

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

---

Theses, Dissertations, and Student Research from  
Electrical & Computer Engineering

Electrical & Computer Engineering, Department of

---

8-2011

# CYCLOSTATIONARY DETECTION FOR OFDM IN COGNITIVE RADIO SYSTEMS

Marcos E. Castro

University of Nebraska-Lincoln, [marcos.castro@us.bosch.com](mailto:marcos.castro@us.bosch.com)

Follow this and additional works at: <http://digitalcommons.unl.edu/elecengtheses>



Part of the [Signal Processing Commons](#), and the [Systems and Communications Commons](#)

---

Castro, Marcos E., "CYCLOSTATIONARY DETECTION FOR OFDM IN COGNITIVE RADIO SYSTEMS" (2011). *Theses, Dissertations, and Student Research from Electrical & Computer Engineering*. 21.

<http://digitalcommons.unl.edu/elecengtheses/21>

This Article is brought to you for free and open access by the Electrical & Computer Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Theses, Dissertations, and Student Research from Electrical & Computer Engineering by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

CYCLOSTATIONARY DETECTION FOR OFDM IN  
COGNITIVE RADIO SYSTEMS

by

Marcos E. Castro

A THESIS

Presented to the Faculty of  
The Graduate College at the University of Nebraska  
In Partial Fulfillment of Requirements  
for the Degree of Master of Science

Major: Electrical Engineering

Under the Supervision of Professor M. Cenk Gursoy

Lincoln, Nebraska

August, 2011

# CYCLOSTATIONARY DETECTION FOR OFDM OVER COGNITIVE RADIO SYSTEMS

Marcos E. Castro M.S.

University of Nebraska 2011

Advisor: M. Cenk Gursoy

Research on cognitive radio systems has attracted much interest in the last 10 years. Cognitive radio is born as a paradigm and since then the idea has seen contribution from technical disciplines under different conceptual layers. Since then improvements on processing capabilities have supported the current achievements and even made possible to move some of them from the research arena to markets.

Cognitive radio implies a revolution that is even asking for changes in current business models, changes at the infrastructure levels, changes in legislation and requiring state of the art technology.

Spectrum sensing is maybe the most important part of the cognitive radio system since it is the block designed to detect signal presence on the air.

This thesis investigates what cognitive radio systems require, focusing on the spectrum sensing device. Two voice applications running under different Orthogonal Frequency Division Multiplexing (OFDM) schemes are chosen. These are WiFi and Wireless Microphone. Then, a Cyclostationary Spectrum Sensing technique is studied and applied to define a device capable of detecting OFDM signals in a noisy environment. One of the most interesting methodologies, in terms of complexity and computational requirements, known as FAM is developed. Study of the performance and frequency synchronization results are shown, including the development of a blind synchronization technique for offset estimation.

*A mi hijo Gabriel, la motivación de todos mis objetivos*

*A mi esposa Norka, el soporte de todos mis logros*

*To my son Gabriel, the motivation of all my objectives*

*To my wife Norka, the support of all my achievements*

# Contents

<b>1</b>	<b>Cognitive Radio and Software Defined Radio</b>	<b>1</b>
	1.1 Cognitive Radio (CR) .....	1
	1.2 Software Defined Radio (SDR) .....	4
	1.3 Cognitive Radio applications .....	4
	1.4 SDR design .....	5
	1.5 Topologies .....	6
	1.6 The RF Chain .....	8
<b>2</b>	<b>Spectrum Sensing Techniques</b>	<b>12</b>
	2.1 Spectrum Sensing .....	12
	2.2 Energy Detection .....	13
	2.3 Matched Filter .....	14
	2.4 Cyclostationary Feature Detection .....	14
	2.5 Cooperative spectrum Sensing .....	15
	2.6 Other Methods .....	16
<b>3</b>	<b>Orthogonal Frequency Division Multiplexing OFDM</b>	<b>17</b>
	3.1 OFDM in Cognitive Radio Systems .....	17
	3.1.1. Advantage and Disadvantages .....	18
	3.2 Definition .....	19
	3.3 The FFT/IFFT in OFDM .....	21
	3.4 OFDM frame structure .....	21
	3.4.1. Guard Interval and Cyclic Extension .....	22
	3.4.2. Pilots .....	22

	3.4.3. Preambles .....	23
	3.4.4. OFDM Symbols .....	24
	3.5 OFDM Implementation .....	25
	3.5.1 Transmitter .....	26
	3.5.2 Receiver .....	29
<b>4</b>	<b>Cyclostationary Detection</b>	<b>34</b>
	4.1 Cyclostationary Theory .....	34
	4.2 Cyclic Autocorrelation Function .....	36
	4.3 The spectral-correlation function .....	37
	4.4 Cyclic Spectrum Estimation .....	38
	4.5 FAM Implementation .....	40
	4.6 Applying FAM for OFDM signals.....	47
	4.7 FAM test bench .....	51
	4.8 FAM Performance Analysis .....	53
	4.8.1. Average size, number of data frames vs. SNR .....	56
	4.9 Narrow the Target .....	59
	4.10 A voice application .....	62
	4.11 FAM Computation .....	67
<b>5</b>	<b>FAM vs. Frequency Offset</b>	<b>69</b>
	5.1 Frequency offset in FAM algorithms .....	69
	5.2 Frequency Offset estimation and correction .....	72
	5.3 How and where to reduce computation .....	76
	5.3.1. The Sliding FFT .....	76
	5.3.2. One-Bit Correlation .....	78

<b>References</b>	<b>79</b>
<b>Appendix A : Matlab Implementation</b>	<b>83</b>
A.1 OFDM Transmitter and Receiver 802.11g .....	83
A.2 FAM implementation .....	98
A.3 Frequency Estimation .....	104

# Chapter 1

## Cognitive Radio and Software Defined Radio

### 1.1 Cognitive Radio (CR)

The cognitive radio is an intelligent wireless communication system that is aware of its surrounding environment and under a certain methodology is able to use the current available spectrum momentarily without interfering with the primary user who paid to be served in that area.

#### **An example**

Imagine a portable radio that is able to communicate to its base which is relatively close. Let's call this pair the secondary system and picture it as a relative local service.

Now assume the system is working at the same spectrum of the cellular phone system, which is the primary system.

Such secondary system should work in a kind of opportunistic way to borrow spectrum without interfering with the primary users or degrading the quality of its service.

Cognitive radio system should be able to scan and sense the spectrum around and find any available spot in frequency to establish its communication, that has to be released once a primary user comes back claiming the spot.

What follows is the definition adopted by the FCC: "A cognitive radio is a radio or system that senses its operational electromagnetic environment and can dynamically and autonomously adjust its radio operating parameters to modify system operation, such as maximize throughput, mitigate interference, facilitate interoperability, access secondary markets"



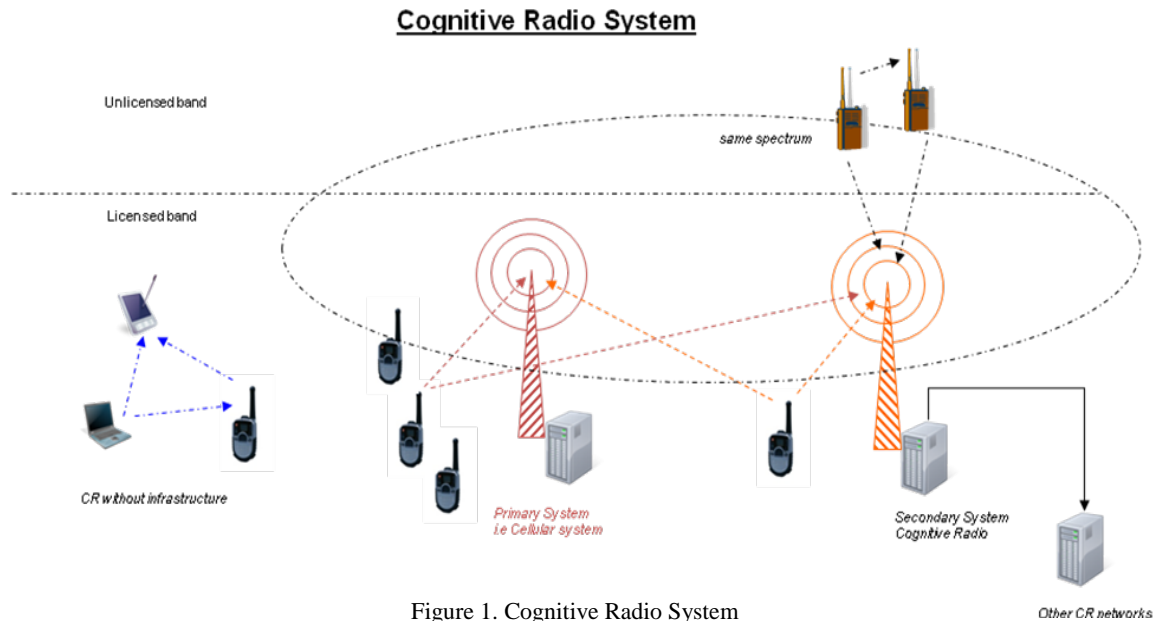


Figure 1. Cognitive Radio System

The spectrum awareness is defined as multi-dimensional and mainly measures signal's presence in the frequency spectrum at a certain time at certain locations. Although other dimensions could be involved as coding or angle dimensions [7]. Spread spectrum signals or frequency hopping could allow new vacant possibilities while angle dimension is made possible to account since the inclusion of smart antennas capable of detecting the arrival direction.

This kind of knowledge and awareness requires a high grade of flexibility and sensing capabilities in the radio architecture that just a software defined radio (SDR) would be able to support.

The CR system has to support a dynamic spectrum allocation (DSA). This capability is a matter that involves technology, standardization and spectrum policy and even requires changes in the business model.

Measurements over the used spectrum at many different geographical locations show that the average occupancy is less than %6. Some regions of the spectrum are more interesting than others due to technical reasons that will support mobile service models.

CR is then moved by these commercial interests and the scarcity of the spectrum. Using intelligent systems DSA could be achieved thanks to digital technology like digital signal processing and faster processors available.

Digital communication systems are more flexible and provide better bandwidth and energy efficiency than the analog counterparts.

Multimedia services require voice, data and even video transfer nowadays and digital radios are suitable for this purpose.

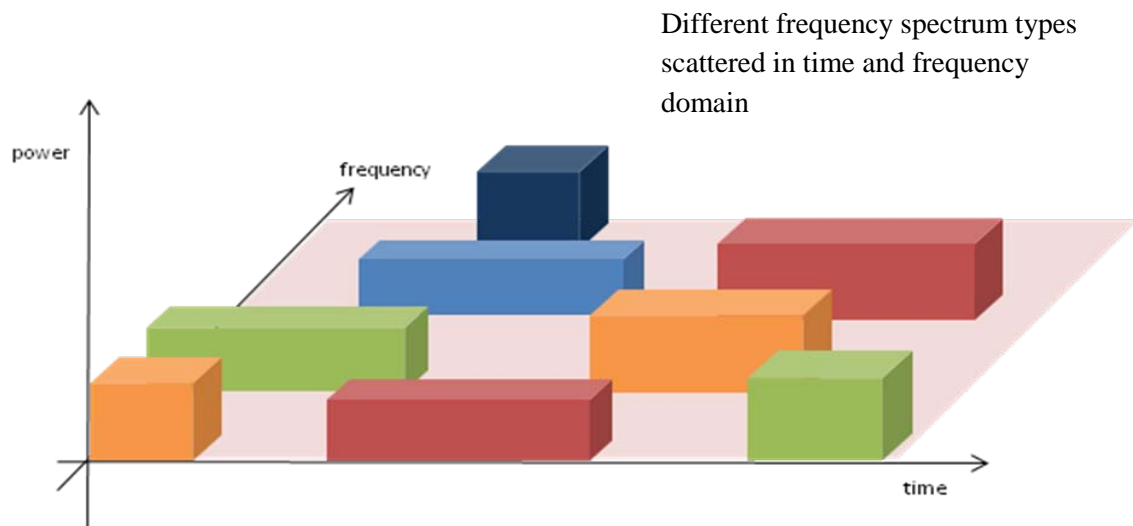


Figure 2. Space frequency, power & time

## 1.2 Software Defined Radio (SDR)

SDR is a radio architecture that shows some considerable grade of flexibility over the fixed hardware radio model. This flexibility is achievable by means of software control changing radio configuration as needed. Digital signal processing techniques applied to SDR bring flexibility and adaptability.

CR technology lies as a layer over the SDR system model that introduces intelligence to the radio system. SDR makes possible foresee the CR paradigm achievable. SDR possesses some grade of flexibility by programmability along the receiver path. At the front end, i.e. some filtering process takes place and these filters could be implemented by analog devices where the parameters that define the bandwidth or frequency allocation could be modified at will by means of digital potentiometers. Change in local oscillators to tune different frequencies by means of digital controls are some examples of flexibility. Further flexibility is achieved when filter processing, modulation and demodulation is done in digital domain allowing to even perform different modulation or encoding schemes using the same hardware.

The grade of flexibility comes with some trade off that makes harder to achieve the CR paradigm and forces to constraints the system to a less broad set of applications.

Then, an exact definition of a SDR is under discussion and there is no defined level of reconfigurability needed to be qualified as SDR. Despite discrepancies a good common definition for SDR concept, is a system that most of the physical layer is defined and altered by software.

## 1.3 Cognitive Radio applications

CR technology is still under development and involves different areas of study. Some research focuses on development of strategies at the receiver itself, like improvements on digital signal processing for spectrum sensing, signal awareness and even better architectures to achieve this.

Other researchers focus in creating modifications to the current wireless networks for a system support that include base stations and central awareness and control.

Many of the results coming from the research efforts are being applied already at some levels into specific applications. Emergency radios are able to get into the spectrum and establish communication in free spots of the frequency domain using its SDR capabilities to adapt to the available spectrum. Wireless microphones market is also planning a regulation that includes cognitive radio capabilities as spectrum sensing to keep policies of usable frequencies.

Examples of this are 802.22, 802.h and 802.11k that intent to apply some CR proven practices to WAN and WLAN arena.

## 1.4 SDR design

The goal of a flexible SDR lies in placing the ADC as close as possible to the antenna. This ideal architecture will allow digital processing of the signal as will. This will also bring some benefits such as avoiding analog devices and reducing noise related with this kind of electronics.

This model will allow modification at the physical layer by software; change in filter characteristics, waveforms, bandwidth response, and even on the fly reconfigurability.

However the ideal model requires fast ADC/DAC and such a requirement with current technology implies high cost, noise and power consumption. Also the processor will require to process high amount of data at faster pace, making it even more expensive and consuming more power. Power constraint is imposed by mobile or portable devices.

Then it is clear to foresee that backing up the ADC/DAC from the antenna will reduce exigencies in speed processing and power requirement. This is done by down converting the signal to intermediate frequencies by means of mixers, PLLs and local oscillators. Some pre and post

filtering is needed for this process and now some analog devices are added and some flexibility is lost.

These added components are the RF front end and its characteristics like non-linearity and dynamic range will affect and limit the SDR performance.

## 1.5 Topologies

The tuned Radio Frequency Receiver (TRF) shown in Fig. 3 consists of an antenna, the bandpass filter which isolates the incoming signal, Low Noise Amplifier (LNA) that amplifies the signal rejecting on band noise, the Automatic Gain Control (AGC) which increases the signal level to be compatible with the Analog to Digital Converter (ADC).

Although this topology shows simplicity, it results highly impractical. Due to Nyquist requirements the ADC has to work at high sampling rates that translate in high consumption and increase the price. Additional filtering and post-processing has to be done in the digital domain by digital signal processing techniques, increasing current and time consumption.

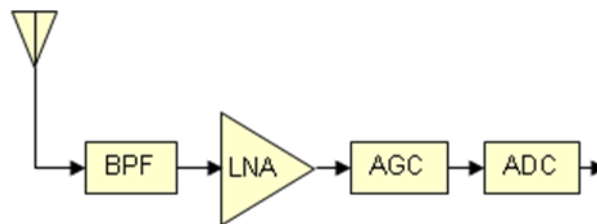


Figure 3. Tuned radio frequency

As depicted in Fig. 4, the single conversion receiver also known as homodyne include a mixer that down convert the incoming signal to a lower central frequency or even base band. This way it requires a lower sampling rate easing the ADC selection. In the case of phase or frequency modulation, one mixer for each branch of I and Q has to be used, since both lower and higher bands contain information.

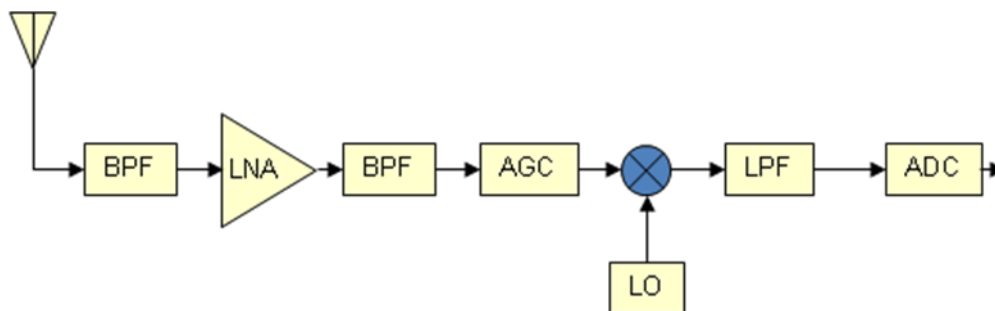


Figure 4. Single conversion receiver AM,BPSK

The local oscillator (LO) makes possible to down convert the signal but is also a new source of noise. Leakage from the LO comes across the inputs that causes the mixer to down convert a received version of itself, called self-mixing. This results in a DC bias that has to be corrected and it could be a hard task to perform. On the other hand, any non-ideal I&Q conversion will result in phase and amplitude errors for the quadrature demodulation.

Some solutions for the problems described above could be addressed in the digital domain by means of compensation and feedback for correction.

It is sometimes more convenient to down sample to an intermediate frequency rather than directly to baseband. This reduces also the sample frequency used in the digital domain and allows better filter designs.

A heterodyne receiver, shown in Fig. 5, down translates the signal spectrum to an intermediate frequency. A super-heterodyne is when the signal is moved to a frequency that is higher than its bandwidth and lower than the central frequency. It is common for this topology to have two intermediate frequency stages. The dual conversion relaxes the filtering requirements. Gain can also be applied by stages and this will reduce the power at the LO and relax also the isolation needs on the mixers. But, a second LO requires more power, probably not a good selection for a mobile application.

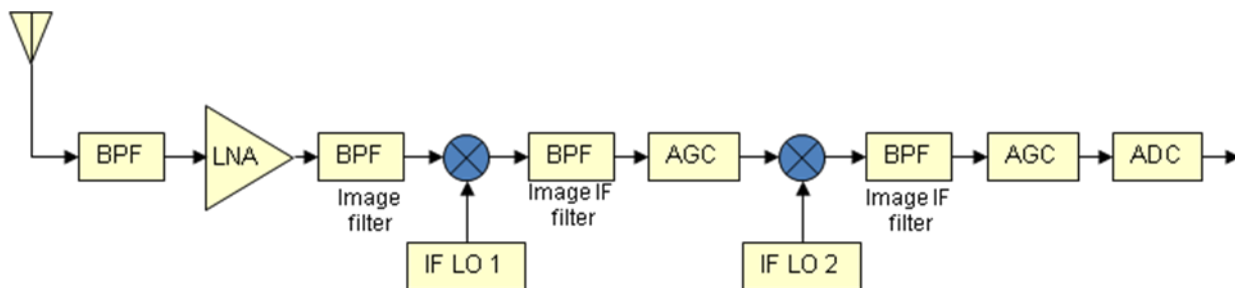


Figure 5. Dual conversion dual super heterodyne receiver

The first image filter is used to reject any interference already existing in the band where one of the images of the signal will be placed. The second filter rejects the upper band.

Single conversion is probably the most convenient topology to be used for a software defined radio. It is probably the best balance between pros and cons assuming that all correction needed in phase, frequency, amplitude, etc. would be achievable in the digital domain, on time, by means of digital signal processing algorithms.

## 1.6 The RF Chain

The front end of a radio system is also understood as the RF chain and its components and design will contribute to the digital base band processing on SDR.

The antenna is the first one at the receiver chain and a good selection will enhance gain and signal quality. Although several antenna forms exist, most of them has a bandwidth of 10% related to the carrier frequency. Thus, for supporting more than one mode at the radio system, more than one antenna is needed. A simple version of a multi antenna system is used for diversity where the signal at both antennas is always under monitor and switching to the best incoming signal in terms of power is used to pass to the baseband processing.

The duplexer and diplexer shown in Fig. 6 are devices that exist in the chain to isolate the transmitter from the receiver that use a common antenna. The duplexer takes place when the transmitter and receiver are in a common frequency range while the diplexer is for the case of different frequency range between the transmitter and receiver.

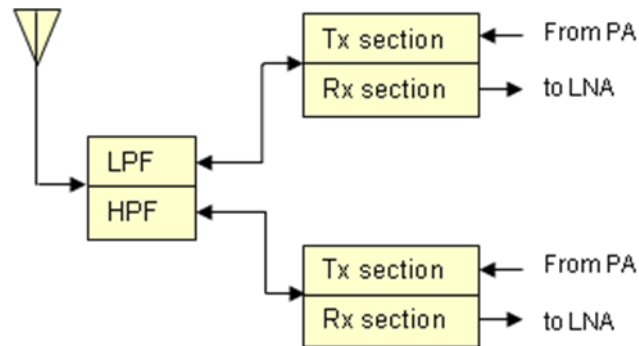


Figure 6. Duplexer and diplexer

The RF filtering is the first filtering performed on the incoming signal. It has to reject out of band signals and noise and provide as much selectivity as possible for the working bandwidth without losses. This bandwidth will define the flexibility of the system to operate at different frequencies under a range of the spectrum.

The low noise amplifier LNA has to boost the signal to a manageable range without adding noise into the signal.

The RF mixer is used to downconvert the incoming signal and can be a source of intermodulation distortions. These are due to a non-linear behavior of the mixer that could be overcome by increasing the LO power to the mixer. But this could be prohibitive in mobile applications that need to save batteries.



The local oscillator will generate the frequency used by the mixer to accomplish the down conversion and good phase stability is required. Phase noise at this part of the chain could be a significant source of interference.

The automatic gain controller (AGC) has the function of maintaining the signal between the ADC range. To get the best of the quantization of the ADC, the signal should be amplified to use most of the range. Clipping on the ADC must be avoided since for amplitude modulated signal as QAM, it will cause damage to the encoded information.

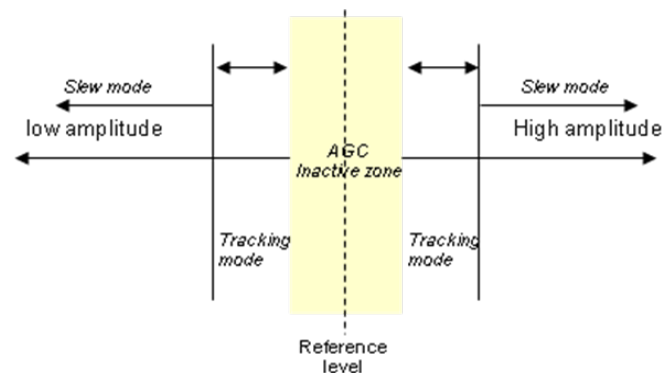


Figure 7. AGC operating modes

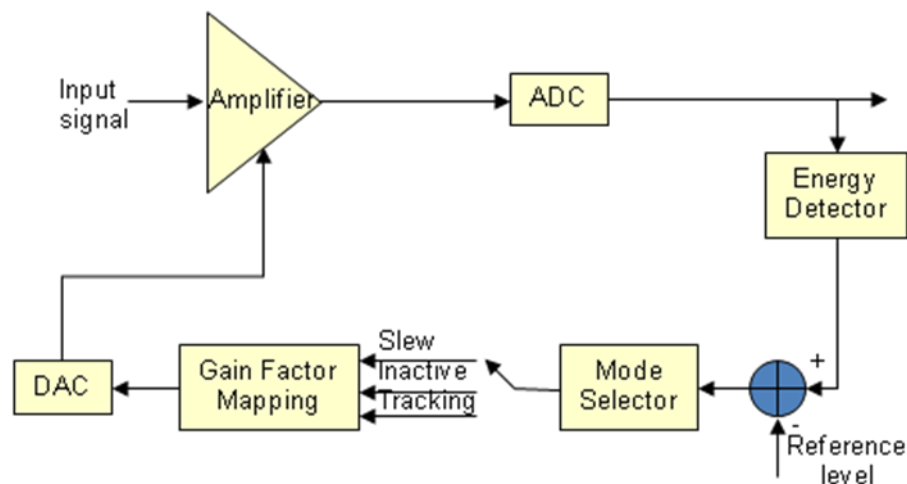


Figure 8. AGC block diagram

An AGC has to react fast enough to increase the signal gain but has to avoid jumping around of a ripple behavior of the signal. There is a need of some hysteresis to handle this situation (Fig. 7).

The AGC (Fig. 8) is a closed loop system that is based on energy measurement of the signal which is constantly compared with a reference value. The difference is used to increase the gain until the energy becomes the selected. The amount of gain applied and the time to reach the desired level establish three modes of operation, slew mode, tracking and inactive mode.

The slew mode is selected when the error inside the loop is high, then the gain is incremented in larger step sizes. The tracking mode becomes the opposite situation where smaller steps will be applied. Once the signal is around the desired value, the inactive mode is reached. The gain factor is usually mapped into a look up table, according to the operation modes, that drives a voltage gain amplifier (VGA) by means of a DAC.

Finally the ADC converter is the key component of the chain. Sampling rate, dynamic range, resolution and power consumption are the characteristics to be evaluated in ADC. Also the price and power consumption of the ADC increase with the sampling rate. A faster ADC will allow having more flexible radio system architecture as the ADC moves closer to the antenna and would reduce components as mixers, LOs and amplifiers. For a mobile device the power requirements become prohibitive.

## Chapter 2

# Spectrum Sensing Techniques

### 2.1 Spectrum Sensing

Detecting the presence of signals in the frequency spectrum is called spectrum sensing. An empty spot in frequency will be a candidate to allocate a new communication link. Under the definition of a primary and secondary user, this last one has to be able to detect or may be informed of an incoming primary user and move on the fly to another vacant spot. This action requires some level of certainty on the process to find an empty spot as well as fast allocation of this.

There are centralized methodologies that rely on a central unit that updates and holds information about the spectrum utilization either using its own spectrum sensing capabilities or even collecting and gathering spectrum information from portable or mobile units.

Regardless where the spectrum sensing device is located, it has to be capable detect signal presence over noise levels and even identify and recognize in some applications specific signals or services using the spectrum. The challenge of the spectrum sensing is to perform the detection reliably and within a required time response.

Of course some flexibility is also needed to scan the spectrum and digital processing takes place regardless of the used detection method. The processing is usually performed out of band since transmission or reception cannot be interrupted.

There are mainly three methods to determine signal presence, all of them with some trade-off. They are energy detection, matched filtering and cyclostationary detection.

It is important to note that the evaluation of these methods to use for spectrum sensing is heavily biased on the CR application requirements. There are applications where there is not prior

knowledge of the signals on air. Some others consider some specifications about the signal as partial range of the spectrum, bandwidth, modulation, etc. Some applications look for identification of specific signals.

## 2.2 Energy Detection

Energy detection is the most common way of spectrum sensing due to the lower computation required and no need of any knowledge about the possible signal. The energy detected is compared with a threshold established over the noise floor.

The detection could be done in time or frequency domain. Time domain implementation consists of averaging the square of the signal. Frequency domain requires an FFT whose size  $N$  improves frequency resolution helping narrow band signal detection and later averaging over the observation time. Frequency domain gives a second degree of freedom to improve the algorithm. Finally in both cases, the result is compared with the threshold.

Some algorithms include adaptive threshold calculation since the noise floor could vary under certain circumstances. But the presence of in-band interference would not be differentiated by the energy detector.

Some experimental works [2] assure that by increasing the average time the energy detection algorithm could go at negative SNR until certain limits where it fails since the noise modeling is not real. In any case, the time response makes such implementation prohibitive.

Energy detector is not able to recognize between noise, signal or interference and these reasons rule out this method for many CR applications although it could be used to support more complex strategies.

Although this is the less expensive and more generic method, it performs poor at low SNR levels and could lead to false detection. Also energy detection does not work for spread spectrum [3].

There are many improvements over the basic idea of energy detector that mostly focus on the threshold calculation adaptively [1][2]. Others require even synchronization to the primary user network [4]. There is also a work that improves the basic energy detection for the purpose of using it on WRAN systems exploiting the maximum to mean power ratio [6] to provide even identification.

### 2.3 Matched Filter

Matched filter approach gives the best SNR since it matches a specific signal. The approach requires demodulation of the signal which means having in advance deep knowledge of primary user signal.

Coherent detection is required and it implies time and frequency synchronization, knowledge of modulation scheme, bandwidth, frame format, etc. It could be impractical to implement a CR which could hold capabilities for all signal types. Although this information could be saved in the spectrum sensing device as a catalog of possible signals, it would come costly in terms of processing and memory.

### 2.4 Cyclostationary Feature Detection

In general any modulated signal include some periodicity by definition and some others added for synchronization or signaling purposes such as preambles, pilots, cyclic prefix, etc. It means that autocorrelation of the signal exhibit an observable grade of periodicity.

Instead of power spectral density, cyclic correlation function is used, and the algorithms are able to differentiate noise from signals, since noise is not correlated.

The Cyclic Spectral Density CSD is formed after the spectrum and will output peaks when cyclic frequencies are present.

These algorithms are based on a statistic approach which means an average has to be performed and it requires time to give an output. Also the process involves more than one FFT calculation and correlation making it computationally pricy compared with some other methods.

Cyclostationary detection deteriorate with the sampling frequency offset [7] the reason is that the spectral correlation function is estimated based on the correlation of the FFT coefficients, that due to any variable offset, could cancel each other instead of adding up.

## 2.5 Cooperative spectrum Sensing

There are some situations where a radio system is not able to be aware of the surrounds, like in fading channels, local interference, hidden source, etc. Data recovering could become impossible is such scenarios. But if multiple radios are performing around and could share their knowledge acquired by its individual spectrum sensing devices, then the probability of detection would increase [5].

Another way to see cooperative sensing is to picture it as a multi-antenna system and diversity included. This cooperation is based on a network concept and could be handled by a central unit that manages the spectrum, based on updated information coming from the multiple radios as part of its data base. This data base also includes primary user's information and frequencies vacant by city permits.

Cooperative strategies imply a network layer that will connect bases and also a back channel between radio and its base. This last detail would consume bandwidth unless a separate RF link is used for these purposes.

## 2.6 Other Methods

Waveform based, multitaper spectral estimation, radio identification, wavelet transform are some of the other different approaches found in the literature that are applied for spectrum sensing. Also it is common to find a combination of them to improve time response or save computation. Some solutions are more specific regarding the application and exploit some prior degree of knowledge about the sensing environment either as part of the sensing strategy or by definition of the approach.

## Chapter 3

# Orthogonal Frequency Division Multiplexing OFDM

### 3.1 OFDM in Cognitive Radio Systems

OFDM is a modulation scheme that uses multiple carriers to transmit data. Each of these carriers could be modulated using any variation from BPSK to N-QAM. Historically frequency division multiplexing was used to allocate different data channels. OFDM multiplex in frequency too but use all the carriers in order to send data from one channel. The idea is to split the data to be transmitted over multiple lower rate channels, making it more robust but getting higher bit rates in the overall transmission. Such a scheme was improved by defining orthogonality between the used carriers, allowing them to be closer to each other and reducing the needed bandwidth.

OFDM has many decades of existence but the current technology available for digital signal processing allows implementing such multi-carrier modulation on the digital domain meeting the real-time requirements. It is worth noting that the use of the FFT makes OFDM implementation easy.

Digital domain implies flexibility and it is known that cognitive radios lay over software defined radios, which in turn demands great grades of programmability. OFDM definition makes it easier to be adapted to different bands and different performance requirements by changing parameters on the implementation. These are some of the reasons why OFDM is the technology that best fit for cognitive radio systems.

A cognitive radio system requires spectrum sensing capabilities that are usually implemented by means of the FFT. OFDM already has an FFT machine that in many cases could be shared for spectrum sensing algorithms.



Definition of OFDM carriers takes place in the digital domain before the FFT. This allows the manipulation of individual carriers as a strategy to reduce the current bandwidth in case needed. This is where adaptability of OFDM resides, making possible to modify power on individual carriers, suppressing any of them, modulation order and even spectrum shaping. This flexibility allows even to get more advantage of available frequency spots, since it is possible to adapt the transmission to the size of the spot. In fact Multiband OFDM uses even scattered free spots to allocate transmission thanks to the ability to pick and shape carriers to be used. Along with the adaptability of OFDM, another good reason is that many current radio technologies are running with this scheme like Wi-Fi, Wi-Max, DVB-T between others.

OFDM signal is also appreciated by its robustness against multipath and this is accomplished by defining cyclic guard intervals as part of its structured design. This feature is susceptible to be easily modified to meet different environments.

Although cognitive radio is an emerging technology still under development, some of the ideas developed in this discipline are already moving into existent radio systems. The reason is that current OFDM based systems are ready for these changes due to the inherent flexibility. OFDM requires high synchronization in frequency and time domain as well as channel estimation. The inclusion of pilots, preambles and cyclic extension make possible for the receiver to achieve the exigent requirement of synchronization. The price to be paid as an overhead reflects as less bandwidth efficiency.

### 3.1.1 Advantage and Disadvantages

OFDM present also some challenges that together with cognitive radio broad requirements, makes this technology subject to more investigation [32][37]. But, it is worth mentioning the advantages and disadvantages of OFDM [33] itself.

Advantages:

- Simple implementation by means of the FFT
- High spectral efficiency considering the number of sub-carriers.
- Adaptability over frequency spectrum
- Capable of rate adaptation due to different modulations by carriers
- The anti ICI and ISI design concept makes OFDM receiver less complex since almost no equalizer is needed.

Disadvantages:

- High peak-to-average power ratio (PAPR) requiring high linear amplifiers.
- Requires accurate time and frequency synchronization.
- Sensitive to Doppler effects.
- Guard time introduce overhead
- The quality of the transmitter and receiver oscillators could influence in-phase noise.

### 3.2 Definition

An OFDM signal is made of multiple carriers closely spaced in the frequency domain. The orthogonality between carriers, required by OFDM, allow them to be closer without causing interference. Such arrangements reduce the used bandwidth. Every single carrier is digitally modulated carrying 2 bits for BPSK, 4 for QPSK up to  $2^n$  for  $n$ -QAM. For a single carrier the defined complex signal is

$$S_c = A_c(t)e^{j[2\pi f_c t + \phi_c(t)]} \quad (1)$$

Expression (1) defines a signal with magnitude and phase that vary with the time. Now the OFDM signal is the summation of an N number of carriers expressed as in (1).

$$S_s(t) = \frac{1}{N} \sum_{n=0}^{N-1} A_n(t)e^{j[2\pi f_n t + \phi_n(t)]} \quad (2)$$

During the symbol length the amplitude and the phase remains constant

$$S_s(t) = \frac{1}{N} \sum_{n=0}^{N-1} A_n e^{j[2\pi f_n t + \phi_n]}$$

Since the carriers are centered around some central frequency as  $f_0$  then  $f_n = f_0 + n\Delta f$ , but  $f_0$  could be zero reference, then  $f_n = n\Delta f$ . On the other hand, in a discrete domain,  $t$  becomes  $kT$ , where  $T$  is the period of the sampling frequency.

$$S_s(kT) = \frac{1}{N} \sum_{n=0}^{N-1} A_n e^{j[(2\pi n\Delta f)kT + \phi_n]}$$

This in turn could be expressed by

$$S_s(kT) = \frac{1}{N} \sum_{n=0}^{N-1} A_n e^{j\phi_n} e^{j[(2\pi n\Delta f)kT]} \quad . \quad (3)$$

Equation (3) resemble the IFFT where  $A_n e^{j\phi_n}$  is the representation in frequency domain.

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{j2\pi \frac{nk}{N}} \quad . \quad (4)$$

Expression (4) is the IFFT that by definition is the summation of orthogonal components in the frequency domain. Then, if we want to assure in (3) the orthogonality from the exponentials

$2\pi n\Delta f k T = 2\pi n \frac{k}{N}$  and after simplifications, this becomes the condition to meet. For a data

sequence made of complex elements  $d_n = a_n + jb_n$  where  $a_n$  and  $b_n$  have values according the modulation scheme (BPSK, QPSK and N-QAM). Replacing  $N$  values of this complex data into equation (3)

$$S_s(kT) = \sum_{n=0}^{N-1} d_n e^{j[2\pi f_n t_k]}$$

Then after complex multiplication the real part the symbol duration becomes

$$\text{Re}\{S_s\} = y(t) = \sum_{n=0}^{N-1} [a_n \cos(2\pi f_n t_k) + b_n \sin(2\pi f_n t_k)] \quad (5)$$

### 3.3 The FFT/IFFT in OFDM

After the mathematical description of OFDM, it seems clear that the IFFT is used at the transmitter side after mapping the bits and the FFT is used at the receiver to recover data.

The size of the IFFT is directly related to the number of carriers. For a given bandwidth if the number of carriers increases, the length in time of the OFDM symbol increases too. Then, a large FFT gives a larger OFDM symbol which is more robust for the multipath problem, but carrier spacing is reduced and intersymbol interference (ISI) could appear. This is a trade-off to be evaluated. If there is no need to use all the carriers, it is common to make them zero. This way the design can have the carrier spacing needed, the data rate defined and a good length in time domain for the OFDM symbol to face multipath effects. Although amplitude attenuation and phase rotations remain, the channel equalization is now achieved by means of a complex multiplier.

### 3.4 OFDM frame structure

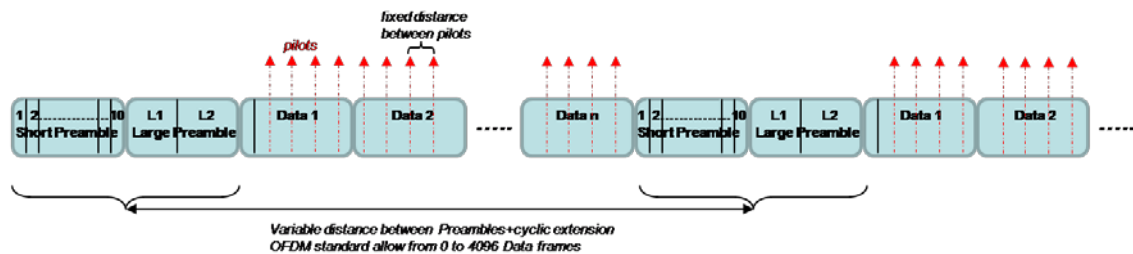


Figure 9. OFDM structure

OFDM is currently used for DVB-T, Wi-Fi, Wi-MAX and some particular or even proprietary applications. In all cases OFDM follow some common structure designed to help data recovery and fight channel impairments. Specifics appear in cases such as wired channel, wireless mobile

or wireless steady application. The common OFDM structure is packet oriented, based on frames which are in turn based on OFDM symbols. The packet starts with a preamble with the purpose to allow the receiver to recover synchronization. The preamble is followed by subsequent frames holding data encoded into the OFDM symbols. Fig. 9 shows the OFDM structure for the 802.11a/g standard.

### 3.4.1 Guard Interval and Cyclic Extension

OFDM structure inserts between symbols what is called “guard interval”. Intersymbol interference (ISI) and intercarrier interference (ICI) could appear due to transmission channel distortion. An efficient way to fight this is the use of the “cyclic extension” guard interval. If the guard interval is longer than the channel impulse response then the ISI is eliminated. The cyclic extension is a copy of part of the OFDM symbol that is placed at the beginning of the symbol. This action mimics an infinite behavior in the signal (instead of a truncated one) that a linear-time-invariant transmission channel would expect. This improves robustness to multipath effects. ISI in single carrier systems was historically solved by means of channel estimation and equalizers on the receiver side. For higher data rates this implementation becomes very complex and expensive in resource terms. OFDM appeared with multiple lower rates carriers with better immunity to fading and achieving overall higher data rates.

### 3.4.2 Pilots

Another key ingredient in the structure of OFDM signals is the existence of pilot tones. Pilots are included to help receiver accomplish channel estimation, frequency estimation and in some cases

even carry management data. Usually some of the carriers are chosen as pilots to modulate by BPSK or even QPSK some selected complex values. Basically the receiver looks for them and apply minimum mean square error (MMSE) criterion for maximum-likelihood estimation.

There are many different pilot schemes, some of them use pseudo-random values to avoid spectral lines, some of them apply the tones at the same spectral position every OFDM symbol, and some move the pilots between symbols to preset locations. Reference [38] study the performance of the channel estimator for two different pilot schemes where one of them use the strategy of even using one complete OFDM symbol full of pilots every fixed number of regular OFDM symbols. Scattered pilots or fixed pilots, in any case pilots are transmitted with the same power and assure a cyclic character over OFDM symbol or frame oriented.

It is also important to consider for the pilot design that its influence in rising the power-to-peak average ratio of the signal. As it will be detailed later, this is of great concern and value and also placement of the pilots help in the efforts to be made at the front-end stage.

### 3.4.3 Preamble

The preamble in an OFDM signal is designed with the objective of providing the means to perform channel estimation, frequency offset estimation and to identify the beginning of the OFDM signal. It has to have good correlation properties and avoid complex algorithmic implementation for the recovery tasks.

The task of identifying the beginning of the OFDM signal is performed by time correlating the incoming signal with a local copy of the preamble.

The frequency offset between the transmitter and receiver is estimated basically by multiplying the incoming signal with the conjugate of the local copy. This operation takes out the modulation influence. The preamble is usually made of several copies of a short signal with period  $\gamma$ .

$y(t) = x(t)e^{j2\pi\Delta_f t}$  and  $y(t - \gamma) = x(t)e^{j2\pi\Delta_f t}$  where  $\gamma$  is the period of the preamble.

Then  $y(t - \gamma)y^*(t) = |x(t)|^2 e^{-j2\pi\Delta_f \gamma}$  and taking the angle, the frequency offset becomes

$$\Delta_f = \frac{\text{angle}[y(t - \gamma)y^*(t)]}{2\pi\gamma} \quad (6)$$

OFDM in 802.11a/g include a combination of 10 short preambles to help the receiver AGC to be adjusted and for timing synchronization. Another two large preambles follow for channel estimation purposes. Although there are some strategies that base channel estimation on the preamble, most of them make exclusive use of the pilots [37].

### 3.4.4 OFDM Symbols

OFDM symbols hold data of different purposes on the stream. Is common the have of the first symbol after the preamble to carry information about the connection to the receiver such as data rate, packets length, etc. Complex data is transported by a number of carriers as detailed in expression (5).

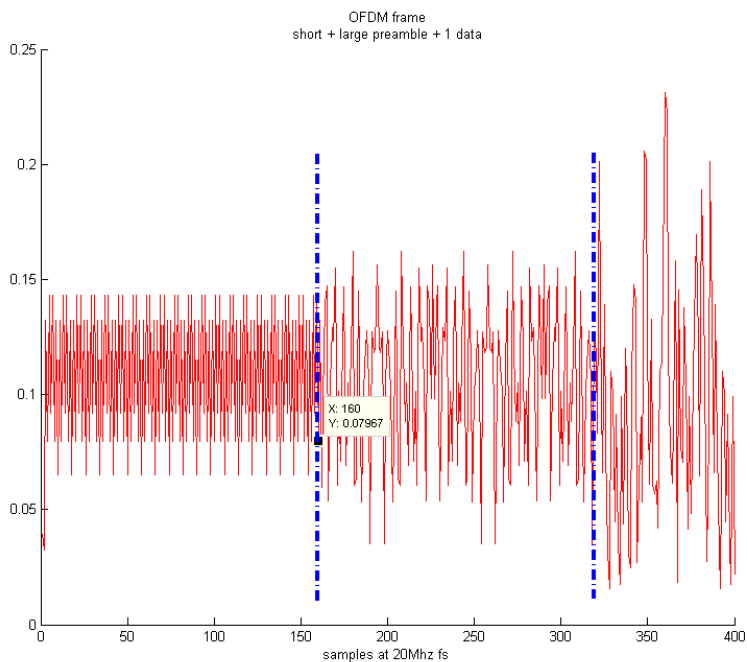


Figure 10. OFDM symbol in time domain

### 3.5 OFDM Implementation

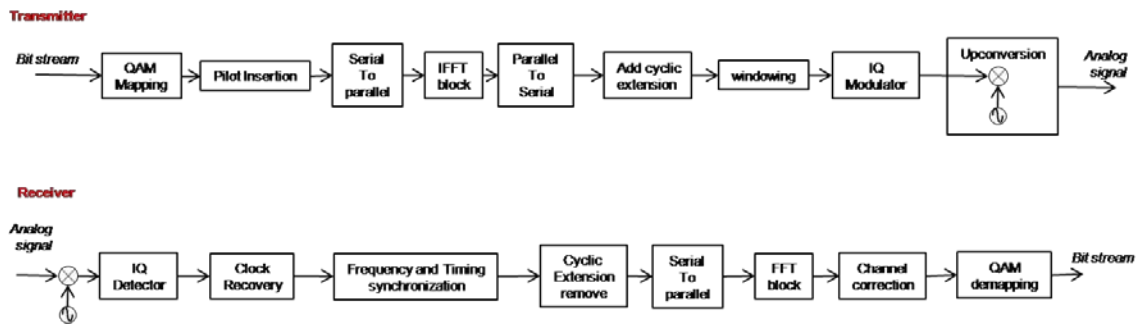


Figure 11. Transmitter & Receiver diagram

Fig. 11 shows a block diagram of the receiver and transmitter for an OFDM system. The Fig. focuses on a digital domain implementation. Current technology makes more convenient to implement OFDM in the digital signal processor (DSP) to allow flexibility or even into FPGA if faster processing is needed.

The bit stream coming into the QAM mapping usually has been submitted to some pre-processing such as scrambler, FEC encoder, interleaver. These blocks make some new arrangement of the incoming bits and the FEC increase the number of bits as a result of its algorithm.

Usually the IFFT/FFT block is shared by the transmitter and receiver. The IQ modulator and demodulator could be a hardware piece or perform in digital domain. At the transmitter a DAC converter with a low pass filter is used to convert the signal to analog domain at some inter media sample rate, before up-converting the signal by the mixer.

The receiver front-end transform the incoming analog signal to suitable levels to be apply to the ADC. The ADC and the IQ detector are usually tuned their respective frequency and phase of operation, by some algorithm.

This thesis implements an OFDM system shown in Fig. 11. The transmitter creates an OFDM signal for 802.11g from the incoming bit stream. The spectrum of this signal is used to apply and detect cyclic features by using cyclostationary detection. Although the receiver is not really



needed for the cyclic detection, it is also implemented in this thesis. This allows validation of the transmitter design and introduces us the basic receiver tasks. Appendix A contains the Matlab script files developed for this purpose.

### 3.5.1 Transmitter

QAM mapping is the block that groups bits according the chosen modulation scheme in groups of  $2^n$ , where  $n$  is 1 for BPSK, 2 for QPSK, and any higher corresponds to n-QAM. These grouped bits are compared with a table where an equivalent complex number is extracted. These values are located in the frequency domain and correspond to each of the carriers.

According 802.11a/g there are 52 data carriers where 48 are data and 4 of them are pilots. There are zeros defined for DC and the extremes of the 20 MHz band. Table 1 shows the basic parameters.

Pilot Insertion is performed next by placing the defined values into the defined carrier places. 802.11a/g uses a pseudo-random sequence of BPSK modulated pilots. 64 carriers at 20 MHz sample rate defines frequency spacing between carriers of 0.325Mhz. and also the size of the IFFT to be performed.

$F_s = 20\text{Mhz}$	IFFT/FFT size = 64
Sub-carrier spacing = 0.325Mhz	BW = 20Mhz
Data Carriers = 48	Pilot Carriers = 4
Null Carriers = 12	Long Preamble = 2 x 4us
Short Preamble = 10 *0.8us	Carrier Spacing = BW / Carriers

Table 1

The IFFT block has to be fed in parallel fashion to perform the conversion over the 64 carriers (DC included). The output is a sequence now in time domain.

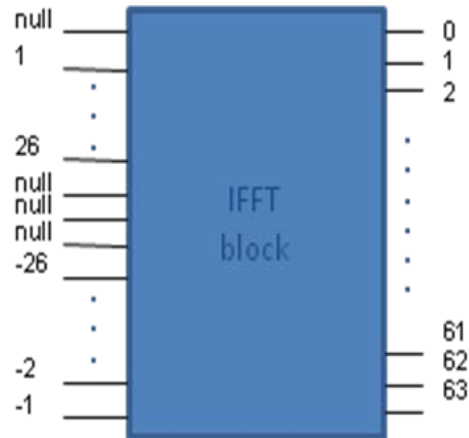


Figure 12. IFFT block

Cyclic extension is performed next by copying last 16 samples of the serial sequence, into the beginning of each OFDM symbol. The result is a new sequence of length of 80. The preamble made of short and large preambles account for 320 samples. Fig. 13 is an OFDM signal 64-QAM, 2 packets made of 2 data frames.

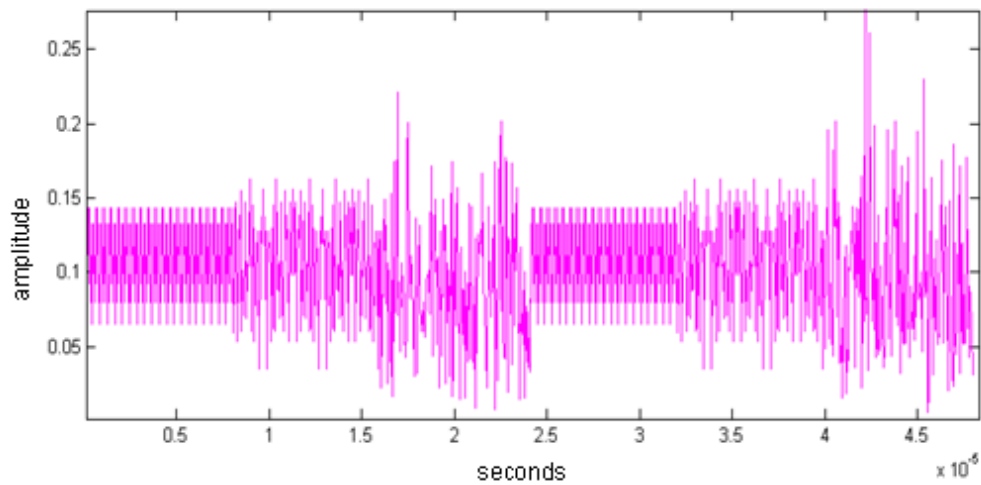


Figure 13. Preamble &amp; data OFDM

In order to meet the spectrum mask conditions detailed in reference [33] windowing is applied, as the reference suggest in a digital implementation with a 20 MHz sample rate.

The signal generated is in the time domain and in digital format. It has to be converted with a DAC with enough bits and maybe oversampled to reduce quantization errors. The output of the DAC is then passed through a low pass filter to get rid of the quantization and make it analog. Later this signal has to be up-converted to the operation band, which in the version g of 802.11 is around 2.4 GHz.

Fig. 14 shows the spectrum of a QAM-16 OFDM 802.11a/g signal generated with the implemented Matlab code.

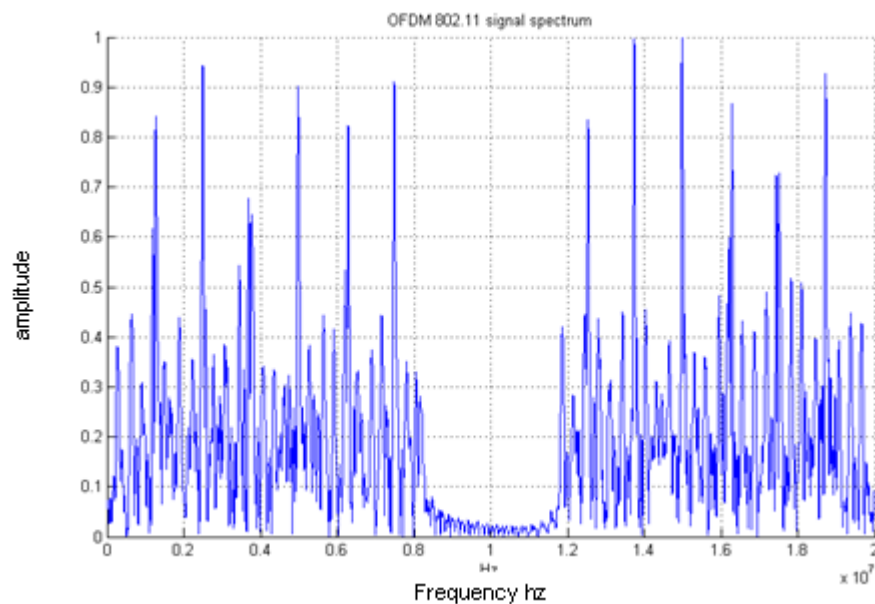


Figure 14. OFDM frequency spectrum

There could be other blocks involved at the transmitter since they do not contribute to the thesis target of cyclostationary, they are not included. Some of this could be those efforts to reduce the PAR or “crest factor”.

One of the issues or trade-off in digital modulation systems that use some form of QAM is the requirement of high power and highly linear power amplifiers (PA). Those kinds of amplifiers are the least efficient ones (Class A). Mobile devices designs using OFDM have to put huge effort to deal with this problem since power consumption is of concern.

Crest factor or peak-to-average power ratio (PAPR) is calculated as the ratio of the peak amplitude over the RMS value of the signal. OFDM implementation shows inherently high PAPR affecting the PA efficiency [43].

The target is to reduce those peaks that become bigger when data changes from some value of the QAM constellation to maybe another far away one. There is research on the topic, using different strategies to cope with the problem. Solutions go from applying Golay sequences theory to avoid such high peak combinations, to measuring and correcting iteratively the peak values based on power calculations and a Gaussian window and even clipping.

There could be also efforts to get more linearization of cheaper and less efficient PA by means of pre-distortion applied to the signal [43].

### 3.5.2 Receiver

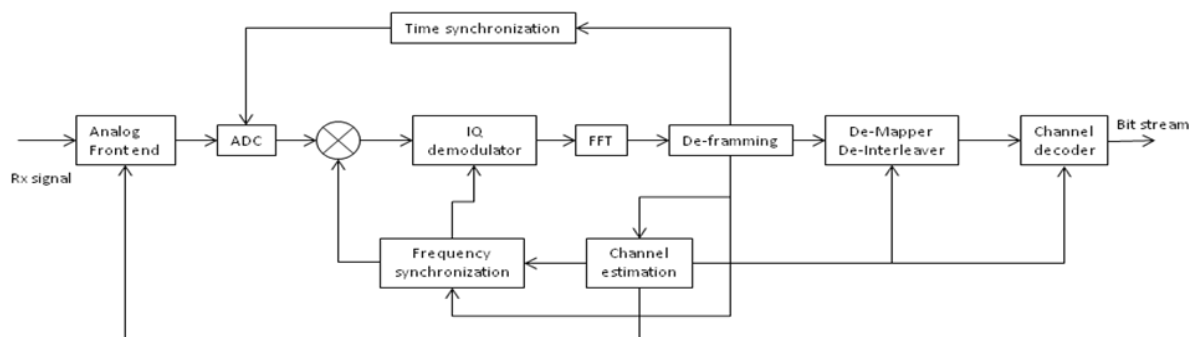


Figure 15. OFDM receiver

Fig. 15 describes a block diagram at the receiver that includes most of the process that takes place at the OFDM receiver.

The Analog Front-End is a conglomerate of analog hardware. Some of the most relevant are the SAW filter, LNA and AGC. The SAW perform the first pass band process followed by the down conversion to some lower intermediate frequency. The Low Noise Amplifier LNA amplifies the incoming signal to adequate levels. The Automatic Gain Control adjusts the incoming signal level

to best fit the ADC range. The AGC is a closed loop design that could be done by many hardware pieces or even totally in digital domain. Channel estimation block applies the amplitude correction by means of the AGC.

The clock running for the ADC has to be corrected by the timing synchronization block. For most of the OFDM receivers a 10bits conversion is usually enough although fewer bits could be used for QAM below 64.

The local oscillator is also corrected by the frequency synchronization block after estimation of any drift. The I&Q demodulator is also corrected by the frequency synchronization avoiding mismatch between I and Q branches. The I&Q demodulator could be also a chipset or could be implemented in digital domain.

Fig. 15 depicts a receiver with the I&Q and the local oscillator in digital domain. The decision between digital or analog solution for these blocks depends on the application. Moving to digital domain is not always convenient if computation is not able to be performed on time. Also quantization has to be evaluated accordingly and whether or not a fixed point implementation is possible or a more expensive floating point processing unit is needed.

After the FFT is performed, de-framing block is where the preamble, cyclic extension and pilots are recognized. Most of the algorithms to perform channel estimation, frequency and timing synchronization, are pilot aided and make use of cyclic extension and/or preambles.

Once the synchronization is achieved the receiver decoding, the correct start of symbol is found and the FFT could be applied over the data portion.

For our purposes, the cyclostationary detector, it is assumed that perfect synchronization on time, frequency and phase has been achieved. Then having the base band signal and after identifying the preamble, the cyclic extension has to be removed on every OFDM symbol before FFT is applied.

Having knowledge of the number of samples that cyclic extension is made of, the next 64 samples has to be converted to frequency domain using the FFT. Once this is done the pilot and nulls have to be extracted and then submit the 48 left samples to the demodulator.

The demodulator gets the complex sample and tries to match it with the respective N-QAM constellation value. It is common to use the Euclidean distance to determine which point of the theoretical constellation is closer to the complex sample. An important role for this recognition is usually played by the FEC decoding. Trellis decoding use what is call soft decision or hard decision mode to perform. A hard decision has clear threshold defined over the constellation where the comparison is made. The complex sample is then changed to the theoretical value, while soft decision uses the actual received complex sample and uses it to establish modified threshold to use with the next sample. This development is out of the scope for this thesis as it was explained and hard threshold decision without FEC is used. After comparing the complex sample to the I&Q thresholds, the sample is changed for the theoretical one and then pass to the QAM de-mapping.

QAM de-mapping usually consists of an efficient implementation of a LUT (look up table) to output the right group of bits.

Fig. 17 and 18 show a 64 QAM and 32QAM constellation respectively. Fig. 19 details a 64QAM constellation of the received signal depicted in Fig. 11. The red circles are the theoretical values and the black ones are the received ones. Although there are no errors in the transmission, is possible to observe some imperfections that seem to suggest a small mismatch in phase and also in magnitude. Dotted lines suggest the thresholds that in some solutions could be dynamically shifted to apply corrections based on boundary changes due to channel impairments.

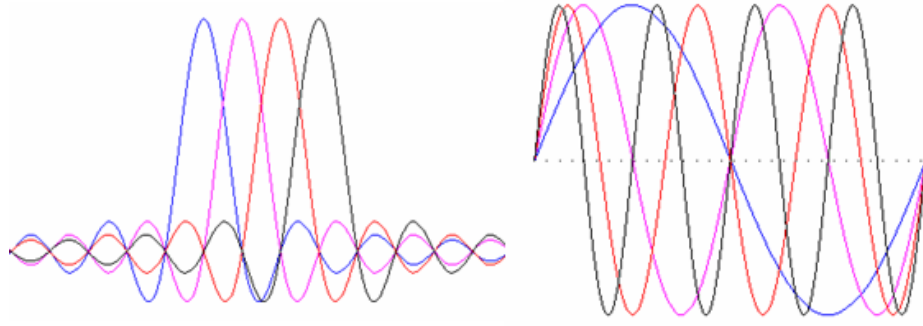


Figure 16. OFDM frequency & time domain

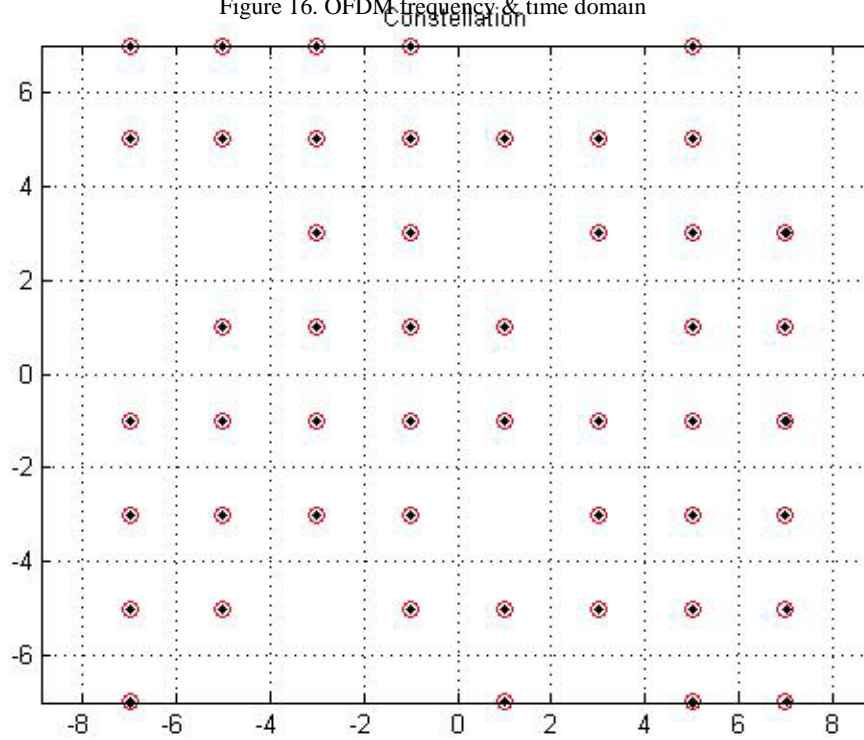


Figure 17. QAM 64 constellation

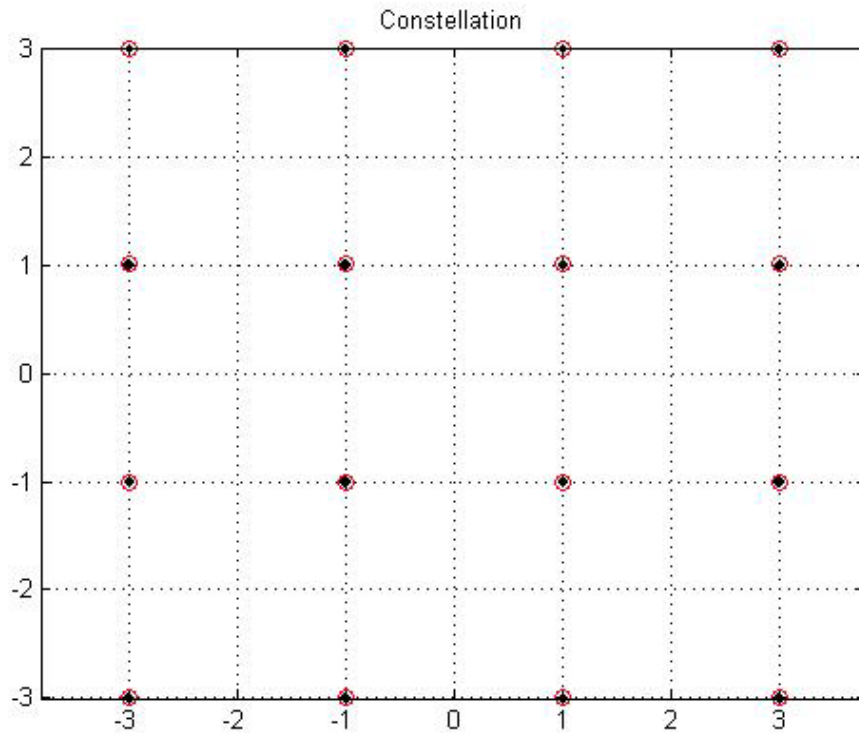


Figure 18. QAM 16 constellation

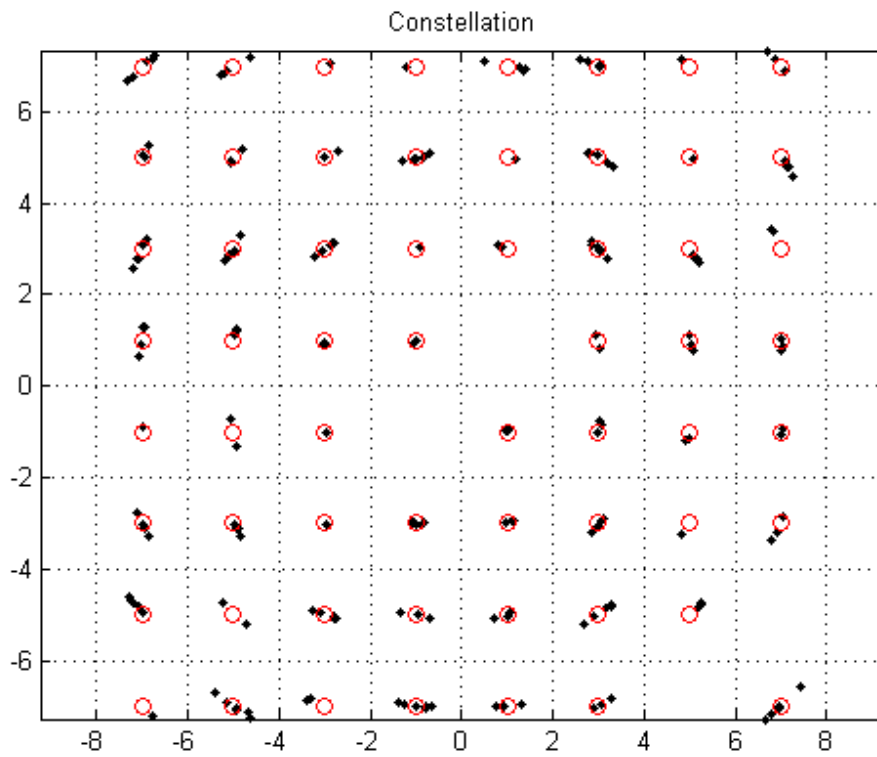


Figure 19. QAM 64 constellation showing frequency noise



## Chapter 4

### Cyclostationary Detection

#### 4.1 Cyclostationary Theory

In the literature, we can find many different approaches to get into the idea behind signal feature detection. A good start is to get into the definition of periodicity in a signal  $x(t)$ . First order periodicity could be pictured like

$$x(t) = x(t + T_0)$$

Fourier series coefficients become a helpful tool to describe it

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_0 t}$$

The types of signals we find in communication systems have been formed by means of sine waves that are periodic in nature. Although modulated signals are not really periodic, by means of transformations and Fourier analysis, periodicity information can be extracted.

A simple example of a modulated signal would be  $x(t) = a(t) \cos(2\pi f_0 t)$  where  $f_0$  represents the carrier which is a determined value in the frequency domain and  $a(t)$  is of random nature. In order to characterize the random part of the expression we can claim it as a wide-sense stationary process.

Power spectrum density, autocorrelation become now part of tool set needed to analyze such modulated signals.

If the autocorrelation of  $a(t)$  is  $R_a = E\{a(t)a(t-\tau)^*\}$  its power spectrum density is calculated by  $S_a(f) = F\{R_a(\tau)\}$  and in turn power spectrum density of  $x(t)$  could be expressed as:

$$S_x(f) = \frac{1}{4} S_a(f + f_0) + \frac{1}{4} S_a(f - f_0)$$

Clearly no spectral lines or sine wave components appear in this last expression.

As it was mentioned, some transformation could be applied to introduce or better said enhance any built in periodicity that would be more visible with the appearance of spectral lines. The most common example used to picture the idea is the use of a simple quadratic transformation like

$y(t) = x(t)^2 = a(t)^2 \cos^2(2\pi f_0 t)$  Using trigonometric identities it becomes:

$$y(t) = \frac{1}{2} [b(t) + b(t) \cos(4\pi f_0 t)] \text{ where } b(t) = a(t)^2$$

We can say that  $b(t)$  is positive and with this, it has a DC component or spectral line that would appear together with the spectrum at  $f=0$  on the PSD.

This way, we can in turn assure that the PSD of  $y(t)$  contain scaled copies of  $b(t)$  PSD (including the spectral line) at  $\pm 2f_0$  and  $f=0$ .

$$S_y(f) = \frac{1}{4} \left[ K\delta(f) + S_b(f) + K\delta(f \pm 2f_0) + \frac{1}{4} S_b(f \pm 2f_0) \right]$$

The graphic below clarifies it.

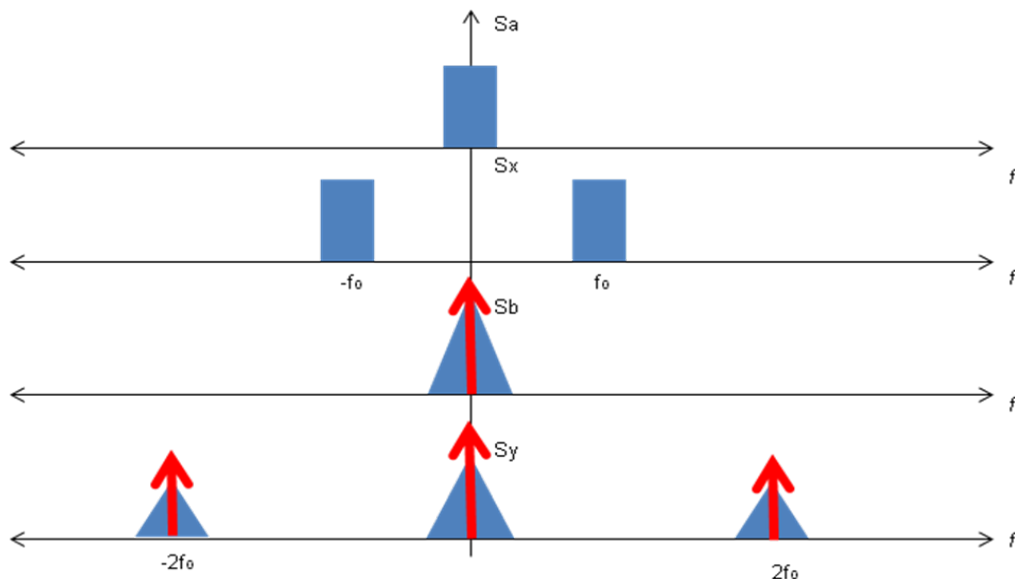


Figure 20. Generating spectral lines

## 4.2 Cyclic Autocorrelation Function

The quadratic transformation used in the example above, the squarer, pictures good enough how the periodicity involved in a modulated signal can be bold making spectral lines appear into the scene. But squarer transformation doesn't work for all cases then a delay could be included in the transformation. This idea come from the example of having a pulse modulation of unique magnitude like +/-1 that after square hide any spectral line but the dc one.

Then the transformation  $y(t) = x(t).x(t - \tau)$  assures spectral lines for  $m. f_0$  where  $m$  is an integer. Defining  $\alpha = m f_0$  we declare

$$\begin{aligned} M_y^\alpha &= \langle y(t)e^{-j2\pi\alpha t} \rangle \quad (1) \\ &= \langle x(t).x(t - \tau)e^{-j2\pi\alpha t} \rangle \neq 0 \end{aligned}$$

At this point is time to say that  $x(t)$  has second order periodicity if and only if the delay product contains spectral lines. For convenience theory use the symmetric delay product instead.

$$y_\tau = x(t - \tau/2)x^*(t + \tau/2) \quad (2)$$

According to [11] the conjugate is included to accommodate complex values.

$$M_y^\alpha = \langle x(t - \tau/2)x^*(t + \tau/2)e^{-j2\pi\alpha t} \rangle \quad (3)$$

This is the place where the autocorrelation appear into the theory. Equation (3) is viewed as the generalization of the autocorrelation when  $\alpha = 0$ .

$$R_x^0(\tau) = \langle x(t - \tau/2)x^*(t + \tau/2) \rangle$$

This way from now on the statement (3) be re named as cyclic autocorrelation function

$$R_x^\alpha = \langle x(t - \tau/2)x^*(t + \tau/2)e^{-j2\pi\alpha t} \rangle \quad (4)$$

The reason of the name is due that the autocorrelation involved seems to be weighted by the factor  $e^{-j2\pi\alpha t}$  for any  $\alpha$ .

If some re-arrangement is performed we can make it (4) as

$$R_x^\alpha(\tau) = \left\langle [x(t - \tau/2)e^{+j2\pi\alpha(t-\tau/2)}][x(t + \tau/2)e^{-j2\pi\alpha(t+\tau/2)}]^* \right\rangle \quad (5)$$

If we make:

$$\begin{aligned} u(t) &= x(t)e^{-j2\pi\alpha t} \\ v(t) &= x(t)e^{+j2\pi\alpha t} \end{aligned}$$

Then expression (5) could have the interpretation of the conventional autocorrelation of two signals that happen to be the same one but shifted in the frequency domain by  $\pm \alpha/2$ . That is if  $X(f)$  exist

$$\begin{aligned} U(f) &= X(f + \alpha/2) \\ V(f) &= X(f - \alpha/2) \end{aligned}$$

As another conclusion is possible to say that  $x(t)$  has second order periodicity if there exist correlation between two shifted symmetric versions of  $x(t)$ .

### 4.3 The spectral-correlation function

The Spectral Correlation function definition comes from the basic idea of finding the average power in the frequency domain as  $R_x(0) = \langle |x(t)|^2 \rangle$ . If the correlation in the frequency domain between the shifted versions  $v(t)$  and  $u(t)$  has to be found then the expression becomes

$$R_x^\alpha(0) = \langle u(t)v^*(t) \rangle = \langle |x(t)|^2 e^{-j2\pi\alpha t} \rangle$$

The Power Spectral Density PSD could be pictured as passing the signal  $x(t)$  by a narrowband pass filter and calculating the average power, where the filter is replicated all over the spectrum.

In the limit where the bandwidth (B) of the filter approaches zero:

$$S_x(f) = \lim_{B \rightarrow 0} \frac{1}{B} \left\langle |h_B(t) \otimes x(t)|^2 \right\rangle \quad (6)$$

This expression gives PSD at any particular  $f$  where the narrowband filter is located. Now using the same expression for the shifted versions  $v(t)$  and  $u(t)$ :

$$S_x(f) = \lim_{B \rightarrow 0} \frac{1}{B} \langle |h_B(t) \otimes u(t)| |h_B(t) \otimes v(t)|^* \rangle \quad (7)$$

with the filters located at  $f$ .

Expression (7) is called Spectral Correlation Density (SCD) function. Then the following expressions are proved to be true

$$S_x(f) = \int_{-\infty}^{\infty} R_x(\tau) e^{-j2\pi f\tau} d\tau \quad \text{Fourier Transform of autocorrelation}$$

$$S_x^\alpha(f) = \int_{-\infty}^{\infty} R_x^\alpha(\tau) e^{-j2\pi f\tau} d\tau \quad \text{Fourier Transform of cyclic autocorrelation}$$

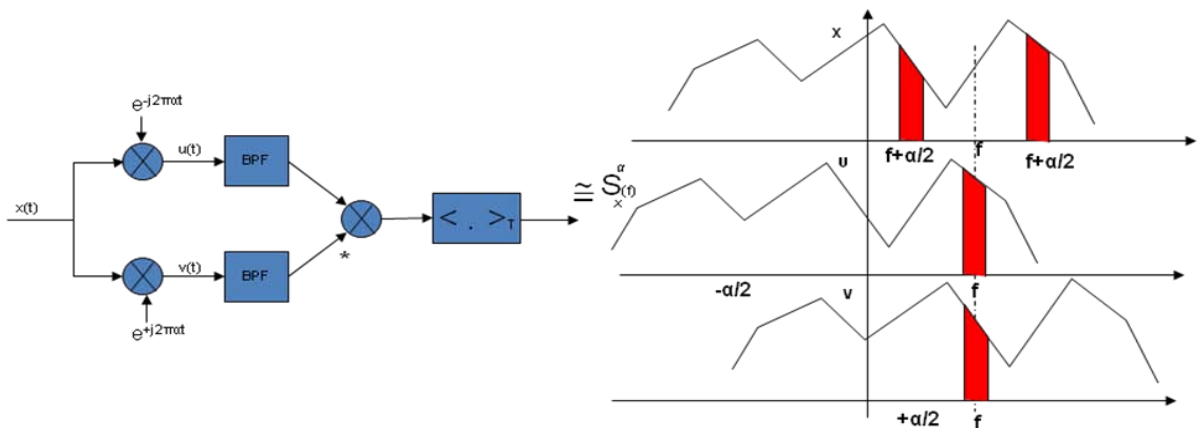


Figure 21. The spectral correlation function procedure

#### 4.4 Cyclic Spectrum Estimation

Nowadays communication systems are more and more developed in the discrete domain by means of digital conversion. This is how  $x(t)$  becomes a series  $x[n]$  and the FFT is used instead and that is how everything falls under the Digital Signal Processing discipline. At this point, it is clear to preview that computation by means of a processing unit as a DSP or further hardware implementation of the algorithms is needed.

There are a variety of algorithms that are born from the need to reduce the amount of computation involved in the SCD calculation. The impact of the time to be spent in such calculations is weighted against the application. Some applications show exigent real-time orders to be met. As it is expected, trade-offs appear as a result of either the efforts to reduce calculations or due to the quantization at different levels on the algorithms involved.

Among these algorithms, some apparent classifications could be made. References [28], [30] discuss in detail different approaches for the estimation and underline these under time-smoothed and frequency-smoothed algorithms. Two methods become the most efficient in terms of calculation, the Strip Spectral Correlation Analyzer (SSCA) and FFT accumulation (FAM), both under the time-smoothing classification.

The SCD function of  $x[n]$  is defined as  $S_x^\alpha(f) = \sum_{k=-\infty}^{\infty} R_x^\alpha(k) e^{-j2\pi f k}$  using the discrete fourier transform, where

$$R_x^\alpha(k) = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N [x(n+k) e^{-j2\pi\alpha(n+k)}][x(n) e^{-j2\pi\alpha n}]^*$$

This way:

$$S_x^\alpha(n, f)_{\Delta t} = \frac{1}{T} \left\langle X_T(n, f + \frac{\alpha}{2}) X_T^*(n, f - \frac{\alpha}{2}) \right\rangle_{\Delta t} \quad (8)$$

Where  $X_T(n, f \pm \frac{\alpha}{2})$  are the complex demodulators that by definitions are band pass signals shifted to DC.

The band pass filter is implemented as a data tapering window in the time domain of a length  $T = N'T_s$  with sample rate  $f_s = 1/T_s$ . The way complex demodulators are calculating:

$$X_T(n, f) = \sum_{r=-N/2}^{N/2} a(r) x(n-r) e^{-j2\pi f(n-r)T_s} \quad (9)$$

Data tapering window  $a(r)$  is well accomplished by a Hamming window.

## 4.5 FAM Implementation

FAM is one of the methods under time-smoothing classification which has good efficiency, computation wise. There are parameters involved that are used to trade-off resolution, reliability and of course computation reduction.

FAM consists of capturing in a time length  $\Delta_t$  a piece of the incoming signal  $x[n]$  which is the result of  $x(t)$  sampled at  $f_s$ . Estimation of the  $S_x^\alpha(n, f)_{\Delta_t}$  is performed over this time length. This computation is performed iteratively over consecutive pieces in the time domain until acceptable results for a summation of several  $S_x^\alpha(n, f)_{\Delta_t}$  satisfy the application, in terms of time of computation and objective to meet.

The demodulators are calculated by expression (9) and they require the band pass filter which is applied as a data tapering window of length  $T$  in the time domain. This window reflects in frequency domain as  $1/T$  that in turn defines the frequency resolution as  $\Delta f = 1/T$ . This is the frequency interval where the FFT is performed. At this point it has to be recalled that the calculation of the SCF requires of course the computation in the frequency domain. Clarifying that, what we have so far is the FFT performed over a duration (of length  $T$ ) inside an observation time of  $\Delta_t$ , and then it should be clear that  $T$  has to be moved to complete the observation time  $\Delta_t$ . This is consistent with the concept of recursive Discrete Fourier Transform by means of the technique of block oriented window [16].

$$\begin{aligned}
 X_m &= \frac{1}{N} \sum_{k=0}^{M-1} x_{kL} e^{-j\frac{2\pi}{N}mkL} + \frac{1}{N} \sum_{k=0}^{M-1} x_{kL+1} e^{-j\frac{2\pi}{N}m(kL+1)} + \dots + \frac{1}{N} \sum_{k=0}^{M-1} x_{kL+L-1} e^{-j\frac{2\pi}{N}m(kL+L-1)} \\
 &= \frac{1}{L} \sum_{L=0}^{L-1} DFT_M(m, l) e^{-j\frac{2\pi}{N}mL}
 \end{aligned}$$

M=size of the L blocks, l=index within blocks. (m,l) mth frequency component of the lth M-size DFT (N=ML)

Then,  $T$  is slide all over  $\Delta$ . It is possible to foresee that serious number of computations has to be performed, some of them on the number of FFTs and later into the resulting correlation of them. But, it is possible to introduce some decimation to reduce computation and that would be change the sliding FFT process to a hopping FFT one, made by blocks  $T$  size hopping by step  $L$ . Recall the fact that a second transformation brought us a second order periodicity which is the frequency domain where every cyclostationarity should appear, there is this other dimension  $\alpha$ .  $\alpha$  has its own resolution  $\Delta\alpha$  to describe the cyclic frequency.

Fig. 22 describes the parameters involved and their relationship in time, frequency and cyclic-frequency domains. Finally  $\Delta$ , and  $T$  represent some number of samples in the discrete domain and  $n$  are the instants of each FFT accomplished.

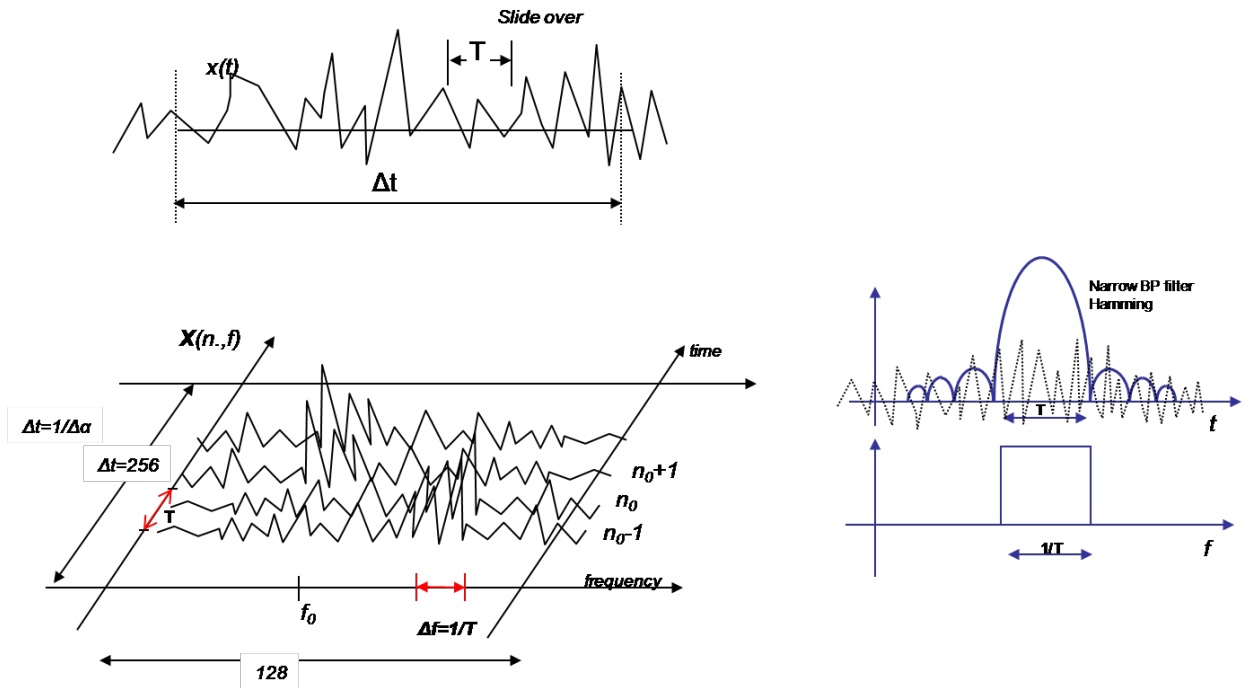
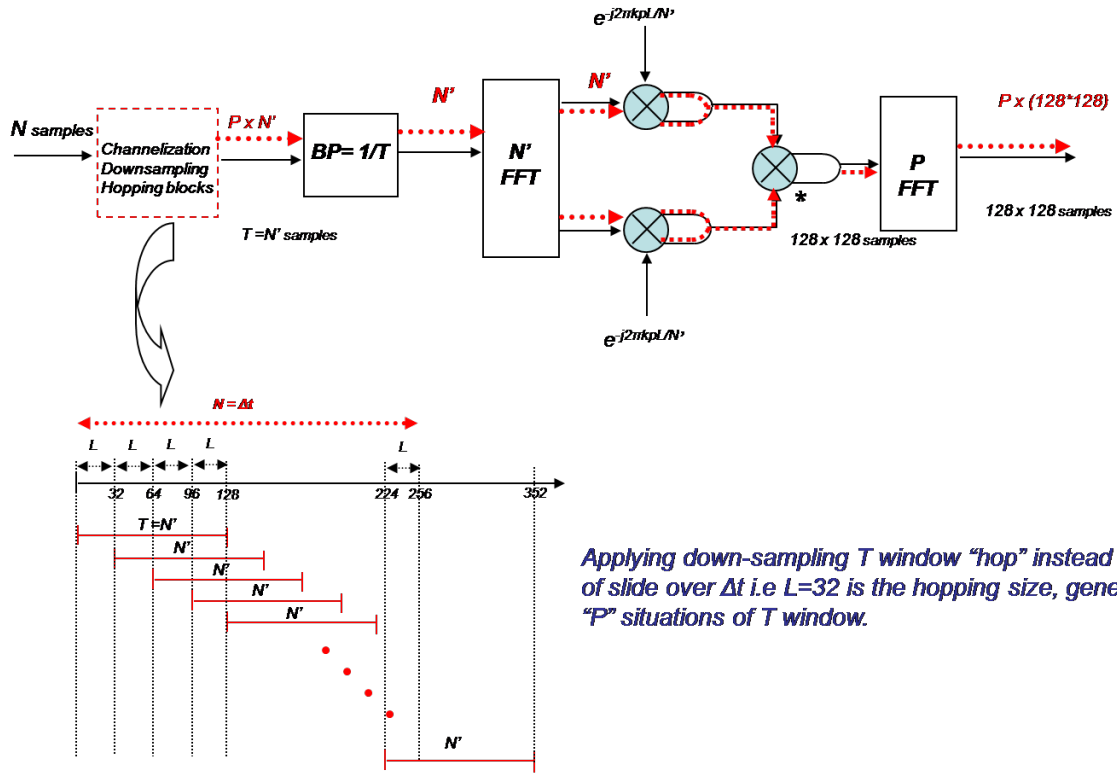


Figure 22. FAM procedure

Now having in mind expressions (8) and (9) and what so far was discussed and described, it is time to present the FAM implementation in detail. Fig. 23 shows the block diagram of FAM where the red dashed lines describe the efforts made to reduced the computation complexity by



means of down sampling and channelization.  $P$  is the number of windows of size  $T$  generated by the hop size  $L$ .  $N'$  is equal to  $T$  or better said is the number of samples equivalent to  $T$ .



Applying down-sampling  $T$  window “hop” instead of slide over  $\Delta t$  i.e  $L=32$  is the hopping size, generating “ $P$ ” situations of  $T$  window.

*Spectral Correlation Function SCF also known as Cyclic Spectral Density function  
Correlation between two spectral components of  $x(t)$  at frequencies  $(f+\alpha/2)$  and  $(f-\alpha/2)$   
over an interval of length  $\Delta t$*

Figure 23. FAM implementation

As it is shown in Fig. 21, the two branches of Fig. 23 comes from versions of  $x[n]$  shifted in frequency domain  $\pm \alpha/2$  that has to be correlated, one of them after the conjugate operation. This operation is represented on a bi-frequency domain as Fig. 24 depicts.

Over the x-axis where  $f_0$  is the target frequency, the band pass filter bandwidth is represented located  $\pm \alpha_0/2 f_0$  apart from  $f_0$ . The content of this frequency segment is multiplied (correlated) and is represented at y-axis for this  $\alpha_0$  as the red segment. The next step requires modified

$\alpha$  according to its resolution  $\Delta\alpha$ . Since  $\alpha_0$  move (in the example) the position of the narrow band pass filters move accordingly for the frequency domain, becoming closer. Now performing the cross multiplication, the blue segment is obtained. The green segment is performed applying the same process again, now decreasing to  $\alpha_0 - 2\Delta\alpha$ .

Although the graphic doesn't depict it, the same process has to be applied for positive increments of  $\Delta\alpha$ , which in turn move apart the narrow filters sequentially by  $\Delta f$  increments. The diamond shape appears now clear for the bi-frequency plane.

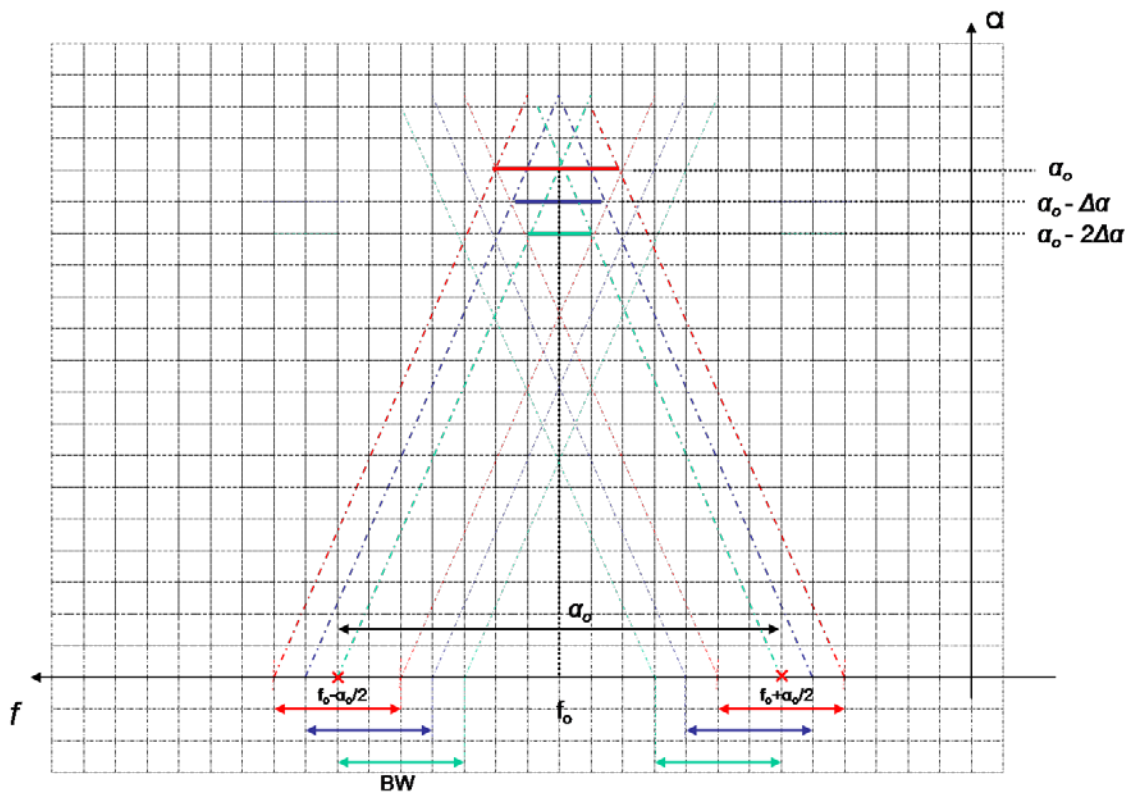


Figure 24. Bi-frequency plane

The resolution for the frequency and the cyclic frequency seems to be related and determine the resolution of the FAM. Reference [19] describes how reliability and resolution reside on  $\Delta f$  and  $\Delta\alpha$ .  $\Delta f$  is  $1/T$  and determine the resolution of the FFT performed on  $x[n]$  segments.

Also a good amount of smoothing has to be applied to the cyclic periodogram in order to overcome random results.

This brings that  $\Delta f \Delta t \gg 1$ .  $\Delta \alpha$  is considered to be  $1/\Delta t$  so:

$$\Delta f / \Delta \alpha \gg 1 \quad (10)$$

Fig. 25 and 26 picture this relationship on the bi-frequency plane for the example of  $\Delta f = 2\Delta \alpha$ .

The process described above cover frequency domain and cyclic frequency domain for the entire spectrum confined by the sample rate holding the relations between  $\Delta \alpha, \Delta f, T$  and  $\Delta t$ . The

discrete implementation calls for a definition of  $N$  and  $N'$  which one related by:

$$\Delta f = \frac{f_s}{N'} \quad \Delta \alpha = \frac{f_s}{N}$$

And by relation (10)  $NN' \gg 1$

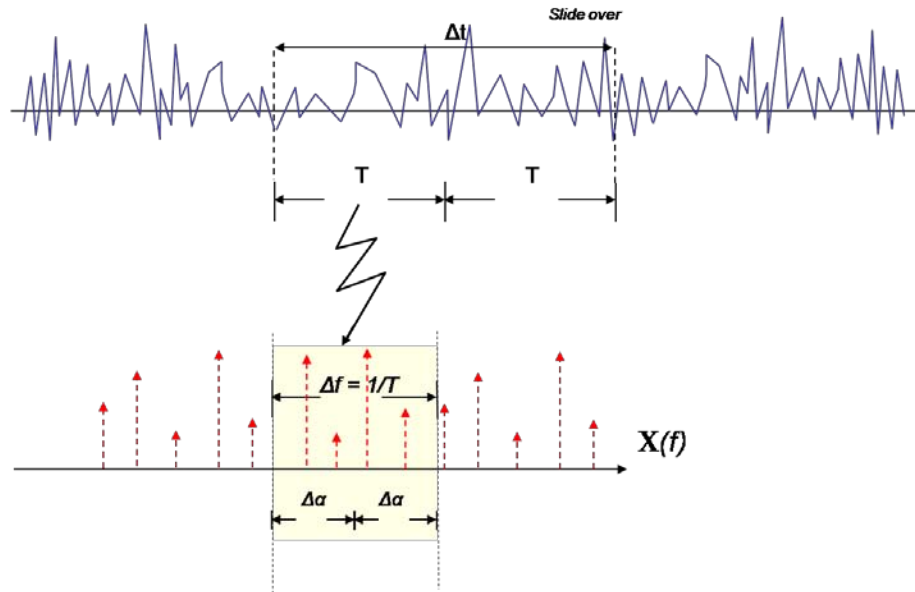


Figure 25. relationship between frequency and cyclic frequency

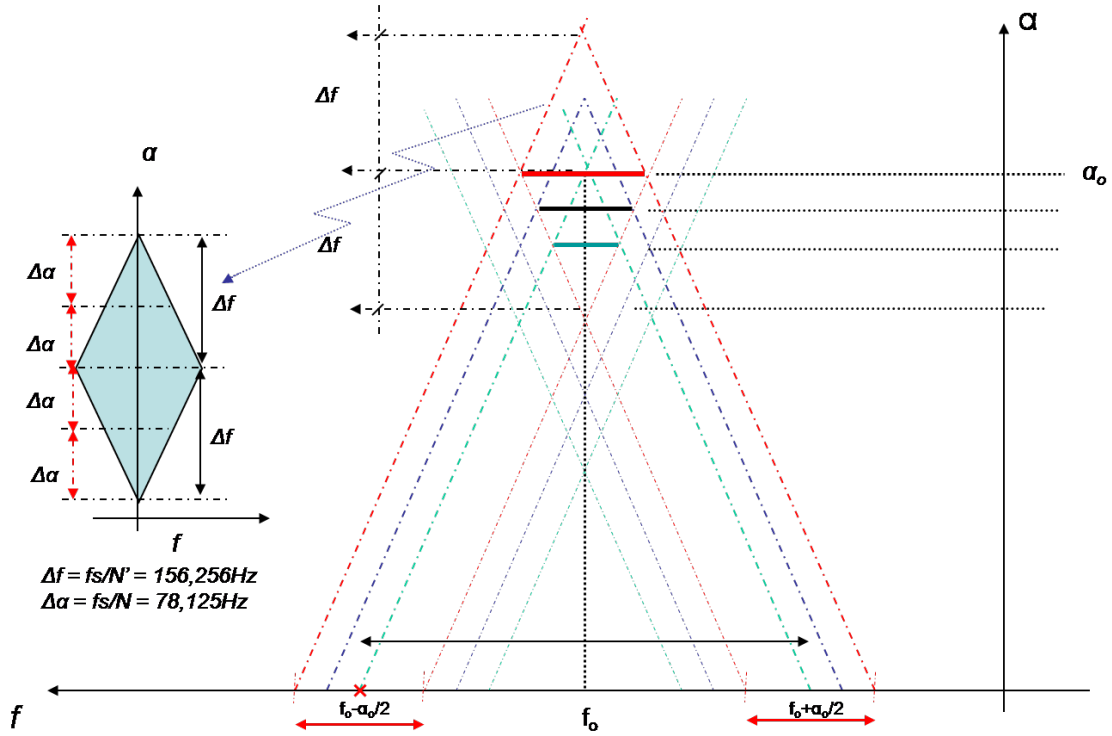


Figure 26. Bi-frequency plane

**P windows of length T**

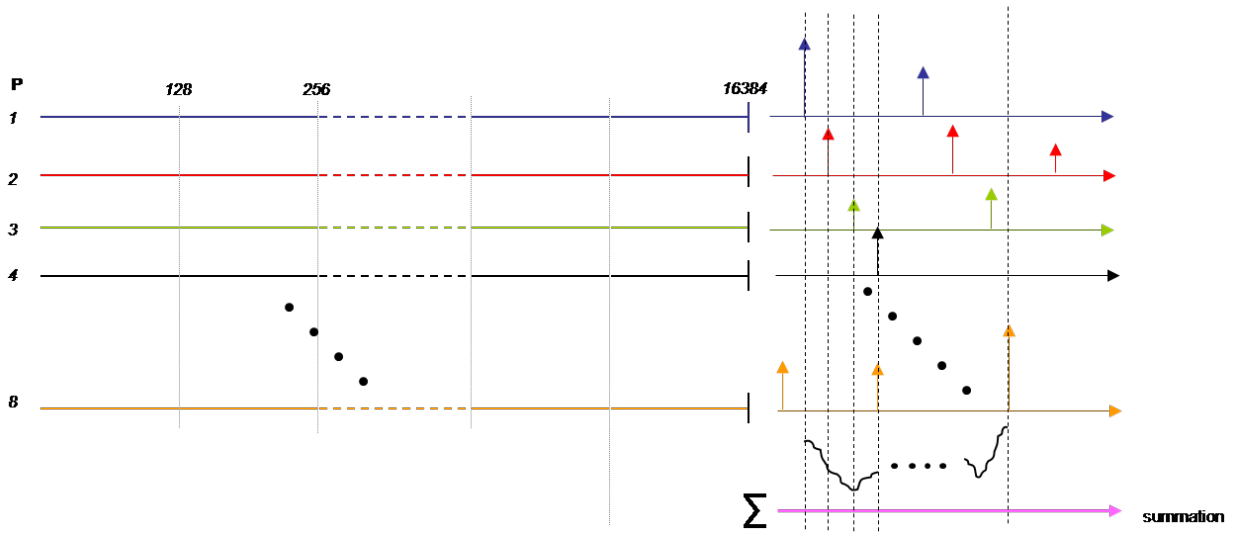


Figure 27. FFT for a summation

Fig. 27 shows what down sampling and hopping FFTs mean for the FAM algorithm. The  $P$  number appears due to the  $L$  step size of hopping  $T$  over  $\Delta t$ . This  $P$  segments have to added

considering that they are shifted on frequency between each other. Each window is depicted by different colors and at the end it is possible to observe that the summation has a resolution clearly due to the number of windows ( $P$ ) included in the operation.

This kind of de-phase summation is well accomplished by means of an FFT. This is the reason why the second FFT appears in the FAM algorithm and has size  $P$ .

Now let's define the environment described until this point for the discrete implementation. Segments of size  $N$  forming the signal  $x[n]$  are captured. Inside this segment, a hopping FFT is performed of size  $N'$  corresponding to  $T$ . The hop size  $L$  determines a  $P$  de-phase summation to be performed.

$P$  sub-segments of size  $N'$  are formed out of the  $N$  size segment hopping by  $L$ . The sub-segments are passed by the Hamming window narrow filter and then through a size  $N'$  FFT that generates two branches that are shifted by multiplying them by the oscillator. One of these branches is passed through the conjugate operation. The next step is the cross multiplication of these two branches that generates an  $N' \times N'$  result. This happens for each of the  $P$  windows of size  $N'$  in order to complete the treatment of the size  $N$  segment ( $\Delta t$ ).

The description above lets us also know the way the bi-frequency is filled out. Including the zero center of the plane, frequency domain extends for  $\pm f_s / 2$  and  $\pm f_s$  for the cyclic frequency  $\alpha$ . It is then  $N'+1$  for  $f$  axis and  $2N'+1$  for  $\alpha$  axis.

Fig. 28 described the process of filling the bi-frequency domain. A zoomed version of one of the calculated SCF for some specific  $\alpha_0$  and  $f_0$  is depicted. Each block of  $N'$  results coming from the result vector of  $N' \times N'$ , represent two lines size  $N'/2$  (blue dotted) where vertical aligned points are the result of cross multiplication centered at  $f_0$ . The reason of the existence of the size  $N'/2$  and its representation is that resolution of  $\alpha$  axis is twice the resolution of  $f$ , for the example used.

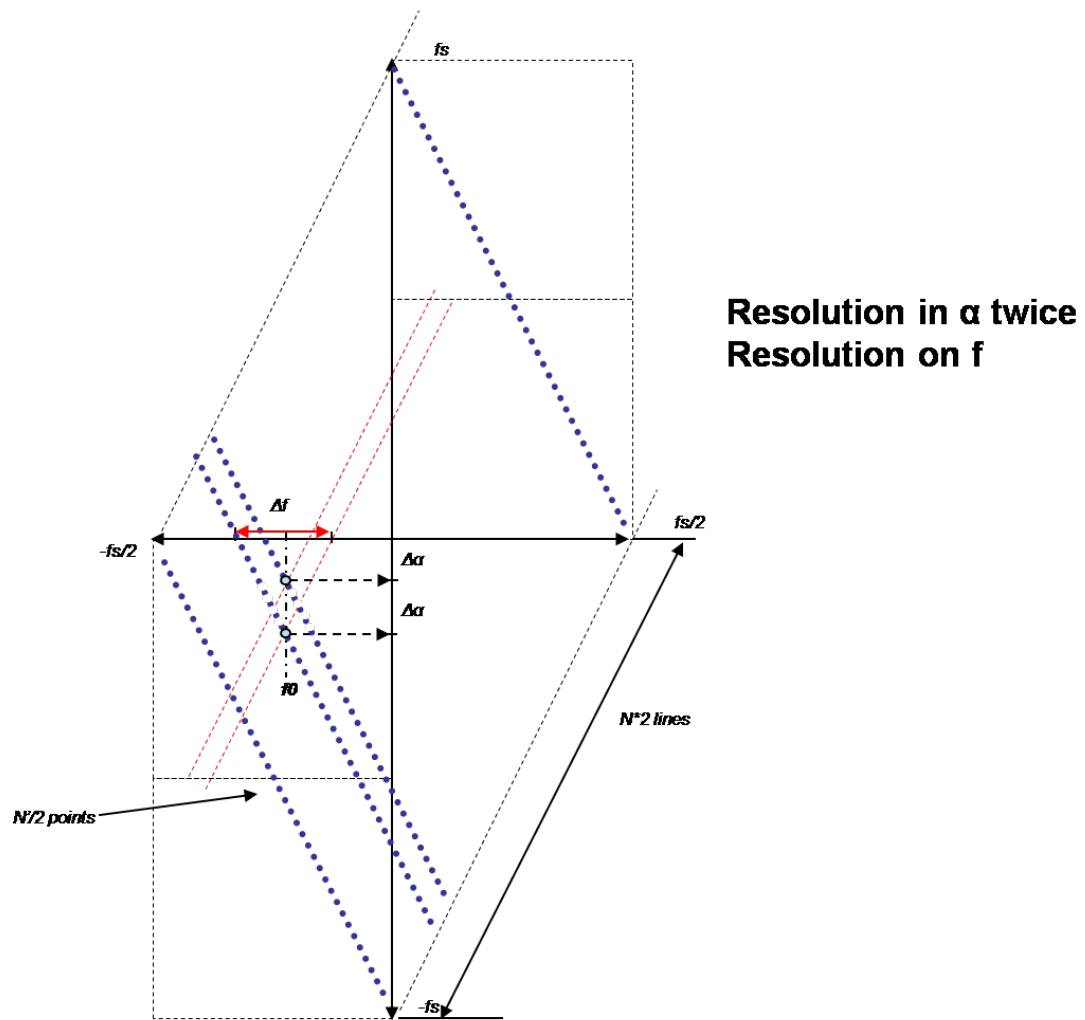


Figure 28. filling the bi-frequency

#### 4.6 Applying FAM for OFDM signals

Cyclostationary detection presents a valuable and viable tool when energy detection algorithms are not sufficient to detect users over the scanned bandwidth. There is no doubt that in terms of the computational complexity, cyclostationary detections is more demanding when compared with energy detection; the resolve time could very well meet many applications real time requirements.

Such applications could consist of signal identification or merely signal presence. It has to be considered data transfer, video, video and voice, video and voice on real time, voice alone, etc. Probably the most complex scenario consists of voice alone running on a Cognitive Radio System that requires continuous awareness of possible primary users ready to claim bandwidth, rapid allocation of new spectral holes and rapid switching of secondary user communication without losing the link.

These types of scenarios remain under study and require more complex assistance than local capabilities as in the case of central-base-station-based strategies. Other applications could be as simple as recognizing empty channels and stepping into as a co-existence rule. It is also true that the spectrum sensing device could be working as out-of-band, independent of the receiver and transmitter duties.

OFDM signals become nowadays a good choice to follow for many communication systems. Wi-Fi, Wi-Max, DVB-T are among of the most recognized. The truth is that OFDM offers higher data rates and reliability even for mobile devices, of course under certain constraints.

The structure of OFDM is almost common between the different applications. There is a preamble, a cyclic header, framed data packages of variable or even fixed length, and inserted pilot signal to help synchronization at the receiver. Modulation of several carriers under BPSK, QPSK or N-QAM is also part of OFDM design. Usually pilots and preambles are modulated in BPSK or QPSK.

This OFDM structure and the modulation involved contain cyclostationary features that would allow detecting and even recognizing signals in the spectrum.

Data is of random character but the presence of pilots is almost fixed although different strategies are used among different OFDM schemes. Pilots are fixed in the sense of its presence in the signal spectrum and the power used on them. For example, Wi-Fi uses one scheme to place its pilots and they are present on every data frame that forms a variable packet size although their value is pseudo-random. Wi-Max has two different schemes to place the pilots according of the

operation mode, others applications vary the location between data frames although it is also cyclic.

Cyclostationary detection search for these repetitive patterns and according the application the target could be any of them requiring more or less efforts. For example reference [45] uses the pilots of OFDM to perform signal identification between Wi-Fi and different Wi-Max modes. Since pilots are fixed in the spectrum and contain more power, they should highly protrude on the final SCF over data and noise. But, the existence of preamble, whatever its values or modulation are, could overcome the pilot energy shown on the final SCF, burying them with data. So, this work extracts the preamble and cyclic extension before applying cyclostationary detection, process that requires finding the beginning of the frame and some synchronization. It uses a correlation with a local version of the preambles for Wi-Fi and Wi-Max.

But some other applications look for signal presence and not signal identification as a first target. Correlation in time domain will tell about the signal with no need to proceed to perform cyclostationary detection, assuming the application can afford correlation and the positive results already show signal presence. Another scenario is when one that doesn't know which kind of signal is around, although some assumptions has to be made, using the fewer possible required to run the spectrum sensing device, search the spectrum and find signal presence.

Again the reliability and the performance of the spectrum sensing device seem to be framed around the application. The application set the requirement of signal detection or signal identification, computation on band or out of band, real-time or empty band detection on access time (co-existence rules). Real time definition set basic boundaries to meet on top of above detailed.

Audio with video applications could afford delays on arrival time as long as they arrived together. An audio alone application that is more exigent in terms of real time is a wireless microphone in front of an audience.



There is now a clear interest to use 802.11 Wi-Fi to transmit voice WVOIP (Wireless Voice Over IP) and the respective committee for standards even releases the new standard 802.11r that set a maximum of 100ms to deal with hand-off between access points that route voice.

Real time voice over 802.11 could be one of these applications that require adjustment of layers as transportation, MAC and the physical one (ISO tower model).

The current size of data packets, made of a number of data frames, defined for Wi-Fi 802.11g is variable with a maximum of 4096. Larger packets mean higher payload data rate but if errors are detected the whole packet has to be retransmitted. Real time voice cannot afford such retransmissions and would make use of smaller packets. These could be even of a fixed size according rate of digital conversion (voice quality) and SNR.

This thesis shows how the cyclostationary spectrum sensing device could target the preamble existence avoiding the correlation needed to work with just data and pilots.

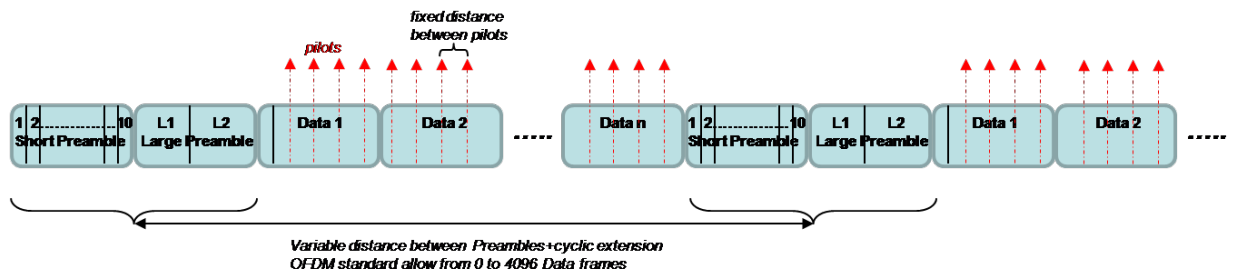


Figure 29. OFDM 802.11

Fig. 29 shows the structure of an 802.11g packet. The standard defines two preambles, a variable number of data frames but each of them holding the 4 pilots defined. The estimation of SCF averages the presence of random data, fixed pilots and preamble appearance. Pilot presence should protrude over data, preambles present some random character due to the fact of a variable number of data frames (0-4096).

The random character of the preamble appearance would make them appear or not appear in the block oriented FFTs and so in the average of the SCF calculation of the incoming signal. This

means that preamble cyclic frequency is moving around the cyclic spectrum, and in so, messing the average calculation all over the range.

The test performed with FAM implementation show the strong presence of preambles on the final SCF, even over the pilots with no doubt. It is the random characteristic that works against its use on the SCF as a first glance. If this randomness could be reduced then preamble should become the best signature of signal presence.

#### 4.7 FAM test bench

To apply the FAM device some assumptions has to be made in relation with the spectrum where the signal appear. For example let's work with an 802.11g signal that was down converted to base band, 20 MHz bandwidth, and 20 MHz sample rate on the sensing device. FAM parameters as frequency and cyclic frequency resolutions would be defined to meet such signals.

$$\Delta t \Rightarrow N = 256, \text{ define } \Delta f = 20 \text{ MHz} / 256 = 78.125 \text{ KHz}$$

$$T \Rightarrow N' = 128 \text{ define } \Delta \alpha = 20 \text{ MHz} / 128 = 156.25 \text{ KHz}$$

$$P = 8 \text{ hopping FFT step size } 32$$

Cyclostationary detection is based on average and this is why results would be enhancing with the number of times the SCF estimation is run. More runs reduce the presence of randomness.

To show the presence of pilots and preambles in the SCF, two tests were run on the FAM device.

On the first one, the preambles and cyclic extension were extracted by previous correlation synchronization. The second one includes preamble and cyclic extension as in the whole 802.11.

The signal under test 12dB SNR, has a fixed number of data frames of 8, 4-QAM and the average size for the SCF is equal to 80. FAM performs the estimation over 80 (average size) segments 256 samples length  $N(\Delta t)$ .

The top part of Fig. 30 corresponds to the SCF estimation for the first case, no preamble no cyclic extension. The 2D graph is at  $f = 0$  and the 3D shows the whole bi-frequency plane and arrows

pointing to the pilots that prompt into the 2D graphic. The bottom part shows the case where the whole signal is included, pilots are still above the data but preamble & cyclic extension appear as 12 big peaks.

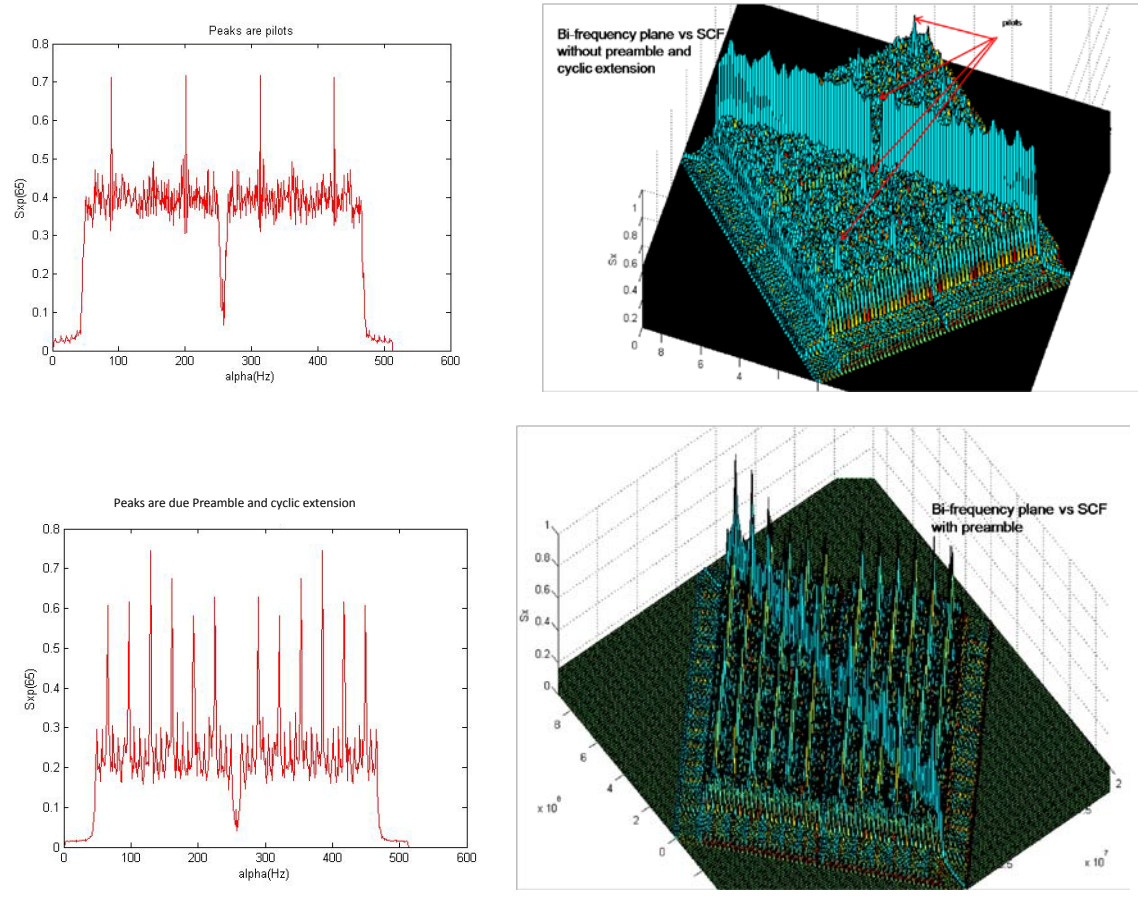


Figure 30. FAM SCF estimation

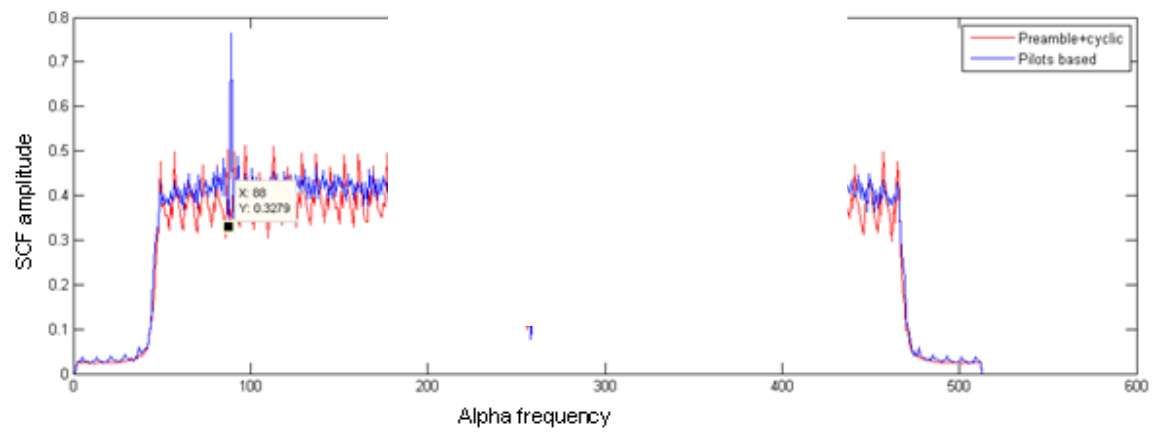


Figure 31. Comparison with extracted preamble

Fig. 31 is a FAM test for an 802.11g signal with variable number of data frames, random sizes from 0 to 4096. The red one is the SCF estimation when the whole signal is considered; the blue one has no preamble and cyclic extension. The average size is 300 segments of  $N$  (256) size.

#### 4.8 FAM Performance Analysis

FAM performance could be analyzed by defining a spectrum sensing device capable of outputting a value that indicates the presence of a particular signal like 802.11. The output has to be a ratio metric defined since wireless signals does not assure fixed signal energy levels.

A good choice for the device output could be the summation of the peaks where preambles should appear and also include the pilot's peaks.

**Device Output:** Setting  $f_0=0$  from bi-freq plane we get the graph below  
From this the output is =  $4\text{Pilots}+12\text{Preamble}$  divided by all Data (others).  
The center one is discarded.

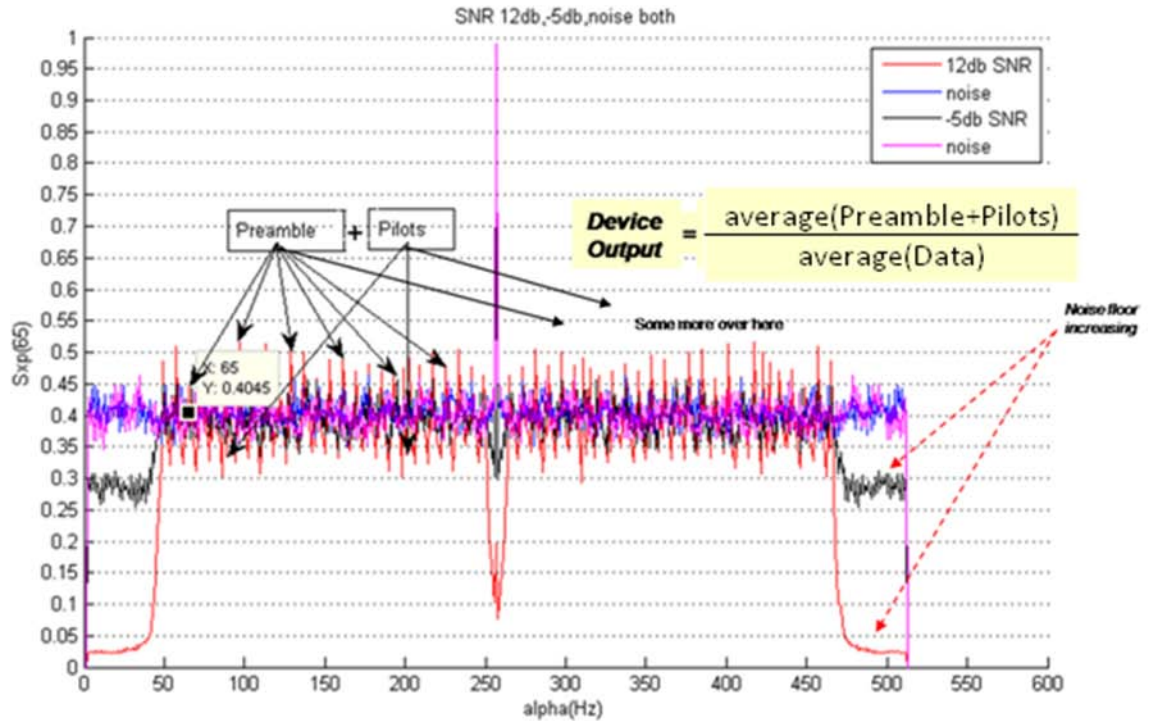
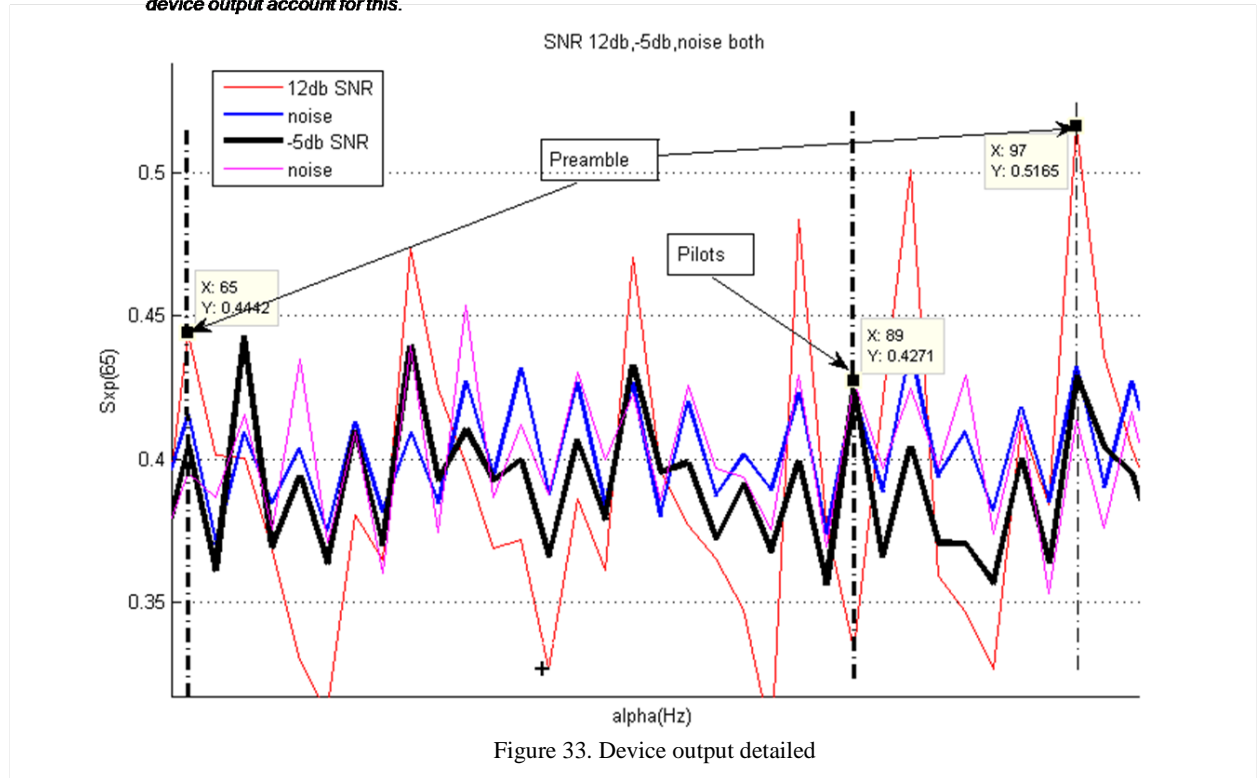


Figure 32. FAM spectrum sensing device

Fig. 32 shows the device output details running the FAM for 12dB, noise alone and -5dB SNR.

Next Fig. 33 is a zoom view of Fig. 32 that show the benefits of the ratio metric approach.

*The zoom shows that comparing with noise level, magnitud of pilots and preambles at -5db SNR, could be even below noise (blue). But for -5db the Data carriers are also below noise. The ratio metric character of the device output account for this.*



Now that our device is defined it will be put under test. It is a good idea to test the device for the signal absence, noise alone in order to establish a base reference. Then, it could be tested for a range of SNR that allows us to finally set a threshold for signal detection.

Recall that performance of the device is a trade-off between average sizes and the randomness, defined as the variability of the number of data frames. The test designed should involve these parameters.

Average size was defined as the number of  $N$  (256) samples to be used on the SCF estimation or device output. This is reflected as the time needed for the device to collect enough information to estimate SCF, let's call this "Estimation- time". For example in OFDM 802.11 it means:

$$\text{Estimation-time} = \text{average-size} \times 256(1/20 \text{ MHz})$$

If average-size is equal to 500 then:

$$\text{Estimation-time} = 500 \times 256(1/20 \text{ MHz}) = 6.4\text{ms}$$

In other words the device won't be able to give an output before 6.4ms and computation time is not included.

For sake of reference too, Fig. 34 shows results of an Energy Detection algorithm running for different SNR and noise alone using the same 802.11 vectors that are used later in FAM.

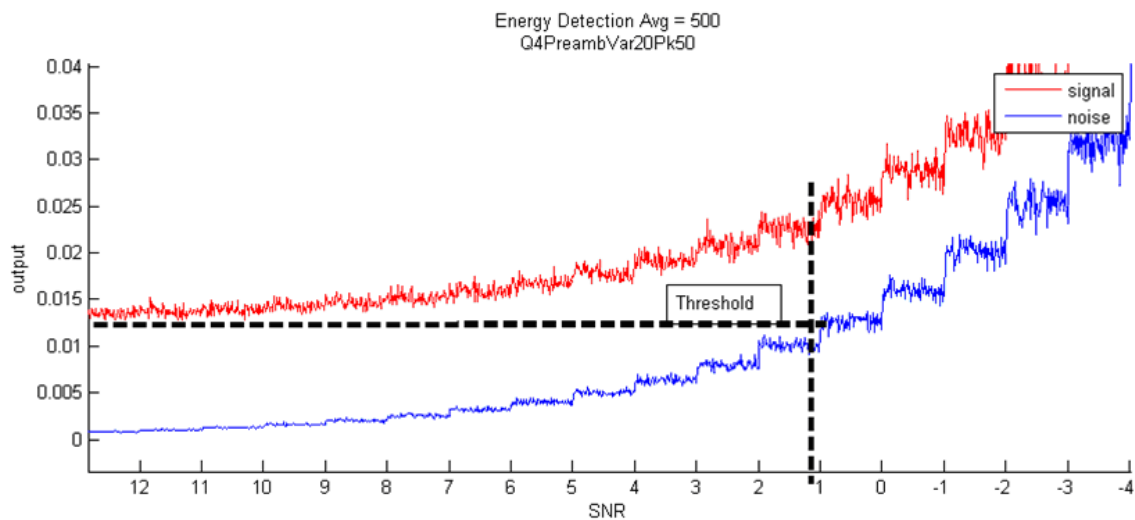


Figure 34. Energy detector

The output of this Energy Detector shows that if a threshold has to be defined for signal detection, the energy based detection algorithm is capable of helping us above 1dB SNR. Below this point, it is not able to differentiate between noise and signal.

Fig. 35 is the FAM output for an 802.11 signal made of 50 packets that include variable number of data frames (0 to 4096). It is a big file and the average-size is the length of the file.

The graphics shows that even under -10dB SNR, FAM is able to help and establish difference between signal and noise.

For a useful, viable implementation average-size has to be reduced (trade-off). So the next test results focus on average-size and also on the variability, randomness of the number of data frames, or even for a fixed size.

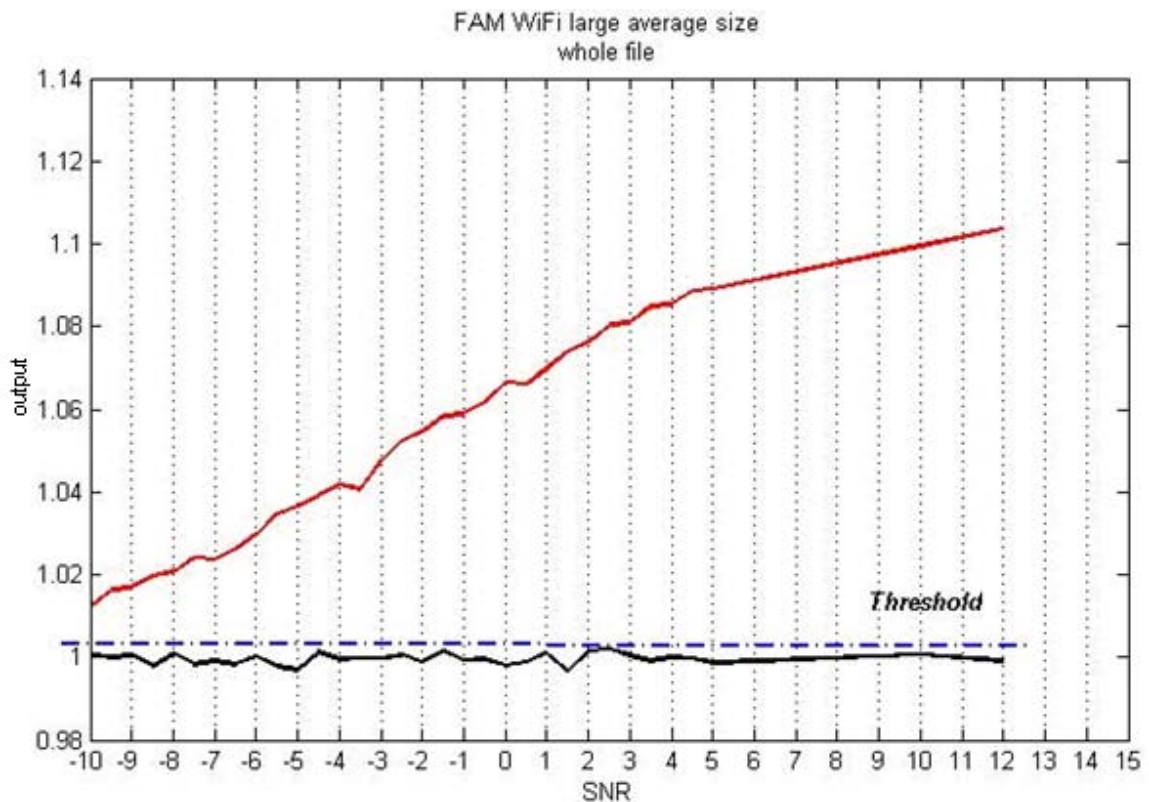


Figure 35. FAM output

#### 4.8.1 Average size, number of data frames vs. SNR

Fig. 36 shows device performance for a variable number of data frames (0-4096) run for different average-size. As it was expected performance increases with the Estimation-time. The depicted noise allow us to set a threshold that for average-size above 300 could make the device go as far as -8dB of SNR into the signal detection purpose.

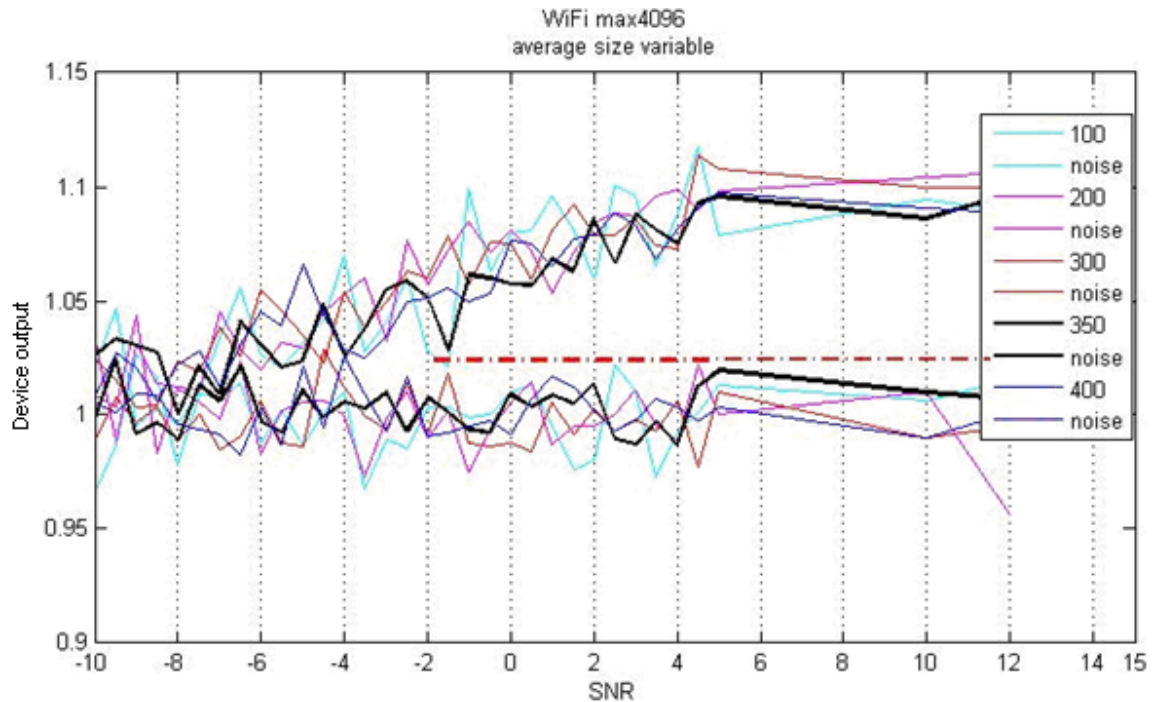


Figure 36. FAM output reducing average size

Estimation –time for average-size 300 is 3.84ms, for average-size equal 400 is 5.12ms and 6.4ms for 500.

For the next test, the variability of the number of data frames is added. A maximum number of frames is set and randomly generate packets with different number of data frames as before. This time the test runs for an average-size of 100 to 400.

Fig. 37 shows the results of the proposed test for a maximum number of data frames equal to 200. It is possible to observe that compared with Fig. 17, curves seems to be more smooth. It shows that for average-size the device could go even close to -10dB.



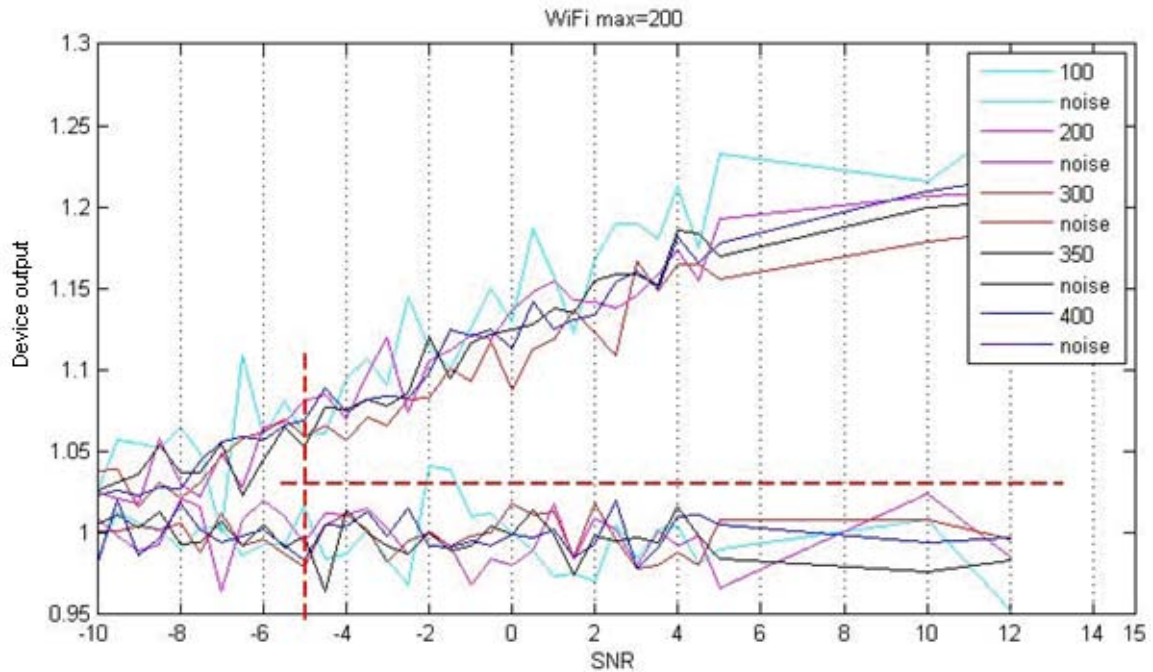


Figure 37. WiFi maximum frame size 200

For Fig. 38 the maximum number of data frames is now set to 50. It is clear how the device output becomes even more smooth, allowing to set a threshold that go further than -10dB SNR.

From this graphic we could pick up an average-size of 100 with Estimation-time of 1.28ms and be able to detect signal presence below -10dB SNR.

There exist applications that could even use fixed number of data frames that are around 20.

There is also no need to go maybe down below -5dB. These two reasons allow us at this point to think that such cyclostationary detector is achievable and not really expensive in terms of computation and Estimation-time.

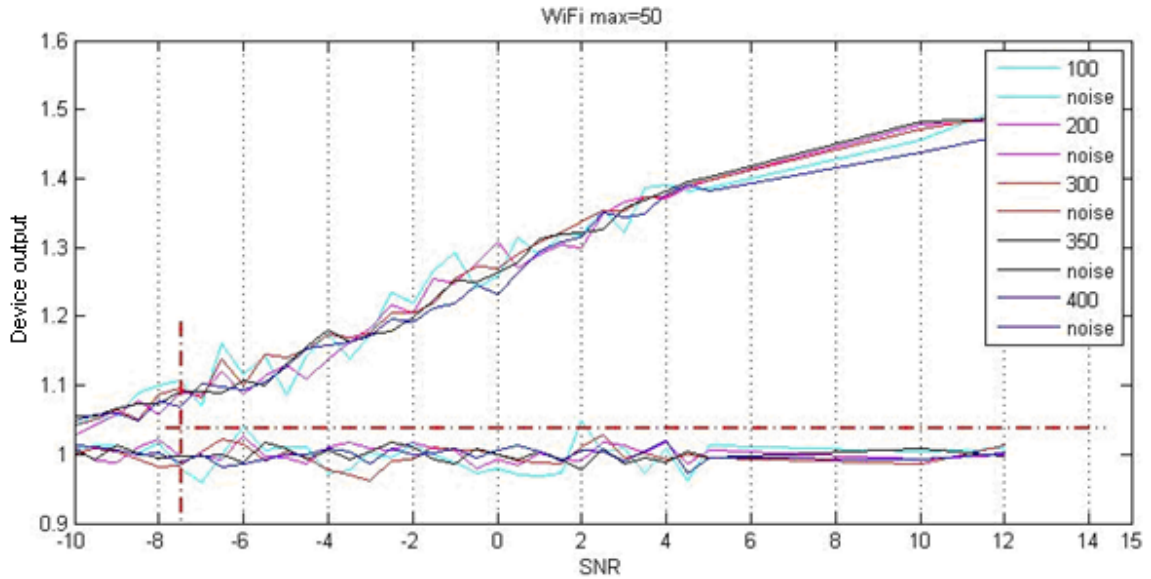


Figure 38. WiFi maximum frame size 50

#### 4.9 Narrow the Target

The FAM used above works over the whole spectrum defined by the 20 MHz sample rate under Nyquist criteria. But if we expect that primary users appear with 802.11 Wi-Fi signals (in example), the FAM could be set to run its calculations just for smaller range of the spectrum like where some of the relevants appear.

Recall that there is a cross-multiplication involved in the SCF estimation that implies  $N' \times N'$  (i.e. 16,384) multiplications. If we are not interested in other frequencies than the selected range where the peaks appear, then we can make them zero. Any zero value multiplication won't be performed reducing the above described multiplication process.

Many others strategies at many different levels of the implementation could be applied, but this multiplication reduction shows a “cheap enough” FAM. Let's call this “single FAM” and see how the performance looks.

Fig. 39 shows in red the portion of whole FAM that is used by the single FAM to estimate SCF. Fig. 40 shows that just few of the peaks we are looking for, are considered in the average calculations as well as number of data peaks involved.

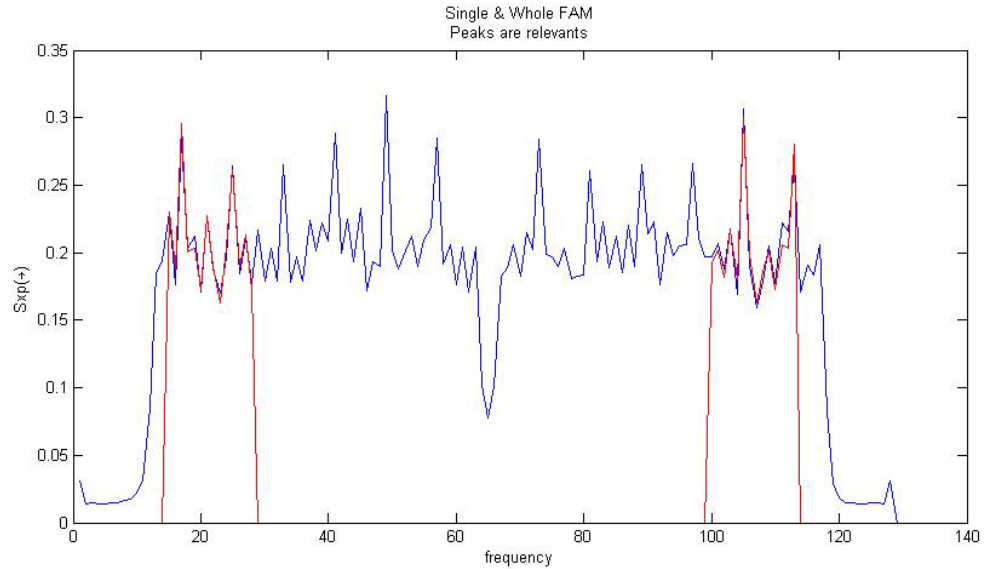


Figure 39. FAM single vs. whole

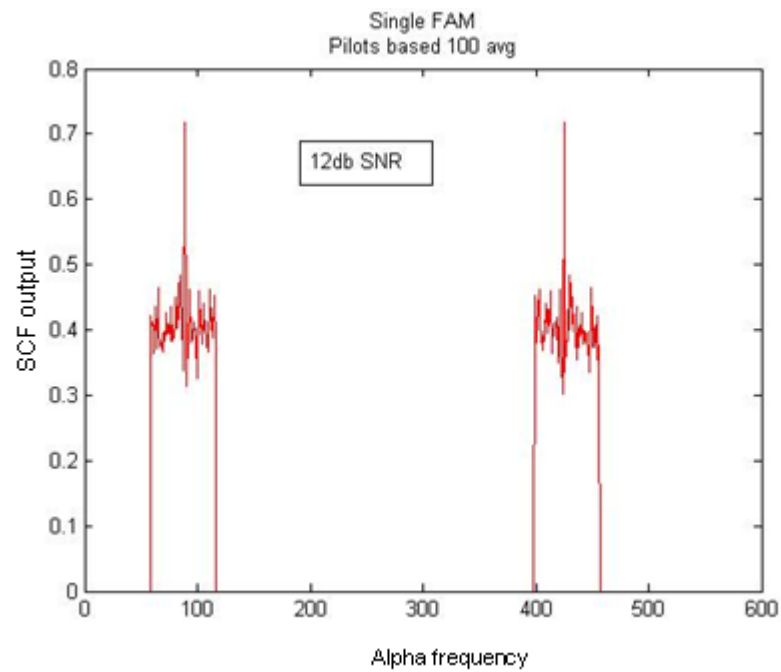


Figure 40. Single FAM SCF output

Fig. 41 shows the performance of the single FAM for the large vector file, variable number of data frames (0 to 4096), that make 50 packages and SNR from -10dB to 12dB.

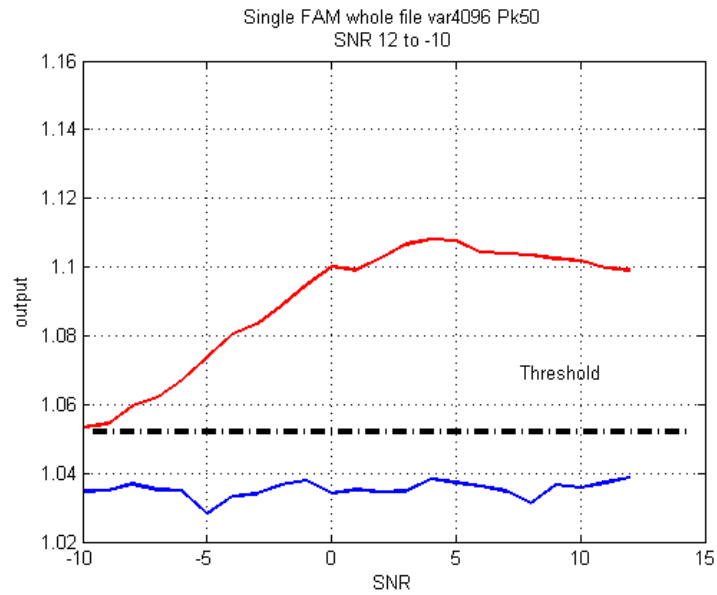


Figure 41. Single FAM

Now as with the whole FAM, let's apply different average-size values. Fig. 42 shows the results.

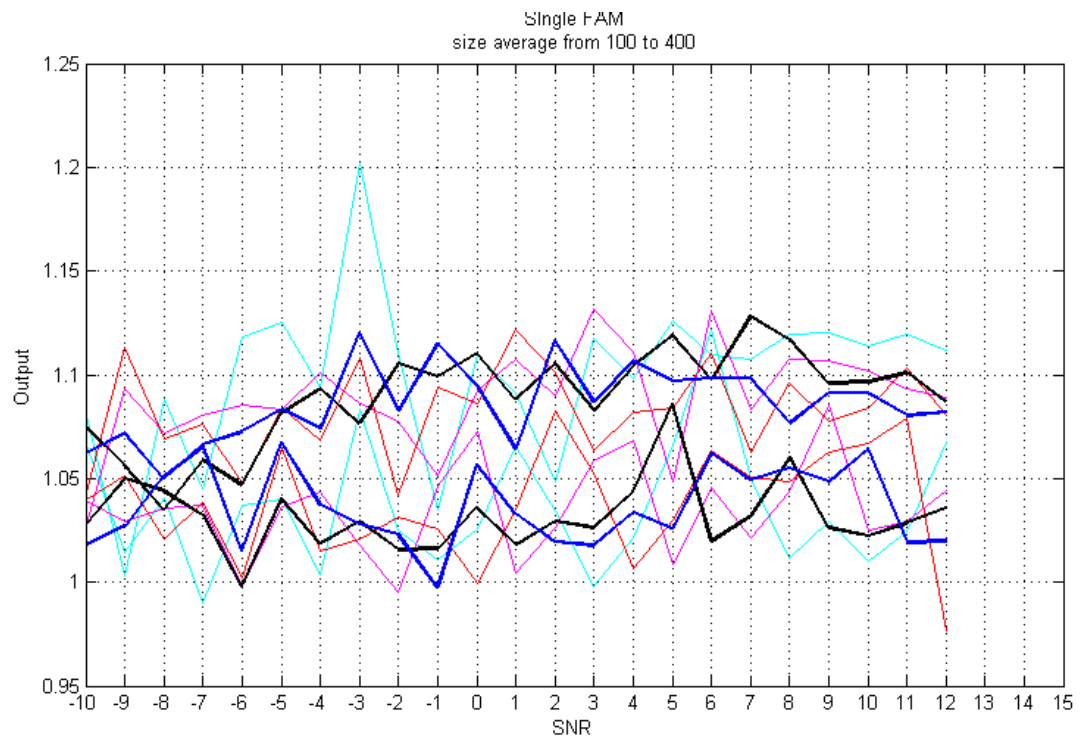


Figure 42. Reducing average size

The results for the Single FAM at average-size from 100 to 400 for a variability or randomness from 0 to 4096 data frames are really poor.

It is expected that if the maximum number of data frames is reduced, the performance will be enhanced.

Fig. 43 shows how the single FAM performs in the test. It looks better for an average size of 200 and 300 and for the cases of maximum number of data frames of 50 and 200. The threshold is still really good and could give us signal detection over noise for -5dB in one case and more than -9dB in the other.

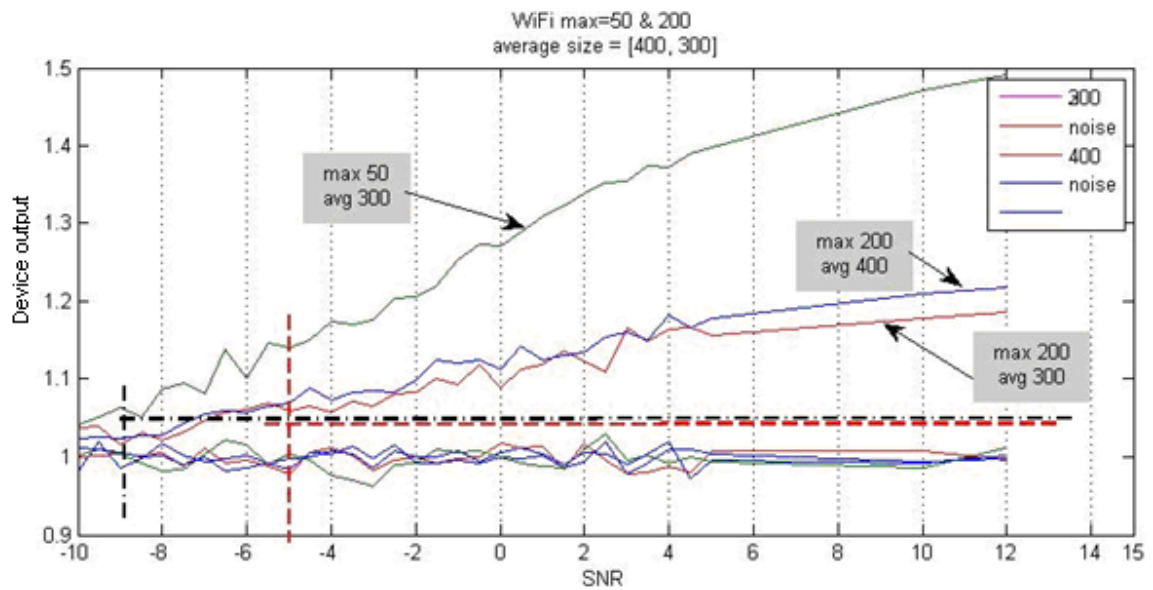


Figure 43. Average size comparison

#### 4.11 A voice application

To have a better idea of the viability of using cyclostationary detection on real applications and FAM performing, a voice application is described.

A wireless microphone would normally require the best latency achievable since for a live performance the audience has to find congruence between the moving mouth and the hearing on the speakers. The quality of the voice application as voice alone or with musical instruments set bit rate requirements for the digital modulated versions.

Wireless microphone is one application that currently is close to received some of the benefits that Cognitive Radio systems have achieved so far. There is an intention in Europe of standardize wireless microphones running already under TV spectrum. One of these requirements could be at least co-existence rules. The use of these frequencies requires normally permits around the city and pre-establish them for the duration of the event where the microphone, belt packs or base station used. Football games, artistic performances, etc. requires this kind of permits.

Some microphones run under FM, QPSK, and AFSK. There is also in the wireless inter-communicators market system running in 802.11 under different versions (a,b,g) for voice applications, requiring normally less quality and maybe with less exigent latency than real-time event microphone.

Many of the digital modulated microphones are forced to work in a fixed size framing scheme and even further usually work for a fixed bit rate. The reason is that to perform bit rate adaptation according SNR, both size of the communications have to agree on this. A microphone is by definition a transmitter with no receiver capabilities communicating with a receiver base station. Pre-setting profiles are allowed if bit rate has to be reduced or increase as well as the sample rate for the quantization resolution.

The chosen example is a wireless microphone that use OFDM with some N-QAM (no need to reveal) fixed for the communication time. It works on UHF around 640Mhz, spends some BW and requires a fixed number of data frame of 20.

The OFDM has its proprietary definitions for FFT length, number of carriers, guard intervals, preamble values and modulation as well its own pilot scheme.

The FAM modifications correspond to change of sample rate, and target peaks to perform the device output.

Fig. 44 shows the results for a whole spectrum FAM under different average-size from 350 to 700. Results seem to be promising even below -8dB SNR.

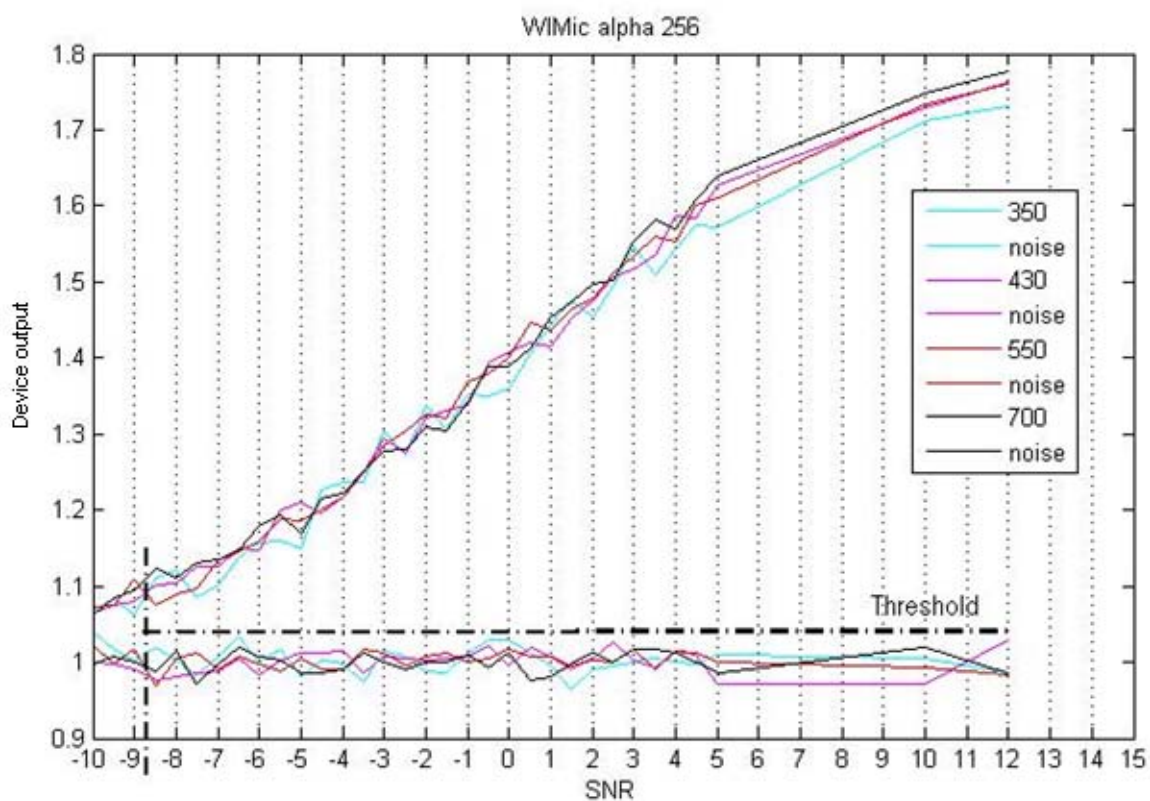


Figure 44. FAM for Microphone

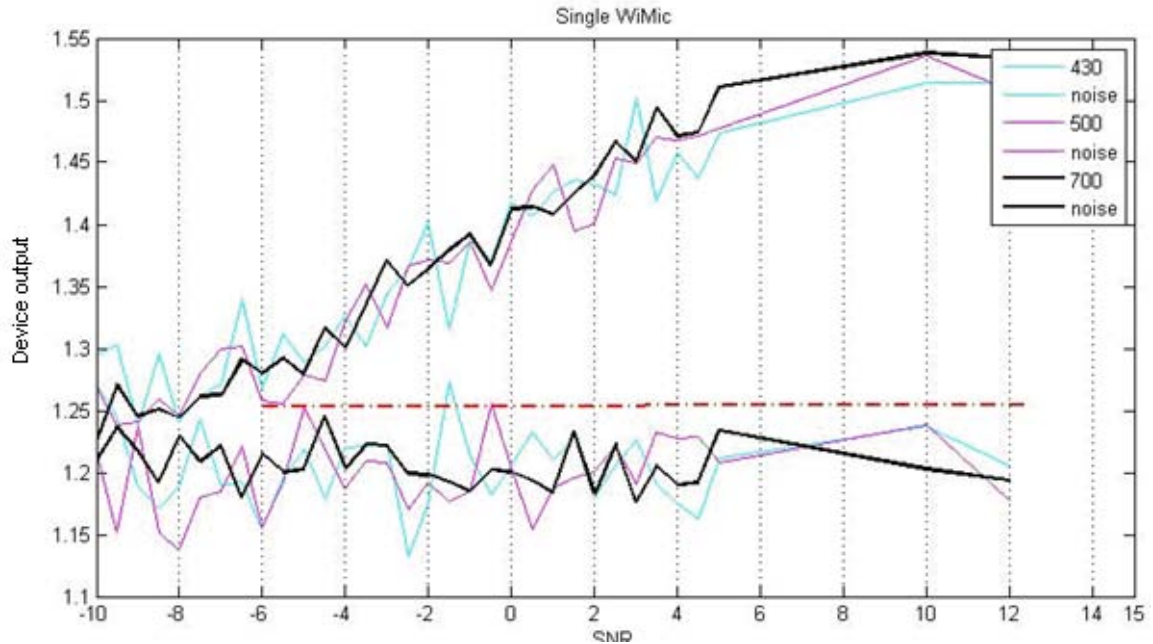


Figure 45. Single FAM Microphone

Fig. 45 shows the application of the properly adjusted Single FAM. It shows that with an average-size of 430 the device make difference between noise and signal above -6db.

Fig. 46 & 47 show the results of applying the device making use of the FAM based on the cyclic frequency alpha centered at 257. With a fixed frame size of 20 for both the 802.11 and the Wireless Microphone, the FAM is performed. The test runs along SNR and different average sizes.

It is possible to observe that the device output is clearly defined for both at even lower average size, although the WiFi signal shows lower range.

Recall the pilot schemes are different and also the preambles, some differences are expected. The pilot amplitude relative to the data carriers is also a key factor to have in mind, since it would tend to protrude on the SCF.



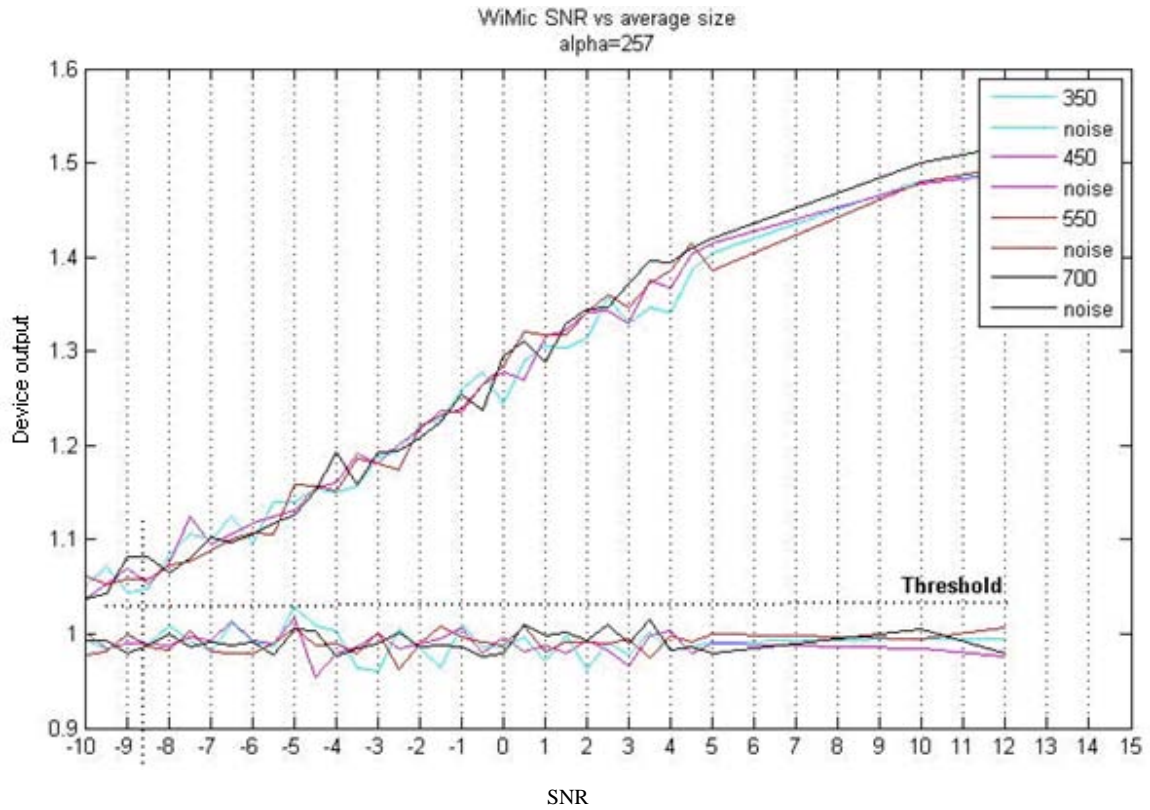


Figure 46. Average size

From the entire test performed some conclusions could be detailed. The FAM would contain relevant peaks that could be formed by the frequent appearance of the preamble, pilots, and of course the cyclic extension would add to it too. The fixed pilot scheme like the one used in 802.11 will have more presence than a hopping pilot scheme. But the preamble design could have a strong autocorrelation and under variable random frame size, it will spread over the cyclic frequency. Power increase on pilots help the FAM detection even over preambles, this maybe considering fixed frame size.

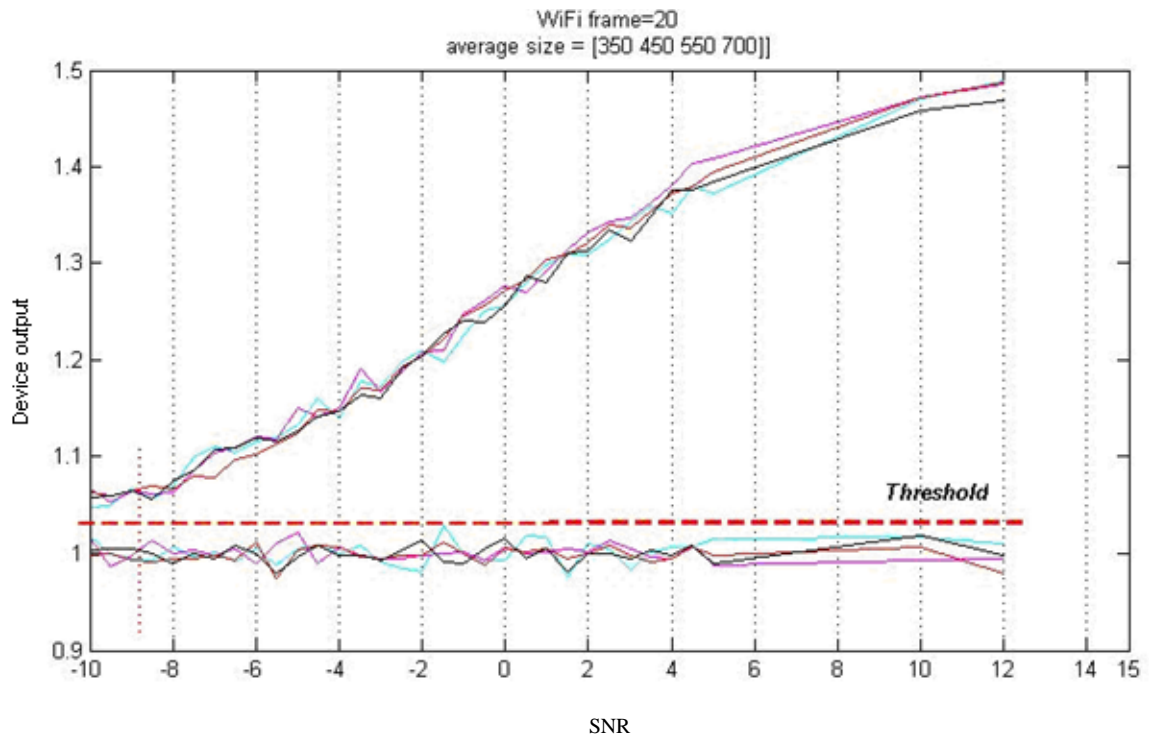


Figure 47. FAM microphone

## 4.12 FAM Computation

FAM computation requires two FFT stages and a correlation performed by a cross-multiplication. The first FFT has a size of  $N'$  and the second one of size  $P$ . The cross-multiplication produce a vector of size  $N' \times N'$ . There are even more multiplications involving down conversion of complex conjugates, summations and conjugate operation.

Looking for efficiency of implementation and computation at this level, would have to be framed under the chosen resolution for frequency and cyclic frequency. These conditions require to chose values for  $N$  and  $N'$  large enough to achieve reliability. Once  $N$  and  $N'$  are set the average-size would have to be set to a value that allow a good signal detection above some SNR.  $N'$  Set the size of the first FFT.

The second FFT of size  $P$  comes from the hopping FFT that result in a down sample operation. Fig. 8 shows the significance on the final average. This is usually a small FFT and close observation and further test could show that at some point, increasing  $P$  not apply useful contributions.

Then for the general case where the whole spectrum has to be use for SCF estimation, the next concerns is the correlation and as usual the FFT implementation.

The work reference in [17], focus on these two concerns using the concept of sliding FFT calculation and something called one-bit correlation.

The FFT calculation is normally efficiently designed, since digital communication systems as OFDM make use of such module for receiver and transmitter duties. In many cases in order to share resources the FFT machine is used by both receiver and transmitter circuits.

Anyway a shared FFT machine or a dedicated one for the spectrum sensing device could be use. Either it is a flexible FFT machine capable of work for different sizes; it is implemented on the DSP by firmware or is running into a separate FPGA unit.

Below the cost in computation based in complex multiplication is detailed [20]:

Data tapering:	$2N' P$
$N'$ point FFT:	$PN' \log_2 N'$
Frequency shift:	$2N' P$
Cross Multiply:	$P(N')^2$
FFT product sequences :	$(N')^2 (P/2) \log_2 P$

## Chapter 5

### FAM vs. frequency offset

#### 5.1 Frequency offset in FAM algorithms

The radio receiver requires frequency synchronization to be able to decode the incoming data. There are many approaches with some trade-offs for each of them but certainly all of them required knowledge of physical layer scheme. Normally some features like preambles, are included in the signal to facilitate such procedures and apply maybe correlation based on targeting i.e. encoded preamble.

As it was expressed before, the final application would certainly frame the spectrum sensing design as it is for signal presence detection all the way to signature recognition. Cyclostationary detection requires knowledge of some parameters related to sample frequency and the range of frequency at the spectrum to be targeted.

A shift on the spectrum of the frequency domain should be expected if any offset exist between transmitter and receiver. Cyclostationary detection based device, would experience this offset affecting the relevant peaks position, therefore the calculation of the ratio used in this thesis between the magnitudes of peaks vs. data would be miscalculated.

To avoid this situation, frequency offset detection and posterior frequency offset correction processing blocks, could be needed (designed and included).

Saying again that application wise the cyclostationary detection could go further on the receiver chain and then rely in the frequency detection and correction blocks already used on the receiver.

If the application is maybe an independent spectrum sensing unit, maybe force to work out-of-band and maybe a more open device tunable to work at different parts of the spectrum and with

no previous knowledge of possible signal scheme coming to help synchronization (i.e. preambles), then there is an additional issue to solve.

The results arrived in this thesis show that some improvement could be applied to the spectrum sensing device to cope with the frequency offset problem and even add a new frequency offset detection system, coming with this solution.

Fig. 48 shows the SCF for a wireless OFDM system where the pilots were exaggerated in magnitude so they will protrude here in the cyclostationary spectrum. The plot shows in frequency domain for a fixed alpha equal to the center value of  $\alpha=257$ , how the pilots move according to the offset applied with a resolution equal to  $df$  (frequency resolution).

The original position of the pilot that use to be placed and measure at 55units of  $df$ , moves to 52 for and offset  $3*df$  and to 45 at  $10*df$ .

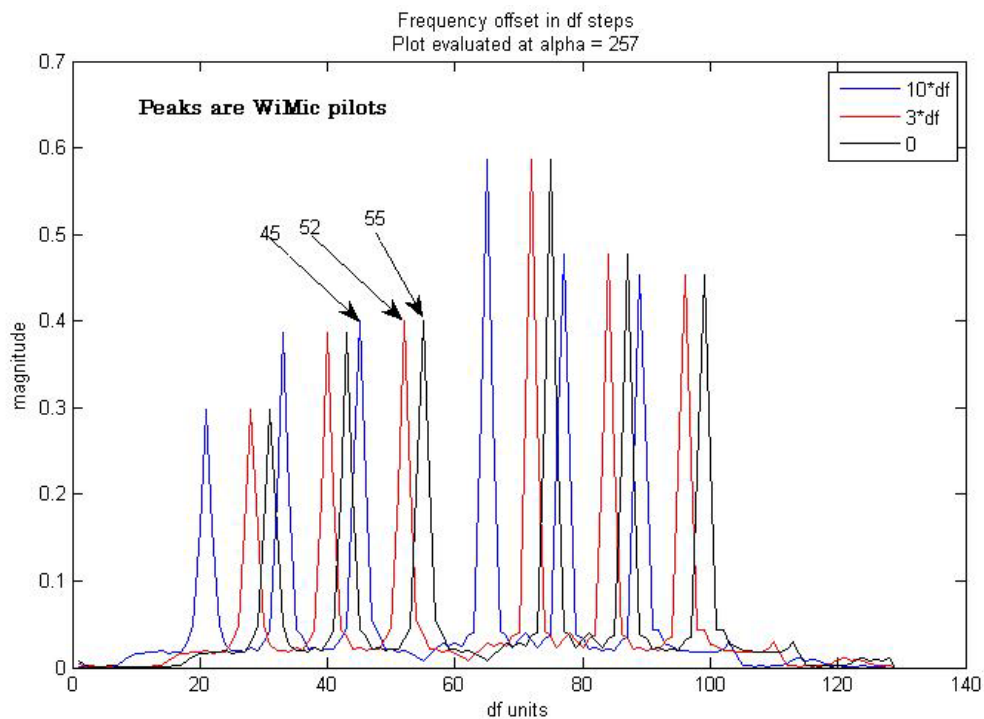


Figure 48

The next Fig. 49 shows the SCF now with the regular pilot amplitude and  $3*df$  offset with no noise and SNR applied for values of -3 and 3 db.

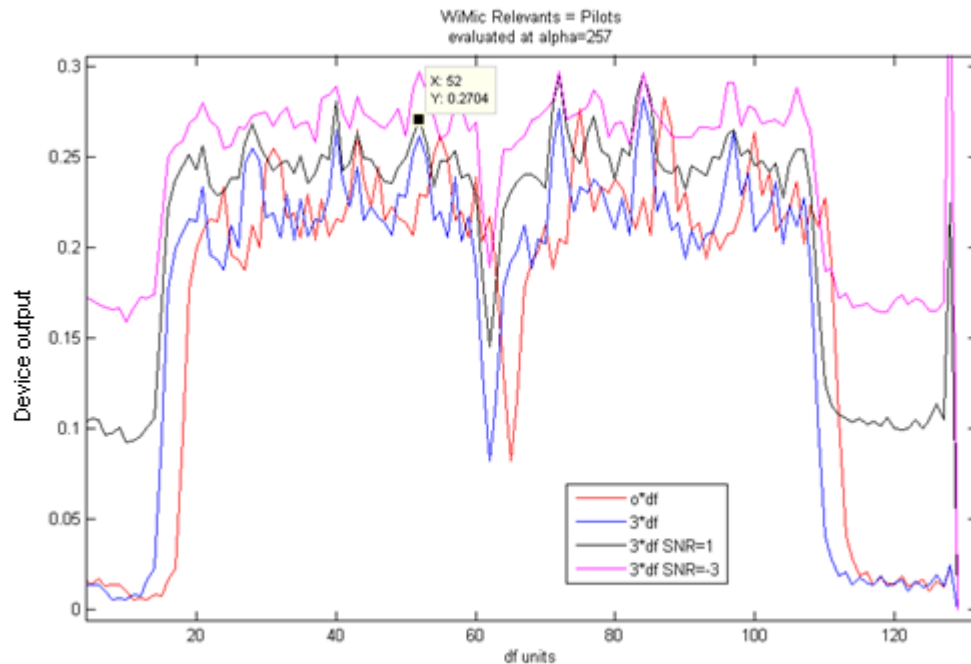


Figure 49. Frequency offset in FAM

It clearly shows how the pilot position at 52 instead of the original 55 is kept and still protrudes at the respective value. It is possible to observe that as the noise increase the level of the whole SCF. But it is also true that the relative distance between pilots remains the same.

Then from both figures it is possible to arrive to some conclusion. The difference between the expected pilot position and the current one, gives the frequency offset and here we have a tool to detect it.

It was found then that the FAM could also give us a frequency offset device with resolution of  $df$ .

In order to right calculate the spectrum sensing device output, this new peaks location have to be known. Locating one of them would allow knowing the others as the relative distance remains the same.

Then is not necessary to correct the frequency offset and it is possible to recover from this if a new algorithm is applied to detect the new peaks position, as find the max values at fixed distance between each other.

Maybe the approach should consider that if the offset is big enough some of the peaks may disappear on the side. Then consider less peaks and relative less data on average could be a solution.

DC carrier at the center could be part of this improved algorithm to reference and recognize in example, the first maximum peak value as the first pilot to the left. Now we have a new layer of detection on the spectrum sensing device that considers distance between relevant peaks on the SCF.

More intelligence could be included or even offset correction applied before SCF estimation. But it looks that these improvements would be paid with more computation in a system already costly in this sense.

## 5.2 Frequency Offset estimation and correction

It is clear then that FAM is affected by a frequency offset by shifting the relevants and still keeps the relative distance between them. But for an offset big enough, then some of the relevants will be push out of the SCF estimation. Even though smarter algorithms could be applied to detect and work with the remainder relevants, maybe a frequency detection block and correction could be used to simplify the spectrum sensing device output. Again it is also another example where the application itself will decide. In example, when the sensing device works out of band as an independent unit.

There are many strategies to frequency offset detection with some trade offs, but something common for an OFDM signal comes from its cyclic extension components. Although different OFDM designs have their own preambles to help time and frequency synchronization, these models usually count with the cyclic extension by OFDM symbol.

The idea behind this blind estimation is correlate the incoming symbol having in mind the sample size of the cyclic extension and its distance with the replica at the end of the OFDM symbol. The correlation will peak when the copy is found.

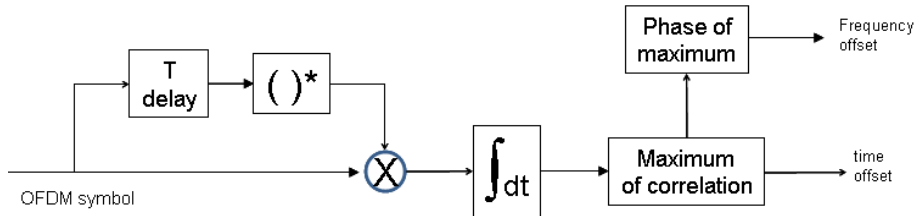


Figure 50. Block diagram

The peak position will give the time offset inside the receiver buffer of size 2 OFDM symbols. The angle of the complex output value at the peak found, gives the frequency offset. A complex multiplication of the OFDM symbol will allow correcting the incoming signal.

The weakness of the algorithm comes of the fact that the data based correlation of this cyclic extension is random, and then the correlation peak won't be a stable value per OFDM symbol. Also the incoming preamble will affect the output of the algorithm if its own cyclic extension correlation performs close or far to the data correlation peaks.

Also the algorithm is constrained to frequency offset detection in the range of  $\pm \text{carrier-distance}$ . It is also necessary to perform some averaging on time over several symbols to improve the correlation output, making it of some latency, which delays response on cases of fast frequency changes like the fast fading situation.



Anyway this becomes a very good approach for many applications where fine frequency detection is required and weakness are affordable either by the application or some other blocks supporting it.

Topic 6.3 includes the Matlab code of the results shown in this chapter. Fig. 51 detail the correlation output for a WiFi kind OFDM structure.

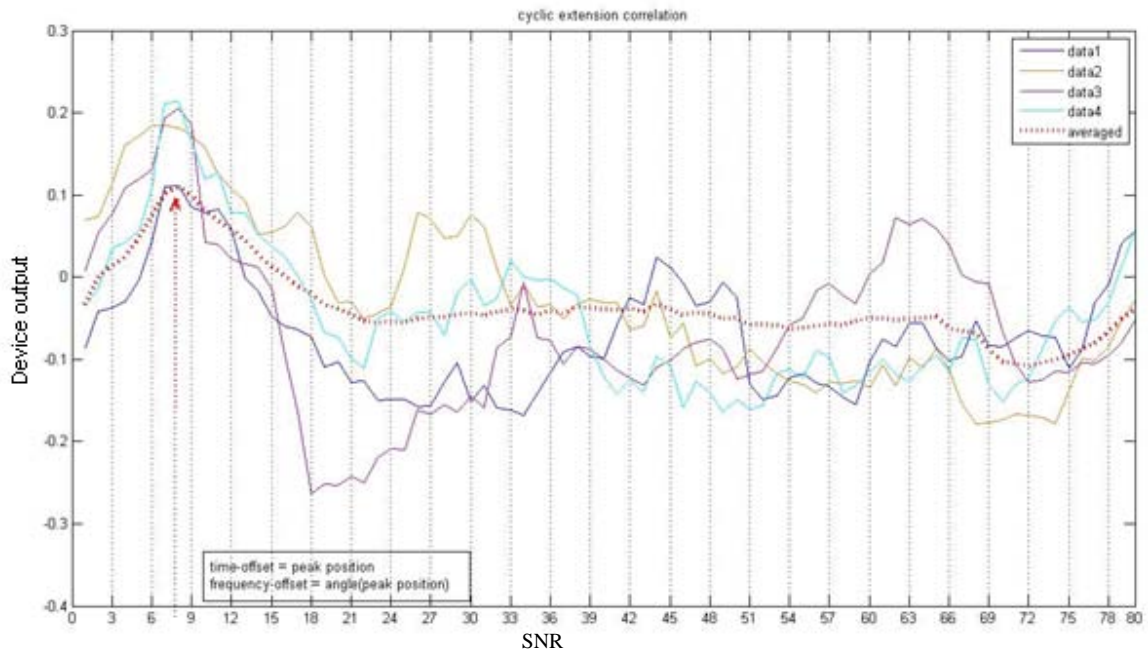


Figure 51. Correlation output

The averaging will reduce the noisy output. For some cases where the preamble is highly correlated with itself at its whole length, then a flat output of the correlation could spoil any peak detection. One way to cope with this is instead of averaging the peak position at the buffer, better average the whole output of the correlation. The peak finding over this average result will be more stable and even takes care of any flat result too. The price to pay is a buffer to hold the average correlation although it could be any smart average implementation.

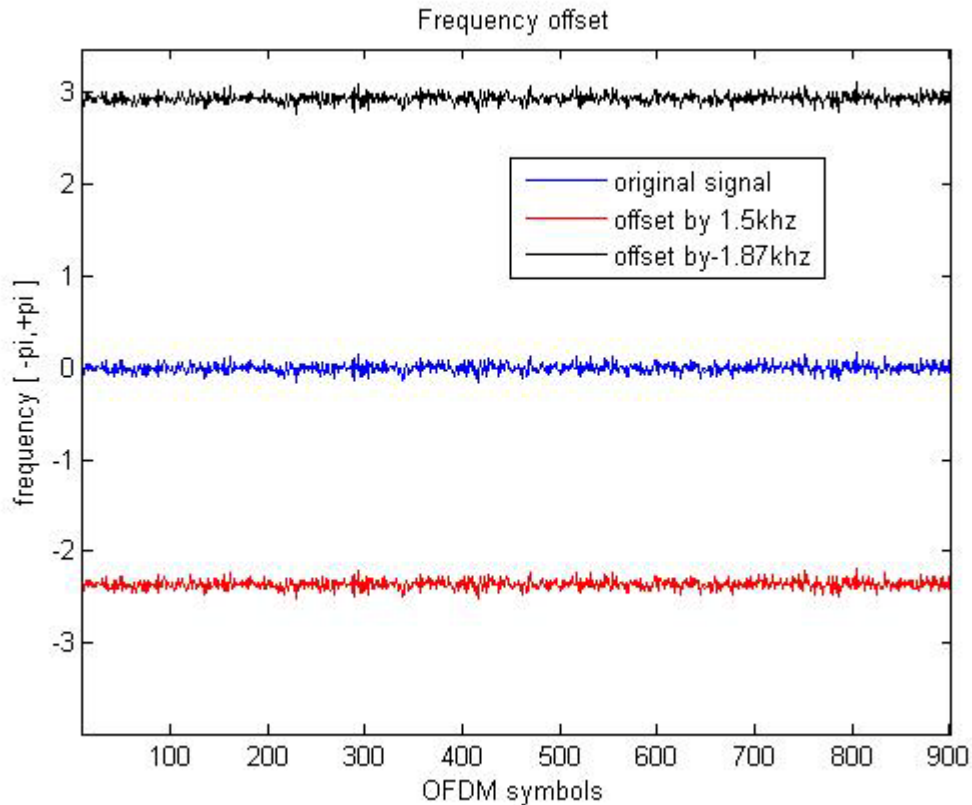


Figure 52

If the signal has no time offset it will happen that the max peak of the correlation will appear half around position one and half around the end of the buffer. Then the maximum finding will be jumping between sample 1 and the last sample. If this situation is not avoided, then the algorithm will get confuse. One way to handle this is force the ADC to re-start getting samples by giving a new start to run to the clock. Since DMA is normally involved turning off and back this process is usually enough to fall any other place in buffer. It is also possible to not start any process until maybe i.e. the peak falls close to the optimum position at the center of the buffer. It is Optimum in the sense of the latency for the receiver application.

### 5.3 How and where to reduce computation

It has been discussed that in order to reduce FAM computation a single FAM version could be used to target some specifics relevant and reduce the overall calculation. But the frequency offset needs to be found to relocate this single FAM.

FAM computation is mainly based on correlations and FFTs, which increase in size, will improve resolution. Any improve on FFT calculation will lead to a considerable reduction on calculation time and MIPS.

One way to do it is the use of the Sliding-FFT which is suited to target specific frequencies instead of the whole spectrum.

#### 5.31 The Sliding FFT

The Sliding FFT calculates an N point DFT of window of samples sliding on time domain. The algorithm start calculating a DFT, then the time window slide one sample and again the DFT is calculated. But the new calculation is based in the previous DFT value, leading a highly efficient overall calculation.

The base of this algorithm is the well known shifting theorem that state that if  $X(k)$  is the DFT of a sequence  $x[n]$  and if this sequence is shifted by one sample then the DFT of the resulting sequence is the previous sequence spectral components, multiplied by  $e^{j2\pi k/N}$ .

$$S_k(n) = e^{j[2\pi k/N]} [S_k(n-1) + x(n) - x(n-N)]$$

The expression above state the Sliding FFT procedure being  $k$  the bin position. Then for a single bin target Fig. show the implementation based in a comb filter and a resonator.

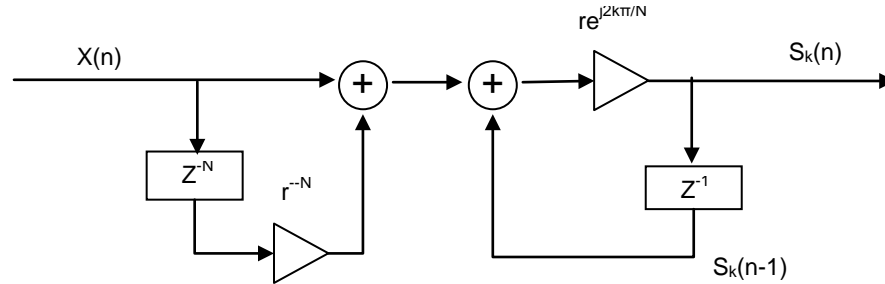


Figure 53. Block diagram

Now if the complete  $N$  points DFT has to be computed, then  $N$  resonators will be needed tied to a common comb filter.

The computational load is reduced to  $O(N)$  compared with  $O(N^2)$  of the regular DFT and  $O[N\log_2(N)]$  of the FFT.

Then is ease to note that target single specific bins (frequencies) is possible with such structure. This characteristic plus the reduced computation required makes the Sliding DFT very attractive to the idea of creating the relevant peaks for the SCF.

In order to give stability to the Sliding DFT Fig. 53 includes  $r$  which is value close to the unit. To reduce the spectral leakage some modification are applied to this original design. A Hanning window on frequency domain facilitates the computation.

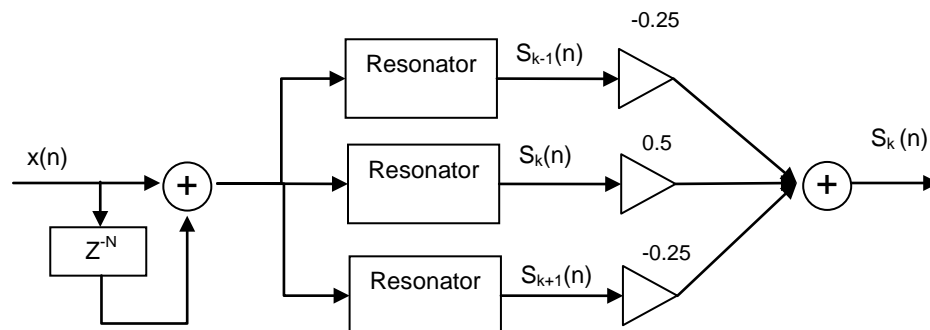


Figure 54. Block diagram

### 5.32 One-Bit Correlation

This a technique based on Bussgangs's theorem which in synthesis replaces complex multiplications with sign changes. This advantage and the data multiplexing operation involved are more appreciated in hardware implementation on FPGA or ASIC where greatly simplifies size and complexity.

It is recognized that correlation computation can be simplified by clipping one of the correlated signals to  $\pm 1$ , which in turn becomes just sign information. So the multiplications of the correlated will become sign changes of the other signal.

The Bussgang theorem state that the cross correlation between two signals could be equal to a scaled version of the correlation of one of the signals with a transformed version of the other according to a non linear memory less fashion.

$$R_{uv}(0) \cong R_{v'v}(0)R_{uv'}(0)$$

This is valid for Gaussian process and the components on frequency domain highly approximate stationary and Gaussian.

$$\Phi[v] = \frac{1}{\sqrt{2}} [\text{sign}(\text{Re}[v]) + j\text{sign}(\text{Im}[v])]$$

The expression above is the transformation used for the purpose. If this is rotated 45 degrees then the complexity is reduced to a sign results [-1,-1]. Then we have now a redefinition of the FAM equations.

$$S_{xy_{\Delta t}}^{\alpha_o}(n, f_0)_{\Delta f} = \frac{b(f_0 - \alpha_o/2)}{\Delta t} \langle X_{\Delta t}(n, f_0 + m/\Delta t + \alpha_o/2) \cdot (\Phi[Y_{\Delta t}(n, f_0 + m/\Delta t - \alpha_o/2)]e^{j\pi/4})^* \rangle_{\Delta f}$$

$$b(f_0 - \alpha_o/2) = \frac{1}{2} \langle \Phi[Y_{\Delta t}(n, f_0 + m/\Delta t - \alpha_o/2)]e^{j\pi/4} \cdot Y_{\Delta t}^*(n, f_0 + m/\Delta t - \alpha_o/2) \rangle_{\Delta f}$$

Simulation studies results could be find in reference [15].

## References

- [1] F.Weiding, D.Datla, "A framework for RF spectrum measurements analysis", in Proc. IEEE Int. Symposium on new Frontiers in Dynamic Spectrum Acces Networks, vol 1 Baltimore Maryland.
- [2] T. Yucek and H. Arslan, " A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications", IEEE Communications Surveys & Tutorials, Vol. 11 2009.
- [3] M.P. Olivieri, G. Barnett, " A scalable dynamic spectrum allocation system with interference mitigation for teams of spectrally agile software defined radios," in Proc. IEEE Int. Symposium on new Frontiers in Dynamic Spectrum Access Networks, Vol. 1 Baltimore Maryland.
- [4] S.Lal and A.Mishra, "A look ahead scheme for adaptive spectrum utilization", in Proc. Radio and Wireless Conf. Boston, Massachusetts.
- [5] A.Ghasemi and E.Sousa, "Collaborative Spectrum Sensing for opportunistic acces in fading environments", in Proc. IEEE Int. Symposium on new Frontiers in Dynamic Spectrum Acces Networks, vol 1 Baltimore Maryland.
- [6] G. Zheng, N. Han, "An Enhanced Energy Detector for WRAN Systems Using Maximum-to-Mean Power Ratio", IEEE ISWCS2007
- [7] D. Cabric, "Addressing the Feasibility of Cognitive Radios", IEEE Signal Processing Magazine [85] November2008
- [8] Akyildiz, Won-Yeol Lee, M. C. Vuran, "A Survey on Spectrum Management in Cognitive Radio Networks", Georgia Institute of Technology
- [9] B. Ackland, D. Raychaudhuri, "High Performance Cognitive Radio Platform with Integrated Physical and Network Layer Capabilities", Rutgers University
- [10] W.A.Gardner, "Exploitation of Spectral Redundancy in Cyclostationary Signals", 1999
- [11] W.A.Gardner, "The spectral correlation theory of cyclostationary time series", Signal Processing - July 1986
- [12] W.A.Gardner, "Measurement of spectral correlation", IEEE Trans. Oct. 1986

- [13] W. A. Brown, "Digital Implementation of Spectral Correlation Analyzers", IEEE Trans. Oct. 1993.
- [14] T.Zhang, G. Yu, C. Sun, "Performance of Cyclostationary Features Based Spectrum Sensing Method in A Multiple Antenna Cognitive Radio System", IEEE Trans. WCNC 2009
- [15] W.A Gardner and R.S. Roberts "One-Bit Spectral-Correlation Algorithm", IEEE Transactions on Signal Processing, vol. 41 1993
- [16] Annamaria R., Varkonyi-Koczy, "A Recursive Fast Fourier Transformation Algorithm", IEEE Transactions on Circuits and Systems Vol 42, No. 9 1995.
- [17] G. Yu-Long, C. Yan, "Modification and Digital Implementation of FAM algorithm for Spectral Correlation", Wireless Communication and Signal Processing IEEE 2009
- [18] A. Al-Dulaimi, N. Radhi, "Cyclostationary Detection of Undefined Secondary Users", IEEE Computer Society September 2009
- [19] R. Roberts, W. A. Brown, H. Lomis, "A review of Digital Spectral Correlation Analysis: Theory and Implementation" IEEE Press,Pages 38-48, April 1991
- [20] R. Roberts, W. A. Brown, H. Lomis, "Computational Balance in Real-Time Cyclic Spectral Analysis", Proceedings of the 1994 International Conference in Acoustics.
- [21] D. Noguét, L. Biard, and M. Laugeois, "Cyclostationarity Detectors for Cognitive Radio: Architectural Tradeoffs", EURASIP Journal on Wireless Communication and Networking, Volume 2010. Hindawi Publishing Corporation and Networking.
- [22] P. D. Sutton, K. E. Nolan, "Cyclostationary Signatures in Practical Cognitive Radio Applications", IEEE Journal on Selected Areas in Communications, Vol. 26 No. 1 2008
- [23] W. A. Gardner, "The Spectral Correlation Theory of Cyclostationary Time-Series", In Signal Processing, vol. 11, Nr. 1 1986.
- [24] K. PO and J. Takada "Signal Detection Based on Cyclic Spectrum Estimation for Cognitive Radio in IEEE 802.22 WRAN System" Technical report of IEICE.

- [25] W. A. Gardner, "Exploitation of Spectral Redundancy in Cyclostationary Signals" IEEE Signal Processing Magazine, pp. 4-35, April 1991.
- [26] D. Cabric, S. M. Mishra, "Implementation Issues in Spectrum Sensing for Cognitive Radios", Proc of Asilomar conference on signals, systems and computers, Nov 2004, Vol. 1
- [27] W. A. Gardner, "Measurement of Spectral Correlation", IEEE Transactions on Acoustics, Speech and Signal Processing Vol. No. 5 1986.
- [28] R. Roberts, W. A. Brown and H. Loomis, "Computationally Efficient Algorithms for Cyclic Spectral Analysis, IEEE SP Magazine.
- [29] J. Wang, T. Chen, "Cyclo-Period Estimation for Discrete-Time Cyclostationary Signals", IEEE Transactions on Signal Processing, Vol. 54 No.1 January 2006.
- [30] W. A. Brown, H. Loomis "Digital Implementation of Spectral Correlation Analyzers", IEEE Transactions on Signal Processing, Vol. 41 No.2.
- [31] W. Yue, B. Zheng, "Spectrum Sensing algorithms for primary detection based on reliability in cognitive radio systems", Computers and Electrical Engineering journal, ELSEVIER. 2009.
- [32] Mahmoud, Tevfik and Arslan, "OFDM for Cognitive Radio: Merits and Challenges", IEEE Wireless Communications, Vol. 16 Issue 2, 2009
- [33] Part 11: Wireless frequency division multiplexing (OFDM) PHY specification for the 5Ghz band, IEEE Std. 802.11-2007
- [34] S. Hwan, N. Han, J. Moun, J. Wan, "OFDM Signal Sensing Method Based on Cyclostationary Detection", IEEE 2nd. International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications.
- [35] T. Yucek and H. Arslan, "Spectrum characterization for Opportunistic Cognitive Radio Systems", Proceeding MILCOM 06 2006 IEEE conference on military.
- [36] R. K. Sharma and J. W. Wallace, "A novel correlation sum method for a cognitive radio spectrum sensing", Proceedings of XXIXth General Assembly of International Union of Radio Science, Chicago, Aug. 7-16,2008.



- [37] N. Kim, M. Kim, E. Kim, "Robust Cognitive-Radio-Based OFDM Architecture with Adaptive Traffic Allocation in Time and Frequency", ETRI Journal, Volume 30. 2008.
- [38] Negi and Cioffi, "Pilot Tone Selection for Channel Estimation in a Mobile OFDM", IEEE Trans August 1998.
- [39] Z. Zheng, "Robust channel estimation for OFDM-based WLAN systems with imperfect synchronization", IEEE Trans. In Communications, Vol. 46, 2008
- [40] Jose de Souza Lima, "Dissect PA Distortion from OFDM Signals", News RF magazine article 22326.
- [41] J. E. Kleider, X. Ma, A. Smith, "OFDM transmitter and receiver performance improvements for small factor SDR hand-held radios", 2008 SDR forum.
- [42] K. Fazel S. Kaiser, "Multi-Carrier and Spread Spectrum Systems", John Wiley Publications.
- [43] J. H. Reed, "Software Radio: A Modern approach to Radio Engineer", Prentice Hall.
- [44] Hou-Shin, W. Gao, D. G. Daut, "Spectrum Sensing for OFDM Systems Employing Pilot Tones", IEEE Transactions on Wireless Communications, Vol. 8 No.12 December 2009.
- [45] S. R. Schnur, "Identification and Classification of OFDM based signals using preamble correlation cyclostationary feature extraction", Naval Postgraduate School, Monterrey CA 2009.

# Appendix

## Matlab Implementation

### A.1 OFDM Transmitter and Receiver 802.11g

```

% -----
% Initial values
% -----
global bits_per_sym
random = 1;
if random==2

fp1=fopen('nr.dat','r');fp2=fopen('ni.dat','r');fp3=fopen('nb.dat
','r');
    s_r = fscanf(fp1,'%f\n',[inf])'; %real branch
    s_i = fscanf(fp2,'%f\n',[inf])'; %imaginary branch
    file_bits = fscanf(fp3,'%f\n',[inf])'; %imaginary branch

    indice=1;
end

data_carriers = 48;
bits_per_sym = 6; %N-QAM
QAM_N = bits_per_sym;
bit_error=zeros(1,1);
% -----
% Basis functions, sample points per symbol interval
% -----

N_FFT = 64;
BW = 20e6;
Ncarriers = 64;
Carrier_Spacing = BW / Ncarriers;% Carrier Spacing
TFFT = 1/Carrier_Spacing;%Time of FFT

T = TFFT/N_FFT;
%Guard band
G = 0;%1/4;% 1/8, 1/16, and 1/32
delta = G*TFFT; %guard band duration

Tshort = 10*TFFT/4;
Tlong = TFFT/2 + 2*TFFT;

fc = 2.4e9+40e6; %802.11g 2.4Ghz
tx_clock = 20e6;%4*fc/20;
up_clock = 4*fc; %Upsample clock

```

```

NOISE_ALONE = 0;
N_FRAMES = 20;
N_PACKETS = 200; %Number of packets of OFDM analyzed
UPDOWNConv = 0; %UPDOWNConv = 1 to implement Up-Conversion.
Memory needed
PILOT_FAM = 0; %1 = Pilots based, 0 = Preamble + Cyclic
extension
RECEIVER = 0; %bypass receiver
VARFRAME = 0; %makes N_FRAMES useless
MAXVAR = 4096; %max size of random size of N_FRAMES
AVERAGE_SIZE = 0;%size of average for FAM,overules the whole
input signal
pmagnitude = 1;
% -----
% Map received gains to symbol bits
% -----
if (bits_per_sym==2)
    % 4-QAM table
    SGain = sqrt(2);
    qam_table=(1/SGain)*[-1 1];

end;

if (bits_per_sym==4)
    % 16-QAM table
    SGain = sqrt(10);
    qam_table = (1/SGain)*[-3 -1 3 1];

end;

if (bits_per_sym==6)
    % 64-QAM table
    SGain = sqrt(42);
    qam_table=(1/SGain)*[-7 -5 -1 -3 7 5 1 3];

end;

pilot=[ 1,1,1,1, -1,-1,-1,1, -1,-1,-1,-1, 1,1,-1,1, -1,-1,1,1,...
-1,1,1,-1, 1,1,1,1, 1,1,-1,1, 1,1,-1,1, 1,-1,-1,1, 1,1,-1,1, ...
-1,-1,-1,1, -1,1,-1,-1, 1,-1,-1,1, 1,1,1,1, -1,-1,1,1,...
-1,-1,1,-1, 1,-1,1,1, -1,-1,-1,1, 1,-1,-1,-1, -1,1,-1,-1, ...
1,-1,1,1, 1,1,-1,1, -1,1,-1,1, -1,-1,-1,-1, -1,1,-1,1, 1,-1,1,-
1,...
1,1,1,-1, -1,1,-1,-1, -1,1,1,1, -1,-1,-1,-1, -1,-1,-1 ];
```

%OFDM Generator



```

[arr,carriers0]=IFFT_Block(short_seq);

carriers0 = [carriers0,carriers0,carriers0(1:33)];
carriers0(1)=carriers0(1)*0.5;
carriers0(161)=carriers0(161)*0.5;

% 2. ----- Long symbols -----
-----

TFFT2 = TFFT;
G = 1/2;                                %no guard for preamble
delta = G*TFFT2;                         %guard band duration
TFFT2 = 2*TFFT2 + delta;                  %now we take in consideration the
guard time

long_seq = [ 1  1 -1 -1  1  1 -1  1 ...
            -1  1  1  1  1  1  1 -1 ...
            -1  1  1 -1  1 -1  1  1 ...
             1  1 ...
             0 ...
             1 -1 -1  1  1 -1  1 -1 ...
             1 -1 -1 -1 -1 -1  1  1 ...
            -1 -1  1 -1  1 -1  1  1 ...
             1  1 ];

%----- IFFT -----
-----
% input:      map_complex
% output:     carriers
%-----
-----

[arr,carriers2]=IFFT_Block(long_seq);

%----- Apply Cyclic Prefix -----
-----

carriers2 = [carriers2, carriers2];
carriers2 = [carriers2(length(carriers2)-
31:length(carriers2)),carriers2];
carriers2(1) = carriers2(1)*0.5;
carriers2(161) = -carriers2(1)*0.5;

carriers = [];
Ncarriers= [];
ibits = [];

% 3. ----- Signal Field BPSK and pilots -----
-----

```

```

training_seq =
[carriers0(1:160),carriers0(161)+carriers2(1),carriers2(2:161)];
%These are going to be fixed values anyway
Reserved = -1;
LENGTH = [-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1];
Parity = [-1];
SIGNAL_TAIL = [ -1 -1 -1 -1 -1 -1 ];
Rate = [-1 -1 -1 -1];
Sfield0 = [Rate,Reserved,LENGTH,Parity,SIGNAL_TAIL];
% The convolutional encoding will output 2*24bits rate=1/2
[Sfield1]= Convolutional(Sfield0); %mimic anyway is not physical
layer understood

%----- pilot insertion -----
-----
Sfield_complex = [
                Sfield1(1:5)   pilot(1) ...
                Sfield1(6:18)  pilot(1) ...
                Sfield1(19:24) 0     ...
                Sfield1(25:30)  pilot(1) ...
                Sfield1(31:43) -1*pilot(1) ...
                Sfield1(44:48)  ];

[arr,Signal_Field]=IFFT_Block(Sfield_complex);
Signal_Field = [Signal_Field(length(Signal_Field)-
16:length(Signal_Field)),Signal_Field];
Signal_Field(1) = 0.5*Signal_Field(1);
Signal_Field(81) = 0.5*Signal_Field(81);

% ----- Prepare the preamble = short_seq + large_seq +
signal_Field
preamble = [
training_seq(length(training_seq)),training_seq(1:length(training
_seq)-2)];
%preamble = [
training_seq(length(training_seq))+Signal_Field(81),training_seq(
1:length(training_seq)-1)];

if PILOT_FAM == 1
    preamble = [];
end

% 4. ----- Data generator -----
-----
% output:   input_bits
% generate random data if random = 1 or declare fixed data bits
vector
%-----
-----
G = 1/4;% 1/8, 1/16, and 1/32
delta = G*TFFT;           %guard band duration
TFFT1 = TFFT + delta;    %now we take in consideration the guard
time

```

```

TFFT1 = TFFT1*N_FRAMES;

for N_pack = 1:N_PACKETS
    if VARFRAME == 1
        N_FRAMES = randint(1,1,[1,MAXVAR]);
    end
    data = [];%zeros(1,1);
    Ncarriers = [];
    for nframes = 1:N_FRAMES
        carriers1 = [];
        Data_Generator;

        %----- QAM mapping -----
        %   Map according to bits_per_sym N-QAM output
        %   the output will merge into map_complex
        %   Output : r_branch + li*i_branch
        %-----

        QAM_mapping;
        %----- pilot insertion -----
        map_complex = [ r_branch(1:5)    + li*i_branch(1:5)
pmagnitude*pilot(2) ...    r_branch(6:18)  + li*i_branch(6:18)
pmagnitude*pilot(2) ...    r_branch(19:24) + li*i_branch(19:24) 0
...
...                          r_branch(25:30) + li*i_branch(25:30)
pmagnitude*pilot(2) ...    r_branch(31:43) + li*i_branch(31:43) -
pmagnitude*pilot(2) ...    r_branch(44:48) + li*i_branch(44:48) ];

        %----- IFFT -----
        % input:    map_complex
        % output:   carriers
        %-----
        pilot=circshift(pilot,[0,-1]); % pilot polarity sequence
        [arr,carriers1]=IFFT_Block(map_complex);

        %----- Apply Cyclic Prefix -----
        -----
        if PILOT_FAM == 1 %For FAM purposes
            carriers1 = [carriers1];
            data = [data(1:length(data)),carriers1];
        else
            carriers1 = [carriers1(length(carriers1)-
16:length(carriers1)),carriers1];
            carriers1(1) = 0.5*carriers1(1);
            carriers1(81) = 0.5*carriers1(81);
            data = [data(1:length(data)),carriers1(2:81)];
        end
    end
end

```

```

    end
    %===..... all together to send to transmitter
    .....===
    Ncarriers(1,:) =
[preamble(1:length(preamble)),data(1:length(data))];
    carriers = [carriers,Ncarriers(1,:)];

end % N_PACKETS
PACKET_SIZE = length(Ncarriers(1,:));

clear Ncarriers;clear data;clear carriers0;clear carriers1;clear
carriers2;

% Plot time domain of tx signal
plot_N = 1 ; plots();
% to display partially use below to replace carriers

    % TFFT1 = 0;TFFT2 = 0; carriers = carriers0; %simulate only
short symbols
    % TFFT0 = 0;TFFT2 = 0; carriers = carriers1; %simulate only
data symbols
    % TFFT0 = 0;TFFT1 = 0; carriers = carriers2; %simulate only
long symbols

TFFT_T = N_PACKETS*(TFFT0+TFFT1+TFFT2);
t = 0:1/tx_clock:(length(carriers)-1)/tx_clock;      % now base
time

%Plot each packet of the OFDM frame
%plot_N = 3 ; plots();

% 5. ----- Upconverter -----
-----
%   input:  carriers
%   output: s=real(s_tilde); at 9.6Ghz sample rate = 4*2.4Ghz
%-----
-----

if (UPDOWNConv == 1)
% %       t_up = 0:1/up_clock:TFFT_T; %Sample rate at 9.6Ghz
t_up = 0:1/up_clock:(length(carriers))/tx_clock;      % now
base time
carriers = carriers(1:length(t));
clear t;
%fdac = interp1(t',carriers',t_up)';
fdac = interpft(carriers,length(t_up));
sizeof = length(fdac);
t = t_up;

clear t_up;clear carriers;

```





```

    r_sample = real(r_info(1:(up_clock/adc_clock):sizeof)) ...
               -1i*imag(r_info(1:(up_clock/adc_clock):sizeof));
    %Now recover samples at 20Mhz
    r_data = r_sample(1:adc_clock/ tx_clock:length(r_sample));

    clear r_sample;clear r_info;
%plot time domain
    plot_N = 3 ; plots;

else %without up/down conversion
    r_data = carriers;
    clear carriers;
end

if RECEIVER == 0
    fprintf(fp1, '%6.6f\n',real(r_data));
    fprintf(fp2, '%6.6f\n',imag(r_data));
    fprintf(fp3, '%d\n',ibits);
    fclose(fp1);
    fclose(fp2);
    fclose(fp3);

end

%----- RECEIVER
if RECEIVER == 1
obits = [];

% 4. ----- Detection of 802.11 Preamble -----
-----
sample_r_data = r_data(1:450); %Buffer of input signal to perform
xcorr
[Correlation,lags] = xcorr(preamble,sample_r_data);
[max_v,pos] = max(abs(Correlation));
plot(abs(Correlation));
maximo = max_v;
maxpos = pos;
shiftpos = length(sample_r_data)-pos+1;
%start with complete valid frame
r_data = r_data(shiftpos:length(r_data));
rx_signal = [];

% 5. ----- Demodulation process -----
-----
for m=1:N_PACKETS
    %extract packets and discard preamble
    start = (m-1)*PACKET_SIZE+1;
    r_pack = r_data(start:start+PACKET_SIZE-1);
    r_pack = r_pack(321:length(r_pack));

    for k=1:N_FRAMES % Multiframe

```

```

%----- Remove cyclic extension -----
-----

r_frame = r_pack(80*(k-1)+17:80*(k-1)+17+63);
r_frame(64) =r_frame(64)*2;
rx_signal = [rx_signal r_frame]; % Collect them all no cyclic
extension
%----- FFT -----
-----
%   input:  r_data
%   output: rdata_h
%-----
-----
%if NO_RX_FULLL == 1

rdata_h = fft(r_frame,64);

%----- Pilot and Null extraction -----
-----
data_h = SGain*[rdata_h(39:64),rdata_h(2:27) ]; %re-arrange
according IFFT algorithm
data_h =
[data_h(1:5),data_h(7:19),data_h(21:32),data_h(34:46),data_h(48:5
2)];
%-----
-----

%----- data extraction -----
-----
%   Call QAM_Demapping :  output_bits
%-----
-----
exact_complex = zeros(1,data_carriers);
for k=1:data_carriers
    b2=threshold(imag(data_h(k)) ,bits_per_sym);
    a2=threshold(real(data_h(k)) ,bits_per_sym);
    exact_complex(k)=a2 +1i*b2;
    output_bits = QAM_Demapping([a2,b2]);
    obits = [obits, output_bits];
end;
%   plot_N = 6;plots();

%end
end

end
bits_in_error = sum(xor(ibits,obits))

end

```

```

%-----
% data_generator
%-----
if (random == 1)
% Create incoming bits randomly for each symbol

    input_bits=round(rand(1,bits_per_sym*data_carriers));    % 0
or 1

else if random == 0
    input_bits = [...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0    0    0    0    0    ...
0    0    0    0];
    input_bits=input_bits(1:bits_per_sym*data_carriers);
else
    input_bits = file_bits(indice:bits_per_sym*data_carriers);
    indice = indice+length(input_bits);

end
end

ibits = [ibits,input_bits];

```

```

% -----
% QAM mapping
% Input: input_bits
% Output: r_branch, i_branch
% -----

for i=1:data_carriers
    index=0;
    in_bits = input_bits((i-1)*bits_per_sym+1:(i-
1)*bits_per_sym+bits_per_sym);

    % Read values of in-phase and quadrature-phase (out-of-phase)
    gains a,b from table

    if (bits_per_sym==2)
        indx = in_bits(1);
        indy = in_bits(2);
    else if(bits_per_sym==4)
        indx = in_bits(1)*2+in_bits(2);
        indy = in_bits(3)*2+in_bits(4);
    else if(bits_per_sym==6)
        indx = in_bits(1)*4+in_bits(2)*2+in_bits(3);
        indy = in_bits(4)*4+in_bits(5)*2+in_bits(6);
    end
end
end

r_branch(i) = qam_table(indx+1);
i_branch(i) = qam_table(indy+1);

end

%Demapping
%Input:a2,b2 decoded complex received vectors
%Output:output_bits

function [salida] = QAM_Demapping(k,vector)
global bits_per_sym
salida=[];
%4_QAM -----
if (bits_per_sym==4)
    for m=1:2
        if (vector(m)==-3)
            qam_rx_bits = [0 0];
        else if (vector(m)==-1)
            qam_rx_bits = [0 1];
        end
    end
end
end

```

```

        else if (vector(m)==3)
            qam_rx_bits = [1 0];
        else if (vector(m)==1)
            qam_rx_bits = [1 1];
        end
    end
    end
    end
    salida = [salida,qam_rx_bits];
end

end

%2_QAM -----
if (bits_per_sym==2)
    for m=1:2
        if (vector(m)==-1)
            qam_rx_bits = [0];
        else if (vector(m)== 1)
            qam_rx_bits = [1];
        end
    end
    salida = [salida,qam_rx_bits];
end

end %2_QAM

%6_QAM -----
if (bits_per_sym==6)
    for m=1:2
        if (vector(m)==-7)
            qam_rx_bits = [0 0 0];
        else if (vector(m)==-5)
            qam_rx_bits = [0 0 1];
        else if (vector(m)==-1)
            qam_rx_bits = [0 1 0];
        else if (vector(m)==-3)
            qam_rx_bits = [0 1 1];

        else if (vector(m)==7)
            qam_rx_bits = [1 0 0];
        else if (vector(m)==5)
            qam_rx_bits = [1 0 1];
        else if (vector(m)==1)
            qam_rx_bits = [1 1 0];
        else if (vector(m)==3)
            qam_rx_bits = [1 1 1];
        end
    end
end
end
end
end
end

```

```

        end
        end
        end
        salida = [salida,qam_rx_bits];
    end

end

end

function [tx,y] = IFFT_Block(x)
tx(1) = 0;
tx(2:27) = x(28:53);
tx(28:38) = 0;
tx(39:64) = x(1:26);
y = ifft(tx,64);

switch plot_N
    case 3

        pwelch(r_data,[],[],[],2/T);
        case 1
            figure(11);
            subplot(311);plot(t,real(carriers),'r');
            subplot(312);plot(t,imag(carriers),'m');
            subplot(313);plot(t,abs(carriers),'b');

        case 2

            figure(20);
            if TFFT0 ~= 0
                frac = round(round(TFFT0/(1/tx_clock))/length(carriers0));
                subplot(3,1,1)
                plot(t(1:frac:round(TFFT0/(1/tx_clock))+1),real(carriers0))
            end
            if TFFT1 ~= 0
                subplot(3,1,2)
                frac = round(round(TFFT1/(1/tx_clock))/length(carriers1));
                plot(t(1:frac:round(TFFT1/(1/tx_clock))),real(carriers1))
            end
            if TFFT2 ~= 0
                subplot(3,1,3)
                frac = round(round(TFFT2/(1/tx_clock))/length(carriers2));
                plot(t(1:frac:round(TFFT2/(1/tx_clock))+1),real(carriers2))
            end
        end
    end
end

```

```

case 4
figure(40);
va_fft(s,4096*48,up_clock,'r',1);

case 6
figure(60);
plot(data_h((1:data_carriers)),'.k');
title('Constellation')
hold on;
plot(exact_complex((1:data_carriers)), 'or');
axis square;
axis equal;
grid on;

case 7
% Principal frequency
figure(71)%alpha,
plot(Sxp(floor((Np+1)/2)+1,:), 'r');
xlabel('alpha(Hz)');
ylabel(['Sxp(',Fo,')'])
title('Peaks are pilots')

% 3D graph
figure(72)
surfl(alphao,fo,Sxp);
view(-37.5,60);
xlabel('alpha');
ylabel('f');
zlabel('Sx');
colormap jet
axis([-20e6 20e6 -9e6 9e6 0 1])
title('Spectral Correlation Density')

end

```

```

function[y] = va_fft(x,N,FS,color,norm)
X=fft(x,N);
XX=(abs(X));
XXX=XX(1:length(XX));
y=XXX;
f=[0:1:length(y)-1];

if (norm == 0)
plot(f*FS/N,y,color);
else
plot(f*FS/N,y/max(abs(y)),color);

```



```

end
%axis([0,3,0,1e3]);

grid on;

```

## A.2 FAM algorithm

```

%=====
% FAM Marcos Castro
%=====

clc
clear
close all;
fclose all;
%Init values

global SINGLE
global WIMIC
ind=1;

%-----
SINGLE      = 0;
FREQ_OFFSET = 1;
WIMIC      = 1;
FIX_PREAMBLE = 0;
LONG_PREAMBLE = 0;
LONG_PILOT = 0;
FIX_PILOT = 0;
SNR_TEST = 0;
AWGN      = 1;
%-----

path = 'C:\Working Thesis\Files\';
if LONG_PREAMBLE == 1
    LFlsize = 0; %N = 256. Just to avoid loading the whole file
    file = ['Q6xPreCyclxVar4096Pk100'];
else if WIMIC == 1
    file = ['NWMicSigP125D075']; %pilots over 25% and data at 75%
    LFlsize = 0;AVG_LENGTH = 0;
else if FIX_PREAMBLE == 1
    file = ['Q6PreCyclxFr20Pk200'];
    LFlsize = 0;AVG_LENGTH = 0;
else if LONG_PILOT == 1
    file = ['Q4PilotVar4096Pk50'];
    LFlsize = 0; %N = 256. Just to avoid loading the whole file
else if FIX_PILOT == 1
    file = ['Q6PilotFr20Pk100'];
    LFlsize = 0;AVG_LENGTH = 0;
end
end
end
end

```

```

end
file_r = [path,file,'_r.dat'];file_i = [path,file,'_i.dat'];%file_b =
[path,file,'_b.dat'];
file_th = ['Th',file,'.dat'];
fp1=fopen(file_r,'r');fp2=fopen(file_i,'r');%fp3=fopen(file_b,'r');

if LFflsize == 0      %to reduce the data load from the file
    s_r = fscanf(fp1,'%f\n',[inf]); %real branch
    s_i = fscanf(fp2,'%f\n',[inf]); %imaginary branch
else
    s_r = fscanf(fp1,'%f\n',[LFflsize]); %real branch
    s_i = fscanf(fp2,'%f\n',[LFflsize]); %imaginary branch
end

rx_s = (s_r+ 1i.*s_i);

if FREQ_OFFSET
    Off = 3;
    if WIMIC
        Foffset = -2e3*Off;
        ts = 1/256e3;
    else
        Foffset = -156.25e3*Off;
        ts = 1/20e6;
    end

    t = 0:ts:ts*(length(rx_s)-1);
    rx_s = rx_s.*exp(-1i*2*pi*Foffset*t);
end

rx_s = rx_s(160:end);

fclose all;clear s_r;clear s_i;

if SNR_TEST
    SNR = [12 10 5 4.5 4 3.5 3 2.5 2 1.5 1 0.5 0 -0.5 -1 -1.5 -2 ...
-2.5 -3 -3.5 -4 -4.5 -5 -5.5 -6 -6.5 -7 -7.5 -8 -8.5 -9 -9.5 -10
];
    avg = [430 500 700];
else
    SNR = 1;
    avg = [430 500 700];
end

for indx = 1:length(avg)

    AVERAGE_SIZE = avg(indx);

    Hyp2=[];Hyp1=[];
    for ee=1:2

```

```

ind=1;
for m=1:length(SNR)
    SNR(m)
    if ee==2
        NOISE_ALONE = 1; %to force noise in the same test
    else
        NOISE_ALONE = 0;
    end
    if AWGN
        rx_baseb = awgn(rx_s,SNR(m),'measured') ; % Additive White Gaussian
Channel
    else
        rx_baseb = rx_s;
    end
    if NOISE_ALONE == 1
        rx_baseb = rx_baseb - rx_s;
    end

%----- FFT Accumulation Method FAM -----
----
% FAM Initialization

if WIMIC
    fam_clock = 256e3;
    dalpha=ceil(fam_clock/(256)); % delta frequency
    df=ceil(fam_clock/(128)); % delta frequency
else
    fam_clock = 20e6;
    dalpha=ceil(fam_clock/(256)); % delta frequency
    df=ceil(fam_clock/(128)); % delta frequency
end

% FAM call
TheFAM_Ciclostationary;
plot_N = 7; plots;
plot_N = 8; plots;
plot_N = 9; plots;

if NOISE_ALONE == 1
    Hyp1(ind) = [AVERAGE_PEAKE / AVERAGE_DATA];

else
    Hyp2(ind) = [AVERAGE_PEAKE / AVERAGE_DATA];

end
ind= ind+1;

end

end
vec1(:,indx) = Hyp1;
vec2(:,indx) = Hyp2;

```

```

end

color = ['c','m','r','k','b','g','y','w'];

figure(88)
for k=1:length(avg)
    if k==9
        plot(SNR,vec1(:,k),'.-m');
        hold on;
        plot(SNR,vec2(:,k),'.-m');
    else
        plot(SNR,vec1(:,k),color(k));
        hold on;
        plot(SNR,vec2(:,k),color(k));
    end
end

end
fclose all;

%=====
%====
% Marcos Castro
%=====
%====

%define parameters
global SINGLE
global WIMIC
Np = pow2(nextpow2(fam_clock/df));
L = Np/4;
P = pow2(nextpow2(fam_clock/dalpha/L));
N = P*L;
Sxp = zeros(Np+1,2*N+1);
Length_1 = (P-1)*L+Np;
%initialize
if AVERAGE_SIZE==0
    avg_size = round(length(rx_baseb)/N)-1;
else
    avg_size = AVERAGE_SIZE;
end
%clear and define
X = zeros(Np,P);
Xproduct = zeros(P,Np^2);
oscl = zeros(Np,P);
f_Center = Np/2+1;
alphao = -fam_clock:fam_clock/N:fam_clock;
fo = -fam_clock/2:fam_clock/Np:fam_clock/2;
a = hamming(Np);
%prepare oscillator for downconversion
for k=-Np/2:Np/2-1
    for k0=0:P-1

```

```

        osc1(k+Np/2+1,k0+1)=exp(-1i*2*pi*k*k0*L/Np);
    end
end
for t=1:1:avg_size

    x=rx_baseb(1,(N)*(t-1)+1:(N)*t); % x of 256 samples

    xx = [x,zeros(1,Length_1-N)]';

    for k=0:P-1    % P = 8 sliding windows of length Np = 128. Sliding
L=32
        X(:,k+1)=xx(k*L+1:k*L+Np);
    end

    %windowing

    Xw=diag(a)*X;

    %first FFT
    fft_1 = fft(Xw);
    fft_1 = [fft_1(Np/2+1:Np,:) ; fft_1(1:Np/2,:)];

    X_shifted=fft_1.*osc1;
    X_shifted=conj(X_shifted');

    %multiplication
if SINGLE
    if WIMIC==0
        f1 = 49;f2 = 81;
    else
        f1 = 69;f2 = 113;
    end

    for k=1:Np
        if ((k<(f1-6) || k>(f1+6)) && (k<(f2-6) || k>(f2+6)))
            X_shifted(:,k)=0;
        end
    end
    for k=1:Np
        for c=1:Np
            if X_shifted(:,k)~=0
                Xproduct(:,(k-
1)*Np+c)=(X_shifted(:,k).*conj(X_shifted(:,c)));
            end
        end
    end

else
    for k=1:Np
        for c=1:Np
            Xproduct(:,(k-
1)*Np+c)=(X_shifted(:,k).*conj(X_shifted(:,c)));
        end
    end
end
end

```

```

end

%second FFT
fft_2=abs(fft(Xproduct));

fft_2 = [fft_2(P/2+1:P,:); fft_2(1:P/2,:)];

Sx=zeros(Np+1,2*N+1);
alpha_step = 0;

for k1 = 1:P
    l1=1;
    freq_start = -Np;
    alpha_start = -3 + alpha_step;
    alpha_step = alpha_step+1;
    for k3=0:Np^2-1
        alpha = alpha_start*dalpha/fam_clock -
2*(mod(k3,Np))*(dalpha/(fam_clock/1));
        y_alpha = N + (alpha*N+1);
        frequency = freq_start*dalpha/fam_clock +
1*(mod(k3,Np))*(dalpha/(fam_clock/1));
        x_frequency=round(1+Np*(frequency+0.5));

        if mod(k3+1,Np)==0
            alpha_start = alpha_start + 2;
            freq_start = freq_start + 1;
        end

        if (alpha>-1 && alpha<1)
            Sx(x_frequency,round(y_alpha)) = fft_2(k1,k3+1);

        end

    end

end

Sxp=Sxp+Sx;

end
Sxp=Sxp./max(max(Sxp));

Alpha=num2str(round(1+N*(alpha/fam_clock+1)));
Fo=num2str(floor((Np+1)/2)+1);

alpha0=257;
% ----- Threshold comparison -----
-----
if WIMIC
    if SINGLE
        SUM_PEAKS= Sxp(f1,alpha0)+Sxp(f2,alpha0);
        TOTAL = sum(Sxp(f1-6:f1+6,alpha0))+ sum(Sxp(f2-6:f2+6,alpha0));
        AVERAGE_DATA = (TOTAL - SUM_PEAKS)/(13*2-2);
    end
end

```

```

AVERAGE_PEAKS = SUM_PEAKS/2;

else

    SUM_PEAKS= Sxp(53,alpha0)+Sxp(79,alpha0)+Sxp(81,alpha0)+ ...
               Sxp(111,alpha0)+Sxp(123,alpha0)+Sxp(127,alpha0);
    TOTAL = sum(Sxp(2:127,alpha0));
    AVERAGE_DATA = (TOTAL - SUM_PEAKS)/(126-6);
    AVERAGE_PEAKS = SUM_PEAKS/6;
end

else
if SINGLE
    SUM_PEAKS= Sxp(f1,alpha0)+Sxp(f2,alpha0);
    TOTAL = sum(Sxp(f1-6:f1+6,alpha0))+ sum(Sxp(f2-6:f2+6,alpha0));
    AVERAGE_DATA = (TOTAL - SUM_PEAKS)/(13*2-2);
    AVERAGE_PEAKS = SUM_PEAKS/2;

else
    SUM_PEAKS= Sxp(57,alpha0)+Sxp(49,alpha0)+Sxp(41,alpha0)+ ...
               Sxp(33,alpha0)+Sxp(25,alpha0)+Sxp(17,alpha0)+ ...
               Sxp(73,alpha0)+Sxp(81,alpha0)+Sxp(89,alpha0)+ ...
               Sxp(97,alpha0)+Sxp(105,alpha0)+Sxp(113,alpha0);
    TOTAL = sum(Sxp(11:64,alpha0))+ sum(Sxp(66:119,alpha0));
    AVERAGE_DATA = (TOTAL - SUM_PEAKS)/(54*2-12);
    AVERAGE_PEAKS = SUM_PEAKS/12;
end

end
end

```

### A.3 Frequency offset estimation

```

% Blind Synchronization
% Marcos Castro

```

```

clear all;
close all;

```

```

WiFi = 1;
TIME_OFF = 1;
Time_Offset = 7;

```

```

FREQ_OFF = 1;
Foffset = -1.87e3;
AVG = 20;

nOFDM = 80;
nGI = 16;
nFFT = 64;

old_block = zeros(1,nOFDM);
new_block = [];vec1=[];vec2=[];
PeakIDavg = 0;Davg=0;

if WiFi == 1
    file = ['Q6PreCyclxFr20Pk200'];
else
    file = ['VP125DP75WMicSig'];
end
%----- Files management -----
path = 'C:\Working Thesis\Files\';
file_r = [path,file,'_r.dat'];file_i =
[path,file,'_i.dat'];
file_th = ['Th',file,'.dat'];
fp1=fopen(file_r,'r');fp2=fopen(file_i,'r');

s_r = fscanf(fp1,'%f\n',[inf]); %real branch
s_i = fscanf(fp2,'%f\n',[inf]); %imaginary branch
rx_s1 = (s_r+ 1i.*s_i);
%----- Offset in Time domain-----

if TIME_OFF
    rx_s1 = [rand(1,Time_Offset),rx_s1 ];
end

%----- Offset in Frequency Domain -----

if FREQ_OFF
    if WiFi
        ts = 1/20e6;
    else
        ts = 1/256e3;
    end
    t = 0:ts:ts*(length(rx_s1)-1);
    rx_s1 = rx_s1.*exp(-1i*2*pi*Foffset*t);
end

%-----

```



```

for cont=0:floor(length(rx_s1)/nOFDM -1 )
    %OFDM incoming symbol arrangement
    new_block= rx_s1(cont*nOFDM+1:cont*nOFDM+nOFDM);
    SymbolBuffer = [old_block,new_block]';%new in front,old
to the back
    old_block = new_block;

    CorrelationOutput = zeros(nOFDM+nGI,1);
    RealDomainPeaks = zeros(nOFDM,1);
    ComplexDomainPeaks = zeros(nOFDM,1);
    %Correlate samples nFFT apart
for indx1 = 1:2*nOFDM-nFFT
    CorrelationOutput(indx1)=SymbolBuffer(indx1) .* conj(
SymbolBuffer(indx1+nFFT) ) ;
end
    %average CorrelationOutput with algorithm
for indx1 = 1:nOFDM
    ComplexIntgr1 = 0;PowerIntgr1 = 0;
    for indx2=1:nGI
        ComplexIntgr1 = ComplexIntgr1 +
CorrelationOutput(indx1+indx2);
        PowerIntgr1 = PowerIntgr1 + abs(
SymbolBuffer(indx1+indx2) ).^2;
    end
    RealDomainPeaks(indx1) = 2*abs(ComplexIntgr1)-
PowerIntgr1;
    ComplexDomainPeaks(indx1) = ComplexIntgr1;
end

Davg = Davg+(RealDomainPeaks-Davg)/AVG;

[maxvalue,position]=max(Davg);
% figure(5);hold on;plot(Davg,'r');hold off;

time_offset = position;
freq_offset = angle(ComplexDomainPeaks(position));

end

```