

**UNIVERSIDADE PRESBITERIANA MACKENZIE
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA E COMPUTAÇÃO**

THIAGO MONTANARO SAPIA

**DESENVOLVIMENTO DE ALGORITMOS DE
SINCRONISMO E ESTIMAÇÃO DE CANAL NO GNU
RADIO COMPANION PARA O SISTEMA ISDB-T**

São Paulo
2017

**UNIVERSIDADE PRESBITERIANA MACKENZIE
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA E COMPUTAÇÃO**

Thiago Montanaro Sapia

**Desenvolvimento de algoritmos de sincronismo e estimação de
canal no GNU Radio Companion para o sistema ISDB-T**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Computação da Universidade Presbiteriana Mackenzie, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Professor Dr. Cristiano Akamine

São Paulo
2017

S241d Sapia, Thiago Montanaro
Desenvolvimento de algoritmos de sincronismo e
estimação de canal no GNU radio companion para
o sistema ISDB-T. / Thiago Montanaro Sapia - 2017.
62f.: il., 30 cm

Dissertação (Mestrado em Engenharia Elétrica e
Computação) – Universidade Presbiteriana Mackenzie,
São Paulo, 2017.

Orientadores: Prof. Dr. Cristiano Akamine

Bibliografia: f. 49-51

1. ISDB-T. 2. SDR. 3. Rádio definido por software.
4. GNU rádio. Título.

CDD 621.38807

THIAGO MONTANARO SAPIA

DESENVOLVIMENTO DE ALGORITMOS DE SINCRONISMO E ESTIMAÇÃO
DE CANAL NO GNU RADIO COMPANION PARA O SISTEMA ISDB-T

Dissertação de Mestrado apresentada ao
Programa de Pós-graduação em Engenharia
Elétrica e Computação da Universidade
Presbiteriana Mackenzie, como requisito
parcial para obtenção do título de Mestre em
Engenharia Elétrica.

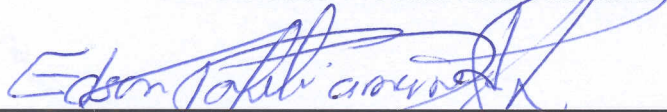
Aprovado em 20 de fevereiro de 2017.

BANCA EXAMINADORA



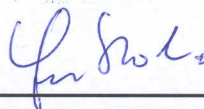
Prof. Dr. Cristiano Akamine

Universidade Presbiteriana Mackenzie



Prof. Dr. Edson Tafeli Carneiro dos Santos

Universidade Presbiteriana Mackenzie



Prof. Dr. Guido Stolfi

Universidade de São Paulo

AGRADECIMENTOS

Meus pais Percio e Valéria, e à minha irmã Isabella, por todo carinho e apoio. Sem vocês eu não teria chegado até aqui.

Laboratório de TV Digital da Universidade Presbiteriana Mackenzie pela concessão da minha bolsa de estudos.

Universidade Presbiteriana Mackenzie por ter permitido que realizasse meus estudos e experiências em suas dependências.

Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq)

Laboratório de TV Digital por ter me cedido espaço e equipamentos, sem os quais seria impossível realizar o trabalho aqui descrito.

Programa de Pós-Graduação em Engenharia Elétrica e Computação da Universidade Presbiteriana Mackenzie por ter me dado a oportunidade de estudar e a todos os professores que compartilharam seus conhecimentos comigo.

Professor Doutor Cristiano Akamine pela paciência, dedicação, conselhos e sugestões triviais para o desenvolvimento desse trabalho.

Todos os colegas que fazem, ou fizeram, parte do Laboratório de Televisão Digital da Universidade Presbiteriana Mackenzie e que me ajudaram na realização deste trabalho.

RESUMO

Este trabalho apresenta uma descrição sobre o Sistema Brasileiro de TV Digital (SBTVD) e *Software Defined Radio* (SDR). O objetivo deste trabalho é implementar um estimador de canal e estágios de sincronismo para receber sinais ISDB-T_B via RF. Com isso, sendo possível a análise das constelações dos sinais recebidos. Para tal, foram utilizados conceitos de SDR e as linguagens de programação C++ e *Python*. Por meio da ferramenta aberta GNU *Radio*, foram realizadas implementações de sincronismo no tempo, frequência (inteira e fracionária) e de quadro OFDM. Além disso, um estimador de canal foi implementado. Simulações computacionais foram realizadas para verificar o funcionamento das implementações.

Palavras-chave: *ISDB-T, SDR, rádio definido por software, GNU Radio.*

ABSTRACT

This essay presents theoretical concepts of the Brazilian Digital TV System (SBTVD) and Software Defined Radio (SDR). The purpose of this essay is to implement timing synchronization stages and a channel estimator to enable ISDB-T_B reception via RF. Thus, enabling the analysis of the constellations of the received signals. To achieve this goal, it was used SDR concepts and the C++ and Python programming languages. Through the use of the GNU Radio tool, time, frequency and OFDM frame synchronization were implemented. Besides that, a channel estimator was also implemented. Computer simulations were carried out to verify the performance of the implementations.

Key words: *ISDB-T, SDR , software defined radio, GNU Radio.*

Lista de Figuras

| | | |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1 | Padrões de TV digital pelo mundo. | 2 |
| 2 | Acesso à internet no Brasil. | 3 |
| 3 | Transmissão do ISDB-T _B | 7 |
| 4 | Codificação de canal do ISDB-T _B | 8 |
| 5 | Espectro com BST-OFDM. | 10 |
| 6 | Entrelaçador de frequência. | 12 |
| 7 | <i>Zero padding</i> | 13 |
| 8 | Inserção do intervalo de guarda. | 14 |
| 9 | Estrutura de um SDR. | 15 |
| 10 | Placa USRP B210. | 16 |
| 11 | Demodulação em quadratura (<i>chip</i> AD9361). | 17 |
| 12 | Exemplo de fluxograma criado no GNU <i>Radio</i> | 19 |
| 13 | Espectro do canal 60 (TV Mackenzie/SP). | 20 |
| 14 | Diagrama de sincronismo. | 21 |
| 15 | Autocorrelação para sincronismo temporal. | 22 |
| 16 | <i>Flow graph</i> para o teste de remoção de zeros e sincronismo temporal | 29 |
| 17 | Constelação do sinal após FFT. | 30 |
| 18 | Zeros no domínio do tempo. | 30 |
| 19 | Valores de autocorrelações realizadas. | 31 |
| 20 | Constelação do sinal ISDB-T sem sincronismo temporal. | 32 |
| 21 | Constelação do sinal ISDB-T após sincronismo temporal. | 32 |
| 22 | <i>Flow graph</i> para os testes de detecção de erros fracionários | 33 |
| 23 | Valores de MER em função de erro fracionário com variação senoidal de 1 Hz: (a)Erro fracionário gerado e detectado; (b) MER para sistema com e sem realimentação | 34 |
| 24 | Valores de MER em função de erro fracionário com variação de onda quadrada de 1 Hz: (a)Erro fracionário gerado e detectado; (b) MER para sistema com e sem realimentação | 34 |
| 25 | Valores de MER em função de erro fracionário com variação de onda dente de serra de 0.25 Hz: (a)Erro fracionário gerado e detectado; (b) MER para sistema com e sem realimentação | 35 |
| 26 | Valores de MER em função de erro fracionário com variação senoidal de 2 Hz: (a)Erro fracionário gerado e detectado; (b) MER para sistema com e sem realimentação | 36 |

| | | |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 27 | Valores de MER em função de erro fracionário com variação senoidal de 5 Hz: (a) Erro fracionário gerado e detectado; (b) MER para sistema com e sem realimentação | 36 |
| 28 | Valores de MER em função de erro fracionário com variação senoidal de 10 Hz: (a) Erro fracionário gerado e detectado; (b) MER para sistema com e sem realimentação | 37 |
| 29 | Valores de MER em função de erro fracionário constante (Sem realimentação) | 38 |
| 30 | Valores de MER em função de erro fracionário constante (Com realimentação direta) | 38 |
| 31 | Valores de MER em função de erro fracionário constante (Com realimentação controlada por PID) | 39 |
| 32 | <i>Flow graph</i> para os testes de detecção de erros inteiros de frequência. . . . | 40 |
| 33 | Valores correlação cruzada (sem erro inteiro de frequência). | 40 |
| 34 | Valores da correlação cruzada (com erro inteiro de frequência). | 41 |
| 35 | Valor do erro de frequência (1200 Hz) | 41 |
| 36 | Constelação de um sinal ISDB-T com erro inteiro de frequência. | 42 |
| 37 | <i>Bits</i> demodulados da TMCC | 42 |
| 38 | <i>Flow graph</i> para os testes de estimação de canal. | 43 |
| 39 | Relação E/D para diferentes estimadores de canal. | 44 |
| 40 | Curva de MER em função do SNR. | 45 |
| 41 | Relação E/D com todos os blocos interligados sem ruído branco e com ruído branco. | 46 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|-----------------------------|---------------------------------------------------------------------------------|
| <i>16 – QAM</i> | <i>16 Quadrature Amplitude Modulation</i> |
| <i>64 – QAM</i> | <i>64 Quadrature Amplitude Modulation</i> |
| <i>AVC</i> | <i>Advanced Television System Committee</i> |
| <i>AVC</i> | <i>Advanced Video Coding</i> |
| <i>AGC</i> | <i>Automatic Gain Control</i> |
| <i>AC</i> | <i>Auxiliary Channel</i> |
| <i>BPSK</i> | <i>Binary Phase Shift Keying</i> |
| <i>BST – OFDM</i> | <i>Band Segmented Transmission - Orthogonal Frequency Division Multiplexing</i> |
| <i>BTS</i> | <i>Broadcast Transport Stream</i> |
| <i>DBPSK</i> | <i>Diferential Binary Phase Shift Keying</i> |
| <i>DQPSK</i> | <i>Diferential Quadrature Phase Shift Keying</i> |
| <i>DTMB</i> | <i>Digital Terrestrial Multimedia Broadcast</i> |
| <i>DVB – T</i> | <i>Digital Video Broadcasting - Terrestrial</i> |
| <i>FFT</i> | <i>Fast Fourier Transform</i> |
| <i>FPGA</i> | <i>Field Programmable Gate Array</i> |
| <i>FIR</i> | <i>Finite Impulse Response</i> |
| <i>FEC</i> | <i>Forward Error Correction</i> |
| <i>FI</i> | <i>Frequência Intermediária</i> |
| <i>GRC</i> | <i>GNU Radio Companion</i> |
| <i>HDTV</i> | <i>High Definition Television</i> |
| <i>HE – AAC</i> | <i>High Efficiency Advanced Audio Coding</i> |
| <i>ISDB – T</i> | <i>Integrated Services Digital Broadcasting Terrestrial</i> |
| <i>ISDB – T_B</i> | <i>Integrated Services Digital Broadcasting Terrestrial - Version B</i> |
| <i>ICI</i> | <i>Inter-Carrier Interference</i> |
| <i>ISI</i> | <i>Intersymbol Interference</i> |
| <i>IG</i> | <i>Intervalo de Guarda</i> |
| <i>LTE</i> | <i>Long Term Evolution</i> |

| | |
|--------------|------------------------------------------------------------|
| <i>MFN</i> | <i>Multi-Frequency Network</i> |
| <i>QPSK</i> | <i>Quadrature Phase Shifting Key</i> |
| <i>RF</i> | <i>Rádio Frequência</i> |
| <i>RS</i> | <i>Reed Solomon</i> |
| <i>SP</i> | <i>Scattered Pilots</i> |
| <i>SBTVD</i> | <i>Sistema Brasileiro de TV Digital</i> |
| <i>SDR</i> | <i>Software Defined Radio</i> |
| <i>SDTV</i> | <i>Standard Definition Television</i> |
| <i>TI</i> | <i>Time Interleaver</i> |
| <i>TMCC</i> | <i>Transmission and Multiplexing Configuration Control</i> |
| <i>TS</i> | <i>Transport Stream</i> |
| <i>TSP</i> | <i>Transport Stream Packet</i> |
| <i>USB</i> | <i>Universal Serial Bus</i> |
| <i>USRP</i> | <i>Universal Software Radio Peripheral</i> |

Sumário

| | | |
|----------|------------------------------------------------------|-----------|
| 1 | INTRODUÇÃO | 1 |
| 1.1 | Objetivo | 3 |
| 1.2 | Justificativa | 3 |
| 1.3 | Metodologia | 4 |
| 1.4 | Estrutura e organização do trabalho | 5 |
| 2 | SISTEMA BRASILEIRO DE TELEVISÃO DIGITAL | 6 |
| 2.1 | Transmissão do Sistema ISDB-T _B | 7 |
| 2.2 | Codificação de Canal | 7 |
| 2.2.1 | Codificador Externo | 8 |
| 2.2.2 | Separador de camadas | 8 |
| 2.2.3 | Dispersor de energia | 9 |
| 2.2.4 | Ajuste de Atraso | 9 |
| 2.2.5 | Entrelaçador de <i>byte</i> | 9 |
| 2.2.6 | Codificador Convolutacional | 9 |
| 2.2.7 | Entrelaçador de <i>bit</i> | 10 |
| 2.2.8 | Mapeamento | 10 |
| 2.2.9 | Combinador de camadas | 11 |
| 2.2.10 | Entrelaçador de tempo | 11 |
| 2.2.11 | Entrelaçador de frequência | 11 |
| 2.2.12 | Estrutura de quadro OFDM e IFFT | 12 |
| 2.2.13 | Intervalo de guarda | 13 |
| 3 | DESENVOLVIMENTO DO PROJETO | 15 |
| 3.1 | <i>Software Defined Radio (SDR)</i> | 15 |
| 3.2 | GNU Radio | 17 |
| 3.3 | Demodulador ISDB-T no GRC | 20 |
| 3.3.1 | Desenvolvimento dos blocos | 21 |
| 3.3.2 | Sincronismo temporal | 22 |
| 3.3.3 | Sincronismo de frequência | 24 |
| 3.3.4 | Sincronismo de quadro OFDM | 25 |
| 3.3.5 | Estimação de canal assistida por pilotos | 27 |

| | |
|-----------------------------------|-----------|
| 4 Resultados | 29 |
| 5 CONCLUSÃO | 47 |
| 5.1 Artigos publicados | 48 |
| 5.2 Trabalhos futuros | 48 |
| REFERÊNCIAS BIBLIOGRÁFICAS | 49 |

1 INTRODUÇÃO

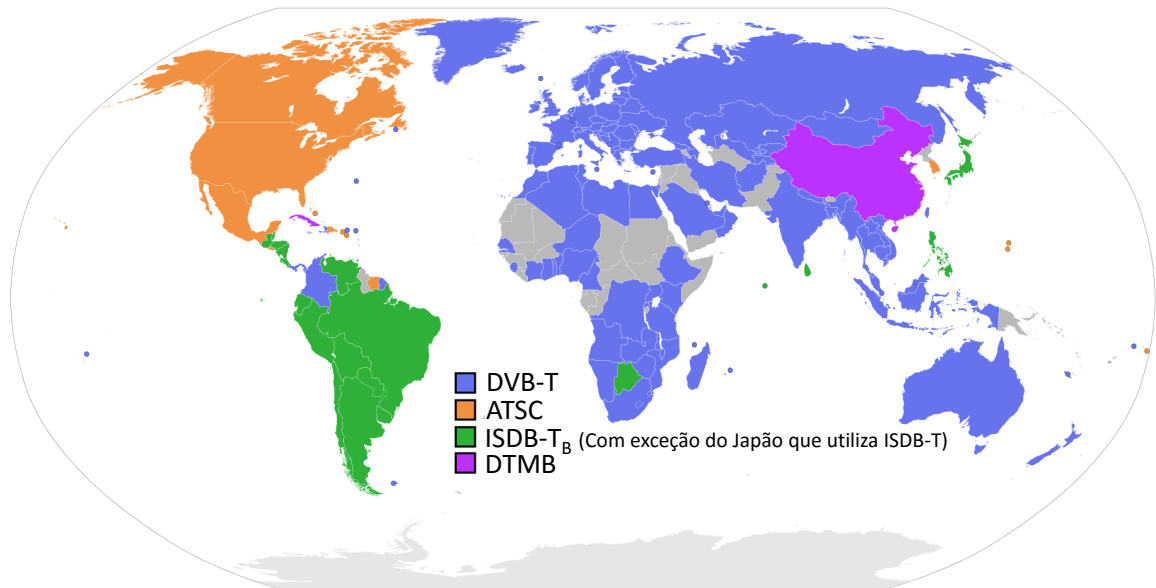
Em 31 de outubro de 1963, no Brasil, os serviços de radiodifusão foram regulamentados. Esses serviços são gratuitos para a população e englobam as transmissões de áudio (rádio) e de áudio e vídeo (televisão). A finalidade desses serviços é transmitir conteúdos educativos e culturais, sendo permitido a exploração comercial dos mesmos (BRASIL, 1963).

Um dos serviços de radiodifusão mais utilizados no dia-a-dia pelos brasileiros é a televisão. Com o advento da TV digital os conteúdos passaram a ser transmitidos com alta qualidade de áudio e vídeo, assim melhorando a experiência do telespectador.

O padrão de TV digital utilizado no Brasil é baseado no *Integrated Services Digital Broadcasting Terrestrial* (ISDB-T), sistema de radiodifusão terrestre criado pelo Japão e inaugurado em 2003. No Brasil foram feitas algumas modificações nesse sistema, assim criando a versão chamada *Integrated Services Digital Broadcasting Terrestrial - Version B* (ISDB-T_B) (AKAMINE, 2011).

O ISDB-T_B foi, e está sendo adotado por diversos países. Na Figura 1 pode ser observada a distribuição de países e seus respectivos padrões adotados. Além do ISDB-T/ISDB-T_B, também existem os padrões *Digital Video Broadcasting Terrestrial* (DVB-T)/DVB-T2, *Advanced Television System Committee* (ATSC) e *Digital Terrestrial Multimedia Broadcast* (DTMB).

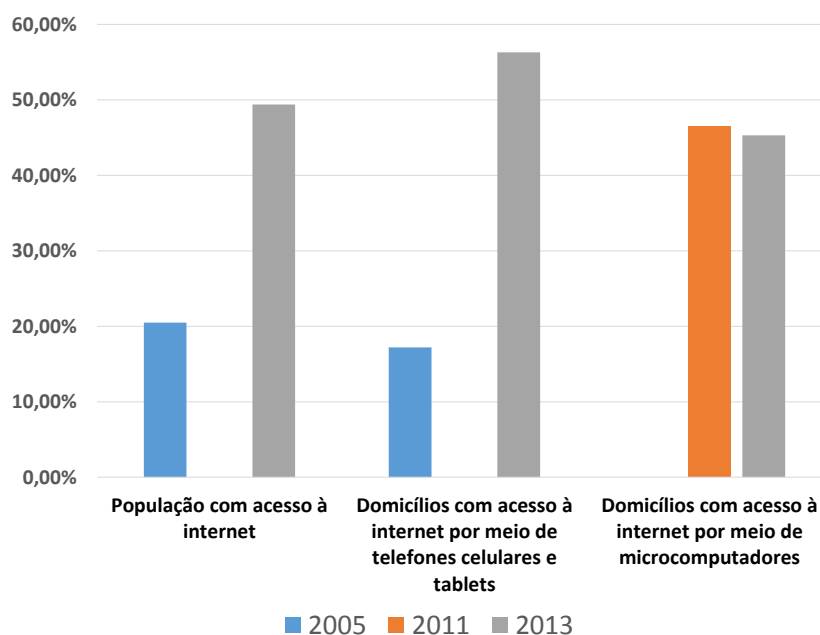
Figura 1 – Padrões de TV digital pelo mundo.



Fonte: Adaptado de DIBEG (2015).

Além da radiodifusão, outras tecnologias também são muito utilizadas pelos brasileiros, como a internet e os telefones celulares. Entretanto, a demanda por maior velocidade e qualidade desses serviços cresce de maneira acelerada. Segundo o Instituto Brasileiro de Geografia e Estatística, IBGE (2015), em 2013, estimou-se que 49,4% da população tinha acesso à internet, enquanto em 2005 esse valor era de 20,9%. Além disso, os domicílios com acesso à internet por meio de telefones celulares e *tablets* era de 53,6% e 17,2%. Em 2011, domicílios com acesso à internet por meio de microcomputadores era de 46,5%. Entretanto, em 2013, houve uma retração, para 45,3%. Isso mostra que outros dispositivos também estão sendo utilizados para o acesso à internet. Esses dados podem ser melhor observados na Figura 2.

Figura 2 – Acesso à internet no Brasil.



Fonte: Adaptado IBGE (2015).

Portanto, é necessário espaço no espectro para que essas tecnologias possam transmitir seus sinais. Autoridades reguladoras do espectro estão reivindicando faixas de frequência do espectro para que sejam leiloadas, assim satisfazendo a demanda das novas tecnologias. Com o rápido crescimento dessas tecnologias e a demanda de espectro, a televisão digital vem perdendo seu espaço. Assim, sendo importante estudos sobre maneiras de utilizar o espectro de forma mais eficiente.

1.1 Objetivo

O objetivo deste projeto é implementar estágios de sincronismo e um estimador de canal modelado e simulado no ambiente de desenvolvimento GNU *Radio Companion* (GRC), que pode utilizado em conjunto com uma placa *Universal Software Radio Peripheral* (USRP). O propósito é demodular um sinal de RF ISDB-T e analisar sua constelação.

1.2 Justificativa

A implementação de um demodulador ISDB-T utilizando algoritmos desenvolvidos a partir da revisão da literatura, juntamente com otimizações, capacitam o desenvolvi-

mento tecnológico do sistema de TV digital brasileiro. Com a adição desses algoritmos é possível realizar a demodulação em tempo real ou de sinais digitalizados das emissoras de TV digital. Além disso, muitos trabalhos não levam em consideração os algoritmos de sincronismo, assim realizando simulações assumindo um sincronismo ideal.

O Laboratório de TV Digital da Universidade Presbiteriana Mackenzie está desenvolvendo um conjunto de bibliotecas para os padrões de TV digital, portanto os algoritmos desenvolvidos servirão como base para futuras pesquisas. A literatura disponível não explora amplamente o sistema brasileiro de TV digital, principalmente porque a inovação tecnológica presente nos *chips* de demodulação são considerados um segredo industrial. Portanto, os algoritmos de sincronismo e estimação de canal apresentados nesse trabalho servirão como base para futuras pesquisas do Laboratório de TV Digital e em desenvolvimentos de um novo padrão de TV digital brasileiro. Também servirão como base para implementações de sistemas de demodulação de TV digital mais complexos, permitindo que a tecnologia de transmissão e recepção permaneçam no Brasil.

Outra grande relevância é que implementações por meio do *software* GRC mostram a flexibilidade e eficácia do SDR. Com a criação de blocos com códigos individuais é possível o desenvolvimento de sistemas de rádio sem a necessidade de criação de novos *hardwares* (HILBURN, 2015).

1.3 Metodologia

O primeiro ponto tratado na pesquisa teórica foi o ISDB-T_B. Para desenvolver esse assunto foram utilizados artigos, especificações técnicas e normas. Além disso, foi realizada a pesquisa teórica acerca de SDR e do *software* GRC. Em relação a implementação, a pesquisa foi baseada, principalmente, em artigos e fóruns especializados.

No desenvolvimento prático deste trabalho foram implementadas as etapas de estimação de canal, sincronismo de quadro *Orthogonal Frequency Division Multiplex* (OFDM), sincronismo temporal e sincronismo de frequência. Para tal, foi utilizado o GRC. Com um computador pessoal, blocos com códigos individuais foram programados e simulações foram realizadas nesse *software* para avaliar os blocos implementados.

Como *hardware*, foi utilizada a placa USRP B210 conectada a um computador pessoal

via *Universal Serial Bus* (USB). Por meio dessa placa foram gravados alguns sinais de RF para servirem de fonte de sinal no GRC. O computador é responsável pelo processamento da lógica programada no *software*, e a placa, pela recepção e conversão analógico/digital do sinal recebido em RF.

1.4 Estrutura e organização do trabalho

Esta dissertação é composta de 4 partes, como descritas a seguir.

- Capítulo 1: Introdução, objetivo, justificativa, metodologia e organização do texto.
- Capítulo 2: Trata o SBTVD e algumas de suas características quanto a codificação de canal, transmissão e recepção.
- Capítulo 3: Apresenta o SDR de uma forma geral e as principais características da plataforma GNU *Radio*, utilizada no desenvolvimento deste projeto. Também aborda como foi realizado o desenvolvimento do projeto, bem como as técnicas de sincronismo e estimação de canal necessárias para o funcionamento de um receptor.
- Capítulo 4: São apresentados os testes realizados e resultados obtidos
- A última parte do texto apresenta as conclusões finais e trabalhos futuros.

2 SISTEMA BRASILEIRO DE TELEVISÃO DIGITAL

O sistema ISDB-T foi desenvolvido para oferecer alta qualidade de vídeo e áudio. Além disso, proporciona flexibilidade nos parâmetros de modulação e possibilita a combinação de diversos serviços e programas em um único canal com largura de banda de 6, 7 ou 8 MHz. O sistema opera com serviços *High Definition Television* (HDTV), *Standard Definition Television* (SDTV) e 1SEG (TAKADA; SAITO, 2006).

O ISDB-T_B (Versão B) foi desenvolvido a partir do padrão Japonês de TV digital ISDB-T e possui algumas inovações, tais como: interatividade (DTV_i) e os formatos de codificação de áudio e vídeo. A alocação de canal também difere do sistema Japonês.

A interatividade permite que o usuário tenha acesso, por meio de aplicativos, a informações como detalhes sobre a programação e jogos. Desenvolvido no Brasil, o *middleware* Ginga foi incorporado ao SBTVD (ABNT, 2008d). Os padrões de codificação de áudio e vídeo utilizados pelo sistema japonês foram substituídos por padrões mais eficientes e de maior poder de compressão. Para a codificação de vídeos foi adotado o padrão H.264 *Advanced Video Coding* (AVC) e para a codificação de áudio, o padrão H.264 *High Efficiency Advanced Audio Coding* (HE-AAC) (ABNT, 2008b; ABNT, 2008c).

Em relação à alocação de canal, no ISDB-T_B são utilizadas as bandas *Very High Frequency* (VHF) alto e *Ultra High Frequency* (UHF) com canais de 6 MHz de banda. Os canais 7 ao 13 (174 a 216 MHz) são pertencentes a banda VHF alto, enquanto os canais 14 ao 69 (470 a 806 MHz) são pertencentes a banda UHF (ABNT, 2008a). Uma exceção é o canal 37 (608 a 614 MHz), que é destinado à radioastronomia.

Entretanto, devido ao programa de desligamento do sinal analógico e a expansão das redes móveis para faixas de frequência antes destinadas à radiodifusão, a alocação de canal do ISDB-T_B sofrerá algumas alterações.

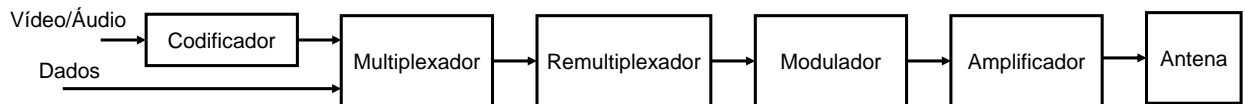
No dia 31 de setembro de 2014 ocorreu o leilão da faixa de 700 MHz para a utilização do serviço 4G da rede móvel, também chamado de *Long Term Evolution* (LTE). O LTE irá operar nas faixas de frequência de 708 a 748 MHz (*uplink*) e 763 a 803 MHz (*downlink*). As faixas de 703 a 708 MHz e 758 a 763 MHz serão destinadas ao Serviço Limitado Privado

(Agência Nacional de Telecomunicações, 2013).

2.1 Transmissão do Sistema ISDB-T_B

O sistema de transmissão ISDB-T_B consiste nos seguintes estágios: codificação da fonte de áudio e vídeo, multiplexação/remultiplexação, modulação e amplificação, para então o sinal ser enviado à antena (ABNT, 2008a). A Figura 3 apresenta o sistema brasileiro de transmissão terrestre.

Figura 3 – Transmissão do ISDB-T_B.



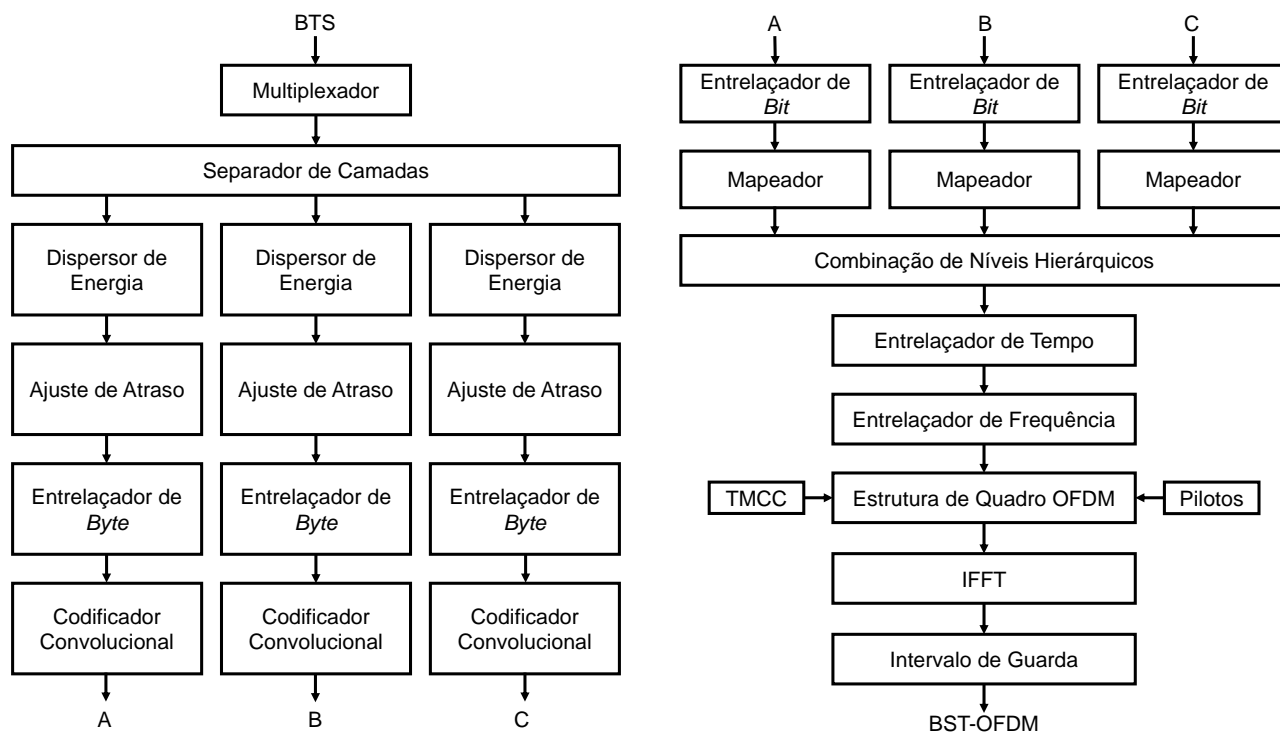
Fonte: ABNT (2008a, p.4).

O codificador é responsável por comprimir o sinal de áudio e vídeo, assim reduzindo a taxa de *bits* para que seja possível a transmissão pelo ar. A seguir o multiplexador recebe até três sequências de pacotes de dados codificados, chamados de *Transport Stream* (TS), que são multiplexados em um único TS. Esse sinal é enviado para o remultiplexador, que tem como objetivo inserir as informações de parâmetros de modulação e de sincronismo. O sinal de saída do remultiplexador é chamado de *Broadcast Transport Stream* (BTS) e é modulado e então enviado para um estágio de amplificação para ser transmitido pelo ar (AKAMINE, 2011; TAKADA; SAITO, 2006).

2.2 Codificação de Canal

O ISDB-T_B utiliza um sistema de codificação de canal juntamente com a modulação OFDM, como apresentado na Figura 4, para garantir a flexibilidade e a robustez do sistema. Os estágios da Figura 4 são explicados nas próximas seções.

Figura 4 – Codificação de canal do ISDB-T_B.



Fonte: (ABNT, 2008a, p.12).

2.2.1 Codificador Externo

O codificador externo possibilita que erros aleatórios e ruídos em rajada, como o ruído impulsivo, sejam corrigidos pelo receptor. Isso é possível por meio de um código *Reed Solomon* (RS). Na entrada do codificador são recebidos pacotes com o tamanho de 188 *bytes* provenientes do *Transport Stream Packet* (TSP). O RS acrescenta 16 *bytes* de paridade em cada pacote, com isso o tamanho do mesmo é acrescido para 204 *bytes*. Com a utilização desse código corretor de erros, o receptor tem a capacidade de corrigir até 8 *bytes* errados de cada pacote (ABNT, 2008a; AKAMINE, 2011).

2.2.2 Separador de camadas

O separador de camadas, ou divisor hierárquico, tem como função dividir o BTS em até três fluxos de dados TSP a partir de informações inseridas em cada pacote. Isto é, após a divisão, cada TSP é associado à sua respectiva camada hierárquica (ABNT, 2008a; AKAMINE, 2011).

2.2.3 Dispersor de energia

Os pacotes de 204 *bytes* recebidos pelo dispersor de energia podem conter longas sequências de *bits* zeros ou uns. Essas repetições podem ocasionar interferência entre símbolos. Para isso, o dispersor de energia tem como função distribuir os *bits* recebidos de forma aleatória utilizando uma sequência *Pseudo Random Binary Sequence* (PRBS) (ABNT, 2008a; AKAMINE, 2011).

2.2.4 Ajuste de Atraso

A função do ajuste de atraso é fazer com que o tempo de atraso de cada camada hierárquica seja igual. Esse atraso ocorre devido aos diferentes parâmetros de transmissão, que geram diferentes taxas de transmissão para cada camada. Calculando como período de tempo, as diferentes taxas geram diferença no tempo entre as camadas (ABNT, 2008a; AKAMINE, 2011).

2.2.5 Entrelaçador de *byte*

O entrelaçador de *bytes* tem como função embaralhar os dados recebidos. Essa operação é realizada a fim de minimizar os efeitos de ruído impulsivo e erros de bloco (ABNT, 2008a; MACIEL et al., 2015). Certos tipos de erros ocorrem de forma concentrada, portanto, quando as informações são desembaralhadas, os erros também são espalhados. Os códigos corretores de erros não são capazes corrigir erros concentrados, mas quando esses erros estão espalhados é possível detectar e corrigir os erros presentes no sinal (AKAMINE, 2011).

2.2.6 Codificador Convolutacional

Por meio da técnica *Forward Error Correction* (FEC), o codificador convolutacional acrescenta robustez ao sinal. A FEC trabalha com taxas de 1/2, 2/3, 3/4, 5/6 ou 7/8 (ABNT, 2008a; AKAMINE, 2011). No caso da taxa 2/3, por exemplo, significa que a cada 2 *bits* de informação é acrescentado 1 *bit* de correção. Pode-se concluir, portanto, que quanto maior o número de *bits* de correção, maior será a robustez do sinal. Entretanto,

a quantidade de informação útil enviada é reduzida.

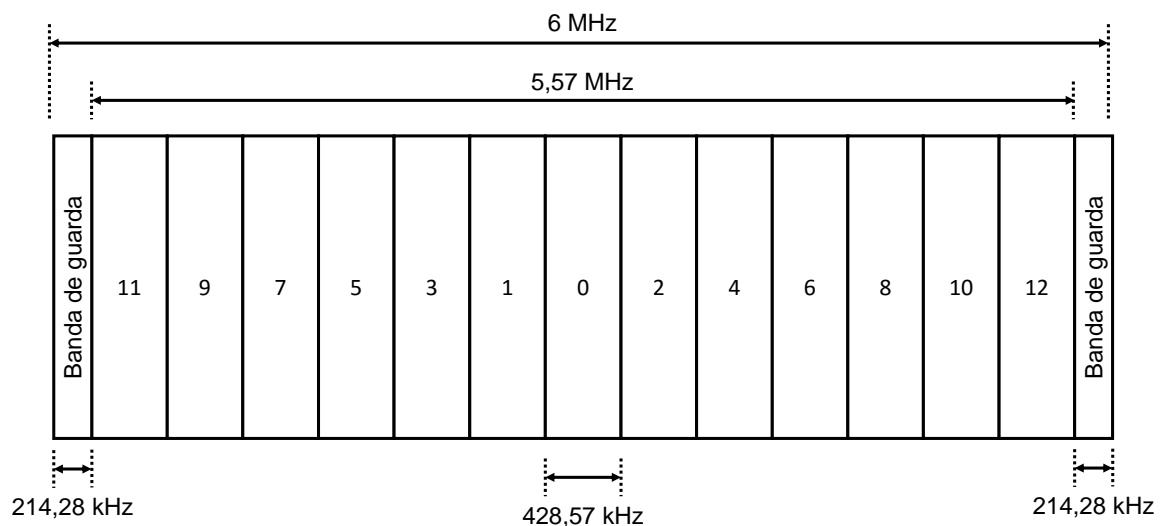
2.2.7 Entrelaçador de *bit*

O entrelaçador de *bit* é composto por um conversor serial/paralelo. Os *bits* de entrada são convertidos para paralelo em pacotes de 2 *bits*, 4 *bits* e 6 *bits* para as modulações QPSK, 16-QAM E 64-QAM, respectivamente (ABNT, 2008a; AKAMINE, 2011).

2.2.8 Mapeamento

A modulação utilizada pelo ISDB-T é a *Band Segmented Transmission - Orthogonal Frequency Division Multiplexing* (BST-OFDM). A modulação BST-OFDM divide um canal de TV (6 MHz) em 14 segmentos, como pode ser observado na Figura 5. Cada segmento tem uma banda de aproximadamente 428,571 kHz, e um deles é utilizado como banda de guarda do canal adjacente superior e inferior (AKAMINE, 2011; TAKADA; SAITO, 2006). Ao distanciar os canais adjacentes com a banda de guarda, a relação de proteção entre eles é melhorada.

Figura 5 – Espectro com BST-OFDM.



Fonte: (ABNT, 2008a, p.4).

A BST-OFDM permite combinar os 13 segmentos restantes em até três camadas hierárquicas, denominadas *layer A*, *layer B* e *layer C*, que podem ser moduladas de for-

mas distintas. Com isso, é possível utilizar até três tipos diferentes de serviços, como: 1SEG, SDTV e HDTV. Isso é chamado de transmissão hierárquica. No ISDB-T_B as camadas podem ser moduladas em *Quadrature Phase Shifting Key* (QPSK), 16 *Quadrature Amplitude Modulation* (16-QAM), e 64 *Quadrature Amplitude Modulation* (64-QAM) (ABNT, 2008a; TAKADA; SAITO, 2006).

2.2.9 Combinador de camadas

O combinador de camadas tem como objetivo concatenar os sinais das diferentes camadas hierárquicas para obter um único fluxo de dados na saída (ABNT, 2008a; AKAMINE, 2011).

2.2.10 Entrelaçador de tempo

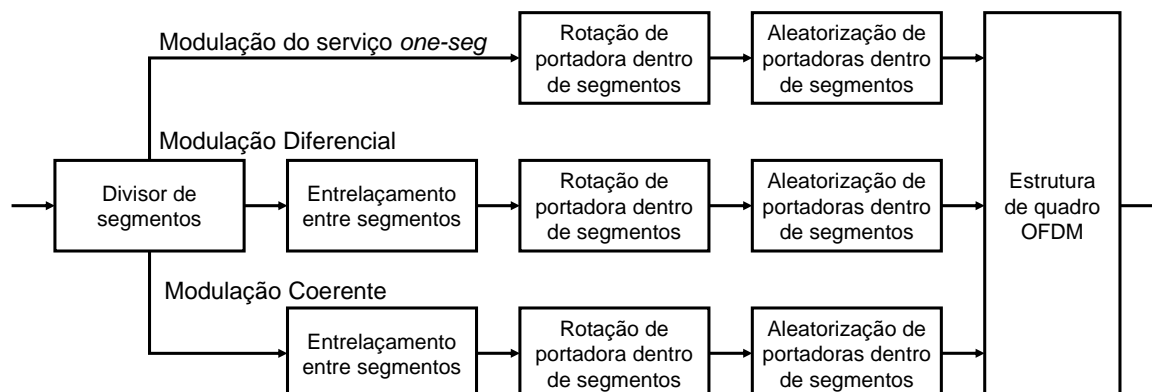
Após a concatenação dos sinais das diferentes camadas, o sinal resultante é entrelaçado no tempo. A configuração do tamanho do entrelaçamento pode ser feita para cada camada hierárquica variando-se o parâmetro I . Entretanto, os valores de comprimento do entrelaçamento dependem do modo de transmissão adotado (ABNT, 2008a). A função do entrelaçador de tempo, também chamado de *Time Interleaver* (TI), é aumentar a robustez contra desvanecimento e ruído impulsivo (AKAMINE, 2011).

2.2.11 Entrelaçador de frequência

Como pode ser observado na Figura 6, o entrelaçador de frequência é dividido em três entrelaçadores.

A primeira etapa é a divisão dos segmentos. Isto é, os números 0 a 12 dos segmentos de dados são enviados aos ramos de recepção parcial (*one-seg*), modulação diferencial e modulação coerente. O ramo de modulação diferencial é utilizado para os segmentos modulados em *Differential Quadrature Phase Shift Keying* (DQPSK), mas a utilização dessa modulação não é recomendada pela (NBR 15608-1). Já o ramo de modulação coerente é utilizado para os segmentos modulados em QPSK, 16-QAM e 64-QAM (ABNT, 2008a).

Figura 6 – Entrelaçador de frequência.



Fonte: (ABNT, 2008a, p.30).

Após a divisão dos segmentos, ocorre o entrelaçamento entre segmentos. Esse entrelaçador é responsável por distribuir as portadoras entre os segmentos de dados. Entretanto, se o serviço *one-seg* estiver habilitado, as portadoras do segmento de dados destinadas à recepção parcial não são entrelaçadas. Em seguida, ocorre o entrelaçamento dentro do segmento, o qual é dividido em rotação e randomização das portadoras. Nessas etapas as portadoras são deslocadas e depois randomizadas, sendo que a randomização é baseada em tabelas específicas que variam com o modo de transmissão adotado (ABNT, 2008a).

O objetivo da rotação e randomização é eliminar a periodicidade das portadoras para aumentar a eficiência da correção de erros. O entrelaçador de frequência evita que um símbolo seja perdido quando uma determinada faixa de frequência é afetada por um ruído de banda estreita ou multipercurso (ASSOCIATION OF RADIO INDUSTRIES AND BUSINESSES., 2003).

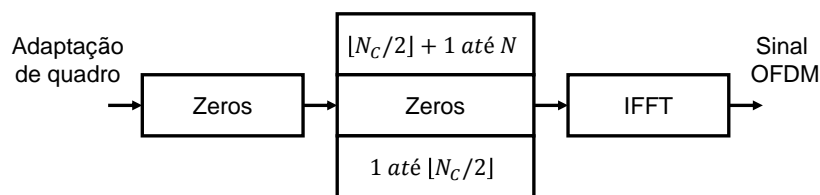
2.2.12 Estrutura de quadro OFDM e IFFT

A transmissão de um sinal ISDB- T_B é estruturada em quadros OFDM. Um quadro é formado por 204 símbolos OFDM. O número de subportadoras presentes em cada símbolo depende do modo de transmissão adotado. A quantidade de subportadoras totais em cada símbolo é igual a 1405, 2809 e 5617 para os modos 1, 2 e 3, respectivamente (ABNT, 2008a).

Nos símbolos OFDM são inseridos os sinais *Transmission and Multiplexing Configuration Control* (TMCC) e o *Auxiliary Channel* (AC). Esses sinais são responsáveis por indicar os parâmetros de modulação, codificação de canal, sincronismo de quadro e transmissão de dados auxiliares. Além desses sinais, são inseridos também os *Scattered Pilots* (SP). Esses sinais são adicionados para que seja possível realizar a sincronização em frequência e estimação do canal (ASSOCIATION OF RADIO INDUSTRIES AND BUSINESSES., 2003).

Após a adaptação de quadro, é necessário realizar o *zero padding*, ilustrado na Figura 7. Isso é feito para adicionar zeros nos dados do quadro OFDM para obter um número de amostras com um valor de 2^N (AKAMINE, 2011). Isso é necessário, pois a *Inverse Fast Fourier Transform* (IFFT) trabalha apenas com valores na potência de base 2. Por exemplo, no caso de uma transmissão no modo 3, são utilizadas 5617 portadoras. Como o valor mais próximo é 8192, deverão ser adicionados 2575 zeros.

Figura 7 – *Zero padding*.

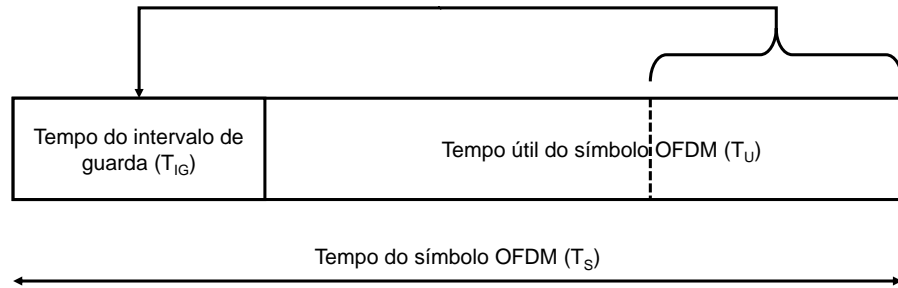


Fonte: (AKAMINE, 2011, p.84).

2.2.13 Intervalo de guarda

Na saída da IFFT, parte do símbolo é copiada e anexada ao seu início, como ilustrado na Figura 8. Essa cópia é denominada intervalo de guarda, ou prefixo cíclico. O tempo de duração de um símbolo OFDM (T_S) é igual ao tempo útil do símbolo (T_U) mais o tempo do intervalo de guarda (T_{IG}).

Figura 8 – Inserção do intervalo de guarda.



Fonte: Adaptado de ABNT (2008a, p.44).

A presença de multipercuro em um canal de comunicação provoca *Intersymbol Interference* (ISI) nos símbolos OFDM. Isso acontece devido ao espalhamento temporal causado pelo multipercuro, que faz com que informações de um símbolo vizinho também sejam processados pela *Fast Fourier Transform* (FFT). Para resolver esse problema, o intervalo de guarda é utilizado. Se o atraso do multipercuro for menor do que o tempo de duração do intervalo de guarda o símbolo é recuperado. Entretanto, se o atraso do multipercuro for maior, resultará em ISI (AKAMINE, 2011).

O intervalo de guarda pode ser configurado para $1/4$, $1/8$, $1/16$ ou $1/32$. Quanto maior o tempo do intervalo, maior é a robustez do sinal, pois aumenta o tempo de duração do prefixo cíclico, assim aumentando a proteção do sinal contra multipercuro (ABNT, 2008a; AKAMINE, 2011). Portanto, um sinal com um intervalo de guarda de $1/4$ possui maior imunidade ao multipercuro do que um sinal com um intervalo de $1/32$. Entretanto, quanto maior o intervalo de guarda, menor é a quantidade de informação útil no símbolo.

3 DESENVOLVIMENTO DO PROJETO

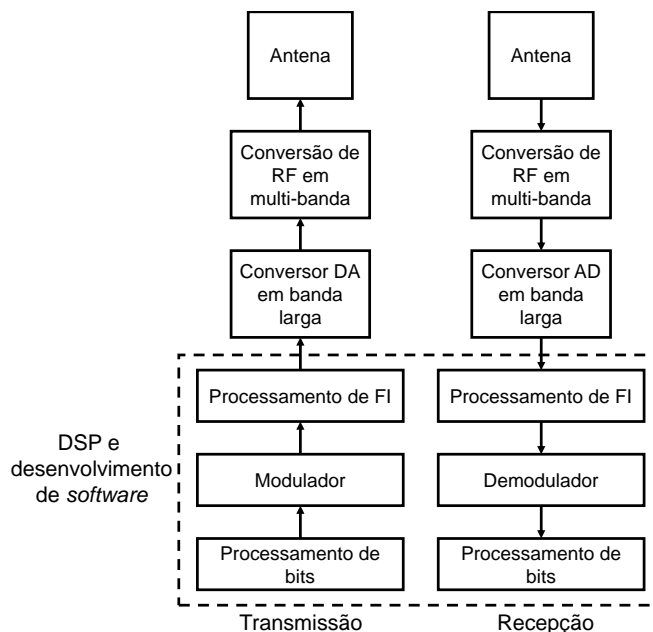
Nesta seção são abordadas as principais características de SDR e da plataforma GNU *Radio*. Também são explanadas a recepção de sinais RF, estimação de canal e técnicas de sincronismo. Por fim, são detalhados os blocos criados, testes realizados e resultados obtidos.

3.1 *Software Defined Radio (SDR)*

Entre os anos de 1970 e 1980, alguns sistemas de rádio migraram do analógico para o digital. Entretanto, a principal desvantagem desse tipo de sistema é que qualquer tipo de atualização no sistema o *hardware* deve ser trocado. Já com *Software Defined Radio* (SDR) basta reprogramar o rádio para que o mesmo efetue diferentes funções. Com isso, o *hardware* pode ser o mesmo para diversas aplicações, o que torna o sistema muito mais flexível. Obviamente, as funções implementadas serão limitadas pelo *hardware* (MITOLA, 1995).

A arquitetura de um SDR é mostrada na Figura 9.

Figura 9 – Estrutura de um SDR.

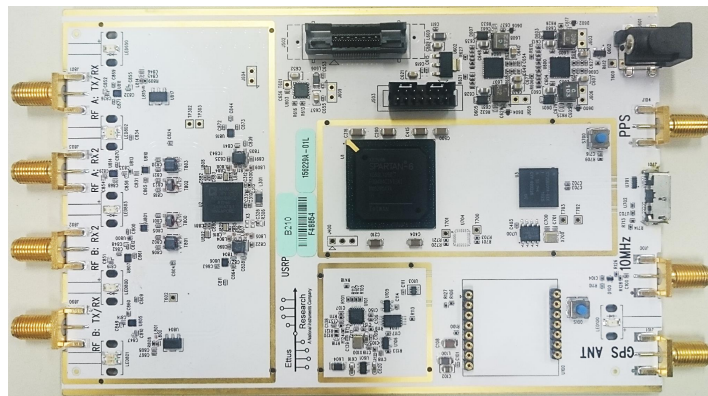


Fonte: Adaptado de Mitola (1999, p.517).

Como pode ser observado, um SDR pode trabalhar como um receptor e/ou transmissor. Para a transmissão, as informações são processadas e então moduladas. O sinal resultante tem sua frequência deslocada para uma Frequência Intermediária (FI). Entretanto, existem alguns modelos de SDR que não utilizam FI. Após ser convertido de digital para analógico, esse sinal é irradiado para o ar. No caso da recepção, o processo inverso é realizado.

A placa USRP B210 da empresa Ettus, exibida na Figura 10, ilustra um *kit* real de SDR.

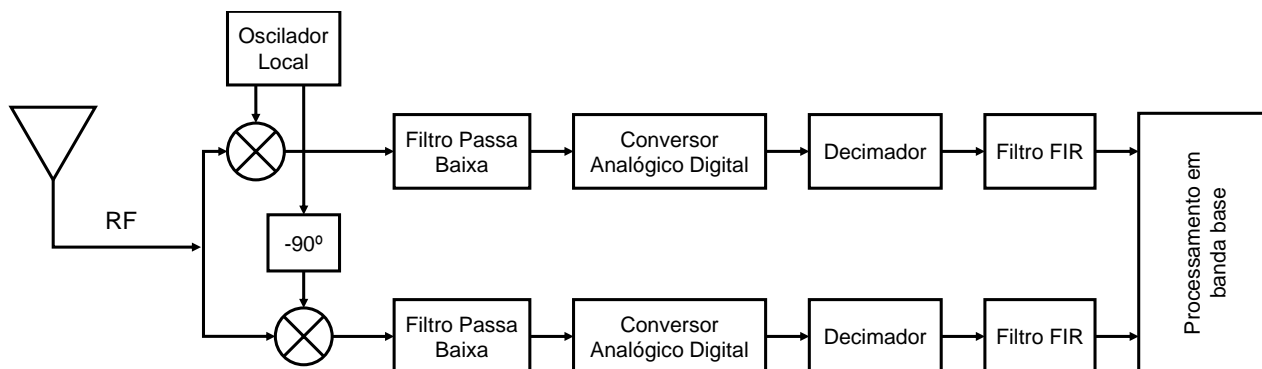
Figura 10 – Placa USRP B210.



Fonte: Acervo próprio.

Essa placa é capaz de realizar transmissões ou recepções simultaneamente e em tempo real, além de operar com frequências de 70 MHz até 6 GHz e trabalhar com larguras de banda de até 56 MHz. A B210 também possui uma *Field Programmable Gate Array* (FPGA) Spartan 6. A recepção de sinais de RF é realizada com o *chip* AD9361 (ANALOG DEVICES, 2014; ETTUS RESEARCH, 2014). O diagrama de blocos simplificado do AD9361 pode ser observado na Figura 11.

Figura 11 – Demodulação em quadratura (*chip* AD9361).



Fonte: Adaptado de Analog Devices (2014, p.1).

O sinal recebido passa por um *Automatic Gain Control* (AGC). A seguir, o sinal é convertido para banda base por meio de uma demodulação em quadratura. Para isso, o sinal é multiplicado por um seno e um cosseno, de forma que se obtenha os valores de fase e quadratura do sinal. Logo após, um filtro passa baixa é utilizado para reduzir a interferência de canal adjacente e filtrar as replicações espectrais geradas pelas multiplicações (AKAMINE, 2011). Agora em banda base, o sinal é digitalizado por conversor analógico/digital com uma quantização de 12 *bits*/amostra. O sinal resultante passa por um filtro de decimação e *Finite Impulse Response* (FIR), assim produzindo um sinal de 12 *bits* com taxa de amostragem apropriada (ANALOG DEVICES, 2014). O sinal resultante é enviado para o estágio de processamento.

3.2 GNU Radio

O GNU *Radio* é um *software* de desenvolvimento gratuito e aberto. Esse *software* possui uma interface gráfica, GRC, similar a programas como *Simulink* e *LabView*. Desta forma, permitindo interligar blocos na forma de um fluxograma, o que facilita a visualização do sistema implementado. O GRC trabalha com o conceito de *flow graph*. Isto é, um fluxograma por onde ocorre todo fluxo de dados. Os processamentos são realizados individualmente por cada bloco, o que torna o sistema modular e flexível. Além disso, todo processamento é realizado em tempo real (BRAUN, 2014; HILBURN, 2015).

Os blocos criados no GNU Radio podem ser desenvolvidos com duas linguagens de

programação: C++ ou *python*. Entretanto, por questões de desempenho, é preferível que a linguagem c++ seja utilizada na criação de funções para os blocos criados (RONDEAU, 2016). Para a organização, controle e conexão entre os blocos é utilizada a linguagem *Python*.

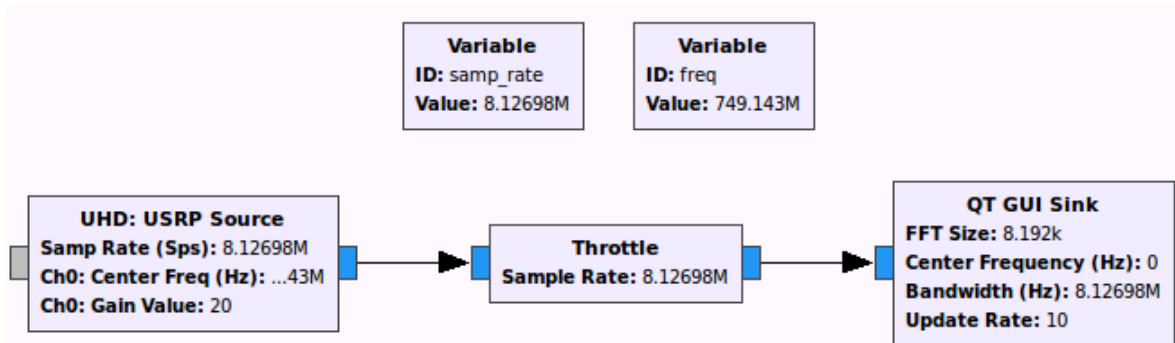
O GRC permite a geração de *flow graphs* sem a necessidade de escrever códigos em *Python*. Por meio do GRC é possível acessar diversos blocos com funções pré-definidas, além de blocos criados pelo próprio usuário. A interface gráfica dos blocos é escrita na linguagem *eXtensible Markup Language* (XML) (VACHHANI; MALLARI, 2015).

A ligação entre C++ e *Python* é realizada com a ferramenta *Simplified Wrapper and Interface Generator* (SWIG). Ao executar um *flow graph*, um arquivo *Python* é gerado. Nesse arquivo estão as linhas de código que representam o *flow graph* criado. Além disso, nesse arquivo é explicado como os objetos e funções correspondentes da biblioteca C++ compartilhada devem ser chamados por meio da ferramenta SWIG (VACHHANI; MALLARI, 2015).

Cada bloco possui um *buffer* de entrada e saída, com exceção dos blocos dedicados somente à escrita ou leitura. Quando um *flow graph* é executado, os blocos verificam o tamanho do *buffer* de saída para determinar a quantidade de elementos de entrada necessários para preencher o *buffer*. Entretanto, os tamanhos de entrada e saída podem ser diferentes, como acontece em blocos de interpolação ou decimação, por exemplo. Se existirem elementos de entrada suficientes para o *buffer* de saída o bloco é executado. Caso contrário, o bloco entrará em um modo de espera (VACHHANI; MALLARI, 2015).

As saídas e entradas do blocos podem ser configuradas para amostras ou vetores. Também é possível trabalhar com sinais de diferentes tipos, como reais, complexos, entre outros (RONDEAU, 2016). Um exemplo de fluxograma com alguns blocos é ilustrado na Figura 12. Nesse exemplo, foi utilizada a placa USRP B210, com uma antena, conectada ao computador.

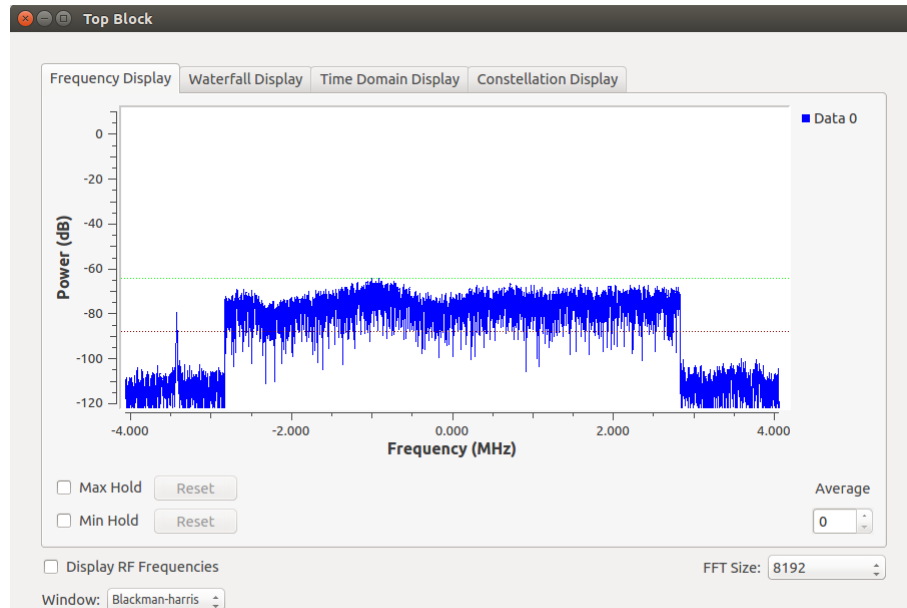
Figura 12 – Exemplo de fluxograma criado no GNU *Radio*.



Fonte: Acervo próprio.

O bloco “UHD: USRP *Source*” capta as informações recebidas pela USRP. A frequência central adotada foi de 748.143MHz , que corresponde ao canal 60 (TV Mackenzie). Como taxa de amostragem (*sample rate*) foi utilizado o valor $512 \times 10^6 / 63$ [*amostras/segundo*], equivalente ao valor da taxa de amostragem da FFT de um sinal ISDB- T_B no modo 3. O bloco “*Throttle*” é utilizado para limitar a taxa de transmissão de dados. Isso deve ser feito, pois, na ausência desse bloco, 100% do processador será utilizado na execução do *flow graph*. Assim, fazendo com que os elementos da interface gráfica não respondam. Entretanto, isso acontece apenas em *flow graphs* que são apenas simulações. Isto é, que não possuem blocos de áudio ou blocos para se conectarem a uma USRP. Após o bloco “*Throttle*”, o sinal é enviado ao bloco “QT GUI *Sink*”, que mostra o sinal no domínio do tempo, na frequência e a constelação do sinal. Na Figura 13 é possível observar o espectro do sinal gravado do ar.

Figura 13 – Espectro do canal 60 (TV Mackenzie/SP).



Fonte: Acervo próprio.

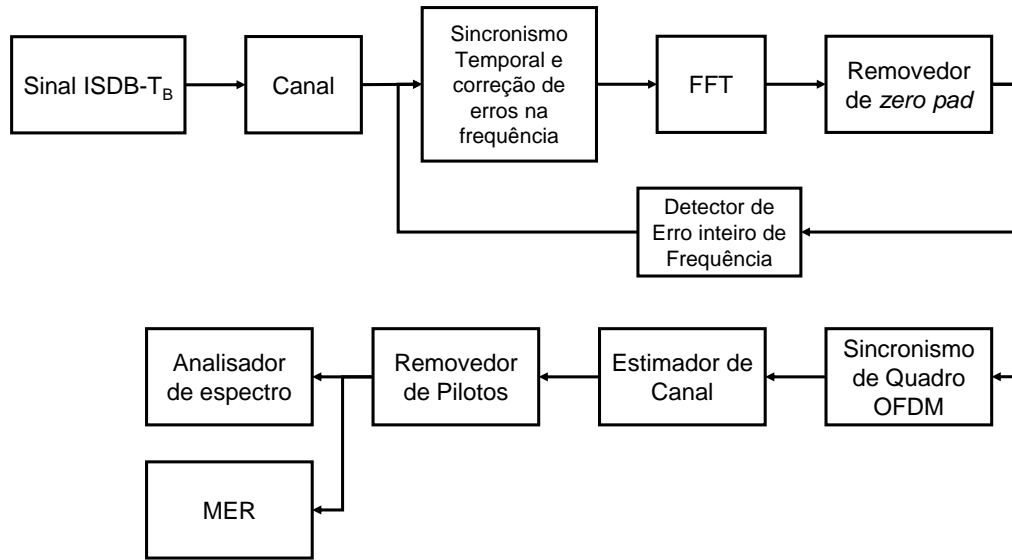
Por meio deste ambiente de desenvolvimento é possível criar, simular e implementar sistemas reais de rádio. Entretanto, como o GRC trabalha apenas com sinais digitais, para uma aplicação real é necessário um *hardware* capaz de realizar as conversões necessárias para que seja possível transmitir e receber sinais de RF (BRAUN, 2014).

3.3 Demodulador ISDB-T no GRC

Um sinal ISDB-T em banda base deve passar por alguns estágios de sincronismo antes de ser demodulado, como ilustrado na Figura 14. Após os estágios de sincronismo é realizada a estimação de canal.

O sincronismo temporal é necessário para detectar o início dos símbolos OFDM recebidos. Na ausência de um sincronismo temporal a FFT não processa o símbolo OFDM no lugar correto, assim resultando em ISI. A próxima etapa é o sincronismo de frequência, que corrige o desvio de frequência do sinal. Essa correção permite que os símbolos corretos sejam demodulados, além de evitar *Inter-Carrier Interference* (ICI). A seguir deve ser corrigido o desvio de frequência da frequência de amostragem do conversor A/D, pois esse desvio causa o deslocamento da janela FFT. Entretanto esse bloco não foi implementado, pois está fora do escopo do trabalho. Para detectar a posição das portadoras pilotos

Figura 14 – Diagrama de sincronismo.



Fonte: Acervo próprio.

espalhadas, e o início dos quadros OFDM, é realizado o sincronismo de quadro OFDM (AKAMINE, 2011; LARROCA et al., 2015).

Após os estágios de sincronismo deve ser realizada a estimação de canal, necessária para a recuperação do sinal devido ao desvanecimento seletivo. Devido ao conhecimento da posição e valores das portadoras pilotos transmitidas, é possível obter a função de transferência das pilotos na recepção. A partir da função de transferência das pilotos é possível obter a resposta em frequência do canal (MA et al., 2013; PADERNA; HIGASHINO; OKADA, 2014).

3.3.1 Desenvolvimento dos blocos

No GRC foi criado um *flowgraph* com os blocos apresentados na Figura 14. Os blocos “Sincronismo temporal”, “Detecção de erro inteiro de Frequência”, “Removedor de *zero pad*”, “Sincronismo de quadro OFDM” e “Estimador de Canal” foram desenvolvidos por meio das linguagens de programação C++. Essa linguagem foi utilizada para escrever o processamento de sinal que cada bloco deve realizar.

Como fonte de sinal foram utilizados sinais ISDB-T_B em banda passante, com intervalos de guarda de 1/8 e 1/16 e sem nenhum ruído. Esses sinais foram gerados pelo mo-

dulador criado por Akamine (2011). Com a placa USRP foram gravados sinais ISDB-T_B com as mesmas configurações a partir de equipamentos disponibilizados pelo Laboratório de TV Digital da Universidade Presbiteriana Mackenzie.

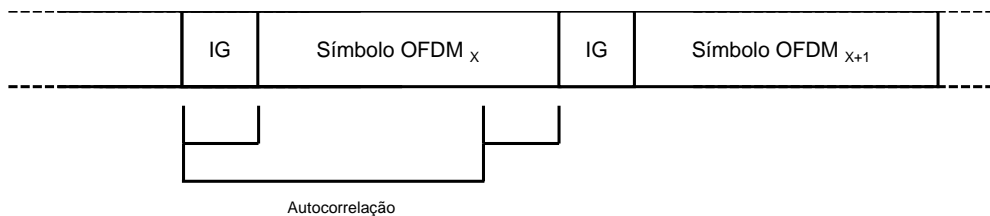
3.3.2 Sincronismo temporal

O bloco “Sincronismo temporal” foi criado para realizar o sincronismo temporal, além de corrigir desvios fracionários de frequência. Para realizar o sincronismo temporal foi utilizada uma técnica pré-FFT (KELLER et al., 2001; LARROCA et al., 2015). Essa técnica, apresentada na Equação 1, consiste em uma autocorrelação, que se aproveita do intervalo de guarda para identificar o ponto ótimo do símbolo OFDM.

$$C(i) = \sum_{n=0}^{N_{ig}-1} Z(i-n) \times Z(i-n-N_{FFT})^* \quad (1)$$

As multiplicações da autocorrelação são realizadas entre amostras complexas recebidas, $Z(i)$, e amostras complexas conjugadas, $Z^*(i)$, espaçadas de N , que é o tamanho da FFT. Isto é, amostras mais recentes são multiplicadas por amostras mais antigas. O índice i representa a amostra mais recente e é incrementado até $N_{FFT} - N_{ig}$. O termo N_{FFT} indica o tamanho da FFT, que é, também, o tamanho do símbolo. N_{ig} é o tamanho do intervalo de guarda. Logo, pode-se concluir que são necessários dois símbolos OFDM para realizar cada autocorrelação. Quando i for incrementado até $N_{FFT} - N_{ig}$, a amostra mais nova fará parte de outro símbolo. A Figura 15 ilustra essa autocorrelação.

Figura 15 – Autocorrelação para sincronismo temporal.



Fonte: Acervo próprio.

A cada incremento de i são realizadas as somas dos valores das multiplicações. Esses valores são representados pela função $C(i)$. A quantidade de somas e multiplicações é estabelecida pelo termo N_{ig} . O pico da autocorrelação é obtido quando a janela da

correlação está alinhada com o intervalo de guarda.

Para normalizar os valores obtidos na autocorrelação foi utilizada a Equação 2, responsável por calcular a energia do sinal recebido (AKAMINE, 2011).

$$P(i) = \sum_{n=0}^{N_{ig}-1} |Z(i+n)|^2 \quad (2)$$

Finalmente, com a Equação 3, é possível detectar o início do símbolo ao encontrar o índice i do maior valor obtido (AKAMINE, 2011). Entretanto, essa equação é válida somente se a frequência de amostragem estiver correta. Dessa forma, o sinal pode ser sincronizado no tempo e ter o intervalo de guarda removido. A próxima etapa é corrigir o desvio de frequência.

$$M = \arg \max_i \left(\frac{|C(i)|^2}{P(i)^2} \right) \quad (3)$$

Com o intuito de reduzir a complexidade computacional neste projeto, as autocorrelações foram implementadas de forma que o índice i fosse incrementado em diferentes passos (AKAMINE, 2011). Isto é, a primeira correlação é realizada em passos de 250 amostras. Ao achar a posição com maior valor, uma nova autocorrelação é feita, porém somente em algumas amostras próximas ao índice encontrado. A cada nova correlação o número de passos é reduzido pela metade. Esse processo é repetido até que o passo da correlação seja igual a um. Dessa forma o processamento computacional é reduzido drasticamente.

O desvio de frequência (Δ) é composto por dois termos que se somam. São eles o desvio inteiro (Δ_i) e desvio fracionário (Δ_f) (AI, 2004). Assim obtendo a Equação 4.

$$\Delta = \Delta_i + \Delta_f \quad (4)$$

O termo Δ_i representa um múltiplo inteiro n de Δ_F . Isso significa que as subportadoras têm suas posições deslocadas em n posições. As subportadoras mantêm a ortogonalidade, porém, como os símbolos não estão em suas devidas posições não será possível decodificá-los, assim, gerando erros na decodificação. Já o termo Δ_f representa um desvio

de valor menor que Δ_F . Isso provoca um espalhamento de energia que causa a perda da ortogonalidade e ICI (AKAMINE, 2011).

O termo Δ_f pode ser calculado por meio da Equação 5.

$$\Delta_f = \frac{\tan^{-1}\left(\frac{\text{Im}(M)}{\text{Re}(M)}\right)}{2\pi} \Delta_F \quad (5)$$

Como pode ser observado, o cálculo do desvio fracionário utiliza o valor obtido pela Equação 3, que é o pico da correlação realizada para o sincronismo temporal.

Para corrigir o Δ_f do sinal deve-se multiplicar o mesmo por uma exponencial complexa, assim deslocando o sinal para a frequência correta, como apresentado na Equação 6.

$$Z(i) = Z_e(i)e^{-j2\pi\Delta_f t} \quad (6)$$

O termo t representa o tempo, e $z_e(i)$, o sinal recebido com Δ_f . Neste projeto, como a implementação é realizada com amostras, a Equação 6 pode ser reescrita como mostrado na Equação 7.

$$Z(n) = Z_e(n) \times e^{\frac{-j2\pi\Delta_f n}{F_s}} \quad n = 0, 1, 2, \dots, N \quad (7)$$

Sendo F_s a frequência de amostragem da FFT, N o número total de amostras, e n o índice da amostra atual.

O próximo estágio a ser realizado é o sincronismo de frequência.

3.3.3 Sincronismo de frequência

Outro bloco criado foi o “Estimação de Frequência”, responsável por calcular Δ_i . Para tal, foi utilizada a técnica de correlação cruzada. Essa correlação, mostrada na Equação 8 (AI, 2004), se aproveita das portadoras piloto *Auxiliary Channel* (AC) e *Transmission*

and *Multiplexing Configuration Control* (TMCC) do sinal ISDB-T_B.

$$\varepsilon_i = \arg \max_l \left\{ \sum_{k=l}^{N_p} |Y_{m,k} \cdot P_{m,k+l}| \right\} \quad (8)$$

O termo N_p representa é o número de pilotos contínuas, $Z_{m,k}$ é o valor complexo recebido da k -ésima piloto no m -ésimo símbolo OFDM depois da FFT, e l é o índice das portadoras piloto. $P_{m,k+l}$ é um sinal de referência, isto é, um sinal que possui as pilotos contínuas nos lugares corretos. O pico da correlação, obtido quando as pilotos do sinal recebido e do sinal de referência estão sobrepostas, indica o valor de posições deslocadas (ε_i) das subportadoras. O cálculo de Δ_i pode ser observado na Equação 9.

$$\Delta_i = \varepsilon_i \cdot \Delta_F \quad (9)$$

O bloco “Estimação de Frequência” envia o valor de Δ_i para o bloco “sincronismo temporal”. Assim, com os valores de Δ_i e Δ_f , o bloco consegue corrigir o sinal recebido por meio da Equação 10.

$$Z(n) = Z_e(n) \cdot e^{\frac{-j2\pi(\Delta_f + \Delta_i)n}{F_s}} \quad n = 0, 1, 2, \dots, N \quad (10)$$

Após o sincronismo de frequência deve-se realizar a operação inversa do *zero padding*. Isto é, deve-se retirar os zeros presentes no sinal. Para isso foi criado o bloco “Removedor de *zero pad*”. A seguir é realizada a estimação de canal.

3.3.4 Sincronismo de quadro OFDM

Antes de realizar a estimação de canal é necessário detectar o início do quadro OFDM. Para tal, são utilizadas as pilotos TMCC, as quais são inseridas na etapa de estrutura de quadro OFDM, como mostra a Figura 4. A quantidade de pilotos em cada símbolo depende do modo escolhido. No caso desse projeto, foi utilizado o modo 3, que possui 52 pilotos TMCC, moduladas em *Differential Binary Phase Shift Keying* (DBPSK), em cada símbolo. Para identificar o *bit* da TMCC em cada símbolo são necessários alguns passos. Para a realização desse sincronismo foi implementado o bloco “Sincronismo de quadro OFDM”.

A primeira etapa é realizar a demodulação *Binary Phase Shift Keying* (BPSK) em cada piloto utilizando a Equação 11 (AKAMINE, 2011),

$$D_{m,k} = \begin{cases} 1, & \text{para } \text{Re}\{TMCC_{m,k}\} < 0 \\ 0, & \text{para } \text{Re}\{TMCC_{m,k}\} \geq 0 \end{cases} \quad (11)$$

onde as variáveis m e k representam, respectivamente, o índice do símbolo OFDM e o índice da portadora dentro do símbolo. O valor demodulado é representado por $D_{m,k}$.

Após a demodulação, é aplicada uma lógica majoritária para determinar o valor da TMCC em cada símbolo, como pode ser observado na Equação 12 (AKAMINE, 2011).

$$D_m = \begin{cases} 1, & \text{para } \frac{1}{N_{TMCC}} \sum_{k=0}^{N_{TMCC}} D_{m,k} \geq 0,5 \\ 0, & \text{para } \frac{1}{N_{TMCC}} \sum_{k=0}^{N_{TMCC}} D_{m,k} < 0,5 \end{cases} \quad (12)$$

A seguir deve ser realizada a decodificação diferencial, como mostra a Equação 13. Dessa forma o *bit* da TMCC é recuperado (ASSOCIATION OF RADIO INDUSTRIES AND BUSINESSES., 2003).

$$B_m = \begin{cases} D_m \oplus W_m, & \text{para } m = 0 \\ D_{m-1} \oplus D_m, & \text{para } 1 \leq m \leq 203 \end{cases} \quad (13)$$

Ao realizar a decodificação no primeiro símbolo deve ser utilizado o sinal de referência para modulação diferencial W_m , pois não é possível utilizar um valor de símbolo anterior. Esse sinal de referência é baseado em um circuito PRBS (ASSOCIATION OF RADIO INDUSTRIES AND BUSINESSES., 2003). A próxima estapa é uma decodificação cíclica para detectar e corrigir *bits*. Entretanto, sem essa etapa já é possível identificar os *bits* na TMCC. Essa etapa não foi implementada neste projeto.

Na Tabela 1, são mostradas as informações indicadas pelos 204 *bits* da TMCC.

Os sinais de sincronização W_0 e W_1 são utilizados na detecção do início do quadro OFDM. Isso é feito por meio de uma correlação cruzada entre os 204 *bits* da TMCC e a sequência de 16 *bits* dos sinais de sincronização. A sequência de *bits* de W_0 e W_1 são, respectivamente, 0011010111101110 e 0011010111101110.

Tabela 1 – Atribuição dos *bits* da TMCC.

| | |
|-------------------|-------------------------------------------------------------|
| B_0 | Referência para demodulação diferencial |
| B_1-B_{16} | Sinal de sincronização (W_0 ou W_1) |
| $B_{17}-B_{19}$ | Identificação do tipo de segmento (diferencial ou síncrono) |
| $B_{20}-B_{121}$ | Informações da TMCC |
| $B_{122}-B_{203}$ | Bits de paridade |

Fonte: Adaptado de ASSOCIATION OF RADIO INDUSTRIES AND BUSINESSES. (2003, p.54)

Entre os *bits* de índice 20 e 121 estão contidas seguintes informações: sistema de identificação, indicador de comutação de parâmetros de transmissão, indicador para alarme de emergência de radiodifusão, indicador de recepção parcial, esquema de modulação de portadora, taxa de codificação convolucional, comprimento entrelaçador temporal e número de segmentos. Todos esses *bits* são codificados com um código cíclico encurtado (ASSOCIATION OF RADIO INDUSTRIES AND BUSINESSES., 2003).

Após o sincronismo de quadro OFDM é realizada a estimação de canal.

3.3.5 Estimação de canal assistida por pilotos

A estimação de canal é necessária para recuperar o sinal ISDB-T após o mesmo ter sofrido interferências por desvanecimento seletivo. A estimação de canal assistida por pilotos necessita da resposta em frequência de cada piloto espalhada, representadas por H_{SP} na Equação 14 (BEEK et al., 1995).

$$H_{SP} = Y_{SP}/X_{SP} \quad (14)$$

onde Y_{SP} e X_{SP} são, respectivamente, as SP recebidas e transmitidas. As posições das SP são diferentes em cada símbolo OFDM. No caso do ISDB-T, existem 4 posições possíveis: 1, 4, 7 e 10. Esses valores representam a posição inicial de uma SP em um símbolo OFDM. A cada símbolo essas posições iniciais são trocadas. Isto é, se em um símbolo a SP começou na posição 1, então no próximo símbolo a posição inicial será em

4. Em cada símbolo as SP estão espaçadas de 12 portadoras. Esses espaços entre as SP precisam ser interpolados no domínio tempo e na frequência, conhecido como interpolação bidimensional (2D), ou somente no domínio da frequência. Existem diversos tipos de estimação de canal, mas nesse projeto foram implementados 4 tipos: *Linear Piecewise*, *Linear Piecewise 2D*, *Cubic Spline* e *Cubic Spline 2D*. A implementação da interpolação linear é simples e possui baixa complexidade computacional. A Equação 15 representa a interpolação linear,

$$H_i(k) = (1 - r).H_{SP}(k) + r.H_{SP}(k + 1) \quad (15)$$

onde $H_i(k)$ é o valor estimado, k é a posição da SP, $H_{SP}(k)$ é a função de transferência da SP atual e $H_{SP}(k+1)$ é a função de transferência da próxima posição da SP. A variável r representa a razão l/L , onde L é o número de amostras entre as SP, e l , a distância entre a SP atual e a k -ésima portadora (KANG; HA; JOO, 2003).

A interpolação *Cubic Spline* utiliza um polinômio de terceira ordem para estimar os valores entre cada SP. Essa técnica obtém os coeficientes dos polinômios a partir de 4 sinais adjacentes de referência e suas derivadas de segunda ordem, como é mostrado na Equação 16.

$$H(k) = A(r).H_{SP}(k) + B(r).H_{SP}(k + 1) + C(r).z(k) + D(r).z(k + 1) \quad (16)$$

As constantes $A(r)$, $B(r)$, $C(r)$ e $D(r)$ são determinadas pelo valor da razão r . As funções $z(k)$ e $z(k + 1)$ representam as derivadas de segunda ordem do n -ésimo sinal de referência (KANG; HA; JOO, 2003). Após esses cálculos, os valores interpolados são obtidos ao inserir os coeficientes adquiridos em um polinômio de terceira ordem. Essa interpolação é mais efetiva do que a linear. Entretanto, a implementação dessa técnica é mais complicada e possui uma alta complexidade computacional.

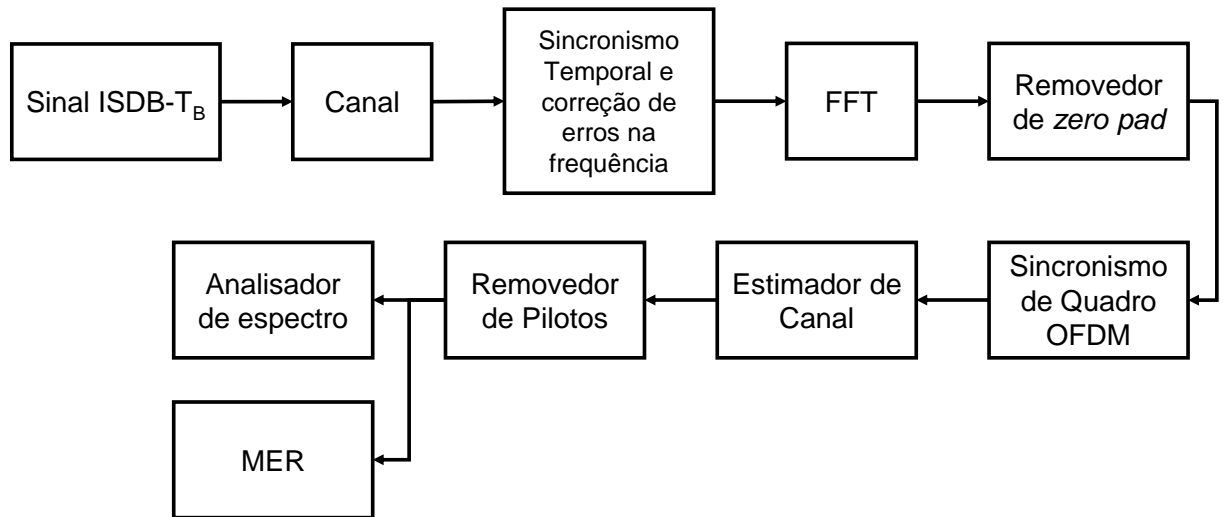
Após a realização da estimação de canal, as pilotos AC, TMCC e SP são removidas. Dessa forma, por meio de um analisador de espectro, é possível observar a constelação do sinal recebido sem as pilotos.

4 Resultados

Neste capítulo são apresentados os testes realizados, bem como os resultados obtidos.

Na Figura 16 está ilustrado o *flow graph* criado no GRC para testar os blocos de remoção de zeros e de sincronismo temporal. Para os testes de detecção e correção de erros inteiros e fracionários de frequência são realizadas pequenas modificações no *flow graph*, as quais serão mostradas ao longo do texto.

Figura 16 – *Flow graph* para o teste de remoção de zeros e sincronismo temporal

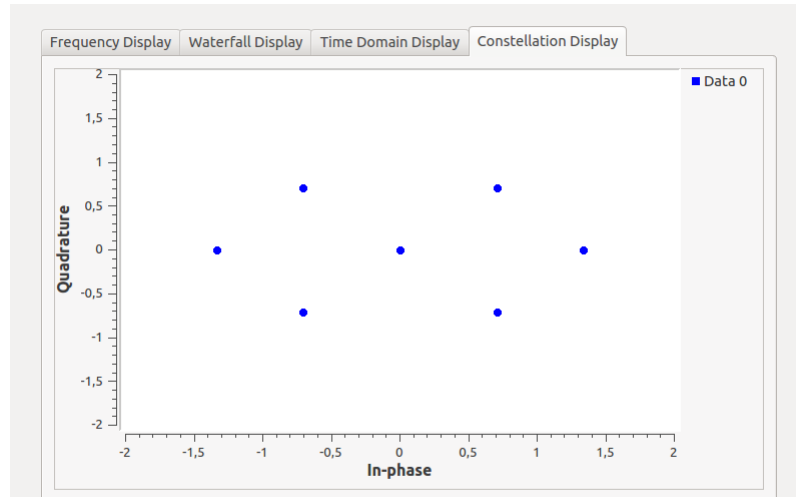


Fonte: Acervo próprio.

O bloco “*File source*” é responsável por transmitir as amostras do sinal ISDB-T_B armazenadas no computador. O bloco “*Delay*” insere zeros no começo da transmissão, assim provocando um atraso no tempo. Por meio do bloco “*Channel model*” é possível configurar um canal de comunicação. Nesse bloco é possível configurar parâmetros como: amplitude de ruído branco, desvio de frequência e multipercurso. Também é possível emular quando há taxas diferentes entre o transmissor e receptor. Após passar por esse bloco, o sinal é transformado em vetores pelo bloco “*Stream to vector*”. O sinal gravado possui 8192 portadoras e um intervalo de guarda de 1/8, logo, o tamanho dos vetores entregues à FFT devem ter um tamanho de 9216. Com a realização do sincronismo temporal o intervalo de guarda é removido, assim, o tamanho dos vetores volta a ser de 8192. A constelação do sinal após a FFT pode ser observado na Figura 17. Também

foram testados sinais com intervalo de guarda de 1/16.

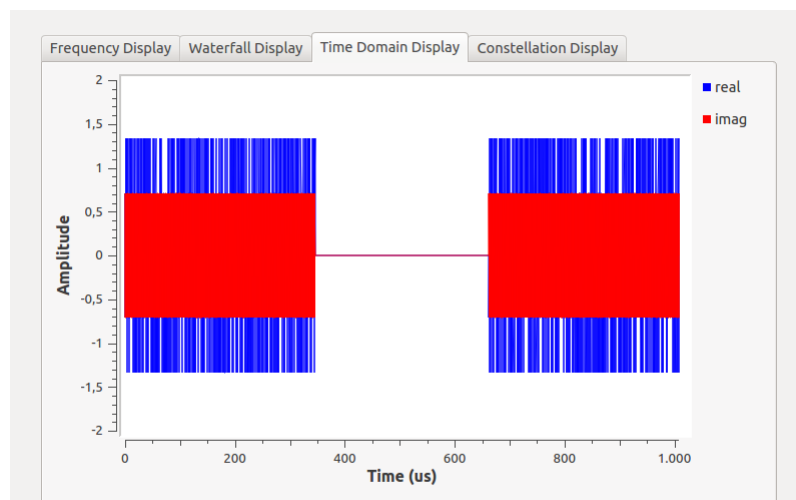
Figura 17 – Constelação do sinal após FFT.



Fonte: Acervo próprio.

Como é possível observar na constelação, as portadoras piloto e os zeros, inseridos devido a técnica de *zero padding*, ainda estão presentes no sinal. Os zeros e as portadoras piloto também podem ser observados no domínio do tempo, como ilustra a Figura 18.

Figura 18 – Zeros no domínio do tempo.

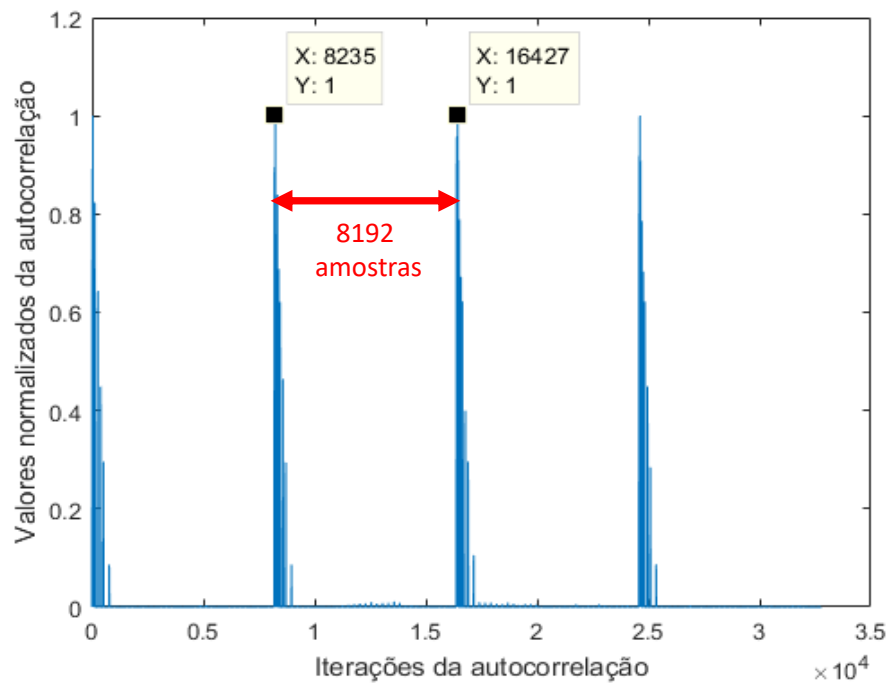


Fonte: Acervo próprio.

O próximo passo é retirar os zeros, por meio do bloco “*zero pad remover*”, e as pilotos, com o bloco “*FrameDeStr*”. Esse bloco foi utilizado por Maciel et al. (2015) e cedido pelos autores.

Para avaliar a implementação de sincronismo temporal foram analisados valores obtidos nas autocorrelações realizadas. Alguns desses valores podem ser observados na Figura 19.

Figura 19 – Valores de autocorrelações realizadas.



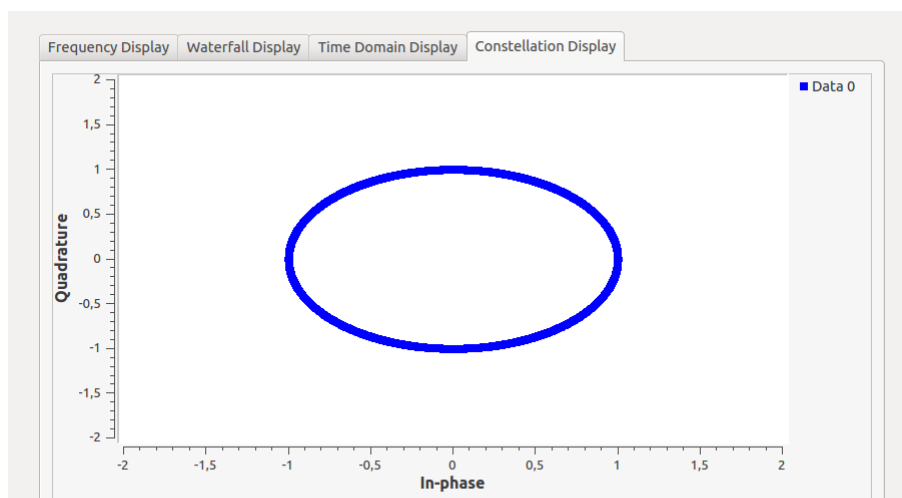
Fonte: Acervo próprio.

A distância entre os picos é de 8192 amostras, uma vez que esse é o tamanho da IFFT. A posição dos picos indicam a posição do início do símbolo OFDM. Portanto, pode-se concluir da 19 que o símbolo OFDM está atrasado no tempo o equivalente a 43 amostras, uma vez que um dos picos está na posição 8235. Isto é, $8235 - 8192 = 43$. Se não houvesse um atraso no tempo, as posições dos picos seriam múltiplos inteiros de 8192.

Com a posição do início do símbolo agora é possível remover o intervalo de guarda e enviar os símbolos OFDM a partir dos seus respectivos inícios.

Na Figura 20 é possível observar um sinal QPSK sem sincronismo no tempo.

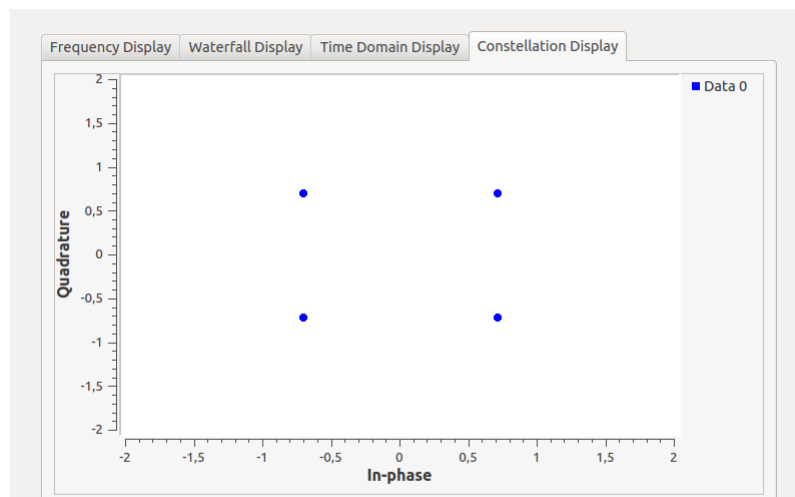
Figura 20 – Constelação do sinal ISDB-T sem sincronismo temporal.



Fonte: Acervo próprio.

Ao realizar o sincronismo temporal a constelação do sinal fica como ilustrado na Figura 21.

Figura 21 – Constelação do sinal ISDB-T após sincronismo temporal.



Fonte: Acervo próprio.

Para avaliar a detecção de erros fracionários de frequência foram inseridos erros constantes e erros com oscilações no sinal recebido, e, para medir o desempenho, o MER foi avaliado. O *flow graph* desse teste pode ser observado na Figura 22.

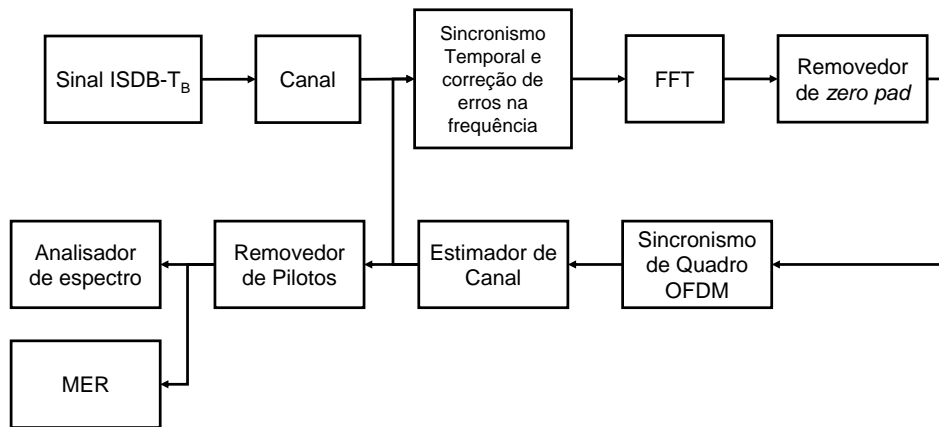
Para os erros com oscilações foram testadas ondas quadradas, triangulares e senoidais, como mostram as Figuras 23, 24 e 25. As variáveis $\Delta_f G$ e Δ_f representam, respectiva-

mente, o erro fracionário gerado e o erro fracionário detectado.

No caso da Figura 23, por exemplo, o @ $1Hz$ significa que o erro gerado tem uma oscilação de $1Hz$. Ou seja, o erro gerado está entre os valores 0 e 2, mas com uma variação de $1Hz$.

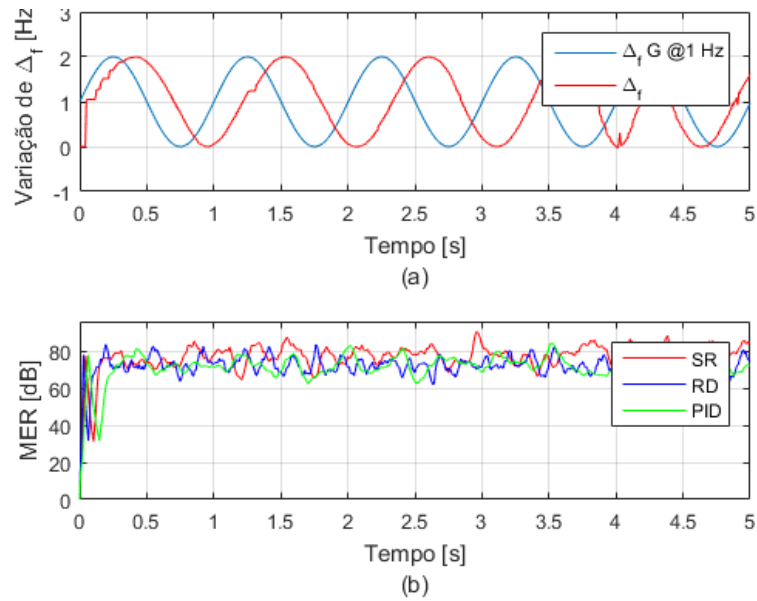
As siglas SR, RD e PID significam, respectivamente, sem realimentação, com realimentação direta e com realimentação direta controlada por um PID. A realimentação direta envia valores de erros residuais de frequência calculados por meio uma implementação cedida por (AKAMINE, 2011). Esses erros são calculados dentro do bloco de estimação de canal e são enviados para o bloco de correção de erro de frequência.

Figura 22 – *Flow graph* para os testes de detecção de erros fracionários



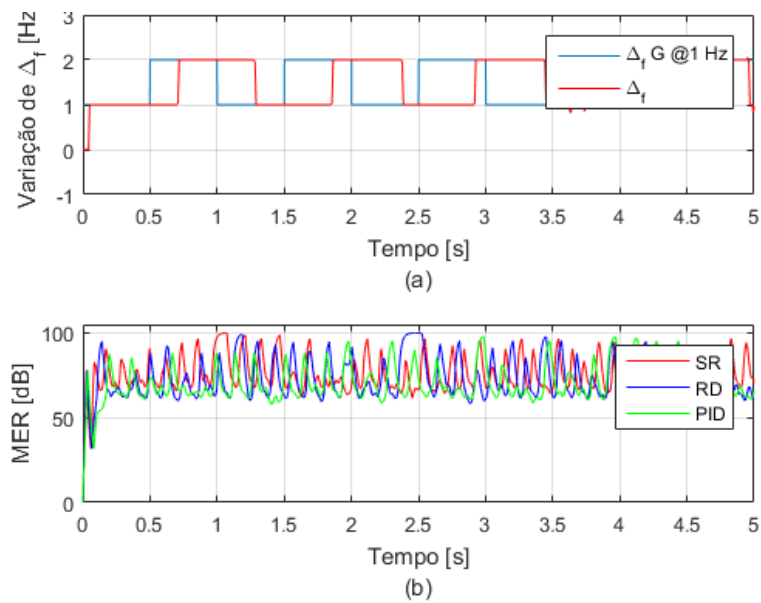
Fonte: Acervo próprio.

Figura 23 – Valores de MER em função de erro fracionário com variação senoidal de 1 Hz: (a) Erro fracionário gerado e detectado; (b) MER para sistema com e sem realimentação



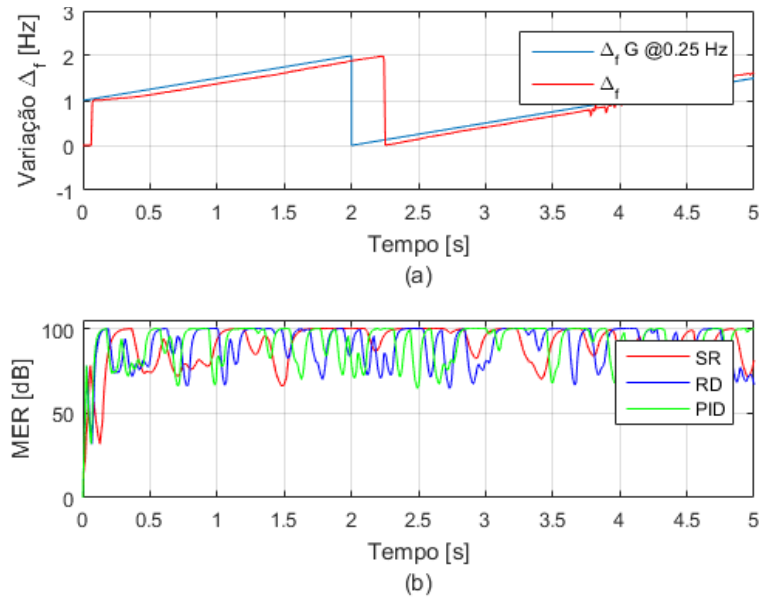
Fonte: Acervo próprio.

Figura 24 – Valores de MER em função de erro fracionário com variação de onda quadrada de 1 Hz: (a) Erro fracionário gerado e detectado; (b) MER para sistema com e sem realimentação



Fonte: Acervo próprio.

Figura 25 – Valores de MER em função de erro fracionário com variação de onda dente de serra de 0.25 Hz: (a) Erro fracionário gerado e detectado; (b) MER para sistema com e sem realimentação

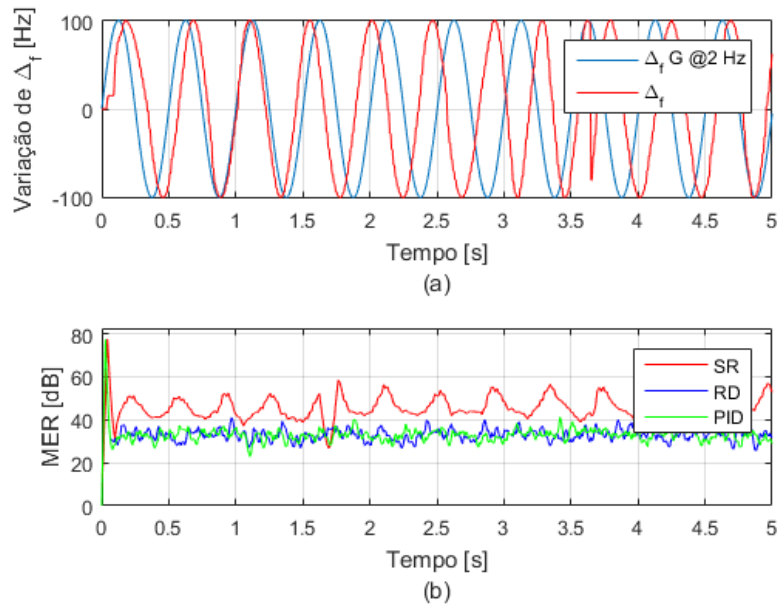


Fonte: Acervo próprio.

Como pode ser observado, devido ao tempo de processamento existe uma diferença entre o tempo do erro gerado e de sua detecção. Além disso é possível observar que o desempenho do sistema foi melhor sem a utilização de realimentação de erros residuais. Note que os valores de MER obtido são valores impossíveis de serem encontrados na prática. Esses valores foram obtidos, pois são testes simulados em computador.

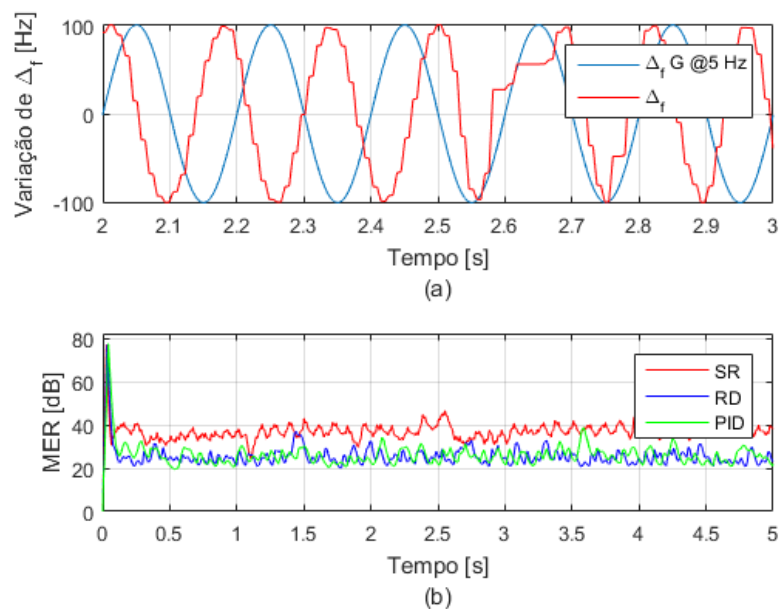
A seguir, nas Figuras 26, 27 e 28, são mostrados os testes com erros fracionários com valores maiores de oscilação.

Figura 26 – Valores de MER em função de erro fracionário com variação senoidal de 2 Hz:
 (a) Erro fracionário gerado e detectado; (b) MER para sistema com e sem realimentação



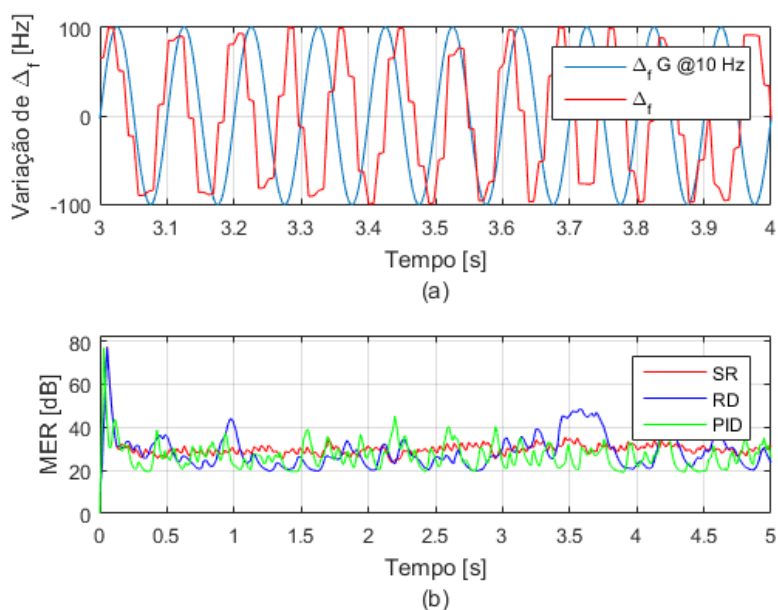
Fonte: Acervo próprio.

Figura 27 – Valores de MER em função de erro fracionário com variação senoidal de 5 Hz:
 (a) Erro fracionário gerado e detectado; (b) MER para sistema com e sem realimentação



Fonte: Acervo próprio.

Figura 28 – Valores de MER em função de erro fracionário com variação senoidal de 10 Hz: (a) Erro fracionário gerado e detectado; (b) MER para sistema com e sem realimentação



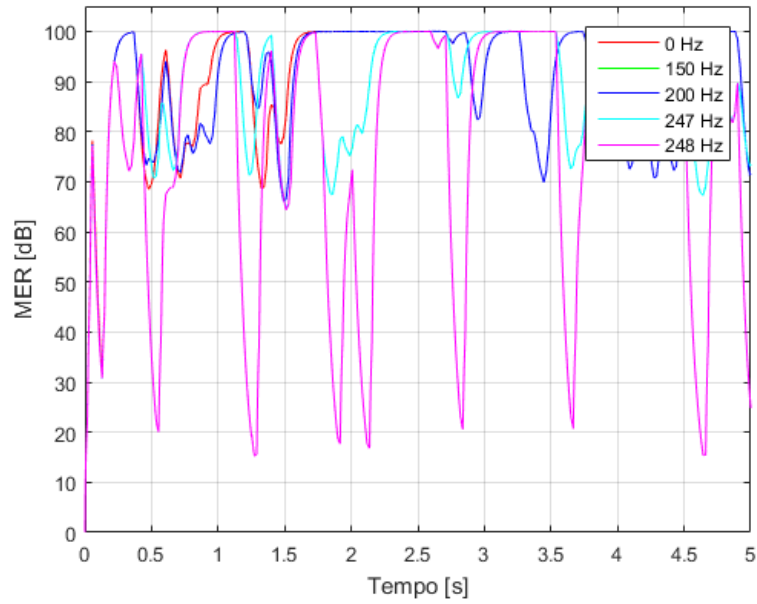
Fonte: Acervo próprio.

É possível observar que quanto maior a oscilação do erro de frequência gerado, maior é a dificuldade do detector e os valores de MER obtidos são piores. Também pode-se concluir, ao observar os valores obtidos de MER, que o sistema sem realimentação obteve um melhor resultado.

É possível perceber, também, que quanto maior a oscilação, maior é o tempo de resposta do detector. Com isso, no caso de uma oscilação considerada alta para esse sistema, o erro pode não ser detectado corretamente.

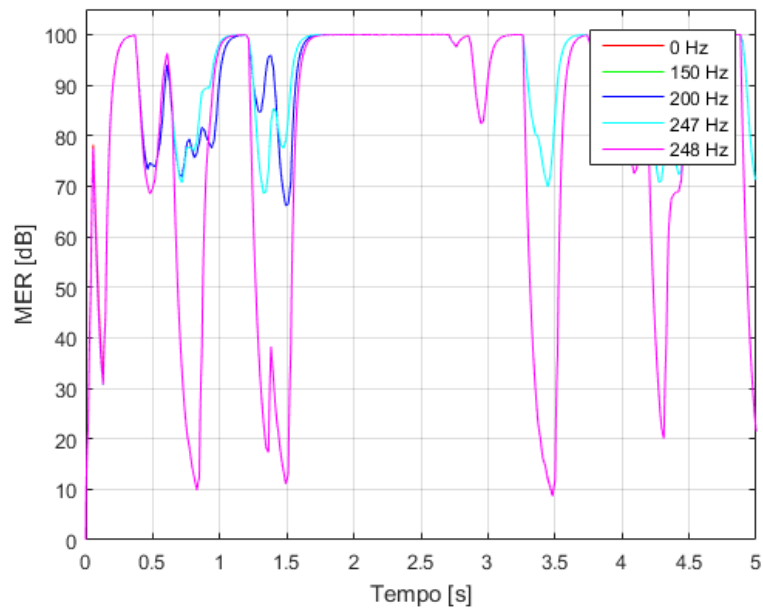
Os testes com erros fracionários constantes, isto é, sem oscilação, podem ser observados nas Figuras 29, 30 e 31. Foram realizados testes com e sem realimentação. O objetivo do teste foi identificar o limite de detecção de erro fracionário.

Figura 29 – Valores de MER em função de erro fracionário constante (Sem realimentação)



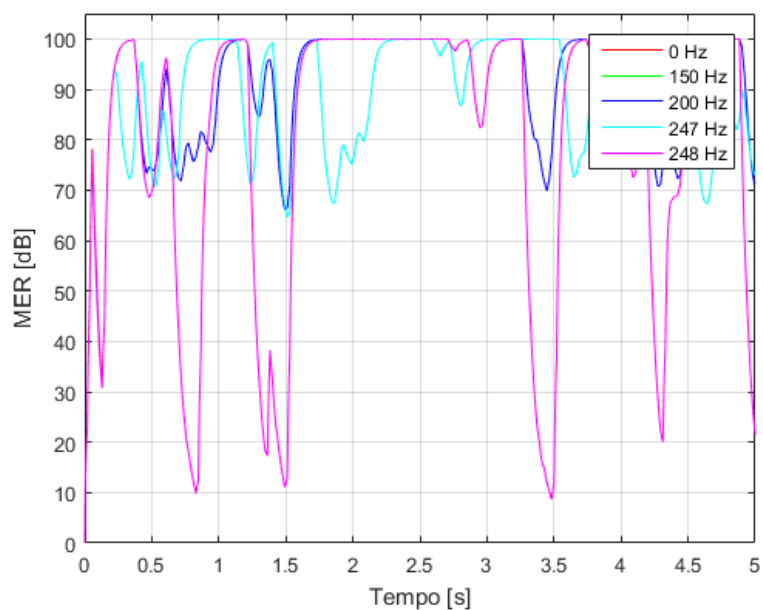
Fonte: Acervo próprio.

Figura 30 – Valores de MER em função de erro fracionário constante (Com realimentação direta)



Fonte: Acervo próprio.

Figura 31 – Valores de MER em função de erro fracionário constante (Com realimentação controlada por PID)

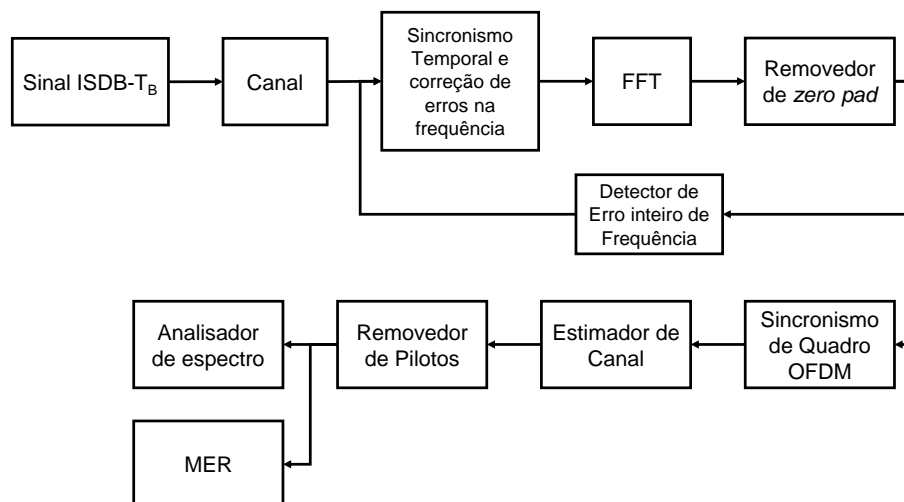


Fonte: Acervo próprio.

Como pode ser observado, nos casos com e sem realimentação, os valores de MER são praticamente os mesmos para valores abaixo de 247Hz . Entretanto, com valores próximos do limite, como é o caso do valor 240Hz , os valores de MER já apresentam resultados piores.

Para avaliar a detecção de erros inteiros de frequência, foram analisados valores obtidos nas correlações cruzadas ao inserir um erro inteiro no sinal recebido. O *flow graph* para esse teste é mostrado na Figura 32.

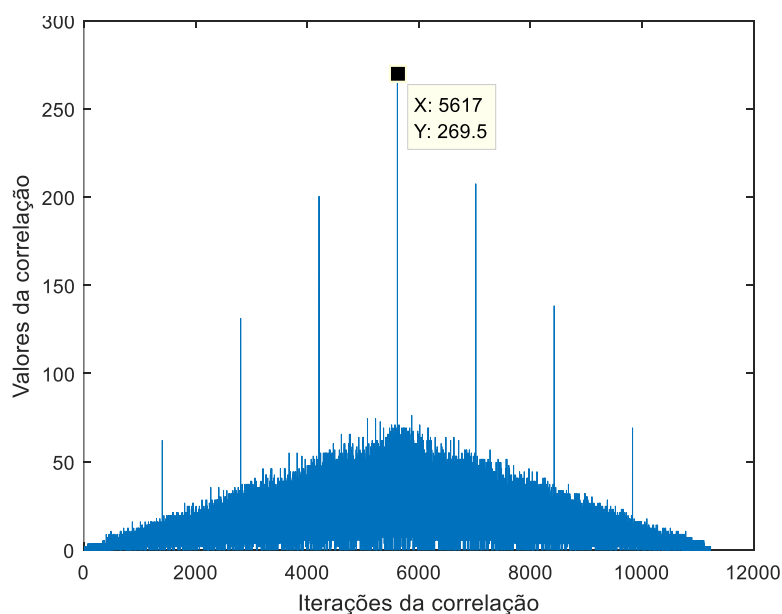
Figura 32 – *Flow graph* para os testes de detecção de erros inteiros de frequência.



Fonte: Acervo próprio.

Na Figura 33 podem ser observados os valores obtidos da correlação de um sinal sem erro inteiro.

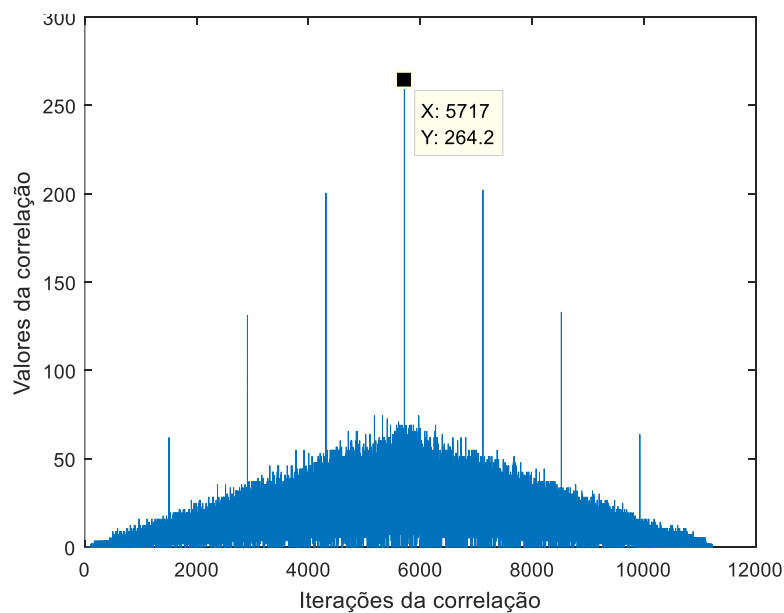
Figura 33 – Valores correlação cruzada (sem erro inteiro de frequência).



Fonte: Acervo próprio.

Ao inserir um erro de frequência equivalente a 100 vezes a distância entre as portadoras OFDM, ocorre um deslocamento de 100 amostras na correlação, como pode ser observado na Figura 34.

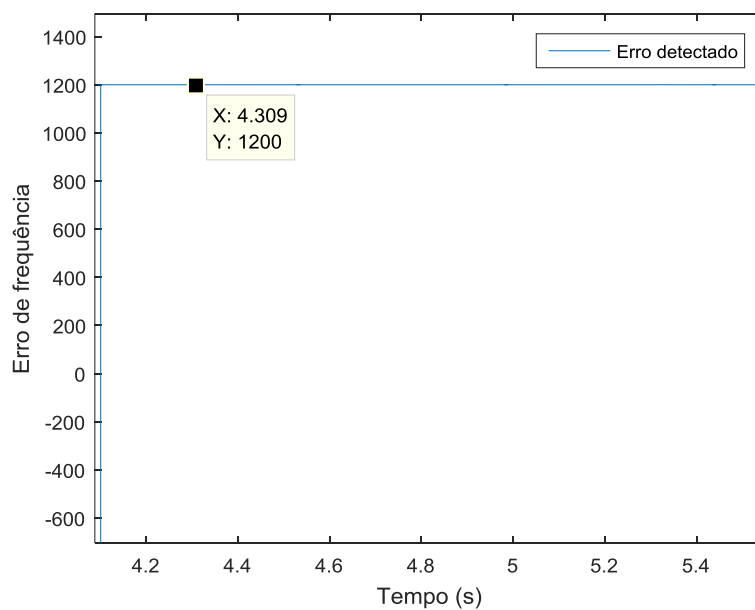
Figura 34 – Valores da correlação cruzada (com erro inteiro de frequência).



Fonte: Acervo próprio.

Na Figura 35 pode ser observada a detecção de um erro de frequência de 1200 Hz . Isto é, um erro inteiro mais um erro fracionário.

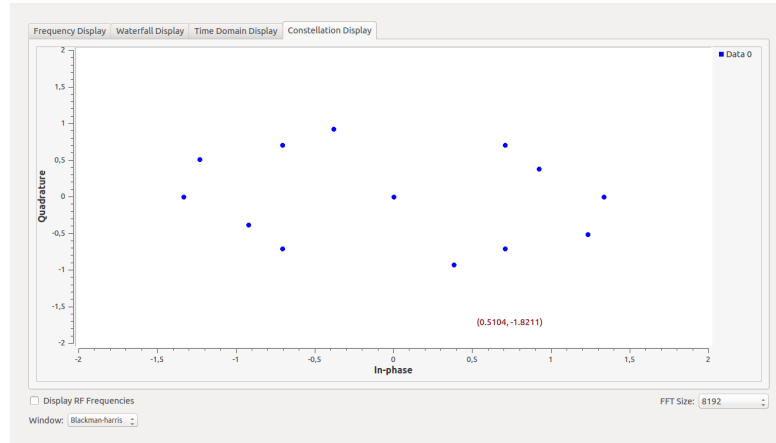
Figura 35 – Valor do erro de frequência (1200 Hz)



Fonte: Acervo próprio.

O efeito causado na constelação de um sinal ISDB-T devido um erro inteiro de frequência pode ser observado na Figura 36.

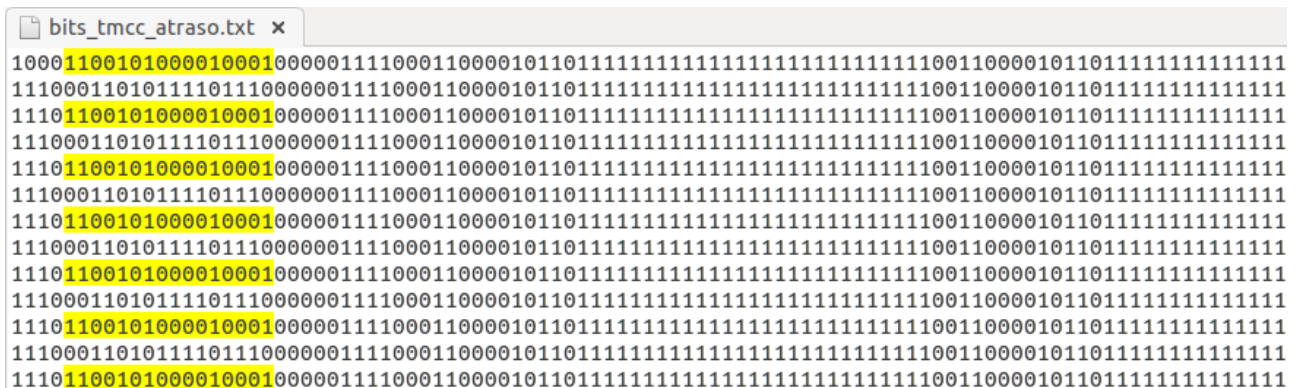
Figura 36 – Constelação de um sinal ISDB-T com erro inteiro de frequência.



Fonte: Acervo próprio.

Como resultado da detecção do início dos quadros OFDM, pode ser observada na Figura 37, os bits demodulados da TMCC.

Figura 37 – Bits demodulados da TMCC



Fonte: Acervo próprio.

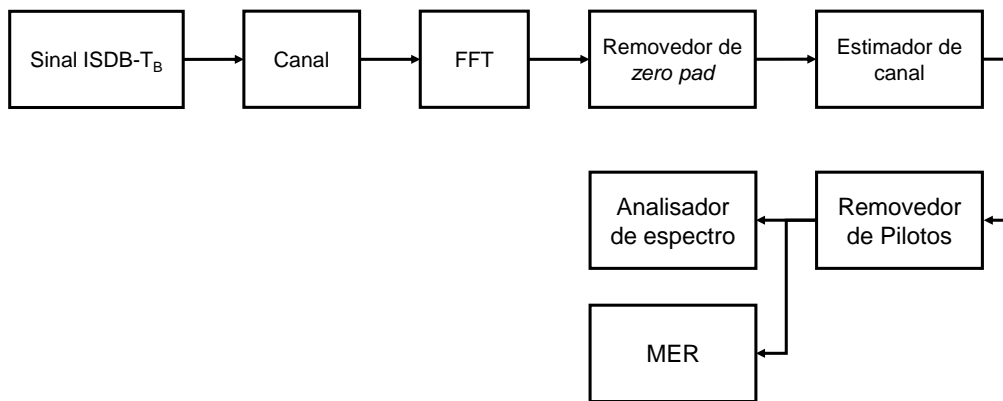
Pode-se notar a seqüência de 16 bits dos quadros ímpares (em amarelo), utilizadas para a sincronização. Note como os bits desse seqüência são invertidos a cada linha.

Cada linha da Figura 37 possui todos os 204 bits da TMCC de um quadro OFDM. Também é possível notar que o sinal recebido não estava sincronizado, pois o primeiro

bit de cada linha não é o *bit* de referência para demodulação diferencial, como mostra a Tabela 1.

Para a avaliação dos estimadores de canal implementados foi utilizado um canal de comunicação sem ruído AWGN e com um único caminho de multipercurso. Um sinal ISDB-T 64-QAM, no modo 3(8K) e com intervalo de guarda de 1/8 foi inserido nesse canal de comunicação. O sinal testado estava sincronizado e sem erros de frequência. O *flow graph* para esse teste pode ser observado na Figura 38. Nesse teste foi assumido um sincronismo perfeito de tempo, frequência e de quadro OFDM.

Figura 38 – *Flow graph* para os testes de estimação de canal.

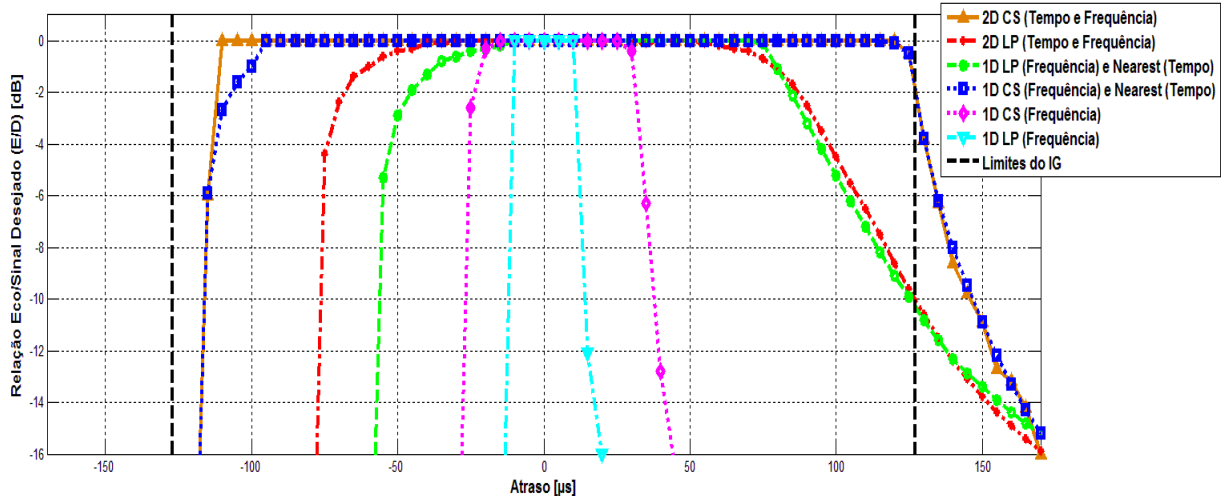


Fonte: Acervo próprio.

Nesse trabalho foram implementados os estimadores *Linear Piecewise* (LP) e *Cubic Spline* (CS). Cada um desses possui 3 variações, que são: interpolação somente no domínio da frequência (1D), interpolação no domínio do tempo e frequência (2D) e interpolação no domínio da frequência com interpolação *Nearest* no tempo (2D *Nearest*). Essa última interpolação utiliza os valores das pilotos SP do símbolo anterior para que a interpolação no domínio da frequência seja realizada.

A Figura 39 mostra curvas de relação eco por sinal desejado (E/D) em dB para obter valores de MER iguais ou maiores que 19 dB. Os valores de MER foram medidos na saída do bloco de estimação de canal.

Figura 39 – Relação E/D para diferentes estimadores de canal.



Fonte: Sapia e Akamine (2016).

A Equação 17 mostra como é calculado o limite do intervalo de guarda, isto é, o seu tempo de duração.

$$T_{IG} = T_s \times IG \quad (17)$$

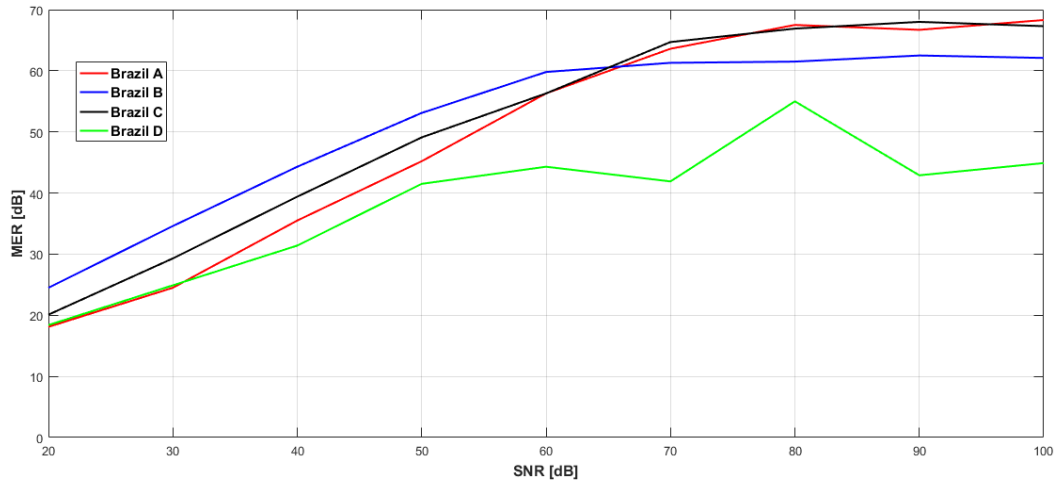
Isto é, o tempo de duração do intervalo de guarda é equivalente a multiplicação dos valores de tempo de duração de um símbolo OFDM pela razão desejada de intervalo de guarda. Portanto, como foi utilizado o modo 3 e a razão do IG é de 1/8, chega-se ao valor de $T_{IG} = 125\mu s$

Como pode ser observado, os estimadores 2D CS têm um desempenho superior aos estimadores 1D e 2D LP. Os estimadores 2D CS e 1D *Nearest* CS tiveram desempenhos próximos. Entretanto, o estimador 2D CS teve um comportamento levemente superior na presença de pré-eco. Esses estimadores obtiveram resultados próximos, pois nesse teste foi utilizado um canal estacionário. Os estimadores 1D tiveram resultados muito abaixo dos outros estimadores, pois a interpolação é linear e somente na frequência. Sem a estimação no tempo os valores obtidos sequer se aproximaram dos limites teóricos do intervalo de guarda.

Outro teste realizado foi do valor de MER em função do SNR, como pode ser observado na Figura 40. As curvas obtidas foram traçadas para diferentes configurações de

multipercurso. Essas configurações foram estabelecidas por (ITU, 2003).

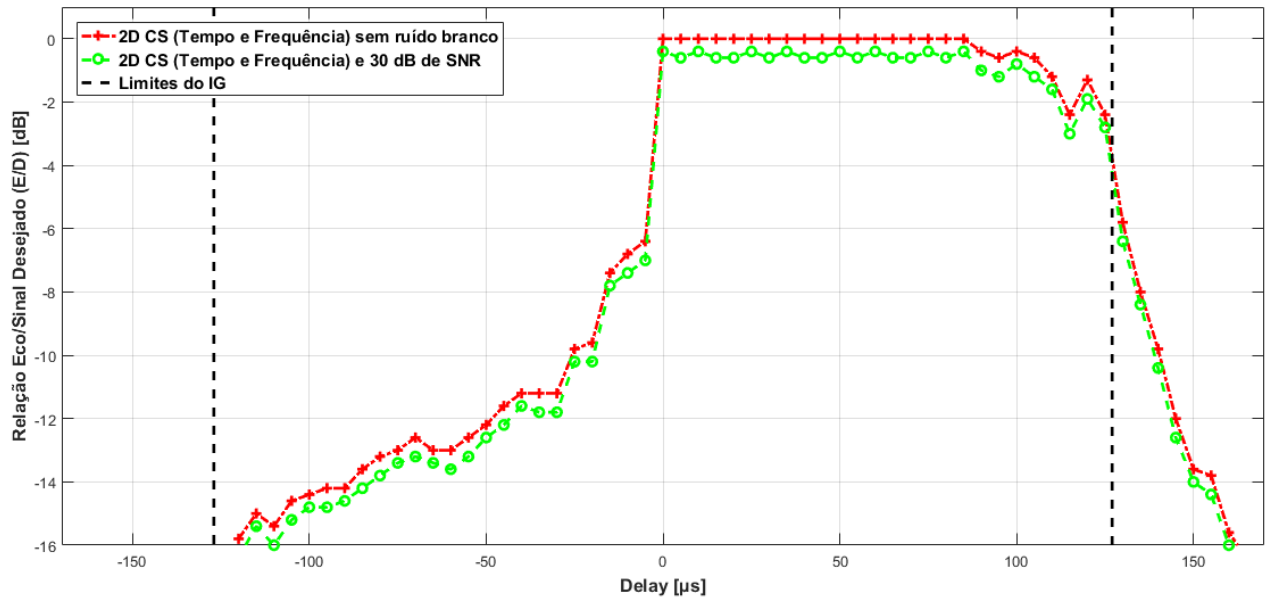
Figura 40 – Curva de MER em função do SNR.



Fonte: Acervo próprio.

Para verificar o comportamento do sistema como um todo, foi realizado um teste onde todos os blocos criados foram utilizados. Como canal de comunicação foi escolhido um erro de inteiro de frequência somado a um erro fracionário de frequência, no valor de, $1.0921kHz$, que equivale a um erro inteiro mais $100Hz$ de erro fracionário. Como multipercurso foi utilizada a mesma configuração do teste da Figura 39. Nesse teste apenas a estimação cúbica no tempo e na frequência foi testada, pois foi esta que obteve o melhor resultado, como verificado na Figura 39. Foram geradas duas curvas: uma sem ruído branco e outra com um valor de 30 dB de SNR. Os resultados podem ser observados na Figura 41.

Figura 41 – Relação E/D com todos os blocos interligados sem ruído branco e com ruído branco.



Fonte: Acervo próprio.

Como pode ser observado na 41, em relação ao resultado da Figura 39 o desempenho no pré-eco foi muito inferior. O pós-eco teve o resultado mais parecido com o da Figura 39, entretanto, ainda obteve valores inferiores. O que pode ser verificado também é que na presença de um SNR de 30 dB o desempenho foi inferior ao teste sem ruído, o que já era um resultado esperado.

5 CONCLUSÃO

Neste projeto foi realizada uma ampla pesquisa acerca da transmissão do sistema ISDB-T, o que possibilitou a compreensão dos estágios de recepção para esse sistema. Entretanto, as técnicas de sincronismo e estimação canal necessárias para a demodulação do sistema ISDB-T foram estudadas mais a fundo, possibilitando dessa maneira a implementação de um demodulador. Estudos sobre o GRC e sobre SDR também foram realizados para permitir as implementações realizadas.

Com o bloco para a realização de sincronismo temporal foi possível obter resultados da autocorrelação realizada nos símbolos OFDM para encontrar a posição inicial dos mesmos. Dessa forma, foi verificado que o bloco identifica corretamente o início dos símbolos. Além disso, foi possível verificar o efeito causado pela falta de sincronismo na constelação de um sinal ISDB-T, bem como a eficácia do sincronismo temporal. Ainda nesse bloco, foi verificada a capacidade de detecção de erros fracionários de frequência. Essa detecção apresentou certas limitações, como no caso erros fracionários com grandes oscilações na frequência. Entretanto, a correção do erro funciona normalmente se a detecção for correta.

Para testar o bloco de detecção de erros inteiros de frequência, foram gerados gráficos das correlações cruzadas. A partir dos gráficos foi possível verificar que o bloco é capaz de detectar corretamente o valor de erro inteiro presente no sinal. Também foi possível verificar o efeito causado na constelação do sinal na presença de um erro inteiro de frequência.

O bloco de sincronismo de quadro OFDM foi avaliado por meio dos *bits* da TMCC. Avaliando a sequência de *bits* foi verificado que o bloco inicia a sua transmissão de dados apenas quando o início de um quadro OFDM é detectado.

O desempenho da estimação de canal foi avaliada utilizando um canal de comunicação sem ruído AWGN e com um único canal de multipercurso. Ao verificar o MER do sinal recebido na saída do estimador para diferentes atrasos de multipercurso, foi possível traçar curvas E/D para as diferentes estimações implementadas. Por meio do gráfico gerado foi verificado que as estimações com interpolações cúbicas realizadas no tempo e na frequência alcançaram os melhores resultados.

Por meio deste projeto foi possível concluir que a utilização de técnicas de sincronismo temporal e de frequência permite que o sinal afetado pelo canal de comunicação seja de-

modulado da forma correta. Isto é, o receptor consegue identificar o início de cada símbolo OFDM no tempo, além de corrigir os desvios de frequência do sinal, assim permitindo que o sinal demodulado não apresente erros. Também foi possível verificar a importância dos estimadores de canal na recuperação de um sinal na presença de multipercurso.

5.1 Artigos publicados

1. “*Channel estimation evaluation in an ISDB-T system using GNU Radio*”, Conference on Electronics, Telecommunications and Computers, Lisboa, 2016. Autores: Thiago Sapia e Cristiano Akamine.

5.2 Trabalhos futuros

- Implementação de um controle para a realimentação dos erros de frequência
- Implementação de sincronismo de amostragem
- Implementação de blocos para a recuperação do TS.
- Implementação de técnicas mais eficazes de estimação de canal

REFERÊNCIAS BIBLIOGRÁFICAS

- Agência Nacional de Telecomunicações. *Resolução nº 625, de 11 de novembro de 2013. Aprova a atribuição, a Destinação e o Regulamento Sobre Condições de Uso de Radiofrequências na Faixa de 698 MHz a 806 MHz*. 2013.
- AI, B. Frequency offset estimation for OFDM in wireless communications. *IEEE Transactions on Consumer Electronics*, v. 50, n. 1, p. 73–77, feb 2004.
- AKAMINE, C. *Contribuições para distribuição, modulação e demodulação do sistema de TV digital ISDB-TB*. Tese (Doutorado) — Universidade Estadual de Campinas, 2011.
- ANALOG DEVICES. *RF Agile Transceiver*. [S.l.], nov. 2014. Rev. E.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 15601: Televisão digital terrestre - Sistema de transmissão*. Rio de Janeiro, 2008a.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 15602-1: Televisão digital terrestre - Codificação de vídeo, áudio e multiplexação - Parte 1: Codificação de vídeo*. Rio de Janeiro, 2008b.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 15602-2: Televisão digital terrestre - Codificação de vídeo, áudio e multiplexação - Parte 2: Codificação de áudio*. Rio de Janeiro, 2008c.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 15606-5: Televisão digital terrestre - Codificação de dados e especificações de transmissão para radiodifusão digital - Parte 5: Ginga-NCL para receptores portáteis - Linguagem de aplicação XML para codificação de aplicações*. Rio de Janeiro, 2008d.
- ASSOCIATION OF RADIO INDUSTRIES AND BUSINESSES. *ARIB STD-B31: Transmission System for Digital Terrestrial Television Broadcasting*. Japão: [s.n.], jul. 2003.
- BEEK, J.-J. van de et al. On channel estimation in OFDM systems. In: *1995 IEEE 45th Vehicular Technology Conference. Countdown to the Wireless Twenty-First Century*. [S.l.]: IEEE, 1995. v. 2, p. 815–819.

BRASIL. *Decreto nº52.795 de 31 de outubro de 1963*. Brasília, 1963. Diário Oficial da União, 12 de nov. 1963.

BRAUN, M. *Core concepts of GNU Radio*. 2014. Disponível em: <<http://gnuradio.org/redmine/projects/gnuradio/wiki/TutorialsCoreConcepts>>.

DIGITAL BROADCAST EXPERTS GROUP. *ISDB-T Adopting Countries*. [S.l.], 2015. Disponível em: <<http://www.dibeg.org/world/world.html>>.

ETTUS RESEARCH. *Specification Sheet: USRP B200/210*. 2014. Disponível em: <https://www.ettus.com/content/files/b200-b210_spec_sheet.pdf>.

HILBURN, B. *What is GNU Radio and why do I want it?* 2015. Disponível em: <<http://gnuradio.org/redmine/projects/gnuradio/wiki/WhatIsGR>>.

IBGE. *Acesso à Internet e à Televisão e Posse de Telefone Móvel Celular para Uso Pessoal*. Rio de Janeiro, 2015.

INTERNATIONAL TELECOMMUNICATION UNION. *ITU-R BT.2035: Guidelines and techniques for the evaluation of digital terrestrial television broadcasting systems*. [S.l.], 2003.

KANG, S. G.; HA, Y. M.; JOO, E. K. A comparative investigation on channel estimation algorithms for ofdm in mobile communications. *IEEE Transactions on Broadcasting*, v. 49, n. 2, p. 142–149, June 2003. ISSN 0018-9316.

KELLER, T. et al. Orthogonal frequency division multiplex synchronization techniques for frequency-selective fading channels. *IEEE Journal on Selected Areas in Communications*, v. 19, n. 6, p. 999–1008, jun 2001.

LARROCA, F. et al. gr-isdbt: An ISDB-T 1-segment receiver implementation on GNU Radio. In: *2015 Latin American Computing Conference (CLEI)*. [S.l.]: IEEE, 2015. p. 1–8.

MA, Z. et al. Low-complexity channel estimation for ISDB-T over doubly-selective fading channels. In: *2013 International Symposium on Intelligent Signal Processing and Communication Systems*. [S.l.]: IEEE, 2013. p. 114–118.

- MACIEL, Y. P. et al. ISDB-Tb transmission in software-defined radio. In: *2015 7th IEEE Latin-American Conference on Communications (LATINCOM)*. [S.l.]: IEEE, 2015. p. 1–6.
- MITOLA, J. The software radio architecture. *IEEE Communications Magazine*, v. 33, n. 5, p. 26–38, may 1995.
- MITOLA, J. Software radio architecture: a mathematical perspective. *IEEE Journal on Selected Areas in Communications*, v. 17, n. 4, p. 514–538, apr 1999.
- PADERNA, R.; HIGASHINO, T.; OKADA, M. Improved channel estimation for ISDB-T using Modified Orthogonal Matching Pursuit over fractional delay TU6 channel. In: *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*. [S.l.]: IEEE, 2014. p. 1–5.
- RONDEAU, T. *Out-Of-Tree Modules*. 2016. Disponível em: <<http://gnuradio.org/redmine/projects/gnuradio/wiki/OutOfTreeModules>>.
- SAPIA, T.; AKAMINE, C. Channel estimation evaluation in an ISDB-T system using GNU Radio. In: *Conference on Electronics, Telecommunications and Computers*. Lisboa: [s.n.], 2016. p. 2.
- TAKADA, M.; SAITO, M. Transmission System for ISDB-T. *Proceedings of the IEEE*, IEEE, v. 94, n. 1, p. 251–256, jan 2006.
- VACHHANI, K.; MALLARI, R. A. Experimental study on wide band fm receiver using gnuradio and rtl-sdr. In: *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. [S.l.: s.n.], 2015. p. 1810–1814.