Reducing Computational Complexity of Eigenvalue Based Spectrum Sensing for Cognitive Radio

Sener Dikmese¹, Jiunn Lin Wong², Ahmet Gokceoglu¹, Elena Guzzon³, Mikko Valkama¹ and Markku Renfors¹

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

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

³Depart. 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:

$$H_0 : y(n) = w(n) \approx N(0, \sigma_n^2)$$

$$H_1 : y(n) = s(n) \otimes h(n) + w(n) \approx N(0, \sigma_s^2 + \sigma_n^2)$$

(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:

$$R_{yy} = E(\hat{y}\hat{y}^\dagger) ; \quad R_{ss} = E(\hat{s}\hat{s}^\dagger) ; \quad R_{ww} = E(\hat{w}\hat{w}^\dagger)$$

(2)

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

$$R_{yy} = H_H R_{ss} H_H^T + R_{ww}$$

(3)

The channel matrix $H$ is given in [2] and eigenvalues of $R_{yy}$ and $H_H R_{ss} H_H^T$ are defined as $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_M$ and $\rho_1 \geq \rho_2 \geq \ldots \geq \rho_M$, 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
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:

\[
R_{yy}(N) = \frac{1}{N} \sum_{n=M-L-1}^{L-2+N} \tilde{y}(n)\tilde{y}^*(n)
\]

(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:

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

(5)

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

\[
R_{yy} = \begin{pmatrix}
    r_y(0) & r_y(1) & \cdots & r_y(ML-1) \\
r_y(1) & r_y(0) & \cdots & r_y(ML-2) \\
    \vdots & \vdots & \ddots & \vdots \\
r_y(ML-2) & r_y(ML-3) & \cdots & r_y(1) \\
r_y(ML-1) & r_y(ML-2) & \cdots & r_y(0)
\end{pmatrix}
\]

(6)

Next, the largest and smallest eigenvalues \( \lambda_{\text{max}}, \lambda_{\text{min}} \) of the sample covariance matrix \( R_{yy}(N) \) are computed and the ratio of \( \lambda_{\text{max}}/\lambda_{\text{min}} \) 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]:

\[
\gamma_1 = \frac{(N + \sqrt{ML})^2}{(N - \sqrt{ML})^2} \left[ 1 + \frac{(N + \sqrt{ML})^{2/3}}{(NML)^{1/3}} F_1^{-1}(1 - P_{fa}) \right]
\]

(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:

\[
F_1(t) = \exp\left( -\frac{t^2}{2} \int q(u) + (u-t)q^2(u) \, du \right)
\]

(8)

where \( q(u) \) is the solution of the nonlinear Painlevé II differential equation \( q''(u) + uq(u) + 2q^3(u) = 0 \).

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

<table>
<thead>
<tr>
<th>TABLE I. NUMERICAL TABLE FOR THE TRACY-WIDOM DISTRIBUTION OF ORDER 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t )</td>
</tr>
<tr>
<td>( F_1(t) )</td>
</tr>
</tbody>
</table>

When \( \lambda_{\text{max}}/\lambda_{\text{min}} > \gamma_1 \), 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 \( R_{yy}(N) \) and the smallest eigenvalue \( \lambda_{\text{min}} \) 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:

\[
T(N) = \frac{1}{MN} \sum_{n=0}^{NML-1} |y(n)|^2
\]

(9)

The threshold value \( \gamma_2 \) is calculated with the inverse Q-function \( Q^{-1} \) using random matrix theorem [11] as follows

\[
\gamma_2 = \left( 1 - \frac{1}{\sqrt{NML}} - \frac{1}{NML} \right) \frac{N}{(NML)^{1/2}}
\]

(10)

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

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 \( M^2NL \) complex multiplications and \( M^2L(N-1) \) additions. However, since the covariance matrix is block Toeplitz and Hermitian, the complexity of the first part can be obtained as \( MLN \) multiplications and \( ML(NM-1) \) additions. For the second part, calculation of the eigenvalues requires \( O(M^2L^2) \) 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 \( MLN + O(M^2L^2) \) 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 $\{v_i, \lambda_i\}$ be the $m$ eigen-pairs of a symmetric positive definite matrix $R_{yy}$. Since $R_{yy}$ is symmetric, we assume that the eigenvectors $\{v_i\}$ form an orthogonal basis of $\mathbb{R}^N$. Hence, if we begin with an initial eigenvector guess $v$, $v$ can be expressed as a linear combination of the basis $v_i$:

$$v = \sum_{i=1}^{m} \alpha_i v_i$$  \hspace{1cm} (11)

where $\alpha_i \neq 0$. Assume $v_1$ to be the largest and distinct eigenvector (in absolute), then multiplying $R_{yy}$ yields

$$R_{yy}v = \sum_{i=1}^{m} \alpha_i R_{yy}v_i$$

$$= \sum_{i=1}^{m} \alpha_i \lambda_i v_i \quad \text{(by definition of eigenvalue)}$$

Let $\lambda_i$ be the dominant eigenvalue, then divide (12) by $\lambda_i$ to give

$$R_{yy}v/\lambda_i^k = \alpha_1 v_1 + \sum_{i=2}^{m} \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^k v_i$$

Since $\lambda_i > \lambda_1$, 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 $v_1$, scaled by some non-zero constant $\alpha_1$.

The power method thus runs as follows:

\begin{verbatim}
Input: $R_{yy}$, the matrix; $v_0$, an initial guess of an eigenvector such that $\|v_0\| = 1$

For $k = 1, 2, 3, ..., do$

$w \leftarrow R_{yy}v_{k-1}$

$v_k \leftarrow w/\|w\|$\hspace{1cm} $\lambda_k \leftarrow v_k^T R_{yy}v_k$

End for

Output: $\lambda_k$, the approximation of the maximum eigenvalue of $R_{yy}$ after the $k^{th}$ iteration.
\end{verbatim}

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

$$R_{yy}x = \lambda_i x \Rightarrow R_{yy}^{-1}R_{yy} \left(\frac{1}{\lambda_i}x\right) = R_{yy}^{-1}x$$

$$\Rightarrow \frac{1}{\lambda_i}x = R_{yy}^{-1}x$$

Hence, by definition of eigenvalue, all $1/\lambda_i$ are eigenvalues of $R_{yy}^{-1}$. 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 $O(M^4L^3)$ operations using naive Gaussian elimination. A common implementation of the inverse power iteration uses LU decomposition, which has a lower complexity of $O(4/3M^4L^3)$. Hence, the overall computational complexity of the inverse power method is $O(M^4L^4)$, 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. $R_{yy} = CC^T$, where $C$ is an upper or lower triangular matrix. For a general Hermitian positive definite matrix, this process takes $O(M^3L^2/3)$ 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 $O(M^4L^4)$ 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 $T$ fulfills the following relationship:

$$T - ZT^T = D$$  \hspace{1cm} (15)

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

$$D = G \Sigma G^T$$  \hspace{1cm} (16)

where

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad G = [u \ v]$$  \hspace{1cm} (17)

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

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

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 $R_{yy}$ be a non-singular $m \times m$ matrix i.e. $\lambda_i$ are non-zero for all $1 \leq i \leq m$, then $R_{yy}^{-1}$ has eigenvalues $1/\lambda_i$ for all $1 \leq i \leq m$ [14], [16].
Assume we have $G = [u \ v]$, we can make the first element of $v$ to be 0 by applying a hyperbolic transformation $H$ on $G$ such that:

$$H_i G = [\tilde{u}, \tilde{v}]$$  \hspace{1cm} (18)

Where $\tilde{u} = [\tilde{u}_1, \tilde{u}_2, ..., \tilde{u}_n]^T$ and $\tilde{v} = [0, \tilde{v}_1, \tilde{v}_2, ..., \tilde{v}_n]^T$

With $H_i$ satisfying the relation $H_i^T \Sigma H_i = \Sigma$. Hence, the transformed generator $H_i G$ is also a generator for $T$:

$$(H_i G)^T \Sigma (H_i G) = G \Sigma G^T$$  \hspace{1cm} (19)

The transformed vector $\tilde{u}$ is the first column of the Cholesky factor of $T$. Also introduce $T_5 = T_5 - \tilde{u}^T \tilde{u}$ to be the Schur complement of $T$. Then, we observe that applying the shift matrix $Z$ to the generator after the hyperbolic transformation yields

$$Z \tilde{u} \tilde{u}^T Z^T - \tilde{v} \tilde{v}^T = T_5 - ZT_5 Z^T$$  \hspace{1cm} (19)

which actually generates the Schur complement. As a result, applying further hyperbolic transformations on expression (19) introduces zero to the first element of $\tilde{v}$. This is the second column of the Cholesky factor of $T$. Successive iterations of the above process yield the Cholesky factor of the Toeplitz matrix. The positivity of the Schur complement $S_T$ at each stage ensures that the hyperbolic transformation matrix $H_i$ 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 $L = \{4, 8 \text{ and } 16\}$ 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 Vehicular A 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.
using the 

\[ O((ML)^2) \] 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 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 \( O((ML)(\log, ML)^2) \).

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 Teknillan 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

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


### TABLE II. COMPUTATIONAL COMPLEXITY OF POWER ITERATION BASED SPECTRUM SENSING

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Trad. Alg.</td>
<td>Alg. 1 (max. eig. / min. eig.)</td>
<td>MLN</td>
<td>$O(M^3L^3)$</td>
<td>$MLN + O(M^3L^3)$</td>
</tr>
<tr>
<td></td>
<td>Alg. 2 (average / min. eig.)</td>
<td>MLN</td>
<td>$O(M^3L^3)$</td>
<td>$MLN + O(M^3L^3)$ + $MN$</td>
</tr>
<tr>
<td>Prop. Alg.</td>
<td>Alg. 1 (max. eig. / min. eig.)</td>
<td>Cholesky factorization</td>
<td>$O(kML)$</td>
<td>$O(M^3L^3 / 3)$</td>
</tr>
<tr>
<td></td>
<td>Schur Algorithm</td>
<td>MLN</td>
<td>$O(M^2L^2)$</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Alg. 2 (average / min. eig.)</td>
<td>Cholesky factorization</td>
<td>$O(M^3L^3 / 3)$</td>
<td>MLN</td>
</tr>
<tr>
<td></td>
<td>Schur Algorithm</td>
<td>MLN</td>
<td>$O(M^2L^2)$</td>
<td>MN</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>ALGORITHMS</th>
<th>Smoothing Factor (L)</th>
<th>Number of samples (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$10^4$</td>
<td>$5 \times 10^4$</td>
</tr>
<tr>
<td>Trad. Alg.</td>
<td>Alg. 1 (max. eig. / min. eig.)</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Alg. 2 (average / min. eig.)</td>
<td>8</td>
</tr>
<tr>
<td>Prop. Alg.</td>
<td>Cholesky</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Schur</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Cholesky</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Schur</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Cholesky</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Schur</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Cholesky</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Schur</td>
<td>16</td>
</tr>
</tbody>
</table>
### Table IV. Some Numerical Values of Computational Complexities for Sensing Methods (M=2x-Oversampled)

<table>
<thead>
<tr>
<th>ALGORITHMS</th>
<th>Smoothing Factor (L)</th>
<th>Number of samples (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$10^4$</td>
</tr>
<tr>
<td>Trad. Alg.</td>
<td>Alg. 1 (max. eig. / min. eig.)</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>64 768</td>
</tr>
<tr>
<td></td>
<td>Alg. 2 (average / min. eig.)</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>66 768</td>
</tr>
<tr>
<td>Prop. Alg.</td>
<td>Alg. 1 (max. eig. / min. eig.)</td>
<td>Cholesky</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>46 123</td>
</tr>
<tr>
<td></td>
<td>Schur</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>36 224</td>
</tr>
<tr>
<td></td>
<td>Alg. 2 (average / min. eig.)</td>
<td>Cholesky</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>44 923</td>
</tr>
<tr>
<td></td>
<td>Schur</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>35 024</td>
</tr>
</tbody>
</table>

### Table V. Some Numerical Values of Computational Complexities for Sensing Methods (M=4x-Oversampled)

<table>
<thead>
<tr>
<th>ALGORITHMS</th>
<th>Smoothing Factor (L)</th>
<th>Number of samples</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$10^4$</td>
</tr>
<tr>
<td>Trad. Alg.</td>
<td>Alg. 1 (max. eig. / min. eig.)</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>326 144</td>
</tr>
<tr>
<td></td>
<td>Alg. 2 (average / min. eig.)</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>330 144</td>
</tr>
<tr>
<td>Prop. Alg.</td>
<td>Alg. 1 (max. eig. / min. eig.)</td>
<td>Cholesky</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>157 780</td>
</tr>
<tr>
<td></td>
<td>Schur</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>74 496</td>
</tr>
<tr>
<td></td>
<td>Alg. 2 (average / min. eig.)</td>
<td>Cholesky</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>155 380</td>
</tr>
<tr>
<td></td>
<td>Schur</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>72 096</td>
</tr>
</tbody>
</table>