Reducing Computational Complexity of Eigenvalue Based Spectrum Sensing for Cognitive Radio

Sener Dikmese[1], Jiunn Lin Wong[2], Ahmet Gokceoglu[1], Elena Guzzon[3], Mikko Valkama[1] and Markku Renfors[1]

[1]Depart. of Elect. and Com. Eng.
Tampere University of Technology
Tampere, FINLAND

[2]Depart. of Elect. and Com. Eng.
Jacobs University
Bremen, GERMANY

[3]Depart. of Sig. Proc. Tel. Eco.
University of Roma TRE
Rome, ITALY

{sener.dikmese, ahmet.gokceoglu, mikko.e.valkama, markku.renfors}@tut.fi,
ji.wong@jacobs-university.de,elena.guzzon@uniroma3.it

***Abstract*— Spectrum sensing of primary users under very low signal-to-noise ratio (SNR) and noise uncertainty is crucial for cognitive radio (CR) systems. To overcome the drawbacks of weak signal and noise uncertainty, eigenvalue-based spectrum sensing methods have been proposed for advanced CRs. However, one pressing disadvantage of eigenvalue-based spectrum sensing algorithms is their high computational complexity, which is due to the calculation of the covariance matrix and its eigenvalues. In this study, power, inverse power and fast Cholesky methods for eigenvalue computation are investigated as potential methods for reducing the computational complexity.**

Keywords-component; Energy detector based spectrum sensing, eigenvalue based spectrum sensing, AWGN, frequency selective channel and noise uncertainty

I. INTRODUCTION

Cognitive radio is emerging as an attractive solution to the problem of spectral congestion in wireless communications, as it allows opportunistic use of unoccupied spectrum by secondary users [1], [2], [3]. A key challenge of cognitive radio technology lies in reliable and efficient spectrum sensing techniques to detect the presence of primary users, so that secondary users can adjust their operating parameters to exploit the unused spectrum. Various methods such as energy detection, matched filter detection, autocorrelation [4], and cyclostationary feature detection [3] have emerged, but each has its own shortcomings in operational requirements or performance under noise uncertainty [1], [5].

Recently, considerable attention has turned to eigenvalue-based spectrum sensing, which requires only the covariance matrix of the received signal to be estimated, and no knowledge of the signal, noise power and channel are needed. Therefore, these methods have the potential of overcoming the problem of noise uncertainty [5]. Results from simulations have confirmed the robust performance of this method, especially under low signal-to-noise ratio (SNR) [1], [6]. However, implementing eigenvalue-based spectrum sensing requires high computational complexity, primarily from the computation of the covariance matrix and finding its eigenvalues [5]. Improvement on this complexity issue is critical to many real-time applications, such as TV white space utilization by cognitive radios [3]. Hence, the power iteration eigenvalue algorithm with deflation has been proposed in [7] to approximate all eigenvalues of the covariance matrix with reduced computational complexity. However, since only the largest and smallest eigenvalues are required for eigenvalue-based spectrum sensing [2], methods and algorithms with lower complexity should be explored. The purpose of this paper is to propose a combination of iterative eigenvalue algorithms: the power method and inverse power method to estimate the maximum and minimum eigenvalues respectively, thus reducing the computational complexity of the implementation.

In Section II, the main ideas of eigenvalue-based spectrum sensing will be introduced. In Section III, we outline the power and inverse power methods and their computational complexities, and improve the inverse method by using the Schur algorithm for fast Cholesky factorization of the covariance matrix. Then, numerical simulations will be shown in Section IV. Finally, our conclusions are presented in Section V.

II. TRADITIONAL EIGENVALUE-BASED SPECTRUM SENSING

Spectrum sensing is commonly considered as a binary hypothesis testing problem. Under the H_0 hypothesis, the primary user is absent and the primary user is present under H_1 hypothesis:

$$\begin{aligned} H_0 &: y(n) = w(n) \approx N(0, \sigma_w^2) \\ H_1 &: y(n) = \overbrace{s(n) \otimes h(n)}^{x(n)} + w(n) \approx N(0, \sigma_x^2 + \sigma_w^2) \end{aligned} \tag{1}$$

Here $s(n)$ is the primary user signal, $h(n)$ is the channel impulse response and $w(n)$ is assumed to be additive white Gaussian noise (AWGN).

The statistical covariance matrices of the received signal, transmitted signal and noise are defined as:

$$\mathbf{R}_{yy} = E(\hat{\mathbf{y}}\hat{\mathbf{y}}^{\dagger}); \quad \mathbf{R}_{ss} = E(\hat{\mathbf{s}}\hat{\mathbf{s}}^{\dagger}); \quad \mathbf{R}_{ww} = E(\hat{\mathbf{w}}\hat{\mathbf{w}}^{\dagger}) \tag{2}$$

Here $\dagger$ denotes the conjugate transpose. The covariance matrix of the received signal can be further expressed as [2]:

$$\mathbf{R}_{yy} = \mathbf{H}_c \mathbf{R}_{ss} \mathbf{H}_c^{\dagger} + \mathbf{R}_{ww} \tag{3}$$

The channel matrix $\mathbf{H}_c$ is given in [2] and eigenvalues of $\mathbf{R}_{yy}$ and $\mathbf{H}_c \mathbf{R}_{ss} \mathbf{H}_c^{\dagger}$ are defined as $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_{ML}$ and $\rho_1 \geq \rho_2 \geq ... \geq \rho_{ML}$, respectively. Due to the transmitter waveform processing and/or due to the multipath channel, the received signal values are correlated and the eigenvalues appear far away from each other. Zeng and Liang presented in

CROWNCOM 2013, July 08-10, Washington DC, United States

DOI 10.4108/icst.crowncom.2013.252014

[2] two eigenvalue based algorithms for spectrum sensing, which are described in the following.

A. Algorithm 1: Max-Min eigenvalue based sensing (MME)

Statistical covariance matrix can be estimated using the sample covariance matrix, which is obtained by averaging N sample covariance matrices:

$$\mathbf{R}_{yy}(N) = \frac{1}{N}\sum_{n=ML-1}^{L-2+N} \hat{\mathbf{y}}(n)\hat{\mathbf{y}}(n)^{\dagger} \tag{4}$$

Here n indicates the last sample used in the calculation of each covariance estimate.

The sample autocorrelations of the received signal which are defined as follows:

$$r_y(l) = \frac{1}{N}\sum_{n=0}^{N} y(n)y^*(n-l), \quad l = 0,1,2\ldots ML-1 \tag{5}$$

where * denotes complex-conjugation. The sample covariance matrix can then be expressed as

$$\mathbf{R}_{yy} = \begin{pmatrix} r_y(0) & r_y(1) & \cdots & r_y(ML-1) \\ r_y^*(1) & r_y(0) & \ddots & r_y(ML-2) \\ \vdots & \ddots & \ddots & \vdots \\ r_y^*(ML-2) & r_y^*(ML-3) & \vdots & r_y(1) \\ r_y^*(ML-1) & r_y^*(ML-2) & \cdots & r_y(0) \end{pmatrix} \tag{6}$$

Next, the largest and smallest eigenvalues $(\lambda_{\max}, \lambda_{\min})$ of the sample covariance matrix $\mathbf{R}_{yy}(N)$ are computed and the ratio of $\lambda_{\max}/\lambda_{\min}$ is compared with the threshold γ_1 which is calculated according to the distribution of the noise sample covariance matrix. The threshold value is calculated using desired false alarm probability (i.e. P_{FA}), N, M and L as follows [2]:

$$\gamma_1 = \frac{(\sqrt{N}+\sqrt{ML})^2}{(\sqrt{N}-\sqrt{ML})^2}\cdot\left(1+\frac{(\sqrt{N}+\sqrt{ML})^{-2/3}}{(NML)^{1/6}}F_1^{-1}(1-P_{FA})\right) \tag{7}$$

F_1 is the cumulative distribution function (CDF) of the Tracy-Widom distribution of order 1, which was derived using random matrix theorem [8] as:

$$F_1(t) = \exp\left(-\frac{1}{2}\int_t^{\infty}(q(u)+(u-t)q^2(u))du\right) \tag{8}$$

where $q(u)$ is the solution of the nonlinear Painleve II differential equation $q''(u) = uq(u) + 2q^3(u)$

TABLE 1 gives the values of F_1 at some points. Also F_1^{-1} can be obtained using same table

TABLE I. NUMERICAL TABLE FOR THE TRACY-WIDOM DISTRUBITON OF ORDER 1

t	-3.90	-2.78	-1.91	-1.27	-0.59	0.45	2.02
$F_1(t)$	0.01	0.10	0.30	0.50	0.70	0.90	0.99

When $(\lambda_{\max}/\lambda_{\min}) > \gamma_1$, the primary signal is assumed to be present, otherwise it is assumed that there is no transmitted signal in the band of interest at this time.

B. Algorithm 2: Energy with min eigenvalue based sensing (EME)

The sample covariance matrix $\mathbf{R}_{yy}(N)$ and the smallest eigenvalue $\lambda_{\min}$ of the sample covariance matrix are calculated in the same way with *Algorithm 1*. The average energy of the received signal is computed like in the traditional energy detector as:

$$T(N) = \frac{1}{MN}\sum_{n=0}^{NM-1}\left|y(n)\right|^2 \tag{9}$$

The threshold value γ_2 is calculated with the inverse Q-function Q^{-1} using random matrix theorem [11] as follows

$$\gamma_2 = \left(\sqrt{\frac{1}{MN}}Q^{-1}(P_{FA})+1\right)\frac{N}{(\sqrt{N}-\sqrt{ML})^2} \tag{10}$$

When $(T(N)/\lambda_{\min}) > \gamma_2$, the signal is assumed to be present, otherwise it is expected that there is only noise in the band of interest.

In the calculation of the threshold values for the two algorithms, M, N, L and P_{FA} parameters are only used. No previous knowledge of the estimated noise variance is needed; hence, these two algorithms are very robust against noise uncertainty.

C. Computational complexity of the traditional eigenvalue based spectrum sensing techniques

The main computational complexities of the aforementioned methods arise from computation of the covariance matrix and finding its largest and smallest eigenvalues. In the first part, calculation of the covariance matrix of received signal from (4) includes M^2L^2N complex multiplications and $M^2L^2(N-1)$ additions. However, since the covariance matrix is block Toeplitz and Hermitian, the complexity of the first part can be obtained as MLN multiplications and $ML(MN-1)$ additions. For the second part, calculation of the eigenvalues requires $O(M^3L^3)$ multiplications and additions [2]. An additional complexity of $O(M^2L^2)$ might also be required to sort all eigenvalues to determine the minimum and maximum. The total complexity is $MLN + O(M^3L^3)$ from (6). Since N is usually much greater than ML, the complexity required for computing the covariance matrix may dominate. In [10], Liu *et al.* suggest a cooperative sensing method to reduce N without significantly deteriorating the performance of the spectrum sensing algorithms. However, assuming the simplest scenario of no cooperation and single receiver antenna, and considering the value of ML becoming substantially large for real life applications, the computational complexity of eigenvalue decomposition can no longer be neglected.

III. PROPOSED EIGENVALUE BASED ALGORITHMS

Since only the largest and smallest eigenvalues are required, there exist a few algorithms which approximate the extreme eigenvalues of a matrix by exploiting its Hermitian or Toeplitz structure [11]-[17]. However, some of them suffer from high computational complexities or numerical instability, whilst others are difficult to implement in practice. Therefore, we consider the power method and inverse power method, due to

their low computational complexities and ease of implementation.

A. The Power Method to find the largest eigenvalue

The power method is an iterative algorithm which approximates the largest dominant eigenvalue of a symmetric positive definite matrix in $O(kML)$ operations, where k is the number of iterations under a certain error threshold. The theoretical basis of the algorithm is as follows [18]:

Let $\{\mathbf{v}_i, \lambda_i\}_i^n$ be the m eigen-pairs of a symmetric positive definite matrix $\mathbf{R}_{yy}$. Since $\mathbf{R}_{yy}$ is symmetric, we assume that the eigenvectors $\{\mathbf{v}_i\}_i^m$ form an orthogonal basis of $\mathbb{R}^N$. Hence, if we begin with an initial eigenvector guess $\mathbf{v}$, $\mathbf{v}$ can be expressed as a linear combination of the basis $\mathbf{v}_i$:

$$\mathbf{v} = \sum_{i=1}^{m} \alpha_i \mathbf{v}_i \tag{11}$$

where $\alpha_1 \neq 0$. Assume $\mathbf{v}_1$ to be the largest and distinct eigenvector (in absolute), then multiplying $\mathbf{R}_{yy}^k$ yields

$$\mathbf{R}_{yy}^k \mathbf{v} = \sum_{i=1}^{m} \alpha_i \mathbf{R}_{yy}^k \mathbf{v}_i \tag{12}$$

$$= \sum_{i=1}^{m} \alpha_i \lambda_i^k \mathbf{v}_i \quad \textit{(by definition of eigenvalue)}$$

Let λ_1 be the dominant eigenvalue, then divide (12) by λ_1^k to give

$$\frac{\mathbf{R}_{yy}^k \mathbf{v}}{\lambda_1^k} = \alpha_1 \mathbf{v}_1 + \sum_{i=2}^{m} \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^k \mathbf{v}_i \tag{13}$$

Since $\lambda_1 > \lambda_i$, for $i = 2,...,m$, the term in summation vanishes for some large positive integer k. Hence, the term on the left (known as the Rayleigh quotient) eventually converges to the largest eigenvector $\mathbf{v}_1$ scaled by some non-zero constant α_1.

The power method thus runs as follows:

> ***Input:*** $\mathbf{R}_{yy}$, *the matrix;* $\mathbf{v}_0$, *an initial guess of an eigenvector such that* $\|\mathbf{v}_0\| = 1$
>
> ***For*** $k = 1, 2, 3, \ldots,$ ***do***
>
> $\mathbf{w} \leftarrow \mathbf{R}_{yy}\mathbf{v}_{k-1}$
>
> $\mathbf{v}_k \leftarrow \mathbf{w}/\|\mathbf{w}\|$
>
> $\lambda_k \leftarrow \mathbf{v}_k^{\dagger}\mathbf{R}_{yy}\mathbf{v}_k$
>
> ***End for***
>
> ***Output:*** λ_k, *the approximation of the maximum eigenvalue of* $\mathbf{R}_{yy}$ *after the* k^{th} *iteration.*

B. The Inverse Power Method to find the smallest eigenvalue

The inverse power method is an iterative algorithm which approximates the smallest eigenvalue, without finding and sorting all eigenvalues.

Theorem 1*: Let* $\mathbf{R}_{yy}$ *be a non-singular* $m \times m$ *matrix i.e.* λ_i *are non-zero for all* $1 \le i \le m$*, then* $\mathbf{R}_{yy}^{-1}$ *has eigenvalues* $1/\lambda_i$ *for all* $1 \le i \le m$ [14], [16].

Proof: $\in \mathbf{R}_{yy}$ *is non-singular, there exists* $\lambda_i \neq 0$ *and there exists* $x \neq 0$ *such that*

$$\mathbf{R}_{yy}\mathbf{x} = \lambda_i \mathbf{x} \Rightarrow \underbrace{\mathbf{R}_{yy}^{-1}\mathbf{R}_{yy}}_{I}\left(\frac{1}{\lambda_i}\mathbf{x}\right) = \mathbf{R}_{yy}^{-1}\mathbf{x}$$
$$\Rightarrow \frac{1}{\lambda_i}\mathbf{x} = \mathbf{R}_{yy}^{-1}\mathbf{x} \tag{14}$$

Hence, by definition of eigenvalue, all $1/\lambda_i$ are eigenvalues of $\mathbf{R}_{yy}^{-1}$. The inverse power method is therefore applying the power method on the inverse of the matrix for approximating the smallest dominant eigenvalue. However, computing R_{yy}^{-1} explicitly is numerically expensive and unstable, and generally takes $O(M^4L^4)$ operations using naive Gaussian elimination. A common implementation of the inverse power iteration uses LU decomposition, which has a lower complexity of $O(4/3M^3L^3)$ Hence, the overall computational complexity of the inverse power method is $O(M^3L^3)$, as the complexity of LU factorization dominates. However, since a covariance matrix is Hermitian, the LU decomposition can be simplified to Cholesky decomposition, i.e. $\mathbf{R}_{yy} = \mathbf{C}\mathbf{C}^{\dagger}$,where $\mathbf{C}$ is an upper or lower triangular matrix. For a general Hermitian positive definite matrix, this process takes $O(M^3L^3/3)$ operations.

C. Fast Cholesky factorization of Toeplitz Matrix

As such, there are many well-known algorithms which factorize Toeplitz or Toeplitz-like matrices in less than $O(M^3L^3)$ steps, amongst them the popular Levinson algorithm and Schur algorithm, both of which achieve triangulation of a Toeplitz matrix (or its inverse) with $O(M^2L^2)$ operations [19]. While both classes of algorithms are of comparable computational complexities, the Schur algorithm has the additional advantage of being able to exploit parallel computing to further reduce its computational latency to a linear relation [20], [21], [22]. Therefore, we propose the Schur algorithm for fast Cholesky factorization of our covariance matrix. An outline of the Schur algorithm for finding the Cholesky factor of positive definite Toeplitz matrices is presented here based on [23], [24], [25] and [26].

It is known that a symmetric Toeplitz matrix $\mathbf{T}$ fulfills the following relationship:

$$\mathbf{T} - \mathbf{Z}\mathbf{T}\mathbf{Z}^T = \mathbf{D} \tag{15}$$

where $\mathbf{Z}$ is a shift-down matrix and $\mathbf{D}$ the *displacement* of $\mathbf{T}$, which can be further expressed as:

$$\mathbf{D} = \mathbf{G}\boldsymbol{\Sigma}\mathbf{G}^T \tag{16}$$

where

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} \mathbf{u} & \mathbf{v} \end{bmatrix} \tag{17}$$

with $\mathbf{u} = [t_0, t_1, ..., t_{n-2}, t_{n-1}]^T$, $\mathbf{v} = [0, t_1, ..., t_{n-2}, t_{n-1}]^T$ and t_i being the i^{th} column vector of $\mathbf{T}$.

A Schur algorithm operates only on the generating vectors $\mathbf{u}$ and $\mathbf{v}$ instead of the entire Toeplitz matrix to compute the columns of the Cholesky factor.

Assume we have $\mathbf{G} = [\mathbf{u} \quad \mathbf{v}]$, we can make the first element of $\mathbf{v}$ to be 0 by applying a hyperbolic transformation $\mathbf{H}_t$ on $\mathbf{G}$ such that:

$$\mathbf{H}_t\mathbf{G} = [\tilde{\mathbf{u}}, \tilde{\mathbf{v}}] \tag{18}$$

Where $\tilde{\mathbf{u}} = [\tilde{u}_0, \tilde{u}_1 ..., \tilde{u}_{n-2}, \tilde{u}_{n-1}]^T$ and $\tilde{\mathbf{v}} = [0, \tilde{v}_1 ..., \tilde{v}_{n-2}, \tilde{v}_{n-1}]^T$ With $\mathbf{H}_t$ satisfying the relation $\mathbf{H}_t^T \boldsymbol{\Sigma} \mathbf{H}_t = \boldsymbol{\Sigma}$. Hence, the transformed generator $\mathbf{H}_t\mathbf{G}$ is also a generator for $\mathbf{T}$:

$$(\mathbf{H}_t\mathbf{G})^T \boldsymbol{\Sigma} (\mathbf{H}_t\mathbf{G}) = \mathbf{G}\boldsymbol{\Sigma}\mathbf{G}^T \tag{19}$$

The transformed vector $\tilde{\mathbf{u}}$ is the first column of the Cholesky factor of $\mathbf{T}$. Also introduce $\mathbf{T}_S = \mathbf{T} - \tilde{\mathbf{u}}\tilde{\mathbf{u}}^T$ to be the *Schur complement of* $\mathbf{T}$. Then, we observe that applying the shift matrix $\mathbf{Z}$ to the generator after the hyperbolic transformation yields

$$\mathbf{Z}\tilde{\mathbf{u}}\tilde{\mathbf{u}}^T\mathbf{Z}^T - \tilde{\mathbf{v}}\tilde{\mathbf{v}}^T = \mathbf{T}_S - \mathbf{Z}\mathbf{T}_S\mathbf{Z}^T \tag{19}$$

which actually generates the Schur complement. As a result, applying further hyperbolic transformations on expression (19) introduces zero to the first element of $\tilde{\mathbf{v}}$. This is the second column of the Cholesky factor of $\mathbf{T}$. Successive iterations of the above process yield the Cholesky factor of the Toeplitz matrix. The positivity of the Schur complement $\mathbf{T}_S$ at each stage ensures that the hyperbolic transformation matrix $\mathbf{H}_t$ can be found.

IV. Simulations and Numerical Results

The numerical stability of the inverse power method using Cholesky and Schur algorithms is investigated and shown in Figure 1. Different sizes of sample covariance matrices are generated with three different smoothing factors as $L = \{4, 8$ and $16\}$ to see the deviation of the two algorithms. In Figure 1, the inverse method using both the Schur algorithm and Matlab's Cholesky function are shown and compared with the traditional eigenvalue process. We observe no significant numerical instability for the Schur algorithm, as the matrix dimension increases. Numerical instability of Schur algorithm for high size of covariance matrix has been discussed in [21], [25], where in [25]. In particular, Stewart and Van Dooren provided in [25] a detailed analysis which showed that implementation of the hyperbolic transformation plays a pivotal role in ensuring the numerical stability of Schur algorithm. However, since in our study the size of the sample covariance matrix (ML) is small, the Schur algorithm is numerically stable. When applying the inverse power iteration with the Schur algorithm for spectrum sensing, we observe almost the same performance as when using the traditional eigen-decomposition method.

The detection probabilities of simple energy detector, as well as traditional and proposed eigenvalue based spectrum sensing methods have been evaluated using two different channel models (i.e., Indoor and ITU-R Vehicular A channels [27]). The 1 dB noise uncertainty case is considered as the worst-case scenario in terms of noise variance estimation. In all the addressed cases, the time record length is 10000 complex samples, the smoothing factor of $L = 16$ and 1000 Monte Carlo simulations are applied to evaluate the detection probability reliably. The bandwidth is chosen as 20 MHz and the Vehicular A channel model has 6 taps and its maximum delay spreads is about 2.5μs. The number of iterations k is chosen as 100 in the following figures to obtain similar detection probability performances. The oversampling process uses a lowpass FIR filter designed with Kaiser window.

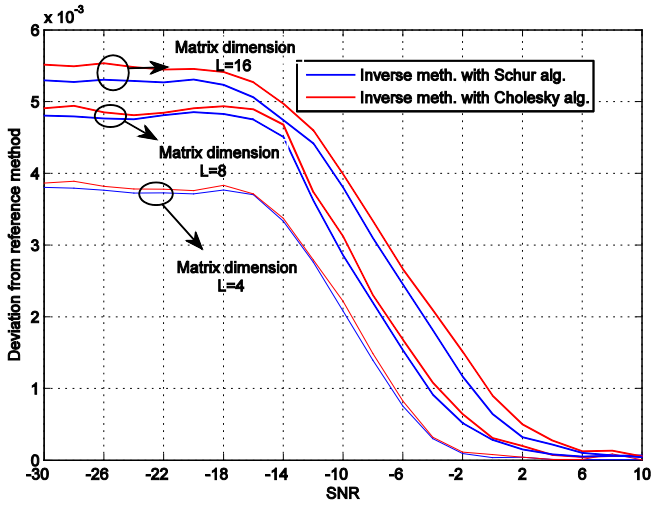

Figure 1. Difference of computed smallest eigenvalue using Schur or Cholesky based inverse power methods in comparison with eigen-decomposition method, for varying matrix dimension under different SNR cases.

The detection probabilities of traditional and proposed eigenvalue based spectrum sensing with non-oversampled signal under Indoor channel and 4x-oversampled signal under ITU-R VehicularA with 0.1 target P_{FA} are depicted in Figure 2 and Figure 3, respectively.

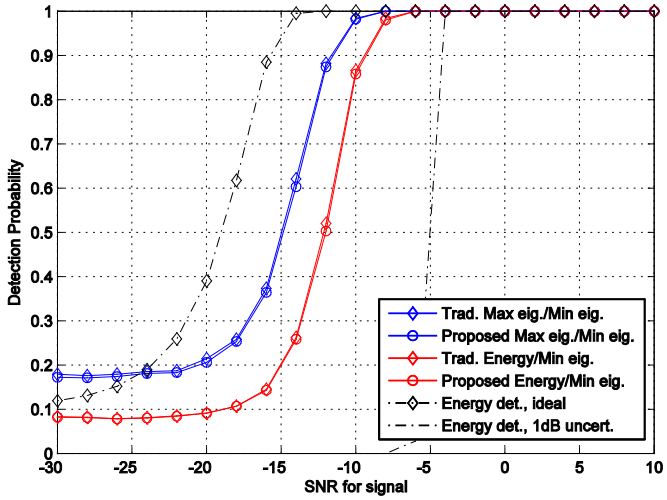

Figure 2. Simulated detection probabilities using traditional and proposed eigenvalue-based spectrum sensing algorithms with $M = 1$ (non-oversampled), $L = 16$ and Indoor channel. Theoretical performance of energy detector without noise uncertainty and with 1 dB noise uncertainty included as reference.

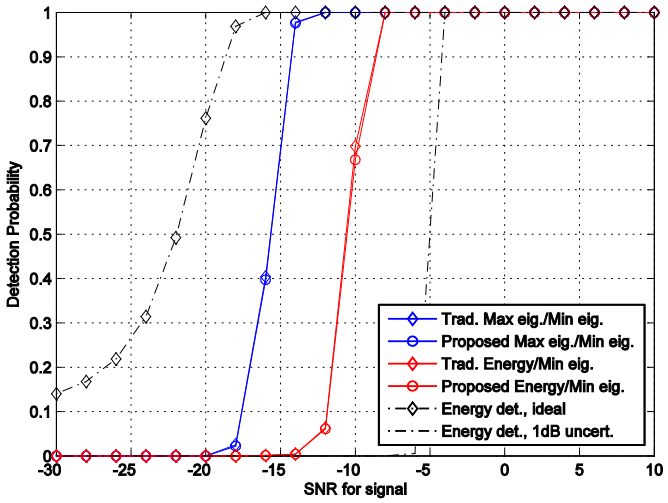

Figure 3. Simulated detection probabilities using traditional and proposed eigenvalue-based spectrum sensing algorithms with oversampling by $M = 4$, $L = 16$, and ITU-R Vehicular A channel. Theoretical performance of energy detector without noise uncertainty and with 1 dB noise uncertainty included as reference.

Frequency selective channels increase the detection probability of eigenvalue based spectrum sensing algorithms.

Quite similar detection performance is obtained with both traditional and proposed eigenvalue based spectrum sensing techniques, in all the addressed cases. TABLE II shows the computational complexities of traditional and proposed eigenvalue based spectrum sensing techniques. TABLE III, TABLE IV and TABLE V include some numerical results for all the considered algorithms. Depending on the values of L, M and N, significant reduction of the computational complexity can be reached. The stability of our proposed iterative algorithms for spectrum sensing is also confirmed under the given simulation cases.

V. Conclusion

An improvement to the computational complexity of eigenvalue-based spectrum sensing has been presented in this paper, based on the simple power iteration and inverse power iteration using the Schur algorithm. In general, the max/min eigenvalue method provides consistently better performance/complexity tradeoff than the energy with min eigenvalue method. On the other hand, the Schur algorithm has in many cases significanlty lower complexity than the Cholesky approach. The overall computational complexity can be reduced from $O((ML)^3)$ to $O((ML)^2)+O(kML)$ using the Schur algorithm. For instance from TABLE V, when the number of samples is 10^3 with 4x-oversampling, the overall computational complexity (multiplication and addition) of the traditional algorithm 1 is 326144 whereas it is 74496 Schur algorithm. Hence upon using the Schur algorithm, the complexity is reduced by about 80 percent.

In the future, improvements to the stability of Schur algorithm can be explored [19]-[25], as well as implementation of a class of asymptotically superfast algorithms [22],[26] or eigenvalue algorithms based on superfast Toeplitz solvers [16], [17], which might further reduce the computational complexity to $O(ML(\log_2 ML)^2)$. Besides numerical algorithms, other aspects of cognitive radio, such a cooperative sensing should also be investigated to reduce the computational complexity.

Acknowledgment

This work was partially supported by Tekniikan Edistämissäätiön (TES), GETA Graduate School, the European Commission, COST Action IC0902 (Cognitive Radio and Networking for Cooperative Coexistence of Heterogeneous Wireless Networks) and Tekes (the Finnish Funding Agency for Technology and Innovation) under the project ENCOR in the Trial Program.

References

[1] A. Kortun, T. Ratnarajah, M. Sellathurai, C. Zhong and C.B. Papadias, "On the Performance of Eigenvalue-Based Cooperative Spectrum Sensing for Cognitive Radio," *Selected Journal of Topics in Sign. Proc., IEEE*, vol.5, no.1, pp.49-55, Feb. 2011.

[2] Yonghong Zeng and Ying-Chang Liang, "Eigenvalue-based spectrum sensing algorithms for cognitive radio," *Trans. Comm.* 57, June 2009.

[3] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Comm. Surv. & Tut.s*, vol. 11, no. 1, pp. 116–130, March 2009.

[4] E.Guzzon, F.Benedetto and G.Giunta, "Performance Improvements of OFDM Signals Spectrum Sensing in Cognitive Radio" *IEEE 76th Vehicular Technology Conf. (VTC2012)*,3-6 Sept. 2012.

[5] S. Dikmese and M. Renfors, "Performance Analysis of Eigenvalue Based Spectrum Sensing under Frequency Selective Channels," *7th CROWNCOM, Stockholm, Sweden*, 19 June 2012.

[6] L.S. Cardoso, M. Debbah, P. Bianchi and J. Najim, "Cooperative spectrum sensing using random matrix theory," *Wireless Pervasive Computing, 2008.* ISWPC 2008. pp.334-338, 7-9 May 2008.

[7] B. Zayen and A, Hayar, "Cooperative Spectrum Sensing Technique Based on Sub Space Analysis for Cognitive Radio Networks," *COGIS'09, COGnitive systems with Interactive Sensors Conf.*, Paris,France,16-18 Nov. 2009

[8] A. M. Tulino and S. Verdu, "Random Matrix Theory and Wireless Communications" *Now Publishers Inc., 2004.*

[9] C. A. Tracy and H. Widom, "The distribution of the largest eigenvalue in the gaussian ensembles," *in Calogero-Moser-Sutherland Models, J. van Diejen and L. Vinet, eds.*, pp. 461-472. Newyork: Springer, 2000.

[10] Z. Liu, H. Zhang, S. Sun and D. Zhu, "Analysis and Improvement of Eigenvalue Spectrum Sensing in Cognitive Radio Networks," *Wireless Comm. and Sign. Proc. (WCSP), 2010 Intern. Conf.*, 21-23 Oct. 2010

[11] I. Dimov, V. N. Alexandrov and A. Karaivanova, "Implementation of Monte Carlo Algorithms for Eigenvalue Problem Using MPI," Proc. of the 5th European PVM/MPI *Vassil Alexandrov and Jack Dongarra (Eds.). Springer-Verlag, London, UK*, pp. 346-353, 1998.

[12] W. Mackens and H. Voss, "A projection method for computing the minimum eigenvalue of a symmetric positive definite Toeplitz matrix", *Linear Alg. and its App.* , Volumes 275–276, pp. 401-415, 15 May 1998

[13] P. Drineas, R. Kannan, and M. W. Mahoney, "Fast Monte Carlo Algorithms for Matrices I: Approximating Matrix Multiplcation," *Siam Journ.on Comp.*, Vol. 36, No. 1, pp.132–157, 2006.

[14] B. Jansen, B. C. Roos and T. Terlaky, "A New Algorithm for the Computation of the Smallest Eigenvalue of a Symmetric Matrix and Its Eigenspace," *Research Report Delf Uni. Of Technolgy* 1995

[15] Y. Okamoto and H. J. Maris, "A novel algorithm for calculation of the extreme eigenvalues of large Hermitian matrices," *Computer Physics Comm.* vol 76, Issue 2, pp.191-202, July 1993.

[16] A, Kostic and H. Voss, "A method of order $1+\sqrt{3}$ for computing the smallest eigenvalue of a symmteric Toeplitz matrix," review paper,

accessed from *http://www.tu-harburg.de/ins/forschung/rep/rep45.pdf on 30 July 2012*.

[17] H. Voss, "A symmtery exploiting Lanzcos method for symmteric Toeplitz matrices," *Numerical Algorithms*, vol 25, Issue 1-4, pp. 377-385, September 2000.

[18] L. Reichel, "Lecture 14: eigenvalue computations," *lecture note for MATH 42202/52202: Introduction to Numerical Computing II*, Kent State University, *accessed from http://www.math.kent.edu/~reichel/courses/intr.num.comp.2/lecture21/evmeth.pdf on 25 July 2012*.

[19] I. C. F. Ipsen, "Some Remarks On the Generalised Bareiss and Levinson Algorithms," *Research Report, Yale University 1990*.

[20] P. G. Martinsson, V. Rokhlin, and M. Tygert, "A fast algorithm for the inversion of general Toeplitz matrices" *Comput. Math. Appl*. 50, pp. 741-752, 5-6 Sept. 2005.

[21] S. Chandrasekaran and Ali H. Sayed**,** "Stabilizing the generalized Schur algorithm," *SIAM. J. Matrix Anal. & Appl., vol 17, Issue 4, pp. 950–83, Oct 1996.*

[22] T. Huckle, "Implementation of a superfast algorithm for symmtric positive-definite linear equations of displacement rank 2," *Adv.Sign. Proc.Alg., Arch., and Impl. V , Proc. SPIE Vol. 2296, pp. 494-503, Franklin T. Luk; Ed, Oct 1994.*

[23] M. Stewart, "Cholesky factorization of semi-definite Toeplitz matrices," *Lin. Alg. Appl*, vol 254, pp. 497-526, 1997.

[24] K. A. Gallivan, S. Thirumalai, P. Van Dooren and V. Vermaut, "High performance algorithms for Toeplitz and block Toeplitzmatrices," *Lin. Alg. Appl*, vol. 241–243, pp. 343–388, 1996.

[25] M. Stewart and P. V. Dooren, "Stability Issues in the Factorization of Structured Matrices," *SIAM J. Matrix Anal. Appl. 18, 1*, pp. 104-118, Jan 1997.

[26] G. Ammar and W. Gragg, "The generalized Schur algorithm for the superfast solution of Toeplitz systems," *Rational App.and its Appl. in Math. and Physics, Lect. Notes in Math., vol 1237, Springer Berlin/Heidelberg, ISBN 978-3-540-17212-3, 1987.*

[27] R. Jain, *"Channel Models: A Tutorial," WiMAX Forum AATG, February 2007.*

TABLE II. COMPUTATIONAL COMPLEXITY OF POWER ITERATION BASED SPECTRUM SENSING

ALGORITHMS			***Cov. Matrix***	***Eigen. Decom.***		***Ave. Test Stat.***	***Total***
				max	**min**		
Trad. Alg.	**Alg. 1 (max. eig. / min. eig.)**		MLN	$O(M^3L^3)$			$MLN + O(M^3L^3)$
	Alg. 2 (average / min. eig.)		MLN	$O(M^3L^3)$		MN	$MLN + O(M^3L^3) + MN$
Prop. Alg.	**Alg. 1 (max. eig. / min. eig.)**	***Cholesky factorization***	MLN	$O(kML)$	$O(M^3L^3 \ / \ 3)$	-	$MLN + O(kML) + O(M^3L^3 \ / \ 3)$
		Schur Algorithm			$O(M^2L^2)$		$MLN + O(kML) + O(M^2L^2)$
	Alg. 2 (average / min. eig.)	***Cholesky factorization***	MLN	-	$O(M^3L^3 \ / \ 3)$	MN	$MLN + O(M^3L^3 \ / \ 3) + MN$
		Schur Algorithm			$O(M^2L^2)$		$MLN + O(M^2L^2) + MN$

TABLE III. SOME NUMERICAL VALUES OF COMPUTATIONAL COMPLEXITIES FOR SENSING METHODS (M=1, NON- OVERSAMPLED)

ALGORITHMS			***Smoothing Factor (L)***	***Number of samples (N)***			
				10^3	$5x10^3$	10^4	$2x10^4$
Trad. Alg.	**Alg. 1 (max. eig. / min. eig.)**		**8**	8 512	40 512	80 512	160 512
			16	20 096	84 096	164 096	324 096
	Alg. 2 (average / min. eig.)		**8**	9 512	45 512	90 512	180 512
			16	21 096	89 096	174 096	344 096
Prop. Alg.	**Alg. 1 (max. eig. / min. eig.)**	**Cholesky**	**8**	8 971	40 971	80 971	160 970
			16	18 965	82 965	162 970	321 856
		Schur	**8**	8 864	40 864	80 864	160 864
			16	17 856	81 856	161 856	321 856
	Alg. 2 (average / min. eig.)	**Cholesky**	**8**	9 171	45 171	90 171	180 170
			16	18 365	86 365	171 370	341 370
		Schur	**8**	9 064	45 064	90 064	180 064
			16	17 256	85 256	170 256	340 256

TABLE IV. SOME NUMERICAL VALUES OF COMPUTATIONAL COMPLEXITIES FOR SENSING METHODS (M=2X-OVERSAMPLED)

ALGORITHMS			*Smoothing Factor (L)*	*Number of samples (N)*			
				10^3	$5x10^3$	10^4	$2x10^4$
Trad. Alg.	**Alg. 1 (max. eig. / min. eig.)**		**8**	20 096	40 512	164 096	324 096
			16	64 768	84 096	352 768	672 768
	Alg. 2 (average / min. eig.)		**8**	22 096	45 512	184 096	364 096
			16	66 768	89 096	372 768	712 768
Prop. Alg.	**Alg. 1 (max. eig. / min. eig.)**	**Cholesky**	**8**	18 965	40 971	162 970	322 970
			16	46 123	82 965	334 120	654 120
		Schur	**8**	17 856	40 864	161 856	321 856
			16	36 224	81 856	324 224	644 224
	Alg. 2 (average / min. eig.)	**Cholesky**	**8**	19 365	45 171	181 370	361 370
			16	44 923	86 365	350 920	690 920
		Schur	**8**	18 256	45 064	180 256	321 856
			16	35 024	85 256	341 024	681 024

TABLE V. SOME NUMERICAL VALUES OF COMPUTATIONAL COMPLEXITIES FOR SENSING METHODS (M=4X-OVERSAMPLED)

ALGORITHMS			*Smoothing Factor (L)*	*Number of samples*			
				10^3	$5x10^3$	10^4	$2x10^4$
Trad. Alg.	**Alg. 1 (max. eig. / min. eig.)**		**8**	64 768	84 096	352 768	672 768
			16	326 144	192 768	902 144	1 542 144
	Alg. 2 (average / min. eig.)		**8**	68 768	94 096	392 768	752 768
			16	330 144	202 768	942 144	1 622 144
Prop. Alg.	**Alg. 1 (max. eig. / min. eig.)**	***Cholesky***	**8**	46 123	82 965	334 120	654 120
			16	157 780	174 120	733 780	1 373 800
		Schur	**8**	36 224	81 856	324 224	644 224
			16	74 496	164 224	650 496	1 290 496
	Alg. 2 (average / min. eig.)	***Cholesky***	**8**	46 923	91 365	370 920	730 920
			16	155 380	180 920	767 380	1 447 400
		Schur	**8**	37 024	90 256	361 024	721 024
			16	72 096	171 024	684 096	1 364 096