
Neeraj Jaggi, Sreenivas Madakasira, Sandeep Reddy Mereddy, Ravi Pendse

Abstract

Upcoming sensor networks would be deployed with sensing devices with energy harvesting capabilities from renewable energy sources such as solar power. A key research question in such sensor systems is to maximize the asymptotic event detection probability achieved in the system, in the presence of energy constraints and uncertainties. This paper focuses on the design of adaptive algorithms for sensor activation in the presence of uncertainty in the event phenomena. Based upon the ideas from increase/decrease algorithms used in TCP congestion avoidance, we design an online and adaptive activation algorithm that varies the subsequent sleep interval according to additive increase and multiplicative decrease depending upon the sensor’s current energy level. In addition, the proposed algorithm does not depend on global system parameters, or on the degree of event correlations, and hence can easily be deployed in practical scenarios. We analyze the performance of proposed algorithm for a single sensor scenario using Markov chains, and show that the proposed algorithm achieves near-optimal performance. Through extensive simulations, we demonstrate that the proposed algorithm not only achieves near-optimal performance, but also exhibits more stability with respect to sensor’s energy level and sleep interval variations. We validate the applicability of our proposed algorithm in the presence of multiple sensors and multiple event processes through simulations.

Keywords: Adaptive Algorithms, Sensor Activation, Temporal Correlations, Energy harvesting Sensor Systems

*Department of Electrical Engineering and Computer Science, Wichita State University, Wichita KS 67226 USA, Email: {neeraj.jaggi, sxmadakasira, sxmereddy, ravi.pendse}@wichita.edu

**This paper significantly extends the results that appeared in [1].
1. Introduction

Sensor networks have profound implications in the areas of environmental surveillance and monitoring, health-care and defense. For long term monitoring of the targeted environments, sensors are envisioned to be deployed with rechargeable batteries, which are capable of harnessing energy from renewable sources in the environment. For instance, solar energy-harvesting platforms such as Heliomote [2, 3] and Enhants [4], demonstrate the self-sustaining capability of a sensing device. These devices are not only extremely constrained in resources, particularly energy and computational power, but they are also expected to operate in the presence of uncertainties, and under dynamically changing and hostile environmental conditions. These factors necessitate the design of adaptive and distributed algorithms for efficient operations and management of sensor networks. Sensing devices capable of harvesting solar power [2, 3, 5, 6] utilize high availability of a renewable energy source to enable near-perpetual operation of the sensor network. One of the most important issues in the efficient operation of such sensor networks lies in the design of intelligent store-and-use energy-harvesting frameworks for energy management [7, 8, 9]. Design of energy-efficient algorithms for sensor operations is vital towards realization of such framework.

The performance achieved in a renewable energy based sensor system is measured using the quality of event detection and reporting attained in the system. The goal of this paper is to design a sensor activation algorithm which can provide performance guarantees in the presence of resource constraints and varying environmental conditions. Particularly, the algorithm should be implementable through low computational overhead, and in an online manner requiring only information currently available to the sensor node. In other words, the algorithm should not depend on global system parameters (such as energy harvesting rate and correlation probabilities) and should be robust to the variations of those parameters and other operational conditions.

Typically sensors are tiny, energy-constrained devices, with low recharge rates (dependent upon energy harvesting) and may have to spend a significant fraction of their time in inactive (sleep) state. The application specific events, which the sensor system is required to detect (and report), would occur randomly in the region of interest and could potentially exhibit temporal correlations across their occurrences. The overall objective is to maximize the time-average event detection probability achieved in the system. The
discharge of an active sensor depends on the activation algorithm as well as on the event occurrence process, while the recharge is based upon harnessing renewable energy sources. We address the following important question in a renewable energy based sensor system – How should a sensor be activated (deactivated) so as to maximize the overall event detection probability? Note that a dynamic algorithm is desired as predetermined scheduling is not feasible due to the randomness in the system.

In this paper, we use the sensor energy model first proposed in [10] to model renewable energy based sensor system. [10] also discusses various sensor activation algorithms in the presence of temporal correlations in the event phenomena. Particularly, it shows that a correlation-dependent wakeup policy, which employs a time-invariant sleep interval derived appropriately using system parameters, achieves near-optimal performance. However, this activation algorithm depends on all global system parameters (discussed in detail in Section 3), which in practice, the sensor may not have knowledge about, and may not be able to estimate with sufficient ease and accuracy. Also these parameters could change over time, due to the variability in environmental conditions. Hence, there is a need to design efficient algorithms for sensor operations which could easily be deployed in practice.

In this paper, we propose an activation algorithm, which does not depend on global system parameters, and is still able to achieve performance close to optimal. We borrow insights from TCP congestion avoidance algorithms which allow the end host to vary the congestion window size dynamically, probing the level of uncertainty (network congestion) in the process. It has been shown [11] that additive increase and multiplicative decrease (AIMD) of congestion window size at the end hosts leads to the best performance amongst the linear increase-decrease algorithms, in terms of throughput as well as fairness. In wired networks, the feedback or signal to trigger window increase or decrease could be the binary feedback from the network or the packet losses or the queue backlogs at the individual nodes. Similarly, in a renewable energy based sensor system, the sensor’s current energy level is a strong indicator of whether the sensor should act aggressively (by decreasing) or conservatively by increasing its subsequent sleep interval. We design an algorithm which varies its sleep interval dynamically based upon the sensor’s current energy level, and show that the proposed algorithm performs close to optimal. Since the algorithm does not rely on global systems parameters, and its decisions are based solely upon the sensor’s current energy level, it can easily be implemented in practice.
The main contributions of this paper include:

- Proposed design of adaptive algorithms for sensor activation to maximize the event detection probability in the presence of temporally correlated event phenomena,
- Performance analysis of additive increase multiplicative decrease based adaptive algorithm,
- Comparison of various linear increase/decrease schemes employed to vary the sensor’s subsequent sleep interval,
- Investigation of the performance of proposed algorithm through extensive simulations to demonstrate its near-optimality, stability and feasibility towards practical deployment, and
- Validation of the applicability of proposed algorithm in the presence of multiple sensors and multiple event processes.

This paper is organized as follows. Section 2 discusses related work in activation algorithms in renewable energy based sensor systems, and in increase/decrease algorithms for congestion avoidance. Section 3 describes sensor system components and their modeling, and also formulates the problem in terms of event detection probability. Section 4 discusses the design of adaptive algorithms for sensor activation. We evaluate the performance of proposed algorithm in Section 5 and discuss simulation results in Section 6. Section 7 discusses the performance of proposed algorithm in a distributed system with multiple sensors and multiple event processes. We summarize the conclusions in Section 8.

2. Related Work

[12, 13, 14] have considered the dynamic activation question in the context of energy harvesting sensor systems. [12] considers the single sensor, and modeling it as a closed three-queue system, obtains Norton’s equivalent of the system to evaluate the structure of the optimal rate control policy. [13] considers the sensor activation problem under a sensor energy model similar to [15], and demonstrates the optimality of threshold based policies for a broad class of utility functions and state dynamics. [14] considers the resource allocation problem in presence of rechargeable nodes, proposes a policy which
decouples admission control and power allocation decisions, and shows that it achieves asymptotically optimal performance for sufficiently large battery capacity to maximum power ratio.

Sensor node activation algorithms for rechargeable sensor systems in the presence of temporally correlated event phenomena have been previously considered in [10]. Particularly, [10] proposes a correlation-dependent wakeup (CW) policy, wherein the sensor employs a deterministic sleep interval which is derived using energy balance during a renewal interval of sensor operation. The proposed algorithm is shown to achieve performance close to optimal. However, this activation algorithm depends explicitly on global system parameters, which in practice, the sensor may not have accurate knowledge about. Moreover, these parameters may not be easy to estimate in practice, particularly because they are susceptible to sudden changes due to the variability in environmental conditions. Therefore, in this paper we focus on the design of algorithms which only rely on the available information such as sensor’s current energy level in making activation decisions.

[11] considered different linear increase and decrease algorithms for congestion avoidance in computer networks, and showed that additive increase with multiplicative decrease (AIMD) of congestion window converges to an efficient and fair state regardless of the starting state of the network. AIMD based approach has been previously applied in sensor networks for power control [16], for achieving fair wake-up rates among equivalent nodes [17], and for low power listening [18]. In this paper, however, we design an algorithm for a sensor node, wherein the sensor varies its subsequent sleep interval according to additive increase and multiplicative decrease, depending upon the sensor’s current energy level.

3. Sensor System and Components

In this section, we model the renewable energy based sensor system, the temporally correlated event phenomena and formulate the problem in terms of event detection probability. The energy bucket of a sensor stores energy in units of a quantum. A discrete time model is assumed such that in each time slot, a recharge event occurs with a probability $q$ and charges the sensor with a charge of $c$ quanta. The size of the sensor energy bucket is denoted by $K$.

The discharge process at the sensor depends upon its (activation) state, as well as on the state of the event phenomena. The sensor having non-
zero energy level is said to be in active (sleep) state if it has been activated (deactivated) in the current time slot. The sensor discharges energy only when in active state. The sensor expends a charge amount of $\delta_1$ quanta (operational cost) during each time slot it is active. In addition, if an event occurs and is detected by the active sensor, the sensor expends an additional charge amount of $\delta_2$ quanta (detection and transmission cost). We assume that if the sensor is active during time slot $t$, and an event occurs during this time slot, the event is detected (w.p. 1). Note that the model and results presented here could be generalized to take detection probability into account. We also assume that the sensor is available for activation as long as it has sufficient energy to operate for at least one time slot, i.e., its energy level is at least $\delta_1 + \delta_2$. In addition, we assume that the average recharge rate is less than the discharge rate of the sensor in active state i.e. $qc \leq \delta_1$.

We model the event phenomena, which the sensor system is required to detect and report, as a correlated stochastic process in order to characterize the inherent randomness and temporally correlated event occurrence. The extent of temporal correlations is specified using correlation probabilities $p_{on}^c$ and $p_{off}^c$, such that $\frac{1}{2} \leq p_{on}^c, p_{off}^c \leq 1$. If an event occurs during time slot $t$, then in the next time slot ($t+1$) a similar event occurs with probability $p_{on}^c$, while no such event occurs with probability $1 - p_{on}^c$. Similarly, if no event occurred during the current time slot, no such event occurs in the next time slot with probability $p_{off}^c$. The event occurrence process used to model the event phenomena comprises of an alternating sequence of periods...
where events occur (On period) and do not occur (Off period). In most application scenarios, events of interest occur rarely, therefore the Off periods are expected to be significantly larger than the On periods (i.e., $p_{c}^{\text{off}} \geq p_{c}^{\text{on}}$). However, we do not make this assumption in our analysis. Figure 1 depicts the energy discharge-recharge model of a typical sensor, where the recharge rate depends on $q, c$ and the discharge rate depends on $\delta_1, \delta_2$, the state of event process and the activation algorithm.

The objective is to maximize the asymptotic event detection probability achieved in the system. Let $E_o(T)$ denote the total number of events that occur in the region during time interval $[0\ldots T]$. Let $E_d(T)$ denote the total number of event detected during this interval by sensor operating under the activation algorithm $\Pi$. The asymptotic event detection probability, $U(\Pi)$, is the time-average fraction of events detected, i.e.,

$$U(\Pi) = \lim_{T \to \infty} \frac{E_d(T)}{E_o(T)}.$$  

(1)

The decision problem is that of finding the activation algorithm $\hat{\Pi}$ such that $\hat{\Pi} = \text{arg max } U(\Pi)$. In this paper, we design an activation algorithm for a sensor node, which takes dynamic decisions (when to stay active, and when to sleep, and for how long), which are independent of global system parameters including the temporal correlation probabilities, and is able to achieve near optimal performance.

4. Design of Adaptive Algorithms

We first discuss the design of a class of adaptive algorithms for sensor activation in a renewable energy based sensor system. Later, we compare the performance of the algorithms in this class with that of the Energy Balancing Correlation-dependent Wakeup (EB-CW) policy, which has been shown to perform near-optimally [10]. The CW algorithm satisfies the following criteria -

- Active sensor with sufficient energy remains active if an event occurred during the previous time slot,

- Active sensor goes to sleep (for an appropriately derived sleep interval) if no event occurred during the previous time slot or if it has insufficient energy, and
• The sensor activates itself at the end of the sleep interval, if it has sufficient energy ($\geq \delta_1 + \delta_2$).

For the sensor to employ EB-CW algorithm, it must have the knowledge of all global system parameters ($\delta_1$, $\delta_2$, $q$, $c$, $p^{on}_c$, and $p^{off}_c$). For EB-CW, the sleep interval is derived using energy balance during a renewal interval of sensor operation [10], and is given by,

$$SI_{EB-CW} \approx \frac{\delta_1}{qc} \left[ \frac{1}{qc} \left( \frac{1}{2 \cdot (1 - p^{on}_c - p^{off}_c)} \right) \right] - \frac{2}{qc} - 1.$$  (2)

In practice, it may not be possible for the sensor to know or estimate the value of these system parameters. We propose a class of adaptive and online algorithms, which do not depend on the knowledge of these system parameters, and are able to dynamically converge to the behavior of EB-CW algorithm, thus achieving near-optimal performance.

This class of activation algorithms, denoted $\Pi (c_1, c_2)$, satisfies all of the criteria described above for the CW policy. However, each time the sensor goes to sleep, it calculates its new sleep interval based upon the sleep interval previously used, and upon its current energy level. If the sensor’s current energy level is $< \frac{K}{2}$, it acts conservatively by choosing its subsequent sleep interval to be larger than its previously employed sleep interval. Otherwise, the sensor acts aggressively by choosing its subsequent sleep interval to be smaller than its previously employed sleep interval. We consider only linear (additive or multiplicative) increase and decrease algorithms for subsequent sleep interval computation.\(^1\)

Let $SI_{prev}$ denote the sleep interval last employed by the sensor. And let $SI_{next}$ denote the sleep interval to be employed the next time sensor decides to go to sleep. If the sensor’s current energy level $L < \frac{K}{2}$, it increases its sleep interval as -

• **Additive Increase**: $SI_{next} = SI_{prev} + c_1$, or

• **Multiplicative Increase**: $SI_{next} = SI_{prev} \cdot c_1$.

If $L \geq \frac{K}{2}$, the sensor decreases its sleep interval as -

• **Additive Decrease**: $SI_{next} = SI_{prev} - c_2$, or

\(^1\)The choice of $\frac{K}{2}$ is arbitrary, and other similar choices are also possible.
- **Multiplicative Decrease**: \( S_{\text{next}} = S_{\text{prev}} \cdot \frac{1}{c_2} \).

Considering all possible combinations of the increase and decrease functions results in the following algorithms - (a) Additive Increase Multiplicative Decrease (AIMD), (b) Additive Increase Additive Decrease (AIAD), (c) Multiplicative Increase Multiplicative Decrease (MIMD), and (d) Multiplicative Increase Additive Decrease (MIAD).

The additive parameter could take any value \( > 0 \), while the multiplicative parameter could take any value \( > 1 \). Other hybrid algorithms are also feasible, such as Multiplicative Additive Increase Multiplicative Decrease (MAIMD), wherein the increase of sleep interval is multiplicative if sensor’s current energy level \( < \frac{K}{4} \) (say), and is additive if sensor’s current energy level lies in \([\frac{K}{4}, \frac{K}{2}]\). Similarly, MAIMAD employs multiplicative as well as additive increase and decrease functionalities, with multiplicative decrease employed if the sensor’s current energy level lies in \([\frac{3K}{4}, K]\).

**Algorithm 1**: Adaptive Sleep Interval Computation for the AIMD Algorithm

```
Input: \( S_{\text{prev}}, L, K \)
Output: \( S_{\text{next}} \)

if \( L < \frac{K}{2} \) then
    \( S_{\text{next}} = S_{\text{prev}} + 1; \)
else
    \( S_{\text{next}} = \frac{S_{\text{prev}}}{2}; \)
end if
```

We show in Section 6 that the AIMD algorithm with \( c_1 = 1 \) and \( c_2 = 2 \) achieves the best performance among all the algorithms in the class \( \Pi(c_1, c_2) \). This is because this AIMD algorithm leads to the fastest and most stable convergence of the sensor’s energy level and sleep interval to the optimal operating region, resulting in the best performance. This result is interesting in lieu of the facts shown in literature [19, 20] which suggest that the general AIMD schemes for congestion avoidance are TCP-friendly for the \((c_1, c_2)\) values given by \((1, 2)\). Algorithm 1 describes this AIMD activation algorithm. Since the sensor could be easily configured to store the values required as input to the algorithm \((S_{\text{prev}}, L, \text{and } K)\), this algorithm is easily deployable in practice.
5. Performance Analysis

We analyze the performance of AIMD algorithm using discrete time Markov chains. The state of the sensor system at time slot \( t \) is represented as \((E, SD)\), where \( E \) denotes the state of event process and \( SD \) denotes the remaining sleep duration (in terms of number of time slots) of the sensor. If an event occurs during time slot \( t \), \( E = 1 \), otherwise \( E = 0 \). Note that \( SD = 0 \) implies that the sensor is active during time slot \( t \). The system state evolution for the sensor operating under the AIMD algorithm follows a discrete time Markov chain, as depicted in Figure 2. Note that the sensor chooses a different sleep interval each time it goes to sleep under the AIMD algorithm, and therefore the value of \( SI \) in Figure 2 would be different each time the sensor goes to sleep. In other words, every time the system enters state \((0,0)\), a new value of \( SI \) is computed. Also note that we are assuming that the sensor’s energy level is always \( \geq \delta_1 + \delta_2 \), and that the sensor never dies. This is a reasonable assumption given the fact that the sensor’s energy level always remains close to \( K/2 \) operating under the AIMD algorithm, as we show in Section 6. We first analyze the performance of the algorithm for a fixed value of \( SI \) in Section 5.1, and then analyze the AIMD algorithm and show its near-optimal performance in Section 5.2.

5.1. Performance computation for a given sleep interval

Solving for the steady-state probability distribution for the Markov chain in Figure 2, we get,

\[
\pi(1,0) = \frac{p_c^{\text{on}}}{1 - p_c^{\text{on}}} \pi(1,1) + \frac{1 - p_c^{\text{off}}}{1 - p_c^{\text{on}}} \pi(0,1). \tag{3}
\]

\[
\pi(0,0) = (1 - p_c^{\text{on}})(\pi(1,1) + \pi(1,0)) + p_c^{\text{off}} \pi(0,1). \tag{4}
\]

\[
\pi(0,SI) = p_c^{\text{off}} \pi(0,0) \quad \text{and} \quad \pi(1,SI) = (1 - p_c^{\text{off}}) \pi(0,0). \tag{5}
\]

For \( k \in [1 \ldots (SI - 1)] \),

\[
\pi(0,k) = p_c^{\text{off}} \pi(0,k+1) + (1 - p_c^{\text{on}}) \pi(1,k+1) \quad \text{and} \quad \pi(1,k) = (1 - p_c^{\text{off}}) \pi(0,k+1) + p_c^{\text{on}} \pi(1,k+1). \tag{6}
\]

Also note that, \( \forall k \in [1 \ldots SI] \),

\[
\pi(0,k) + \pi(1,k) = \pi(0,0). \tag{7}
\]
Figure 2: Discrete time Markov chain for system state evolution under the AIMD algorithm.

The above equation is true for \( k = SI \) from (5), and is true recursively for \( k < SI \) using (6). Using (6), (7), and substituting \( x = p^\text{on}_c + p^\text{off}_c - 1 \), \( y = 1 - p^\text{off}_c \), \( z = 1 - p^\text{on}_c \), we get, \( \forall k \in [1 \ldots (SI - 1)] \),

\[
\pi(1,k) = (1 - p^\text{off}_c)\pi(0,k+1) + p^\text{on}_c\pi(1,k+1)
= (1 - p^\text{off}_c)\left(\pi(0,0) - \pi(1,k+1)\right) + p^\text{on}_c\pi(1,k+1)
= x\pi(k+1) + y\pi(0,0).
\] (8)

Therefore,

\[
\pi(1,1) = x\pi(1,2) + y\pi(0,0) = x\left(x\pi(1,3) + y\pi(0,0)\right) + y\pi(0,0)
= x^2\pi(1,3) + (x + 1)y\pi(0,0) = ...
= x^{SI-1}\pi(1,SI) + \left(x^{SI-2} + x^{SI-3} + \ldots + x^2 + x + 1\right)y\pi(0,0)
= x^{SI-1}\pi(1,SI) + \frac{1 - x^{SI-1}}{1 - x}y\pi(0,0) = x^{SI-1}y\pi(0,0) + \frac{1 - x^{SI-1}}{1 - x}y\pi(0,0)
= \frac{1 - x^SI}{1 - x}y\pi(0,0).
\] (9)
Using equations (8) and (9), we get,

\[ \pi_{(1,k)} = \frac{1 - x^{SI-(k-1)}}{1 - x} y \pi_{(0,0)}, \quad \forall k \in [1 \ldots SI]. \] (10)

Using equations (7) and (10), we get (recall that \( x + y + z = 1 \)),

\[ \pi_{(0,k)} = \frac{z + y x^{SI-(k-1)}}{1 - x} \pi_{(0,0)}, \quad \forall k \in [1 \ldots SI]. \] (11)

Using equations (3), (10) and (11), we get,

\[ \pi_{(1,0)} = \frac{y}{(1 - x)z} (1 - x^{SI+1}) \pi_{(0,0)}. \] (12)

Using \( \sum_{i,j} \pi_{(i,j)} = 1 \), we get,

\[ \pi_{(0,0)} = \left[ (SI + 1) + \frac{y}{z(1 - x)} (1 - x^{SI+1}) \right]^{-1}. \] (13)

From (10), we have,

\[ \sum_{i=1}^{SI} \pi_{(1,i)} = (x^{SI+1} - (SI + 1)x + SI) \frac{y}{(1 - x)^2} \pi_{(0,0)}. \] (14)

From Figure 2, the steady-state probability of event occurrence is given by \( \sum_{i=0}^{SI} \pi_{(1,i)} \). Using equations (12), (14) and simplifying, we get,

\[ \sum_{i=0}^{SI} \pi_{(1,i)} = \pi_{(1,0)} + \sum_{i=1}^{SI} \pi_{(1,k)} = \frac{y}{1 - x}. \] (15)

The performance of the system is measured in terms of event detection probability given by (1). In terms of steady-state probabilities, the system performance for a sensor operating under a constant sleep interval of \( SI \) can be expressed as,

\[ U(\Pi) = \frac{\pi_{(1,0)}}{\sum_{i=0}^{SI} \pi_{(1,i)}}. \] (16)

Here the numerator represents the steady-state probability of event detection and the denominator represents the steady-state probability of event occurrence in the system. Using equations (15) and (16), we get,
\[ U(\Pi) = \frac{(1 - x^{SI+1})(1 - x)}{y(1 - x^{SI+1}) + z(1 - x)(SI + 1)}. \] (17)

In the absence of temporal correlations (when \(p^\text{on}_c = p^\text{off}_c = 0.5\), we have \(x = 0, y = 0.5, z = 0.5\), and the system performance is given by \( U(\Pi) = \frac{1}{2 x^{SI}} \). As the sleep interval increases, the system performance decreases. However, note that the choice of \(SI = 0\) is not feasible since it would lead the sensor to lose all its energy and put it in dead state sooner or later, and hence the above Markov chain analysis would not apply. In fact, the system performance given by (17) is valid only if the sleep interval \(SI \geq SI_{EB-CW}\), given by (2). For sufficiently large sleep interval \(SI \gg 0\) and \(0 < x < 1, x^{SI+1} \to 0\), and the system performance approaches,

\[ U(\Pi) \approx \frac{1}{1 + zSI - \frac{zx}{1-x}}. \] (18)

Putting the value of sleep interval \(SI_{EB-CW}\) given by (2), in (18), we shall show that the EB-CW algorithm achieves the maximum achievable performance for any CW algorithm, which is in line with the results (Lemma 4 and Theorem 2) in [10].

From [10], the maximum achievable performance of any CW algorithm \(\Pi_{CW}\) is upper-bounded by \(U_{CW}\) which is given by,

\[ U_{CW} = \frac{qc}{(\delta_1 + \delta_2) \left( \frac{y}{1-x} \right) + \delta_1 z}. \] (19)

The performance of EB-CW policy \(\Pi_{EB-CW}\) using equations (2) and (18) is given by,

\[
U(\Pi_{EB-CW}) \approx \frac{1}{1 + zSI_{EB-CW} - \frac{zx}{1-x}} \\
= \frac{1}{1 - \frac{zx}{1-x} + z \left[ \frac{\delta_1}{qc} + \frac{y(\delta_1 + \delta_2 - qc)}{qc(1-x)z} \right] - 1} \\
= \frac{1}{1 - z - \frac{zx}{1-x} - \frac{y}{1-x} + \frac{\delta_1}{qc} + \frac{y(\delta_1 + \delta_2 - qc)}{qc(1-x)}} \\
= \frac{1}{1 - z - \frac{zx}{1-x} - \frac{y}{1-x} + \frac{\delta_1}{qc} + \frac{y(\delta_1 + \delta_2)}{qc(1-x)}}
\]
5.2 Near-optimal performance of AIMD algorithm

According to AIMD algorithm, the sleep interval is computed each time the system reaches state (0, 0) in Figure 2. Let $L$ denote the current energy level of the sensor in state (0, 0). Then, the next sleep interval is computed as,

$$SI_{next} = \begin{cases} SI_{prev} + 1, & \text{if } L < K/2, \\ SI_{prev}/2, & \text{else} \end{cases}$$

Thereafter, the system state evolution follows the Markov chain depicted in Figure 2 with $SI = SI_{next}$. Thus, the system evolution under AIMD algorithm can be represented using Figure 3. Let $R_i$ denote the amount of energy gained by the sensor through recharge during the time spent in block $i$. Similarly, let $D_i$ denote the amount of energy spent by the sensor in block $i$. Let $N_i$ denote the net energy gain in block $i$, i.e. $N_i = R_i - D_i$. Let time $T$ denote the total time spent by the system in blocks $1 \ldots j$. Then the expected cumulative net energy gain $N_c$ during time interval $[0 \ldots T]$ is given by $N_c = \sum_{i=1}^{j} E[N_i]$. Assuming that the sensor’s initial energy reserve equals $K/2$, the condition that during a time interval the expected cumulative net energy gain $> 0$, is equivalent to the fact that at the end of that time interval, the expected energy level of sensor $> K/2$. Using the
expected cumulative net energy gain $N^e$, the sleep interval for block $(j + 1)$ operating under the AIMD algorithm can be computed as -

$$
\text{If } N^e < 0, \quad SI_{next} = SI_{prev} + 1, \quad \text{else } SI_{next} = SI_{prev}/2.
$$

(22)

Note that we use the above method of computation since keeping track of sensor's current energy level $L$ at all times is more cumbersome than computing the expected net energy gain in each block. Nevertheless, the computation above accurately captures the dynamics of the system operating under the AIMD algorithm.

Let the probability that the system state equals $(1, 0)$ when the sensor wakes up (i.e. when the system leaves state $(0, 1)$ or $(1, 1)$) be denoted $\hat{p}$. Then, the expected duration of time spent by the system in block $i$ when sleep interval $SI$ is employed, denoted $T_i$ equals,

$$
T_i = 1 + SI + \hat{p} \frac{1}{1 - p_{\text{on}}^e}.
$$

(23)

The system spends one time slot in state $(0, 0)$, $SI$ time slots in sleep states of the form $(s,t)$, $s \in \{0, 1\}, 1 \leq t \leq SI$, and expected $\frac{1}{1 - p_{\text{on}}^e}$ time slots in state $(1,0)$. The expected amount of recharge gained in block $i$ equals

$$
E[R_i] = qcT_i = qc \left(1 + SI + \hat{p} \frac{1}{1 - p_{\text{on}}^e}\right).
$$

(24)

The expected amount of discharge in block $i$ equals

$$
E[D_i] = \hat{p}(\delta_1 + \delta_2) + \delta_1.
$$

(25)

Therefore, the expected net energy gained in block $i$ equals

$$
E[N_i] = E[R_i] - E[D_i] = qc(1 + SI) - \delta_1 - \hat{p}(\delta_1 + \delta_2 - qc) \frac{1}{1 - p_{\text{on}}^e}.
$$

(26)

Let $X_t$ denote the state of the system at time $t$. Then the probability $\hat{p}$ can be expressed as

$$
\hat{p} = \Pr[X_t = (1,0)|X_{t-SI-1} = (0,0)].
$$

(27)
Let \( p(i, j) = \Pr [X_t = (i, j) | X_{t-SI-1+j} = (0, 0)] \) \( \forall i \in \{0, 1\}, 0 \leq j \leq SI \). Then \( \hat{p} = p(1, 0) = 1 - p(0, 0) \). Note that \( p(0, j) + p(1, j) = 1 \) \( \forall j : 0 \leq j \leq SI \). Now, from Figure 2, \( p(0, j) \) and \( p(1, j) \) can be recursively expressed \( \forall j : 0 \leq j < SI \) as

\[
p(0, j) = p(0, j+1) \frac{p_{off}}{1 - p_{off}} + p(1, j+1) (1 - p_{off})
\]

\[
p(1, j) = p(1, j+1) \frac{p_{on}}{1 - p_{on}} + p(0, j+1) (1 - p_{on})
\]

(28)

Therefore, \( \hat{p} \) equals,

\[
\hat{p} = p(0, 1) = xp(1, 1) + y = x (xp(1, 2) + y) + y
\]

\[
= x^2 p(1, 2) + y (1 + x) = \ldots = x^{SI} p(1, SI) + y \left( 1 + x + x^2 + \ldots + x^{SI-1} \right)
\]

\[
= x^{SI} (1 - p_{off}) + y \left( 1 + x + x^2 + \ldots + x^{SI-1} \right)
\]

\[
= \frac{y}{1 - x}
\]

(29)

For sufficiently large values of \( SI \) and since \( 0 < x < 1 \), \( x^{SI+1} \rightarrow 0 \) and \( \hat{p} \rightarrow \frac{y}{1-x} \). From (26), the expected net energy gained in block \( i \) is given by

\[
E[N_i] = qc(1 + SI) - \delta_1 - \frac{y (\delta_1 + \delta_2 - qc)}{z(1 - x)}.
\]

(30)

It is worth noting here that if \( SI = SI_{EB-CW} \), the expected net energy gained \( E[N_i] = 0 \).

The state of the system at each block \( i \) in Figure 3 is represented as \( X'_i = (N'_c, SI_i) \), where \( N'_c \) denotes the expected cumulative net energy gained before the block \( i \) is entered (i.e. \( N'_c = \sum_{j=1}^{i-1} E[N_j] \)), and \( SI_i \) denotes the sleep interval value used in block \( i \). The next state \( X'_{i+1} = (N'_{c+1}, SI_{i+1}) \) at block \( i + 1 \) is computed using (30) and (22) as

\[
N'_{c+1} = N'_c + E[N_i] \quad \text{and} \quad SI_{i+1} = SI_i + 1 \quad \text{if} \quad N'_c < 0; SI_i / 2 \quad \text{otherwise}.
\]

(31)
Thus starting from an arbitrary initial state (e.g., $X'_0 = (0, 0)$), the state of the system at each block is computed. The performance achieved in block $i$ is computed using (18) by substituting $SI = SI_i$, and is denoted $P_i$. The performance of the AIMD algorithm is computed as

$$U(\Pi_{AIMD}) = \lim_{N \to \infty} \frac{\sum_{i=1}^{N} P_i T_i}{\sum_{i=1}^{N} T_i}. \quad (32)$$

Here $T_i$ is the expected duration of time spent by the system in block $i$, given by (23).

**Theorem 1.** AIMD algorithm achieves near-optimal performance, i.e., $U(\Pi_{AIMD}) \approx U_{CW} \approx U(\Pi_{EB-CW})$.

**Proof.** Let us consider the sensor system operating under the AIMD algorithm at steady-state. Let the distinct sleep intervals employed in different blocks (in Figure 3) during a cycle of operation be denoted $SI_1, SI_2, \ldots, SI_n$. Note that the value of $n$ is arbitrary. In this cycle, the sleep interval is increased additively in the first $n-1$ blocks, i.e., $SI_2 = SI_1 + 1$, $SI_3 = SI_2 + 1$, \ldots, $SI_n = SI_{n-1} + 1$, since the expected cumulative net energy gained in blocks $1, \ldots, (n-1)$ is less than zero. And the sleep interval is decreased multiplicatively in block $n$, i.e., $SI_{n+1} = \frac{SI_n}{2}$, since the expected cumulative net energy gained in block $n$ is greater than 0. Now, if $SI_{n+1} = SI_1$, and if the expected cumulative net energy gained in block $n+1$ is less than zero, this cycle of system operation repeats itself over time. We show that such a cycle exists.

We have, $SI_n = SI_1 + n - 1 = 2SI_1$, which implies, $SI_1 = n - 1$. Using (30), the expected cumulative net energy gained in block $n$ is given by,

$$N_{n+1}^c = \sum_{i=1}^{n} E[N_i] = \left[ qc - \delta_1 - \frac{y(\delta_1 + \delta_2 - qc)}{z(1-x)} \right] n + qc \sum_{i=1}^{n} SI_i$$

$$= \left[ qc - \delta_1 - \frac{y(\delta_1 + \delta_2 - qc)}{z(1-x)} \right] n + \frac{3}{2} (n-1) nqc. \quad (33)$$

We want $N_{n+1}^c = \epsilon$, for some $\epsilon > 0$. Assuming $\epsilon$ is arbitrarily close to 0, and equating $N_{n+1}^c = 0$, we get,

$$n - 1 = \frac{2}{3} \left[ \frac{\delta_1}{qc} + \frac{y(\delta_1 + \delta_2 - qc)}{qc z(1-x)} - 1 \right] = \frac{2}{3} SI_{EB-CW}. \quad (34)$$
Thus, $SI_1 = \frac{2}{3}SE_{EB-CW}$, and $SI_n = \frac{2}{3}SE_{EB-CW}$.\(^2\)

Next, we show that for blocks $1, \ldots, (n - 1)$, the expected cumulative net energy gained is less than zero. Using (30), $\forall i \in \{1, \ldots, n - 1\}$, we have,

$$E[N_i] = \left[qc - \delta_1 - \frac{y(\delta_1 + \delta_2 - qc)}{z(1 - x)}\right] + qcSI_i = qc (SI_i - SE_{EB-CW}).$$

(35)

We have, $N^c_2 = N^c_1 + E[N_1] = N^c_{n+1} + E[N_1] = E[N_1] < 0$, since $SI_1 = \frac{2}{3}SE_{EB-CW} < SE_{EB-CW}$. Using (35), and the fact that $SE_{EB-CW} = \frac{3}{2}(n - 1), \forall i \in \{2, \ldots, n\}$, we have,

$$N^c_i = \sum_{j=1}^{i-1} E[N_j] = qc \sum_{j=1}^{i-1} SI_j - \frac{3}{2}qc(i - 1)(n - 1)$$

$$= qc(i - 1)(n - 2 + \frac{i}{2}) - \frac{3}{2}qc(i - 1)(n - 1) < 0.$$  

(36)

The inequality above holds since $n - 2 + \frac{i}{2} < \frac{3}{2}(n - 1)$ for all $i \in \{2, \ldots, n\}$. Thus this cycle would repeat itself in steady-state. The performance of the AIMD algorithm is the same as the performance achieved during this cycle, and is given by,

$$U(\Pi_{AIMD}) = \frac{\sum_{i=1}^{n} P_i T_i}{\sum_{i=1}^{n} T_i},$$

(37)

where $P_i \approx \frac{1}{1 + zSI_i - \frac{y}{z(1-x)}}$ and $T_i = 1 + SI_i + \frac{y}{z(1-x)}$. Using, $\sum_{i=1}^{n} SI_i = \frac{3n(n - 1)}{2}$, we have,

$$\sum_{i=1}^{n} T_i = n \left[1 + \frac{y}{z(1-x)} + \frac{3(n - 1)}{2}\right].$$

(38)

Using, $z \left[1 + \frac{y}{z(1-x)}\right] = 1 - \frac{zx}{1-x}$, we have,

$$\sum_{i=1}^{n} P_i T_i \approx \sum_{i=1}^{n} \frac{1 + SI_i + \frac{y}{z(1-x)}}{1 + zSI_i - \frac{zx}{1-x}} = \frac{n}{z}.$$  

(39)

\(^2\)Note that the values $SI_1$ and $SI_n$ may not be integers in practice, and hence the steady state behavior may differ from the cycle described here.
Simplifying, and using (19), we get,

\[ U(\Pi_{AIMD}) \approx \frac{n}{z} \left[ 1 + \frac{y}{z(1-x)} + \frac{3(n-1)}{2} \right] = \frac{1}{z + \frac{y}{1-x} + \frac{3z(n-1)}{2}} = \frac{1}{z(1 + S_{EB-CW})} + \frac{y}{1-x} = \frac{qc}{z\delta_1 + \frac{y(\delta_1 + \delta_2)}{1-x}} = U_{CW}. \] (40)

Next, we study the performance of the various activation algorithms outlined in Section 4 using extensive simulations. For comparison, we also consider AW (Aggressive Wakeup) algorithm [10], which tries to activate the sensor whenever possible, i.e., as long as the sensor’s energy level \( \geq \delta_1 + \delta_2 \).

6. Simulation Results

We study the performance of different activation algorithms using discrete event simulation [21] of the sensor system, where the system performance is computed using (1). The simulation code is written in C language and can be found in [22], and the results are obtained using Linux i686 kernel release 2.6.26. The parameters used are \( q = 0.5, c = 1, \delta_1 = 1, \delta_2 = 6, \) and \( K = 3000 \). The initial energy level of the sensor is assumed to be zero. In all the experiments, the values of increase/decrease parameters are chosen such that the additive parameter equals 1 and the multiplicative parameter equals 2, and the correlation probabilities are given by \( p^{on}_c = 0.7, \) and \( p^{off}_c = 0.9, \) unless stated otherwise. We observe similar performance trends with other choices of system parameters as well. We observe that the performance of the AIMD algorithm computed using (32) is very close to that obtained using simulations.

Figure 4 plots the steady-state performance of the four types of algorithms outlined in Section 4, with \( p^{off}_c = 0.9 \). As \( p^{on}_c \) increases from 0.5 to 0.9, more events occur during the interval \([0 \ldots T]\), and since the recharge system parameters are kept constant, the performance in terms of fraction of events detected decreases for all the algorithms. We observe that the AIMD algorithm achieves the best performance among these algorithms, closely followed by AIAD and MIMD, whereas MIAD achieves the worst performance.\(^3\)

\(^3\)Even though AIAD and MIMD perform well, the sensor behavior is more stable w.r.t.
Figure 4: Performance comparison of different adaptive algorithms.

Figure 5: Performance comparison of AIMD with other hybrids, AW and EB-CW algorithms.

Figure 5 compares the performance of some of the hybrid algorithms with that of AIMD, with $p_c^{off} = 0.9$. $U_{CW}$, and the performance of EB-CW and AW algorithms are also plotted for comparison. We observe that the AIMD algorithm performs very well compared to AW, and performs quite close to EB-CW. From the above figures, AIMD, AIAD, MIMD, MAIMD and MAIMAD perform very close to each other, when $p_c^{on} = 0.9$.

Figure 6 depicts the performance of AIMD algorithm for various values of increase and decrease parameters, $c_1$ and $c_2$. The performance is observed for values of $c_1 \in [1 \ldots 10]$, and $c_2 = 2^i \forall i \in [1 \ldots 10]$. We observe that $c_1 = 1$, and $c_2 = 2$ achieves the best performance among all AIMD algorithms. Figures 7 - 9 depict similar performance trends with respect to $c_1$ and $c_2$ for other adaptive algorithms as well. Particularly, additive (increase or decrease) parameter of 1, and multiplicative parameter of 2 achieve the best performance, except for algorithms with additive decrease (where $c_2 = 1$ is not the best choice). Additionally, the AIMD algorithm with $c_1 = 1$ and $c_2 = 2$ is observed to achieve the best performance among all the algorithms in the class $\hat{\Pi} (c_1, c_2)$.

Note that MIAD algorithm performance is much lower than that of the other algorithms. Since the initial energy level of the sensor is zero, and it takes a while for it to reach above $\frac{K}{2}$, the sensor operating under the energy level and sleep interval variations under AIMD.
MIAD algorithm increases its sleep interval to a very large value using a multiplicative increase of two each time. Even when the energy level becomes greater than the above threshold, the sleep interval is decremented slowly (additively by one each time). Therefore, the sensor spends most of its time in the sleep state, and detects a lower fraction of events. On the other hand, AIMD is able to perform better since the sleep interval of the sensor operating under AIMD converges to that of EB-CW faster, as we show next.

Figure 6: AIMD achieves the best performance with $c_1 = 1$ and $c_2 = 2$.

Figure 7: AIAD Performance with various values of $c_1, c_2$.

Figure 8: MIAD Performance with various values of $c_1, c_2$.

Figure 9: MIMD Performance with various values of $c_1, c_2$.

Figures 10 and 11 plot the sensor’s sleep interval over time operating under various activation algorithms. $SI_{EB-CW}$ (computed using (2)) is around
Figure 10 shows that the sleep interval values under various adaptive algorithms oscillate around (and try to converge to) $SI_{EB-CW}$, except for MIAD, where the sleep interval diverges to a very large value. Among these, the AIMD exhibits the fastest and more stable convergence. The sleep interval employed by the sensor operating under the AIMD activation algorithm oscillates around $SI_{EB-CW}$ in steady-state, as shown in Figure 11. Note that the AIMD algorithm is able to estimate the value of $SI_{EB-CW}$ automatically, by dynamically adapting the sleep interval in such a way so as to keep the sensor’s energy level close to $\frac{K}{2}$. As a result, sensor’s behavior operating under the AIMD algorithm approaches the behavior under EB-CW, resulting in near-optimal performance achieved by the AIMD algorithm.

Figure 11: Sleep interval oscillations with AIMD algorithm.

Figure 12 plots the energy level of the sensor operating under various activation algorithms. The energy level of the sensor under AIMD algorithm converges to around half of energy bucket size, i.e., in steady-state $L_{AIMD} \rightarrow \frac{K}{2}$. Energy level of the sensor operating under algorithms AIAD and MIMD also converge close to $\frac{K}{2}$, albeit after a longer time and with larger oscillations. This behavior is a direct consequence of the fact that all the adaptive algorithms increase the sensor’s sleep interval when $L < \frac{K}{2}$, and decrease it when $L \geq \frac{K}{2}$. Since the sensor’s energy level stabilizes close to $\frac{K}{2}$ quite rapidly operating under the AIMD algorithm, the use of hybrid algorithms like MAIMD and MAIMAD does not provide additional performance...
improvement over that of AIMD (as the sensor’s energy level never reaches close to 0 or $K$).

![Figure 12: Energy level evolution for different adaptive activation algorithms.](image1)

![Figure 13: Energy level under AIMD and CW algorithms in steady-state.](image2)

Figure 12 depicts the sensor’s energy level in steady-state for AIMD and EB-CW activation algorithms. Recall that $SI_{EB-CW} = 11$. We observe the energy level of the sensor for two different CW algorithms, one with $SI = 11$ (EB-CW), and another with $SI = 12$ (denoted $CW^1$). The sensor operating under AIMD activation algorithm maintains its energy level close to $\frac{K}{2}$, whereas under the CW algorithms the energy level of the sensor either stays close to full or close to empty, and also exhibits larger variations. This suggests that AIMD algorithm is able to stabilize the sensor’s energy level more than the CW algorithm.\(^4\)

Figure 13 depicts the stability of the sensor’s behavior, and the performance achieved, operating under AIMD activation algorithm, in comparison with the CW activation algorithm. The performance of CW algorithm is shown for different values of sleep intervals. The performance achieved by the AIMD algorithm is quite close to the performance of the best CW algorithm. The figure also depicts the steady-state energy levels (denoted $L^{ss}$)

\(^4\)Note that a probabilistic scheme could be used with CW algorithm to choose the sleep interval 11 w.p. $p$, and 12 w.p. $1 - p$, so as to stabilize the sensor’s energy level. However, such a scheme would still exhibit larger variations in energy level, and would also depend heavily on global system parameters.
of the sensor under the AIMD and CW algorithms. We observe that the energy level of the sensor under AIMD algorithm converges close to $K^2$, whereas under CW algorithms the energy level of the sensor is either close to $K$ or close to 0.

6.1. Comparison of EB-CW and AIMD Algorithms

AIMD algorithm performs well because it tries to quickly estimate and converge the sensor’s behavior to that under the EB-CW algorithm. Thereafter, the AIMD algorithm maintains the sensor’s sleep interval value oscillating around $SI_{EB-CW}$, and its energy level close to $K^2/2$. Moreover, the AIMD algorithm is more adaptive to dynamic changes in the environment. For instance, if one of the system parameters changes, AIMD algorithm would automatically converge to the new high performance and stability region, whereas the sensor operating under the EB-CW algorithm would need to estimate the new value of the changed system parameter, and recompute $SI_{EB-CW}$ in order to achieve good performance. In addition, the AIMD algorithm does not depend on the system parameters ( unlike EB-CW), and thus can easily be deployed in practice. The sensor’s behavior (with respect to energy level and sleep interval variations) operating under the AIMD algorithm is also more stable compared to the EB-CW and other adaptive algorithms.
7. Extensions to Multiple Sensors and Multiple Event Processes

In this section, the AIMD algorithm is extended to a distributed network scenario with multiple sensors and multiple event processes. A system with $N$ sensors, each having a circular coverage range of $r$, are deployed uniformly at random in a region of size 100x100. Sensors’ recharge processes are governed by parameters $q$ and $c$, and are assumed to be independent. $M$ independent event processes are located uniformly at random in the region, each with correlation probabilities $p_{c_{on}}^j$ and $p_{c_{off}}^j$. The system parameters used are $q = 0.5$, $c = 1$, $\delta_1 = 1$, $\delta_2 = 6$, $K = 3000$, $p_{c_{on}}^j = 0.7$, $p_{c_{off}}^j = 0.9$ and $r = 20$. An event that occurs during time slot $t$ is considered detected if there is at least one active sensor during time $t$ that covers the corresponding event process. If a sensor detects (and transmits) $k$ ($k > 1$) events during time slot $t$, it expends $k\delta_2$ energy quanta as detection cost during time $t$. The performance of the system operating under activation policy $\Pi$ is computed as follows. Let $E_{j_{on}}^j(T)$ denote the total number of events corresponding to event process $j$ that occur in the region during time interval $[0 \ldots T]$. Let $E_{j_{d}}^j(T)$ denote the total number of events detected corresponding to event process $j$ during this interval by the sensor system operating under the activation algorithm $\Pi$. The asymptotic event detection probability, $U_M(\Pi)$, is the time-average fraction of events detected over all event processes, i.e.,

$$U_M(\Pi) = \lim_{T \to \infty} \frac{\sum_{j=1}^{M} E_{j_{d}}^j(T)}{\sum_{j=1}^{M} E_{j_{on}}^j(T)}.$$  \hspace{1cm} (41)

We consider algorithms wherein the sensors take activation decisions in a completely distributed manner (without taking into account the decisions taken by other sensors). Thus, such algorithms involve minimum coordination and message overhead. AW, EB-CW and AIMD algorithms are simple extensions of the respective algorithms in the single sensor case. In case of EB-CW and AIMD algorithms, however, a sensor decides to go to sleep only if all the event processes in its coverage range are in the Off period. For AIMD algorithm, we consider $c_1 = 1$ and $c_2 = 2$.

Figures 15 - 17 depict the performance of various algorithms for various choices of $N$, $M$ and $r$. Each data point is shown along with 90% confidence interval. Note that sometimes the interval is too small to be depicted. Figure 15 depicts the performance of various algorithms with $M = 5$ event processes for a range of sensor densities. We observe that AIMD algorithm
performs better than both the AW and the EB-CW algorithms at all sensor densities. In order to achieve a desired coverage quality (e.g. ≥ 85%), only 100 sensors are needed when using the AIMD algorithm, while 250 or more sensors are needed when using the EB-CW or the AW algorithm. Figure 16 depicts the system performance under various algorithms with $N = 150$ sensors for various values of $M$. We observe that as the number of event processes increase, the system performance decreases. However, the decrease is more graceful when using the AIMD algorithm, when compared with AW and EB-CW algorithms. Figure 17 depicts the performance under varying coverage radius at the sensors. At large radius the performance of AW policy decreases significantly, however the performance decrease is minimal under AIMD algorithm. We observe that AIMD algorithm performs better than both EB-CW and AW under all scenarios considered. Since AIMD algorithm is based only upon local information, and performs well compared to AW and EB-CW algorithms, it is more suitable to deploy in a practical and distributed network scenario.

8. Summary and Conclusions

We have designed a class of online and adaptive activation algorithms for sensor operation in a renewable energy based sensor system. The performance of the sensor is measured in terms of asymptotic event detection.
probability achieved, in the presence of temporally correlated event phenomena. We have shown that an algorithm which increases sensor’s sleep interval conservatively (Additive Increase), and decreases the sleep interval aggressively (Multiplicative Decrease) performs close to optimal. In particular, our proposed AIMD algorithm not only achieves near-optimal performance, but also maintains a stable energy level for the sensor with finite energy bucket size. Since the AIMD algorithm does not rely on global system parameters, it is more adaptive to changes in these parameters, and is also more suitable for deployment. We validate the applicability of our proposed algorithm in a distributed network environment with multiple sensors and multiple event processes through simulations.

References


