

Expectation Maximization Aided Modified Weighted Sequential Energy Detector for Distributed Cooperative Spectrum Sensing

Mohammed Rashid, *Member, IEEE*, and Jeffrey A. Nanzer, *Senior Member, IEEE*

Abstract

Distributed cooperative spectrum sensing usually involves a group of unlicensed secondary users (SUs) collaborating to detect the primary user (PU) in the channel, and thereby opportunistically utilize it without causing interference to the PU. The conventional energy detector (ED) based spectrum sensing ignores the dynamic nature of the PU by using energy statistic only from the present sensing interval for the PU detection. However, for a dynamic PU, previous studies have shown that improved detection capabilities can be achieved by aggregating both present and past energy samples in a test statistic. To this end, a weighted sequential energy detector (WSED) has been proposed, but it is based on aggregating all the collected energy samples over an observation window. For a highly dynamic PU, that involves also combining the outdated samples in the test statistic. In this paper, we propose a modified WSED (mWSED) that uses the primary user states information over the window to aggregate only the highly correlated energy samples in its test statistic. In practice, since the PU states are a priori unknown, we also develop a joint expectation-maximization and Viterbi (EM-Viterbi) algorithm based scheme to iteratively estimate the states by using the energy samples collected over the window. The estimated states are then used in mWSED to compute its test statistics, and the algorithm is referred to here as EM-mWSED. Simulation results are presented to demonstrate the states estimation performance

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

This effort was sponsored in whole or in part by the Central Intelligence Agency (CIA), through CIA Federal Labs. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Central Intelligence Agency. (*Corresponding author: Jeffrey A. Nanzer.*)

The authors are with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824 (e-mail: rashidm@ieee.org, nanzer@msu.edu).

of EM-Viterbi and the PU detection performance of EM-mWSED. The results show that, for both highly dynamic as well as slowly time-varying PU, these algorithms outperform the ED and WSED at PU detection, and their performances improve by either increasing the average number of neighbors per SU in the network, or by increasing the SNR or the number of samples per energy statistic.

Index Terms

Cognitive radio systems, dynamic primary user, distributed cooperative spectrum sensing, expectation maximization, energy detector, modified weighted sequential energy detector.

I. INTRODUCTION

A cognitive radio system is an intelligent wireless communication system that learns from its surrounding radio environment and adapts its operating parameters (e.g., carrier frequency, transmit power, and digital modulation scheme) in real-time to the spatiotemporal variations of the RF spectrum. The primary objective of the cognitive scheme is to enable the unlicensed (secondary) users to opportunistically utilize the spectrum owned by the licensed (primary) users, where the reconfigurability of the radio is accomplished using software-defined radio based platforms [1]. Since the secondary users (SUs) are a lower priority for spectrum access than the primary users (PUs), an indispensable condition for the SUs is to avoid causing interference to the PUs during their spectrum use, which can be achieved by improving the spectrum sensing capabilities of the SUs. Several sensing algorithms have been reviewed in [2], [3] for cognitive radio systems. Among them, matched filtering is considered an optimal method when the PU's transmitted signal is known to the SUs; however, when the signal knowledge is not available then energy detection emerges out as a favorable choice due to its low computational and implementation complexities.

Spectrum sensing (or PU detection) can be done by the SUs either by using a non-cooperative scheme or a cooperative scheme. In a non-cooperative scheme, each SU performs PU detection individually without any direct communication with the other SUs or a fusion center (FC). In contrast, in cooperative spectrum sensing, a group of SUs communicate with each other or with a fusion center to collaboratively perform the PU detection. Consequently, in comparison, the cooperative sensing approach is resilient to the deep fading and shadowing at an SU level, aids in eliminating the hidden terminal problem, reduces the sensing time per SU, and demonstrates a better performance for PU detection [4], [5].

Cooperative spectrum sensing schemes can be further categorized into either a centralized scheme or a distributed scheme. In a centralized scheme, a fusion center collects the sensing information from the SUs, detects the unused band, and broadcasts that information via a control channel to the SUs [4], [6]–[8]. However, the centralized approach is not scalable to large networks as the available communication resources are limited at the FC. Furthermore, an FC involvement defines a single point of failure for the centralized network. In comparison, in a distributed scheme, the SUs share their sensing statistics with their neighboring users in the network and use a consensus protocol to collaboratively decide on the presence or absence of PU in the channel [9], [10]. This approach not only eliminates the single point of failure from the network, but it is also scalable as the communication resources need to be shared only among the neighboring users.

The distributed cooperative spectrum sensing (DCSS) scheme usually has three critical phases, namely the sensing phase, the consensus phase, and the transmission/wait phase. In the sensing phase, a group of SUs observes the same PU channel for a certain time duration to collect a sufficient number of samples for computing the summary statistics (e.g., energy statistics [3], [11]). Next, in the consensus phase, the SUs locally share their summary statistics and use, e.g., an average consensus protocol [12], [13] to iteratively compute a weighted average of the globally shared values across the network. Upon consensus in such an approach, the final value is compared against a threshold at each SU to locally detect the presence or absence of the PU in the channel. Finally, in the transmission/wait phase, the detection outcome is used to either transmit in the channel or wait for some duration before restarting the cycle. This DCSS scheme was proposed in [9] wherein the authors analyzed its convergence speed as well as the detection performance for varying false alarm rates. In [14], [15], DCSS was extended to protect against the eavesdropper attack by encrypting the summary statistics shared between the SUs, whereas in [16], [17], the authors considered the scenarios in which some malicious SUs (aka Byzantines) may inject falsified data into the network and proposed a data-driven approach to mitigate the Byzantine attacks in DCSS.

The above-mentioned DCSS algorithms use the conventional approach in which each SU uses energy samples only from the current sensing time period to make the PU detection. However, in the case of a dynamic PU whose activity varies over the consecutive sensing periods, aggregating present and past samples at each SU usually results in an improved detection performance. In [7], [8], the dynamic PU is modeled using a two-state Markov chain model and a weighted sequential

energy detector (WSED) is proposed in which the present and past samples over an observation window are weighted appropriately and aggregated to achieve improved detection capability. For a slowly varying PU, equal weighting of the samples is suggested whereas for a highly dynamic PU, exponential weighting is proposed to reduce the impact of out-dated measurements. For the highly dynamic PU, a two-stage detector is also proposed in [8] in which a threshold is used at the first stage to detect the change in the PU's state between the consecutive sensing periods, based on which a decision is made to either include or ignore completely the past out-dated samples in the WSED statistic. However, due to hard detection on the first stage and exclusion of all the past samples during a state change, only a slight improvement in performance was observed with the two-stage detector as compared to WSED. Finally, [7], [8] assume a centralized scheme for cooperative spectrum sensing which as discussed before is not a scalable approach.

In this paper, we also consider the problem of DCSS in which the PU follows a two-state Markov chain model for switching between the active and idle states over the consecutive sensing periods [8]. However, a modified WSED (mWSED) is proposed in which instead of aggregating all the present and past samples over an observation window¹, we aggregate only those samples that correspond to the state of the PU in the present sensing period. An underlying assumption in mWSED is that the actual states of the PU are known over the observation window. In practice, the states are unknown, and thus we also develop an algorithm to iteratively estimate them using the samples collected over the window. Specifically, we first develop an expectation maximization (EM) based algorithm to estimate the model parameters of the joint probability distribution over the observation and the state vectors. Next, using their estimate, we use the Viterbi algorithm [18] to estimate the state vector by the maximization and back tracing operations. The estimated state vector produced by the joint EM and Viterbi (EM-Viterbi) algorithm is then used in mWSED to aggregate only the highly correlated samples in its test statistic. This approach avoids aggregating the outdated samples in computing the detection statistic and thus manifests a better detection performance than WSED for a dynamic PU. Since EM is the main algorithm that enables the use of the Viterbi algorithm, the resulting algorithm is referred to here as the EM-mWSED algorithm. For fair comparison, the WSED algorithm of [8] is also extended to the DCSS scheme. Simulation results are included which show that both EM-Viterbi and EM-mWSED outperform WSED and

¹An observation window is defined herein as a vector of length D containing all the energy detection statistics from the $D - 1$ past sensing periods as well as the energy statistic from the present sensing period.

the conventional energy detector for both slowly varying PU and a highly dynamic PU in all the considered scenarios. Furthermore, the results demonstrate that their performances improve by either increasing the average number of connections per SU in the network, or by increasing the SNR or the number of samples per energy statistics.

This paper is outlined as follows. Section II provides a brief review of the energy detection based spectrum sensing. Distributed cooperative spectrum sensing is discussed in Section III, also including a review of WSED and presentation of our proposed mWSED. Next, Section IV delivers an expectation maximization and Viterbi algorithm based scheme for estimating the PU states over an observation window and using it with mWSED. Simulation results are presented in Section V. Finally, Section VI summarizes this work.

II. ENERGY DETECTOR BASED SPECTRUM SENSING

We consider a distributed spectrum sensing system in which a network of N SUs are spatially distributed and cooperating with each other to detect the PU in the channel. As discussed in the previous section, we assume that the SUs deploy an energy based statistic to sense the channel. As such, the energy computed by an i -th SU under the null hypothesis (H_0) and the alternate hypothesis (H_1) can be written as follows.

$$x_i = \begin{cases} \sum_{l=1}^L |n_{i,l}|^2, & \text{if } H_0 \\ \sum_{l=1}^L |h_i s_l + n_{i,l}|^2, & \text{if } H_1 \end{cases} \quad (1)$$

in which L is the total number of samples collected over the sensing interval, h_i is the channel gain for SU i , s_l represents the PU signal at time index l , and finally, $n_{i,l}$ denotes the noise in the sensing interval which is assumed to be normally distributed with zero mean and variance σ_n^2 . The signal to noise ratio (SNR) at the SU is defined by $\eta_i = \frac{\sum_{l=1}^L |h_i s_l|^2}{\sigma_n^2}$ which is L times the SNR at the output of the energy detector.

Now when the PU is idle under H_0 , i.e., the channel is unoccupied, the normalized energy statistics $z_i = \frac{x_i}{\sigma_n^2}$ follow a central chi-square distribution with L degrees of freedom. Its probability density function (pdf) is written as

$$p(z_i|H_0) = \frac{z_i^{\frac{L}{2}-1} e^{-\frac{z_i}{2}}}{2^{\frac{L}{2}} \Gamma\left(\frac{L}{2}\right)}, \quad (2)$$

for $z_i \geq 0$, in which $\Gamma(\cdot)$ is the gamma function [19]. Using the pdf in (2), the probability of false alarm can be computed in closed-form as

$$P_f(\lambda) = \int_{\lambda}^{\infty} p(z_i|H_0) dz_i = \frac{\Gamma\left(\frac{L}{2}, \frac{\lambda}{2}\right)}{\Gamma\left(\frac{L}{2}\right)}, \quad (3)$$

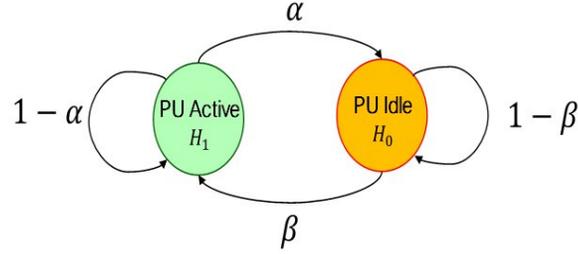


Fig. 1. A graphical representation of the two-state Markov chain model describing the change in primary user activity over the sensing intervals. The parameters α and β represent the transition probabilities of switching between the two states in the Markov model.

where λ is the threshold for energy detection, and $\Gamma(\cdot, \cdot)$ is the upper incomplete gamma function [19]. Note that for a selected value of false alarm probability, the threshold λ can be computed from (3) by using the inverse of the incomplete gamma function. In contrast, when the PU is active under H_1 , i.e., the channel is busy, then z_i follows a non-central chi-square distribution with L degrees of freedom and a non-centrality parameter η_i . Thus, its pdf is written as

$$p(z_i|H_1) = \frac{e^{-\frac{\eta_i}{2}} F_{0,1}\left(\frac{L}{2}, \frac{\eta_i z_i}{4}\right)}{2^{\frac{L}{2}} \Gamma\left(\frac{L}{2}\right)} e^{-z_i/2} z_i^{\frac{L}{2}-1}, \quad (4)$$

for $z_i \geq 0$, where $F_{0,1}(\cdot, \cdot)$ is the hypergeometric function [19] and parameter η_i is the SNR at the i -th SU. Assuming a Rayleigh fading channel, the probability of detection for the SU can be computed in a closed-form as follows [8].

$$\begin{aligned} P_d(\lambda) &= \int_{\lambda}^{\infty} p(z_i|H_1) dz_i \\ &= \frac{\Gamma\left(\frac{L}{2} - 1, \frac{\lambda}{2}\right)}{\Gamma\left(\frac{L}{2} - 1\right)} + e^{-\frac{\lambda}{2+\bar{\eta}_i}} \left(1 + \frac{2}{\bar{\eta}_i}\right)^{\frac{L}{2}-1} \left[1 - \frac{\Gamma\left(\frac{L}{2} - 1, \frac{\lambda \bar{\eta}_i}{4+2\bar{\eta}_i}\right)}{\Gamma\left(\frac{L}{2} - 1\right)}\right], \end{aligned} \quad (5)$$

where $\bar{\eta}_i = \frac{\sum_{l=1}^L \mathbb{E}[|h_i s_l|^2]}{\sigma_n^2}$ denotes the average value of the SNR due to randomness in the channel, and the threshold λ can be identified based on a selected P_f as discussed above.

Next we consider a practical scenario wherein the PU's activity is dynamic over the sensing intervals and follows a two-state Markov chain model as shown in Fig. 1. Specifically, the current state visited by the PU depends only upon its immediate previous state. Accordingly, in this figure, the parameter α denotes the transition probability of switching to an idle state (H_0) given that previously the PU was in the active state (H_1), whereas β represents the transition probability of switching to an active state (H_1) if previously the PU was in the idle state (H_0). Thus, higher

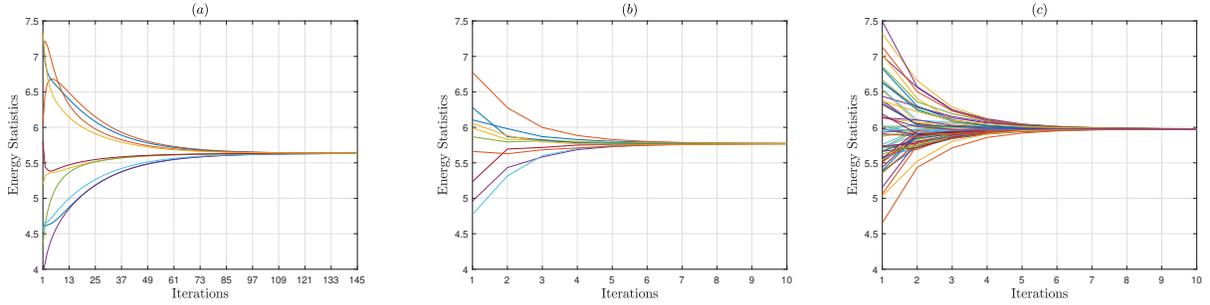


Fig. 2. Energy statistics of SUs vs. DCSS iterations for (a) $N = 10$, $c = 0.2$, (b) $N = 10$, $c = 0.5$, and (c) $N = 60$, $c = 0.2$, when $\text{SNR} = -3$ dB, number of samples per energy statistic $L = 12$, and the PU follows the two-state Markov model with $\alpha = \beta = 0.1$.

values of α and β imply a highly dynamic PU, whereas their smaller values represent its slowly time-varying behavior. For such modeling of the PU, a WSED algorithm was proposed in [7], [8] in which the SUs combine a fixed number of present and past observations (i.e., the energy statistics from (1)) to calculate a weighted sequential energy test statistic. The observations in WSED are weighted either by using uniformly distributed weights for a slowly varying PU, or exponentially distributed weights for a highly dynamic PU wherein the weights over the past observations are reduced exponentially by a factor of e . However, the authors in [7], [8] considered a centralized scheme for the cooperative spectrum sensing where the existence of a fusion center makes it a non-scalable approach. In the following, we first describe a scalable distributed cooperative spectrum sensing scheme as considered here, followed with a brief review of the WSED algorithm and our modified WSED algorithm.

III. DISTRIBUTED COOPERATIVE SPECTRUM SENSING

Consider a network of SUs represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in which $\mathcal{V} = \{1, 2, \dots, N\}$ is the set of N number of SUs in the network, and $\mathcal{E} = \{(i, j), \forall i, j \in \mathcal{V}\}$ represents the set of all possible bidirectional communication links between them. The connectivity of the network in any realization is denoted by c which is defined as a ratio of the number of active connections in the network (N_a) to the number of all possible connections among the SUs ($N(N-1)/2$). We consider a connected network in which the neighboring users that are one hop away from each other share their information to reach a consensus. Therefore, wireless and computing resources such as bandwidth, processing power, and data storage capabilities,

need to be only locally managed at the SUs, and scale proportionally to the average number of connections per SU in a network.

A distributed cooperative spectrum sensing algorithm is a scalable and a fully distributed approach which deploys a consensus protocol at an SU. The protocol iteratively updates the sensing information at the user, by using the locally shared information, to reach consensus with the other users in the network. To elaborate, let $y_i(0) = x_i$ represents the initial energy statistic for an i -th SU, then in iteration k of the DCSS algorithm, the SU updates its estimate by using a weighted average method as follows.

$$y_i(k) = y_i(k-1) + \sum_{j \in N_i} w_{ij}(y_j(k-1) - y_i(k-1)) \quad (6)$$

in which N_i is the set of neighboring users of the i -th SU. The weight w_{ij} can be selected as $w_{ij} = 1/\max(d_i, d_j)$ where d_i and d_j represent the number of neighbors of SU i and SU j , respectively. This selection of weights results in a doubly-stochastic Metropolis-Hasting weighting matrix which guarantees convergence of the consensus algorithm [13]. Thus, starting with a set of initial values $\{y_i(0), \text{ for } i = 1, 2, \dots, N\}$, the algorithm running locally at each SU iteratively updates the values using (6) until it converges to an average of the globally shared values across the network. The average value is defined by $y^* = \frac{\sum_{i=1}^N x_i}{N}$. Upon convergence, the decision can be made locally at the i -th SU by using the following rule

$$d_i = \begin{cases} H_1 & \text{if } y^* \geq \lambda \\ H_0 & \text{otherwise} \end{cases} \quad (7)$$

A. Simulation Results

Herein, we analyze the consensus performance of an energy detector (ED) based DCSS scheme when the network of SUs is randomly generated for different number of SUs N and with varying connectivity c . The primary user follows a two-state Markov chain model for switching states between H_0 and H_1 over the multiple sensing intervals. Each SU uses $L = 12$ samples for computing the energy statistic following (1), and has an SNR= -3 dB for the PU channel. In Fig. 2, we demonstrate the convergence performance of the ED-based DCSS algorithm for $N = 10$ and 60 users when the connectivity is either $c = 0.2$ or 0.5. It is observed that when $N = 10$ and $c = 0.2$, the consensus occurs in about 96 iterations, but when the connectivity increases to $c = 0.5$, it happens in about 7 iterations. Similar observation is made if N increases from 10 to 60 SUs for $c = 0.2$. This is because the average number of connections per SU is

represented by $R = c(N - 1)$, and thus when either c or N increases, then the local averages computed at the SUs using (6) are more accurate and stable resulting in the faster convergence speed.

Next, we first briefly review the WSED detector of [8] and discuss its extension for using it with the DCSS algorithm in (6). After that, we propose a modified WSED which as shown later in Section V outperforms the DCSS-based WSED algorithm.

B. Weighted Sequential Energy Detector

As proposed in [7], [8], the WSED algorithm computes a weighted sum of all the present and past samples over an observation window of length D to define a new test statistic, which for the i -th SU is given by

$$S_i = \sum_{d=1}^D w_d x_{i,d}, \quad (8)$$

where $x_{i,d}$ represents the energy statistic of the SU i (as in (1)) in the sensing interval d , with $x_{i,D}$ representing the energy at the present sensing interval. Thus, a total number of D present and past observations are combined in the WSED statistic. The weights obey $\sum_{d=1}^D w_d = 1$ and the authors in [7], [8] propose to use equal weights ($w_d = 1/D$) for a static PU and exponential weights ($w_d = e^d / \sum_{d=1}^D e^d$) for a highly dynamic PU. Specifically, the exponential weighting is motivated to reduce the impact of aggregating the outdated past samples in (8) in a highly dynamic scenario. However, a centralized scheme is considered in [7], [8] wherein the SUs forward their statistics in (8) to a fusion center where a decision is made using an OR rule. As the use of a fusion center is a non-scalable approach, so in Section V, we extend WSED to the DCSS scheme with consensus relating to the energy samples aggregated in (8) as in (6). This aids in improving the SNR at each SU as discussed in Section V. Thus, the decision can be made locally at each SU by comparing S_i against a threshold. Finally, as pointed out in [8], the exact closed-form expressions for the probability of detection and the probability of false alarm for WSED are in general intractable to compute analytically, due to the aggregation of observations that may correspond to different states of the PU. However, the authors in [8] have derived approximated expressions which are also applicable for the DCSS scheme based WSED.

$s_{i,1}$	$s_{i,2}$...	$s_{i,D}$
$x_{i,1}$	$x_{i,2}$...	$x_{i,D}$

Fig. 3. A notional view of the correspondence between the primary user states ($\{s_{i,d} \in \{0, 1\}, \forall d = 1, 2, \dots, D\}$) and the energy samples ($\{x_{i,d}, \forall d = 1, 2, \dots, D\}$) collected by the i -th secondary user in an observation window of length D .

C. Modified Weighted Sequential Energy Detector

In this subsection, we present our proposed modified WSED (mWSED). It is based on the motivation that instead of combining all the present and past energy samples upon consensus over the observation window of length D , we combine only those energy samples (observations) in the summary statistic that belong to the present state of the PU. As such, in mWSED, we begin by assuming that the states visited by the PU over the observation window are known to each SU. Notably, this assumption provides a starting point to derive mWSED, but later on in Section IV we also develop an EM algorithm to compute an estimate of those states at each SU, using the energy samples collected over the window as shown in Fig. 3. Thus, by comparing the present and past states over the observation window, each SU locally combines only those samples that correspond to the state of the PU in the present sensing interval. Therefore, the test statistics computed in mWSED at the i -th SU in the present D -th sensing interval is defined as

$$T_i = \sum_{d=1}^D x_{i,d} \mathbb{1}(s_{i,d} = s_{i,D}, w_d), \quad (9)$$

where $x_{i,d}$ represents the energy computed by SU i in the d -th sensing interval and $s_{i,d}$ is the PU's state in that interval with $s_{i,d} = 0$ denoting H_0 and $s_{i,d} = 1$ implying H_1 . $\mathbb{1}(A, w_d)$ is a weighted indicator function which outputs a non-zero weight w_d if A is true, and outputs $w_d = 0$ if A is false. Thus, given the state information of PU over the observation window, either $x_{i,d}$ is included or excluded from T_i . Parameter w_d is the weight assigned to the aggregated sample, and note that depending on the output of the indicator function, the non-zero weights on the aggregated samples in T_i can be distributed either uniformly or exponentially as for WSED. Finally, at each SU, the statistic T_i is compared against the threshold in Algorithm 1 to make the PU detection.

Now, we derive the expressions for the probability of false alarm and the probability of detection for mWSED as follows. Let C be the number of samples with non-zero weights in (9), then for the purpose of derivation, we use the fact that the sum of C independent chi-square random variables with L degrees of freedom is a chi-square random variable with CL degrees of freedom, and that the sum of C independent non-central chi-square random variables with L degrees of freedom and non-centrality parameter η is a non-central chi-square random variable with CL degrees of freedom and non-centrality parameter $C\eta$ [19]. Thus, when the PU is idle in the present sensing interval, the normalized statistic $z_i = \frac{T_i}{\sigma_n^2}$ follows a central chi-square distribution with CL degrees of freedom. Its pdf is given by

$$p(z_i|H_0) = \frac{z_i^{\frac{CL}{2}-1} e^{-\frac{z_i}{2}}}{2^{\frac{CL}{2}} \Gamma\left(\frac{CL}{2}\right)}, \quad (10)$$

for $z_i \geq 0$, and where $\Gamma(\cdot)$ is the gamma function [19]. Consequently, the probability of a false alarm can be computed in a closed-form as

$$\begin{aligned} P_f(\lambda) &= \int_{C\lambda}^{\infty} p(z_i|H_0) dz_i \\ &= \frac{\Gamma\left(\frac{CL}{2}, \frac{C\lambda}{2}\right)}{\Gamma\left(\frac{CL}{2}\right)} \end{aligned} \quad (11)$$

in which λ is the threshold for PU detection, and $\Gamma(\cdot, \cdot)$ is the upper incomplete gamma function [19]. Hence, by selecting a suitable value for the false alarm probability, the threshold λ can be estimated from the above equation using an inverse of the incomplete gamma function.

Next, when the PU is active and the channel is quasi-static, then z_i follows a non-central chi-square distribution with CL degrees of freedom and non-centrality parameter $C\eta$. Then, the pdf of z_i is given by

$$p(z_i|H_1) = \frac{e^{-\frac{C\eta}{2}} F_{0,1}\left(\frac{CL}{2}, \frac{C\eta z_i}{4}\right)}{2^{\frac{CL}{2}} \Gamma\left(\frac{CL}{2}\right)} e^{-\frac{z_i}{2}} z_i^{\frac{CL}{2}-1}, \quad (12)$$

for $z_i \geq 0$, and where $F_{0,1}(\cdot, \cdot)$ is the hypergeometric function [19]. In particular, when the channel is a Rayleigh fading channel, we can write the detection probability for mWSED using (5) as follows.

$$\begin{aligned} P_d(\lambda) &= \int_{C\lambda}^{\infty} p(z_i|H_1) dz_i \\ &= \frac{\Gamma\left(\frac{CL}{2} - 1, \frac{C\lambda}{2}\right)}{\Gamma\left(\frac{CL}{2} - 1\right)} + e^{-\frac{C\lambda}{2+C\eta_i}} \left(1 + \frac{2}{C\eta_i}\right)^{\frac{CL}{2}-1} \left[1 - \frac{\Gamma\left(\frac{CL}{2} - 1, \frac{C^2\lambda\eta_i}{4+2C\eta_i}\right)}{\Gamma\left(\frac{CL}{2} - 1\right)}\right], \end{aligned} \quad (13)$$

where $\bar{\eta}_i$ is the average SNR due to the random variations in the channel. Thus, for the threshold λ computed using a false alarm probability in (11), an SU's operating point is given by $(P_f(\lambda), P_d(\lambda))$.

IV. EXPECTATION MAXIMIZATION BASED STATE ESTIMATION FOR DYNAMIC PRIMARY USER

The mWSED algorithm described in the previous section assumes that the actual states visited by the PU over the observation window are a priori known to the SUs. In practice, this may not be a valid assumption, and thus in this section we aim to compute an estimate of the states locally at each SU from the samples collected over the observation window.

To begin, let an SU i collect D energy samples over consecutive sensing intervals using (1), denoted by $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,D}]^T$ with T representing the transpose operation. Using the central limit theorem assumption [17], [19], it can be shown that x_{id} follows a normal distribution represented by $\mathcal{N}(x_{i,d}|\mu_h, \sigma_h^2)$ with mean μ_h and variance σ_h^2 , and with $h = 0$ when the PU is idle, and $h = 1$ when the PU is active. These means and variances in the binary hypothesis setting can be easily computed as

$$\begin{aligned}\mu_0 &= L\sigma_n^2 \\ \mu_1 &= (L + \eta_i)\sigma_n^2 \\ \sigma_0^2 &= 2L\sigma_n^4 \\ \sigma_1^2 &= 2(L + 2\eta_i)\sigma_n^4\end{aligned}\tag{14}$$

Further, if for the i -th SU the state of the PU at the sensing interval d is denoted by $s_{i,d} \in \{0, 1\}$, then for $\boldsymbol{\theta}_0 \triangleq \{\mu_0, \sigma_0^2\}$ and $\boldsymbol{\theta}_1 \triangleq \{\mu_1, \sigma_1^2\}$, the conditional probability distribution of \mathbf{x}_i can be written as

$$\begin{aligned}p(\mathbf{x}_i|\mathbf{s}_i, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1) \\ = \prod_{d=1}^D (\mathcal{N}(x_{i,d}|\mu_1, \sigma_1^2))^{\mathbb{1}(s_{i,d}=1)} (\mathcal{N}(x_{i,d}|\mu_0, \sigma_0^2))^{\mathbb{1}(s_{i,d}=0)},\end{aligned}\tag{15}$$

where $\mathbf{s}_i = [s_{i,1}, s_{i,2}, \dots, s_{i,D}]^T$ denotes the PU state vector, and $\mathbb{1}(A)$ is an indicator function which is one if A is true, and is zero otherwise [19]. Next, as discussed before and shown in Fig.

1, we assume that the state vector \mathbf{s}_i follows a two-state Markov chain model with the transition probabilities α and β . Thus, the probability distribution of \mathbf{s}_i is written as

$$\begin{aligned} p(\mathbf{s}_i|\alpha, \beta) &= p(s_{i,1}) \prod_{d=2}^D p(s_{i,d}|s_{i,d-1}) \\ &= p(s_{i,1}) \prod_{d=2}^D \left[(1-\alpha)^{\mathbb{1}(s_{i,d-1}=1)} \beta^{\mathbb{1}(s_{i,d-1}=0)} \right]^{\mathbb{1}(s_{i,d}=1)} \left[\alpha^{\mathbb{1}(s_{i,d-1}=1)} (1-\beta)^{\mathbb{1}(s_{i,d-1}=0)} \right]^{\mathbb{1}(s_{i,d}=0)}. \end{aligned} \quad (16)$$

where considering the steady-state distribution for the Markov process, we assume $s_{i,1}$ is Bernoulli distributed with mean $\frac{\beta}{\alpha+\beta}$.

Now if the model parameters of the above probability distributions are defined by $\Theta = \{\theta_0, \theta_1, \alpha, \beta\}$, an optimal scheme for estimating both Θ and \mathbf{s}_i for SU i involves solving the following optimization problem

$$\begin{aligned} (\mathbf{s}_i^*, \Theta^*) &= \arg \max_{(\mathbf{s}_i, \Theta)} p(\mathbf{s}_i, \Theta | \mathbf{x}_i) \\ &= \arg \max_{(\mathbf{s}_i, \Theta)} p(\mathbf{x}_i | \mathbf{s}_i, \theta_0, \theta_1) p(\mathbf{s}_i | \alpha, \beta), \end{aligned} \quad (17)$$

where for the sake of simplicity, we assumed a uniform prior distribution on Θ . Note that due to the large dimensionality of the search space, jointly optimizing for \mathbf{s}_i and Θ is computationally difficult. Alternatively, we can aim to sequentially optimize for \mathbf{s}_i and Θ which involves solving the following two optimization problems:

$$\begin{aligned} \hat{\Theta} &= \arg \max_{\Theta} \log p(\mathbf{x}_i | \Theta) \\ &= \arg \max_{\Theta} \log \sum_{\mathbf{s}_i} p(\mathbf{x}_i, \mathbf{s}_i | \Theta) \end{aligned} \quad (18)$$

where (18) maximizes the likelihood function of Θ . Then using $\hat{\Theta}$ we can solve,

$$\begin{aligned} \hat{\mathbf{s}}_i &= \arg \max_{\mathbf{s}_i} p(\mathbf{s}_i | \mathbf{x}_i, \hat{\Theta}) \\ &= \arg \max_{\mathbf{s}_i} p(\mathbf{x}_i | \mathbf{s}_i, \hat{\theta}_0, \hat{\theta}_1) p(\mathbf{s}_i | \hat{\alpha}, \hat{\beta}), \end{aligned} \quad (19)$$

However, note that due to the log-sum in (18), directly optimizing for the elements of Θ , e.g., using the derivative trick, does not result in the closed-form update equations, whereas using the numerical methods for optimization have the inherent complexity with the tuning of the step-size parameter [20]. Furthermore, the optimization problem in (19) is still a complex combinatorial search problem where the dimensionality of the search space increases exponentially with D . In

the following, we propose an expectation maximization based algorithm to estimate \mathbf{s}_i and Θ in a computationally efficient way using the closed-form update equations.

A. Expectation Maximization Algorithm

An expectation maximization algorithm [21]–[23] is an iterative algorithm which can be derived by first selecting a complete data model in order to compute an objective function of the model parameters. Next, given an initial estimate of the parameters, it tends to improve this estimates in each iteration by maximizing the objective function which in turn maximizes the likelihood function [22]. The EM algorithm has been developed for a variety of estimation problems in recent years [22], [24], [25], and in this subsection, we develop it to facilitate joint PU states and the model parameters estimation in order to enable distributed cooperative spectrum sensing by the SUs.

To begin, let the complete data model for the i -th SU be denoted by $[\mathbf{x}_i^T, \mathbf{s}_i^T]^T$, and suppose $\Theta^{(l-1)}$ is the $(l-1)$ -st estimate of the model parameters, then in the l -th iteration it computes an expectation step (E-step) and a maximization step (M-step). In the E-step, it computes an expectation of the complete data log-likelihood function as follows

$$\mathcal{Q}(\Theta; \Theta^{(l-1)}) = \mathbb{E}_{p(\mathbf{s}_i|\mathbf{x}_i, \Theta^{(l-1)})} [\log p(\mathbf{x}_i, \mathbf{s}_i|\Theta)], \quad (20)$$

where we note that the expectation is with respect to the posterior distribution on \mathbf{s}_i given \mathbf{x}_i and an old estimate $\Theta^{(l-1)}$. In the M-step, it maximizes the objective function in (20) with respect to Θ by solving

$$\Theta^{(l)} = \arg \max_{\Theta} \mathcal{Q}(\Theta; \Theta^{(l-1)}), \quad (21)$$

in which $\Theta^{(l)}$ represents the new estimate of Θ in the l -th iteration. The above E-step and M-step are repeated iteratively by replacing the old estimate with the new one until convergence is achieved.

Now using the distributions in (15) and (16), and the following notation for the expectation operations, i.e., $\gamma(s_{i,d} = h) \triangleq \mathbb{E}[\mathbb{1}_{(s_{i,d}=h)}]$ and $\xi(s_{i,d} = h, s_{i,d-1} = g) \triangleq \mathbb{E}[\mathbb{1}_{(s_{i,d}=h)}\mathbb{1}_{(s_{i,d-1}=g)}]$, for $g, h \in \{0, 1\}$, it can be easily derived that the objective function in (20) can be written as shown in (22). Further, note that $\gamma(s_{i,d} = h) = p(s_{i,d} = h|\mathbf{x}_i, \Theta^{(l-1)})$ and $\xi(s_{i,d} = h, s_{i,d-1} = g) = p(s_{i,d} = h, s_{i,d-1} = g|\mathbf{x}_i, \Theta^{(l-1)})$ for $g, h \in \{0, 1\}$, where these probability distributions are derived in the Appendix.

$$\begin{aligned}
\mathcal{Q}(\Theta; \Theta^{(l-1)}) &= \sum_{d=1}^D [\gamma(s_{i,d} = 1) \log \mathcal{N}(x_{i,d} | \mu_1, \sigma_1^2) + \gamma(s_{i,d} = 0) \log \mathcal{N}(x_{i,d} | \mu_0, \sigma_0^2)] + \\
&\sum_{d=2}^D [\xi(s_{i,d} = 1, s_{i,d-1} = 1) \log(1 - \alpha) + \xi(s_{i,d} = 1, s_{i,d-1} = 0) \log \beta + \\
&\xi(s_{i,d} = 0, s_{i,d-1} = 1) \log \alpha + \xi(s_{i,d} = 0, s_{i,d-1} = 0) \log(1 - \beta)] + \text{const}, \tag{22}
\end{aligned}$$

In order to compute the M-step in (21), we use the sequential optimization approach [24], [25] for simplicity, i.e., we maximize $\mathcal{Q}(\Theta; \Theta^{(l-1)})$ with respect to each parameter individually by keeping the others fixed to their current estimate. To that end, we use the derivative trick, and thus to maximize $\mathcal{Q}(\Theta; \Theta^{(l-1)})$ with respect to μ_h for $h \in \{0, 1\}$, we compute its derivative and set it equal to zero as follows.

$$\begin{aligned}
\frac{\partial \mathcal{Q}(\Theta; \Theta^{(l-1)})}{\partial \mu_h} &= 0 \\
\sum_{d=1}^D \left[\gamma(s_{i,d} = h) \frac{(x_{i,d} - \mu_h)}{\sigma_h^2} \right] &= 0, \tag{23}
\end{aligned}$$

solving it gives us a new estimate of μ_h , in the l -th iteration of EM, which is written as

$$\mu_h^{(l)} = \frac{\sum_{d=1}^D \gamma(s_{i,d} = h) x_{i,d}}{\sum_{d=1}^D \gamma(s_{i,d} = h)}, \tag{24}$$

for $h = 0, 1$. Now to maximize $\mathcal{Q}(\Theta; \Theta^{(l-1)})$ with respect to σ_h^2 , we solve

$$\begin{aligned}
\frac{\partial \mathcal{Q}(\Theta; \Theta^{(l-1)})}{\partial \sigma_h^2} &= 0 \\
\sum_{d=1}^D \left[\gamma(s_{i,d} = h) \left(\frac{1}{2\sigma_h^2} - \frac{(x_{i,d} - \mu_h)^2}{2\sigma_h^4} \right) \right] &= 0, \tag{25}
\end{aligned}$$

from which we get the update equation for σ_h^2 as

$$(\sigma_h^2)^{(l)} = \frac{\sum_{d=1}^D \gamma(s_{i,d} = h) (x_{i,d} - \mu_h^{(l)})^2}{\sum_{d=1}^D \gamma(s_{i,d} = h)}, \tag{26}$$

where $h = 0, 1$. Similarly, using the same approach, it can be easily shown that the update equations for the Markov chain transition probabilities α and β are given by

$$\alpha^{(l)} = \frac{\sum_{d=2}^D \xi(s_{i,d} = 0, s_{i,d-1} = 1)}{\sum_{d=2}^D [\xi(s_{i,d} = 0, s_{i,d-1} = 1) + \xi(s_{i,d} = 1, s_{i,d-1} = 1)]}, \quad (27)$$

$$\beta^{(l)} = \frac{\sum_{d=2}^D \xi(s_{i,d} = 1, s_{i,d-1} = 0)}{\sum_{d=2}^D [\xi(s_{i,d} = 1, s_{i,d-1} = 0) + \xi(s_{i,d} = 0, s_{i,d-1} = 0)]}. \quad (28)$$

Thus, all the model parameters are updated iteratively in EM using the closed-form update Eqns. (24), (26), (27), and (28) until convergence is achieved.

Finally, upon the convergence of EM, the state vector $\mathbf{s}_i = [s_{i,1}, s_{i,2}, \dots, s_{i,D}]^T$ for the i -th user can be estimated, in a computationally efficient way, by using the Viterbi algorithm [18]. Thus, at SU i , let the EM estimate of the model parameters is denoted by $\hat{\Theta}$, then the Viterbi algorithm uses it to recursively solve the following optimization problem

$$\omega_{i,d}(s_{i,d}) = \max_{s_{i,d-1}} \left[p(x_{i,d}|s_{i,d}, \hat{\Theta}) p(s_{i,d}|s_{i,d-1}, \hat{\Theta}) \omega_{i,d-1}(s_{i,d-1}) \right], \quad (29)$$

for $d = 2, 3, \dots, D$ with the initialization $\omega_{i,1}(s_{i,1}) = p(x_{i,1}|s_{i,1}, \hat{\Theta})p(s_{i,1}, \hat{\Theta})$. The distributions $p(x_{i,d}|s_{i,d}, \hat{\Theta})$ and $p(s_{i,d}|s_{i,d-1}, \hat{\Theta})$ are given in (15) and (16), respectively, and note that they are computed in (29) using only the required parameters estimate from the set $\hat{\Theta}$. Hence, by keeping track of the maximizing sequence at each time instant in (29) and by finding $\max_{s_{i,D}} \omega_{i,D}(s_{i,D})$ at time instant D , we can back trace the most probable sequence to get $\hat{\mathbf{s}}_i$.

Note that the combination of EM and Viterbi algorithm is named here as the EM-Viterbi algorithm. However, once the state vector of the PU is estimated then we can use it in the mWSED algorithm proposed in Section III-C to combine only the highly correlated energy samples in its test statistic, and the resulting algorithm is referred to here as the EM-mWSED algorithm. Both EM-Viterbi and EM-mWSED are summarized for SU i in Algorithm 1.

The computational complexity of EM-mWSED is dominated by the use of the distributed consensus algorithm of (6) in Step 1. This step has the complexity of $\mathcal{O}(|N_i|)$ per its iteration, where $|N_i|$ is the cardinality of the set of neighboring users of SU i . Furthermore, the forward and backward passes on the observation window in Steps 2 and 3, to compute the distributions in (32) and (33), respectively, as well as the Viterbi algorithm in Step 6 and (29) also dominate the computational complexity. These steps have the complexity of $\mathcal{O}(2D)$ where D is the length of the observation window. Thus, the computational complexity of EM-mWSED is $\mathcal{O}(I_c|N_i| + 2DI_e)$

Algorithm 1: States Estimation Based PU Detection for SU i

Input: $l = 0$, \mathbf{x}_i and \mathbf{x}_j for $j \in N_i$, $\Theta^{(0)}$.

- 1) Use the distributed consensus algorithm of (6) to reach consensus on \mathbf{x}_i with the other users in the network.

while *convergence criterion is not met* **do**

$l = l + 1$

- 2) Use the forward recursion in (32) to compute $\nu_d(s_{i,d} = h)$ for all $d = 1, 2, \dots, D$ and $h = 0, 1$.
- 3) Use the backward recursion in (33) to compute $\pi_d(s_{i,d} = h)$ for all $d = D, D - 1, \dots, 1$ and $h = 0, 1$.
- 4) Compute $\gamma(s_{i,d} = h)$ from (34) for all $d = 1, 2, \dots, D$ and $h = 0, 1$, and compute $\xi(s_{i,d} = h, s_{i,d-1} = g)$ from (35) for all $d = 2, \dots, D$ and $h = 0, 1, g = 0, 1$.
- 5) Update the model parameters $\Theta^{(l)}$ using (24), (26), (27), and (28).

end

- 6) Use the Viterbi algorithm in (29) to estimate the PU state vector $\hat{\mathbf{s}}_i$ for SU i .
- 7) Compute the test statistics for mWSED using (9) and compare it against a threshold to make PU detection.

Output: $\hat{\mathbf{s}}_i, \hat{\Theta}, T_i$

where I_c is the number of consensus iterations whereas I_e denotes the number of EM iterations till convergence. The complexity of the energy detector based DCSS is $\mathcal{O}(I_c|N_i|)$ whereas that of the WSED based DCSS is $\mathcal{O}(I_c|N_i| + D)$. Thus, the performance improvement of EM-mWSED, as demonstrated in the next section, is at the cost of a slight increase in the computational complexity per a single iteration of the EM algorithm.

V. SIMULATION RESULTS

In this section, we demonstrate the performance of our expectation maximization and Viterbi algorithms based PU states estimation scheme, referred to here as EM-Viterbi, as well as that of our EM-mWSED algorithm. For comparison purposes, we compare the performance of the proposed algorithm to the conventional energy detector (ED) and the weighted sequential energy detector (WSED) of [7], [8] under different scenarios, when ED and WSED are used with the DCSS scheme and with the consensus happening on the present and past observations as proposed herein. As suggested in [7], [8] for WSED, we use a total of 3 past energy samples in its test statistics for a highly dynamic PU, whereas ED uses only the present energy sample in its test statistics. Further, we consider a network of N secondary users randomly generated with a

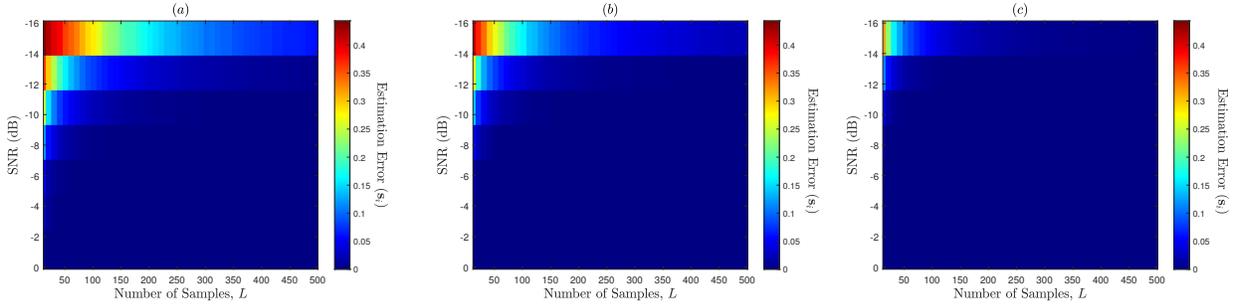


Fig. 4. Primary user states estimation error of EM for an SU as a function of SNR (dB) and the number of samples per energy statistics L , for (a) $N = 10$ SUs, (b) $N = 20$ SUs, and (c) $N = 60$ SUs, and when the network connectivity is $c = 0.2$ and the PU follows the two-state Markov model with $\alpha = \beta = 0.1$.

connectivity c and the weighting matrix as defined in (6). The average number of connections per SU in the network is given by $R = c(N - 1)$. The primary user follows a two-state Markov chain model to switch between the active and idle states with the transition probabilities α and β . The SUs collect L samples individually to compute the energy statistic, and combine $D = 150$ present and past observations over the consecutive sensing intervals after consensus to define an observation window as in Algorithm 1. For the initialization of the EM algorithm, we determine the initial estimate of the means and variances by using the K-means clustering algorithm [22] with $K = 2$, whereas the initial estimate of the transition probabilities can be computed by performing a coarse grid search over the likelihood function in (18) in the $(0, 1)$ interval with grid resolution of 0.1.

In Fig. 4, we demonstrate the performance of the joint EM and Viterbi (EM-Viterbi) algorithm at estimating the state vector of the PU as a function of SNR (dB) and the number of samples per energy statistics L . The states estimation error for SU i is defined here as Estimation Error (s_i) = $\frac{1}{D} \sum_{d=1}^D \mathbb{E} [\mathbf{1}(\hat{s}_{i,d} \neq s_{i,d})]$ where the expectation is computed over several Monte Carlo trials. Further, we assume that the consensus is reached on the energy samples in Step 1 in Algorithm 1 prior to estimation, thus the error plots in this figure are observed at all the SUs in the network. We consider here that the secondary users network has $N = 10, 20,$ and 60 users with connectivity $c = 0.2$. The PU displays a highly dynamic nature with transition probabilities $\alpha = \beta = 0.1$. It is observed that, in general, the estimation error is higher at lower SNR and L values for all the considered cases in Fig. 4. This is because the distribution of the energy samples under the two hypotheses highly overlap at those values making it harder to separate the samples into two clusters. Particularly, we observe that for a fewer number of SUs ($N = 10$) in

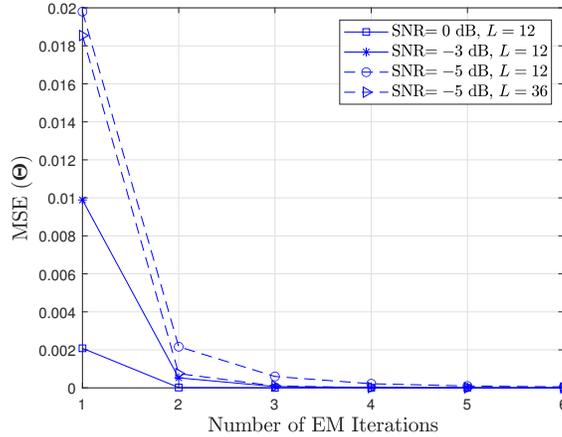


Fig. 5. Mean-squared error (MSE) of estimating the model parameters Θ vs. EM iterations for $N = 10$ SUs and connectivity $c = 0.2$, when SNR = -5 dB, -3 dB, or 0 dB and when the number of samples per energy statistics $L = 12$ or 36 . The PU states transition probabilities are $\alpha = \beta = 0.1$.

the network, the estimation error is higher at the lower SNRs and the lower L values, but when the number of SUs in the network increases from $N = 10$ to 20 and then to 60 , the estimation error decreases significantly even for the lower SNR and L values. This is because the SNR upon consensus in Step 1 of Algorithm 1 is directly proportional to \sqrt{R} because each SU exploits R independent observations of the PU channel's energy statistics. Thus, with the increase in either N or c , the SNR upon consensus improves due to the increase in R which in turns results in decreasing the estimation error. This explains our motivation behind using DCSS scheme prior to the estimation process in Algorithm 1 that improves the performance of EM-Viterbi at the lower SNR and L values for larger networks.

In Fig. 5, we illustrate the convergence performance of the EM algorithm in estimating the model parameters Θ when $N = 10$ SUs are considered in the network with network connectivity $c = 0.2$. The mean-squared error (MSE) of Θ is defined as $\text{MSE}(\Theta) = \mathbb{E} \left[\|\Theta - \hat{\Theta}\|_2 \right]$. It is observed that for $L = 12$ samples per energy statistics, as the SNR increase from -5 dB, to -3 dB, and then to 0 dB, the EM algorithm converges faster in fewer iterations. Similar observation is also made when for a lower SNR value of -5 dB, we increase L from 12 to 36 . This is due to the fact that the initial estimates for EM are improved at the larger SNR and L values which results in its faster convergence response.

Since our joint EM and Viterbi algorithm, referred to here as EM-Viterbi, also outputs an

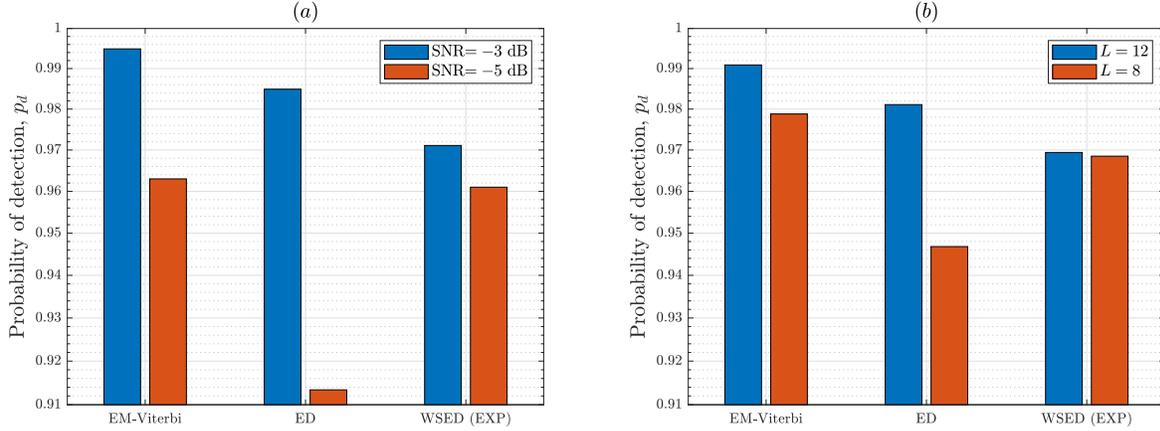


Fig. 6. Illustrating detection performance of the joint EM and Viterbi (EM-Viterbi) algorithm for different SNR (dB) values or the number of samples per energy statistics L , and comparing it to the conventional ED and the weighted sequential energy detector with exponential weighting (WSED (EXP)) for (a) $N = 10$ SUs, connectivity $c = 0.2$, and the number of energy samples $L = 12$, and (b) $N = 20$ SUs, connectivity $c = 0.5$, and SNR = -5 dB. The PU follows the states transition probabilities $\alpha = \beta = 0.1$.

estimate of the present state ($s_{i,D}$) at the i -th SU by going back and forth on the observation window, therefore in Fig. 6, we show its detection performance for different SNR and L values and compare it with the ED and the WSED algorithms. For WSED, we use the exponential weighting on the aggregated energy samples, named here as WSED (EXP), as suggested in [8] and discussed earlier in Section III-B. Two network configurations are considered for demonstration purposes. In configuration (a), we consider $N = 10$ SUs in the network with connectivity $c = 0.2$ and when $L = 12$ samples per energy statistics are used. For this case, either an SNR = -3 dB or -5 dB is assumed and the probabilities of false alarm of 0.8% and 4.4%, respectively, were recorded for EM-Viterbi. In contrast, in configuration (b), it is assumed that we have $N = 20$ SUs in the network with connectivity $c = 0.5$ and SNR = -5 dB. The number of samples L is considered to be either 8 or 12, and similarly the probabilities of false alarm of 2.66% and 1.08%, respectively, were observed for EM-Viterbi. These false alarm probabilities in each case were selected also for ED and WSED (EXP) to define a threshold and determine their probability of detection for comparison purposes. In general, from Fig. 6, it is observed that EM-Viterbi outperforms EM and WSED (EXP) in improving the detection probability of PU, and its performance improves with the increase in SNR and the number of samples L as expected. Specifically, in Fig. 6 (a), when SNR = -5 dB and $L = 12$ the detection probability

of EM-Viterbi is 96.30% and that of WSED (EXP) is 96.14%, however, when the network size increase from $N = 10$ to 20 and connectivity increases from $c = 0.2$ to 0.5 in going from Fig. 6 (a) to Fig. 6 (b), there is an improvement in SNR upon consensus by \sqrt{R} and thus the detection probability of EM-Viterbi reaches 99.09% and that of WSED (EXP) is 96.94%. Further, Fig. 6 (b) also illustrates that the detection performance of EM-Viterbi improves by increasing the number of samples L from 8 to 12 due to further decrease in the estimation error.

While EM-Viterbi outputs a single operating point for SUs in terms of detection probability and false alarm probability, in contrast, by using the estimated state vector in mWSED, the EM-mWSED algorithm can provide a wide range of operating points. As such, in Figs. 7 and 8, we show the receiver operating characteristic (ROC) curves of the proposed EM-mWSED algorithm and compare it with those of the ED, WSED (EXP), and WSED with equal weighting of the present and past energy samples, viz named here as WSED (EQ). Accordingly, in these figures, we demonstrate the performance of EM-mWSED with either equal weighting (EM-mWSED (EQ)) or with the exponential weighting (EM-mWSED (EXP)) of the aggregated energy samples. Notably, we observed that exponential weighting results in better detection performance than equal weighting at lower SNR or L values due to the rise in the estimation error.

In Fig. 7, we consider $N = 20$ SUs in the network with connectivity $c = 0.2$. The PU states transitioning probabilities are selected as $\alpha = \beta = 0.1$. The SNR is assumed to be either -3 dB or -5 dB, whereas the number of samples per energy statistic L is assumed to be either 8 or 12. As expected, it is observed that the detection performance of EM-mWSED improves with the increase in either SNR or L values, due to decrease in the states estimation error. Further, EM-mWSED outperforms both ED and WSED at increasing the detection probability and reducing the false alarm probability, and thereby provides a wide range of operating points for SU.

Fig. 8 compares the ROC curves of EM-mWSED with that of ED and WSED with both exponential and equal weighting of the aggregated energy samples. The PU is either considered to be slowly time-varying with $\alpha = \beta = 0.05$ or highly dynamic with $\alpha = \beta = 0.1$ as considered earlier. There are $N = 10$ SUs in the network with connectivity $c = 0.2$, and the SNR is considered to be -3 dB with the number of samples per energy statistics as either $L = 12$ or 36. Firstly, by comparing Figs. 7 and 8 for $L = 12$, SNR = -3 dB, and $\alpha = \beta = 0.1$, we observe a decay in the detection performance of EM-mWSED (EXP) due to decrease in the value of R in Fig. 8, which reduces the SNR upon consensus as discussed above. Secondly, it is observed in Fig. 8 that for both slowly and highly dynamic natures of the PU, EM-WSED

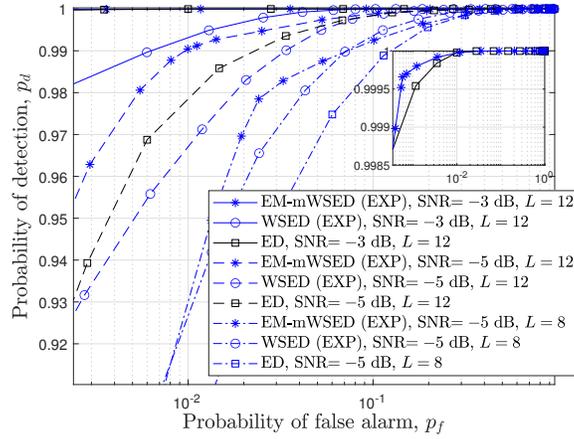


Fig. 7. Receiver operating characteristic curves of EM-mWSED (EXP), conventional ED, WSED (EQ), and WSED (EXP), for different SNR (dB) values or the number of samples per energy statistics L , when $N = 20$ SUs, connectivity $c = 0.2$, and the PU states transition probabilities $\alpha = \beta = 0.1$ are used.

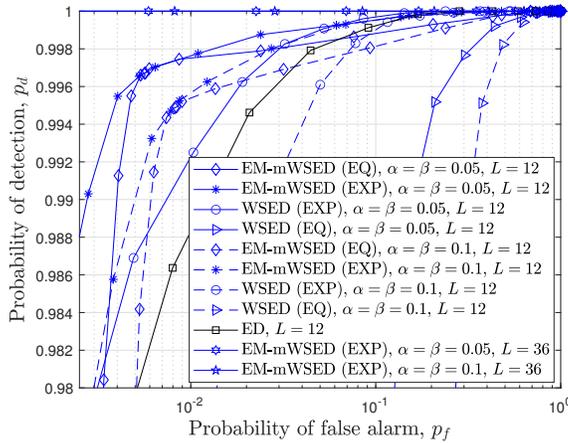


Fig. 8. Receiver operating characteristic curves of EM-mWSED (EXP), EM-mWSED (EQ), conventional ED, WSED (EQ), and WSED (EXP), for slowly time-varying PU with $\alpha = \beta = 0.05$ and a highly dynamic PU with $\alpha = \beta = 0.1$. The number of SUs $N = 10$, connectivity $c = 0.2$, SNR = -3 dB, and the number of samples per energy statistics $L = 12$ or 36.

(EXP) performs better than EM-mWSED (EQ) since it avoids aggregating the outdated energy samples at the lower SNR or L values. Further, we also observe that, for both kinds of PUs, our EM-mWSED algorithm performs better than the other detectors as expected. However, when $L = 12$, its detection performance appears to be dependent on the time-varying nature of the PU, and it is seen to be better in case of slowly time-varying PU than a highly dynamic PU. This is because at the lower SNR or L values, EM-Viterbi can easily characterize the energy

samples, corresponding to the two states of PU, when the PU is slowly time-varying than when it is highly dynamic, and thus results in a lower estimation error in the former case. However, when L increases to 36, then the estimation error of EM-Viterbi decreases for a highly dynamic PU as well, which in turn results in the similar performance of EM-mWSED (EXP) for both kinds of PUs, as shown in this figure.

Finally, while the focus herein is on investigating the detection vs. false alarm probabilities, it is worth noting that, on the one hand, where the throughput performance of an SU under H_1 is proportional to $(1 - p_d)$, on the other hand, the throughput under H_0 is proportional to $(1 - p_f)$ [26]. Thus, the higher detection probability of EM-Viterbi and EM-mWSED as compared with that of ED and WSED implies a higher throughput of SUs with reduced interference to the primary user, whereas its capability of simultaneously decreasing the false alarm probability with the increase in the average connections per SUs in the network, SNR, or L , implies a higher throughput during the idle state of the PU.

VI. CONCLUSION

We considered the problem of DCSS for a dynamic PU when the present and past energy samples are aggregated in a test statistic to enable improved PU detection capabilities. To this end, a modified weighted sequential energy detector is proposed which utilizes the PU states information over an observation window to combine only the highly correlated energy samples in its test statistics. In practice, the states information is unknown, and thus we developed an EM-Viterbi algorithm to iteratively estimate them using the energy samples collected over the window. The estimated states are then used in mWSED to compute its test statistics, and the resulting algorithm is named here as the EM-mWSED algorithm. Simulation results are included to demonstrate the estimation performance of EM-Viterbi and compare the detection performance of both EM-Viterbi and EM-mWSED with that of the conventional energy detector and the WSED algorithm. The results demonstrate that our proposed algorithms perform better than both ED and WSED, and their performances improve by either increasing the average number of connections per SU in the network, or by increasing the SNR or the number of samples per energy statistics, for both slowly varying and highly dynamic PU.

VII. APPENDIX

In this section, we present the derivation of the probabilities $\gamma(s_{i,d} = h) = p(s_{i,d} = h | \mathbf{x}_i, \Theta^{(l-1)})$ and $\xi(s_{i,d} = h, s_{i,d-1} = g) = p(s_{i,d} = h, s_{i,d-1} = g | \mathbf{x}_i, \Theta^{(l-1)})$ for $g, h \in \{0, 1\}$ and the l -th iteration of EM. To begin, the posterior distribution of $s_{i,d}$ given \mathbf{x}_i and $\Theta^{(l-1)}$ can be written as

$$\begin{aligned}
 & p(s_{i,d} | \mathbf{x}_i, \Theta^{(l-1)}) \\
 & \propto p(s_{i,d}, \mathbf{x}_i | \Theta^{(l-1)}) \\
 & = p(s_{i,d}, \mathbf{x}_{i,1:d}, \mathbf{x}_{i,d+1:D} | \Theta^{(l-1)}) \\
 & = p(\mathbf{x}_{i,d+1:D} | s_{i,d}, \Theta^{(l-1)}) p(\mathbf{x}_{i,1:d}, s_{i,d} | \Theta^{(l-1)}) \\
 & \triangleq \pi_d(s_{i,d}) \nu_d(s_{i,d}), \tag{30}
 \end{aligned}$$

where the distributions $\pi_d(s_{i,d})$ and $\nu_d(s_{i,d})$ are computed later herein. The notation $\mathbf{x}_{i,m:n} \triangleq [x_{i,m}, x_{i,m+1}, \dots, x_{i,n}]^T$ which is a shorthand to represent the elements in \mathbf{x}_i from index m to n where $m, n \in \{1, 2, \dots, D\}$. Similarly, we can write the joint distribution of $s_{i,d}$ and $s_{i,d-1}$ as

$$\begin{aligned}
 & p(s_{i,d}, s_{i,d-1} | \mathbf{x}_i, \Theta^{(l-1)}) \\
 & \propto p(\mathbf{x}_{i,1:d-1}, s_{i,d-1}, s_{i,d}, \mathbf{x}_{i,d:D}, \Theta^{(l-1)}) \\
 & = p(\mathbf{x}_{i,1:d-1}, s_{i,d-1} | \Theta^{(l-1)}) p(\mathbf{x}_{i,d:D}, s_{i,d} | s_{i,d-1}, \Theta^{(l-1)}) \\
 & = \nu_{d-1}(s_{i,d-1}) p(x_{i,d}, \mathbf{x}_{i,d+1:D}, s_{i,d} | s_{i,d-1}, \Theta^{(l-1)}) \\
 & = \nu_{d-1}(s_{i,d-1}) \pi_d(s_{i,d}) p(x_{i,d} | s_{i,d}, \Theta^{(l-1)}) p(s_{i,d} | s_{i,d-1}, \Theta^{(l-1)}), \tag{31}
 \end{aligned}$$

where the conditional distribution of $x_{i,d}$ and the conditional prior distribution of $s_{i,d}$ that are used above are both defined in (15) and (16). Next we follow the forward-backward recursion approach in [23] to compute the distributions $\nu_d(s_{i,d})$ and $\pi_d(s_{i,d})$. First, to compute $\nu_d(s_{i,d})$, we write

$$\begin{aligned}
 \nu_d(s_{i,d}) & = p(s_{i,d}, \mathbf{x}_{i,1:d} | \Theta^{(l-1)}) \\
 & = \sum_{s_{i,d-1}} p(s_{i,d}, s_{i,d-1}, \mathbf{x}_{i,1:d-1}, x_{i,d} | \Theta^{(l-1)}) \\
 & = \sum_{s_{i,d-1}} p(x_{i,d} | s_{i,d}, \Theta^{(l-1)}) p(s_{i,d} | s_{i,d-1}, \Theta^{(l-1)}) p(s_{i,d-1}, \mathbf{x}_{i,d-1} | \Theta^{(l-1)}) \\
 & = \sum_{s_{i,d-1}} c(s_{i,d}, s_{i,d-1}) \nu_{d-1}(s_{i,d-1}), \tag{32}
 \end{aligned}$$

where we have defined $c(s_{i,d}, s_{i,d-1}) = p(x_{i,d}|s_{i,d}, \Theta^{(l-1)}) p(s_{i,d}|s_{i,d-1}, \Theta^{(l-1)})$, and in (32) the summation is over $s_{i,d-1} \in \{0, 1\}$. The forward recursion in (32) occurs in the l -iteration of EM for all $d = 2, 3, \dots, D$ with the initialization $\nu_1(s_{i,1}) = p(s_{i,1}|\Theta^{(l-1)})p(x_{i,1}|s_{i,1}, \Theta^{(l-1)})$ which is defined in (15) and (16). Next we write the backward recursion equation to compute $\pi_d(s_{i,d})$ as follows

$$\begin{aligned}
\pi_d(s_{i,d}) &= p(\mathbf{x}_{i,d+1:D}|s_{i,d}, \Theta^{(l)}) \\
&= \sum_{s_{i,d+1}} p(\mathbf{x}_{i,d+1:D}, s_{i,d+1}|s_{i,d}, \Theta^{(l)}) \\
&= \sum_{s_{i,d+1}} p(\mathbf{x}_{i,d+2:D}|s_{i,d+1}, \Theta^{(l)}) p(x_{i,d+1}|s_{i,d+1}, \Theta^{(l)}) p(s_{i,d+1}|s_{i,d}, \Theta^{(l)}) \\
&= \sum_{s_{i,d+1}} \pi_{d+1}(s_{i,d+1})c(s_{i,d+1}, s_{i,d}), \tag{33}
\end{aligned}$$

in which the summation runs over $s_{i,d+1} \in \{0, 1\}$ for all $d = D - 1, D - 2, \dots, 1$ with the initialization $\pi_D(s_{i,D}) = 1$. Finally, the probabilities $\gamma(s_{i,d} = h) = p(s_{i,d} = h|\mathbf{x}_i, \Theta^{(l-1)})$ and $\xi(s_{i,d} = h, s_{i,d-1} = g) = p(s_{i,d} = h, s_{i,d-1} = g|\mathbf{x}_i, \Theta^{(l-1)})$ for $g, h \in \{0, 1\}$ can be computed as

$$\begin{aligned}
\gamma(s_{i,d} = h) &= p(s_{i,d} = h|\mathbf{x}_i, \Theta^{(l-1)}) \\
&= \frac{\nu_d(s_{i,d} = h)\pi_d(s_{i,d} = h)}{\sum_{s_{i,d}} \nu_d(s_{i,d})\pi_d(s_{i,d})}, \tag{34}
\end{aligned}$$

and,

$$\begin{aligned}
\xi(s_{i,d} = h, s_{i,d-1} = g) &= p(s_{i,d} = h, s_{i,d-1} = g|\mathbf{x}_i, \Theta^{(l-1)}) \\
&= \frac{\nu_{d-1}(s_{i,d-1} = g)\pi_d(s_{i,d} = h)c(s_{i,d} = h, s_{i,d-1} = g)}{\sum_{s_{i,d}} \sum_{s_{i,d-1}} \nu_{d-1}(s_{i,d-1})\pi_d(s_{i,d})c(s_{i,d}, s_{i,d-1})}, \tag{35}
\end{aligned}$$

REFERENCES

- [1] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, 2005.
- [2] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 116–130, 2009.
- [3] E. Axell, G. Leus, E. G. Larsson, and H. V. Poor, "Spectrum Sensing for Cognitive Radio : State-of-the-Art and Recent Advances," *IEEE Signal Processing Magazine*, vol. 29, no. 3, pp. 101–116, 2012.
- [4] A. Ghasemi and E. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments," in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.*, 2005, pp. 131–136.

- [5] K. Cichoń, A. Kliks, and H. Bogucka, "Energy-Efficient Cooperative Spectrum Sensing: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1861–1886, 2016.
- [6] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical Communication*, vol. 4, no. 1, pp. 40–62, 2011.
- [7] W. Prawatmuang and D. K. C. So, "Adaptive sequential cooperative spectrum sensing technique in time varying channel," in *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*, 2012, pp. 1546–1551.
- [8] W. Prawatmuang, D. K. C. So, and E. Alsusa, "Sequential Cooperative Spectrum Sensing Technique in Time Varying Channel," *IEEE Transactions on Wireless Communications*, vol. 13, no. 6, pp. 3394–3405, 2014.
- [9] Z. Li, F. R. Yu, and M. Huang, "A Distributed Consensus-Based Cooperative Spectrum-Sensing Scheme in Cognitive Radios," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 1, pp. 383–393, 2010.
- [10] A. Mustafa, M. N. U. Islam, and S. Ahmed, "Dynamic Spectrum Sensing Under Crash and Byzantine Failure Environments for Distributed Convergence in Cognitive Radio Networks," *IEEE Access*, vol. 9, pp. 23 153–23 167, 2021.
- [11] A. Gharib, W. Ejaz, and M. Ibnkahla, "Enhanced Multiband Multiuser Cooperative Spectrum Sensing for Distributed CRNs," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 256–270, 2020.
- [12] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [13] S. Boyd, P. Diaconis, and L. Xiao, "Fastest Mixing Markov Chain on a Graph," *SIAM Review*, vol. 46, no. 4, pp. 667–689, 2004.
- [14] H. Wang, W. Xu, and J. Lu, "Privacy-Preserving Push-sum Average Consensus Algorithm over Directed Graph Via State Decomposition," in *2021 3rd International Conference on Industrial Artificial Intelligence (IAI)*, 2021, pp. 1–6.
- [15] X. Qiao, Y. Wu, and D. Meng, "Privacy-Preserving Average Consensus for Multi-agent Systems with Directed Topologies," in *2021 8th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS)*, 2021, pp. 7–12.
- [16] Q. Yan, M. Li, T. Jiang, W. Lou, and Y. T. Hou, "Vulnerability and protection for distributed consensus-based spectrum sensing in cognitive radio networks," in *2012 Proceedings IEEE INFOCOM*, 2012, pp. 900–908.
- [17] B. Kailkhura, S. Brahma, and P. K. Varshney, "Data Falsification Attacks on Consensus-Based Detection Systems," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 1, pp. 145–158, 2017.
- [18] G. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.
- [19] J. Gubner, *Probability and Random Processes for Electrical and Computer Engineers*. Cambridge University Press, 2006.
- [20] S. Boyd and L. Vandenberghe, *Convex Optimization*, ser. Berichte über verteilte messsysteme. Cambridge University Press, 2004, no. pt. 1.
- [21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.
- [22] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
- [23] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.
- [24] M. Rashid and M. Naraghi-Pour, "Multitarget Joint Delay and Doppler-Shift Estimation in Bistatic Passive Radar," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 3, pp. 1795–1806, 2020.
- [25] M. Rashid and J. A. Nanzer, "Online Expectation-Maximization Based Frequency and Phase Consensus in Distributed Phased Arrays," *IEEE Transactions on Communications*, pp. 1–1, 2023.

- [26] R. A. Rashid, Y. S. Baguda, N. Fisal, M. Adib Sarijari, S. K. S. Yusof, S. H. S. Ariffin, N. M. A. Latiff, and A. Mohd, "Optimizing distributed cooperative sensing performance using swarm intelligence," in *Informatics Engineering and Information Science*. Springer, Berlin, Heidelberg, 2011, pp. 407–420.