

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

GIANCARLO CLOSS ZANUZ

**FERRAMENTA DE ANÁLISE DE OCUPAÇÃO DO ESPECTRO ATRAVÉS DAS
TÉCNICAS DE DETECÇÃO DE ENERGIA E CICLO-ESTACIONÁRIO, UTILIZANDO
UMA PLATAFORMA DE RÁDIO DEFINIDO POR SOFTWARE**

PORTO ALEGRE

JUNHO DE 2015

GIANCARLO CLOSS ZANUZ

**FERRAMENTA DE ANÁLISE DE OCUPAÇÃO DO ESPECTRO ATRAVÉS DAS
TÉCNICAS DE DETECÇÃO DE ENERGIA E CICLO-ESTACIONÁRIO, UTILIZANDO
UMA PLATAFORMA DE RÁDIO DEFINIDO POR SOFTWARE**

TRABALHO DE CONCLUSÃO DE CURSO
APRESENTADO AO DEPARTAMENTO DE
ENGENHARIA ELÉTRICA DA UNIVERSIDADE
FEDERAL DO RIO GRANDE DO SUL, COMO
PARTE DOS REQUISITOS PARA GRADUAÇÃO
EM ENGENHARIA ELÉTRICA.

ORIENTADOR: PROF. DR. CARLOS EDUARDO PEREIRA

PORTO ALEGRE

JUNHO DE 2015

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

REITOR: CARLOS ALEXANDRE NETTO

VICE-REITOR: RUI VICENTE OPPERMANN

ESCOLA DE ENGENHARIA

DIRETOR: LUIZ CARLOS PINTO DA SILVA FILHO

VICE-DIRETORA: CARLA SCHWENGBER TEM CATEN

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

CHEFE DO DEPARTAMENTO: ROBERTO PETRY HOMRICH

CIP - Catalogação na Publicação

Closs Zanuz, Giancarlo

Ferramenta de Análise de Ocupação do Espectro
Através das Técnicas De Detecção de Energia e Ciclo-
estacionário, Utilizando Uma Plataforma de Rádio
Definido por Software / Giancarlo Closs Zanuz. --
2015.

77 f.

Orientador: Carlos Eduardo Pereira.

Coorientador: Jean Michel Winter.

Trabalho de conclusão de curso (Graduação) --
Universidade Federal do Rio Grande do Sul, Escola de
Engenharia, Curso de Engenharia Elétrica, Porto
Alegre, BR-RS, 2015.

1. Rádio Definido por Software. 2. Ciclo-
estacionário. 3. Detecção de Energia. 4. Rádio
Cognitivo. I. Pereira, Carlos Eduardo, orient. II.
Winter, Jean Michel, coorient. III. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UFRGS com os
dados fornecidos pelo(a) autor(a).

GIANCARLO CLOSS ZANUZ

**FERRAMENTA DE ANÁLISE DE OCUPAÇÃO DO ESPECTRO ATRAVÉS DAS
TÉCNICAS DE DETECÇÃO DE ENERGIA E CICLO-ESTACIONÁRIO, UTILIZANDO
UMA PLATAFORMA DE RÁDIO DEFINIDO POR SOFTWARE**

ESTE TRABALHO FOI ANALISADO E JULGADO ADEQUADO PARA A APROVAÇÃO DA DISCIPLINA DE “PROJETO DE DIPLOMAÇÃO”, DO DEPARTAMENTO DE ENGENHARIA ELÉTRICA E APROVADO EM SUA FORMA FINAL PELO ORIENTADOR E PELA BANCA EXAMINADORA.

BANCA EXAMINADORA:

PROF. DR. CARLOS EDUARDO PEREIRA, UFRGS
DOUTOR PELA UNIVERSIDADE DE STUTTGART – STUTTGART, ALEMANHA

PROF. DR. VALNER BRUSAMARELLO, UFRGS
DOUTOR PELA UNIVERSIDADE FEDERAL DE SANTA CATARINA – FLORIANÓPOLIS, BRASIL

PROF. DR. IVAN MULLER, UERGS
DOUTOR PELA UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL – PORTO ALEGRE, BRASIL

APROVADO EM: ___/___/___

Agradecimentos

Primeiramente gostaria de agradecer aos meus pais e a minha família pela motivação e suporte que me ofereceram para tornar este trabalho concreto. Agradeço também aos meus colegas e amigos da Engenharia Elétrica da UFRGS pela união e bons momentos que foram essenciais durante toda a graduação.

Por fim agradeço ao Prof. Carlos Eduardo Pereira pela sua orientação, e ao meu co-orientador Jean Michel Winter que me ajudou no processo de revisão e supervisão.

“O raciocínio matemático pode ser considerado, esquematicamente, como o exercício da combinação de duas faculdades, que podemos chamar de intuição e engenhosidade”

Alan Turing

Resumo

Neste trabalho foi detalhada a construção de uma aplicação desenvolvida no ambiente computacional do LabVIEW e do MATLAB, que tem como objetivo realizar a análise do espectro de radiofrequências na faixa ISM de 2,4GHz até 2,485GHz. Para este sensoriamento foram utilizadas as técnicas de detecção de energia e o método ciclo-estacionário, utilizando uma plataforma de Rádio Definido por Software. As rotinas dos métodos implementados são explicitadas através de fluxogramas e diagramas de blocos, referenciando cada passo com as equações e fundamentação teórica que sustentam as duas técnicas estudadas. Ambas as técnicas foram validadas por simulação computacional utilizando um sinal pseudo-aleatório para o primeiro algoritmo, e um sinal modulado do tipo BPSK para o segundo algoritmo. A aplicação obteve êxito na classificação correta das duas classes de sinais analisados, IEEE 802.11g e IEEE 802.15.4, obtendo uma taxa de acertos superior a 90% para ambos os sinais com uma relação sinal-ruído tão pequena quanto 1,6dB. Também é apresentada a importância das técnicas de sensoriamento de espectro com o conceito de Rádio Cognitivo, desenvolvendo o seu papel no surgimento de novas tecnologias na área de redes de sensores sem fio que possibilitem obter uma comunicação mais robusta e confiável perante o crescimento acentuado do número de dispositivos ocupando as superpopuladas faixas de frequências ISM.

Palavras-Chaves: Sensoriamento do Espectro, Detecção de Energia, Ciclo-Estacionário, Rádio Cognitivo, IEEE 802.11g, IEEE 802.15.4.

Abstract

This work detailed the construction of an application developed in the computing environment of LabVIEW and MATLAB, which aims to conduct the analysis of the radio frequency spectrum in the 2.4GHz to 2,485GHz ISM band. For this sensing, it was used the energy detection technique and cyclo-stationary method, using a Universal Radio Software Peripheral (USRP). The routines of the implemented methods are explained through flowcharts and block diagrams, referencing each step with equations and the theoretical foundations that support both techniques. Both techniques have been validated by computer simulation using a pseudo-random signal to the first algorithm, and a modulated signal of BPSK type for the second algorithm. The application was successful in the correct classification of the two signal classes analyzed, IEEE 802.11g and IEEE 802.15.4, getting a correct rate higher than 90% for both signals with as little as 1,6dB signal to noise ratio. The importance of spectrum sensing techniques is also presented with the concept of Cognitive Radio, by developing its role in the emergence of new technologies in wireless sensor network area to enable a more robust and reliable communication before the enhanced growth of the number of devices occupying the overcrowded channels of the ISM bands.

Keywords: Spectrum Sensing, Energy Detection, Cyclostationary, Cognitive Radio, IEEE 802.11g, IEEE 802.15.4.

Lista de Ilustrações

FIGURA 1: EXEMPLO DE SEGMENTAÇÃO EM BANDAS DA FAIXA ISM DE 2,4GHZ.....	15
FIGURA 2: REPRESENTAÇÃO DAS PORTADORAS NO SISTEMA OFDM.	17
FIGURA 3: ORGANIZAÇÃO DAS PORTADORAS NO PADRÃO 802.11G.	18
FIGURA 4: EXEMPLO DE CONEXÕES DE UMA REDE DA CLASSE IEEE 802.15.4.	19
FIGURA 5: REPRESENTAÇÃO DE OPORTUNIDADES NO ESPECTRO DE FREQUÊNCIA AO LONGO DO TEMPO.....	22
FIGURA 6: COMPARAÇÃO DAS TÉCNICAS DE SENSORIAMENTO DE ESPECTRO.....	23
FIGURA 7: COMPARAÇÃO DAS TÉCNICAS DE SENSORIAMENTO DE ESPECTRO.....	25
FIGURA 8: RÁDIO DEFINIDO POR SOFTWARE NI-USRP 2932.	31
FIGURA 9: DIAGRAMA FUNCIONAL DO HARDWARE NI-USRP 2932.	32
FIGURA 10: PLACA DE DESENVOLVIMENTO PARA OS TESTES COM O PADRÃO IEEE 802.15.4.	34
FIGURA 11: ANALISADOR DE REDES FIELDFOX N9912A	35
FIGURA 12: ENSAIO DE CALIBRAÇÃO REALIZADO PARA CALIBRAR O DISPOSITIVO <i>USRP</i>	36
FIGURA 13: DIAGRAMA DE BLOCOS GERAL.....	37
FIGURA 14: DIAGRAMA DE BLOCOS: CONFIGURAÇÃO INICIAL DO DISPOSITIVO <i>USRP</i>	39
FIGURA 15: DIAGRAMA DE BLOCOS: FREQUÊNCIA CENTRAL DO CANAL A SER ANALISADO.	40
FIGURA 16: DIAGRAMA DE BLOCOS: TAMANHO DAS AMOSTRAS A RECEBER E OBTENÇÃO DAS AMOSTRAS.....	40
FIGURA 17: DETECÇÃO DE ENERGIA: DIAGRAMA DE BLOCOS 1.....	42
FIGURA 18: DETECÇÃO DE ENERGIA: DIAGRAMA DE BLOCOS 2.....	43
FIGURA 19: EXEMPLO DE SINAL ANALISADO PELO MÉTODO DE DETECÇÃO DE ENERGIA.	44
FIGURA 20: DIAGRAMA DE BLOCOS: <i>SCRIPT</i> DE DETECÇÃO DE ENERGIA.....	45
FIGURA 21: MÉTODO CICLO-ESTACIONÁRIO: DIAGRAMA DE BLOCOS 1.	46
FIGURA 22: FORMAÇÃO DA MATRIZ <i>MAINWINDOWED</i>	47
FIGURA 23: MÉTODO CICLO-ESTACIONÁRIO: DIAGRAMA DE BLOCOS 2.	49
FIGURA 24: ESTIMAÇÃO DA FUNÇÃO DE AUTOCORRELAÇÃO CÍCLICA DE UM SINAL EXEMPLO.	50
FIGURA 25: DIAGRAMA DE BLOCOS: <i>SCRIPT</i> DO MÉTODO CICLO-ESTACIONÁRIO.....	50
FIGURA 26: FUNÇÃO DE AUTOCORRELAÇÃO CÍCLICA PARA SINAL COM PADRÃO IEEE 802.11G.....	51
FIGURA 27: FUNÇÃO DE AUTOCORRELAÇÃO CÍCLICA SUAVIZADA PELA MÉDIA DAS JANELAS DE ANÁLISE, PARA SINAL COM PADRÃO IEEE 802.11G.	52

FIGURA 28: FUNÇÃO DE AUTOCORRELAÇÃO CÍCLICA DO PREFIXO DO PACOTE DE UM SINAL DO PADRÃO IEEE 802.11G.	52
FIGURA 29: FUNÇÃO DE AUTOCORRELAÇÃO CÍCLICA PARA SINAL COM PADRÃO IEEE 802.15.4.	53
FIGURA 30: FUNÇÃO DE AUTOCORRELAÇÃO CÍCLICA SUAVIZADA PELA MÉDIA DAS JANELAS DE ANÁLISE, PARA SINAL COM PADRÃO IEEE 802.15.4.	54
FIGURA 31: DIAGRAMA DE BLOCOS: BLOCOS DE DECISÃO E FINAL DO LAÇO DE REPETIÇÃO.	55
FIGURA 32: PAINEL FRONTAL DO PROGRAMA DESENVOLVIDO EM LABVIEW.	56
FIGURA 33: PRIMEIRO TESTE, VERIFICAÇÃO DE ERROS DO TIPO I.	58
FIGURA 34: PRIMEIRO TESTE, VERIFICAÇÃO DE ERROS DO TIPO II.	59
FIGURA 35: SEGUNDO TESTE, VERIFICAÇÃO DE ERROS DO TIPO I.	60
FIGURA 36: SEGUNDO TESTE, VERIFICAÇÃO DE ERROS DO TIPO II.	61
FIGURA 37: TRECHO DO SINAL GERADO BPSK.	62
FIGURA 38: ESPECTRO DE FREQUÊNCIAS DO SINAL MODULADO BPSK.	63
FIGURA 39: FUNÇÃO DE AUTOCORRELAÇÃO CÍCLICA PARA SINAL MODULADO BPSK.	64
FIGURA 40: MÉDIA DA FUNÇÃO DE AUTOCORRELAÇÃO CÍCLICA PARA JANELA DE 0,5MS.	65
FIGURA 41: MÉDIA DA FUNÇÃO DE AUTOCORRELAÇÃO CÍCLICA PARA JANELA DE 1,0MS.	66
FIGURA 42: MÉDIA DA FUNÇÃO DE AUTOCORRELAÇÃO CÍCLICA PARA JANELA DE 2,5MS.	66
FIGURA 43: DIAGRAMA DE BLOCOS PARA TESTE DO SINAL IEEE802.15.4.	68
FIGURA 44: EQUIPAMENTOS UTILIZADOS PARA O TESTE DO SINAL IEEE802.15.4.	68
FIGURA 45: TAXA DE ACERTOS EM FUNÇÃO DA RSR, ENSAIO COM SINAL IEEE 802.15.4.	69
FIGURA 46: DIAGRAMA DE BLOCOS PARA TESTE DO SINAL IEEE802.11G.	70
FIGURA 47: EQUIPAMENTOS UTILIZADOS PARA O TESTE DO SINAL IEEE 802.11G.	70
FIGURA 48: TAXA DE ACERTOS EM FUNÇÃO DA RSR, ENSAIO COM SINAL IEEE 802.11G.	71

Sumário

Agradecimentos.....	5
Resumo	7
Abstract	8
Lista de Ilustrações	9
1 Introdução	13
2 Fundamentação Teórica: Considerações sobre redes de comunicação sem fio	14
2.1 Bandas ISM.....	14
2.2 Protocolos de comunicação sem fio nas faixas ISM	15
2.2.1 IEEE 802.11	16
2.2.2 IEEE 802.15.4	18
2.3 O Rádio Cognitivo	20
2.4 Técnicas de sensoriamento de espectro	22
2.4.1 Método de detecção de energia	24
2.4.2 Método Ciclo estacionário.....	27
3 Proposta da ferramenta de sensoriamento de espectro	30
3.1 Ferramentas e Materiais.....	30
3.2 Diagrama de Blocos	36
3.2.1 Configuração inicial NI-USRP	37
3.2.2 Algoritmo de Detecção de Energia	40
3.2.3 Algoritmo Ciclo-Estacionário	45
3.2.4 Classificação do Sinal	51
3.2.5 Interface Homem-Máquina	55
4 Validação e Testes	57
4.1 Validação do método de detecção de energia	57
4.2 Validação do método ciclo-estacionário	61

4.3	Testes da aplicação desenvolvida.....	67
4.3.1	Testes com sinal IEEE 802.15.4.....	67
4.3.2	Testes com sinal IEEE 802.11g.....	69
5	Conclusão e Trabalhos Futuros	72
	Referências Bibliográficas.....	73
	ANEXO A: Algoritmo de Detecção de Energia em linguagem de MATLAB	75
	ANEXO B: Algoritmo ciclo-estacionário em linguagem de MATLAB	77

1 Introdução

Atualmente há uma tendência de crescimento no número de dispositivos sem fio para automação de processos residenciais e industriais, como parte do processo de modernização da tecnologia de produção de semicondutores que proporcionou uma queda no preço dos circuitos integrados, e a difusão do conceito de *'internet das coisas'* na qual proveu a necessidade de interligar os objetos ao nosso redor. Neste contexto é fundamental compreender que os recursos utilizados pelos dispositivos sem fio não são ilimitados; o espectro de radiofrequência é um recurso regulamentado em cada país, que disponibiliza poucas e estreitas faixas de uso livre para fins industriais, científicos e médicos. São nestas faixas em que a grande maioria de dispositivos sem fio opera, conseqüentemente ocasionando o congestionamento de informações e degradando a qualidade do sinal destes dispositivos.

Considerando estes fatos há uma grande demanda na área de pesquisa por possíveis soluções e tecnologias capazes de enfrentar estes desafios. Uma linha de pesquisa que apresenta grandes expectativas é o conceito de Rádio Cognitivo, que é debatido de forma sucinta no [SUBCAPÍTULO 2.3](#). Este trabalho trata de duas técnicas utilizadas para realizar o sensoriamento do espectro, sendo este um dos pilares no qual o conceito de Rádio Cognitivo está baseado.

Neste trabalho são apresentadas duas técnicas utilizadas para o sensoriamento do espectro: o método de detecção de energia e o método ciclo-estacionário. Ambas as técnicas foram desenvolvidas em ambiente LabVIEW, juntamente com rotinas escritas em linguagem de MATLAB, formando uma aplicação que tem como objetivo detectar e classificar dois sinais distintos das classes IEEE 802.11g e IEEE 802.15.4, operando na faixa de frequência ISM (*Industrial, Scientific and Medical*) centrada em 2,45GHz. O dispositivo utilizado para a aquisição das amostras nesta faixa de frequência é um Rádio Definido por Software da National Instruments, utilizado para realizar o sensoriamento do espectro pela sua ampla gama de configurações e seu hardware robusto.

2 Fundamentação Teórica: Considerações sobre redes de comunicação sem fio

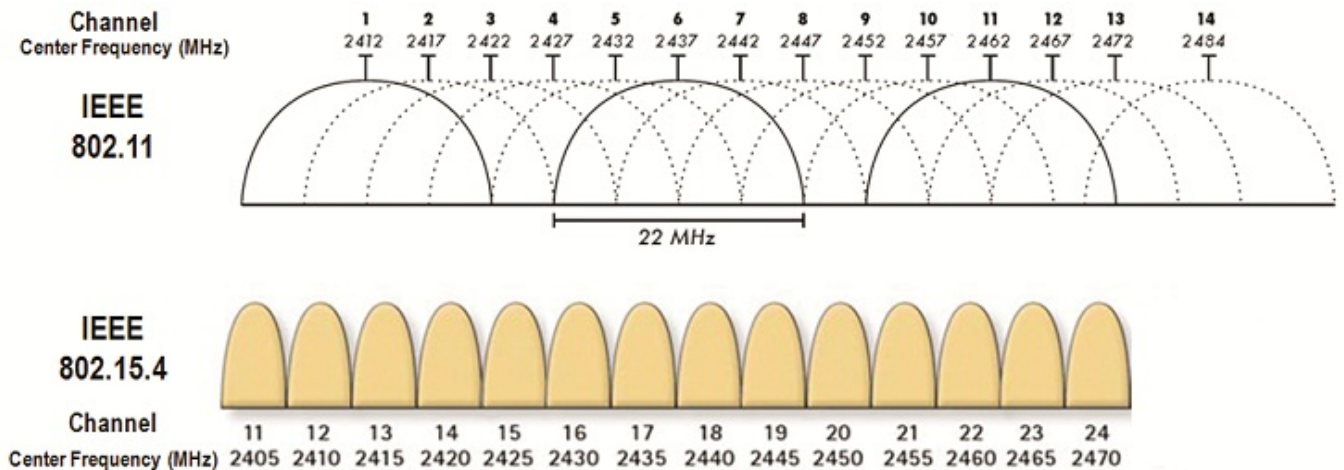
Nesta seção serão apresentados e revisitados alguns conceitos que são necessários para o completo entendimento dos processos aplicados neste trabalho. Muitos dos tópicos apresentam apenas uma revisão superficial, sendo que a documentação completa sobre cada assunto pode ser encontrada junto às referências devidas de cada seção.

2.1 Bandas ISM

As bandas de frequência ISM (*Industrial, Scientific and Medical*), são faixas no espectro de frequência reservadas para fins industriais, científicos e médicas, que possuem padronização internacional para seu uso, divididas em três regiões. Em geral há 12 faixas ISM licenciadas para uso mundialmente, sendo a primeira centrada em 6,780 MHz e a última em 245,0 GHz, com bandas entre 30 kHz e 2 GHz. No Brasil, a ANATEL (Agência Nacional de Telecomunicações) segue os padrões internacionais, permitindo as 11 faixas ISM nesta região ([ANATEL, 2015](#)). Entre as faixas ISM, a mais utilizada atualmente em dispositivos sem fio está situada em 2,4 GHz e 2,5 GHz, com uma banda total de 100 MHz.

Por serem faixas não licenciadas qualquer dispositivo homologado pode operar nestas frequências, o que implica que estes devem tolerar interferência de outros dispositivos. De forma a limitar a banda de frequência utilizada por um dispositivo em determinada faixa, esta é dividida em diversas bandas, organizando assim o espectro. Como exemplo segue a [FIGURA 1](#), no qual a faixa de 2,4 GHz é subdividida em diferentes canais, para dois dispositivos operando em protocolos diferentes.

Figura 1: Exemplo de segmentação em bandas da faixa ISM de 2,4GHz.



Fonte: [\(National Instruments, 2012\)](#).

2.2 Protocolos de comunicação sem fio nas faixas ISM

Atualmente, devido ao aumento no número de dispositivos sem fio operando em setores industriais, comerciais e residenciais, há uma tendência de crescimento na ocupação do espectro de radiofrequência. Em especial, esse crescimento ocorre com maior volume nas faixas ISM, devido a sua característica de uso livre, descomprometendo os dispositivos de taxas por ocupação em uma faixa licenciada. Neste contexto, ocorre que a utilização incorreta destas faixas prejudica todos os dispositivos que nela se situam, degradando a qualidade da comunicação. Como forma de padronizar a comunicação entre diversos dispositivos operando nestas faixas, algumas normas foram criadas, com o objetivo de regulamentar a utilização correta e permitir a coexistência destes dispositivos sem fio. O conjunto de normas mais adotado internacionalmente é dado pelo IEEE (*Institute of Electrical and Electronics Engineers*), sob o nome de IEEE 802.

As normas IEEE 802 tratam especificamente de padrões relacionados a redes locais e publicas. As normas incluem as duas camadas mais inferiores do modelo OSI, sendo elas a camada física (PHY) e a de enlace, esta ainda sendo subdividida em controle de enlace lógico (LLC) e controle de acesso ao meio (MAC). A norma IEEE 802 possui diversas subdivisões dependendo da aplicação e serviço prestado, englobando diversos tipos de redes como sem

fio, fibra óptica e Ethernet. Entre as diversas subdivisões, as que serão abordadas devido à sua relevância são:

- IEEE 802.11: Especificações para a Rede Local sem Fio (WLAN), que forma a base para o padrão WiFi.
- IEEE 802.15: Especificações para a Rede Pessoal sem Fio, no qual desta incluem:
 - IEEE 802.15.1: Serviu como padronização para o Bluetooth em seu desenvolvimento inicial;
 - IEEE 802.15.4: Padrão para dispositivos com baixa taxa de transmissão e baixo consumo de energia, assim como baixa complexidade. Os protocolos ZigBee e WirelessHART, são exemplos de padrões baseados nesta norma.

2.2.1 IEEE 802.11

Lançado em 1997, fornece especificações para a camada física e a subcamada MAC, voltada à implementação da rede local sem fio. Ao longo dos anos foi evoluindo e se aperfeiçoando resultando nas subfamílias 802.11b, 802.11g, 802.11n, 802.11ac, etc. Entre os aperfeiçoamentos estão a diversidade de antenas (sistema MIMO)¹, tipo de modulação, faixas de frequência, largura de banda e outros. A lista completa das subfamílias da norma IEEE 802.11 pode ser encontrada em ([IEEE 802.11, Wikipédia, 2015](#)). Este trabalho utiliza como base a subfamília 802.11g, no qual os testes foram realizados.

O padrão 802.11g é a extensão da subfamília 802.11b, lançado em 2003. Possui como diferencial o aumento da taxa de transmissão para até 54Mbit/s utilizando a faixa de frequência ISM de 2,4GHz. Possui como modulação a técnica OFDM (*orthogonal frequency-division multiplex*), ocupando uma banda total de 20MHz para cada canal.

Esta técnica de modulação realiza a multiplexação dos dados dividindo-os em frequências ortogonais entre si. Ou seja, a informação está alocada em diversas portadoras próximas de banda estreita que formam bases ortogonais, onde cada portadora utiliza uma modulação do tipo QAM ou PSK a uma taxa menor. Também utiliza um sistema de intervalo de guarda, permitindo a diversidade temporal e reduzindo erros devido a interferência inter-

¹ Primeiramente apresentado na subfamília 802.11n.

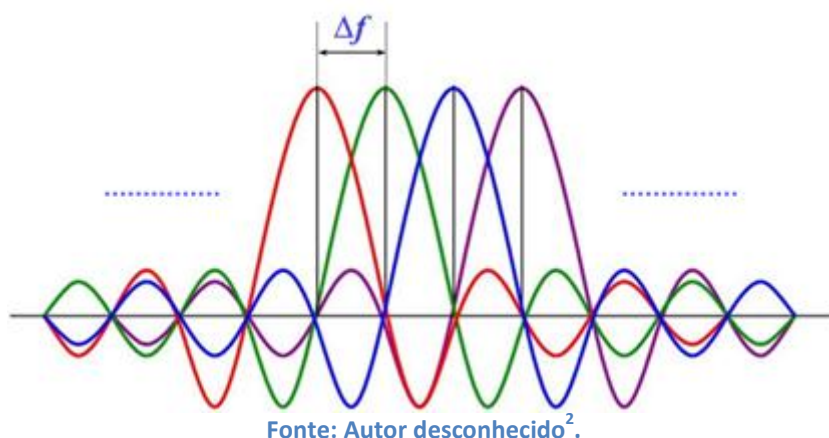
simbólica (ISI). É amplamente utilizada em sistemas que requerem uma alta taxa de dados, boa eficiência espectral e imunidade à interferência e degradação do canal.

A principal característica do sistema OFDM é a utilização de portadoras ortogonais, no qual cada portadora não sofre interferência das demais, aproveitando o máximo do espectro disponível. Assim, diferentemente do sistema FDM (*frequency-division multiplex*), não é necessário um sistema complexo de filtros sintonizados para o processo de transmissão e recepção, facilitando na equalização do canal. A condição de ortogonalidade é verificada quando o espaçamento entre as portadoras satisfaz a EQUAÇÃO 1:

$$\Delta f = \frac{k}{T_{\text{sym}}} \quad 1$$

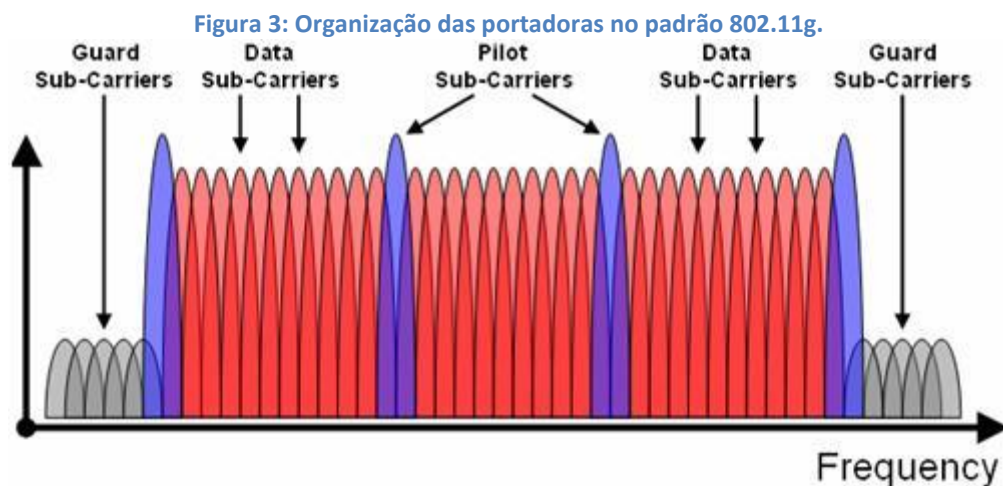
onde k representa uma constante inteira, e T_{sym} é o tempo de duração do símbolo útil, sem o intervalo de guarda. No início de cada pacote é gerado um prefixo de sincronização, que permite ao receptor identificar a portadora correspondente e auxilia na separação dos pacotes. No domínio da frequência, cada portadora aparece como uma função $\text{sinc}(f)$, no qual em cada cruzamento com o zero, ocorre o pico da portadora seguinte. A FIGURA 2 demonstra a alocação de cada portadora.

Figura 2: Representação das portadoras no sistema OFDM.



² Disponível em: <http://www.ecnmag.com/sites/ecnmag.com/files/legacyimages/Figure%201%20-%20OFDM%20Sub-Carriers.png>.

No padrão 802.11g, o distanciamento entre portadoras Δf é de $312,5\text{kHz}$, num total de 52 portadoras igualmente espaçadas, sendo 48 disponíveis para transmissão de dados e 4 utilizadas como piloto. Na [FIGURA 3](#) é mostrado como as portadoras estão alocadas no canal de 20MHz.



Fonte: ([National Instruments, 2009](#))

Pode-se obter mais informações a respeito do esquema de modulação OFDM e do padrão 802.11g em ([Vassis, et al., 2005](#)).

2.2.2 IEEE 802.15.4

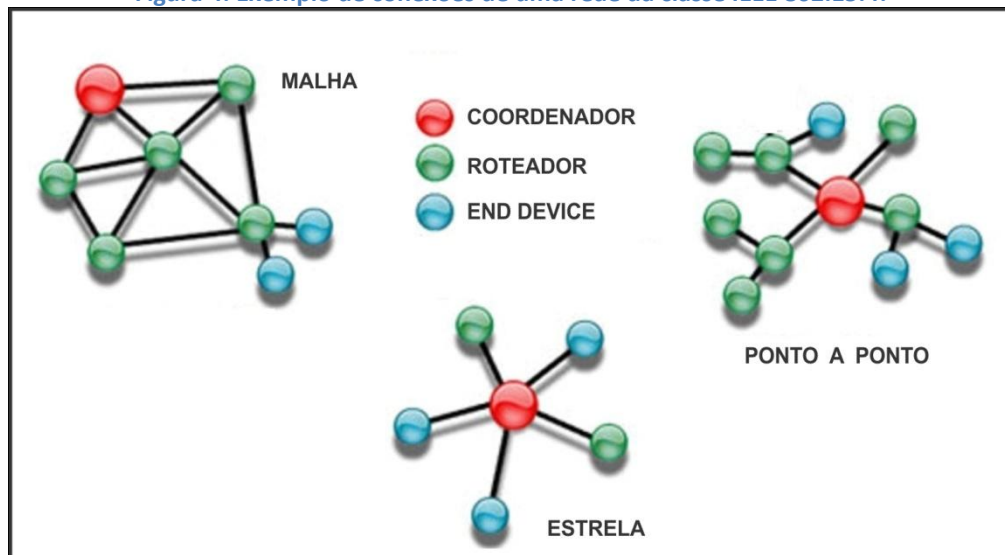
Estabelecido em 2003, incorpora características diferentes dos últimos padrões IEEE 802. Ao contrário da tendência que surgia com dispositivos com taxa de transmissão maior e mais funcionalidades, o padrão IEEE 802.15.4 tem como foco a interação entre diversos dispositivos que possuem como característica a baixa taxa de transmissão, baixo consumo de potência e baixo custo. É pensada para aplicações industriais e domésticas que requerem o uso de diversos sensores espalhados em um determinado ambiente. Os nodos podem ser alimentados por bateria com expectativa de duração de alguns anos, ou ainda, em alguns casos aproveitar o potencial energético do meio em que está situado, como é o caso do processo de 'colheita de energia' (*energy harvesting*, do inglês). ([Adams, 2005](#))

O padrão 802.15.4 especifica ainda o modo de operação da rede sem fio. Cada dispositivo pode ser do tipo totalmente funcional (*Fully-Functional Device*, do inglês), no qual

é capaz de coordenar a rede assim como se comunicar com qualquer dispositivo nela presente, ou o dispositivo pode ser do tipo parcialmente funcional (*Reduced-Functional Device*, do inglês), que são dispositivos mais simples e podem apenas se comunicar com os coordenadores da rede. As redes podem ter topologias simples ou complexas, no qual os coordenadores de rede podem formar novas rotas de comunicação, aumentando a redundância do sistema e assim garantindo robustez à malha. Segue como exemplo a [FIGURA 4](#), no qual apresenta três redes conectadas de modo distinto. São elas:

- Malha: Todos os dispositivos com funcionalidade completa estão conectados entre si através de uma malha de conexões³;
- Árvore: Há uma ramificação nas conexões, ou seja, um coordenador da rede local possui diversos ramos, e estes estão conectados a outros dispositivos subordinados;
- Estrela: Há apenas um coordenador da rede, e este controla todos os dispositivos nele ligado.

Figura 4: Exemplo de conexões de uma rede da classe IEEE 802.15.4.



Fonte: Autor Desconhecido⁴.

Na camada física o padrão 802.15.4 especifica a utilização de três bandas ISM distintas, dependendo da região e disponibilidade. São elas a de 2.4GHz com uma taxa de

³ Não presente no padrão IEEE 802.15.4 original, porém está disponível no protocolo ZigBee.

⁴ Disponível em: http://www.taskblog.com.br/wp-content/uploads/2013/04/ZigBee_interna.jpg.

250kbps, a de 915MHz operando em 40kbps e por fim a faixa em 868MHz com taxa de 20kbps. Ao todo são disponibilizados 27 canais nas 3 faixas, sendo 16 destes na faixa de 2.4GHz, com uma banda total de 2MHz e um intervalo de 5MHz entre canais.

No processo de modulação é utilizada a técnica de espalhamento espectral DSSS (*Direct-Sequence Spread Spectrum*), na qual 4 bits de informações são processados por um código pseudo-aleatório de 32 pulsos (*chips*, do inglês). Esta técnica permite espalhar a potência no canal, permitindo que diversos dispositivos ocupem a mesma faixa, assim coexistindo em um mesmo canal. No receptor, realizando o processo de correlação-cruzada com o mesmo código pseudo-aleatório, é possível sincronizar o relógio e descobrir qual o dispositivo que originou o sinal.

Após passar pelo pseudocódigo, o sinal é então modulado utilizando a técnica O-QPSK (*Offset Quadrature Phase Shift Keying*), que possui boa imunidade ao ruído. O *offset* em O-QPSK refere-se ao fato que há um atraso entre o bit em fase e o em quadratura, permitindo eliminar o cruzamento com o zero, e implicando em um transmissor mais simples e eficiente. ([Adams, 2005](#))

2.3 O Rádio Cognitivo

Conforme visto na [SEÇÃO 2.2](#), diversos dispositivos compartilham a mesma faixa de frequências, operando simultaneamente e sofrendo com interferências cruzadas entre si. Em determinadas ocasiões, a intensidade da interferência pode ser intensa o suficiente a ponto de ocorrer perda de pacotes, restringindo o uso de comunicação sem fio para aplicações com requisitos intolerantes a perdas. Mesmo com técnicas modernas de modulação e espalhamento espectral como DSSS (*direct-sequence spread spectrum*) ou OFDM, esse tipo de situação ocorre, conforme pode ser visto em ([Machado, et al., 2014](#)) e ([Winter, et al., 2014](#)).

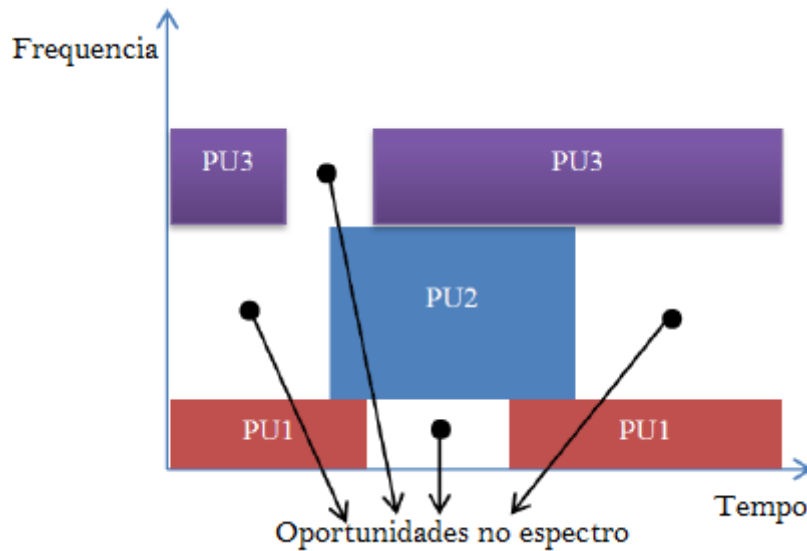
Neste contexto, um conceito em popularização é o chamado Rádio Cognitivo. Por cognitivo subentende-se a aquisição de conhecimento através da percepção e observação exterior. Pode-se definir então o termo rádio cognitivo como um sistema de telecomunicação que está ciente de seus arredores, e é capaz de aprender e se adaptar ao meio em que está inserido com base em uma resposta em tempo real de seus parâmetros e

configurações, permitindo modificar e interagir com o espectro de radiofrequência em que ocupa, com os objetivos de obter uma comunicação mais robusta e confiável com o máximo de eficiência espectral. ([Abdulsattar, et al., 2012 p. 4](#))

Em termos práticos, o rádio cognitivo baseia-se na idéia de gerenciamento do espectro livre, reorganizando e aproveitando espaços úteis em meio ao espectro utilizado por diferentes fontes de transmissão de sinais em rádio frequência. Um rádio cognitivo é capaz de observar uma determinada faixa ou banda em um intervalo de tempo, determinar a partir de técnicas de sensoriamento de espectro se determinado espaço espectral está sendo ocupado, e ainda em alguns casos descobrir qual o dispositivo que ocupa este canal. Com base nestes conhecimentos é capaz de se configurar em tempo real e modificar suas propriedades na camada física, como frequência de modulação, banda ocupada e taxa de transmissão. Quando houver diversos dispositivos que incorporem o conceito de rádio cognitivo em um determinado ambiente, estes poderão interagir entre si, compartilhando conhecimento sobre o espectro no local em que estão inseridos e seus arredores, consequentemente ampliando a área de abrangência do sistema inteligente.

No espectro de radiofrequência, especificamente nas faixas ISM, é comum haver diversos dispositivos compartilhando deste recurso, ocupando canais distintos ou não, coexistindo em um mesmo local. Porém como os dispositivos convencionais não possuem conhecimento dos seus arredores, a ocupação do espectro possui uma baixa eficiência por não ser utilizado de forma correta. Por vezes pode ocorrer de um dispositivo não estar ocupando um canal no espaço espectral, em um exato momento de tempo, em um determinado espaço geográfico. Cada uma dessas dimensões, o tempo, a frequência e o espaço, são independentes entre si. Em determinados instantes, podem ocorrer de alguns espaços neste conjunto de dimensões estarem desocupados, estes chamados de espaços de oportunidade (*spectrum holes*, do inglês) ([Almeida, 2010](#)). A [FIGURA 5](#) demonstra as oportunidades em um determinado local do espectro de frequências, ao longo do tempo.

Figura 5: Representação de oportunidades no espectro de frequência ao longo do tempo.



FONTE: Adaptado de (Golestanian M., 2014)⁵.

Na [FIGURA 5](#) são representadas apenas as dimensões tempo e frequência. Outras dimensões possíveis incluem a localização geográfica, o ângulo de transmissão, a polarização e a utilização de código como no sistema CDMA. ([Almeida, 2010](#))

Para descobrir os espaços de oportunidades disponíveis é necessário utilizar técnicas de sensoriamento de espectro, com o objetivo de reconhecer e armazenar informações a respeito dos dispositivos ao seu redor e com isso aprender a utilizar o espectro de RF de forma eficiente e dinâmica. Entende-se então que o dispositivo que possui tais características está ativamente utilizando-se do conceito de rádio cognitivo, ou seja, está reconhecendo, aprendendo e se adaptando ao meio em que está inserido.

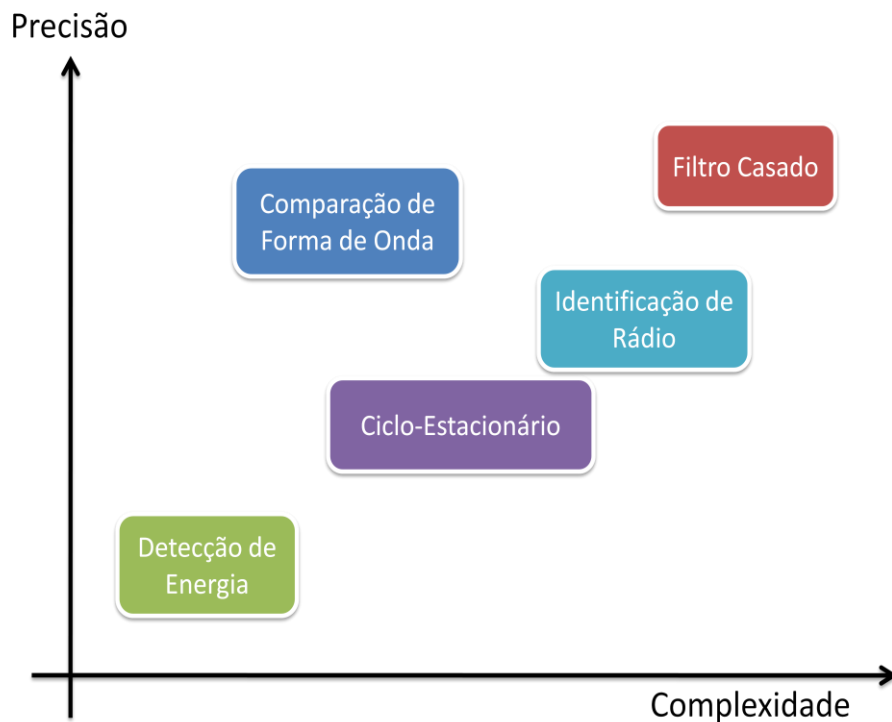
2.4 Técnicas de sensoriamento de espectro

Há diversas maneiras de realizar o sensoriamento do espectro e detectar se um determinado canal está sendo ocupado por alguma fonte de sinal RF. Cada técnica difere em termos de complexidade computacional, matemática e robustez. Técnicas capazes de detectar e identificar qual o agente interferente em um determinado canal com baixa RSR, exigem um hardware completo e um software capaz de implementar algoritmos matematicamente complexos. No outro extremo, um simples detector que indica a presença

⁵ Disponível em: <http://pubs.sciepub.com/iteces/2/3/3/image/fig1.png>.

ou não de algum sinal possui hardware e software com baixo nível de complexidade, porém falha em prover resultados mais detalhados. A relação de compromisso é resolvida com base nas características do sistema, necessidade, custo adicional, e outros fatores. Na [FIGURA 6](#) é possível verificar as principais técnicas de sensoriamento de espectro, qualitativamente explicitando a relação de compromisso entre a precisão e a complexidade.

Figura 6: Comparação das técnicas de sensoriamento de espectro.



Fonte: Elaborada pelo autor.

Cada técnica será brevemente descrita abaixo:

- **Detecção de Energia:** Através do cálculo da potência média, obtido durante um determinado instante de tempo, é feita uma comparação com um valor pré determinado. Caso a potência média obtida seja maior que o nível especificado, o canal é considerado ocupado, caso contrário, o canal é considerado livre. Por ser um método relativamente simples, muitos transceptores já empregam esta função em seu próprio circuito integrado.
- **Ciclo-Estacionário:** Realiza a identificação de parâmetros presentes nas propriedades estatísticas do sinal, como a frequência cíclica. Exige um conhecimento prévio das

características do sinal a ser analisado para a sua identificação, embora consiga distinguir entre diferentes sinais.

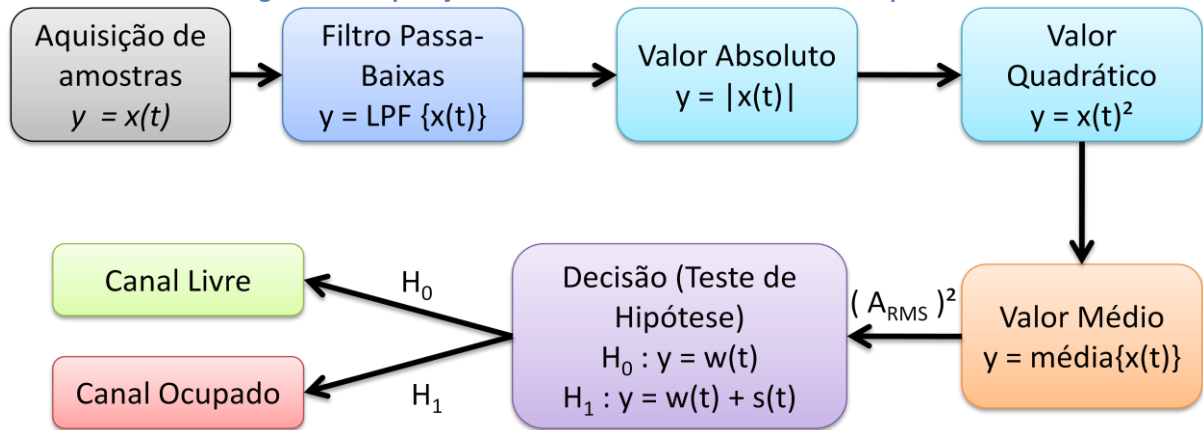
- Comparação de forma de onda: Através da simples comparação entre o sinal adquirido e um sinal armazenado em um banco de dados do dispositivo. Realizando uma correlação cruzada entre estes sinais é possível distingui-los. Necessita de uma memória interna para armazenar os sinais a serem comparados.
- Identificação de rádio: Obtém as principais características do sinal, como frequência central, banda ocupada, número de portadoras, salto de canal, espalhamento espectral, e outras. Estas características são catalogadas e armazenadas no dispositivo.
- Filtro Casado: O sinal obtido é processado através de um filtro casado, maximizando a relação sinal-ruído e realizando a demodulação do sinal. Exige um conhecimento prévio completo do sinal a ser adquirido, o que é impraticável quando se deseja classificar diversos tipos de sinais.

Neste trabalho serão empregados os dois primeiros métodos, a detecção de energia, e o método ciclo-estacionário. Informações detalhadas sobre os outros métodos descritos acima podem ser encontradas em ([Almeida, 2010](#)).

2.4.1 Método de detecção de energia

É o método mais simples e rápido de análise espectral. É necessário adquirir um determinado número de amostras em um canal e calcular a intensidade de potência neste intervalo de tempo. Comparando a potência média com a potência do ruído de fundo, é possível descobrir se o canal está sendo ocupado ou não. Tem como desvantagens o fato de não ser possível identificar qual o dispositivo que está ocupando determinado canal, ou ainda qual a taxa de ocupação temporal. Em alguns casos onde um sinal com relação sinal ruído muito baixa, é possível confundir o sinal com o ruído de fundo, identificando assim erroneamente um canal ocupado de um livre. A [FIGURA 7](#) representa através de um diagrama de blocos as principais etapas para o desenvolvimento do método de detecção de energia.

Figura 7: Comparação das técnicas de sensoriamento de espectro.



Fonte: Elaborada pelo autor.

Após a aquisição das amostras em fase/quadratura do receptor, estas passam por um filtro passa-baixas, removendo o ruído presente nas bandas adjacentes ao canal a ser analisado. Em seguida obtém-se o valor absoluto das amostras complexas, seguido pelo valor quadrático e por fim o valor médio destas. Ao final destes últimos procedimentos o resultado obtido é a amplitude quadrática média $(A_{RMS})^2$ das amostras adquiridas, conforme pode ser visto adiante.

Seguindo para o bloco de decisão, é feito um teste de hipótese, em que definimos a hipótese nula H_0 como sendo aquela em que há apenas o ruído de fundo presente, ou seja, o canal está livre. Caso a hipótese nula não seja verdadeira, é considerada a hipótese alternativa H_1 na qual além do ruído de fundo, algum outro sinal está presente nas amostras, ou seja, o canal está ocupado. Considerando que o ruído de fundo é constituído principalmente por ruído aditivo branco com distribuição gaussiana (AWGN), é definido um nível limite para as duas hipóteses, ou seja, o nível de significância do teste. Decidir qual será este valor limite é o grande desafio do método, uma vez que caso definirmos um limite muito próximo do valor médio quadrático do ruído de fundo, pode-se cometer erros do tipo I, no qual é dito que há um sinal no canal quando na realidade não o há. No outro extremo, definindo um limite muito distante do valor quadrático médio do ruído, podem-se cometer erros do tipo II, quando é dito que o canal está livre, sendo que na realidade há um sinal interferente presente. Ao final da decisão o algoritmo retorna o estado atual do canal para o usuário, indicando o canal como livre ou ocupado, baseado no teste de hipóteses.

O método de detecção de energia requer obter a potência média das amostras recebidas, para realizar a comparação entre o sinal e o ruído. Para isso primeiramente é necessário obter a amplitude RMS (*Root Mean Square*) das amostras de um vetor, que é dada na [EQUAÇÃO 2](#):

$$A_{RMS} = \sqrt{\frac{1}{N} \sum_n |x[n]|^2} \quad 2$$

onde $x[n]$ é o valor da n -ésima amostra, e N o tamanho deste vetor. A partir desta equação, é possível calcular a potência média do sinal, com base na impedância de entrada do sistema Z_{in} . Segue então a [EQUAÇÃO 3](#):

$$P_{m\acute{e}dia} = \frac{A_{RMS}^2}{Z_{in}} \quad 3$$

Para um sistema referenciado à mesma impedância de entrada, com base na [EQUAÇÃO 3](#), a relação sinal-ruído (RSR)⁶ é dada pela [EQUAÇÃO 4](#):

$$RSR = \frac{P_{sinal}}{P_{ru\acute{i}do}} = \frac{A_{RMS\ sinal}^2}{A_{RMS\ ru\acute{i}do}^2} \quad 4$$

onde P_{sinal} é a potência média do sinal presente na banda e $P_{ru\acute{i}do}$ é a potência média do ruído de fundo. Em dB, a relação sinal ruído é dada pela [EQUAÇÃO 5](#):

$$RSR_{dB} = 10 \log_{10}(RSR) \quad 5$$

⁶ Neste trabalho foi utilizada a abreviatura RSR para a relação sinal-ruído em contraste com o termo mais popularmente utilizado em língua inglesa SNR (*signal-to-noise ratio*) por uma questão de padronização.

2.4.2 Método Ciclo estacionário

Sinais modulados possuem algumas características a respeito de suas propriedades estatísticas. Como exemplo, os sinais da classe IEEE 802.11 possuem no início de sua transmissão um prefixo que permite a sincronização dos pacotes, e os sinais da classe IEEE 802.15.4 utilizam um código pseudo-aleatório com uma determinada taxa de pulsos (*chips*) na técnica de espalhamento espectral DSSS. Estes sinais podem ser descritos através de processos ciclo estacionários, ou seja, cujas propriedades estatísticas de segunda ordem, como a autocorrelação, são periódicas no tempo. Então para diferentes intervalos de tempo a autocorrelação apresentará diferentes resultados que tendem a se repetir em uma determinada frequência. Esta frequência recebe o nome de frequência cíclica, e diferentes protocolos de comunicação apresentam frequências cíclicas próprias. A partir destas propriedades é possível diferenciar as classes IEEE 802, baseando-se nos resultados estatísticos distintos que cada tipo de modulação apresenta.

A definição de autocorrelação, no campo de processamento de sinais, é uma função matemática que relaciona as amostras de um determinado sinal com as amostras deste mesmo sinal atrasado em diferentes intervalos. É matematicamente definida como a correlação cruzada entre um sinal por ele mesmo. A autocorrelação discreta é matematicamente definida pela [EQUAÇÃO 6](#) para sinais estacionários no sentido amplo, ou seja, cujas propriedades estatísticas de primeira ordem como média e variância são independentes do tempo:

$$R_{xx}[\tau] = \sum_N x[n] x[n - \tau] \quad 6$$

no qual $x[n]$ é o vetor de amostras e τ é o atraso entre estas amostras e N é o tamanho total do vetor.

Para sinais cujas propriedades estatísticas de primeira ordem são dependentes do tempo, a autocorrelação passa a depender também da variável n , ou seja, nesta situação a autocorrelação não é estática para todo o intervalo de tempo, e sim dependente deste, como é visto na [EQUAÇÃO 7](#):

$$R_{xx}[n, n + \tau] = \sum_N x[n] x[n - \tau] \quad 7$$

A frequência cíclica é obtida a partir da Função de Autocorrelação Cíclica (CAF, do inglês), e esta é derivada da aplicação da função de autocorrelação em diversas janelas no tempo. A Função de Autocorrelação surge do fato que a autocorrelação destes sinais possui um fator de periodicidade, devido às propriedades destes sinais, como os prefixos, taxa de pulsos (*chips*) em sistemas de espalhamento espectral e sinais pilotos. Como a autocorrelação destes sinais é periódica com o tempo, esta pode ser expressa em termos dos seus coeficientes de Fourier, conforme segue o modelo em tempo discreto na [EQUAÇÃO 8](#).

$$R_{xx}[n, n + \tau] = \sum_{\alpha} R_{xx}^{\alpha}[\tau] e^{j2\pi\alpha n} \quad 8$$

Com base na Teoria de Fourier, estes coeficientes são dados pela Série de Fourier de tempo discreto na [EQUAÇÃO 9](#):

$$R_{xx}^{\alpha}[\tau] = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{N-1} R_{xx}[n, n + \tau] e^{-j2\pi\left(\frac{\alpha n}{N}\right)} \quad 9$$

Estes coeficientes formam a Função de Autocorrelação Cíclica $R_{xx}^{\alpha}[\tau]$, onde N corresponde ao tamanho da janela de autocorrelação, e α é a chamada frequência cíclica, dada em Hertz [Hz]. Ainda pelo teorema de Wiener-Khinchin, que relaciona a densidade espectral de potência com a autocorrelação através da transformada de Fourier, pode-se obter a Função de Correlação Espectral (SCF, do inglês), como é visto na [EQUAÇÃO 10](#).

$$S_{xx}^{\alpha}[f] = \mathfrak{F}\{R_{xx}^{\alpha}[\tau]\} \quad 10$$

Para obter a frequência cíclica α , é necessário satisfazer as condições da [EQUAÇÃO 11](#) e da [EQUAÇÃO 12](#).

$$f_s > 2 \max(\alpha, B) \quad 11$$

onde f_s representa a frequência de amostragem e B é a banda máxima do sinal analisado. Esta condição é uma sequência do Teorema de Nyquist, aplicado à frequência cíclica. E a segunda condição:

$$\Delta n \gg N \quad 12$$

onde Δn é o número total de amostras obtidas em um determinado instante de tempo Δt . Ou seja, é necessário que o total de amostras seja muito maior que o tamanho da janela de autocorrelação, para obter uma resolução na frequência cíclica mínima.

3 Proposta da ferramenta de sensoriamento de espectro

Nesta seção será apresentada a aplicação que tem como finalidade realizar o sensoriamento do espectro na faixa ISM de 2,4GHz, e identificar padrões de sinais conhecidos através dos métodos de detecção de energia e ciclo-estacionário. São apresentadas as ferramentas e materiais, bem como o desenvolvimento dos *scripts* utilizados para realizar o sensoriamento do espectro.

3.1 Ferramentas e Materiais

Para realizar a análise do espectro é utilizado um rádio definido por *software* (*USRP – Universal Software Radio Peripheral, do inglês*) que consiste de um transceptor flexível, que pode ser utilizado para analisar uma ampla faixa de espectro através de um conjunto de sincronizadores, *PLLs (Phase Locked Loop)*, conversores AD (Analogico-Digital) e DA (Digital-Analogico), amplificadores LNA (*Low Noise Amplifier*) e outras ferramentas, e pode ser configurado facilmente através de uma variedade de *softwares* comerciais e de código livre citados adiante. Possuem diversos modelos, alguns com mais ou menos funcionalidades, e com faixas de frequências diferentes, alguns capazes de referenciar o seu *clock* interno a partir de um receptor de GPS.

Para o desenvolvimento da aplicação foi utilizado um dispositivo *USRP* da *National Instruments* em conjunto com a *Ettus Research*, o NI-2932, que pode ser utilizado juntamente com diversos *softwares* de alto nível. Este dispositivo pode ser visto na [FIGURA 8](#).

Figura 8: Rádio Definido por Software NI-USRP 2932.



Fonte: Página da *National Instruments*⁷.

Alguns *softwares* compatíveis são:

- LabVIEW: Propriedade da National Instruments, possui uma vasta gama de ferramentas de processamento de sinal. Possui interface gráfica composta por diagrama de blocos, cada um realizando uma função no processamento das amostras.
- GNU Radio: Foi estruturado com o objetivo de ser uma alternativa de código aberto para ser utilizado em Linux. Similar ao LabVIEW, porém com um número de ferramentas reduzido. É possível utilizar programação em Python para realizar processamentos mais complexos, porém exige um nível maior de conhecimento por parte do programador.
- MATLAB/Simulink: O Simulink juntamente com o MATLAB permite ao usuário desenvolver os blocos através da programação em código de MATLAB. Possui um grande poder de processamento e análise de dados. É propriedade da MathWorks.

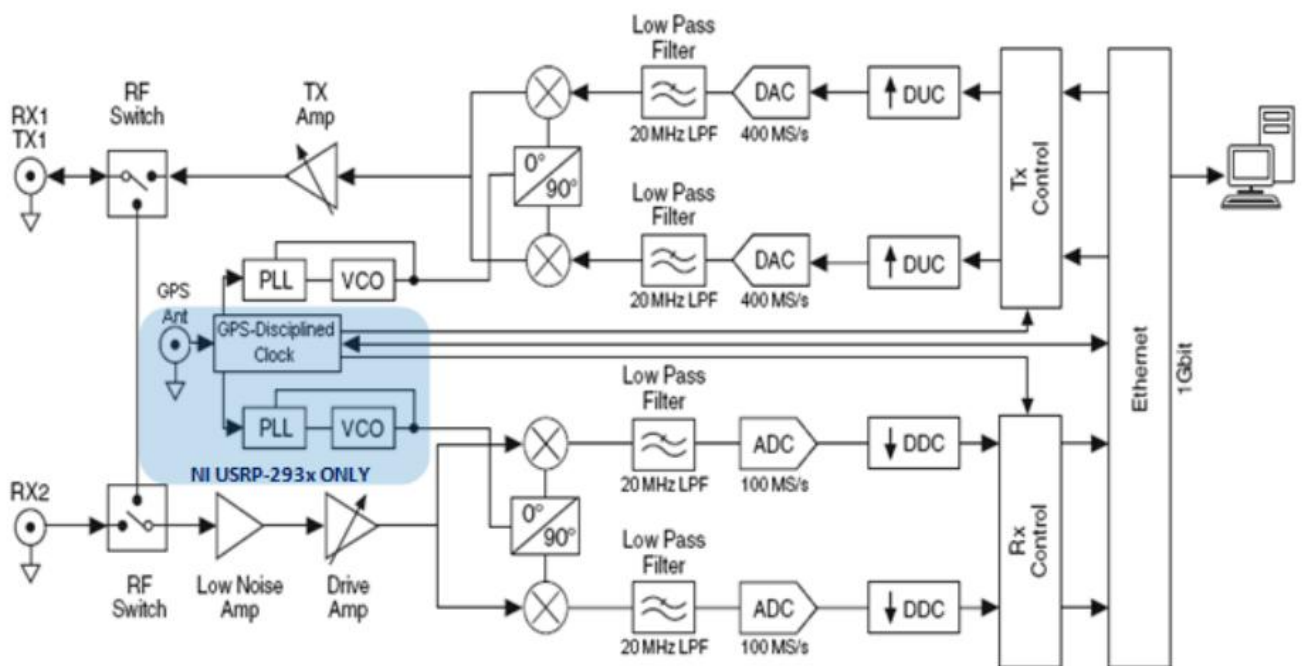
Algumas características deste dispositivo incluem:

⁷ Disponível em: <http://sine.ni.com/nips/cds/view/p/lang/pt/nid/212999>.

- Operação na faixa de 400MHz à 4,4GHz;
- Precisão na frequência de 2,5ppm
- Taxa de amostragem máxima em Fase/Quadratura (I/Q):
 - 16 bits: 25MHz;
 - 8 bits: 50MHz;
- Banda instantânea em tempo real:
 - 16 bits: 20MHz;
 - 8 bits: 40MHz;
- Conversores DA de dois canais, à 100MS/s, em 14 bits.
- Impedância de entrada de 50Ω.
- Operável com precisão *double* em ponto flutuante.

É possível ver pela [FIGURA 9](#) o diagrama de blocos do dispositivo *USRP*:

Figura 9: Diagrama funcional do hardware NI-USRP 2932.



FONTE: Datasheet NI-USRP 292x/293x⁸.

⁸ Disponível em: <http://www.ni.com/datasheet/pdf/en/ds-355>.

O dispositivo pode operar tanto como transmissor pela saída TX1, ou como receptor tanto pela entrada RX1 ou RX2, porém apenas de forma não simultânea. Um par de chaves faz o controle entre o sinal a ser transmitido ou o sinal recebido.

O sinal recebido é amplificado por um amplificador de baixo ruído (LNA), seguido por um amplificador de ganho. O dispositivo possui um gerador de *clock* interno, que pode ter sua precisão aperfeiçoada utilizando o sinal PPS oriundo de uma entrada para GPS. Esse *clock* passa pelo conjunto PLL e VCO (*Voltage Controlled Oscillator*) que geram uma frequência precisa na faixa de microondas. Esse *clock* gera o par em fase/quadratura (0° e 90°) que servirá para demodular o sinal recebido no *mixer* para a banda base. Após, o par de sinais fase/quadratura passa por um filtro de *anti-aliasing* de 20MHz para poder seguir pelo conversor AD em 100MS/s. Esse sinal super-amostrado passa então por um *down-sampling* de acordo com o valor declarado na camada de aplicação. Finalmente o sinal passa pelo bloco de controle, que fará o gerenciamento com a camada de rede. O sinal transmitido segue o caminho inverso do processo descrito anteriormente, porém com um conversor DA com taxa de 400MS/s.

É necessário explicitar que este dispositivo não é calibrado de fábrica, ou seja, as amostras em sua saída não compreendem um valor de tensão elétrica conhecido. As amostras complexas estão situadas na faixa entre -1 e 1, discretizadas de acordo com o número de bits selecionado. Para obter o valor das amostras em nível de tensão elétrica foi realizada uma calibração em laboratório, a partir de um sinal com potência de saída conhecida. A calibração tem como fins obter um valor aproximado para a estimativa da potência do sinal e do ruído de fundo do dispositivo USRP.

O sinal para calibração foi gerado utilizando uma plataforma de desenvolvimento que consiste de um transceptor em conformidade com o padrão IEEE802.15.4 e um microcontrolador ARM Cortex-M4. É produzido pela *Freescale Semiconductor*, modelo MKW24D512V, e pode ser visto na [FIGURA 10](#).

Figura 10: Placa de desenvolvimento para os testes com o padrão IEEE 802.15.4.



Fonte: Página da Digi-Key⁹.

A potência de saída desta placa de desenvolvimento foi medida utilizando um analisador de redes RF, modelo FieldFox N9912A – 6GHz, produzido pela *Agilent Technologies*, visto na [FIGURA 11](#). Na entrada da ponteira de prova do analisador é colocado um atenuador de 30dB de forma a evitar possíveis danos devido a potência elevada.

⁹ Disponível em: <http://media.digikey.com/Photos/Freescale%20Photos/TWR-KW24D512.jpg>.

Figura 11: Analisador de redes FieldFox N9912A .

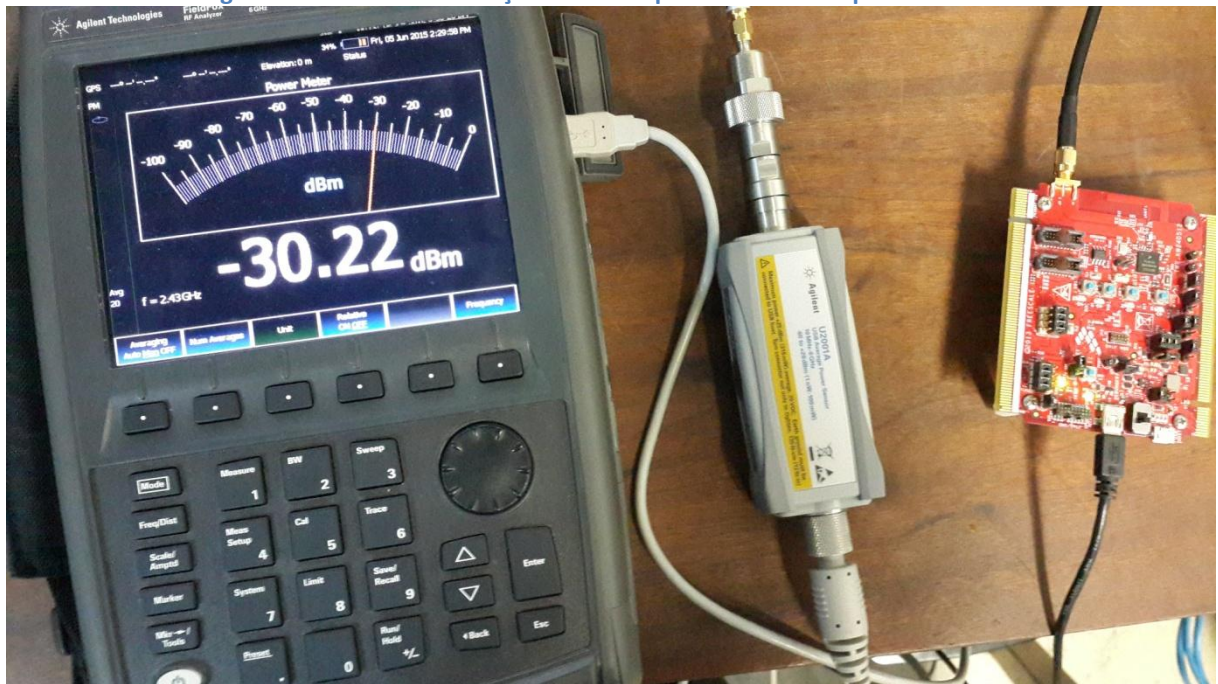


Fonte: Página da Keysight Technologies¹⁰.

A [FIGURA 12](#) mostra a potência média medida, que foi utilizada como base para calibrar o dispositivo *USRP*.

¹⁰ Disponível em: <http://www.keysight.com/en/pd-1456189-pn-N9912A/fieldfox-rf-analyzer-handheld-cable-and-antenna-analyzer-and-handheld-spectrum-analyzer-4-6-ghz?cc=BR&lc=por>.

Figura 12: Ensaio de calibração realizado para calibrar o dispositivo *USRP*.



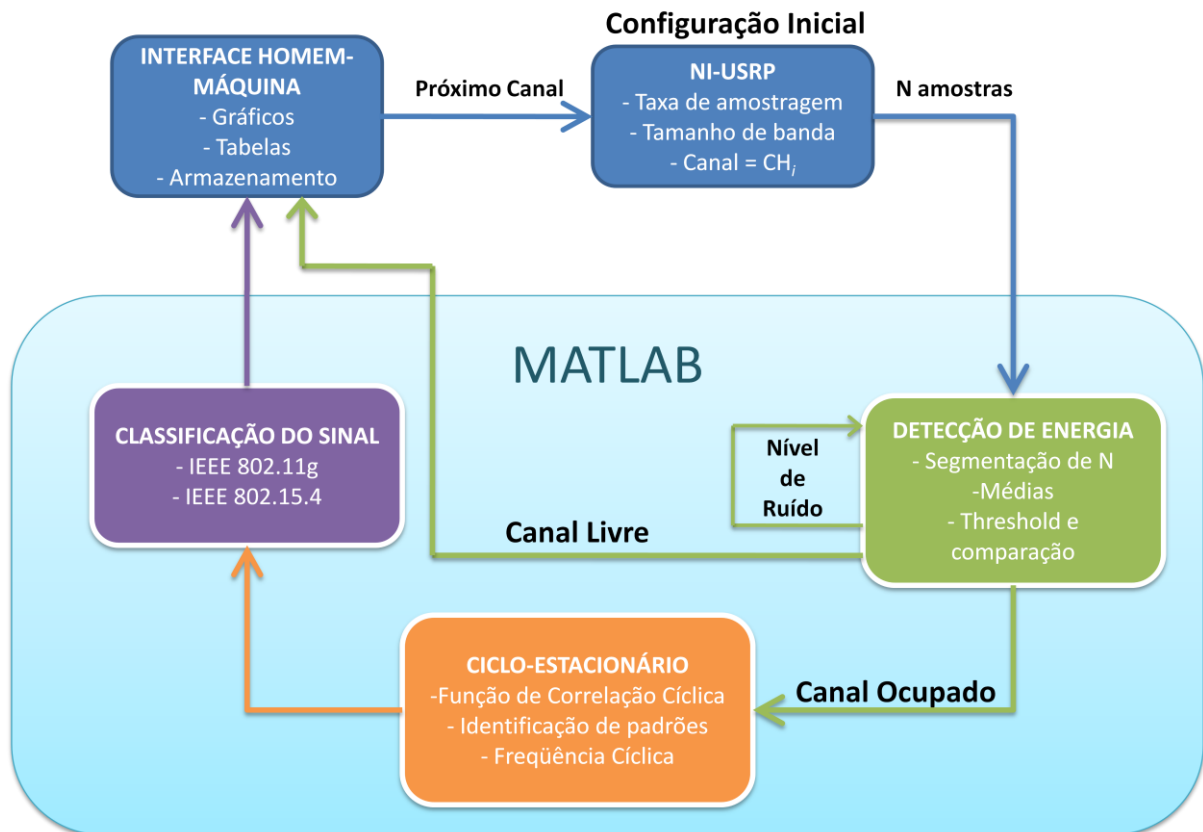
Fonte: Elaborada pelo autor.

Com base no valor obtido na [FIGURA 12](#) pelo analisador de redes, foi possível calibrar o dispositivo *USRP* através do algoritmo de detecção de energia, que possui em sua saída um valor para a estimativa da potência média do sinal presente no canal.

3.2 Diagrama de Blocos

O diagrama geral do funcionamento e operação da aplicação pode ser visto na [FIGURA 13](#).

Figura 13: Diagrama de blocos geral.



Fonte: Elaborada pelo autor.

É possível verificar que o programa é dividido em duas partes, sendo a primeira responsável pela configuração do dispositivo e aquisição das amostras recebidas, assim como da interface homem-máquina, realizada pelo LabVIEW. A segunda é responsável pelo processamento computacional e execução dos métodos de detecção de energia e ciclo-estacionário, realizada em *script* do MATLAB. Cada bloco será detalhado nas subseções seguintes.

3.2.1 Configuração inicial NI-USRP

É o conjunto de operações que controlam o hardware NI-USRP 2932, conectado ao computador através de um cabo Ethernet. Todas as operações e configuração do *hardware* são realizadas via *software* pelo LabVIEW, e são elas:

- **Nome do dispositivo/IP:** É o IP do dispositivo que está conectado ao computador através de uma conexão Ethernet. O IP de fábrica está escrito na memória interna

EEPROM, e pode ser modificado utilizando um executável distribuído pela *National Instruments*. O IP padrão é 192.168.10.2. No hardware utilizado, este foi modificado para 10.1.36.20, para entrar em conformidade com o IP da rede.

- **Taxa de amostragem:** Dada em amostras por segundo [S/sec], especifica a taxa na qual as amostras são capturadas pelo conversor AD. É configurada uma taxa de 25MS/seg com resolução de 16bits. Como o sinal demodulado está centrado na banda base, nesta taxa obtém-se a banda equivalente de 25MHz. Devido ao filtro passa-baixas interno do dispositivo, a banda útil é na realidade de 20MHz.
- **Banda de análise:** A banda total do sinal está diretamente relacionada a taxa de amostragem especificada.
- **Antena Ativa:** Escolha do canal no qual a antena está conectada, RX1/TX ou RX2.
- **Frequência Central:** Frequência de operação do PLL, que realizará a demodulação para a banda base. No programa esta frequência será alterada constantemente, a fim de realizar a varredura no espectro de 2,4GHz. A aplicação permite escolher qual a frequência inicial e a final que deve ser analisada dentro da faixa ISM de 2,4GHz, assim como o intervalo entre cada canal dado por ΔCH . A banda de cada canal é fixa em 5MHz. A frequência central de cada canal é dada pela [EQUAÇÃO 13](#):

$$CH_i = 2,4[GHz] + f_{inic} [MHz] + i \cdot \Delta CH [MHz] \quad | \quad i = \left\{ 0, \dots, \frac{f_{final} - f_{inic}}{\Delta CH} \right\} \quad 13$$

- **Ganho:** Especifica o ganho do amplificador de entrada, dado em dB. O ganho atual está ajustado para 0dB.

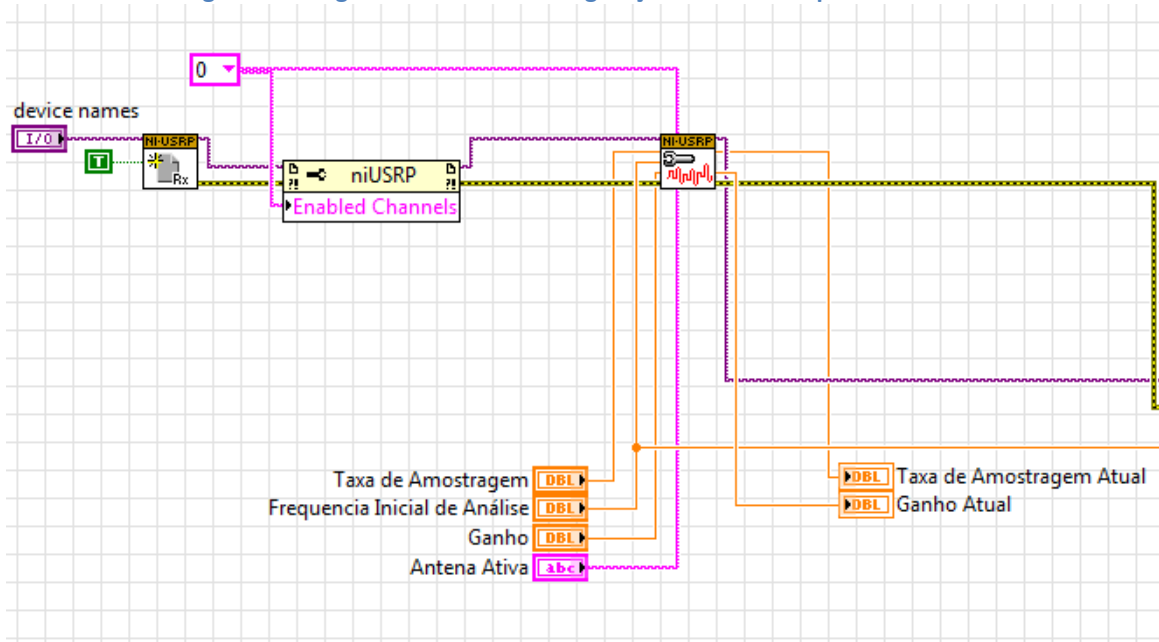
Os blocos (Vis) niUSRP utilizados são:

- **Open Rx Session:** A partir do nome do dispositivo/IP, obtém o endereço para a aquisição dos dados.
- **Configure Signal:** Configura o dispositivo com as informações previamente descritas, como frequência de amostragem, banda, etc.
- **Initiate:** Inicializa o dispositivo USRP para coletar as amostras e guardar em seu buffer interno.

- **Fetch Rx Data:** Obtém as amostras do *buffer* do dispositivo e reorganiza estas em um vetor de dados complexo, no formato *double-precision floating-point*.

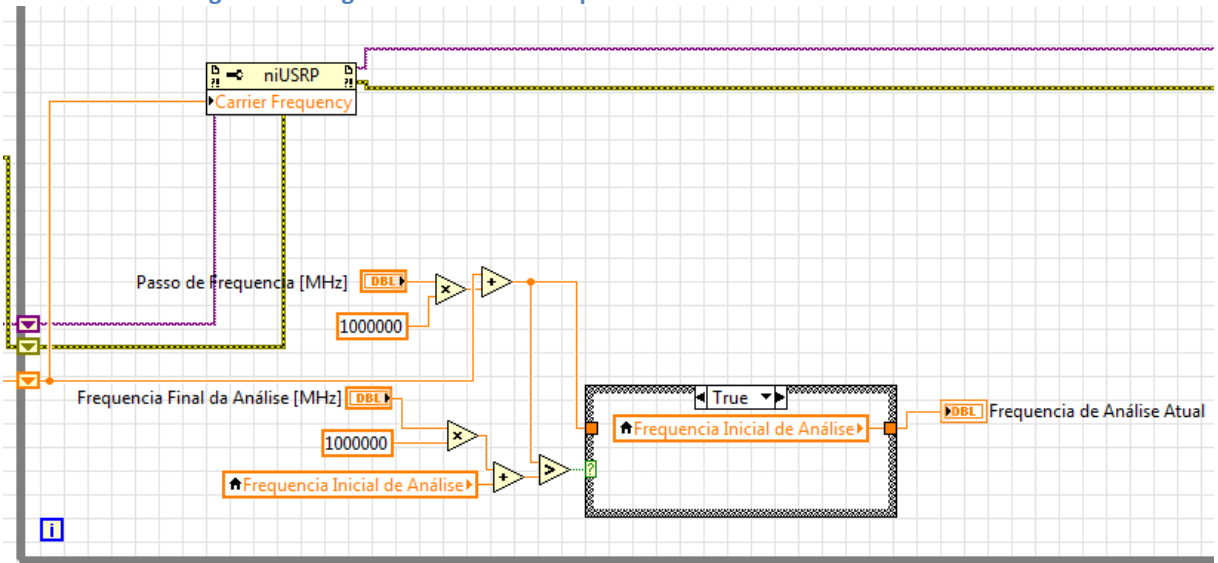
A [FIGURA 14](#) mostra o painel de blocos do LabVIEW para a configuração inicial do dispositivo, enquanto que a [FIGURA 15](#) mostra os blocos utilizados para realizar a varredura do espectro na faixa de 2,4GHz, dentro de um laço de repetição. A [FIGURA 16](#) mostra os blocos responsáveis pelo cálculo do número de amostras a ser recebidas pelo dispositivo USRP, assim como pela coleta destas.

Figura 14: Diagrama de blocos: configuração inicial do dispositivo USRP.



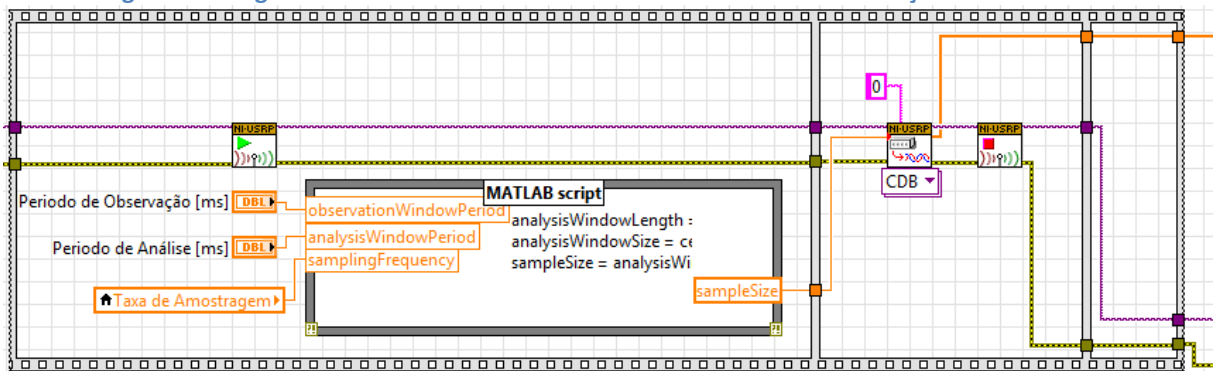
Fonte: Elaborada pelo autor.

Figura 15: Diagrama de blocos: Frequência central do canal a ser analisado.



Fonte: Elaborada pelo autor.

Figura 16: Diagrama de blocos: Tamanho das amostras a receber e obtenção das amostras.



Fonte: Elaborada pelo autor.

3.2.2 Algoritmo de Detecção de Energia

O algoritmo de detecção de energia é implementado em um *script* de MATLAB, rodando conjuntamente com o LabVIEW. Tem o objetivo principal de realizar uma identificação e separação entre os canais que possuem algum sinal e aqueles que não estão sendo ocupados. É um algoritmo recursivo, no qual a potência média do ruído é constantemente recalculada a cada novo vetor de amostras, de forma a convergir ao valor médio presente no ambiente. Desta forma o algoritmo é adaptativo, no sentido que se adapta ao ambiente em que é feita a medição.

As configurações de entrada são repassadas através da interface do LabVIEW, sendo elas:

- **Nível de *threshold* [dB]:** Mínima RSR necessária para distinguir um sinal de ruído. Sinais que possuem potência média inferior ao nível de *threshold* são considerados como parte do ruído de fundo. Um valor muito elevado para *threshold* pode mascarar alguns sinais de baixa amplitude, enquanto que um valor relativamente baixo pode ocasionar o reconhecimento de ruído como sinal. Como padrão este valor é configurado em 3dB.
- **Nível de Ruído Inicial [dBm]:** É o valor inicial atribuído pelo usuário para uma estimativa para a potência de ruído. É utilizado apenas na primeira vez que o *script* é executado. Após o primeiro ciclo de amostras, este valor tende a se estabilizar ao valor médio do ruído de fundo do dispositivo *USRP*. Como padrão este valor é configurado em -76dBm.
- **Período de Observação [ms]:** É o período de tempo total no qual é feita a coleta de amostras no canal especificado. O valor padrão é de 50ms.
- **Período de Análise [ms]:** As amostras recebidas no intervalo de tempo dado pelo Período de Observação são divididas em janelas de análises, com intervalo dado pelo Período de Análise. Estas janelas permitem separar de forma eficiente sinais que são recebidos de forma não contínua. Assim apenas as janelas que possuem sinal são analisadas. A resolução na frequência cíclica é diretamente proporcional ao Período de Análise, conforme [EQUAÇÃO 15](#).

As variáveis de saída são:

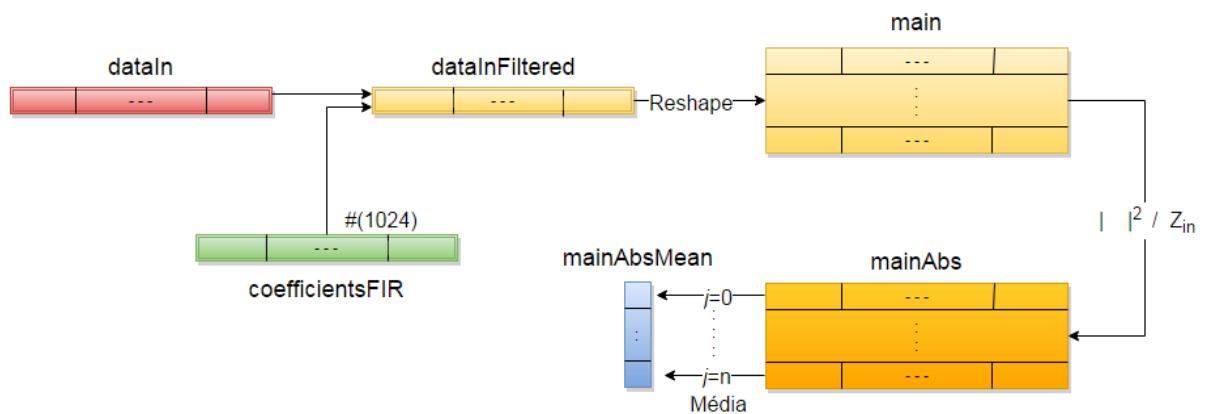
- **Potência Média do Sinal [dBm]:** Estimativa para a potência média do sinal detectado. É dado pelo valor médio das janelas de análise que possuem sinal.
- **Potência Média do Ruído [dBm]:** Estimativa para a potência média do ruído de fundo do dispositivo *USRP*. Nos testes realizados em laboratório este valor se encontra próximo à -76dBm. Uma justificativa para este valor elevado é a falta de resolução do dispositivo, por abranger uma escala relativamente larga de amplitudes. Logo seria

necessária uma resolução muito maior do que 16 bits para obter sinais com amplitudes menores que este valor.

- **RSR [dB]:** Relação Sinal-Ruído entre o sinal adquirido e o ruído de fundo, dado em dB.

O primeiro diagrama de blocos, na [FIGURA 17](#), prepara os dados recebidos para poderem ser tratados. Primeiramente é necessário multiplicar as constantes recebidas por um fator de calibração, obtido a partir do ensaio de calibração previamente realizado em laboratório. Esta constante de calibração transforma o valor adimensional das amostras em nível de tensão elétrica, dado pelo vetor com valores complexos *dataIn*.

Figura 17: Detecção de Energia: diagrama de blocos 1.

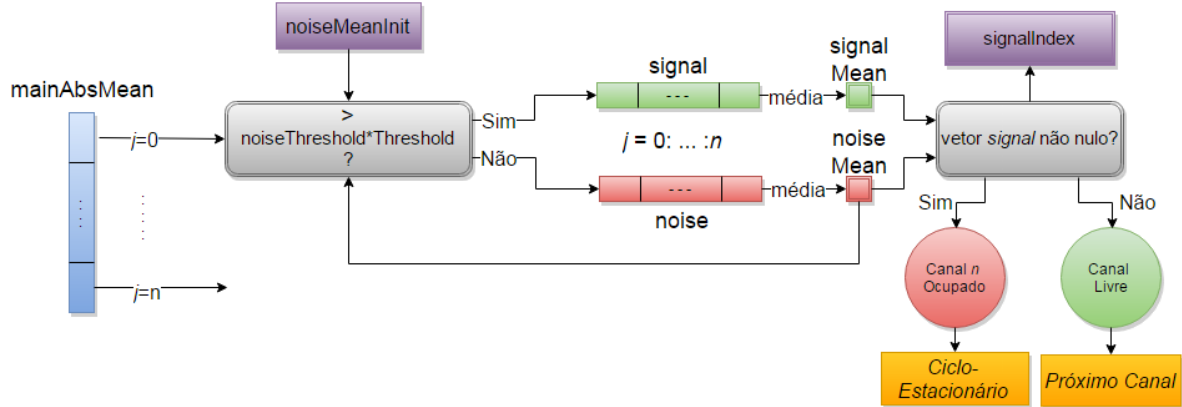


Fonte: Elaborada pelo autor.

Após, este vetor passa por um filtro FIR passa-baixas utilizando os coeficientes armazenados em *coefficientsFIR*. O filtro possui frequência de corte igual a $f_c = 0.4 \frac{f_s}{2}$, onde f_s é a frequência de amostragem. Para $f_s = 25MS/s$, a frequência de corte é de $f_c = 5MHz$, e o resultado é armazenado no vetor *dataInFiltered*. Este vetor é reordenado na matriz *main*, no qual cada linha corresponde a uma janela de análise. Desta matriz é obtido o valor absoluto quadrático normalizado pela impedância de entrada, e é armazenado em *mainAbs*. Realizado a média de cada linha se obtém a potência média para cada janela de análise conforme [EQUAÇÃO 2](#), que é salva no vetor coluna *mainAbsMean*.

A próxima etapa do algoritmo é responsável pela comparação e decisão, que indicará se o canal está livre ou ocupado. O segundo diagrama de blocos é visto na [FIGURA 18](#).

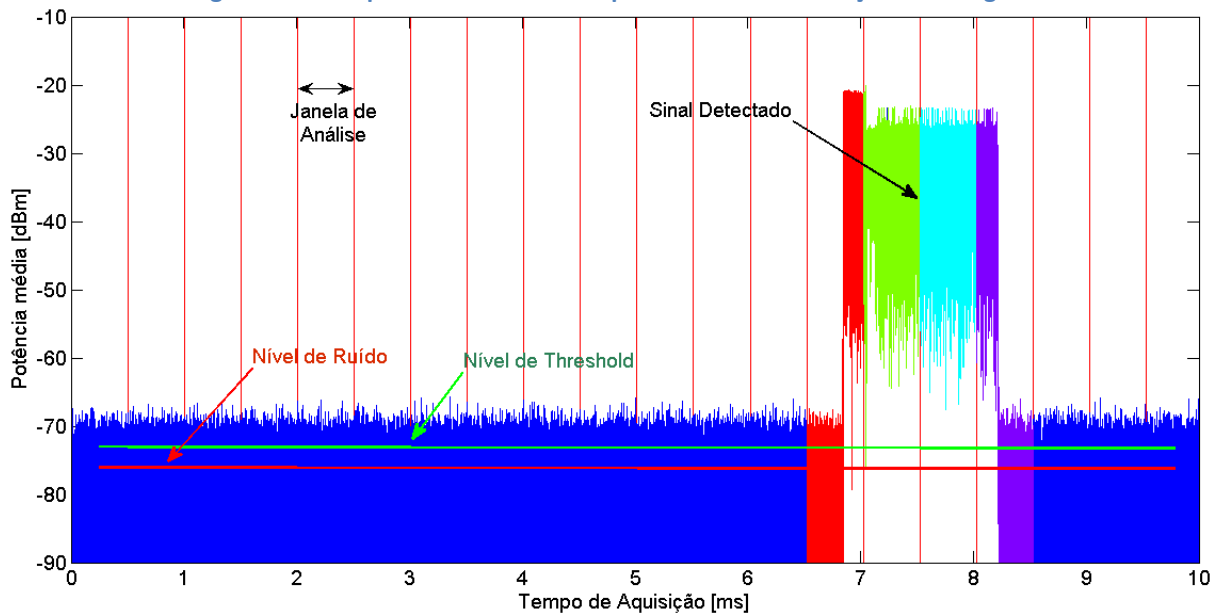
Figura 18: Detecção de Energia: diagrama de blocos 2.



Fonte: Elaborada pelo autor.

Conforme pode ser visto na [FIGURA 18](#), cada elemento de *mainAbsMean* passa por um comparador, que verifica se a potência média da janela de análise é maior que o nível de *threshold*. Isto é, o comparador verifica se a RSR entre o sinal recebido e o nível de ruído é maior que o definido pelo nível de *threshold*. A estimativa inicial para o ruído é dada pela variável *noiseMeanInit*. Caso a resposta do comparador seja afirmativa, então esse valor é colocado no vetor *signal*, ou seja, representa a potência média do sinal para aquela janela de análise. Caso negativo o valor é colocado no vetor *noise*, representando a potência média do ruído, que subsequentemente é realimentada no comparador. Esse processo é repetido para cada elemento de *mainAbsMean*. Ao fim há dois vetores, *signal* e *noise*. Caso haja algum valor armazenado no vetor *signal* é porque foi encontrado um sinal em alguma janela de análise. Logo o canal é classificado como ocupado, e a janela de análise na qual foi encontrado o sinal é analisada pelo algoritmo seguinte, o ciclo-estacionário. O vetor *signalIndex* identifica os índices das janelas de análise que possuem sinal. Caso não haja nenhum valor no vetor *signal* é porque não foi encontrado nenhum sinal ocupando o canal no intervalo de tempo de observação. A [FIGURA 19](#) representa um sinal exemplo analisado pelo algoritmo de detecção de energia.

Figura 19: Exemplo de sinal analisado pelo método de detecção de energia.

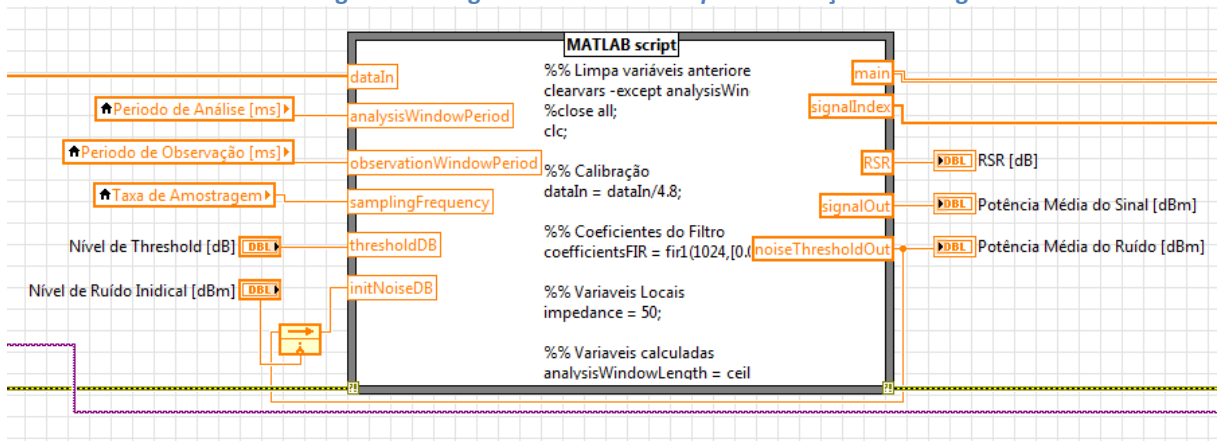


Fonte: Elaborada pelo autor.

Na [FIGURA 19](#) as janelas de análise representadas por retas verticais paralelas são as linhas da matriz *main* onde em cada janela é comparada a potência média do ruído (listra em vermelho), com a potência média das amostras nesta janela. Caso a potência média seja maior que o nível de *threshold* (listra em verde), classificam-se as amostras desta janela como parte de um sinal. O índice desta janela é dado pelo vetor *signalIndex*, e as janelas que possuem algum sinal são posteriormente analisadas pelo algoritmo ciclo-estacionário. Caso não haja nenhum sinal nas amostras analisadas, a aplicação prossegue para o canal seguinte. Neste exemplo é possível observar um sinal com duração aproximada de 1,5ms que é identificado em quatro janelas de análise entre 6,75ms e 8,25ms. Estas janelas serão posteriormente analisadas pelo algoritmo ciclo-estacionário.

A [FIGURA 20](#) mostra o *script* de detecção de energia sendo executado dentro do ambiente do LabVIEW. O *script* completo utilizado se encontra no [Anexo A](#).

Figura 20: Diagrama de blocos: *Script* de detecção de energia.



Fonte: Elaborada pelo autor.

3.2.3 Algoritmo Ciclo-Estacionário

Conforme abordado na [SEÇÃO 2.4.2](#), pode-se classificar diversos tipos de sinais a partir de suas características de modulação, portadoras pilotos, tamanho do pseudocódigo em técnicas de espalhamento espectral, e prefixos de sincronização no início da transmissão. Estas características apresentam propriedades estatísticas periódicas com o tempo, que podem ser obtidas através de sua frequência cíclica.

Após as amostras passarem pelo método de detecção de energia, cada janela de análise que possui sinal é processada pelo algoritmo ciclo-estacionário, onde os índices destas janelas são dadas pelo vetor *signalIndex*. As configurações de entrada do algoritmo são:

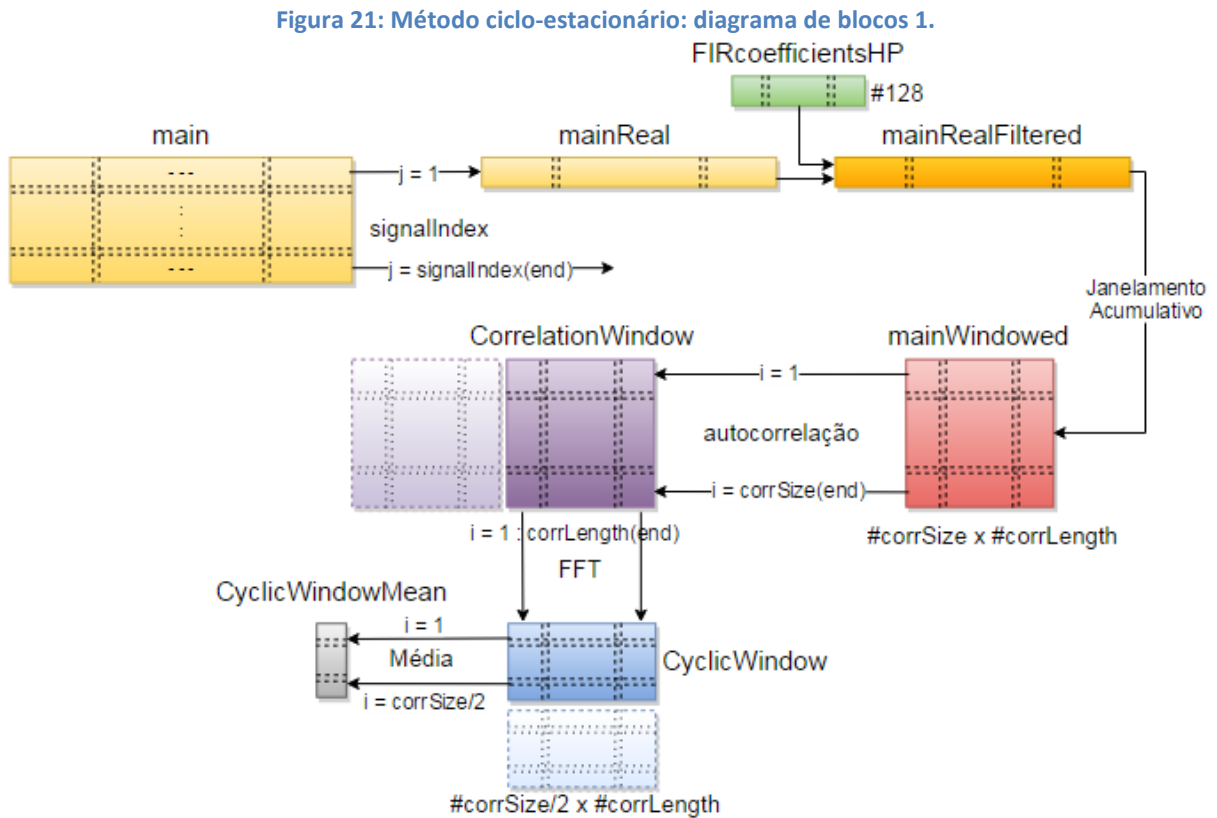
- **Período da Autocorrelação [μs]:** É o período de tempo no qual a janela de autocorrelação é processada. Define também o atraso máximo entre amostras τ na função de autocorrelação, conforme [EQUAÇÃO 7](#). Padrão igual a $4\mu\text{s}$.
- **Espaçamento entre Janelas de Autocorrelação [Δn amostras]:** É representado por n na [EQUAÇÃO 7](#), porém a autocorrelação não é realizada para todas as n amostras, e sim a cada Δn amostras. Por esta razão, este parâmetro define a frequência de amostragem cíclica, uma vez que é diretamente proporcional ao número de janelas de autocorrelação, e conseqüentemente ao tamanho da FFT da frequência cíclica. Como padrão foi definido igual a 6 amostras.

As variáveis recebidas do algoritmo de detecção de energia são:

- **main**: Matriz que contém cada uma das janelas de análise em suas linhas.
- **signalIndex**: Vetor de índices que indicam quais as linhas de *main* que contém sinal a ser analisado.

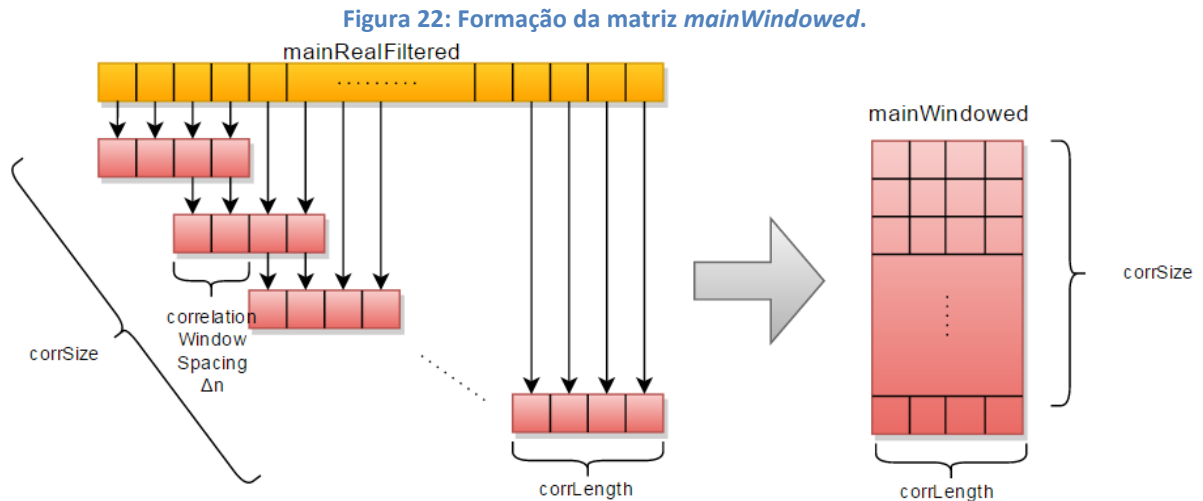
A saída do algoritmo é o resultado da classificação obtida. O sinal é classificado como pertencente ao padrão IEEE 802.11g ou ao padrão IEEE 802.15.4, ou como um padrão desconhecido, quando é distinto destes dois tipos de sinais.

O primeiro diagrama de blocos do método aplicado pode ser visto na [FIGURA 21](#).



Cada janela de análise indicada pelo vetor *signalIndex* com provável sinal é analisada separadamente dentro de um laço de repetição. Desta janela é obtido sua componente em fase *mainReal* e o nível médio é filtrado através de um filtro FIR passa-altas, dado pelos

coeficientes armazenados em *FIRcoefficientsHP*. O sinal filtrado é armazenado no vetor *mainRealFiltered*. Em seguida as amostras são reordenadas na matriz *mainWindowed*, a partir de um processo de janelamento cumulativo, conforme pode ser visto na [FIGURA 22](#).



Fonte: Elaborada pelo autor.

A matriz *mainWindowed* possui o número de colunas dado pela variável *corrLength*, que representa o tamanho da janela de autocorrelação em amostras; e número de linhas dado pela variável calculada *corrSize*. Esta variável é obtida conforme segue na [EQUAÇÃO 14](#).

$$corrSize = \frac{length(main) - corrLength}{\Delta n} \quad 14$$

Em seguida é obtido dentro de um laço de repetição, a autocorrelação de cada linha da matriz de *mainWindowed*, gerando a matriz *corrWindow* com número de linhas dado por *corrSize* e número de colunas dado por duas vezes *corrLength*. Uma vez que a autocorrelação de um sinal real é uma função par, são descartadas as colunas simétricas da matriz. Esta matriz resultante representa a [EQUAÇÃO 7](#) no qual cada linha representa um instante no tempo discreto n , e as colunas o atraso entre amostras indicado por $n + \tau$.

Conforme visto na [EQUAÇÃO 8](#), pode-se representar a função de autocorrelação de sinais com propriedades cíclicas através de suas componentes dadas pela série de Fourier de tempo discreto, conforme [EQUAÇÃO 9](#). A estimação desta série de Fourier é dada pela FFT (Fast Fourier Transform), que é computacionalmente mais eficiente. Então é obtida a FFT de

cada coluna da matriz *corrWindow*, resultando na matriz *cyclicWindow*, com dimensão de linhas dado por *corrSize* e de colunas por *corrLength*. Uma vez que a FFT de um vetor real é uma função par, novamente são eliminadas as linhas simétricas da matriz. A matriz *cyclicWindow* corresponde à função de autocorrelação cíclica, no qual suas linhas correspondem ao atraso de amostras $n + \tau$, e suas colunas representam a frequência cíclica α .

A frequência de amostragem cíclica α_s é dada pela [EQUAÇÃO 15](#).

$$\alpha_s = \frac{1}{\Delta t_\alpha} [Hz] \quad 15$$

no qual Δt_α representa o intervalo de tempo cíclico, dado pela [EQUAÇÃO 16](#).

$$\Delta t_\alpha = \frac{\Delta n}{f_s} [seg] \quad 16$$

onde Δn é o espaçamento entre janelas de autocorrelação dado pela variável *correlationWindowSpacing* e f_s é a frequência de amostragem do dispositivo *USRP*. A resolução na frequência cíclica é dada na [EQUAÇÃO 17](#):

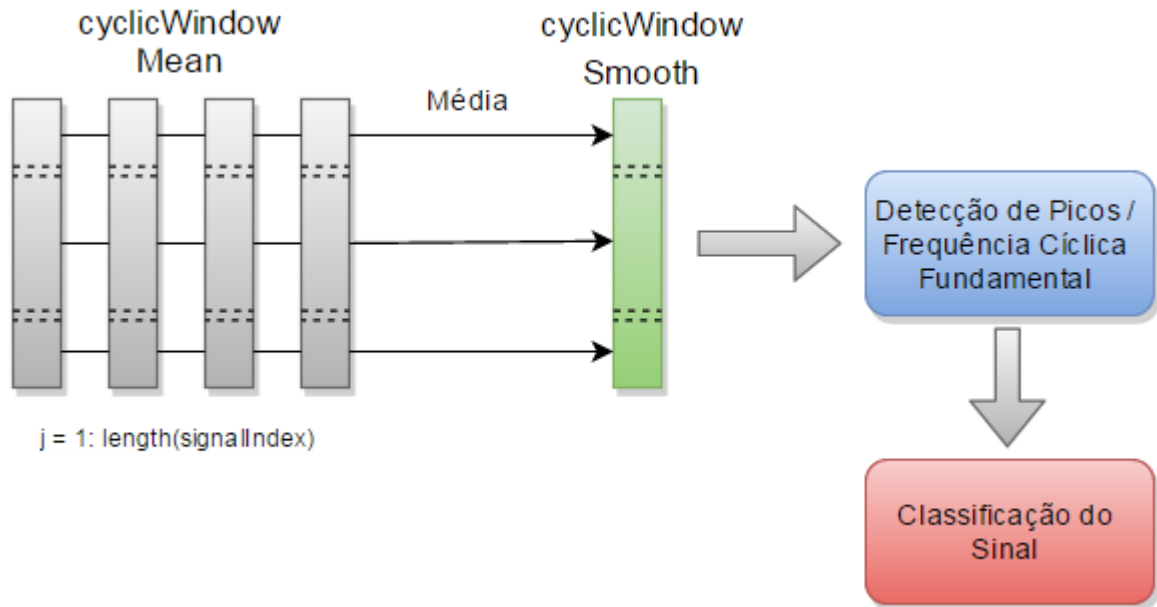
$$\Delta\alpha = \frac{\alpha_s}{corrSize} [Hz] \quad 17$$

A partir da matriz *cyclicWindow* é obtida a média de cada linha, dentro de um laço de repetição, resultando no vetor coluna *cyclicWindowMean*, que possui comprimento igual a *corrSize/2*. Neste vetor estão representadas as componentes da frequência cíclica que formam a função de autocorrelação cíclica, como pode ser representado pela [EQUAÇÃO 18](#).

$$R_{xx}[\alpha] = \frac{\sum_{\tau} R_{xx}[\alpha, \tau]}{length(\tau)} \quad 18$$

Após obter o vetor de autocorrelação cíclica, o laço de repetição em *main* segue até que todas as janelas de análise dadas pelos índices de *signalIndex* sejam processadas. Este processo pode ser visto no segundo diagrama de blocos, na [FIGURA 23](#).

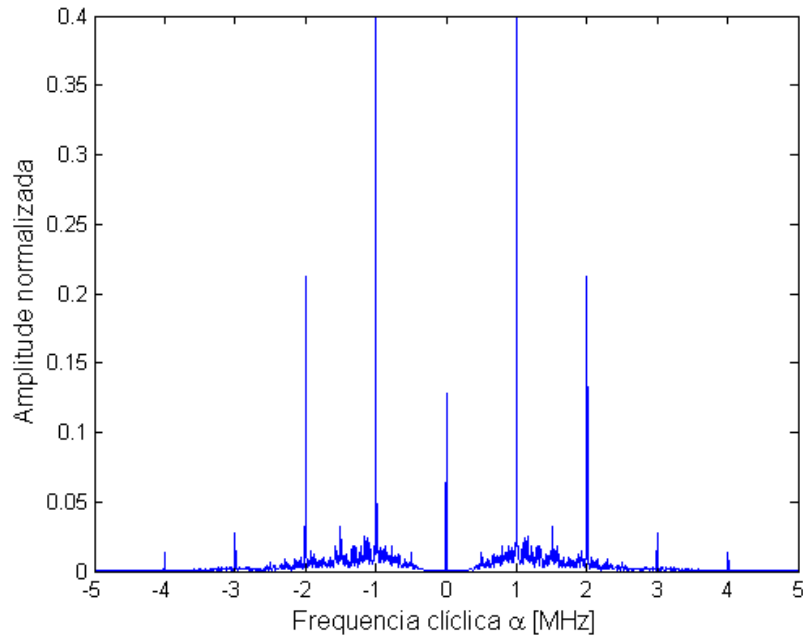
Figura 23: Método ciclo-estacionário: diagrama de blocos 2.



Fonte: Elaborada pelo autor.

Após, é realizado um processamento de suavização nas amostras através da média do resultado de cada janela de análise, gerando o vetor coluna final *cyclicWindowSmooth*. Deste vetor é possível obter as componentes da frequência cíclica através de detecção de picos. A [FIGURA 24](#) apresenta a função de autocorrelação cíclica de um sinal exemplo. É possível ver que as componentes do sinal estão representadas através de funções Delta de Dirac ($\delta(x)$).

Figura 24: Estimação da Função de Autocorrelação Cíclica de um sinal exemplo.

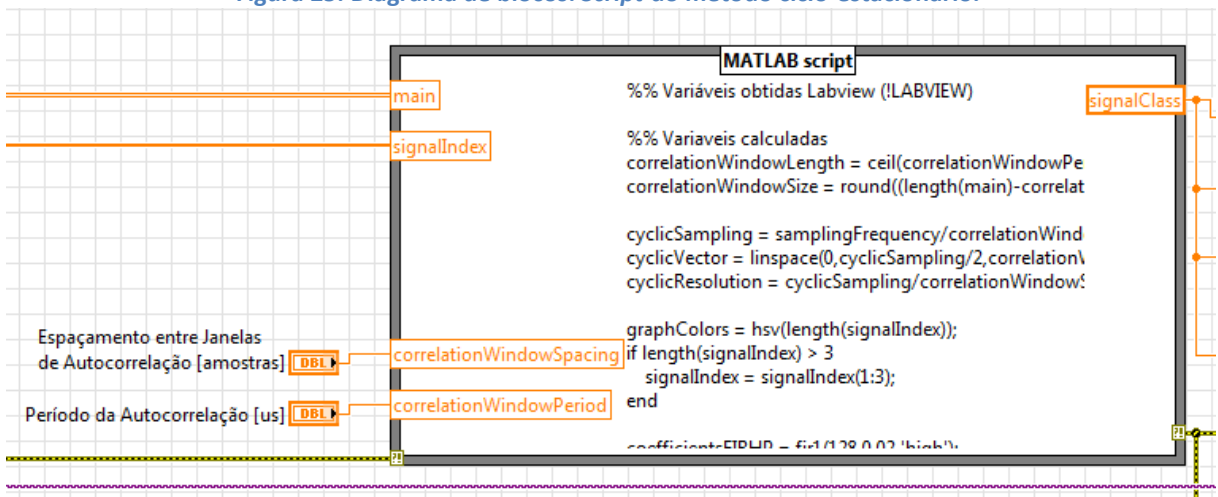


Fonte: Elaborada pelo autor.

Obtendo a frequência cíclica fundamental de cada sinal é possível classificá-los de acordo, uma vez que possuem diferentes fundamentais e harmônicas distintas.

A [FIGURA 25](#) mostra o *script* do método ciclo-estacionário sendo processado no ambiente do LabVIEW. O algoritmo completo se encontra no [Anexo B](#).

Figura 25: Diagrama de blocos: *Script* do método ciclo-estacionário.



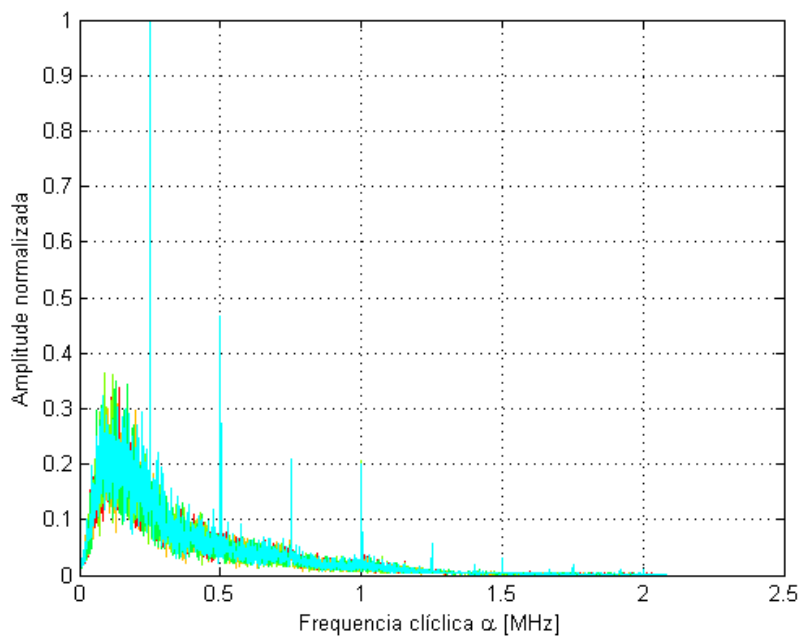
Fonte: Elaborada pelo autor.

3.2.4 Classificação do Sinal

O processo de classificação do sinal é feito com base nas diferentes frequências cíclicas que os padrões IEEE 802.11g e IEEE 802.15.4 apresentam. O padrão IEEE 802.11g apresenta duas frequências cíclicas fundamentais distintas, em 250kHz e 1MHz, sendo que esta última ocorre devido ao prefixo no início do pacote de transmissão. Estes valores estão em acordo com os resultados encontrados por (Figueiredo e Silva, et al., 2014).

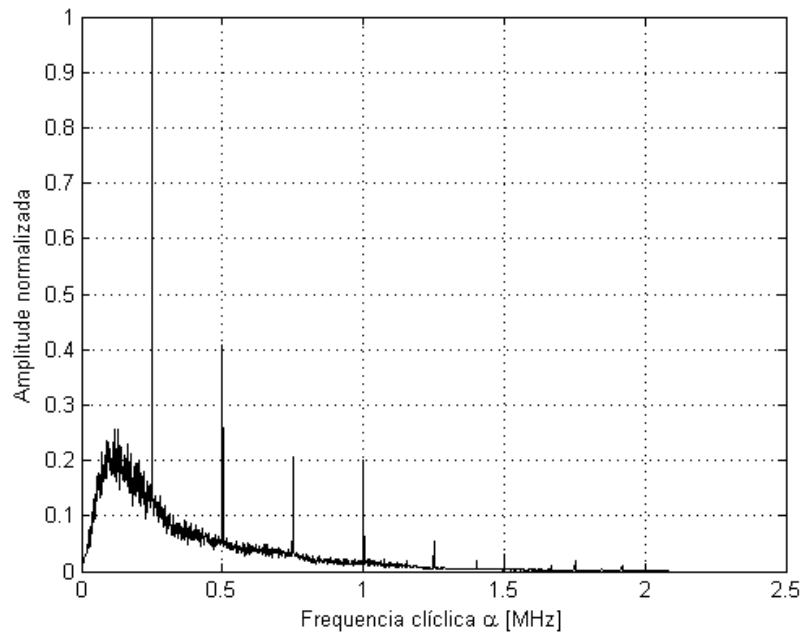
Na [FIGURA 26](#) é representada a função de autocorrelação cíclica para diversas janelas de análise sobrepostas, para o padrão IEEE 802.11g, enquanto que na [FIGURA 27](#) é demonstrado o efeito de suavização obtido pela média do resultado de diversas janelas de análise. Pelas figuras é evidente a frequência fundamental situada em 250kHz.

Figura 26: Função de autocorrelação cíclica para sinal com padrão IEEE 802.11g.



Fonte: Elaborada pelo autor.

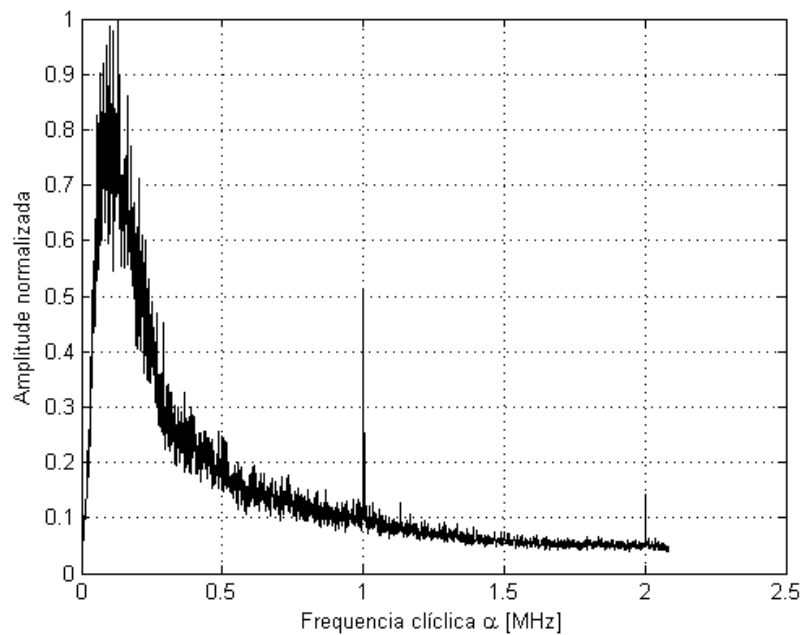
Figura 27: Função de autocorrelação cíclica suavizada pela média das janelas de análise, para sinal com padrão IEEE 802.11g.



Fonte: Elaborada pelo autor.

A [FIGURA 28](#) mostra a ocorrência da segunda frequência cíclica do padrão IEEE 802.11g em 1MHz, devido ao prefixo do pacote de dados.

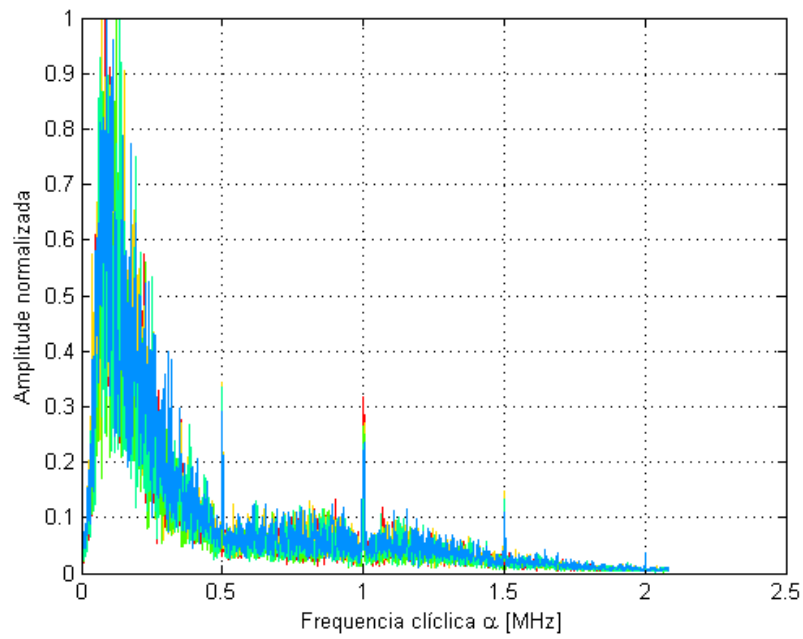
Figura 28: Função de autocorrelação cíclica do prefixo do pacote de um sinal do padrão IEEE 802.11g.



Fonte: Elaborada pelo autor.

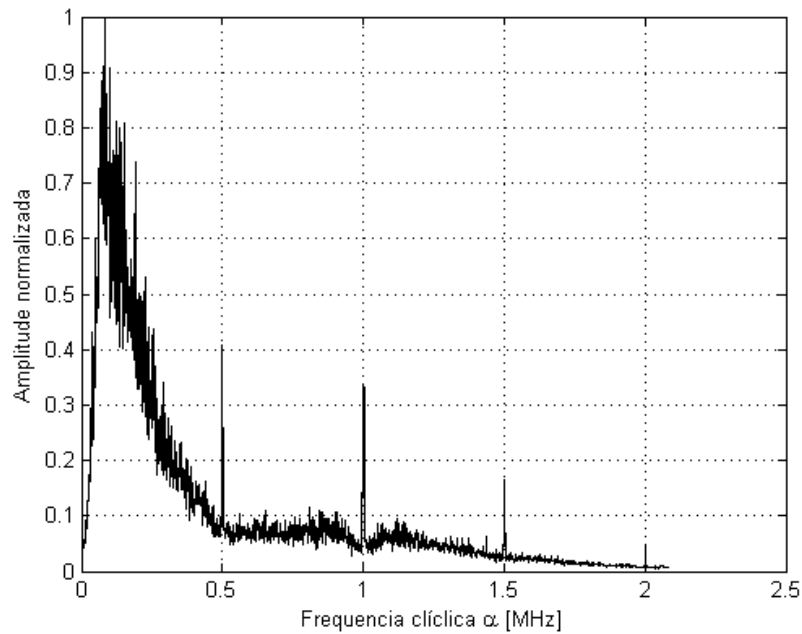
Para o padrão IEEE 802.15.4, a frequência cíclica fundamental ocorre em 500kHz, devido a taxa de transmissão de 250kbps. A [FIGURA 29](#) e a [FIGURA 30](#) ilustram o resultado obtido.

Figura 29: Função de autocorrelação cíclica para sinal com padrão IEEE 802.15.4.



Fonte: Elaborada pelo autor.

Figura 30: Função de autocorrelação cíclica suavizada pela média das janelas de análise, para sinal com padrão IEEE 802.15.4.

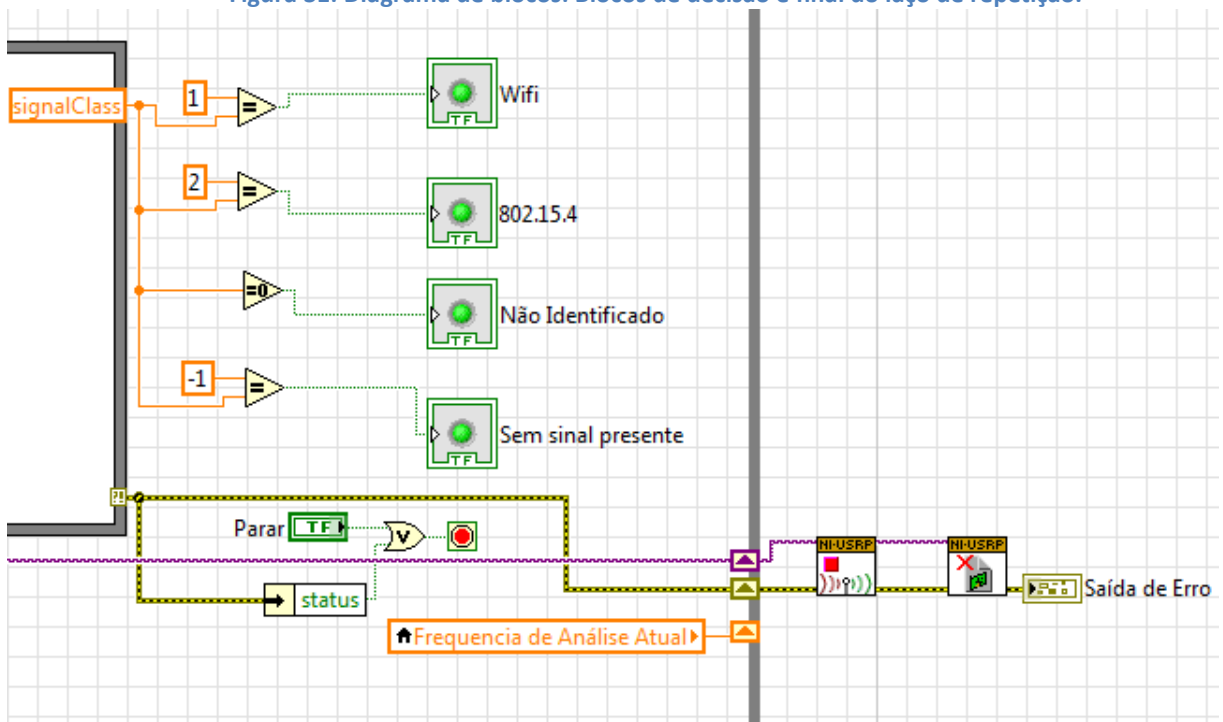


Fonte: Elaborada pelo autor.

Para determinar se uma frequência está presente, é calculado o valor do pico em relação à média local. Caso o pico desta frequência for maior que 50% a média local, pode-se considerar que esta frequência está presente.

A [FIGURA 31](#) mostra a variável *signalClass* que é obtida ao final do *script* do método ciclo-estacionário e que possui o resultado da classificação do sinal. A variável é utilizada para mostrar no painel frontal do LabVIEW o padrão de sinal encontrado no canal analisado.

Figura 31: Diagrama de blocos: Blocos de decisão e final do laço de repetição.

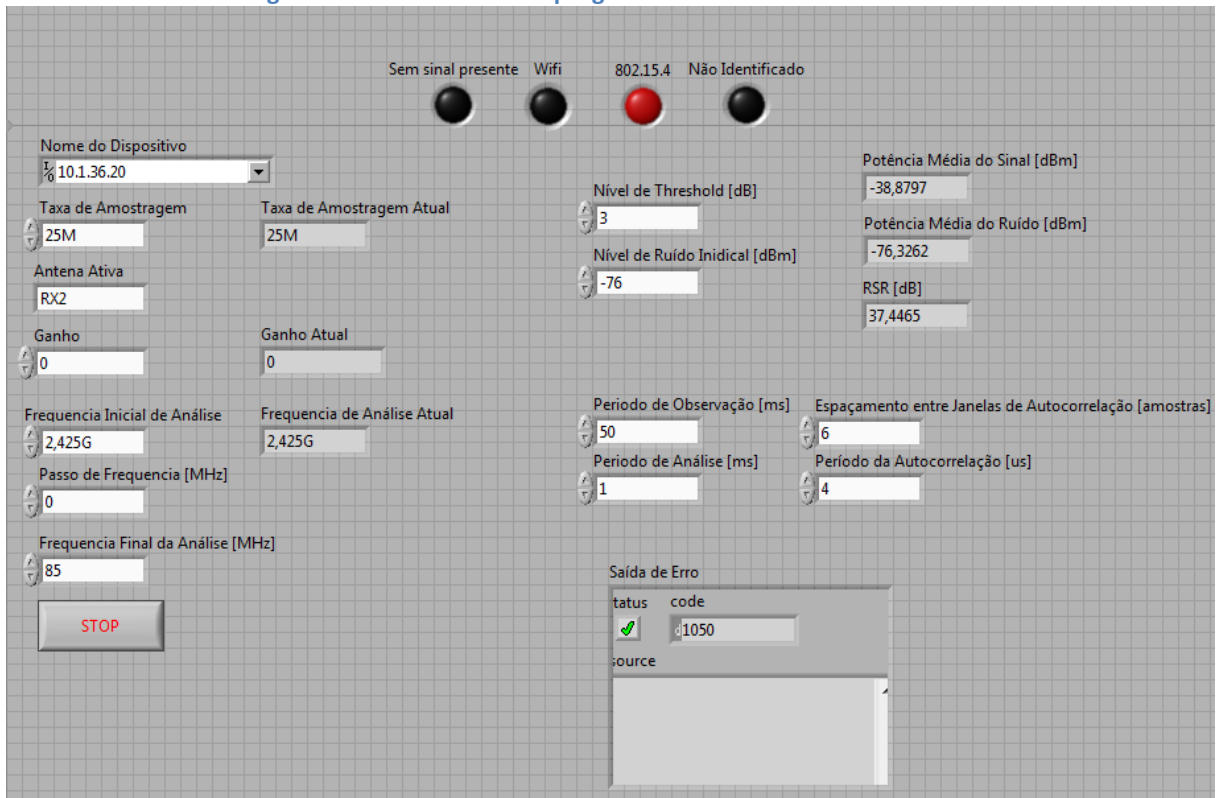


Fonte: Elaborada pelo autor.

3.2.5 Interface Homem-Máquina

Nesta etapa é apresentado ao usuário os resultados obtidos pelo programa. A interface homem-máquina foi desenvolvida no ambiente do LabVIEW, que apresenta os gráficos das amostras recebidas, assim como a função de autocorrelação cíclica do sinal encontrado. Também é indicado no painel frontal qual o padrão de sinal encontrado de acordo com o resultado da classificação. A [FIGURA 32](#) mostra o painel frontal do programa.

Figura 32: Painel frontal do programa desenvolvido em LabVIEW.



Fonte: Elaborada pelo autor.

No painel frontal estão presentes os resultados obtidos para o canal observado, assim como todos os parâmetros utilizados para configurar os *scripts*. Após analisar determinado canal, a aplicação segue para realizar o processamento do canal seguinte.

4 Validação e Testes

Nesta seção são apresentados os métodos utilizados para realizar a validação dos algoritmos de detecção de energia e o ciclo-estacionário. A validação é realizada através de simulação computacional, utilizando o software MATLAB para gerar os sinais que serão analisados. Também são retratados os testes realizados em laboratório utilizando sinais reais obtidos através do dispositivo USRP. Os sinais analisados são os da classe 802.11g utilizando um ponto de acesso WiFi, e sinais da classe 802.15.4 utilizando um transceptor genérico.

4.1 Validação do método de detecção de energia

Para realizar esta validação foi gerado no MATLAB um vetor pseudo-aleatório do tipo complexo para simular um ruído branco, utilizando a função *wgn*. É adicionado a esse ruído uma janela de um sinal pseudo-aleatório com diferentes níveis de potência média, com o objetivo de identificar a presença de erros por falsa detecção. Para o teste foram consideradas as seguintes condições, descritas na [TABELA 1](#):

Tabela 1: Variáveis definidas para o primeiro teste

Variável	Valor	Unidade
Potência média do Ruído	-100	[dBm]
Potência média do Sinal	-96,8 ~ -97,2	[dBm]
Impedância de referência	50	[Ω]
RSR	2,8 ~ 3,2	[dB]
Diferença de RSR entre ensaios (passo entre ensaios)	0,025	[dB]
Número de ensaios para cada passo	300	[ensaios]

Fonte: Elaborada pelo autor.

Os valores de entrada para o algoritmo de detecção de energia encontram-se na

[TABELA 2](#):

Tabela 2: Variáveis de entrada para o algoritmo de detecção de energia

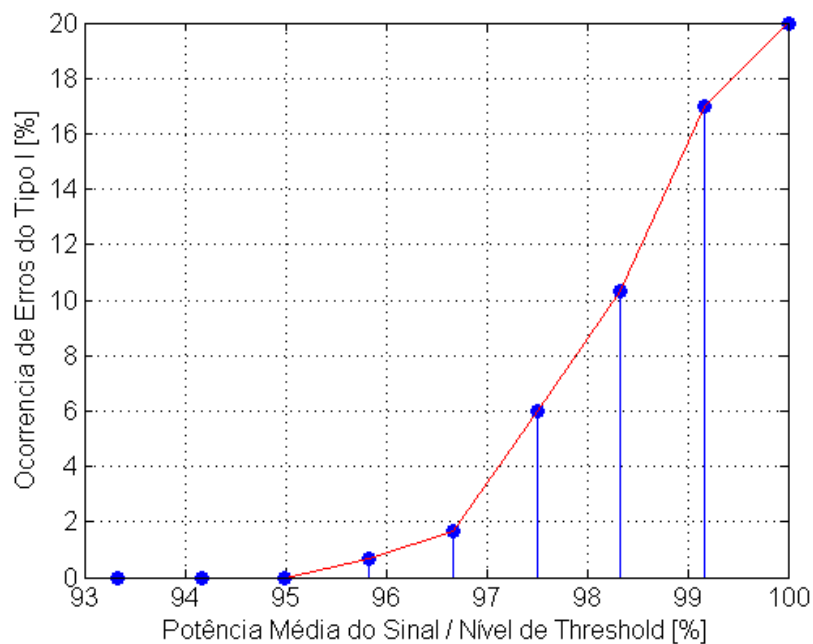
Variável	Valor	Unidade
Tempo total de aquisição	10	[ms]
Tempo da janela de sinal	2,5	[ms]
Tempo da janela de análise	~0,353	[ms]
Valor inicial de estimativa de ruído (<i>initNoiseDB</i>)	-104	[dBm]
Nível de <i>threshold</i> (<i>thresholdDB</i>)	3	[dB]

Fonte: Elaborada pelo autor.

Neste teste foi considerado o nível de amplitude mínima para diferenciar um sinal do ruído (variável *thresholdDB*) igual a $3dB$. Este valor define o nível de significância do teste de hipótese, ou seja, caso um sinal interferente tenha uma potência média acima deste valor e o algoritmo não detectá-lo, considera-se que ocorreu um erro do tipo II (aceitar a hipótese nula H_0 quando esta é falsa). De forma simétrica, no caso em que há presente um sinal com potência média abaixo deste nível e o algoritmo detectar a presença do sinal, é dito que ocorreu um erro do tipo I (rejeitar a hipótese nula H_0 quando esta é verdadeira), conforme visto no [SUBCAPÍTULO 2.4.1](#).

A [FIGURA 33](#) mostra os resultados obtidos quando a potência média do sinal é menor que o nível de *threshold*, e a quantidade de ocorrências de falso-positivos (erros do tipo I).

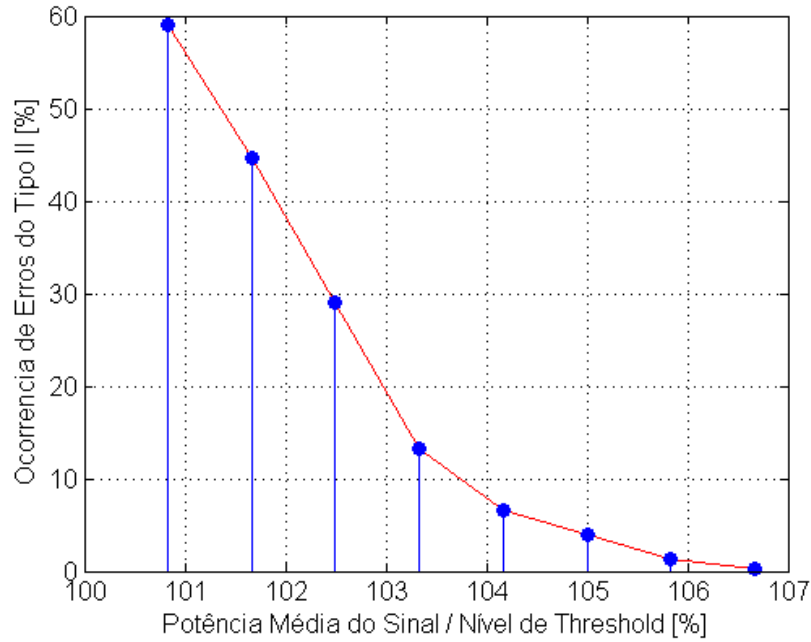
Figura 33: Primeiro teste, verificação de erros do tipo I.



Fonte: Elaborada pelo autor.

Através da [FIGURA 33](#) pode-se afirmar com base no teste estatístico realizado, que para sinais com potência média até 96% do nível de *threshold* a ocorrência de erros do tipo I é menor do que 2%. De mesma forma são apresentados na [FIGURA 34](#) os resultados quando a potência do sinal é maior que o nível de *threshold* e a ocorrência de falso-negativos.

Figura 34: Primeiro teste, verificação de erros do tipo II.



Fonte: Elaborada pelo autor.

Pela [FIGURA 34](#) pode-se afirmar pelo teste realizado que para sinais com potência média 5% acima do nível de *threshold*, a ocorrência de erros do tipo II é menor que 5%.

Foi realizado um segundo teste com condições mais restritas, especificado pela [TABELA 3](#) e [TABELA 4](#):

Tabela 3: Variáveis definidas para o segundo teste

Variável	Valor	Unidade
Potência média do Ruído	-100	[dBm]
Potência média do Sinal	-98,8 ~ -99,2	[dBm]
Impedância de referência	50	[Ω]
RSR	0,8 ~ 1,2	[dB]
Diferença de RSR entre ensaios (passo entre ensaios)	0,025	[dB]
Número de ensaios para cada passo	300	[ensaios]

Fonte: Elaborada pelo autor.

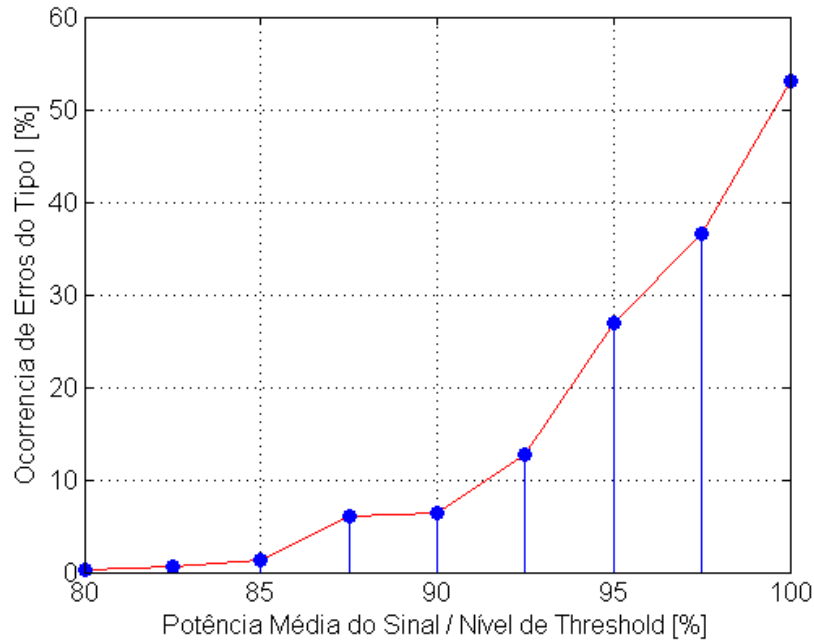
Tabela 4: Variáveis de entrada para o algoritmo de detecção de energia

Variável	Valor	Unidade
Tempo total de aquisição	10	[ms]
Tempo da janela de sinal	1,25	[ms]
Tempo da janela de análise	~0,353	[ms]
Valor inicial de estimativa de ruído (<i>initNoiseDB</i>)	-104	[dBm]
Nível de <i>threshold</i> (<i>thresholdDB</i>)	3	[dB]

Fonte: Elaborada pelo autor.

Neste teste a janela do sinal possui um período menor que a anterior, com duração de 1,25ms. O valor de *threshold* também foi reduzido para 1dB. Novamente foram realizados os ensaios com o objetivo de identificar a ocorrência de erros dos tipos I e II. A primeira verificação de erros é dada na [FIGURA 35](#).

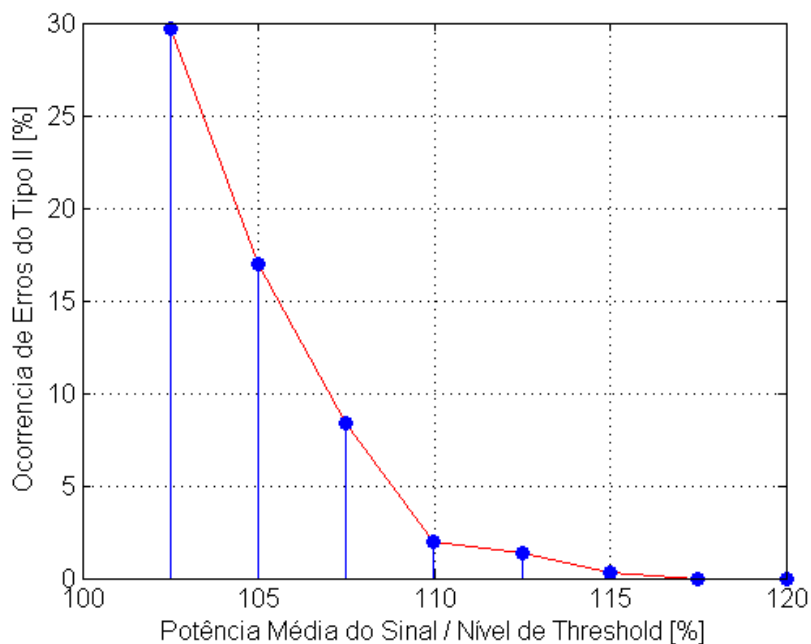
Figura 35: Segundo teste, verificação de erros do tipo I.



Fonte: Elaborada pelo autor.

Neste teste observa-se uma ocorrência de erros do tipo I menor que 10% para um sinal com potência média 10% inferior ao nível de threshold. Em sequência, a [FIGURA 36](#) representa a ocorrência de erros do tipo II.

Figura 36: Segundo teste, verificação de erros do tipo II.



Fonte: Elaborada pelo autor.

no qual é possível afirmar uma ocorrência de erros do tipo II menor que 5% para uma potência média do sinal 10% maior que o nível de *threshold*.

4.2 Validação do método ciclo-estacionário

Para a validação foi gerado no MATLAB um sinal pseudo-aleatório modulado do tipo BPSK (Binary Phase Shift Keying). O sinal possui as características descritas na [TABELA 5](#):

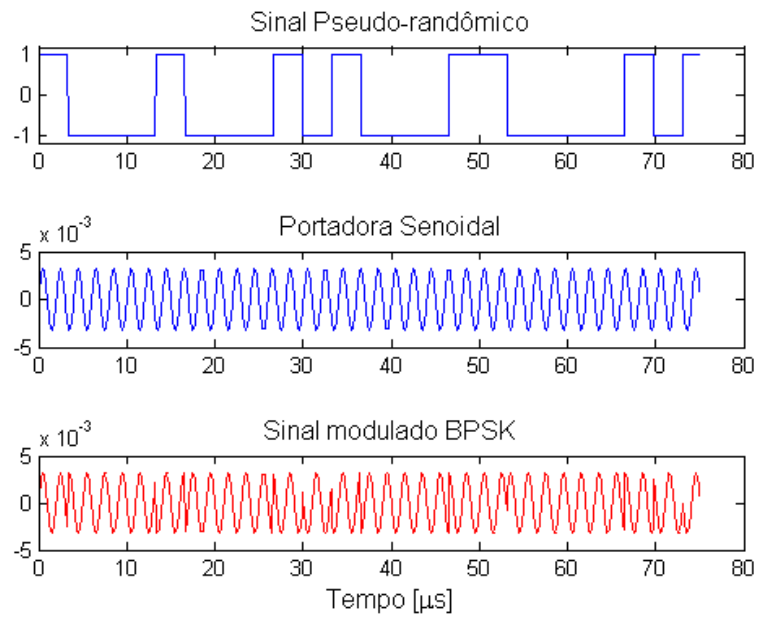
Tabela 5: Características do sinal BPSK gerado

Variável	Valor	Unidade
Frequência da Portadora	500	[kHz]
Frequência do sinal pseudo-aleatório	300	[kHz]
Frequência de amostragem	25	[MHz]
Período do símbolo	3,33	[μ s]
Tempo total do sinal	10	[ms]

Fonte: Elaborada pelo autor.

A [FIGURA 37](#) mostra um trecho do sinal BPSK resultante gerado a partir do sinal pseudo-aleatório e da portadora senoidal.

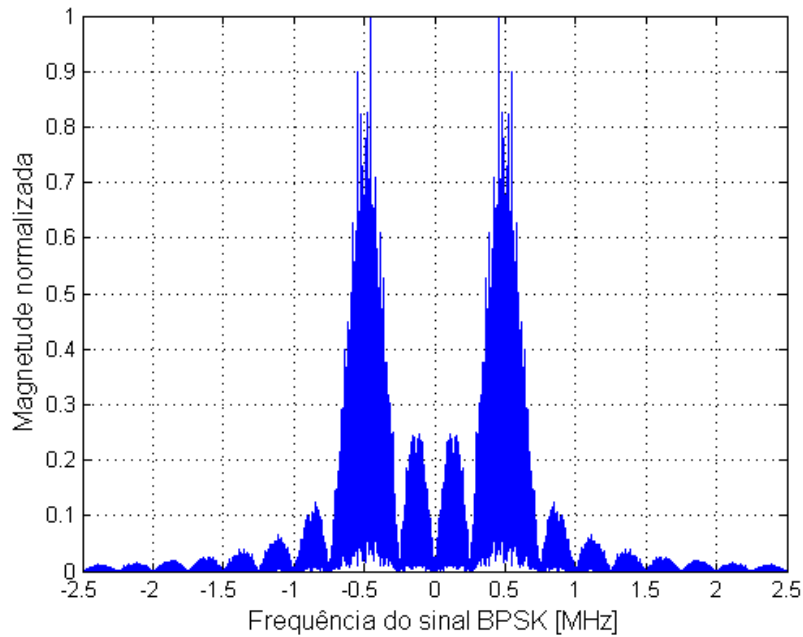
Figura 37: Trecho do sinal gerado BPSK.



Fonte: Elaborada pelo autor.

Pode-se ver na [FIGURA 38](#) a FFT do sinal resultante, onde é possível verificar a existência dos picos principais centrados na frequência da portadora senoidal em 500kHz.

Figura 38: Espectro de frequências do sinal modulado BPSK.



Fonte: Elaborada pelo autor.

Foram gerados três ensaios utilizando diferentes parâmetros para a validação do algoritmo ciclo-estacionário. Em cada um dos ensaios é alterado progressivamente o tamanho da janela de observação, o que resulta em um incremento na resolução da frequência cíclica. Os parâmetros utilizados encontram-se na [TABELA 6](#).

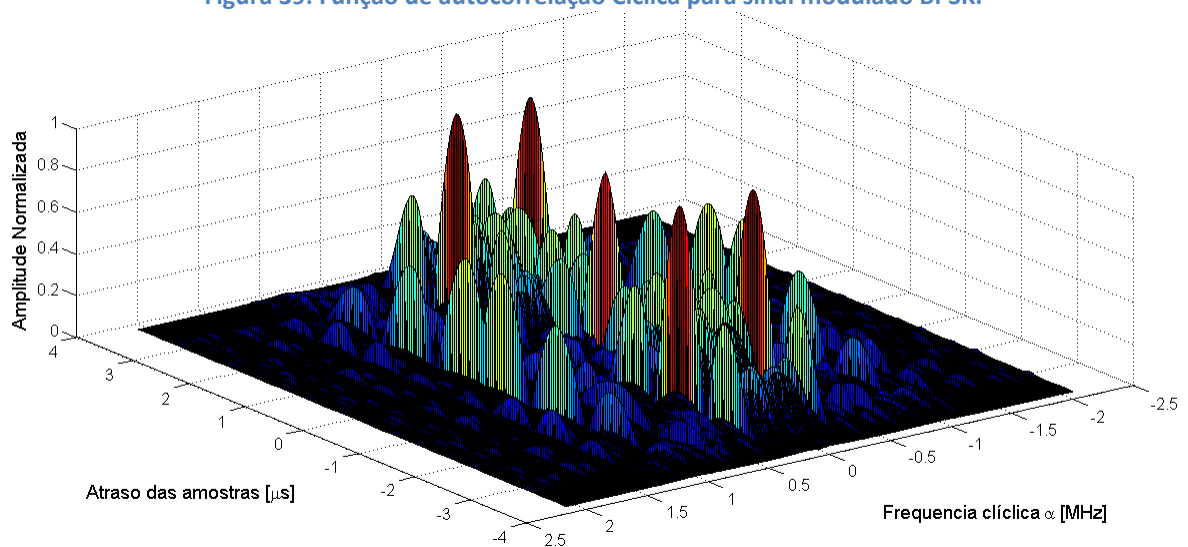
Tabela 6: Parâmetros utilizados para validação do algoritmo ciclo-estacionário

Variável	Valores	Unidade
Período da janela de análise	0,5 1,0 2,5	[ms]
Período da janela de autocorrelação	3,84	[μ s]
Espaçamento entre janelas de autocorrelação (Δn)	6	[amostras]
Número de janelas de autocorrelação (<i>corrSize</i>)	2.075 4.145 10.401	[janelas]
Resolução da frequência cíclica ($\Delta\alpha$)	2,0 1,0 0,4	[kHz]

Fonte: Elaborada pelo autor.

A partir do algoritmo foi gerada a função de autocorrelação cíclica, que pode ser vista na [FIGURA 39](#).

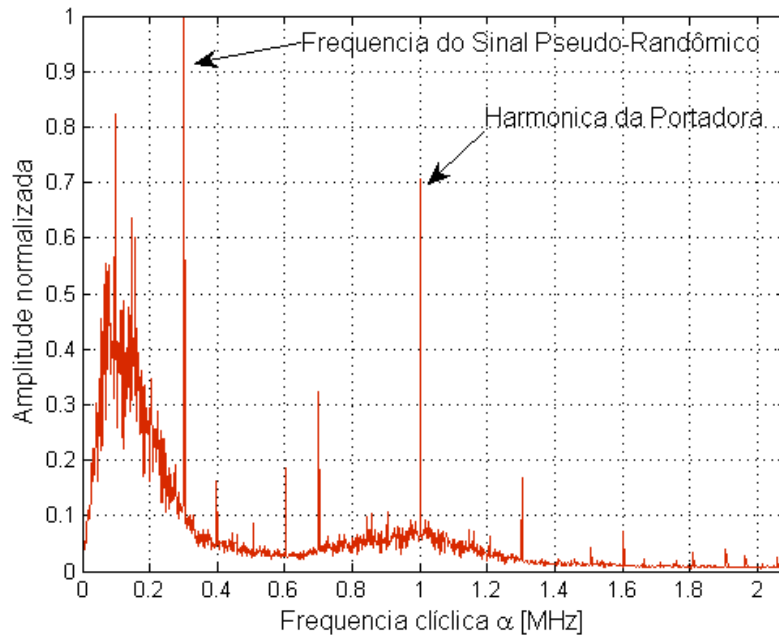
Figura 39: Função de autocorrelação Cíclica para sinal modulado BPSK.



Fonte: Elaborada pelo autor.

É possível verificar pela [FIGURA 39](#) os picos de amplitudes localizados em aproximadamente 300kHz , referentes à frequência do sinal pseudo-aleatório gerado. É possível identificar ainda picos secundários localizados na frequência cíclica de 1MHz , que corresponde à duas vezes o valor da portadora senoidal, conforme pode ser referenciado por um experimento similar realizado por ([Zhou, et al., 2012](#)). A partir da função de autocorrelação cíclica, pode-se obter o valor médio das amostras no eixo correspondente ao atraso entre amostras τ . Com isso é possível analisar em um único vetor os picos na qual as portadoras estão presentes. A [FIGURA 40](#) mostra o resultado deste processo utilizando o eixo positivo da frequência cíclica e uma janela de análise de $0,5\text{ms}$.

Figura 40: Média da função de autocorrelação cíclica para janela de 0,5ms.

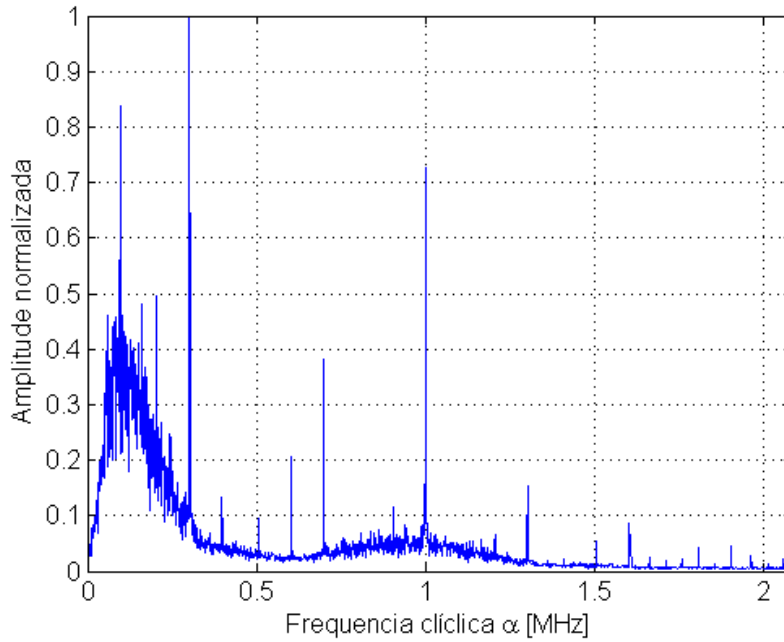


Fonte: Elaborada pelo autor.

Na [FIGURA 40](#) é possível verificar de forma mais nítida as portadoras presentes no sinal modulado. O pico principal corresponde a frequência do sinal pseudo-aleatório em 300kHz e o pico secundário em 1MHz corresponde a primeira harmônica da portadora senoidal. É possível observar ainda algumas harmônicas secundárias que são geradas a partir da composição destes picos principais.

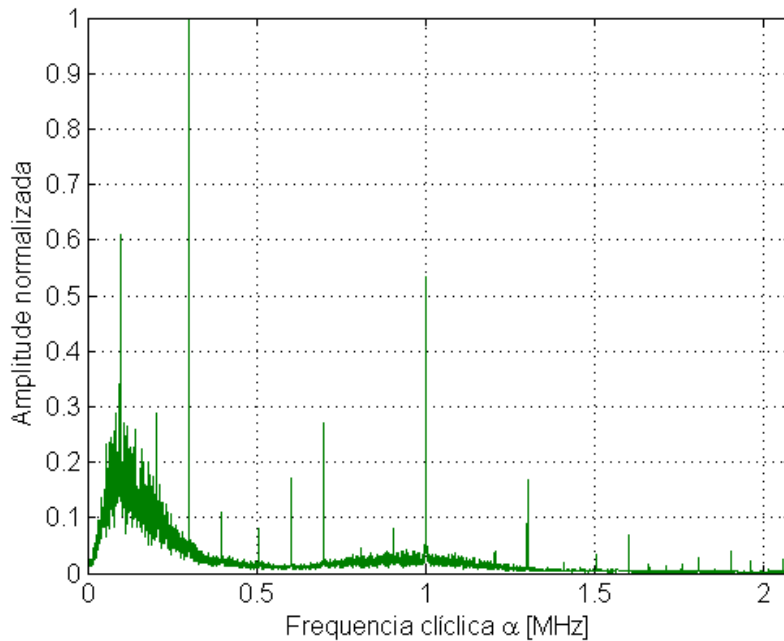
Os resultados obtidos para janelas de análise de $1,0\text{ms}$ e $2,5\text{ms}$ estão representados na [FIGURA 41](#) e [FIGURA 42](#).

Figura 41: Média da função de autocorrelação cíclica para janela de 1,0ms.



Fonte: Elaborada pelo autor.

Figura 42: Média da função de autocorrelação cíclica para janela de 2,5ms.



Fonte: Elaborada pelo autor.

A partir das figuras geradas é possível verificar um incremento na resolução da frequência cíclica, conforme esperado pela [EQUAÇÃO 17](#).

4.3 Testes da aplicação desenvolvida

Foram realizados testes em laboratório com o objetivo de identificar o grau de confiabilidade da aplicação desenvolvida, a partir das taxas de acerto e correta classificação das duas classes de sinais analisados, em função da RSR entre o sinal recebido e o ruído de fundo do dispositivo USRP. Com este teste é possível verificar qual o nível mínimo de *threshold* que deve ser imposto para obter uma taxa de confiabilidade adequada. Foram realizados dois ensaios distintos para cada sinal analisado, conforme segue nas subseções seguintes.

4.3.1 Testes com sinal IEEE 802.15.4

O teste é realizado utilizando a placa de desenvolvimento em conformidade com o padrão IEEE 802.15.4, descrita no [SUBCAPÍTULO 3.1](#). O dispositivo é configurado através de uma porta USB para a transmissão de pacotes genéricos, mantendo fixa a potência de saída e o canal de transmissão. A potência do sinal transmitido é controlada através de um atenuador variável de RF. Este é então conectado a entrada do dispositivo USRP. A [FIGURA 43](#) representa o diagrama de blocos para este teste, e a [FIGURA 44](#) mostra as conexões utilizadas.

Figura 43: Diagrama de blocos para teste do sinal IEEE802.15.4.



Fonte: Elaborada pelo autor.

Figura 44: Equipamentos utilizados para o teste do sinal IEEE802.15.4.

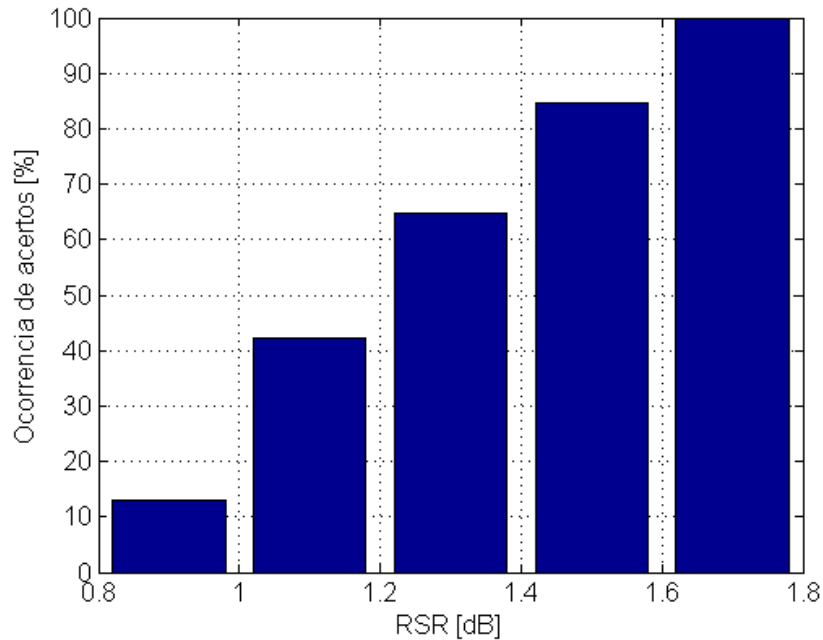


Fonte: Elaborada pelo autor.

Para o ensaio foram realizadas 525 análises, onde o resultado consiste na RSR entre o sinal recebido e o ruído de fundo do dispositivo USRP, e o resultado da classificação do sinal pelo algoritmo ciclo-estacionário. Como o ambiente é fechado e o sinal é conhecido, é

possível identificar a taxa de erros e acertos do algoritmo em função da RSR do sinal. A [FIGURA 45](#) representa os resultados obtidos para este ensaio.

Figura 45: Taxa de acertos em função da RSR, ensaio com sinal IEEE 802.15.4



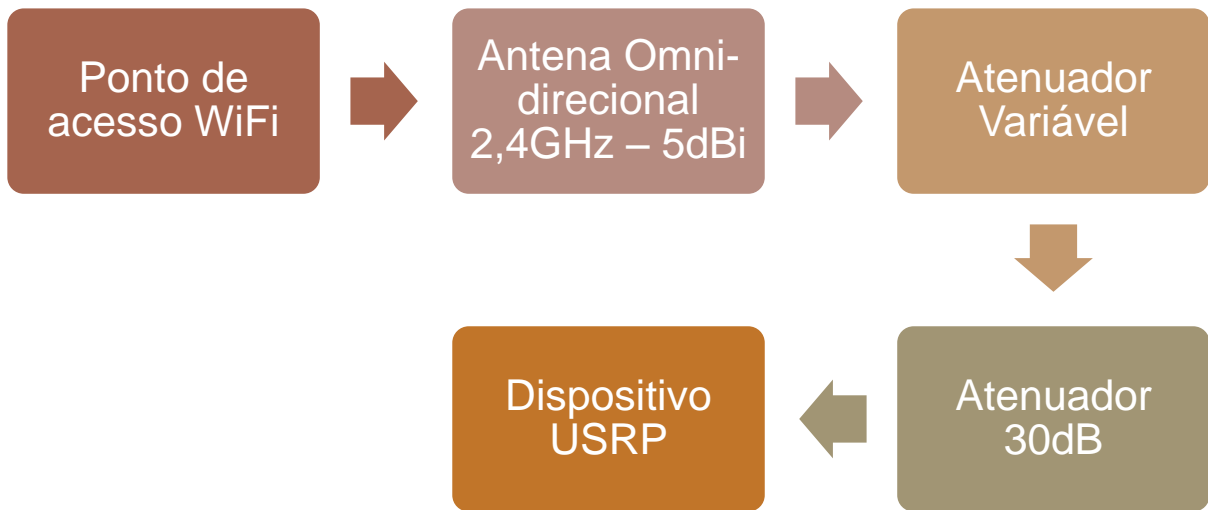
Fonte: Elaborada pelo autor.

Pela [FIGURA 45](#) é possível afirmar que a aplicação desenvolvida tem uma taxa de acertos superior a 80% para a classificação de sinais IEEE 802.15.4, com uma RSR de até 1,4dB em relação ao ruído de fundo do dispositivo USRP.

4.3.2 Testes com sinal IEEE 802.11g

O segundo ensaio de classificação do sinal com padrão 802.11g foi realizado utilizando um ponto de acesso em espaço aberto, controlando a potência recebida pela antena utilizando um atenuador variável. A [FIGURA 46](#) mostra o diagrama de blocos, e a [FIGURA 47](#) representa a conexão entre os equipamentos utilizados para o ensaio.

Figura 46: Diagrama de blocos para teste do sinal IEEE802.11g.



Fonte: Elaborada pelo autor.

Figura 47: Equipamentos utilizados para o teste do sinal IEEE 802.11g.

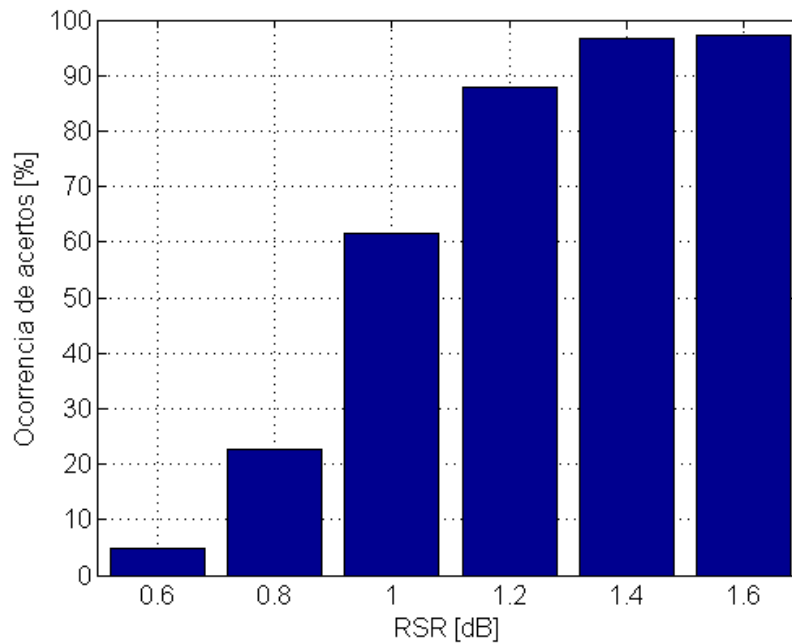


Fonte: Elaborada pelo autor.

Para este ensaio foram realizadas 288 análises para diferentes níveis de RSR, e também observando a taxa de acertos para a detecção de sinais com padrão IEEE 802.11g

oriundos de um ponto de acesso WiFi localizado a uma distância de aproximadamente 2 metros. Os resultados obtidos estão representados na [FIGURA 48](#).

Figura 48: Taxa de acertos em função da RSR, ensaio com sinal IEEE 802.11g.



Fonte: Elaborada pelo autor.

Pela [FIGURA 48](#) pode-se afirmar que para uma RSR maior que 1,3dB a taxa de detecção correta de sinais com padrão IEEE802.11g é maior que 90%.

5 Conclusão e Trabalhos Futuros

Este trabalho teve objetivo como desenvolver em ambiente de LabVIEW e MATLAB uma aplicação capaz de identificar sinais com padrões IEEE 802.11g e IEEE 802.15.4 na faixa ISM de 2,4GHz utilizando um rádio definido por software. A aplicação obteve êxito em realizar a classificação correta dos dois padrões analisados com taxas de acertos relativamente altas: acima de 90% para sinais com RSR próximas a 1,6dB.

A importância das duas técnicas de sensoriamento de espectro implementadas, detecção de energia e ciclo-estacionário foram discutidas no [CAPÍTULO 2](#), que faz uma abordagem sobre a situação atual das redes de dispositivos sem fio operando nas faixas livres ISM, e discute brevemente sobre o conceito de Rádio Cognitivo que possui uma grande expectativa de desenvolvimento e popularização.

Como trabalhos futuros há diversos aspectos que podem ser explorados, entre eles a implementação de algoritmos computacionalmente mais eficientes para realizar o método ciclo-estacionário, como o algoritmo FAM (*FFT Accumulation Method*, do inglês); a utilização do software de código livre GNU Rádio como interface de configuração do dispositivo USRP; e explorar as características cíclicas de outros padrões de sinais encontrados em diferentes dispositivos sem fio.

Referências Bibliográficas

- Abdulsattar, Mahmood A. e Hussein, Zahin A. 2012.** *Energy Detection Technique for Spectrum Sensing in Cognitive Radio: A Survey*. Baghdad : Department of Electrical Engineering, University of Baghdad, 2012. *International Journal of Computer Networks & Communications (IJCNC)* Vol.4, No.5.
- Adams, Jon T. 2005.** *An Introduction to IEEE STD 802.15.4*. Tempe, Arizona, USA : Freescale Semiconductor, Inc., 2005.
- Agilent Technologies. 2002.** *IEEE 802.11 Wireless LAN PHY Layer (RF) Operation and Measurement*. s.l. : Agilent Technologies, 2002. Application Note 1380-2.
- Almeida, E. P. L. 2010.** *Desenvolvimento de Técnicas de Sensoriamento do Espectro Embasada em Detecção de Energia para Aplicações em Sistemas Rádio-Cognitivos*. Brasília : Departamento de Engenharia Elétrica, Universidade de Brasília, 2010.
- ANATEL. 2015.** *Plano de Atribuição, Destinação e Distribuição de Frequências no Brasil*. Brasil : s.n., 2015.
- Castro, Marcos E. 2011.** *Cyclostationary Detection for OFDM in Cognitive Radio Systems*. University of Nebraska, Lincoln : s.n., 2011.
- Figueiredo e Silva, Pedro e Ondrej, Daniel. 2014.** *Cyclostationary features of downsampled 802.11g OFDM signal for cognitive positioning systems*. Dept. of Electron. & Commun. Eng., Tampere Univ. of Technol., Tampere, Finland : s.n., 2014.
- IEEE 802.11, Wikipédia. 2015.** IEEE 802.11. *Wikipédia*. [Online] 05 de Maio de 2015. http://en.wikipedia.org/wiki/IEEE_802.11.
- Jain, Saurav e Raghunadh, M. V. 2014.** *Spectrum Sensing Based on Cyclostationary Detector using USRP*. Department of Electronics and Communication Engineering; National Institute of Technology, Warangal, INDIA : s.n., 2014.
- Machado, T., et al. 2014.** Dept. of Electr. Eng., Fed. Univ. of Rio Grande do Sul, Porto Alegre, Brazil. *IEEE Xplore*. [Online] 30 de Julho de 2014. [Citado em: 05 de Maio de 2015.] <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6945598>.

National Instruments. 2009. Addressing the Test Challenges of MIMO Communications Systems. *National Instruments*. [Online] 17 de Abril de 2009. [Citado em: 11 de Junho de 2015.] <http://www.ni.com/white-paper/5689/en/>.

—. **2015.** Five Factors to Consider When Implementing a Wireless Sensor Network (WSN). *National Instruments*. [Online] 11 de Junho de 2015. [Citado em: 27 de Junho de 2012.] <http://www.ni.com/white-paper/10789/en/>.

Vassis, Dimitris, et al. 2005. *The IEEE 802.11g standard for high data rate WLANs*. University of the Aegean, Greece : s.n., 2005.

Winter, J. M., et al. 2014. Dept. of Electr. Eng., Fed. Univ. of Rio Grande do Sul, Porto Alegre, Brazil. *IEEE Xplore*. [Online] 30 de Julho de 2014. [Citado em: 05 de Maio de 2015.] <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6945541>.

Zhou, Ruolin, et al. 2012. *Real-time Cyclostationary Analysis for Cognitive Radio via Software Defined Radio*. Department of Electrical and Computer Engineering, Western New England University; Department of Electrical Engineering, Wright State University; Inst. of Applied Marine Physics and Undersea Tech., Nat. Sun Yet-Sen Univ.; Acoustics Division, Naval Resear : s.n., 2012.

ANEXO A: Algoritmo de Detecção de Energia em linguagem de MATLAB

```
%% Limpa variáveis anteriores
clearvars -except analysisWindowPeriod observationWindowPeriod initNoiseDB
thresholdDB dataIn samplingFrequency

%% Calibração
dataIn = dataIn/4.8;

%% Coeficientes do Filtro
coefficientsFIR = fir1(1024,[0.01 0.4],'bandpass');

%% Variáveis Locais
impedance = 50;

%% Variáveis calculadas
analysisWindowLength = ceil(sqrt(analysisWindowPeriod*1e-
3*samplingFrequency));
analysisWindowSize = ceil(samplingFrequency*observationWindowPeriod*1e-
3/(analysisWindowLength^2));

%% Fitragem / Reorganização dos dados de entrada
dataInFiltered = filter(coefficientsFIR,1,dataIn);
dataInFiltered = dataInFiltered(analysisWindowLength*analysisWindowLength +
1:end);
main =
reshape(dataInFiltered,analysisWindowLength*analysisWindowLength,analysisWi
ndowSize);

%% Conversão de unidades de potência / Alocação de memória
dataInFilteredDBm = pow2db(abs(dataInFiltered).^2/(impedance*1e-3));
mainAbs = abs(main).^2/impedance;
mainAbsDBm = pow2db(mainAbs/1e-3);
noiseThreshold = db2pow(initNoiseDB)*1e-3;
threshold = db2pow(thresholdDB);
signal = zeros(1,analysisWindowSize);
noise = zeros(1,analysisWindowSize);
noiseThresholdVector = zeros(1,analysisWindowSize);

%% Separação do sinal/ruído
for i = 1:analysisWindowSize
    if mean(mainAbs(:,i)) > noiseThreshold*threshold
        signal(i) = mean(mainAbs(:,i));
    else
        noise(i) = mean(mainAbs(:,i));
    end

    if ~isempty(noise(noise~=0))
        noiseThreshold = mean(noise(noise~=0))*0.1+noiseThreshold*0.9;
    end
    noiseThresholdVector(i) = noiseThreshold;
end

%% Variáveis de saída / Decisão do canal
noiseThresholdVectorDBm = pow2db(noiseThresholdVector/1e-3);
noiseThresholdOut = pow2db(noiseThreshold/1e-3);
```

```

if ~isempty(find(signal,1))
    [signalSorted,signalIndex] = sort(signal,'descend');
    signalSorted(signalSorted==0) = [];
    signalIndex = signalIndex(1:length(signalSorted));
    signalOut = pow2db(mean(signalSorted)/1e-3);
    RSR = signalOut - noiseThresholdOut;
    channelClear = 0;
else
    signalOut = [];
    signalIndex = 0;
    RSR = -1;
    channelClear = 1;
end

%% Gráfico
figure(1)
windowsGrid = -250*ones(1,length(dataInFiltered));
for i = 1:analysisWindowSize
    windowsGrid(1,analysisWindowLength^2*i) = 250;
end

observationWindowVector =
linspace(0,length(dataInFiltered)/samplingFrequency,length(dataInFiltered))
;
plot(observationWindowVector*1e3,windowsGrid,'r'); hold on;
plot(observationWindowVector*1e3,dataInFilteredDBm,'b');

if signalIndex ~= 0
    graphColors = hsv(length(signalIndex));
    for i = 1:length(signalIndex)
        plot((analysisWindowLength^2*signalIndex(i)-
(analysisWindowLength^2-
1))/samplingFrequency:1/samplingFrequency:(analysisWindowLength^2*(signalIn
dex(i)))/samplingFrequency)*1e3,mainAbsDBm(:,signalIndex(i))','color',graph
Colors(i,:));
    end
end

plot((analysisWindowLength^2/2/samplingFrequency:analysisWindowLength^2/sam
plingFrequency:analysisWindowLength^2*analysisWindowSize/samplingFrequency)
*1e3,noiseThresholdVectorDBm,'r','linewidth',2);
plot((analysisWindowLength^2/2/samplingFrequency:analysisWindowLength^2/sam
plingFrequency:analysisWindowLength^2*analysisWindowSize/samplingFrequency)
*1e3,noiseThresholdVectorDBm+thresholdDB,'g','linewidth',2);
ylim([-90 -10]);
xlabel('Tempo de Aquisição [ms]','fontsize',12);
ylabel('Potência média [dBm]','fontsize',12);
hold off;

```

ANEXO B: Algoritmo ciclo-estacionário em linguagem de MATLAB

```
%% Variáveis calculadas
corrLength = ceil(correlationWindowPeriod*1e-6*samplingFrequency);
corrSize = round((length(main)-corrLength)/correlationWindowSpacing)-1;

cyclicSampling = samplingFrequency/correlationWindowSpacing;
cyclicVector = linspace(0,cyclicSampling/2,corrSize/2);
cyclicResolution = cyclicSampling/corrSize;

graphColors = hsv(length(signalIndex));
if length(signalIndex) > 3
    signalIndex = signalIndex(1:3);
end

coefficientsFIRHP = fir1(128,0.02,'high');
%% Inicio do algoritmo

if signalIndex ~= 0
    clear cyclicWindowMean
    cyclicWindowMean = zeros(length(signalIndex),corrSize/2);
    for j = 1:length(signalIndex)
        %% Processamento das janelas com sinal

        clear mainAbs;
        mainReal = real(main(:,signalIndex(j)));
        mainRealFiltered = filter(coefficientsFIRHP,1,mainReal);
        %% Subdivisão da janela

        clear mainWindowed;
        mainWindowed = zeros(corrSize,corrLength);
        for i = 1:corrSize
            mainWindowed(i,:) =
mainRealFiltered(correlationWindowSpacing*(i-
1)+1:correlationWindowSpacing*(i-1)+corrLength);
            waitbarHandle = waitbar(i/corrSize);
        end
        close(waitbarHandle);
        %% Autocorrelação das subdivisões

        clear aux;
        clear correlationWindow;
        aux = zeros(corrSize,2*corrLength-1);
        correlationWindow = zeros(corrSize,corrLength);
        for i = 1:corrSize
            aux(i,:) = fftshift(xcorr(mainWindowed(i,:)));
            correlationWindow(i,:) = aux(i,1:corrLength);
            waitbarHandle = waitbar(i/corrSize);
        end
        close(waitbarHandle);

        %% Função de Autocorrelação Ciclica

        clear aux;
        clear cyclicWindow;
        aux = zeros(corrSize,corrLength);
        cyclicWindow = zeros(corrSize/2,corrLength);
        for i = 1:corrLength
```

```

        aux(:,i) = filter(coefficientsFIRHP,1,correlationWindow(:,i));
        aux(:,i) = abs(fft(aux(:,i)));
        cyclicWindow(:,i) = aux(1:corrSize/2,i);
    end
    cyclicWindowMean(j,:) = mean(cyclicWindow,2);

    figure(2)
    f1 =
    plot(cyclicVector/1e6,cyclicWindowMean(j,:)/max(cyclicWindowMean(j,:)));
    set(f1,'Color',graphColors(j,:));
    grid on;
    xlabel('Frequencia clíclica \alpha [MHz]');
    ylabel('Amplitude normalizada');
    hold on;
end
clear cyclicWindowSmooth
cyclicWindowSmooth = mean(cyclicWindowMean,1);
figure(3)
plot(cyclicVector/1e6,cyclicWindowSmooth/max(cyclicWindowSmooth),'k');
grid on;
xlabel('Frequencia clíclica \alpha [MHz]');
ylabel('Amplitude normalizada');

%% Decisão

cyclicHarmonic = 250e3; % 250kHz
cyclicHarmonicIndex = (cyclicHarmonic)/(cyclicResolution);

cyclicPeaks = zeros(1,4);
for i = 1:4
    cyclicPeaks(i) =
cyclicWindowSmooth((round(cyclicHarmonicIndex*i)+1))/mean([cyclicWindowSmooth
th((round(cyclicHarmonicIndex*i)+1)-3:(round(cyclicHarmonicIndex*i)+1)-1)
cyclicWindowSmooth((round(cyclicHarmonicIndex*i)+1)+1:(round(cyclicHarmonic
Index*i)+1)+3)]);
    end
    cyclicPeaks(3) = [];

    if cyclicPeaks(1)>1.5 || (cyclicPeaks(3)>1.5 && cyclicPeaks(2)<1.5)
        signalClass = 1;
    elseif cyclicPeaks(2)>1.5 && cyclicPeaks(1)<1.5
        signalClass = 2;
    else
        signalClass = 0; %Desconhecido
    end
end

else
    signalClass = -1;
end
end

```