capacity by eight, yields an increase in the utility of 2.03× for EmRep, of 2.41× for PreAct, and 3.60× for DS at $S = 24$. Interestingly, both PreAct and EmRep show an increase up until a capacity of 200 F. Doubling the storage capacity additionally does not increase the utility effectively for all energy managers. The reason behind this is the size, and the power, respectively, of the solar panel. The small 35 ×35 mm panel does not suffice to recharge the capacitor completely; hence, no benefit in utility can be seen. This underlines again that solely increasing the storage capacity—over-dimensioning—only makes sense when the size of the solar panel is increased as well.

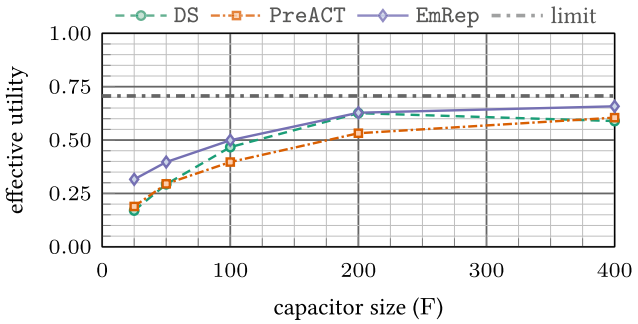
The number of timeslots, however, appears to have a minimal effect, at least at ideal prediction, for EmRep and DS. Although the slot length increases from 1 to 4 h, the effective utility only decreases by 7% for DS and 2% for EmRep for a 100 F capacitor. PreAct on the other hand is quiet sensitive to reduction in slot length which manifests in a loss of 32%. This, however, is mainly caused by extreme reactions of the underlying PID-controller. These high fluctuations in the duty cycle can be corrected with short timeslots and high-capacity storage, c.f. Figure 11c, but otherwise can have severe consequences. We try to adjust PID parameters accordingly, but were not able to improve results. However, these configurations are a bit out of scope since PreAct is mainly designed for long-term ENO systems with large-sized storage.

5.2.2 | Relative results

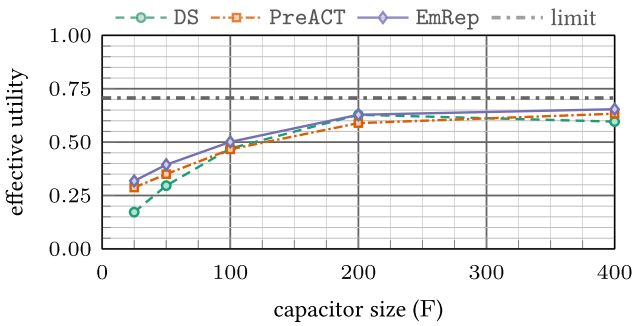
Typically, small-sized energy storage saturates during the times of high energy intake. With EmRep, we aim to mitigate this effect and enhance the effective utility. Hence, we highlight the influence of the capacitor size on the performance of energy



(a) $S = 6$



(b) $S = 12$



(c) $S = 24$

FIGURE 11 Average effective utility with different storage capacity and timeslot lengths with ideal slot-wise prediction; *Const* utility is required throughout the day; EmRep outperforms PreAct and DS at long timeslots and small-sized storage; benefits diminish at 400 F and $S = 24$; best performance should meet the practical limit (average harvest power)

managers in Figure 12. Here, we show the difference in effective utility in our simulation setup compared with the reference, a 200 F capacitor. We choose the 200 F capacitor as reference, since increasing the storage further offers only little improvements for two of the three energy managers.

Across all simulation runs, the energy managers lose between 13.6% and 60.3% of their performance when decreasing the size of the capacitor from 200 to 25 F. However, while DS and PreAct lose 58.7% and 60.3%, EmRep only loses 44.4%. Still, EmRep keeps an average utility level of 0.36 which is 1.42× higher than DS and 1.7× higher than PreAct.

Interestingly, DS is not able to improve the utility in the use of a 400 F capacitor. This is a direct consequence of the

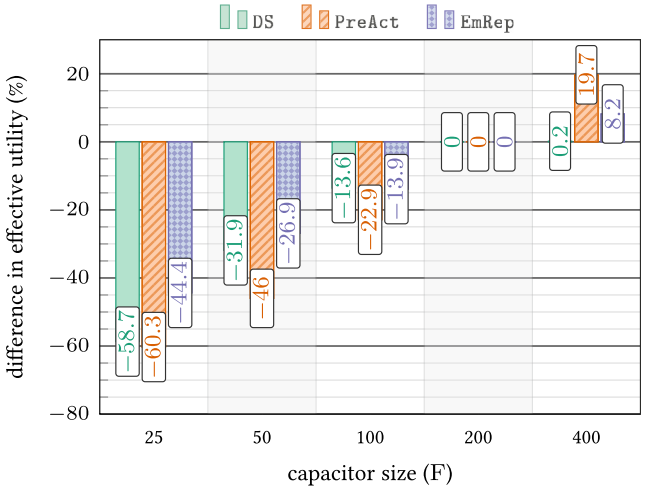


FIGURE 12 Relative difference in utility, compared with 200 F capacitor; EmRep loses significantly less utility with capacitors smaller than 100 F

depletion safe algorithm, which tends to operate at lower SoC levels. In a direct charge circuit, the operating point of the solar panel is, hence, sub-optimal, and the generated power is lower. PreAct, however, tends to keep high SoC levels, which enable the system to profit from over-sized storage. EmRep sits in the middle; building up on DS, it also tends to favour low SoC but profiting also from non-saturating storage.

5.3 | Utility profiles

Next, we highlight the influence of the different utility profiles presented in Section 3.2 and showcase the resulting utility in Figure 13. Each bar incorporates 60 different simulation configurations with varying predictor, cap size, and number of time slots.

The baseline for the other profiles is the *Const* profile. Here, EmRep outperforms both DS and PreAct. In this profile, each time of the day is equally important; hence, surplus activity during daytime is beneficial. However, as visible by the performance of PreAct, spending available energy too greedily harms the overall performance.

The performance of the *Worknight* profile shows how well the energy managers are able to maintain activity during night time. Here, the DS algorithm as a base serves very well since it aims at maintaining an equal performance level throughout the day. Although this profile structure contrasts with the design of EmRep—being designed to use surplus energy—it still matches or outperforms the utility level of DS. However, increasing the utility with *Worknight* profile requires a larger cap; for example, with a 50 F capacitor DS and EmRep both reaching at the utility of 0.48. This image changes with a 400 F capacitor at which DS reaches $U = 0.6$ and EmRep achieves 0.65. Please note, that in this profile, saturation times are generally higher since the surplus energy during daytime is not used to increase activity.

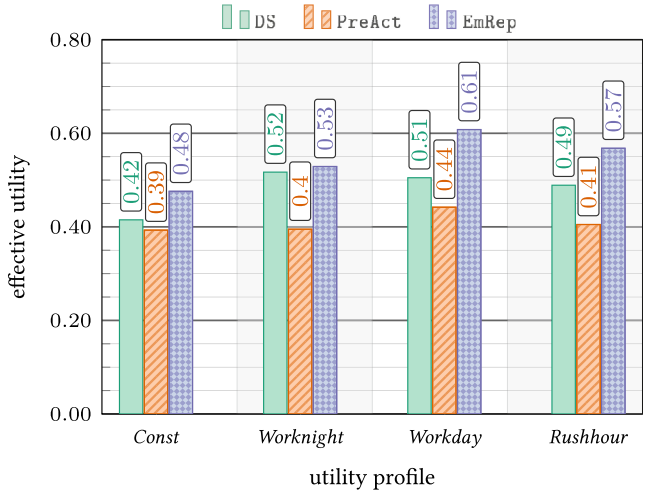


FIGURE 13 Mean utility with different profiles; mean includes cap sizes, prediction methods and slot length as stated in Table 1; EmRep profits most from valued activity at daytime; EmRep matches the performance of DS at sub-optimal profile *Worknight* and outperforms PreAct

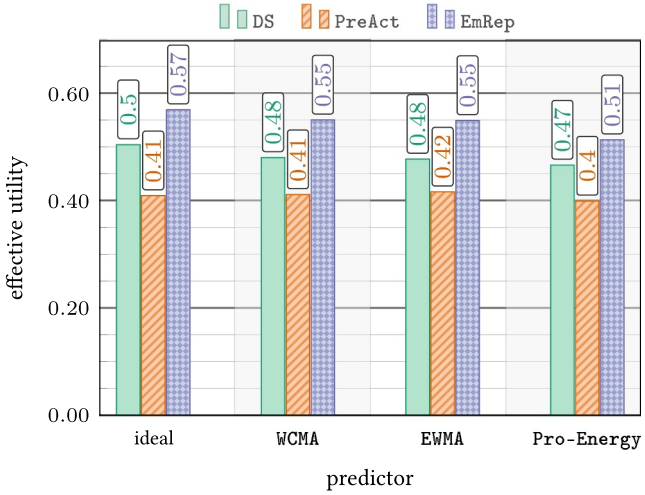


FIGURE 14 Mean utility with different predictors; influence of prediction algorithm is minor, implicating that simple algorithms (EWMA) may be favoured; although EmRep performs worst in combination with Pro-Energy, it still tops the performance of other energy managers with ideal prediction

The *Workday* and *Rushhour* profiles are ideally suited for algorithms that make use of surplus energy. EmRep is able to increase utility by 0.13 (*Workday*) and 0.10 (*Rushhour*). Also, PreAct reaches its best performance here, although it still suffers to match the performance of DS.

5.4 | Prediction influence

Since a fundamental basis of EmRep is the time of SoC extrema, the used harvest prediction method is important for the overall performance. We conduct simulations for three different short-term prediction methods for solar EH and depict the results in Figure 14. For comparison, we also show the results with original data, that is, ideal prediction. However, even the ideal prediction uses slot averages; hence, the deviation within the time slot is still affecting the results.

Under ideal conditions but also with all used prediction methods, EmRep maintains the top-end performance across the simulation configurations. WCMA and EWMA only produce a small drop in utility—for all energy managers—but with Pro-Energy prediction, the performance drops notably. This is due to the fact that the prediction of Pro-Energy may completely change throughout the day. Out of a pool of similar harvest profiles, Pro-Energy continuously selects the profile most similar to the already observed profile. While this decreases the prediction error, it also changes the SoC curve completely; hence EmRep works sub-optimal. This is on par with the findings presented [16] where error rates increase with the increasing forecast horizon up to 2.5 h.

Interestingly, PreAct shows a very stable performance across prediction methods and is even capable of improving its performance with the use of an EWMA-filtered prediction. This effect occurs with long timeslots, that is, 4 h, at which harvest deviation has a high impact. In early mornings, the harvest may rise only at the very end of the timeslot, which increases the slot average. Since PreAct often uses a vast part of incoming energy directly, it suffers from shut-downs in early mornings with little capacitors. This effect is softened by an EWMA filter since the slot average generally is lower; and hence also the used budget.

In general, the influence of the prediction algorithm is rather limited for all energy managers. Although prediction-error-centric work [19] found difference in Mean Absolute Percentage Error (MAPE), the actual implications onto a deployed system remains unclear. This is due to the fact that prediction errors often cancel out throughout the day; that is, the difference for the SoC is smaller than that of the MAPE. Hence, our investigations imply that simple prediction methods often suffice, even if their MAPE is higher.

5.5 | Case study: VAM

Structural health monitoring of bridges or buildings offers great insights to ageing or possible defects of materials but entails a high installation cost of wired power supply and high maintenance effort. Supplying these sensor nodes with solar energy lowers maintenance costs with low impact on measurement performance. VAM, as presented in Refs. [4, 38], uses piezo-electric inducers to generate high frequency pulses. The measurement method requires structural vibration at low frequencies, which are typically caused by cars or trains. Hence, only with a combination of low and high frequency pulses, useful measurements can be obtained. Therefore, it is ideally suited for the *Rushhour* utility profile, at which energy is spent when high traffic, and hence low frequency vibrations are present. Each measurement contains the stimulation of different frequencies typically lasting 100 ms each. Good quality results can be obtained with 50 different frequencies,

but the quality of the measurements strongly increases with the number of tested frequencies.

The deployed piezo-electric transducer uses sinusoidal signals with an amplitude upto 12 V. The signal generator typically draws a current of 20 mA. We feed these parameters into the simulation framework and highlight the results in Figure 15. For simplicity, we simulate hourly resolution and simple harvest prediction with EWMA. The box plot shows the number of individual frequency measurements per hour for different capacitor sizes. With the small 25 F capacitor, DS achieves 110 measurements in 50% of the cases whilst PreAct provides 280 and EmRep 380 measurements. Hence, EmRep enables the VAM method either to increase frequency depth by 35% or upto two additional complete readings with 50 frequencies each, compared with PreAct. Even more interestingly, EmRep achieves a higher median performance with a 50 F capacitor than that of DS and PreAct with a 100 F capacitor. This allows for a much smaller package—supercapacitor volume usually scale linearly with capacity—without sacrificing any performance.

5.6 | Limitations & discussion

EmRep shows a strong performance across various harvest estimation patterns, slot sizes, utility profile and capacitor sizes. Our investigations, however, also reveal deficiencies. The performance benefit in the *Worknight* profile is quite limited, compared with DS. This is dominated by diminishing benefits at small size storage, although EmRep is tailored to small size storage. However, buffering energy for the night requires a certain size, for example, a 50 F capacitor only allows for an average budget of 0.8 mA during the night. The options of shifting energy here are very limited. Furthermore, investigations on the estimation methods reveal that EmRep needs an estimator with stable prediction for T_H. In particular, Pro-Energy, which achieves very low MAPE for forecasts up until 2 h, is no good partner for EmRep since varying extrema prediction hurts performance. Also, large-size storage lets the benefits of EmRep fade: relatively small solar cells combined with large capacitors lead to low variations in SoC. If energy storage is held at 50% of SoC, all incoming energy can be stored and no energy surplus can be used for increasing utility additionally.

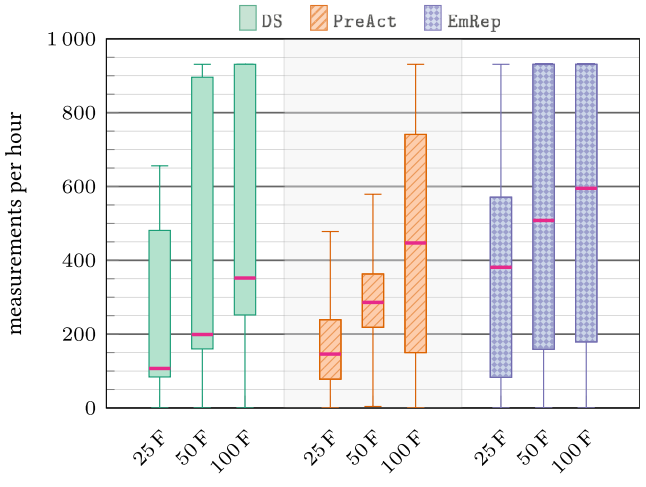


FIGURE 15 Boxplot for VAM case study; *Rushhour* utility profile with EWMA prediction with $S = 24$; even with a 50 F capacitor, EmRep outperforms DS and PreAct in median measurements per hours; for 25 F, EmRep increases measurements per hour by at least 2.5×

6 | CONCLUSION & OUTLOOK

In this study, we presented EmRep, a new energy manager for ENO systems. It uses SoC extrema prediction to decouple energy management in phases where depletion safety is important from phases where energy surplus exists. This allows EmRep to increase the effective utility across a plethora of combinations, for example, doubling utility with 25 F and long timeslots. Our VAM case study also showed that the measurement frequency can be increased by 2.5× for small size storage. EmRep paves the way for smaller sensor nodes powering the EH-WSN. In future, we plan to test EmRep with different harvest sources and perform real-world test at harbour bridges.

ACKNOWLEDGEMENTS

Open Access funding enabled and organised by Projekt DEAL. WOA Institution: N/A Blended DEAL:Projekt DEAL.

CONFLICT OF INTEREST

There are no conflicts of interest.

PERMISSION TO REPRODUCE MATERIALS FROM OTHER SOURCES

None.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Lars Hanschke https://orcid.org/0000-0001-7143-7486

REFERENCES

1. Adkins, J., et al.: The signpost platform for city-scale sensing. In: Proceedings of 17th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN'18, pp. 188–199. IEEE Press, New York (2018)
2. Gaglione, A., et al.: Energy neutral operation of vibration energy-harvesting sensor networks for bridge applications. In: Proceedings of the International Conference on Embedded Wireless Systems and Networks, EWSN 18, pp. 1–12. ACM, Madrid, Feb 2018
3. Magno, M., et al.: SmarTEG: an autonomous wireless sensor node for high accuracy accelerometer-based monitoring. Sensors. 19(12), 2747 (2019)
4. Yang, S., et al.: Ultrasonic wireless sensor development for online fatigue crack detection and failure warning. Struct. Eng. Mech. 69(4), 407–416 (2019)

5. Weber, S., et al.: A decade of detailed observations (2008–2018) in steep bedrock permafrost at matterhorn hörnligrat (zermatt, CH). Earth Syst. Sci. Data. 11, 1203–1237 (2019)
6. Fan, Y., et al.: Astar: sustainable battery free energy harvesting for heterogeneous platforms and dynamic environments. In: Proceedings of the 2019 International Conference on Embedded Wireless Systems and Networks, EWSN'19, pp. 71–82. Junction Publishing, USA (2019)
7. Renner, C., et al.: Perpetual data collection with energy-harvesting sensor networks. ACM Trans. Sens. Netw. 11(1), 12 (2014)
8. Geissdoerfer, K., et al.: Getting more out of energy-harvesting systems: energy management under time-varying utility with PREAcT. In: Proceedings of the 2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), pp. 109–120. IEEE, New York (2019)
9. Hester, J., Jacob, S.: Flicker: rapid prototyping for the batteryless internet-of-things. In: Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems, SenSys'17. Association for Computing Machinery, New York (2017)
10. Bhatti, N.A., Mottola, L.: Efficient state retention for transiently-powered embedded sensing. In: Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks (EWSN), Graz, Feb 2016
11. Majid, A.Y., et al.: Dynamic task-based intermittent execution for energy-harvesting devices. ACM Trans. Sen. Netw. 16(1), 1–24 (2020)
12. Ahmed, S., et al.: Intermittent computing with dynamic voltage and frequency scaling. In: Proceedings of the International Conference on Embedded Wireless Systems and Networks, EWSN'20. Junction Publishing, United States (2020)
13. Fraternali, F., et al.: Ember: energy management of batteryless event detection sensors with deep reinforcement learning. In: Proceedings of the 18th Conference on Embedded Networked Sensor Systems. ACM, New York, Nov 2020
14. Moser, C., et al.: Adaptive power management for environmentally powered systems. IEEE Trans. Comput. 59(4), 478–491 (2010)
15. Sliper, S.T., et al.: Energy-driven computing. Philos. Trans. A Math. Phys. Eng. Sci. 378(2164), 20190158 (2019)
16. Cammarano, A., Petrioli, C., Spenza, D.: Online energy harvesting prediction in environmentally powered wireless sensor networks. IEEE Sensor. J. 16(17), 6793–6804 (2016)
17. Piorno, J.R., et al.: Prediction and management in energy harvested wireless sensor nodes. In: Proceedings of the 1st International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, VITAE'09. IEEE, New York, May 2009
18. Kansal, A., et al.: Power management in energy harvesting sensor networks. ACM Trans. Embed. Comput. Syst. 6(4), 32 (2007)
19. Draskovic, S., et al.: A case for atmospheric transmittance: solar energy prediction in wireless sensor nodes. In: 2018 IEEE International Conference on Green Computing and Communications (GreenCom). IEEE, New York, Jul 2018
20. Sharma, A., Kakkar, A.: A review on solar forecasting and power management approaches for energy-harvesting wireless sensor networks. Int. J. Commun. Syst. 33(8), e4366 (2020)
21. Jackson, N., Adkins, J., Dutta, P.: Capacity over capacitance for reliable energy harvesting sensors. In: Proceedings of the 18th International Conference on Information Processing in Sensor Networks, IPSN'19, pp. 193–204. ACM, New York (2019)
22. Simjee, F., Chou, P.: Everlast: long-life, supercapacitor-operated wireless sensor node. In: Proceedings of the 12th International Symposium on Low Power Electronics and Design, ISLPED'06. ACM, New York, Oct 2006
23. Hester, J., Sitanayah, L., Jacob, S.: Tragedy of the Coulombs: federating energy storage for tiny, intermittently-powered sensors. In: Proceedings of the 13th ACM Conference on Embedded Network Sensor Systems, SenSys'15, pp. 5–16. ACM, New York, Nov 2015
24. Christopher, M., et al.: Adaptive control of duty cycling in energy-harvesting wireless sensor networks. In: Proceedings of the 2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks. IEEE, New York, Jun 2007
25. Buchli, B., et al.: Dynamic power management for long-term energy neutral operation of solar energy harvesting systems. In: Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems-SenSys'14. ACM Press, New York (2014)
26. Ahmed, S., et al.: The betrayal of constant power x time: finding the missing joules of transiently-powered computers. In: Proceedings of the 20th ACM SIGPLAN/SIGBED International Conference on Languages, Compilers, and Tools for Embedded Systems, LCTES'19, pp. 97–109. ACM, New York (2019)
27. Hanschke, L., Renner, C.: Scheduling recurring and dependent tasks in EH-WSNs. Sust. Comput. Informat. Syst. 27, 100409 (2020)
28. Hanschke, L., Renner, C.: Time- and energy-aware task scheduling in environmentally-powered sensor networks. In: Proceedings of 14th International Symposium on Algorithms and Experiments for Wireless Networks, AlgoSensors'18. Springer Nature, New York (2018)
29. Dutta, P., et al.: Energy metering for free: augmenting switching regulators for real-time monitoring. In: Proceedings of 7th International Conference on Information Processing in Sensor Networks, IPSN'08, pp. 283–294. IEEE, New York (2018)
30. Moser, C., et al.: Real-time scheduling for energy harvesting sensor nodes. Real-Time Syst. 37(3), 233–260 (2007)
31. Yang, J., Tilak, S., Rosing, T.S.: An interactive context-aware power management technique for optimizing sensor network lifetime. In: Proceedings of 5th International Conference on Sensor Networks-Volume 1, SENSORNETS'16, pp. 69–76. SciTePress, Setúbal (2016)
32. Renner, C., Turau, V.: CapLibrate: self-calibration of an energy harvesting power supply with supercapacitors. In: Proceedings of 23th International Conference on Architecture of Computing Systems 2010, Hanover, Feb 2010
33. Renner, C., Turau, V.: Adaptive energy-harvest profiling to enhance depletion-safe operation and efficient task scheduling. Sust. Comput. Informat. Syst. 2(1), 43–56 (2012)
34. Hanschke, L., et al.: Light insight—Emulation of radiation traces for analysis and evaluation of solar-harvesting algorithms. In: Proceedings of 5th International Workshop on Energy Neutral Sensing Systems, ENSsys, Delft, Nov 2017
35. Hester, J., Scott, T., Jacob, S.: Ekho: Realistic and repeatable experimentation for tiny energy-harvesting sensors. In: Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems, pp. 330–331. ACM, New York (2014)
36. Geissdoerfer, K., Chwalisz, M., Shepherd, M.Z.: A portable testbed for the batteryless iot. In: Proceedings of the 17th Conference on Embedded Networked Sensor Systems, SenSys'19, pp. 83–95. Association for Computing Machinery, New York (2019)
37. Hanschke, L., Heitmann, J., Renner, C.: Stop waiting: mitigating varying connecting times for infrastructure WiFi nodes. In: Proceedings of 16th GI/ITG KuVS Fachgespräch "Sensornetze", FGSN'17. Hochschule für angewandte Wissenschaften, Hamburg (2017)
38. Peter, O., et al.: Towards structural health monitoring using vibro-acoustic modulation in the real world. In: 18. GI/ITG KuVS FachGespräch SensorNetze, FGSN, pp. 21–24, Magdeburg, 19–20 Sept 2019

How to cite this article: Hanschke, L., Renner, C.: EmRep: energy management relying on state-of-charge extrema prediction. IET Comput. Digit. Tech. 16(4), 91–105, (2022). https://doi.org/10.1049/cdt2.12033