An Alternative Implementation of a Cyclostationary Detector

Francisco Paisana\textsuperscript{1,2}, Neeli Prasad\textsuperscript{1}, António Rodrigues\textsuperscript{2}, Ramjee Prasad\textsuperscript{1}

\textsuperscript{1} Center for TeleInFrastruktur
Aalborg, University, Denmark
Email: \{np, prasad\}@es.aau.dk

\textsuperscript{2} Instituto de Telecomunicações/IST, Technical University of Lisbon, Portugal
Email: francisco.paisana@ist.utl.pt, ar@lx.it.pt

Abstract—Spectrum sensing is the ability on which Cognitive Radio relies to obtain spectrum awareness. In this paper, an experimental study of different time domain cyclostationary detection architectures is made for the special case when the modulation scheme of the transmitted signal is OFDM. The detectors were implemented using GNU Radio as software toolkit for signal processing and two USRP2 as RF front-ends. An alternative implementation of the state of the art cyclostationary detector is proposed for its benefits in terms of complexity, increased performance and flexibility.

Keywords—cognitive radio; spectrum sensing; cyclostationary; OFDM; Goertzel Algorithm; GNU Radio; USRP2.

I. INTRODUCTION

The high data rate demands of modern wireless applications have led to a problem of scarcity of spectrum [1]. According to the Federal Communication Commission (FCC), the current static frequency allocation schemes leave approximately 70\% of the allocated spectrum underutilized [2]. In order to increase the spectrum usage efficiency, Cognitive Radio (CR) was proposed, suggesting the coexistence and share of resources between primary users (PUs) and secondary users (SUs). A CR must be able to obtain information about its surroundings and, by adjusting its radio operating parameters, to access dynamically the unused spectrum without causing any harmful interference to the PUs.

Local Spectrum sensing (SS) is one of the techniques a CR can use to obtain spectral awareness. It simply requires the analysis and processing of the signals transmitted by PUs during their normal operation. In spite of the low infrastructure requirements and broad application areas, the main challenge of spectrum sensing is that CRs must be able to detect PU signals at very low SNRs.

There are several spectrum sensing techniques being Energy Detection (ED), Matched Filtering (MF) and Feature detection (FD) the most popular ones. Energy Detection is one of the most basic sensing schemes and it is optimal if both the signal and the noise are Gaussian. However, it is incapable of distinguishing interference from noise and its performance degrades rapidly when the noise power is not perfectly known. The MF, in spite of having optimal sensing results, requires too much knowledge about the transmitter and becomes overly complex as the number of PU’s signal types increases. Feature detection has better performance than energy detection under low SNRs and is robust to noise uncertainty, however, it requires prior information of the transmitted signal and its complexity and sensing time can sometimes become prohibitive. The ED and FD are good choices for hybrid architectures, as suggested in [3], where the ED can be used for fast coarse sensing followed by a FD for fine sensing.

Much of the recent work in CR has been focused on the detection of Orthogonal Frequency Division Multiplexing (OFDM) signals [4] which is a key technology in modern communication systems such as DVB-T, WLAN, WiMAX and LTE. The cyclic prefix provides important periodic features to OFDM signals that can be used by FDs such as cyclostationary detectors (CD) or autocorrelation-based detectors (AD) [5]. Thus, this work will focus mainly in this type of modulation, although, in the case of CD, it can be easily extendable to other types of signals.

In this study, an alternative cyclostationary detector is described and tested for OFDM signals in particular. The detector was implemented in GNU Radio and its performance was evaluated using USRP2 as RF front-ends. Some comparisons were made with the cyclostationary architecture presented in literature [6].

The paper is organized as follows: Section II presents the theoretical models and architectures of the traditional time-domain cyclostationary detector (TCD) and the alternative architecture proposed (ACD). In sector III, the hardware and software used and the considerations made during the experimental study are described. In section IV, the experimental results are illustrated and compared for both algorithms. Finally, in section V, a conclusion is given and the future work, proposed.

II. THEORETICAL MODELS

A. Principles of Cyclostationarity

Any communication signals exhibit underlying periodicities in their signal structures added by modulation, preambles, pilots and cyclic prefixes for synchronization and signaling purposes. As a result, these signals can be modeled as cyclostationary processes since their mean and autocorrelation are periodic. Such inherent cyclic features can
be used to distinguish primary user signals from AWGN which, by definition, is a stationary process.

A signal \( x[n] \) is considered second order cyclostationary if its time varying autocorrelation function, defined as

\[
R_{xx}[n,l] = E[x[n], x^*[n + l]]
\]

is periodic over \( n \) for a lag parameter \( l (l=\pm 1, \pm 2, \ldots) \). So it can be represented as a Fourier series

\[
R_{xx}[n,l] = \sum_{k=\infty}^{+\infty} R_{xx}^k[l]e^{\frac{2\pi nk}{N}}
\]

(2)

where the sum is taken over integer multiples of fundamental cyclic frequency \( \alpha_k \) (for \( k =0, \pm 1, \pm 2, \ldots \)). The “harmonics” of the Fourier series define the cyclic autocorrelation function (CAF)

\[
R_{xx}^k[l] = \frac{1}{N} \sum_{n=0}^{N-1} x[n]x^*[n + l]e^{-j\frac{2\pi nk}{N}}.
\]

(3)

The Fourier transform of \( R_{xx}^k[l] \) is called the cyclic spectrum (CS) or spectral correlation function (SCF) and has the following equation:

\[
S_{xx}^k[f] = \frac{1}{N} \sum_{n=0}^{N-1} R_{xx}^k[l]e^{-j2\pi nf}.
\]

(4)

When \( \alpha=0 \), CS takes the form of the power spectral density \( S_{xx}[f] \) (PSD) and the CAF takes the form of the autocorrelation function in (1). Under \( H_0 \), since AWGN signals are stationary processes (without cyclostationary features), \( S_{xx}^0[f] = 0 \) and \( R_{xx}^k[l] = 0 \) for any \( \alpha \neq 0 \). Under \( H_1 \), on the other hand, \( R_{xx}^k[l] \neq 0 \) for \( \alpha_k \in \mathcal{A} \), where \( \mathcal{A} \) is the set of cyclic frequencies of \( x[n] \).

### B. Traditional Cyclostationary Detector

Like the ED, the CD can be implemented both in time domain and in frequency domain. In the case of OFDM signals, the correlation created by the cyclic prefix (CP) is used to distinguish primary user signals from AWGN.

The test statistic, based on GLRT, for the detection of signals with a cyclic prefix of \( N_c \) samples and an FFT with size \( N_s \) peaks at \( l=\pm N_c \) for the cyclic frequencies

\[
\mathcal{A} = \left\{ \alpha_k = \frac{k}{(N_d + N_c)T_0}, k = 0, \pm 1, \pm 2, \cdots \right\}
\]

(12)

where \( \#\mathcal{A} = N_d \). The \( R_{xx}^\alpha[l] = N_d \) plot is displayed in Fig. 1. It can be seen that the cyclic frequencies of the OFDM signal are much lower than the range of \( \alpha \) for which the CAF was measured. In [6], the author suggests the usage of decimation before the calculation of the CAF. The main purposes are to control the detection time without altering the size of the FFT employed and to reduce the power consumption by decreasing the sampling rate. A good estimate of the necessary time for the TCD to detect the presence of a signal for each lag parameter would be

\[
\Delta t_{est} \approx T_0 \left( N_{fft}M_{decim} + N_d \right)
\]

(13)

where \( T_0 \) is sampling period and \( M_{decim} \) the decimation factor.

\[
C_{12} = \frac{1}{N} \sum_{k=0}^{N_{fft}-1} \text{Re}\{R_{xx}^\alpha\} \text{Im}\{R_{xx}^\alpha\}
\]

(9)

To improve the detection performance, in [8], test statistics that use several cyclic frequencies were proposed. One that stood out for its simplicity and good results was

\[
T_{sum} = \sum_{\alpha \in \mathcal{A}} T_{\alpha k}.
\]

(10)

Under \( H_0 \), having \( R_{xx}^\alpha[l] \) a zero mean Gaussian distribution, the test statistic \( T_{\alpha k} \) follows a \( \chi^2 \) distribution. \( T_{sum} \), being a sum of approximately independent \( \chi^2 \) distribution variables, has, in turn, a \( \chi^2 N_d \) distribution, where \( N_d \) is the number of cyclic frequencies used in the test (10).

The threshold, for a specific false alarm probability \( P_{fa} \), is derived from the following equation,

\[
F_{\chi^2 N_d} (y_{sum}) > 1 - P_{fa}.
\]

(11)

According to the analysis made in [4], the CAF of OFDM signals with a cyclic prefix of \( N_c \) samples and an FFT with size \( N_s \) peaks has an eigenvalue distribution variables, \( \chi^2 \).

\[
\mathcal{A} = \left\{ \alpha_k = \frac{k}{(N_d + N_c)T_0}, k = 0, \pm 1, \pm 2, \cdots \right\}
\]

(12)

where \( \#\mathcal{A} = N_d \). The \( R_{xx}^\alpha[l] = N_d \) plot is displayed in Fig. 1. It can be seen that the cyclic frequencies of the OFDM signal are much lower than the range of \( \alpha \) for which the CAF was measured. In [6], the author suggests the usage of decimation before the calculation of the CAF. The main purposes are to control the detection time without altering the size of the FFT employed and to reduce the power consumption by decreasing the sampling rate. A good estimate of the necessary time for the TCD to detect the presence of a signal for each lag parameter would be

\[
\Delta t_{est} \approx T_0 \left( N_{fft}M_{decim} + N_d \right)
\]

(13)

where \( T_0 \) is sampling period and \( M_{decim} \) the decimation factor.
The flow graph for this CD is displayed in Fig. 2. The autocorrelation block measures the product $x[n]x^*[n+l]$ for a specified lag $l$ from the set $L$. This requires a RAM memory with size equal to the highest lag value in $L$ which can go from 64 in WLAN up to 8192 in the case of DVB-T signals, for example. The decimation factor $M_{\text{decim}}$ of the decimator block is adjusted according to the time available for detection. In order to obtain a CAF with enough spectral resolution, a FFT block of large size ($N_{\text{in}}$) is needed. However, this will also dramatically increase the memory requirements of the detector. The test statistic block only involves measuring the elements of $\tilde{\Sigma}_{xx}^k$, the test statistics for each $\alpha_k$ $T_{\text{sum}}[l]$ and their sum $T_{\text{sum}}$.

This detector shows great robustness to noise uncertainty and frequency offset. However, it requires previous knowledge of the lag and cyclic frequencies of the PU signals to be able to detect them in real time. In the case of OFDM signals, this corresponds to knowing the $N_p$ and $N_q$ values. If the test statistic is made for several lag parameters (for example, for detecting both 2k and 8k DVB-T signal modes), in order not to increase significantly the detection time, the block diagram from Fig. 2 must be replicated for each different $l \in L$. Such drawback emphasizes the importance of reducing Fig. 2 flowgraph hardware requirements.

C. Alternative Cyclostationary Detector

With this alternative cyclostationary detector (ACD) architecture, it is intended to overcome the limitations of the TCD described before by reducing its hardware requirements. Improvements in detection performance can be also obtained as long as the CAF is measured using a DFT size which is a multiple of the cyclic frequencies of interest. This last requirement isn’t difficult to be met since the new ACD architecture uses separate single frequency DFT blocks instead of a power of two FFT. The FFT already incorporated in the wireless communication devices that use OFDM will, then, remain available for other such operations as the modulation/demodulation of the transmitted/received signals or for other detection algorithms like ED to run in parallel.

Considering the noise AWGN, for low SNRs, the following approximations can be made for the elements of the $\tilde{\Sigma}_{xx}^k$ in (8):

$$C_{12} \approx 0$$

$$\tilde{C} = \frac{1}{2N} \sum_{k=0}^{N_{\text{fft}}-1} |\tilde{R}_{xx}^k|^2 \approx C_{11} \approx C_{22}. \quad (14)$$

According to Parseval’s theorem we have

$$\tilde{C} = \frac{1}{2N} \sum_{k=0}^{N_{\text{fft}}-1} |\tilde{R}_{xx}^k|^2 = \frac{1}{2} \sum_{n=0}^{N_{\text{fft}}-1} |x[n], x^*[n+l]|^2. \quad (15)$$

Thus, $\tilde{C}$ can be measured in time domain, and the $\tilde{R}_{xx}^k[ll]$ for $\alpha_k \in \mathcal{A}$, can be measured using the Goertzel Algorithm (GA) which is more efficient than the FFT when the number of frequencies tested $N_p$ is small. The resulting test statistic can, then, be rewritten as follows,

$$T_{\text{sum}} = \frac{\tilde{C}}{2} \sum_{\alpha_k \in \mathcal{A}} |\tilde{R}_{xx}^k[l]|^2. \quad (17)$$

Some information was, however, lost about $C_{12}$ and the difference between the elements $C_{11}$ and $C_{22}$ values of the covariance matrix which are different than zero as SNR increases. This information can be gathered from the GA DFTs $\tilde{R}_{xx}^k[l], \alpha_k \in \mathcal{A}$. In fact, in absence of noise, $\tilde{R}_{xx}^k[l] = 0, \alpha_k \notin \mathcal{A}$ and $\tilde{R}_{xx}^k[l] \neq 0, \alpha_k \in \mathcal{A}$. As a result, $C_{12}$ and the difference between $C_{11}$ and $C_{22}$ ($\Delta = C_{11} - C_{22}$) can be estimated using the following equations,

$$\tilde{C}_{12} = \frac{1}{N_{\text{fft}}} \sum_{\alpha_k \notin \mathcal{A}} \text{Re}\{\tilde{R}_{xx}^k\}. \text{Im}\{\tilde{R}_{xx}^k\} \quad (18)$$

$$\Delta \approx \frac{1}{N_{\text{fft}}} \sum_{\alpha_k \in \mathcal{A}} \left(\text{Re}\{\tilde{R}_{xx}^k\}^2 - \text{Im}\{\tilde{R}_{xx}^k\}^2\right). \quad (19)$$

The resulting covariance matrix becomes

$$\tilde{\Sigma}_{xx}^k = \begin{bmatrix} \tilde{C} + \Delta/2 & \tilde{C}_{12} \\ \tilde{C}_{12} & \tilde{C} - \Delta/2 \end{bmatrix}. \quad (20)$$

The approximation in (17) is usually sufficient but both (17), called ACD1, and (20), called ACD2, performances will be tested in section V.

The flow graph of the ACD is shown in Fig. 3. The main novelty is the complete removal of the FFT and the insertion of an IIR filter bank formed by $N_p$ elementary Goertzel filters to measure the CAF for $\alpha_k \in \mathcal{A}$ and a $1 \times 1$ block to measure the $\tilde{C}$ parameter as displayed in (16). Another difference was the switch of the variable M decimator block by a fixed M CIC filter. Since the GA provides total freedom in the choice of the DFT size, it can be now used to adjust the detection time. Furthermore, the GA complexity is equal or lower than a FIR filter so, the usage of a high order decimator, which would include FIR filters with a high number of taps, wouldn’t reduce the power consumption of the sensing device. The ACD1 or ACD2 test statistics measurement blocks didn’t suffer any relevant change in complexity when compared with the TCD.

The several Goertzel IIR filters that form the IIR Filter bank have the structure shown in Fig. 4, each one corresponding to a different $\alpha_k \in \mathcal{A}$. The feedback part only involves sums and a real/complex multiplication whereas the forward part only needs to be computed for the last cycle. For the special case $\alpha_k = 0$, the complexity decreases even more since no multipliers are required. The number of multiplications, compared to the generic FFT, is reduced if $N_p < 5 \log_2 (N_{\text{fft}})/6$. However, the greatest advantage of the GA results from the low memory it requires due to the fact it doesn’t use approximately $N_{\text{fft}}$ shift registers to store variables in intermediate steps and doesn’t require a large table of pre-computed sines and cosines. Another advantage of the GA is the fact it can be used for a $N_{fft}$ which is not a power of 2. If the $N_{fft}$ used is a multiple of the cyclic frequency bins analyzed $\alpha_k \in \mathcal{A}$, the scalloping loss effect of the DFT can be dramatically reduced increasing, consequently, the performance of the detector.
III. IMPLEMENTATION

A. Testbed Description

The experimental study was performed in the Real Network module of the S-Cogito Testbed [9], using two USRP2 as RF front-ends for the transmitter and receiver. The USRP2, developed by Ettus Research, LLC, is a hardware platform used in software defined radio. It contains a motherboard that provides the basic components for baseband signal processing such as ADCs, DACs and clock generation and a daughterboard which is responsible for analog operations such as filtering or up/down conversion. The USRP2 daughterboard model used was the XCVR24250 which works both in the 2.4 and 5 GHz bands. The communication between the USRP2 and the computer is made through a Gigabit Ethernet port.

The software used to configure the USRP2 and for baseband processing was GNU Radio version 3.5.2. In this framework, the signal processing blocks are developed in C++ and the interconnections are made using Python language.

The detectors’ performance was tested in an indoor space with both the USRP2 transmitter and receiver in line of sight. A center frequency and sampling rate of 5.5 MHz and 1 MS/s respectively were used by both boards. The transmitter’s signal was an OFDM/BPSK modulated pseudorandom bit sequence with a bandwidth of 500 kHz in the baseband. The OFDM FFT size was N_d=64 with 52 occupied subcarriers and a guard interval composed by N_g=16 samples at 1 MS/s.

In order to test the detector’s performance for different SNRs, the transmitted signal amplitude was adjusted by a numerical amplifier. For each SNR value, 2500 test statistics measured by the flow graphs displayed in Fig. 2 and Fig. 3 in GNU Radio were stored in a file and imported to Matlab for further processing and plotting.

The number of samples used for the estimation of noise power was 82,000,000. This value was, then used as reference for estimating the SNR of the received signal. The tests were made for 12 different SNR values whose range goes approximately from -25 to 0 dB.

B. Detector Architecture

For both the ACD and TCD, during the experimental study, the lag parameter used was l=N_d=64 and the DFT/FFT bins analyzed were

$$\mathcal{A} = \left\{ \alpha_k = \frac{k M_{\text{decim}} N_{\text{fft}}}{(N_d + N_g)}, k = 0, \pm 1, \pm 2 \right\}. \quad (21)$$

For the TCD, a decimation factor of M_{\text{decim}}=16 and a FFT size of 2048 were employed. Therefore, the resulting number of samples per test statistic was N=32832. In the case of the ACD, the CIC decimation factor was M_{\text{decim}}=2 and two different FFT sizes were tested. The first FFT size was N_{\text{fft}}=16384 which allowed the comparison of the ACD with the TCD when the number of samples per test statistic is the same (N=32832). The second FFT size was, on the other hand, N_{\text{fft}}=16360. The purpose of using this value was to show the ACD superior performance when the DFT size is a multiple of the bins analyzed (\alpha_k \in \mathcal{A}). In TABLE I and TABLE II, the hardware requirements of both the TCD, implemented in [6], and the ACD1/ACD2 are shown. As it can be seen, the proposed architecture has clear advantages, especially in terms of (real) memory usage. The number of real multipliers and real adders were also significantly reduced because the ACD controls the detection time by altering the DFT size and not through a high order variable M decimator as in the TCD case.

<table>
<thead>
<tr>
<th>Stages</th>
<th>Operations</th>
<th>Multipliers</th>
<th>Adders</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-cor.</td>
<td></td>
<td>4</td>
<td>2</td>
<td>2N_d = 128</td>
</tr>
<tr>
<td>Decimator</td>
<td></td>
<td>40</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>FFT</td>
<td></td>
<td>4(</td>
<td>\log_2 N_d</td>
<td>- 1) = 20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\bar{\sigma}_n^2$ Estimation</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$T_{\text{sum}}$</td>
<td></td>
<td>6</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>73</td>
<td>152</td>
<td>4308</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stages</th>
<th>Operations</th>
<th>Multipliers</th>
<th>Adders</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-cor.</td>
<td></td>
<td>4</td>
<td>2</td>
<td>2N_d = 128</td>
</tr>
<tr>
<td>CIC decimator</td>
<td></td>
<td>0</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>IIR Filter Bank</td>
<td></td>
<td>6(N_d - 1) = 24</td>
<td>6(N_d - 1) + 2 = 34</td>
<td>4(N_d - 1) + 2 = 18</td>
</tr>
<tr>
<td>$\langle</td>
<td>.</td>
<td>^2 \rangle$ &gt;</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>$T_{\text{sum}}$ (ACD1/ACD2)</td>
<td></td>
<td>1/6</td>
<td>1/9</td>
<td>1/3</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>31/36</td>
<td>50/59</td>
<td>160/163</td>
</tr>
</tbody>
</table>

IV. SIMULATIONS/EXPERIMENTAL RESULTS

Fig. 5 shows the empirical relative error of the ACD1 (see Eq. (17)) and the ACD2 (see Eq. 21) test statistics when
compared to the TCD. The (17) approximation has the largest relative error but its test statistic becomes greater than TCD’s as the SNR value increases. So its performance is slightly better for high SNRs. The ACD2 is almost identical to the TCD in the case the cyclic frequencies of the received signal are completely known. If this requirement can’t be met, its curve will tend to the ACD’s curve. The reason for the ACD1 better performance is the fact that (17) is independent of the $\phi$ phase angle which can be altered by frequency offset.

In Fig. 6, the variation of the probability of detection with SNR is shown. The experimental curves of the TCD and ACDs are overlapping for the same number of samples N, so it can be inferred that their performance is roughly the same. The ACD curve for N=32784 ($N_{\text{dft}}=16360$), on the other hand, in spite of using a lower number of samples, shows better performance since 16360 is a multiple of the bins analyzed ($\alpha_{\text{dft}} \in \mathcal{A}$). In Fig. 6, it can also be seen that the theoretical values, in dash line, match the empirical ones.

V. CONCLUSIONS AND FUTURE WORK

In this paper, an alternative implementation of the time-domain cyclostationary detector was described. Its performance was measured experimentally using USRP2 and GNU Radio and compared to the traditional cyclostationary detector. The proposed method has better performance and lower hardware requirements, showing greater scalability when the number of lag parameters tested increases.

As future work, this alternative implementation could be tested for other types of modulation schemes such as DS-SS and additional studies could be made to test its robustness to frequency offset.

ACKNOWLEDGEMENTS

This experimental work was performed in the Real Network module of the S-Cogito Testbed [9]. The software defined radio platform used includes several USRP2 as RF front ends and personal computers (PC) using GNU Radio software toolkit for baseband processing.

VI. REFERENCES