

Communication Efficient Signal Detection in Correlated Clutter for Wireless Sensor Networks

Srikanth Hariharan	Leonardo R. Bachegea	Ness B. Shroff	Charles A. Bouman
Department of ECE	School of ECE	Departments of ECE and CSE	School of ECE
The Ohio State University	Purdue University	The Ohio State University	Purdue University
Columbus, OH - 43210	West Lafayette, IN - 47906	Columbus, OH - 43210	West Lafayette, IN - 47906
Email: harihars@ece.osu.edu	Email: lbachegea@purdue.edu	Email: shroff@ece.osu.edu	Email: bouman@purdue.edu

Abstract—We study a problem of detecting deterministic signals buried in correlated clutter using wireless sensor networks. We are specifically interested in developing a distributed algorithm over the network to detect the presence of a deterministic signal while keeping low communication delay and energy associated with the distributed computation. In this paper, we deploy a distributed version of the Sparse Matrix Transform (SMT) that decorrelates a signal measured by a number of sensors in order to compute a matched filter. The matched filter represents the sum of the Log-Likelihood Ratios over all the sensors of the two hypotheses corresponding to whether a deterministic signal is present or not. We show through numerical simulations that our algorithm is very efficient in terms of communication energy and delay while sustaining a high Signal-to-Clutter Ratio.

I. INTRODUCTION

We consider a Wireless Sensor Network (WSN) that comprises a number of sensors that are used to monitor the environment in which they are deployed. This network could potentially function as an ad-hoc network where sensors perform a number of operations without the necessity of a powerful centralized node, such as a base station. One of the important advantages of functioning in an ad-hoc manner is that the network may not require much maintenance. On the other hand, in order to minimize the amount of maintenance, sensors need to utilize resources in an efficient manner. For instance, sensors are battery-operated, and need to conserve energy. They also have limited computational resources. In this work, we are interested in one of the important functions of a sensor network, which is to perform signal detection. We wish to do this while utilizing sensor resources efficiently.

Hypothesis testing using the likelihood ratio test [1] is a well-known technique to perform signal detection. In a matched filter problem, one tests the hypothesis of the measurements containing only noise/clutter (null hypothesis) against the alternative hypothesis of the measurements containing a deterministic signal together with noise/clutter. In a WSN environment, since the sensor measurements are prone to errors, acquiring measurements from multiple sensors typically results in a better detection performance. However, measurements from multiple sensors need to be combined carefully

since they could potentially be correlated. Not only could the measurements be correlated, but also the measurement clutters. Therefore, in order to perform signal detection in an ad-hoc sensor network, sensors need to first decorrelate their measurements. Otherwise, the hypothesis testing could result in an erroneous result. In [2], [3], the authors study detection performance as the number of sensors becomes very large. Decentralized detection becomes difficult when measurements are correlated. This is because for an arbitrary covariance matrix, the computation of the matched filter requires global communication of sensor measurements either to a fusion center, or among all sensors in the network. While the former is computationally intensive (since the fusion center must perform all the computation), the latter requires a huge amount of communication.

The Sparse Matrix Transform (SMT) proposed in [4] can be used for decorrelation. One advantage of the SMT technique is that it can provide full-rank estimates of the signal's covariance matrix even when the number of training samples, n is much less than the signal dimensionality, p . In [5], the authors propose the *Graphical-SMT*, an algorithm to design the SMT for graphical data requiring only $O(p \log p)$ computation in average. Once designed, the computational complexity of applying the SMT decorrelating transform to test measurements is $O(p)$.

In this paper, we show how the SMT decorrelating transform can be used in a network of sensors to compute a distributed decorrelation of a measured signal. We analyze the communication costs required by this distributed SMT decorrelation algorithm in terms of communication energy and delay.

We compare our method with two alternatives: (1) Using a shrinkage estimate of the clutter covariance matrix; (2) assuming the clutter is uncorrelated. In our simulations, sensors are deployed to measure a deterministic signal buried in correlated Gaussian random clutter. The sensors communicate among each other using a tree topology. Our results suggest that the our SMT-based method gives the best detection accuracy while keeping the communication costs low.

The main contributions of our work are as follows.

- We design an algorithm for performing signal detection by decorrelating measurements from multiple sensors in a distributed manner in a multi-hop wireless network with

This material is based upon work supported by, or in part by, the U.S. Army Research Laboratory and the U.S. Army Research Office under contract/grant number 56541-CI and ARO MURI Award W911NF-07-1-0376 (SA08-03).

a tree topology. Our algorithms are based on designing SMTs to estimate the covariance matrix, and applying this estimate to test measurements in a distributed manner.

- We analyze the communication costs (in terms of energy and delay) of our algorithm, and show through extensive numerical evaluations that it performs significantly better both in terms of accuracy, and communication costs compared to a number of other existing techniques.

The rest of the paper is organized as follows. In Section II, we describe the system model for the wireless sensor network. In Section III, we provide an overview of the SMT method. In Section IV, we develop the SMT-based distributed signal decorrelation in a multi-hop wireless sensor network, and explain how a signal can be detected using our technique. In Section V, we perform extensive numerical evaluations of our algorithms. Finally, in Section VI, we conclude our work, and consider open problems.

II. SYSTEM MODEL

We consider a multi-hop wireless sensor network organized in the form of a tree, consisting of a number of sensors. The sensors communicate information to a cluster-head (which is the root of the tree) in possibly a multi-hop fashion. Note that the cluster-head is just another sensor, and not a powerful node compared to other nodes in the network. One could, for instance, periodically assign different sensors as cluster-heads. The purpose of the cluster-head is, for instance, to take a decision on whether a signal has been detected, based on the hypotheses ratios received from multiple sensors.

We assume error-free links. The energy consumed by communicating from one sensor to another is a function of the link costs in the path between the sensors. In order to evaluate the delay performance of our algorithms, we consider unit capacity links, and a time-slotted system.

III. THE SPARSE MATRIX TRANSFORM (SMT)

The SMT [4] is used to provide full-rank estimates of the clutter covariance $p \times p$ used during detection, discussed in Section IV. Here we review the concepts of designing and applying the SMT.

A. Design of the SMT transform

The SMT design consists of estimating the full set of eigenvectors and associated eigenvalues for a general p -dimensional signal. More specifically, the objective is to estimate the orthonormal matrix E and diagonal matrix Λ such that the signal covariance can be decomposed as $R = E\Lambda E^t$, and to compute this estimate from n independent training vectors, $Y = [y_1, \dots, y_n]$. This is done by assuming the samples are i.i.d. Gaussian random vectors and computing the constrained maximum log-likelihood (ML) estimates of E and Λ . In [4], we show that these constrained ML estimates are given by

$$\hat{E} = \arg \min_{E \in \Omega_K} \{ |\text{diag}(E^t S E)| \} \quad (1)$$

$$\hat{\Lambda} = \text{diag}(\hat{E}^t S \hat{E}), \quad (2)$$

where $S = \frac{1}{n} Y Y^t$ is the sample covariance matrix, and Ω_K is the set of allowed orthonormal transforms.

If $n > p$ and Ω_K is the set of all orthonormal transforms, then the solution to (1) and (2) is the diagonalization of the sample covariance, i.e., $\hat{E} \hat{\Lambda} \hat{E}^t = S$. However, the sample covariance is a poor estimate of the covariance when $n < p$.

In order to improve the accuracy of the covariance estimate, we will impose the constraint that Ω_K be the set of sparse matrix transforms (SMT) of order K . More specifically, we will assume that the eigen-transformation has the form

$$E = \prod_{k=1}^K E_k = E_1 \cdots E_K, \quad (3)$$

where each E_k is a planar rotation (known as a Givens rotation) over a coordinate pair (i_k, j_k) by an angle θ_k , and K is the model order parameter. So, $E_k = I + \Theta(i_k, j_k, \theta_k)$, where

$$[\Theta]_{ij} = \begin{cases} \cos(\theta_k) - 1 & \text{if } i = j = i_k \text{ or } i = j = j_k \\ \sin(\theta_k) & \text{if } i = i_k \text{ and } j = j_k \\ -\sin(\theta_k) & \text{if } i = j_k \text{ and } j = i_k \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Intuitively, each Givens rotation, E_k , plays the same role as the butterflies of a fast Fourier transform (FFT). In fact, the SMT is a generalization of both the FFT and the orthonormal wavelet transform. However, since both the ordering of the coordinate pairs, (i_k, j_k) , and the values of the rotation angles, θ_k , are unconstrained, the SMT can model a much wider range of transformations. It is often useful to express the order of the SMT as $K = rp$, where r is the average number of rotations per coordinate, being typically very small: $r < 5$. The optimization of (1) is non-convex, so we use a greedy optimization approach in which we select each rotation, E_k , in sequence to minimize the cost. The greedy optimization can be done fast if a graphical constraint can be imposed to the data [5]. The parameter r can be estimated using cross-validation over the training set [4], [5] or using the information criterion proposed in [6].

B. Application of the SMT transform

Typically, r is small (< 5), so that the computation to apply the SMT to a vector of data is very low, i.e., $2r + 1$ floating-point operations per coordinate. Therefore, we can apply the SMT decorrelating transform to p -dimensional random vectors in only $(2r + 1)p$ steps.

IV. WSNS AND COMMUNICATION CONSTRAINTS

In this section, we explain how to apply a SMT decorrelating transform to a multi-dimensional signal measured by a multi-hop wireless network with a tree topology. The main purpose is to use this decorrelating transform as part of a matched filter detection as explained below. We assume that the SMT decorrelating transform had been previously designed from training data as described in Section III-A using one of the methods proposed in [4], [5].

A. Distributed matched filter with the SMT

In our setup, each sensor node has a sequence of rotations containing the nodes (i_k and j_k) performing each of these rotations, and the angle (θ_k) of each rotation. These parameters are obtained from the SMT design. Let $d \in \mathbb{R}^p$ be a deterministic signal buried in additional random clutter $w \sim \mathcal{N}(0, R)$. The random vector x is measured, and one wants to make a decision whether the signal d is present, i.e., $x = d + w$, or the measurement only contains clutter, i.e., $x = w$. This can be done by testing the following hypotheses using a Log-Likelihood Ratio (LLR) test according to a *Neyman-Pearson* criterion [1].

$$\mathcal{H}_0 : x \sim \mathcal{N}(0, R) \quad (5)$$

$$\mathcal{H}_1 : x \sim \mathcal{N}(d, R) \quad (6)$$

In this case, the LLR test has the form of an inner-product: $l(x) = q^t x \geq \eta$, where the vector $q \triangleq R^{-1}d$ is called a *matched filter*, and its detection capability can be obtained by the Signal-to-Clutter Ratio (SCR) [7]:

$$SCR = \frac{(q^t d)^2}{E[(q^t x)^2]} = \frac{(q^t d)^2}{q^t E[xx^t] q} = \frac{(q^t d)^2}{q^t R q}. \quad (7)$$

Note that a node i only has the i^{th} elements of x and d . Therefore, computing $l(x)$, in general, requires global communication of these elements either to the cluster-head, or to all nodes in the network. However, using the graphical SMT, we can compute $l(x)$ in a distributed manner. We do this as follows.

We first rewrite $l(x)$ in the following manner.

$$\begin{aligned} l(x) &= q^t x \\ &= (R^{-1}d)^t x \\ &= (d^t E) \Lambda^{-1} (E^t x) \\ &= z^t \Lambda^{-1} y \\ &= \sum_{i=1}^p z_i \lambda_i^{-1} y_i \end{aligned}$$

where $z^t = d^t E \Lambda^{-1}$, and $y = E^t x$. Clearly, if we determine the i^{th} components of the vectors z and y , i.e. z_i and y_i , at node i , we can compute $z_i \lambda_i^{-1} y_i$ at node i . Each node will then calculate this sum from their sub-trees, and send the sum to their parent. This process terminates at the root of the tree, and the cluster-head will obtain $l(x)$. It can then compare it to η , and make a decision whether signal d was detected. We now explain how to determine z_i and y_i in a distributed manner.

Recall that $E = E_1 E_2 \dots E_K$. Initialize $y_i = x_i$, and $z_i = d_i$, for each node i . For $k = 1$ to K ,

- 1) Suppose that i_k and j_k are the nodes that perform a Givens rotation at the k^{th} rotation. Then, both i_k and j_k possess E_k .
- 2) Node i_k communicates t_{i_k} and x_{i_k} to node j_k , and vice versa.

- 3) Since E_k is a sparse matrix with non-zero elements only at the coordinates (i_k, i_k) , (i_k, j_k) , (j_k, i_k) , and (j_k, j_k) , by only exchanging the above messages between i_k and j_k , we can update y_{i_k} and y_{j_k} in the following manner.

$$y_{i_k} \leftarrow \cos(\theta_k) y_{i_k} - \sin(\theta_k) y_{j_k}, \quad (8)$$

$$y_{j_k} \leftarrow \cos(\theta_k) y_{j_k} + \sin(\theta_k) y_{i_k}. \quad (9)$$

Note that z_{i_k} and z_{j_k} can be updated in a similar manner.

Thus, by only exchanging messages between the nodes that decorrelate during each rotation, we can determine $l(x)$.

B. Communication costs

We are interested in two metrics for the communication cost - energy and delay. The communication energy is the total energy spent by all the nodes in the network for applying the distributed SMT decorrelating transform on test data. The delay is the number of time slots required to apply the distributed decorrelating transform on the given test data.

Computing Energy: Let the energy required to make a transmission over a link l be β_l . β_l could depend on a number of factors such as the distance between the nodes, the SNR required, etc. For each rotation k , two nodes i_k and j_k have to exchange packets. The energy required to send a packet from i_k to j_k is given by $\sum_{l \in P_{i_k j_k}} \beta_l$, where $P_{i_k j_k}$ is the set of links on the path from i_k to j_k . Therefore, the total energy required by a rotation is given by $2 \sum_{l \in P_{i_k j_k}} \beta_l$, since i_k sends a packet to j_k , and vice versa. On the average, in a tree, there are $O(\log p)$ links in the path from one node to another. Therefore, it requires $O(\log p)$ energy for one rotation. Therefore, the total energy required for performing K rotations is $O(K \log p)$. Since K is typically $O(p)$, the total energy required for applying the SMT to perform signal detection is $O(p \log p)$.

Computing Delay: While the energy required is related to the total number of transmissions in the network, the delay to apply the SMT depends on whether transmissions can be performed simultaneously. Assuming unit capacity links, the time required to perform a rotation between two nodes i_k and j_k is given by $2|P_{i_k j_k}|$, i.e., twice the number of hops between i_k and j_k . However, the total time required to perform all K rotations is not the sum of the time required for individual rotations. For instance, if sensor 1 and sensor 2 perform the first rotation, and sensor 10 and sensor 11 perform the second rotation, then these nodes can simultaneously do these rotations, requiring an overall delay of $\max(2|P_{1,2}|, 2|P_{10,11}|)$. On the other hand, if sensor 2 and sensor 3 perform the second rotation instead, the two rotations have to be performed sequentially because y_2 and z_2 get updated after the first rotation. We use the following algorithm to schedule rotations.

- 1) Initialize t_{start}^i and t_{end}^i for each rotation i to be zero. t_{start}^i represents the time slot at which the message exchange for rotation i starts, and t_{end}^i is the time slot at which it ends.

- 2) For each rotation k , look at the list of previous rotations from 1 to $k - 1$. Find the last rotation i in the list from 1 to $k - 1$ such that no rotation in the list from $i + 1$ to $k - 1$ involves either of the nodes in rotation k . Suppose that i_k and j_k are involved in rotation k . Then t_{start}^k and t_{end}^k are given by

$$t_{start}^k \leftarrow t_{end}^i + 1 \quad (10)$$

$$t_{end}^k \leftarrow t_{start}^k + 2|P_{i_k j_k}| \quad (11)$$

- 3) The overall delay is given by $\max_i(t_{end}^i)$.

V. EXPERIMENTAL RESULTS

We now evaluate our algorithms numerically to study its decorrelation performance, and the communication costs associated with it. Before we do this, we first give a brief description of other methods that we compare our method with.

- **Independent:** This method assumes that sensors make independent measurements, and are not correlated. While this method requires the lowest communication costs, it gives the poorest performance when measurements are correlated, which is typical in practice.
- **Sample covariance:** As explained in Section III, the sample covariance matrix is given by $S = \frac{1}{n}YY^t$. However, the matrix \hat{E} in the diagonalization $S = \hat{E}\hat{\Lambda}\hat{E}^t$ is not necessarily a product of sparse matrices. Therefore, in order to apply this estimate to test data in a distributed manner, each node i has to communicate its x_i and d_i to every other node in the tree.
- **Shrinkage estimators:** Shrinkage methods[8], [9] are widely used to estimate high-dimensional covariance matrices. They work by estimating the covariance matrix as a combination of the rank-deficient sample covariance and a positive definite target such as the identity matrix or the diagonal of the sample covariance. Let S be the $p \times p$ sample covariance estimate of the covariance R computed from n samples. The Shrinkage-Identity estimate is given by

$$\tilde{R} = \alpha S + (1 - \alpha) \frac{\text{tr}(S)}{p} I .$$

The Shrinkage-Diagonal is given by

$$\tilde{R} = \alpha S + (1 - \alpha) \text{diag}(S) .$$

The intuition behind these methods is that a combination of an estimator that over-fits the data (i.e, the rank-deficient sample covariance) with an estimator that under-fits the data (i.e, the $p \times p$ identity matrix I , or the diagonal matrix $\text{diag}(S)$) will produce a more accurate final estimate. The right value of the parameter α is usually determined using cross-validation. The eigenvector matrix \hat{E} in the diagonalization $\tilde{R} = \hat{E}\hat{\Lambda}\hat{E}^t$ is dense, therefore, the matrix-vector product $\hat{E}^t y$ required

to decorrelate a vector y requires as much communication as the sample covariance method.

We consider a full binary tree with 63 nodes for the following results. We assume that correlations decreases exponentially with distance, i.e., the actual covariance matrix is given by $R = e^{-D}$, where D is the matrix of physical distances between nodes. We normalize this matrix so that its determinant is one. We generate Gaussian distributed random samples with mean zero, and covariance R for designing the SMT. We generate additional additional samples to test the decorrelation performance.

For calculating the energy, we assume that the cost to communicate one sample over each link is one unit. Similarly, since we assumed unit capacity links, the delay required to communicate one sample over a link is also one unit.

A. SCRR vs. Measurement Samples

One of the metrics for analyzing the decorrelation performance is the *Signal-to-Clutter Ratio Ratio* (SCRR) which is the ratio of the SCR of our algorithm to that of a perfect decorrelation algorithm. Note that a perfect decorrelation algorithm would require infinite measurement samples, and is therefore not practically feasible.

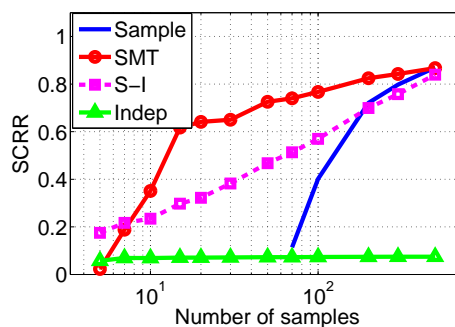


Fig. 1. SCRR vs. #Measurement Samples

Figure 1 shows the SCRR for various amounts of measurement samples used for design. One can see that the SMT performs better than all the other estimators. Also, observe that the sample covariance estimator requires at least p ($= 63$ here) samples for estimating the covariance matrix. The “Independent” estimator assumes that there are no correlations, and therefore performs poorly compared to all the other estimators.

B. SCRR vs. Energy

We now compare the energy required by these methods for application of the respective estimations of the covariance matrices. Figure 2(a) illustrates how much energy is required for given values of SCRR for all these methods. For this experiment, we use $n = 500$ samples for designing the covariance matrix estimators. As explained before, in order to apply the covariance matrices estimated by either the Shrinkage+I or the sample covariance method, each node i needs to communicate its x_i to all the other nodes in the

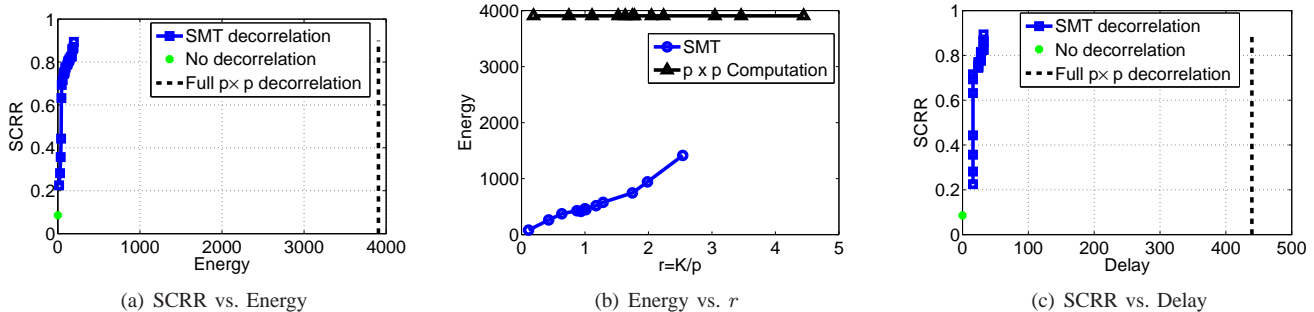


Fig. 2. Communication Performance

network. This is because E need not be a sparse matrix in these cases. Since in a tree with p nodes, there are $p-1$ links, it requires $p-1$ total transmissions in order for every node in the network to obtain x_i . Therefore, these techniques require $O(p^2)$ total transmissions so that every node has the vector x .

We observe that for the same SCRR, the SMT requires significantly lower energy than the other methods. The “Independent” estimator requires an extremely low amount of energy, however it provides very poor SCRR.

C. Energy vs. Rotations

Using the MDL criterion [4] for determining the model parameter K in the design of the SMT, we compare the energy required for application of the respective estimations of the covariance matrices as a function of $r = \frac{K}{p}$. Here, we vary the number of measurement samples for designing the transforms, and obtain K for each value using the MDL criterion. We see that r is always < 3 (Figure 2(b)). As in the previous experiment, the Shrikange+I and the sample covariance estimators require $O(p^2)$ energy for application. The energy required by the SMT is significantly lower than that required by the other methods.

D. SCRR vs. Delay

We finally study the delay performance of our algorithms. Here, the delay is the time required to apply these methods to compute $l(x)$. We again use $n = 500$ samples for designing the covariance matrix estimators. As explained earlier, for the SMT, the time required depends on which rotations can be done simultaneously, and which needs to be done sequentially.

Figure 2(c) shows how the SMT algorithms compare with the other methods. The SMT performs significantly better in terms of delay compared to the sample covariance method, and the shrinkage method, both of which require $p \times p$ computation.

VI. CONCLUSION

We studied a signal detection problem in a multi-hop wireless network with a tree topology. We considered sensor measurements with correlated clutter, and designed distributed algorithms using Sparse Matrix Transforms that could be implemented in these networks in a decentralized manner to perform signal detection. We provided extensive numerical evaluations of our algorithms, and showed that it performs

significantly better than existing methods in terms of decorrelation performance, energy, and delay. A number of interesting open problems need to be addressed. Communication needs to be optimized. For instance, it would be interesting to determine if storing packets in intermediate nodes on the path between two nodes would result in energy savings. One could also study about implementing these algorithms in a wireless network with an arbitrary topology. The problem becomes much harder in these networks since routing and scheduling are significantly more difficult in networks with arbitrary topologies.

REFERENCES

- [1] S. M. Kay, *Fundamentals of Statistical Signal Processing: Vol. 2: Detection Theory*. Prentice-Hall, Inc., 1998.
- [2] J. Chamberland and V. V. Veeravalli, “Decentralized detection in sensor networks,” *IEEE Transactions on Signal Processing*, 2003.
- [3] W. Li and H. Dai, “Distributed detection in large-scale sensor networks with correlated sensor observations,” in *Allerton Conference on Communication, Control and Computing*, 2005.
- [4] G. Cao and C. A. Bouman, “Covariance estimation for high dimensional data vectors using the sparse matrix transform,” in *Neural Information Processing Systems Conference*, 2008.
- [5] L. R. Bacheaga, G. Cao, and C. A. Bouman, “Fast signal analysis and decomposition on graphs using the sparse matrix transform,” in *International Conference on Acoustic, Speech, and Signal Processing (ICASSP)*, 2010.
- [6] J. Theiler, G. Cao, L. R. Bacheaga, and C. A. Bouman, “Sparse matrix transform for hyperspectral image processing,” *Journal of Selected Topics in Signal Processing*.
- [7] G. Cao, C. A. Bouman, and J. Theiler, “Weak signal detection in hyperspectral imagery using sparse matrix transformation (smt) covariance estimation,” in *First Workshop on Hyperspectral Image and Signal Processing*, 2009.
- [8] M. J. Daniels and R. E. Kass, “Shrinkage estimators for covariance matrices,” *Biometrics*, vol. 57, 2001.
- [9] O. Ledoit and M. Wolf, “Improved estimation of the covariance matrix of stock returns with an application to portfolio selection,” *Journal of Empirical Finance*, vol. 10, 2003.