2007

Adaptive Control of Duty Cycling in Energy-Harvesting Wireless Sensor Networks

Christopher M. Vigorito
University of Massachusetts - Amherst

Adaptive Control of Duty Cycling in Energy-Harvesting Wireless Sensor Networks

Christopher M. Vigorito, Deepak Ganesan, and Andrew G. Barto
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01002
Email: {vigorito,dganesan,barto}@cs.umass.edu

Abstract—**Increasingly many wireless sensor network deployments are using harvested environmental energy to extend system lifetime. Because the temporal profiles of such energy sources exhibit great variability due to dynamic weather patterns, an important problem is designing an adaptive duty-cycling mechanism that allows sensor nodes to maintain their power supply at sufficient levels (energy neutral operation) by adapting to changing environmental conditions. Existing techniques to address this problem are minimally adaptive and assume a priori knowledge of the energy profile. While such approaches are reasonable in environments that exhibit low variance, we find that it is highly inefficient in more variable scenarios. We introduce a new technique for solving this problem based on results from adaptive control theory and show that we achieve better performance than previous approaches on a broader class of energy source data sets. Additionally, we include a tunable mechanism for reducing the variance of the node's duty cycle over time, which is an important feature in tasks such as event monitoring. We obtain reductions in variance as great as two-thirds without compromising task performance or ability to maintain energy neutral operation.**

I. INTRODUCTION

Wireless sensor networks and embedded systems are becoming commonplace in many fields of research (e.g., environmental monitoring). The spatially distributed nature of such networks often requires that the individual sensor nodes be powered by batteries. One of the major limitations on performance and lifetime of such networks is the limited capacity of these finite power sources, which must be manually replaced when they are depleted. Recent work [1]–[4] has explored scenarios in which nodes can harvest energy from their environment (e.g., from the sun) and use it to recharge their batteries. In the absence of such energy (e.g., at night in the case of solar energy), nodes can then subsist on their replenished battery supply.

The introduction of energy harvesting capabilities into wireless sensor networks introduces many design questions about the construction of such systems. For example, how can a node use its harvesting abilities intelligently to increase its task performance and lifetime? Can a node adapt its power consumption profile online so as to subsist indefinitely on a given energy source without compromising its task performance? Such questions have been addressed in recent work [1] which presents algorithms for dynamically adapting the duty cycle of a node so as to maximize both lifetime and performance. The concept of "energy neutral operation" is also introduced in [1], which informally amounts to a condition in which the energy consumed by the node is always less than or equal to the energy harvested from the environment. It is a non-trivial task to design duty-cycling algorithms that achieve energy neutral operation, since the dynamics of the energy sources being harvested may not be easily predictable.

Existing approaches [1] to the problem of dynamic duty-cycling of nodes with energy harvesting capabilities attempt to model the energy source and adjust the node's duty cycle in anticipation of expected incoming energy or lack thereof. These methods also attempt to handle stochasticity in the energy profile by comparing observed energy input with expected input (given by the model) and adjusting the duty cycle accordingly. We present an alternative, model-free approach to solving this problem using techniques from adaptive control theory [5] and show that our system achieves better performance on a wider class of environmental data sets than previous approaches. Additionally, our system does this while making no assumptions about the nature and dynamics of the energy source, making our approach more easily implemented in real systems where data about the energy source may not be available beforehand.

Another contribution of this work is the consideration

of duty cycle stability, which to our knowledge has not been addressed in previous work on dynamic duty-cycling algorithms for energy harvesting networks. We contend that in addition to energy neutral operation, certain applications which employ sensor networks require the duty cycling behavior of the node to be as *stable* as possible, meaning that it should have low variance over time. There are a number of reasons why this condition may be considered important. Nodes involved in event monitoring tasks must minimize their sleep time in order to detect fleeting and unpredictable events and report them with low latency (usually to a central location) [6]. In such cases, one would ideally want the node to be equally likely to be awake at any given point in time.

The duty cycle of a node is also important for communication between sensor nodes. For instance, a popular MAC layer protocol, BMAC [7], uses the characteristics of a neighboring node's duty cycle to determine the length of a packet preamble. With these considerations in mind, we augment our duty-cycling algorithm with a parameterized method for reducing the variance of a node's duty-cycling profile to varying degrees without compromising its goals of maintaining energy neutral operation and maximizing its task performance.

In summary, this work presents the following contributions to research involving adaptive duty-cycling of energy harvesting sensor nodes:

- **Generality:** Our approach is model-free with respect to the energy source and thus can be implemented in any energy harvesting system without the need for a priori information about the source. We present results for both periodic (solar) and aperiodic (wind) data sets.
- **Computational Efficiency:** As we will show, our algorithm has only constant time and constant space requirements, making it easily implementable on many low-power, unsophisticated hardware platforms.
- **Adaptivity:** Our approach is designed to allow the system to respond more appropriately in situations where the battery level happens to get perilously low. Our algorithm achieves 0% dead time on all data sets tested in comparison to another algorithm's dead times of as much as 16%.
- **Tunable Stability:** We present a tunable mechanism for minimizing the variance of a node's duty-cycling profile, an important feature for many sensor net applications. Our results show reductions in variance of up to two-thirds using this method.

II. RELATED WORK

Energy-aware tasking protocols for various distributed applications have been studied extensively in recent years [8]–[10]. However, these approaches make tasking decisions based solely on battery level and do not consider the potential of harvesting environmental energy to supplement battery supply. Other recent work has explored incorporating energy harvesting techniques into wireless sensor networks [1]–[3]. Much of this work, however, has been concerned primarily with design, simply taking the extra energy provided by the harvesting system as an overall boost in lifetime and performance. There is little work studying methods of taking harvested energy into account when making energy-aware tasking decisions so as to maximize lifetime and performance. Two exceptions are [3] and [11], which show that incorporating such information into a tasking algorithm can indeed improve performance and lifetime as compared to naive approaches. These methods, however, provide only heuristic techniques for using the harvested energy efficiently.

Kansal et al. [1] present a more principled algorithm for dynamically adapting the duty cycle of a node by observing deviations in energy input from an estimated model of the harvested energy source. In their approach, it is assumed that the energy source is periodic and that the period of the energy profile is known (e.g., one day in the case of solar energy). A single period is discretized into blocks of equal duration, and an estimate of the expected energy input during that block (given in average mW per block for their experiments) is learned from historical data. The estimates are learned using an exponentially weighted moving average of recorded energy input rates for a given block on past days.

Energy input is assumed to be constant over the course of a single block, and the block duration is chosen to be small enough such that this assumption is likely to hold. Initial duty cycles are set based on an initial estimate of the average energy input rates for each of the blocks, which is assumed to be known a priori from existing data. Online changes in duty cycle are based on mismatches between the actual energy received during a block and the expected energy given by the model for that block. Surpluses of energy resulting from these mismatches are utilized by increasing the duty cycle during future blocks of the current period. Similarly, deficits are compensated for by reducing the duty cycle during future blocks accordingly.

Although Kansal et al. formulate their algorithm

within a theoretical framework which they develop and term harvesting theory, much of the power of their algorithm comes from a priori information about the harvested energy source. The theory is used to provide constraints on the choices of duty cycles for particular blocks, but the initial choices are made based on existing data, and subsequent choices are made based on a model with the assumptions described above. Our approach, described in the next section, actually takes a step back from modeling the energy source, using only the current battery level of the node to make duty-cycling decisions. However, as we show, our formulation implicitly accounts for the energy harvested from the environment and makes duty cycling decisions accordingly.

III. ACHIEVING MAXIMUM PERFORMANCE ENERGY NEUTRAL OPERATION WITH ADAPTIVE CONTROL

In this section we introduce some concepts from adaptive control theory and outline our architecture for achieving energy neutral operation (henceforth ENO) while maximizing task performance. In particular, we formulate the problem as a linear-quadratic (LQ) tracking problem and provide a very simple control law for achieving both objectives.

A. A Battery-Centric Objective Function

We first propose an objective function for a given node that is defined only in terms of the node's battery level and argue that, if minimized, it achieves both ENO and maximum task performance subject to maintaining ENO, the dual objectives of an energy harvesting node. We term the condition of satisfying both of these goals the ENO-Max condition. Our informal definition of ENO is as given in [1]—namely that the energy consumed by a node is always less than or equal to the energy harvested.

Here, as in [1], we assume that task performance is strongly correlated with a node's power consumption profile. This is a reasonable assumption in most sensor network tasks, e.g., event monitoring, where maximizing the awake time of a node (which in turn increases power consumption) amounts to maximizing its performance. We also assume that the node has some means of adjusting its power consumption profile online. This can be implemented as in [1] by having the node autonomously set its duty cycle. Alternatives include autonomous dynamic voltage scaling (DVS).

Denote a node's initial battery level as $B_0 \in [0, 1]$ (i.e., percent full) and the battery level at discrete time step t as $B_t \in [0, 1]$. We argue that if a node maintains its battery level such that $B_t = B_0 \ \forall t > 0$, then the

node satisfies the ENO-Max condition. It is not hard to see why this is so if we momentarily assume for clarity that any battery leakage is negligible. Any time at which $B_t < B_0$ indicates a violation of the ENO condition, since the node must have consumed more energy than it has replaced with harvested energy. Thus, as long as $B_t \geq B_0$, the node maintains ENO.

Considering next the condition of maximizing task performance, any situation in which $B_t > B_0$ indicates that the node has harvested energy it has not yet made use of. Thus we can say that when $B_t \leq B_0$, the node has made efficient use of the energy it has harvested. Since satisfying the ENO-Max condition requires satisfying both of these objectives, the node must satisfy $(B_t \geq B_0) \wedge (B_t \leq B_0) \ \forall t > 0$, which is satisfied when $B_t = B_0 \ \forall t > 0$. This is our formal definition of the ENO-Max condition.

In general it will not be possible for a node to adjust its duty cycle to maintain the ENO-Max condition exactly for all $t > 0$. Thus we define the following cost function for the node to minimize to obtain an optimal duty-cycling schedule:

$$\lim_{N \to \infty} \frac{1}{N} \sum_{t=1}^{N} (B_t - B_0)^2. \tag{1}$$

A node minimizing (1) will minimize the average squared deviation of the battery from its initial level, and thus be as close as possible to maintaining the ENO-Max condition given the dynamics of its environment (which includes hardware limitations). Note also that removing the assumption that battery leakage is negligible poses no problem since we can view any leaked energy as a corresponding increase in power consumption, for which the node will compensate when minimizing (1).

B. Optimal Linear Quadratic Tracking

We next present some background on a well-defined problem in adaptive control theory whose solution we adapt in the following section to develop an algorithm for maintaining the ENO-Max condition. Results from research in adaptive control theory have been in wide use for decades in many areas of industry for maintaining optimal control of dynamical systems [12]. One of the problems addressed by adaptive control theory is the optimal tracking problem which, informally stated, attempts to apply external control to a dynamical system so as to keep some output variable at a desired value or sequence of values (trajectory) over time. This is done by minimizing some cost function, which is generally a function of the error between the output variable and

the target value(s). A more specific definition, which is the one we address, concerns the case in which the dynamics of the system are linear in the controls, outputs, and noise of the system, and the cost function to be minimized is quadratic in the outputs. This formulation is referred to as a linear-quadratic (LQ) tracking problem, for which there exist closed-form optimal control laws. We additionally concern ourselves with the special case of this problem in which the desired trajectory of the output is a constant value.

More formally, we consider a first order, discrete-time, linear dynamical system with colored noise. Such a system is assumed to have dynamics that conform to

$$y_{t+1} = ay_t + bu_t + cw_t + w_{t+1}, \qquad (2)$$

where y represents the output of the system, u is the control, w is mean zero input noise, $a, b, c \in \Re$ are real-valued coefficients, and all subscripts indicate discrete time steps. Note that the coefficient c determines the contribution of previous noise terms to the evolution of the system, which is what gives rise to the colored noise.

The objective of the system is to keep $|y_t - y^*|$ small for all values of t, where y^* in this case is the constant value desired for the output. Formally, the system attempts to minimize the average squared tracking error

$$\lim_{N \to \infty} \frac{1}{N} \sum_{t=1}^{N} (y_t - y^*)^2. \qquad (3)$$

It can be shown (e.g., in [5]) that the optimal control law minimizing (3) is given by

$$u_t = \frac{y^* - (a + c)y_t + cy^*}{b}, \qquad (4)$$

which does not depend on w, but does depend on the coefficient c of previous noise terms.

In situations where the coefficients in the control law are not known a priori, as is the case in our problem formulation, they can be estimated online using standard gradient descent techniques [5]. To do this, we introduce a parameter vector $\theta = (a+c, b, c)^T$ to represent the true coefficients and a feature vector $\phi_t = (y_t, u_t, -y^*)^T$. From these definitions it is easy to see that the control law given in (4) can be expressed as $\phi_t^T \theta = y^*$. We can then define a gradient descent update rule for the estimated parameter vector $\hat{\theta}_t$ as described in [5], which is given by

$$\hat{\theta}_{t+1} = \hat{\theta}_t + \frac{\mu}{\phi_t^T \phi_t} \phi_t (y_{t+1} - \phi_t^T \hat{\theta}_t), \qquad (5)$$

where μ is a positive constant step-size parameter.

C. Satisfying ENO-Max with LQ Tracking

We now formulate the problem of a given sensor node maintaining the ENO-Max condition as an LQ tracking problem of the form described in the previous section. Such a formulation requires making certain assumptions, which we will justify in this section. First, we assign the variables of the tracking problem appropriately, letting y_t be the battery level, B_t, of the node at time t and u_t be the node's duty cycle at time t. The colored noise, w_t, will model the moving average of battery level increments produced by the harvested energy. Note that in this formulation harvesting inefficiencies need not be modeled directly because the algorithm only observes the actual harvested energy. This is in contrast to [1], in which harvesting inefficiencies had to be modeled explicitly (with the parameter η) because the harvested source was modeled directly.

By choosing these variable assignments we assume that the dynamics of the node's battery, given now as a function of y, w, and u, evolve according to (2). That is, we assume linearity of the battery level with respect to itself, the incoming energy from the environment, and the power consumption profile as defined by the duty cycle. Most batteries have approximately linear discharge and recharge rates in the middle region of their voltage discharge curves, with non-linearities occurring at the extremes. Thus, the linearity assumption is in general a reasonable one if the range of the reported battery level (y) is clamped around the linear region of the battery's voltage discharge curve.

It is straightforward to transform the objective function we defined in (1) into the cost function for the LQ tracking problem given by (3). We simply let $y^* = B_0$ and $y_{t+1} = B_{t+1}$ and note that the two are then equivalent. However, we mention one possible design choice here that changes our notation slightly. It should be clear that a system designer would want y^* to be some value in the mid-range of the battery level interval $[0, 1]$, so that the battery can act as a buffer both in the case of surpluses of energy as well as in the case of deficits. We suggest a value between 0.5 and 0.75, as values in this range incorporate a bias towards tolerating energy wasting and against tolerating over-depletion.

It may not be plausible or desirable, however, to set the initial charge of a node's battery to exactly the level of y^*. In general, a system designer would likely want the convenience of fully charging all of the nodes in a network before deployment. This does not pose a problem because any mismatch between the initial

battery level B_0 and the desired target for the LQ tracker y^* becomes negligible in terms of the cost function as $t \to \infty$. We therefore change notation slightly and say that y^* is set by the system designer to a "pseudo" initial battery level B^*, rather than to B_0.

It has been suggested [12] that the parameter estimation algorithm (5) will converge must faster if $\hat{\theta}$ is initialized to a reasonable value rather than arbitrarily. Given the dynamics (2) of the system we describe and the variable assignments discussed earlier, it is clear that the coefficient b should be negative (power consumption decreases battery level), a should be positive and near 1, and c should also be positive (energy influx increases battery level). For these reasons we suggest initializing $\hat{\theta}$ to $(2, -1, 1)^T$, recalling that $\hat{\theta}(0) = a + c$. We have observed empirically that $\hat{\theta}$ converges quickly in all of our simulations given this initialization, which should be appropriate for most energy harvesting systems. [1]

D. Reducing Duty Cycle Variance

Another design issue which to our knowledge has not been addressed in previous work on performance-aware energy harvesting is attempting to minimize the variance of the node's duty-cycling profile subject to maintaining the ENO-Max condition. As mentioned in Section 1, this will be of particular interest in applications such as event monitoring where the likelihood of an event may be uniformly distributed over time and the goal of the network is minimum response latency to such events. In such cases, one would ideally want a node to be equally likely to be awake at any given point in time.

Our goal is to have the node smooth its choices of duty cycle over time such that the variance of the duty-cycling profile is minimized, subject of course to maintaining the ENO-Max condition. If a node can realize this third objective, then it will tend to choose duty cycles that take it between periods of high and low energy influx without large changes in its duty cycle. In this case, if the duty cycle represents percentage of time awake, the probability distribution which gives the likelihood that the node will be awake at any given point in time will be closer to a uniform distribution than if this objective is not considered.

There are many possible ways to filter the control outputs of the LQ-Tracker so as to decrease variance of the duty-cycling profile. We present a very simple method for doing so and show that it can be used to

reduce duty-cycling profile variance to varying degrees without violating the ENO-Max condition. Our approach is to smooth the sequence of control outputs $\{u_t\}$ of the LQ-Tracker with a simple exponential weighting scheme. That is, the duty cycle chosen by the node will be an exponentially weighted moving average of previous control outputs of the LQ-Tracker. One advantage of this approach is its simplicity and ease of computation. The update rule for computing the smoothed control signal $\bar{u}_t$ from the control output u_t of the LQ-Tracker at time step t is given by

$$\bar{u}_t = \bar{u}_{t-1} + \alpha(u_t - \bar{u}_{t-1}), \qquad (6)$$

where $\alpha \in (0, 1]$ is a smoothing parameter discussed in more detail shortly.

This has the effect of setting the duty cycle to a recency-weighted average of previous values of u, where the weighting is a function of α. A general rule of thumb is that the number of significant data points that contribute to the average is given approximately by $1/\alpha$ [12]. Thus, for large values of alpha (close to 1), smoothing over only the last few data points will occur. When α is close to 0, the value of $\bar{u}$ will be averaged over increasingly longer histories of the sequence $\{u_t\}$, leading to greater smoothing. Note that if $\alpha = 1$ then no smoothing occurs, and we restrict α to be greater than 0 because $\alpha = 0$ would always keep $\bar{u}$ at its initial value.

The choice of α will be a function of the time step duration and thus will be application dependent. We do not present a method for adjusting α online, but rather discuss guidelines for choosing appropriate values for the parameter. If no information about the energy source is known a priori, then α must be chosen based solely on the heuristic mentioned above. That is, to smooth the sequence of control outputs over approximately x time steps, α should be set to $1/x$. If any information about the nature of the energy source is known, this information along with the heuristic mentioned above, can be used to set α appropriately.

Setting the duty cycle directly to $\bar{u}$ will, in some cases, prevent the node from adapting quickly to large bursts or deficits in energy availability, potentially violating the ENO-Max condition for long periods of time. We therefore introduce a simple, parameterized tradeoff mechanism for allowing both the short time scale control of the LQ-Tracker and the longer time scale control of the weighted average to contribute to the choice of duty cycle. The tradeoff is parameterized by $\beta \in [0, 1]$, which determines the relative contributions of each control signal by setting the duty cycle ρ_t at time step t to a

---

[1]When implementing our algorithm, it may also be useful to introduce the restrictions $\hat{\theta}(0) > 0, \hat{\theta}(1) < 0$, and $\hat{\theta}(2) > 0$ to prevent the values from exiting their appropriate ranges.

```
θ̂ ← (2, -1, 1)^T
B ← initial battery level ∈ [0, 1]
B* ← target battery level ∈ [0, 1]
ρ, u, ū ← initial duty cycle ∈ [0, 1]
φ ← (B, u, -B*)^T
loop forever
  B ← current battery level ∈ [0, 1]
  θ̂ ← θ̂ + μ/(φ^T φ) φ_t(B - φ^T θ̂)
  u ← σ[(B* - θ̂(0)B + θ̂_t(2)B*)/θ̂(1)]
  φ ← (B, u, -B*)^T
  ū ← ū + α(u - ū)
  ρ ← βu + (1 - β)ū
end loop
```

Fig. 1. Sensor node algorithm pseudocode.

weighted combination of u_t and $\bar{u}_t$ according to

$$\rho_t = \beta u_t + (1 - \beta)\bar{u}_t. \qquad (7)$$

The value of β will clearly be application dependent and should be set to 1 in cases where duty cycle variance is irrelevant, since this will produce no contribution from the weighted average control. For applications in which duty cycle variance should be minimized, it is suggested that β be set between 0.25 and 0.75, depending on how critical variance minimization is. Smaller values of β will increase the contribution of the weighted average, thus reducing variance, but may lead to poor performance if the energy source is highly variable, since the node will not be able to respond as well to quick variations in energy availability. We present performance results for differing values of β in Section 5.

E. Algorithm and Implementation Details

Armed with the theoretical formulation of the problem of maintaining the ENO-Max condition outlined in the previous sections, we now discuss methods of implementation in wireless sensor nodes and present a constant time, constant space algorithm for maintaining this condition in a given node. Figure 1 gives pseudocode for the algorithm. We first note that the loop is over discrete time steps whose duration must be chosen a priori. However, the computational demand of the algorithm at each iteration is minimal, even for most low-power systems, and for the most part can be neglected when choosing the time step duration.

It is clear that the node will not be able to perform updates while in sleep mode, and so the actual time between all iterations will not be constant. We assume that the system is designed to perform iterations of the

algorithm at a constant rate when in awake mode, and to begin a series of these equally spaced updates upon awaking from sleep mode until sleep mode is entered again. The asymmetry of the time step durations will not affect the algorithm's performance, however, since the control law does not directly depend on the amount of time that has passed since the last update. Realistically, in most applications it will be sufficient to have the node run one iteration of the algorithm upon awaking from sleep mode. This will have the effect of executing one wake-sleep cycle before deciding whether or not to alter the timing of the next.

Finally, as seen in Figure 1, we have introduced the rectifier function $\sigma(u)$ to constrain the control output u to be a valid duty cycle. That is, $\sigma(u)$ is defined as

$$\sigma(u) = \begin{cases} 0 & \text{if } u < 0 \\ 1 & \text{if } u > 1 \\ u & \text{otherwise} \end{cases} . \qquad (8)$$

IV. EXPERIMENTAL DESIGN

We compared our LQ-Tracking algorithm with the algorithm presented in [1] in simulation on three different solar energy data sets and a wind energy data set. This section presents the details of our simulations and the specifics of the data sets.

A. Sensor Node Simulation

Our sensor node simulation was designed to model the performance characteristics of the Heliomote platform, a solar energy harvesting sensor node [13]. The power consumption of the node was simulated to be 3mW in sleep mode (P_{sleep}) and 100mW in active mode (P_{max}) under a 100% duty cycle. Power consumption $P_c(t)$ in mW under a given duty cycle $\rho_t \in [0, 1]$ at time t was therefore calculated as $P_c(t) = \rho_t P_{max} + (1 - \rho_t)P_{sleep}$.

We then used this energy consumption rate to compute a change in battery voltage level per time step. This was done by choosing a constant that approximated the slope of the linear region of the discharge curve for a standard NiMH rechargeable battery [14]. The voltage values at the beginning and end of this linear region represented 100% full and 0% full, respectively. When the battery level reached zero, all duty cycling decisions by the algorithm were overridden and the duty cycle was manually set to 0 until the battery regained charge through harvesting. Energy recharging was treated as the inverse of power consumption, and the power output by the harvesting system was obtained from the data.

B. Algorithmic Details

We chose a time step duration of one minute for our simulations. This was thought to be fine grained enough given the nature of the energy sources to provide realistic dynamics. Thus, each iteration of our algorithm was run once every minute of real time, a frequency at which our results suggest would be reasonable in a real system.

The framework outlined in [1] described a utility function, parameterized by ρ_{min} and $\rho_{max} \in [0, 1]$, that was a function of a node's duty cycle. Duty cycles were restricted to be in the interval $[\rho_{min}, \rho_{max}]$, and the utility function was assumed to be strictly increasing over that interval. We chose a special case of this function in which $\rho_{min} = 0.01$ and $\rho_{max} = 1$ because we feel that higher values of ρ_{min} are unlikely to be necessary in most applications, and higher duty cycles should in general increase utility, even though there may be diminishing returns. We note, however, that varying ρ_{min} and ρ_{max} produced little effect on the performance of our algorithm except for the trivial cases in which there was not enough energy output by the source for the node to support a minimum duty cycle of ρ_{min}. This validated our choice of utility function.

Our algorithm was implemented as in Figure 1, with the exception that $\rho_{min} = 0.01$ was used as the lower bound of the rectifier function σ, instead of 0. We let $B^* = 65\%$, $\mu = 0.001$, the initial duty cycle be 20%, and the initial battery level be 95% for all of our experiments. Changing these parameters to other reasonable values had little effect on our results, and so we present only the results for these values.

We compared the performance of our algorithm against the algorithm given in [1]. To implement the algorithm, we chose the same time discretization as used in that paper—48 blocks per day representing 30-minute blocks of time. The exponential weighting parameter for estimating the average expected energy for each block was set to 0.5 as in [1]. The initial estimates of the expected energy input for each of the blocks in the algorithm were obtained by averaging the data points in each of those blocks over the entire data set beforehand. Finally, the case of energy harvesting inefficiencies was not considered and thus $\eta = 1$ in our simulations. Experimentation with other reasonable discretizations and time step durations yielded similar results.

C. Data Sets

We obtained data for our evaluations from two sources. The first set was a portion of that used in [1] and consisted of 60 days of solar data collected by a Heliomote in Los Angeles, CA during July and August of 2005. From now on, we refer to this data set as Helio-CA. Data were in the form of average mW flowing out of the solar cells on the device, were sampled once every 10 seconds. One characteristic of this data set is its high predictability and correspondingly low variance. These properties provide a relatively easy task for a dynamic duty-cycling algorithm such as the one developed in [1] which relies on modeling the energy source with simple averaging methods. To illustrate the power and versatility of our model-free approach, we sought additional data sets that contained high variability in energy availability.

Our second source of data was the US Climate Reference Network (USCRN), which maintains a database of environmental data collected from various monitoring stations across the US [15]. In particular, we obtained a year's worth of solar radiation data from two weather stations—one in Durham, NH, and one in Darrington, WA—and a year's worth of wind speed data from the station in NH. We will refer to these data sets as USCRN-NH, USCRN-WA, and USCRN-WIND, respectively. We used data spanning from January 1, 2005, through December 31, 2005, for both stations. Solar data were in the form of W/m^2 and were given as an average over each hour along with the standard deviation for that hour. These were converted to units of mW/m^2 to match the units of the Helio-CA data set. The wind data gave averages of wind velocities in m/s over each hour and the standard deviation for each hour.

To make the USCRN data usable in our simulations, we had to scale them down to represent a reasonable estimate of the power which a Heliomote would likely obtain if placed in the corresponding environment. To do this, we took the maximum-valued data point of the Helio-CA data set, $\max_{CA}$, and the maximum-valued data points of each of the USCRN solar data sets, $\max_{NH}$ and $\max_{WA}$, and scaled all of the data points of each of the latter with the resulting ratios. That is, the USCRN-NH data set was scaled by a factor of $\max_{CA}/\max_{NH}$, and the USCRN-WA data set by a factor of $\max_{CA}/\max_{WA}$.

For the wind data, we chose a somewhat ad-hoc mapping of wind speeds to energy values because we were unable to find specifications for any relatively small wind energy harvester that would have given us a realistic mapping of wind speeds to average mW obtained. However, the purpose of using this data set was to obtain performance results for our algorithm on an aperiodic energy source. Therefore, whether the exact energy harvested by a node for various wind speeds

Data Set	Helio-CA		USCRN-NH		USCRN-WA		USCRN-WIND	
Algorithm	Kansal et al.	LQ-Tracker	Kansal et al.	LQ-Tracker	Kansal et al.	LQ-Tracker	Kansal et al.	LQ-Tracker
Mean Duty Cycle	31.44	33.40	26.00	29.11	18.47	22.43	28.44	37.68
Duty Cycle Var.	19.82	15.73	17.34	14.71	12.85	12.13	18.08	10.64
Time Dead (%)	0.55	0.0	14.97	0.0	16.32	0.0	3.26	0.0
Time Full (%)	2.33	0.0	4.38	0.73	4.68	0.24	17.54	0.93

TABLE I

PERFORMANCE COMPARISON OF ALGORITHM OF KANSAL ET AL. [1] AND THE LQ-TRACKING ALGORITHM WITH NO EXPONENTIAL
SMOOTHING ($\beta = 1$) ON THREE DIFFERENT SOLAR ENERGY DATA SETS AND A WIND DATA SET.

matches our converted values is mostly irrelevant. What matters is that the energy obtained is proportional to the wind speeds observed and so the conversion method for the solar data was applied in the same manner to the wind data. That is, each of the wind speeds was scaled by a factor of $\max_{CA}/\max_{WIND}$, where $\max_{WIND}$ was the maximum wind speed of the entire data set.

Although the algorithm in [1] is designed specifically for dealing with periodic energy sources, it does update its estimates of incoming energy online relatively rapidly, and so we chose to run it on the wind data set as well, keeping the same period and discretization as for the solar data sets. We feel that although this is not a completely fair comparison, it is a reasonable one to make so as to give us a baseline for evaluating the performance of our algorithm on an aperiodic data set.

D. Evaluation Metrics

For each experiment we evaluated each algorithm on four performance metrics:

- **Mean Duty Cycle**: This provided a measure of task performance and was taken to be the average of the duty cycles chosen by the algorithm over all time steps. Note that this average includes the overriding zeros forced on the algorithm when the node's battery level reached 0.
- **Duty Cycle Variance**: A measure of the variability of the node's duty cycle choices over time was computed as the average squared difference between the duty cycle at each time step and the average duty cycle computed for the first metric.
- **Percent Time Dead**: We measured the percent of time during which the node was inoperable. This was taken to be the ratio of the number of time steps for which the node's battery level was 0 to the total number of time steps in the experiment.
- **Percent Time Full**: To asses the percentage of time during which the node was unable to make use of incoming energy, we computed the ratio of time

steps for which the node's battery level was 100% to the total number of time steps.

V. RESULTS

Our first experiments compared the performance of the algorithm in [1] with the performance of our LQ-Tracking algorithm on each of the four data sets. For these experiments, no duty cycle smoothing was used in our algorithm ($\beta = 1$). The results of these experiments are shown in Table 1. We see that for all four data sets the LQ-Tracker produced a higher average duty cycle than the algorithm of Kansal et al. Additionally, the time series of duty cycles produced by the LQ-Tracker have lower variance than the other algorithm in all four cases, even without any explicit smoothing. More importantly, the LQ-Tracker never reached a battery level of 0 on any of the data sets, and the amount of time during which it wasted incoming energy was minimal for each data set (less than 1%). We see also that the aperiodic nature of the wind data set posed no problems for our algorithm.

Conversely, although the algorithm in [1] does well on the Helio-CA data set, its performance decreases drastically when presented with data sets exhibiting higher variability, spending significant portions of time with an either empty or full battery. This is largely a result of the energy source-centric nature of the algorithm. It is unable to adjust its duty cycle choices appropriately when the node's battery level is exceedingly low (as seems often to be the case in the two USCRN solar data sets) or high (as in the case of the wind data set).

The rest of our experiments studied the effects of varying the parameters α and β on the variance of the duty-cycling profile and on the overall performance of the algorithm. In each of these experiments one of the parameters was varied independently of the other and a reasonable value for the fixed parameter was chosen based on existing performance data.

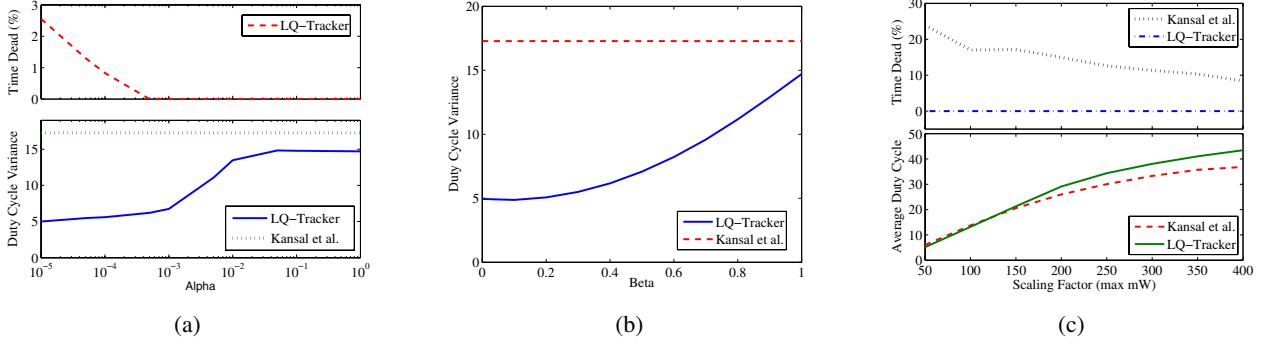Figure 2(a) shows the effect of varying the smoothing parameter α on the variance of the duty cycle for the

Fig. 2.   (a) Effect of α on duty cycle variance and lifetime for the USCRN-NH data set. (b) Effect of β on duty cycle variance for the USCRN-NH data set. (c) Effect varying the scaling factor of the data for the USCRN-NH data set on lifetime and performance.

USCRN-NH data set when $\beta = 0.5$. The variance of the duty cycle time series for the algorithm in [1] is included as a reference point. Note the log scale on the x-axis. These results clearly show the strong effect α can have on duty cycle variance, reducing the variance from just under 15% (when no smoothing is used) down to less than 5% for very small values of α, a two-thirds reduction. Note that the upper bound of the variance is 25%.

However, we note that this effect is more pronounced for certain ranges of α than for others. Values of α greater than 0.1 produce no effect on the variance of the duty cycle, although the variance is still lower in these cases than for the algorithm in [1]. This is because $\bar{u}$ is averaged over only a few data points in these cases, producing little smoothing. Similarly, values smaller than 0.001 have diminishing effects on the variance as they continue to get smaller. This is a result of $\bar{u}$ being smoothed over periods longer than the periodicity of the energy source (the sun in this case), which provides no additional benefit over smoothing across a single period. Values in the region between 0.001 and 0.1, however, affect the variance of the duty-cycling profile to more varying degrees.

Another important result shown in Figure 2(a) is that although they reduce the variance of the duty cycle, very small values of α can also negatively impact the performance of the node, as is evidenced by the time-dead curve for our algorithm, which is zero for many values of α but not for very small values. This is because too much smoothing hinders the ability of the node to respond adaptively to longer deficits in energy availability, even though β is set to 0.5. Thus, a good value of α suggested by these results for this data set when $\beta = 0.5$ is given by where the time-dead curve reaches 0 — around 0.0005. For this value of α,

the algorithm minimizes the variance of the duty cycle subject to maintaining ENO. This value will clearly be different for different energy sources and profiles. For example, although not shown here, the time-dead curve for our algorithm on the Helio-CA data set was 0 for all values of α shown in Figure 2(a), and so a smaller value of α could be chosen to minimize variance further.

The effect of varying β on the variance of the duty-cycling profile for the USCRN-NH data set is illustrated in Figure 2(b) for the case where $\alpha = 0.0005$. We see that for all values of β the variance of the duty cycles chosen by our algorithm are consistently lower than the algorithm in [1], and that the variance decreases monotonically for increasingly small values of β. This is as expected, since smaller values of β give higher weight to the smoothed sequence of LQ-Tracker outputs. We also note that, although not shown here, the time-dead curve of our algorithm for this experiment was zero for all values of β, indicating that no value of β hindered the ability of the algorithm to maintain ENO.

Figure 2(c) illustrates the effect of varying the overall energy availability of the energy source. The purpose of this experiment is to show the robustness of our algorithm in adapting to different environmental conditions such as severe lack or large surpluses of energy. Rather than run our algorithm on many other data sets exhibiting varying degrees of energy availability, we simulated such energy profiles by scaling the numerator, $\max_{CA}$, of the data normalization factor for the USCRN-NH data set to the values given on the x-axis of Figure 2(c). As these results show, our algorithm adapts well to different energy availability profiles, never experiencing any dead time and adapting its average duty cycle appropriately to make use of the available energy. We see that although the algorithm in [1] adapts to some degree as well by adjusting its average duty cycle, it rarely achieves a

higher average duty cycle than our algorithm and, more importantly, produces significant dead times for all of the environmental conditions tested—as high as 25%.

VI. Discussion

We have outlined an efficient, unified approach for achieving three objectives of primary concern in energy harvesting wireless sensor networks—energy neutral operation, performance maximization, and duty cycle stability. As we have shown, our algorithm provides four significant contributions to this area of research.

The battery-centric nature of our approach yields generality of implementation, allowing our algorithm to be realized in most energy harvesting scenarios without making any assumptions about the profile of the harvested energy source or the existence of preliminary data pertaining to it. Our results for all four data sets, especially the USCRN-WIND data set, confirm this.

As our experimental results illustrate, our algorithm is also better equipped to recover from situations in which large variations in the energy profile may drive the battery level into dangerously low regions. This is due to the fact that our algorithm is battery-centric and thus such perilous situations are continually made known to the algorithm until it becomes successful in correcting them. The time-dead performance of our algorithm on all four data sets illustrates this.

Further advantages of our algorithm are its constant computational time and constant storage requirements. The algorithm given in [1] takes time proportional to the number of blocks in the discretization of the energy source's period, and an equivalent amount of space. Our algorithm requires only the storage of a few variables and a negligible amount of computational time at each iteration, making it easily implemented on most low-power, computationally constrained devices.

Finally, our mechanism for reducing variance in a node's duty-cycling profile was shown to achieve significant reductions in this variance (up to two-thirds) for appropriate parameter values without hindering the algorithm's ability to maintain ENO. We also showed that our system is robust to varying those parameters. The tunable property of this mechanism also gives our approach flexibility, making it appropriate both for applications such as event monitoring, in which node availability should be evenly distributed over time, and other applications in which variance in the duty-cycling profile is irrelevant to network performance.

References

[1] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Transactions on Embedded Computing Systems*, 2006.

[2] X. Jiang, J. Polastre, and D. Culler, "Perpetual environmentally powered sensor networks." in *IEEE Information Processing in Sensor Networks*, 2005, pp. 463–468.

[3] A. Kansal and M. Srivastava, "An environmental energy harvesting framework for sensor networks," in *ACM Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, 2003.

[4] M. Rahimi, H. Shah, G. S. Sukhatme, J. Heidemann, and D. Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2003.

[5] P. Kumar and P. Varaiya, *Stochastic Systems: Estimation, Identification, and Adaptive Control*. Englewood Cliffs, New Jersey: Prentice-Hall, 1986.

[6] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler, "Design of a wireless sensor network platform for detecting rare, random, and ephemeral events," in *Proceedings of the Fourth International Conference on Information Processing in Sensor Networks (IPSN'05)*, 2005.

[7] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 2004.

[8] M. Maleki, K. Dantu, and M. Pedram, "Lifetime prediction routing in mobile ad hoc networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, LA, 2003.

[9] R. C. Shah and J. M. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, Orlando, FL, 2002, pp. 350–355.

[10] M. Younis, M. Youssef, and K. Arisha, "Energy-aware routing in cluster-based sensor networks," in *Proc. 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 2002.

[11] T. Voigt, H. Ritter, and J. Schiller, "Utilizing solar power in wireless sensor networks," in *The 28th Annual IEEE Conference on Local Computer Networks (LCN)*, Bonn/Konigswinter, Germany, 2003.

[12] G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction, and Control*. Englewood Cliffs, New Jersey: Prentice-Hall, 1984.

[13] A. Kansal, D. Potter, and M. Srivastava, "Performance aware tasking for environmentally powered sensor networks," in *ACM/IEEE Int'l Symposium on Low Power Electronics and Design (ISLPED)*, 2004.

[14] Energizer no. NH15 AA Rechargeable NiMH battery datasheet. [Online]. Available: http://data.energizer.com/.

[15] US Climate Reference Network Database (USCRN). [Online]. Available: http://www.ncdc.noaa.gov/oa/climate/uscrn/.