# Research Article A Core Algorithm for Object Tracking and Monitoring via Distributed Wireless Sensor Networks

## E. A. Etellisi,<sup>1</sup> A. T. Burrell,<sup>2</sup> and P. Papantoni-Kazakos<sup>1</sup>

<sup>1</sup>Department of Electrical Engineering, University of Colorado Denver, Campus Box 110, P.O. Box 173364, Denver, CO 80217, USA <sup>2</sup>Computer Science Department, Oklahoma State University, 219 MSCS, Stillwater, OK 74078, USA Address correspondence to P. Papantoni-Kazakos, titsa.papantoni@ucdenver.edu

Received 7 May 2011; Revised 20 December 2011; Accepted 25 December 2011

Abstract We consider object tracking and monitoring implemented via the use of distributed wireless sensor networks. We view the signal processing and communications operations performed in such networks, in conjunction with the time constraints imposed on their signal processing objectives and the limited life-spans of their sensors. We identify some of the object identification and network monitoring functions that are embedded in such systems. We subsequently focus on a core algorithm whose various manifestations may serve effectively a variety of network functions, and propose its novel application for image and sound object tracking. We also propose a new distributed version of the algorithm for the monitoring of data rates in the network.

**Keywords** sequential detection of change; object identification; distributed network rate monitoring

## **1** Introduction

Object tracking and monitoring is currently extensively implemented by distributed wireless sensor networks whose architectures, operations, and performance demands are dictated by the tracking and monitoring objectives, but are also constrained by the characteristics and limitations of the environment. Within the statistical inference domain, the tracking and monitoring objectives are classified as either hypothesis testing (detection) or parameter estimation or estimation of the acting data generating process, and the pertinent performance criteria include decision/estimation accuracy and convergence rate, where detection/estimation accuracy is generally monotonically increasing with the number of observation data processed [26]. When time constraints are imposed on high accuracy detection/estimation, the consequence is increased required overall data rates. At the same time, in the distributed wireless networks considered, observation data may be collected and processed by life-limited nodes, whose life-span is a function of the data rates they process [1,2,5,6,15,16,21,22,23,33,38,40,39]. Thus, required

overall data rates, in conjunction with rate-dependent node life-spans, generally necessitate network-architecture and network-operations control/adaptations, so that possible nodes' survivability limitations do not interfere with the required network overall performance [15, 16, 23, 33, 40, 39]. Since the network-architecture and network-operations adaptations are functions of the acting data rates, it is eminent that data rates are then monitored and that rate changes are detected accurately and rapidly.

There has been a plethora of research efforts within the object tracking area, exemplified by the references [8,9, 17, 19, 20, 25, 35, 37, 41, 47], while related marginal issues are addressed in [20, 29], and where some recent results in object identification can be found in [18]. Concurrently, transmission and routing algorithms that are appropriate for the distributed wireless sensor network environment have been proposed in [11, 12, 14, 15, 16, 21, 33, 34, 48], while some other related network issues are addressed in [44, 45, 46, 49], and where network performance monitoring is considered in [31].

The operations performed within an Object Tracking/Monitoring (OTM) Distributed Wireless Sensor Network (DWSN) are many and diverse, and include signal processing as well as communications algorithms. In this paper, we focus on a core algorithm whose various implementations may be deployed at various components of the OTM-DWSN operations. We present and analyze a distributed version of the algorithm for data rate monitoring.

The organization of the paper is as follows: in Section 2, we present some related work. In Section 3, we present the general OTM-DWSN model considered and discuss some of its operations. In Section 4, we present the core algorithm and summarize its properties. In Section 5, we present some implementations of the core algorithm within the OTM-DWSN system, including a new distributed algorithmic form for network data rate monitoring. In Section 6, we draw some conclusions.

In this paper, we focus on a core algorithm and present several of its manifestations and applications, including audio object tracking, object identification from images, and distributed network traffic monitoring. For audio tracking of an object, the audio signature of the object must be monitored and detected accurately when present, while embedded in noise. To attain this objective, the noise and the audio signature of the object must be first modeled. Based on the latter model, algorithms that detect accurately changes from absence to presence of the object audio and vice versa must be designed and deployed. Several researches have been invested in this area [42,43]. While [43] presents a statistical model-based voice activity detection, [42] introduces a new choice activity detection method based on conditional MAP criterion. The difference between the proposed algorithm and the referenced papers is the robustness of the proposed algorithm against the noise; see Section 5.1.2 and [18]. For automatic target detection (airplanes) in images, several related papers have covered this area, for example, [27] presents a technique of how the airplanes in airports could be detected in satellite images using an approach based on visual saliency computation and symmetry detection. The paper [36] presents airplane detection and tracking technique using wavelet features and SVM classifier. The object tracking is considered in this paper as a classification problem by labeling the object in the first name. Next, SVM was trained using the training vectors obtained from image frames. The paper [7] presents the region-based airplane detection in remotely sensed imagery. It is a two-step method of target detection: first, segmentation is performed on the original image, and then the image regions are used to determine which region belongs to candidates of target area. However, the proposed algorithm could detect objects such as airplanes not only from satellite images but also from real-time videos. In addition to that, it could detect objects even when the image is corrupted by extra noise (low SNR), as shown in Figures 5 and 6. For network traffic monitoring case, we focus on the DWSN dynamics, induced by possible mobility as well as expiration of the ENs. Referring to the discussion and Figure 1 in Section 3, the EN mobility and/or expiration induces user population changes within the clusters of the DWSN; thus, necessitating the deployment of random access (RA) algorithms for transmission to the AFNs [12, 31,40,39].

### 3 The OTM-DWSN general model

The architecture of the OTM-DWSN model considered is shown in Figure 1 and consists of elementary nodes (ENs), elementary-node clusters and a backbone network of cluster heads and a fusion center, where the ENs are the sensors that may be mobile. We will respectively term the cluster



Figure 1: The architecture of the OTM-DWSN model.

heads and the fusion center as aggregation and forwarding nodes (AFNs) and base station (BS). The objective of the DWSN architecture is the pursuing of a set of signal processing operations (for detection, identification, etc.) specified by the OTM objective. Given the pre-determined signal processing objectives, the ENs, AFNs, and BS perform the following functions: (a) the ENs are grouped into distinct clusters, where each cluster contains a single AFN. Each EN collects local data and transmits them to its local AFN, via an appropriate protocol (mainly random access in such topologies) [11, 12, 34, 39]. The ENs may generally be low-cost and low-energy; thus, short-life devices. (b) Each AFN collects the data sent by its local ENs and processes them, using an operation determined by the network signal processing objectives; it also receives processed data sent by other neighboring AFNs. The AFN then processes the compounded processed data, utilizing an operation that is determined by the network signal processing objectives, and transmits the outcome to selected neighboring AFNs or the BS. The AFNs have processing capabilities and are devices with energy and life-spans that are much higher than those of the ENs; their life-spans and energy may still be limited, however. (c) The BS fuses data transmitted to it by neighboring AFNs, utilizing an operation that is determined by the network signal processing objectives. The BS has practically unlimited life-span.

When the cluster populations are time-varying due to ENs mobility and/or expirations, they transmit to their assigned AFN via a random access (RA) protocol [31, 33,40,39]; the life-span of each EN is a function of the channel monitoring and the retransmissions it performs, in compliance with the rules of the deployed RA [40,39]. Operations are performed at all nodes of the backbone network: at the AFNs and the BS. The nature of the operations is determined by the signal processing operations performed in the network; as dictated by the OTM, the environment that generates the data and the data rates [33]. At the same time, the energy consumption of the AFNs is a function of the data rates they receive and produce,



Figure 2: The architecture of each EN and CH.

and the complexity of the operations they perform. Each AFN performs operations on its input data rates to produce the data rates it outputs to neighboring AFNs and/or the BS [39]. The OTM imposes strict delay constraints. Thus, the network is required to complete its signal processing operations within given fixed T time units. The general structure of each EN and CH is shown in Figure 2. During this time period, some ENs generally expire, since their lifespans may be generally only fractions of the time period T. This causes changes in the cluster data rates and thus induces dynamics that may dictate rate allocations across the AFNs and/or network architectural configurations [33, 40,39]. Such dynamics are dictated by the specific changes of cluster rates, in conjunction with the characteristics of the deployed RA within its stability region. It is thus important that the cluster rates be continuously monitored.

The OTM problem incorporates various specific signal processing components, including automated object identification in noisy images and automated recognition of audio signal activity in noise, among others. In this paper, we will present a core sequential algorithm whose specific various implementations can address effectively these components, as well as the cluster rate monitoring problem. The various implementations of the algorithm are dictated by the models representing the acting data processes in each application.

#### 4 The core algorithm

The core algorithm presented in this section is the sequential detection of change algorithm that was first presented in 1954 by Page [30], for detecting a change from a given memoryless and stationary process to another such given process. In 1971, Lorden [28] proved the asymptotic optimality of Page's algorithm. Later, Bansal et al. first proved asymptotic optimality of the algorithm for processes with memory and satisfying mixing conditions [3], and then provided outlier resistant generalizations of their algorithm [4]. Burrell et al. extended the algorithm to detect multiple repeated changes among a set of processes and proved asymptotic optimality, first when the processes are parametrically defined [10], and then when the processes are contaminated by data outliers [32]. The algorithm has also been used in various networks applications [12, 13, 14, 33,39], and has recently been used in object recognition in images and in detection of voice activity [18]. Below, we present a summary of the extended algorithm in [10].

Let the process which initially generates the data be known to be the process  $\mu_0$ , called hypothesis,  $H_0$ . Let it be possible that a shift to any one of m-1 distinct processes  $\mu_i$ ; i = 1, ..., m-1, called hypotheses  $H_i$ , i = 1, ..., m-1, may occur at any point in time, where if a  $\mu_0$  to  $\mu_i$  shift occurs, then the process  $\mu_i$  remains active thereafter. The objective is to detect the occurrence of a  $\mu_0$  to  $\mu_i$  shift, as accurately and as timely as possible, including the detection of the process  $\mu_i$  which  $\mu_0$  changed to. Let us denote by  $f_i$ ; i = 0, 1, ..., m - 1, the density or probability functions induced by the processes  $\mu_i$ , i = 0, 1, ..., m - 1, and let us denote conditional density or probability functions similarly. Then, the following algorithm has been proposed and analyzed in [10], where  $x_1^n = [x_1, ..., x_n]$ .

#### 4.1 Extended algorithm

(a) Select a threshold  $\delta_0 > 0$ .

(b) Have m-1 parallel algorithms operating. The *i*th algorithm,  $i = 1, \ldots, m-1$ , is monitoring a  $\mu_0$  to  $\mu_i$  shift.  $T_n^{0i}(x_1^n)$  denotes the operating value of the *i*th algorithm at time *n*, given the observation sequence  $x_1^n$ . The operating value  $T_n^{0i}(x_1^n)$  is updated as follows:

$$T_0^{0i} \equiv 0,$$
  
$$T_n^{0i}(x_1^n) = \max\left(0, T_{n-1}^{0i}(x_1^{n-1}) + \log\frac{f_i(x_n|x_1^{n-1})}{f_0(x_n|x_1^{n-1})}\right).$$
(1)

(c) The algorithmic system stops the first time n when either one of the m-1 parallel algorithms crosses the common threshold  $\delta_0$ . If the *i*th algorithm is the one that first crosses the threshold, then it is declared that a  $\mu_0$  to  $\mu_i$  shift has occurred.

The asymptotic optimality of the algorithm has been proven in [10], where the expected time for a correct decision is asymptotically the fastest among all algorithms that satisfy a specific false alarm constraint. We note that the algorithm is characterized by low complexity. When the processes monitored are memoryless, the algorithm also requires no memory.

A re-initialization extension model is then assumed as follows. At any point in time, let the data be generated by one of m mutually independent and parametrically defined stochastic processes { $\mu_i$ ; i = 0, 1, ..., m-1}. At any point in time, the acting process may shift to either one of the remaining processes, in an equally probable fashion. The objective is to detect such shifts as accurately and as timely as possible. The algorithm below was then proposed.

#### 4.2 Reinitializing algorithm

With each process  $\mu_i$ , we associate a positive threshold value  $\delta_i$ . Let it be known that at time zero the process  $\mu_0$  is acting. Then, at time zero, the extended algorithm is deployed, with common operating threshold  $\delta_0$ . Let  $T_1$ denote the time instant when the above algorithm stops, and let a  $\mu_0 \rightarrow \mu_i$  shift be decided at  $T_1$ . Then, at  $T_1$ , the  $\mu_0 \rightarrow \mu_i$  decision is accepted and the extended algorithm is deployed again, with a common operating threshold  $\delta_i$ , to monitor a shift from the process  $\mu_i$  to either one of the remaining processes. The common operating threshold  $\delta_i$ 



**Figure 3:** False alarm & power curves for the  $\{\mu_k \rightarrow \mu_j; j \neq k\}$  monitoring system. Threshold values  $\eta_k < \eta'_k$ .

is associated with the starting process  $\mu_i$ . In general, let  $\{T_l\}_{l\geq 0}$  denote the sequence of decision/re-initialization time instants induced by the algorithm, with  $T_0 \triangleq 0$ . Then, at  $T_l$  it is decided that some process  $\mu_j$  starts acting, and the extended algorithm with a common operating threshold  $\delta_j$  is immediately deployed, to monitor a change from  $\mu_j$  to either one of the remaining processes. Within the time interval  $[T_l, T_{l+1})$ , it is decided that the process  $\mu_j$  is continuously acting.

Asymptotic performance and stability of the reinitializing algorithm has been studied in [10]. As the Kullback-Leibler distances among the involved processes increase, so does the speed and the accuracy of the algorithms: they can then detect accurately rapid changes.

## 4.3 Threshold selection

The implementation of the algorithm is clearly determined by its non-asymptotic performance: its performance for finite values of the thresholds  $\{\delta_i\}$ . Given a threshold value  $\delta_k$ , the performance of the  $\mu_k$  to  $\mu_j$  monitoring algorithm is basically characterized by two-time curves: the power and false alarm curves, denoted respectively by  $\beta_{kj}(r)$  and  $\alpha_{kj}(r)$ , where r denotes the discrete time or number of data samples collected, and where the following holds:

- $\beta_{kj}(r)$ : the probability that the  $\mu_k \to \mu_j$  monitoring algorithm crosses threshold  $\delta_k$  before or at sample size r, given that the acting process is  $\mu_j$  throughout, named power curve.
- $\alpha_{kj}(r)$ : the probability that the C monitoring algorithm crosses threshold  $\delta_k$  before or at sample size r, given that the acting process is  $\mu_k$  throughout, named false alarm curve.

In Figure 3, we plot the behavior of the power and false alarm curves, for two different values,  $\eta_k$  and  $\eta'_k$ , of the common threshold  $\delta_k$  used by the  $\{\mu_k \rightarrow \mu_j; j \neq k\}$  monitoring system, where  $\eta_k < \eta'_k$ .

From Figure 3, we note that as the value of the decision threshold increases, the false alarm curve decreases, but so does the power curve. The threshold selection for the  $\{\mu_k \to \mu_j; j \neq k\}$  monitoring system may be based on a required lower bound for the power and a required upper bound for the false alarm, at a given time instant r. When all the m-1 algorithms that monitor change from process  $\mu_k$ are considered, the common threshold  $\delta_k$  may be selected based on the following principle: at a given sample size r, have the powers induced by the parallel algorithms be above a predetermined lower bound, while the false alarm induced by each algorithm remains below a predetermined upper bound, where the existence of such a threshold is determined by the selected values of these upper and lower bounds; as the Kullback-Leibler distances between  $\mu_k$ and each one of the  $\{\mu_i; j \neq k\}$  processes increase, the simultaneous attainability of lower false alarms and higher powers increases as well [3,4,10,26,28]. An alternative principle for the selection of the common threshold  $\delta_k$  is the following: at a given sample size r, maximize the distance between false alarm and power induced by the minimum Kullback-Leibler distance pair  $(\mu_k, \mu_j)$ ;  $j \neq k$ , where the latter pair induces the closest to each other power and false alarm curves within the  $\{\mu_k \to \mu_j; j \neq k\}$  monitoring system.

Below, we express the specific forms that the algorithm in (1) takes for three special cases of data generating stochastic processes. As we will discuss in Sections 4.4– 4.6, these special cases correspond to three important operations within the OTM-DWSN system.

#### 4.4 The Bernoulli model

We assume that binary sequences  $x_1^n$  are generated by independent Bernoulli trials, where the Bernoulli parameter for  $P(x_i) = 1$  may shift among two different values, p and q, where p > q and where the parameter p represents the process  $\mu_1$ , while the parameter q represents the process  $\mu_0$ . The two processes  $\mu_1$  and  $\mu_0$  are then memoryless with  $f_1(x) = p^x(1-p)^{1-x}$  and  $f_0(x) = q^x(1-q)^{1-x}$ . Denoting by  $T'(x_1^n)$  the value of the  $\mu_0 \to \mu_1$  monitoring algorithm in (1) and by  $T(x_1^n)$  the value of the  $\mu_1 \to \mu_0$  monitoring algorithm in (1), both at time n, in this case, we obtain in a straight forward fashion and after appropriate scaling the following sequential expressions:

$$q \to p$$
 monitoring, for  $p > q$ :  
 $T'(x^n) = \max\left\{0, T'(x^{n-1}) + [x_n + \gamma(q, p)]\right\},$  (2)

 $p \rightarrow q$  monitoring, for p > q:

$$T'(x^{n}) = \max\{0, T'(x^{n-1}) - [x_{n} + \gamma(q, p)]\}, \qquad (3)$$

where

$$\gamma(q,p) = \frac{\log \frac{(1-p)}{(1-q)}}{\log \frac{p(1-q)}{q(1-p)}}.$$
(4)

It can be shown [26] that  $q < \gamma(q, p) < p$ . Power and false alarm curves can be recursively computed via the methodology shown in [26], and subsequently used for the selection of the two algorithmic thresholds used by the algorithms in (2) and (3), as discussed in Section 4.3.

### 4.5 The Poisson model

We assume that data arrivals are generated by a Poisson process whose rate,  $\lambda$ , in expected number of arrivals per time unit, may shift among a given finite set of values. Thus, shifts among a set  $\{\mu_i; i = 0, 1, \dots, m-1\}$ of processes is here represented by shifts among a set  $\{\lambda_i; i = 0, 1, \dots, m-1\}$  of Poisson rates. Since the Poisson processes are memoryless, the monitoring algorithm in (1) utilizes no memory in this case. We consider fixed-length time intervals named *frames*, within which the number of arrivals are counted, and we measure time in frame units. We then denote by  $n_r$  the number of arrivals in the rth frame, from the beginning of time, and we denote by  $V_{ki}(r)$  the value of the  $\lambda_k \rightarrow \lambda_i$  monitoring algorithm at the *r*th frame from its beginning. After some modifications including scaling, we then find the following sequential evolution of the algorithm in (1) in this case, where d denotes the length of a single frame in time units:

Poisson  $\{\lambda_k \to \lambda_j\}$  monitoring algorithm

$$V_{kj}(r+1) = \max \left\{ 0, V_{kj}(r) + (-1)^{\operatorname{ime}(k,j)} [s_{kj}n_{r+1} - dt_{kj}] \right\},$$
(5)

where

$$\operatorname{ime}(k,j) = \begin{cases} 0, & \text{if } \lambda_j > \lambda_k, \\ 1, & \text{if } \lambda_j < \lambda_k, \end{cases}$$
$$\zeta(\lambda_k \lambda_j) \triangleq \left[\lambda_k - \lambda_j\right] \left[\log\left(\frac{\lambda_k}{\lambda_j}\right)\right]^{-1}, \qquad (6)$$
$$\zeta(\lambda_k, \lambda_j) = \frac{t_{kj}}{s_{kj}}.$$

It can be shown [12] that  $\min(\lambda_k, \lambda_j) < \zeta(\lambda_k, \lambda_j) < \max(\lambda_k, \lambda_j)$ . The  $\zeta(\lambda_k, \lambda_j)$  expression may be approximated by a rational number, as exhibited in (6), where the integers  $t_{kj}$  and  $s_{kj}$  are such that  $t_{kj} < s_{kj}$ . Power and false alarm curves can be computed recursively, as shown in [12], to be used for algorithmic threshold selections, as discussed in Section 4.3. We point out that when only two Poisson processes alternate in generating the observed data, the two algorithmic thresholds involved then in the data monitoring were "learned", instead, via an optimal algorithm in [14].

In this model, it is assumed that the data generating process may shift between a white zero mean stationary Gaussisn process  $\mu_0$  and a stationary memoryless process  $\mu_1$  with perdatum density function that is the convolution of the perdatum density function of the Gaussian process  $\mu_0$  with a Laplacian density function. As induced by the algorithm in (1), the  $\mu_0 \rightarrow \mu_1$  and  $\mu_1 \rightarrow \mu_0$  monitoring algorithms do not require any memory then. Derivation of the per datum density function of process  $\mu_1$ , derivations, and scaling [18] finally leads to the following sequential monitoring algorithms:

Given

- $\sigma$ : the standard deviation of the white stationary Gaussian process  $\mu_0$ ;
- $\alpha$ : the parameter of the Laplacian.

Define

$$\beta \triangleq \alpha \sigma, \quad \xi \triangleq \frac{x}{\sigma},$$

$$h(\xi) \triangleq \exp(-\beta\xi) \Phi(\xi - \beta),$$

$$\Phi(\xi x) = \int_{-\infty}^{x} \frac{1}{\sigma} \varphi(u) du,$$

$$\varphi(u) \triangleq \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^{2}}{2}\right),$$
(7)

where the signal-to-noise ratio (SNR) here is

$$SNR = \frac{2}{\sigma^2 \alpha^2} = \frac{2}{\beta^2}.$$
(8)

Then, the  $\mu_0 \rightarrow \mu_1$  monitoring algorithm evolves sequentially as

$$T(x^{n}) = \max\left[0, T(x^{n-1}) + 1 + \left[\ln\frac{\beta\sqrt{2\pi}}{2} + \frac{\beta^{2}}{2}\right]^{-1} \times \left[\frac{\xi^{2}}{2} + \ln\left\{h(\xi) + h(-\xi)\right\}\right]\right].$$
(9)

The  $\mu_1 \rightarrow \mu_0$  monitoring algorithm evolves sequentially as

$$T(x^{n}) = \max\left[0, T(x^{n-1}) - 1 - \left[\ln\frac{\beta\sqrt{2\pi}}{2} + \frac{\beta^{2}}{2}\right]^{-1} \times \left[\frac{\xi^{2}}{2} + \ln\left\{h(\xi) + h(-\xi)\right\}\right]\right].$$
(10)

Recursive integral equations can be derived that allow the numerical computation of the power and false alarm curves to be subsequently used in the selection of the thresholds used by the algorithms in (9) and (10); see [18].

## 5 Experimental results and analysis of the core algorithm

In this section, we present two implementations of the core algorithm in Section 4, within the OTM-DWSN system: one used for the visual and audio tracking of an object, and another for the monitoring of cluster rates in the network.

## 5.1 Visual and audio tracking of a moving object

The tracking of a moving object may be attained via the processing and monitoring of both sequentially obtained images and progressively emanating sound from the object, as detected by the various network sensors. Some of these sensors may be on the ground, some may be robotic swarms, while others may be satellites. In this section, we present an algorithm that may be used in visual tracking and an algorithm that may be used for audio tracking.

## 5.1.1 Visual tracking

To track a moving object from sequentially obtained images, the object must be first clearly identified and outlined in each image. In this section, we propose a manifestation of the core algorithm in Section 4, for the satisfaction of this objective. Specifically, we propose the adoption of a Bernoulli model and the subsequent deployment of the Bernoulli model algorithm in Section 4.4, for the satisfaction of this objective. The proposed approach involves the following steps:

- (a) the obtained images are converted to black and white;
- (b) from training data, the range of white versus black is decided and each pixel is subsequently represented by a binary number: 1 for black and 0 for white;
- (c) from training data, the percentage range of identity-1 or equivalently identity-0 pixels in the background versus those within the object is quantified. Then, the margins of these ranges that are closest to each other, q and p, are determined;
- (d) the Bernoulli model algorithms in Section 4.4, designed at the p and q values discussed in (c), are subsequently deployed in real time, to clearly outline the position of the object in each of the sequentially obtained images. The algorithms are implemented on sequentially scanned pixels, in the reinitializing mode: as soon as a p → q shift is detected by the p → q monitoring algorithm, the q → p monitoring algorithm is deployed to detect such shift, and so on.

The Bernoulli model algorithms were deployed to identify and clearly outline airplanes, from images obtained by satellites and subsequently corrupted by Additive White Gaussian Noise (AWGN). Figure 4 exhibits the originally obtained images, while Figure 5 shows an example of the effect due to the AWGN corruption, and Figure 6 exhibits the noisy images in black and white.



Figure 4: Original images.



Figure 5: Image on left as that in Figure 4. Image on right corrupted by AWGN.

From training data, the ranges of identity-1 pixel percentages within the plane figures versus the background were found to be (0.045, 0.08) and (0.2, 0.4), respectively. The *p* and *q* values for the design of the two algorithms in Section 4.4 were thus selected equal to 0.08 and 0.2, respectively, where  $0.08 \rightarrow 0.2$  signifies a change from airplane figure to background and where  $0.2 \rightarrow 0.08$  represents a change from background to airplane figure. We selected the algorithmic thresholds following the methodology explained in Section 4.3, requiring false alarm and power bounds, respectively equal to 0.05 and 0.95, for sample size 100. For the  $0.08 \rightarrow 0.2$  algorithm, the selected threshold was 0.2, while for the  $0.2 \rightarrow 0.08$  algorithm, the selected threshold was 12.5. In Figure 7, we exhibit

the images obtained by the Bernoulli model algorithms. The accuracy of the algorithm is exhibited by comparing Figure 7 with Figure 6.

#### 5.1.2 Audio tracking

For audio tracking of an object, the audio signature of the object must be monitored and detected accurately when present, while embedded in noise. To attain this objective, the noise and the audio signature of the object must be first modeled. Based on the latter model, algorithms that detect accurately changes from absence to presence of the object audio and vice versa must be designed and deployed. In this section, we model the noise as additive, white, stationary, and Gaussian, and we model the object



Figure 6: Noisy images in black and white.



Figure 7: The images after being processed by the bernoulli model algorithms.

audio by mutually independent and identically distributed activity intervals with amplitude distribution described by a Laplacian density function (a model frequently adopted for speech [24,43]. Consequently, we deploy the algorithms in the Laplacian-Gaussian hybrid model of Section 4.6 to detect changes from silence to audio object activity and vice versa.

In consistency with the works in [24,43], we selected the standard deviation of the noise equal to 0.0394 and the Laplacian parameter equal to 0.99. Using the methodology in Section 4.3, we required false alarm and power bounds respectively equal to 0.05 and 0.95, for sample size 100. We subsequently selected algorithmic threshold values equal to 0.3 and 0.05, for the detection of change from audio signature presence to absence versus audio signature absence to presence, respectively. We tested the algorithms for various values of the signal-to-noise ratio (SNR), as defined by (8) in Section 4.6. Figure 8 exhibits noisy audio with different SNRs, while Figures 9 and 10 show the performance of the audio recognition algorithms regarding accuracy. We notice that the accuracy improves as the SNR increases, while always remaining significantly good. In [18], comparisons with the methods in [24,43] were made, showing consistent outperforming of the algorithm proposed in this paper. The comparison results are not included here due to lack of space.



Figure 8: Noisy audio with SNR equal to 25 dB and 5 dB, respectively.



**Figure 9:** Algorithmic accuracy for SNR equal to 25 dB.

#### 5.2 DWSN rate monitoring

In this section, we focus on the DWSN dynamics, induced by possible mobility as well as expiration of the ENs. Referring to the discussion and Figure 1 in Section 3, the EN mobility and/or expiration induces user population changes within the clusters of the DWSN; thus, necessitating the deployment of random access (RA) algorithms for transmission to the AFNs [12,31,40,39]. Assuming starting symmetric DWSN topology, where all clusters have identical EN populations randomly distributed, the per-cluster cumulative data generating process may be modeled as homogenous Poisson, where the Poisson rates across different clusters are identical. In [40,39], a specific RA per cluster is deployed, whose throughput-delay characteristics determine the range  $(\lambda_0, \lambda_1)$  of Poisson rates within which the RA attains its best performance. Consequently, an architectural reconfiguration algorithm, facilitated by a rate monitoring algorithm (RMA), has been deployed, [40,39], that reconfigures the

DWSN topology, when so dictated by the RMA, to maintain the system symmetry and the Poisson per-cluster rates within the  $(\lambda_0, \lambda_1)$  range. In [40, 39], the RMA is deployed idependently by each AFN in the topology (see Figure 1) and decides sequences of consecutive  $\lambda_0 \rightarrow \lambda_1$  and  $\lambda_1 \rightarrow \lambda_0$ shifts, where the decisions of the different RMAs are communicated to the BS via the backbone network that connects the BS with the AFNs. In this section, we will present a Distributed Rate Monitoring Algorithm (DRMA), instead, that detects synchronous sequences of consecutive  $\lambda_0 \rightarrow \lambda_1$ and  $\lambda_1 \rightarrow \lambda_0$  shifts, across all clusters in the DWSN.

#### 5.2.1 The DRMA operations and asymptotic performance

The idea here is that, since the system symmetry is maintained by the architectural reconfiguration algorithm in [40, 39], the identical per-cluster rates may by monitored via a distributed algorithm, where decisions of per-cluster RMAs are fused either at the BS or at one of the AFNs, named



Figure 10: Algorithmic accuracy for SNR equal to 5 dB.

Decision AFN (DAFN), and where the BS versus the DAFN respectively makes the final decisions about shifts between the rates  $\lambda_0$  and  $\lambda_1$ . If the BS makes the final decision, it does not contribute local data observations to it. If, on the other hand, the final decision is made by some DAFN, its local data observations contribute to the latter decision. The identical RMAs deployed will be as those in Section 4.5, for m = 2. The DRMA, on the other hand, is presented below, where, for generality purposes, we consider the case where the final decision is made by a DAFN; the final decision being made by the BS is a special case of the former, then. We assume that all EN transmissions are digital and packetized.

As in Section 4.5, we consider consecutive time *frames* whose length in time units is *d*. We then denote by  $g_j(n_1,...,n_q)$ , j = 0,1, the joint distribution of local (from its local ENs) packet arrivals at the DAFN in *q* consecutive frames, as generated by the process  $\mu_j$ , where  $n_i$  is the number of arrivals in the *i*th frame and where  $\mu_j$ , j = 0, 1, is Poisson with rate  $\lambda_j$ . Let the number of the remaining AFNs be *N*, indexed from 1 to *N*, and let  $u_{i,1-j}$ , j = 0, 1, i = 1, ..., N, be defined as follows:

$$u_{i,1-j} = \begin{cases} 1, & \text{if the } i\text{th AFN decides that} \\ & \text{the process } \mu_{1-j} \text{ is active,} \\ 0, & \text{if the } i\text{th AFN decides that} \\ & \text{the process } \mu_j \text{ is active.} \end{cases}$$
(11)

We then propose a Distributed RMA (DRMA) whose only difference from the centralized RMA lies in the updating log likelihood step in (1), Section 4. Indeed, we propose that the log ratio in (1) be substituted by the updating step below:

$$\sum_{i=1}^{N} w_i u_{i,j} + \log \frac{g_j(n_{r+1} \mid n_1, \dots, n_r)}{g_{1-j}(n_{r+1} \mid n_1, \dots, n_r)},$$
(12)

where  $\{w_i\}$  are generally constants whose objective is to weigh the contribution of the various RMAs according to their respective performance characteristics. Let  $\alpha_i$  and  $1 - \beta_i$  denote respectively the false rate, in percentage of false decisions, and the power rate, in percentage of correct decisions, as induced by the RMA employed by the *i*th neighboring node. Then, drawing from some parallelisms with the models considered in [14], we conclude the following form of the set  $\{w_i\}$ :

$$w_i = \log \frac{(1 - \beta_i)(1 - \alpha_i)}{\beta_i \alpha_i}; \quad i = 1, \dots, N.$$
 (13)

When the processes that generate the arrivals are Poisson, the log likelihood ratio in (12), after transformation and scaling, is as the updating step in expression (5), Section 4.5.

When the final monitoring decisions are being made by the BS, no local data are collected and the DRMA. If then the RMAs are also identical, as assumed in this section, the updating step in (12) takes the following form after normalization:

$$wN^{-1}\sum_{i=1}^{N} u_{i,j},$$
 (14)

where  $\alpha$  and  $1 - \beta$  represent the false alarm rate and the power rate per neighboring node, respectively, and where then  $w = \log[[(1 - \alpha)(1 - \beta)]/\alpha\beta]$ .

We note that the weights  $\{w_i\}$  in (13) are positive if and only if the false alarm rate is less than the power rate. We also note from (12) that the decisions of a neighboring RMA contribute then positively to the acceleration of a  $g_{1-j} \rightarrow g_j$ shift decision (by increasing the size of the updating step), when these decisions point to the  $\mu_j$  process as being active. The asymptotic performance characteristics of the DRMA are included in the appendix, where the asymptotic superiority of the DRMA, in comparison to the RMA, is proven, and where the special case of the BS making the final decision is studied as well.

#### 5.2.2 Numerical evaluations

To test the DRMA numerically, we adopted a specific traffic and transmission model. We specifically assumed that the Poisson traffic generated by the ENs consists of messages whose length is random and exponentially distributed, with fixed average length throughout the DWSN. We then assumed that the messages are queued at either the DAFN or the BS (depending on which one is the acting implementation case), and are subsequently transmitted through a transmission channel in a Time Division Multiple Access (TDMA) fashion, one packet per frame for each transmitted message, where the frame lengths are equal to d time units, as in expression (5), Section 4.5, and where the percentage of each frame capacity dedicated to transmissions equals the rate (one of the  $\lambda_0$  or  $\lambda_1$  rates) decided by the DRMA. Since the DRMA induces some false decisions, the allocated to transmissions capacity per frame may cause either wasted capacity or traffic rejections: assuming  $\lambda_0 < \lambda_1$ , if  $\lambda_0$  is true and  $\lambda_1$  is decided, excessive frame capacity allocation will cause capacity waste, while if  $\lambda_1$  is true and  $\lambda_0$  is decided, insufficient frame capacity allocation will cause traffic rejections. The performance metrics for the DRMA are rejection rates, wasted capacity rates, and expected delays of the successfully transmitted messages.

We finally modeled the actual time periods, in time units, during which each of the two Poisson message rates,  $\lambda_0$ and  $\lambda_1$ , are acting as geometrically distributed. We specifically assumed that the distribution of the time period during which a given rate  $\lambda_j$  is continuously acting is geometric, having the form

$$Q_i(k) = (1 - \rho_i)^{-1} \rho_i^{k-1}, \quad k \ge 1,$$

where k represents the number of time slots. The expected time  $E\{l_i\}$  during which  $\lambda_j$  is continuously acting is thus,

$$E\{l_i\} = (1 - \rho_i)^{-1}$$

and the average fraction of time  $\gamma_i$  for the  $\lambda_i$  activity is

$$\gamma_i = \left[\sum_{k=0}^{1} E\{l_k\}\right]^{-1} E\{l_i\}$$

$$= \left[\sum_{k=0}^{1} (1-\rho_k)^{-1}\right]^{-1} (1-\rho_i)^{-1}, \quad i = 0, 1$$

Assuming that the rates are ordered as  $\lambda_0 < \lambda_1$ , the  $\gamma_i$ 's are selected ordered as  $\gamma_1 < \gamma_0$ . Then, for some constant  $C > \max\{\frac{1}{\gamma_0}, \frac{1}{\gamma_1}\}$ , the  $\rho_i$  values are determined as

$$\rho_i = 1 - (C\gamma_i)^{-1}, \quad i = 0, 1.$$

For ease in graphic representation, a geometric structure is adopted in the selection of the  $\gamma_i$ 's. Specifically, for some constant  $\alpha$ ,  $0 \le \alpha \le 1$ , the  $\gamma_i$ 's are generated as follows:

$$\gamma_0 = \gamma_0(\alpha) = (1 - \alpha^2)^{-1} (1 - \alpha), \gamma_1 = \gamma_1(\alpha) = \alpha \gamma_0(\alpha) = (1 - \alpha^2)^{-1} (1 - \alpha) \alpha.$$
(15)

Thus, for any  $\alpha$  value, the generated  $\gamma_i$ 's are such that  $\gamma_1 < \gamma_0$  and  $\gamma_0 + \gamma_1 = 1$ . The following conclusions can be drawn from (15):

- (a)  $\gamma_0(\alpha)$  is a decreasing function of  $\alpha$ ;
- (b)  $\gamma_1(\alpha)$  is an increasing function of  $\alpha$ ;
- (c) in general, as  $\alpha$  decreases, the higher rate becomes increasingly bursty while the frequency with which the lower rate occurs increases monotonically; as  $\alpha \to 0$ ,  $\gamma_0$  approaches 1. As  $\alpha$  increases, the frequencies of occurrence for the two different rates tend to equalization; as  $\alpha \to 1$ , the  $\gamma_i$ , i = 0, 1, values approach 1/2.

In our simulations, we selected a frame length equal to 40 and various numbers of neighboring AFNs contributing to the adaptation steps of the DRMA. We also selected various pair values  $(\lambda_0, \lambda_1)$  of rates for the two Poisson processes which generate the packet arrivals, as well as exponentially distributed message lengths with various average lengths. For the numerical results included in this paper, the selected average message length equals 15 packets (to ensure non insignificant message arrival rates), while the chosen pairs of message arrival rates are (0.2, 0.6) and (0.2, 0.8). We first considered 4, 8, and 12 AFNs feeding a DAFN which also utilizes local data, via expression (12). We also simulated the case where the DRMA is implemented by the BS, via expression (14), utilizing inputs from 24 neighboring AFNs. We selected the threshold values via the methodology in Section 4.3.

Our results are exhibited in Figures 11 to 16, where the performance of the DRMA is compared with that of the RMA: Figures 11, 12, and 13 correspond to the pair (0.2, 0.6) of rates, while Figures 14, 15, and 16 correspond to the pair (0.2, 0.8) of rates. From the figures we observe that, as compared to the RMA and as predicted by the theorems in the appendix, the DRMA generally improves delays at the expense of increased traffic rejection and wasted capacity rates. As the Kullback-Leibler distance between the two



0.8

0.9

0.3 0.4 0.5 0.6 0.7 Alpha

Figure 11: Average message length 15. Rate pair (0.2, 0.6).



Figure 12: Average message length 15. Rate pair (0.2, 0.6)



Figure 13: Average message length 15. Rate pair (0.2, 0.6).

Poisson processes increases (the case of the (0.2, 0.8) pair), however, the DRMA implemented by a DAFN, that incorporates local data, improves delays dramatically at almost no cost in traffic rejection and wasted capacity rates. When the DRMA is implemented by the BS and does not utilize local data (case of 24 neighboring nodes), the highest delay gain is obtained, but the highest penalties are paid as well, regarding traffic rejection and wasted capacity rates.

### **6** Conclusions

Considering the environment of an object tracking and monitoring distributed wireless sensor network, we focussed on a core algorithm and presented several of its manifestations



Figure 14: Average message length 15. Rate pair (0.2, 0.8).



Figure 15: Average message length 15. Rate pair (0.2, 0.8).



Figure 16: Average message length 15. Rate pair (0.2, 0.8).

and applications, including object identification from images, audio object tracking, and distributed network traffic monitoring. For the latter application, we proposed and analyzed a distributed variation of the algorithm. In all cases, we included numerical examples and evaluations. The core algorithm is highly effective and robust, while its applications are also numerous.

#### Appendix A Performance characteristics of the DRMA

Let  $N_{1-j,j}$ , j = 0, 1, denote the extended stopping variable induced by the DRMA algorithm with updating step number as in (12), Section 4. Let also  $I_{1-j,j}$  denote the Kullback-Leibler number of the process  $\mu_j$  with respect to the process  $\mu_{1-j}$ . Let  $N_{1-j,j}^{(i)}$ , j = 0, 1, denote the extended stopping variable induced by the RMA deployed by the *i*th

Rates

0.02

0

0.1 0.2

neighboring AFN and let us define

$$\rho_{i,k|j} \equiv \frac{E\{N_{k,1-k}^{(i)}|\mu_j\}}{E\{N_{k,1-k}^{(i)}|\mu_j\} + E\{N_{1-k,k}^{(i)}|\mu_j\}}, \quad k = 0, 1, \ j = 0, 1.$$
(A1)

Then, the following theorem can be expressed.

**Theorem A1.** Let the processes  $\{\mu_j, j = 0, 1\}$  be stationary, ergodic, mutually independent, and satisfying the general mixing conditions (A) and (B) in [10]. Then, the asymptotic performance of the DRMA is as follows:

$$E\{N_{1-j,j} | \mu_j\} \approx \frac{\log \eta_{1-j}}{\sum_{i=1}^{N} w_i \rho_{i,j|j} + I_{1-j,j}}, \quad as \ \eta_{1-j} \to \infty,$$

$$E\{N_{1-j,j} | \mu_{1-j}\} : \begin{cases} \approx \left[\sum_{i=1}^{N} w_i \rho_{i,j|1-j} - I_{j,1-j}\right]^{-1} \log \eta_{1-j}, \\ if \ \sum_{i=1}^{N} w_i \rho_{i,j|1-j} > I_{j,1-j} \\ 2^{-1} \eta_{1-j}, \\ if \ \sum_{i=1}^{N} w_i \rho_{i,j|1-j} < I_{j,1-j}, \\ if \ \sum_{i=1}^{N} w_i \rho_{i,j|1-j} < I_{j,1-j}, \end{cases}$$
(A2)

At the same time, the performance of the RMA is shown below:

$$E\{N_{1-j,j}|\mu_{j}\} \approx \frac{\log \eta_{1-j}}{I_{1-j,j}}, \quad as \ \eta_{1-j} \to \infty,$$
  
$$E\{N_{1-j,j}|\mu_{1-j}\} \ge 2^{-1}\eta_{1-j}, \quad as \ \eta_{1-j} \to \infty.$$
 (A3)

*Proof.* From the theorems in [4], we conclude that the asymptotic conditional expected values of the stopping times are determined by the conditional expected values of the updating step in (3). Considering the fact that  $\rho_{i,j|j} = E\{u_{i,j}|\mu_j\}$  and  $\rho_{i,j|1-j} = E\{u_{i,j}|\mu_{1-j}\}$ , the latter expected values are equal to

$$\begin{split} &\sum_{i=1}^N w_i \rho_{i,j|j} + I_{1-j,j}, \quad \text{conditional on } \mu_j, \\ &\sum_{i=1}^N w_i \rho_{i,j|1-j} - I_{j,1-j}, \quad \text{conditional on } \mu_{1-j}. \end{split}$$

Then, the results in Theorem A1 follow.

The results in Theorem A1 exhibit clearly the interactive relationship between the DRMA and the RMAs deployed by the neighboring AFNs.

Comparing expressions (A2) with expressions (A3), we observe that the DRMA is clearly superior to the RMA, when the constants  $\{w_i\}$  that reflect the contributions of neighboring nodes and the parameters  $\{\rho_{i,j|1-j}\}$  that represent the performance of the neighboring RMAs are such that their sum  $\sum_{i=1}^{N} w_i \rho_{i,j|1-j}$  is smaller than the minimum between the two Kullback-Leibler numbers  $I_{01}$  and  $I_{10}$ . Then, as compared to the RMA, the DRMA decreases the asymptotic expected stopping times for correct decisions by factors of  $[\sum_{i=1}^{N} w_i \rho_{i,0|0} + I_{01}]^{-1} \sum_{i=1}^{N} w_i \rho_{i,0|0}$  and  $[\sum_{i=1}^{N} w_i \rho_{i,1|1} + I_{01}]^{-1} \sum_{i=1}^{N} w_i \rho_{i,1|1}$ , respectively, while it maintains the exponentially longer asymptotic expected times for erroneous decisions.

We will conclude with an asymptotic result for the case where the DRMA-computing node is the BS. Then, no local data are collected and the DRMA updating step in (12) takes the following form after normalization:

$$wN^{-1}\sum_{i=1}^{N}u_{i,j};$$
 (A4)

where  $\alpha$  and  $1 - \beta$  represent the false alarm rate and the power rate per neighboring node, respectively,  $w = \log[[(1 - \alpha)(1 - \beta)]|\alpha\beta]$ .

We now present another theorem.

**Theorem A2.** Let the processes  $\{\mu_j, j = 0, 1\}$  be as in Theorem A1. Let us then consider the DRMA with updating step as in (A4). Let the N neighboring nodes be independent and identical, resulting in identically distributed and independent variables  $\{u_{i,j}\}$ . Let then  $\rho_{j|j} \triangleq \rho_{i,j|j}$ , for all *i*, and  $\rho_{j|1-j} \triangleq \rho_{i,j|1-j}$ , for all *i*, where  $\rho_{i,j|j}$  and  $\rho_{i,j|1-j}$  are as in Theorem A1. Then, one has

$$E\{N_{1-j|j}|\mu_j\} \approx \frac{\log \eta_{1-j}}{w \rho_{j|j}}, \quad as \ \eta_{1-j} \to \infty,$$
(A5)

$$E\{N_{1-j|j}|\mu_{1-j}\} \approx \frac{\log \eta_{1-j}}{w \rho_{j|1-j}}, \quad as \ \eta_{1-j} \to \infty,$$
(A6)

$$P\left(N_{1-j,j} = \operatorname{ceil}\left(\frac{\eta_{1-j}}{w\rho_{j|j}}\right) \mid \mu_j\right) \xrightarrow[N \to \infty]{} 1, \quad \text{for finite } \eta_{1-j}, \quad (A7)$$

$$P\left(N_{1-j,j} = \operatorname{ceil}\left(\frac{\eta_{1-j}}{w\rho_{j|1-j}}\right) \mid \mu_{1-j}\right) \xrightarrow[N \to \infty]{} 1, \quad \text{for finite } \eta_{1-j}.$$
(A8)

*Proof.* The results in (A5) and (A6) are directly from the results in (A2) of Theorem A1, when the neighboring nodes are identical and no local data are collected.

For  $N \gg 1$ , the variable in (A4) is Gaussian, with mean  $w\rho_{j|j}$ and standard deviation  $wN^{-\frac{1}{2}} \left[ \left[ \rho_{j|j} (1 - \rho_{j|j}) \right] \right]^{\frac{1}{2}}$ , given  $\mu_j$ , and mean  $w\rho_{j|1-j}$  and standard deviation  $wN^{-\frac{1}{2}} \left[ \rho_{j|1-j} (1 - \rho_{j|1-j}) \right]^{\frac{1}{2}}$ , given  $\mu_{1-j}$ . As  $N \to \infty$  the standard deviations converge to zero resulting in respective step sizes  $w\rho_{j|j}$  and  $W\rho_{j|1-j}$ , with probability 1. This gives directly the results in (A7) and (A8).

**Remarks.** From the results in Theorem A2, we observe that when the DRMA-computing node is the BS, then the DRMA performance is controlled by the parameters  $\rho_{j|j}$  and  $\rho_{j|1-j}$ . The latter parameters represent respectively the probabilities of correct versus incorrect decisions induced by the RMA of each AFN. For either N finite and  $\eta_{1-j} \rightarrow \infty$ , or  $\eta_{1-j}$  finite and  $N \rightarrow \infty$ , the ratio  $\rho_{j|j}/\rho_{j|1-j}$  equals the ratio  $E\{N_{1-j,j}|\mu_{1-j,j}\}/E\{N_{1-j,j}|\mu_j\}$  of expected stopping times induced by the DRMA.

#### References

- I. F. Akyildiz, W. Su, Y. Sankkarasubramaniam, and E. Cayirci, Wireless sensor networks: a survey, Computer Networks, 38 (2002), 393–422.
- [2] S. Bandyopadhyay and E. J. Coyle, An energy efficient hierarchical clustering algorithm for wireless sensor networks, in Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03), vol. 3, 2003, 1713–1723.
- [3] R. K. Bansal and P. Papantoni-Kazakos, An algorithm for detecting a change in stochastic process, IEEE Trans. Information Theory, 32 (1986), 227–235.
- [4] R. K. Bansal and P. Papantoni-Kazakos, *Outlier-resistant algorithms for detecting a change in a stochastic process*, IEEE Trans. Information Theory, 32 (1989), 521–535.
- [5] M. Bhardwaj and A. P. Chandrakasan, *Bounding the lifetime of sensor networks via optimal role assignments*, in Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, 2002, 1587–1596.

- [6] D. M. Blough and P. Santi, Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks, in Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MobiCom '02), Atlanta, GA, 2002, 183–192.
- [7] S. Bo and Y. Jing, *Region-based airplane detection in remotely* sensed imagery, in Proceedings of the 3rd International Congress on Image and Signal Processing (CISP), vol. 4, 2010, 1923–1926.
- [8] P. W. Boettcher and G. A. Shaw, A distributed time-difference of arrival algorithm for acoustic bearing estimation, in Proceedings of the 4th International Conference on Information Fusion, vol. 1, Montreal, 2001.
- [9] R. R. Brooks and C. Griffin, *Traffic model evaluation of ad hoc target tracking algorithms*, International Journal of High Performance Computing Applications, 16 (2002), 221–234.
- [10] A. Burrell and P. Papantoni-Kazakos, Extended sequential algorithms for detecting changes in acting stochastic processes, IEEE Trans. Systems, Man, and Cybernetics, 28 (1998), 703– 710.
- [11] A. T. Burrell and T. Papantoni, A class of limited sensing random access algorithms with resistance to feedback errors and effective delay control, Journal of Communications and Networks, 8 (2006), 21–27.
- [12] A. T. Burrell and P. Papantoni-Kazakos, Wireless networks in hybrid topologies: signaling, transmission, and dynamic capacity allocation viewed in an integrated fashion, in ITC Mini-Seminar Performance Modeling and Design of Wireless/PCS Networks, Cambridge, MA, 1996.
- [13] A. T. Burrell and P. Papantoni-Kazakos, *Distributed traffic monitoring and learning for ATM networks*, in Proceedings of the 32nd Annual Conference on Information Sciences and Systems, Princeton, NJ, 1998. invited paper.
- [14] A. T. Burrell and P. Papantoni-Kazakos, On-line learning and dynamic capacity allocation in the traffic management of integrated services networks, European Trans. Telecommunications, 10 (1999), 202–214. Special Issue on Architectures, Protocols, and Quality of Service for the Internet of the Future.
- [15] J.-H. Chang and L. Tassiulas, *Routing for maximum system lifetime in wireless ad hoc networks*, in Proceedings of the 37th Annual Allerton Conference on Communications, Control and Computing, Monticello, IL, 1999, 22–31.
- [16] J.-H. Chang and L. Tassiulas, *Energy conserving routing in wireless ad-hoc networks*, in Proceedings of the 37th Annual Allerton Conference on Communications, Control and Computing, Tel Aviv, Israel, 2000, 22–31.
- [17] J. C. Chen, R. E. Hudson, and K. Yao, A maximum-likelihood parametric approach to source localizations, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01), vol. 5, Salt Lake City, UT, 2001, 3013–3016.
- [18] E. A. Etellisi, Modulation, coding and detection for satellite and space communications, master's thesis, University of Colorado Denver, 2011.
- [19] D. S. Friedlander and S. Phoha, Semantic information fusion of coordinated signal processing in mobile sensor networks, International Journal of High Performance Computing Applications, 16 (2002), 235–241.
- [20] Z. Guo, M. Zhou, and G. Jiang, Adaptive sensor placement and boundary estimation for monitoring mass objects, IEEE Trans. Systems, Man, and Cybernetics, 38 (2008), 222–232.
- [21] Y. T. Hou, Y. Shi, J. H. Reed, and K. Sohraby, *Flow routing for variable bit rate source nodes in energy-constrained wireless sensor networks*, in Proceedings of the IEEE International Conference on Communications (ICC '05), Seoul, Korea, 2005, 3057–3062.

- [22] Y. T. Hou, Y. Shi, and H. D. Sherali, On node lifetime problem for energy-constrained wireless sensor networks, Mobile Networks and Applications, 10 (2005), 865–878.
- [23] Y. T. Hou, Y. Shi, and H. D. Sherali, *Rate allocation and network lifetime problems for wireless sensor networks*, IEEE/ACM Trans. Networking, 16 (2008), 321–334.
- [24] International Telecommunications Union, A silence compression scheme for G.729 optimized for terminals conforming to Recommendation V.70. ITU-T Recommendation G.729 Annex B, 1996.
- [25] L. M. Kaplan, Q. Le, and N. Molnar, *Maximum likelihood methods for bearings-only target localization*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01), vol. 5, Salt Lake City, UT, 2001, 3001–3004.
- [26] D. Kazakos and P. Papantoni-Kazakos, *Detection and Estimation*, Computer Science Press, New York, NY, 1990.
- [27] W. Li, S. Xiang, H. Wang, and C. Pan, *Robust airplane detection in satellite images*, in Proceedings of the 18th IEEE International Conference on Image Processing (ICIP), 2011, 2821–2824.
- [28] G. Lorden, Procedures for reacting to a change in distribution, Ann. Math. Stat., 42 (1971), 1897–1908.
- [29] E. N. Onggosanusi, A. M. Sayeed, and B. D. Van Veen, Canonical space-time processing for wireless communications, IEEE Trans. Communications, 48 (2000), 1669–1680.
- [30] E. S. Page, Continuous inspection schemes, Biometrika, 41 (1954), 100–115.
- [31] P. Papantoni-Kazakos and A. T. Burrell, *Performance monitoring in sensor networks*, in Proceedings of the 6th International Conference on Information Technology: New Generations (ITNG '09), Las Vegas, NV, 2009, 607–612.
- [32] P. Papantoni-Kazakos and A. T. Burrell, *Robust sequential algorithms for the detection of changes in data generating processes*, Journal of Intelligent & Robotic Systems, 60 (2010), 3–17.
- [33] T. Papantoni-Kazakos and A. T. Burrell, *The implementation of dynamic rate allocation in sensor networks*, Journal of Intelligent & Robotic Systems, 58 (2010), 211–238.
- [34] M. Paterakis and P. Papantoni-Kazakos, A simple window random access algorithm with advantageous properties, IEEE Trans. Information Theory, 35 (1989), 1124–1130.
- [35] A. J. Paulraj and C. B. Papadias, *Space-time processing for wireless communications*, IEEE Signal Processing Magazine, 14 (1997), 49–83.
- [36] S. Rastegar, A. Babaeian, M. Bandarabadi, and Y. Toopchi, *Airplane detection and tracking using wavelet features and SVM classifier*, in Proceedings of the 41st Southeastern Symposium on System Theory (SSST '09), 2009, 64–67.
- [37] C. W. Reed, R. Hudson, and K. Yao, *Direct joint source localization and propagation speed estimation*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '99), 1999, 1169–1172.
- [38] V. Rodoplu and T. H. Meng, *Minimum energy mobile wireless networks*, IEEE J. Selected Areas in Communications, 17 (1999), 1333–1344.
- [39] F. Salem, Dynamic architecture reconfiguration algorithms and transmission protocols for clustered sensor network topologies with prioritized data, master's thesis, University of Colorado Denver, 2011.
- [40] F. Salem, A. T. Burrell, and P. Papantoni-Kazakos, *Dynamic architectural reconfigurations of sensor networks*, in Proceedings of the 36th Annual Conference on IEEE Industrial Electronics Society (IECON '10), Glendale, AZ, 2010, 1117–1122.
- [41] A. M. Sayeed and D. L. Jones, Optimal detection using bilinear time-frequency and time-scale representations, IEEE Trans. Signal Processing, 43 (1995), 2872–2883.

- [42] J. W. Shin, H. J. Kwon, S. H. Jin, and N. S. Kim, Voice activity detection based on conditional MAP criterion, IEEE Signal Processing Letters, 15 (2008), 257–260.
- [43] J. Sohn, N. S. Kim, and W. Sung, A statistical model-based voice activity detection, IEEE Signal Processing Letters, 6 (1999), 1–3.
- [44] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, Protocols for self-organization of a wireless sensor network, IEEE Personal Communications, 7 (2000), 16–27.
- [45] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao, *Cooperation in wireless ad hoc networks*, in Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03), vol. 2, 2003, 808– 817.
- [46] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, *Distributed topology control for power efficient operation in multihop wireless ad hoc networks*, in Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01), vol. 3, 2001, 1388–1397.
- [47] K. Yao, R. E. Hudson, C. W. Reed, T. L. Tung, D. Chen, and J. C. Chen, *Estimation and tracking of an acoustic/seismic source using a beamforming arrays based on residual minimizing methods*, in Proceedings of the 1999 Meeting of the MSS Specialty Group on Battlefield Acoustic and Seismic Sensing, 1999, 129–136.
- [48] M. Yu, K. K. Leung, and A. Malvankar, A dynamic clustering and energy efficient routing technique for sensor networks, IEEE Trans. Wireless Communications, 6 (2007), 3069–3079.
- [49] M. Yu, A. Malvankar, and W. Su, An environment monitoring system architecture based on sensor networks, Int. J. Intell. Control and Syst., 10 (2005), 201–209.