

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELÉTRICA
CURSO DE ENGENHARIA ELÉTRICA

JEFERSON GUSTAVO DE ANDRADE

**FRAMEWORK PARA DESENVOLVIMENTO DE
ALGORITMOS PARA ANÁLISE DE SINAIS COM
PROBLEMAS DE QUALIDADE DE ENERGIA**

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO

2019

JEFERSON GUSTAVO DE ANDRADE

**FRAMEWORK PARA DESENVOLVIMENTO DE
ALGORITMOS PARA ANÁLISE DE SINAIS COM
PROBLEMAS DE QUALIDADE DE ENERGIA**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso de Engenharia Elétrica do Departamento Acadêmico de Elétrica – DAELE – da Universidade Tecnológica Federal do Paraná – UTFPR, Câmpus Pato Branco, como requisito parcial para obtenção do título de Engenheiro Eletricista.

PATO BRANCO

2019

TERMO DE APROVAÇÃO

O Trabalho de Conclusão de Curso intitulado **FRAMEWORK PARA DESENVOLVIMENTO DE ALGORITMOS PARA ANÁLISE DE SINAIS COM PROBLEMAS DE QUALIDADE DE ENERGIA** do acadêmico **Jeferson Gustavo de Andrade** foi considerado **APROVADO** de acordo com a ata da banca examinadora N° 237 de 2019.

Fizeram parte da banca examinadora os professores:

Prof. Dr. Giovanni Alfredo Guarneri
Orientador

Prof. Dr. Cesar Claire Rafael Torrico
Convidado 1

Prof. Cesar Augusto Portolann
Convidado 2

A Ata de Defesa assinada encontra-se na Coordenação do Curso de Engenharia Elétrica

DEDICATÓRIA

A finalização deste trabalho é dedicado à minha família, meu orientador e as pessoas que estiveram ao meu lado nessa caminhada. Dedico e agradeço à Deus por todas as promessas cumpridas.

EPÍGRAFE

*"Coisas que os olhos não viram, nem os ouvidos não
ouviram, nem o coração humano imaginou, tais são as
Bençãos que o Senhor preparou para aqueles que o amam."
(1 Corintios 2-9)*

AGRADECIMENTOS

Esse trabalho não seria possível sem a orientação do professor Giovanni. Agradeço pelas inúmeras vezes que teve paciência em lembrar o propósito maior de nosso trabalho.

Ter motivação diária no período da graduação não seria possível sem a companhia de meus amigos, Mayara, Fabio e Ricardo. Os momentos de descontração e os cafés foram fundamentais ao longo destes cinco anos.

Realizar este curso só foi possível graças aos incentivos de meus pais, seus sacrifícios e renúncias pra sempre serão lembrados. A minha namorada Alana, que diariamente ouvia meus medos e sempre me aconselhou e me acalmou. Pelo dom da vida, agradeço à Deus por todas as experiências, tendo cumprido suas promessas em minha vida.

RESUMO

ANDRADE, Jeferson Gustavo de. FRAMEWORK PARA DESENVOLVIMENTO DE ALGORITMOS PARA ANÁLISE DE SINAIS COM PROBLEMAS DE QUALIDADE DE ENERGIA. 53 f. Trabalho de Conclusão de Curso - Curso de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, 2019.

A crescente busca pelo diagnóstico de eventos de Qualidade de Energia Elétrica (QEE) por meio de ferramentas matemáticas aplicadas em métodos computacionais com intuito de proporcionar a observação e a correção de distúrbios na entrega de energia elétrica, tem sido uma evolução na área de estudo de processamento de sinais. Seja por meio de simuladores ou equipamentos capazes de aferir eventos de QEE, há uma grande variedade de fontes capazes de fornecer esses dados para análise. Por serem fontes diferentes, também há entre elas uma diferença no padrão e formato de seus dados. Como consequência, realizar a análise desses dados significa realizar sua decodificação de modo que adquiram uma estrutura compatível de ser trabalhada por *softwares* tradicionais da Engenharia. O objetivo deste trabalho consiste no estudo e desenvolvimento de um *framework* elaborado no *software* Matlab que auxilie pesquisadores da área de QEE no desenvolvimento de algoritmos para análise de sinais provenientes de simuladores ou medidores, livrando-os da necessidade de realizar as etapas anteriores de compatibilização dos dados. O *framework* é capaz de fazer a leitura de arquivos com medidas provenientes de equipamentos analisadores de QEE no formato COMTRADE, além da leitura de outros dois tipos de formatos, proveniente dos simuladores ATP e Simulink, largamente utilizados em simulações no ramo da Engenharia Elétrica. Após realizar a leitura dos dados, inicia-se o pré-processamento, que identifica anormalidades, realiza manipulações em variáveis e a adaptação dos dados para um formato padrão em uma estrutura de entrada passível de ser processada por diversos algoritmos de análise de QEE. Para demonstrar o funcionamento do *framework*, este trabalho apresenta um *template* com um algoritmo baseado na Transformada *Wavelet*. Após o processamento pelo algoritmo, os dados ficam salvos em uma estrutura de saída, sendo utilizada para apresentar os resultados em formatos gráficos e estatísticos, a fim de fornecer a melhor experiência do usuário na observação de seus resultados.

Palavras-chave: *Framework*; Processamento de sinais; Qualidade de Energia Elétrica.

ABSTRACT

ANDRADE, Jeferson Gustavo de. ALGORITHM DEVELOPMENT FRAMEWORK FOR POWER QUALITY PROBLEMS ANALYSIS. 53 f. Trabalho de Conclusão de Curso - Curso de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, 2019.

The growing search for the diagnosis of Electric Power Quality events by means of mathematical tools applied in computational methods with the purpose of providing observation and correction of disturbances in the delivery of electric energy, has been a growing area in the study of signal processing. Whether through simulators or equipment capable of measuring QEE events, there is a wide variety of sources capable of providing this data for analysis. Because they are different sources, there is also a difference in the pattern and format of their data. As a consequence, performing the analysis of these data means performing its decoding so that it acquires a compatible structure to be worked by traditional softwares of Engineering. The objective of this work is the study and development of a framework developed in the Matlab software that helps researchers in the QEE area in the development of algorithms for analyzing signals from simulators or meters, freeing them from the need to perform the previous steps of data compatibility. The framework is capable of reading files with measurements from QEM meters in COMTRADE format, as well as reading two different types of formats, from the simulators ATP and Simulink, widely used in simulations in the field of Electrical Engineering. After reading the data, pre-processing starts, which identifies abnormalities, performs manipulations on variables and adapts the data to a standard format in an input structure that can be processed by several QEE analysis algorithms. To demonstrate the operation of the framework, this work presents a template with an algorithm based on the Wavelet Transform. After processing by the algorithm, the data is saved in an output structure and used to present the results in graphical and statistical formats, in order to provide the best user experience in observing its results.

Keywords: Signal Processing; Frameworks; Power Quality.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de estrutura da categoria CFG do arquivo COMTRADE.	19
Figura 2 – Exemplo de arquivo CSV.	21
Figura 3 – Plano Escala x Tempo da análise por <i>Wavelet</i>	22
Figura 4 – Níveis de decomposição da Transformada <i>Wavelet</i>	24
Figura 5 – Estrutura do <i>framework</i>	25
Figura 6 – Arquivo COMTRADE analisado pelo <i>framework</i>	32
Figura 7 – Caixa de dialogo do parâmetro <i>Comparative</i>	34
Figura 8 – Resultados da janela <i>Comparative</i>	34
Figura 9 – Coeficientes de detalhes da WT do canal 5.	35
Figura 10 – Coeficientes de aproximação da WT do canal 5.	36
Figura 11 – Arquivo .CSV de análise do <i>framework</i>	37
Figura 12 – Janela <i>Comparative</i> do arquivo CSV.	39
Figura 13 – Detalhes do canal 1 do arquivo CSV com nível de decomposição <i>wavelet 7</i>	40
Figura 14 – Aproximação nível 7 do canal X0001A.	41
Figura 15 – Sistema de análise do arquivo Simulink.	42
Figura 16 – Janela <i>Comparative</i> do <i>framework</i> para os arquivos do Simulink.	43
Figura 17 – Coeficientes de detalhes do canal 1 para arquivos do Simulink.	44
Figura 18 – Sistema de análise do arquivo Simulink.	45

LISTA DE TABELAS

Tabela 1 – Exemplo de arquivo semelhante ao ATP.	20
--	----

LISTA DE ABREVIATURAS E SIGLAS

QEE	Qualidade de Energia Elétrica
WT	Transformada <i>Wavelet</i>
SEP	Sistema Elétrico de Potência
PQ	<i>Power Quality</i> - Qualidade de Energia
ANEEL	Agência Nacional de Energia Elétrica
FT	<i>Fourier Transform</i> - Transformada de Fourier
STFT	<i>Short Time Fourier Transform</i> - Transformada de Fourier de curto tempo
SI	Sistema Internacional
CSV	<i>Comma-Separated Values</i> - Valores Separados por Vírgula
COMTRADE	<i>Common Format for Transient Data Exchange</i> - Formato Comum para troca de Dados de Transitórios
CWT	<i>Continuous Wavelet Transformer</i> - Transformada Contínua Wavelet
DWT	<i>Discrete Wavelet Transformer</i> - Transformada Discreta Wavelet
MRA	<i>Multi-Resolutional Analysis</i> - Análise Multirresolucional
db1	<i>Daubechies 1</i>

LISTA DE SÍMBOLOS

ψ *Wavelet-mãe*

SUMÁRIO

	Sumário	12
1	INTRODUÇÃO	13
1.1	OBJETIVO GERAL	15
1.2	OBJETIVOS ESPECIFICOS	15
1.3	ORGANIZAÇÃO DO TRABALHO	15
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	FRAMEWORKS	17
2.2	ARQUIVOS DE LEITURA DO FRAMEWORK	18
2.2.1	Arquivos COMTRADE	18
2.2.2	Arquivos CSV	20
2.2.3	Arquivos Simulink	21
2.3	TRANSFORMADA <i>WAVELET</i>	21
2.4	CONSIDERAÇÕES FINAIS	24
3	METODOLOGIA	25
3.1	LEITURA DOS DADOS PARA ANÁLISE	26
3.2	PRÉ-PROCESSAMENTO	27
3.3	DESENVOLVIMENTO DE ALGORITMOS	28
3.4	FORMATAÇÃO E APRESENTAÇÃO DOS RESULTADOS	29
3.5	CONSIDERAÇÕES FINAIS	30
4	RESULTADOS	32
4.1	ARQUIVOS COMTRADE	32
4.2	ARQUIVOS CSV	36
4.3	ARQUIVOS SIMULINK	41
4.4	CONSIDERAÇÕES FINAIS	46
5	CONCLUSÕES	48
	REFERÊNCIAS	51

1 INTRODUÇÃO

A Engenharia de *Software* tem por objetivo desenvolver sistemas com melhor qualidade, a um custo de menor tempo e esforço. Nisso, cita-se o desenvolvimento de *softwares* com possibilidade de reuso (GIMENES; HUZITA, 2005). Essa utilização de programas já desenvolvidos e depurados, torna o tempo de desenvolvimento, de testes e as possibilidades de erros na nova produção menores (REINALDO et al., 2003). Com um subsistema extensível, para um conjunto de serviços relacionados, constroem-se *frameworks* que favoreçam o usuário no reuso da análise, projeto e código.

Dentro da literatura há diversas definições para o conceito de *framework*. Dentre elas, a de que um *framework* é um conjunto de objetos que colaboram com o objetivo de atender um conjunto de responsabilidades para uma aplicação específica. Em outras palavras, a elaboração de um *framework* para uma aplicação consiste em reunir processos que permitam ao usuário, realizar o objetivo final que se propôs ou apenas adaptar processos já existentes. Esses *frameworks* podem variar desde interfaces avançadas, oferecendo uma enorme quantidade de informações e configurações até rotinas simples, geradas através de linhas de comando.

Atualmente, os *frameworks* estão inseridos nas mais diversas áreas, em vários campos de estudos. Em (HOPP et al., 2009) é apresentado um *framework* avançado, com uma interface gráfica intuitiva desenvolvida pelo *software* Matlab para observação de resultados de tomografias por ultrassom. O *framework* utiliza os mais de 20 GB de dados para uma única imagem produzida por aproximadamente 200 transdutores de ultrassom em uma interface gráfica usando GUI do Matlab. Com isso, é eliminada a complexa relação entre os dados originais e reconstruídos para compreensão dos cientistas e principalmente de novos estudantes, permitindo que desenvolvam suas análises com maior velocidade e facilidade (HOPP et al., 2009).

Em áreas de concentração da Engenharia elétrica, pode-se entender a viabilidade de *frameworks* para as áreas de processamento de sinais. Algoritmos de análise de sinais com eventos de QEE rotineiramente precisam passar por etapas anteriores, de preparação, conversão e adaptação desses dados, para que exista uma compatibilidade com a linguagem da plataforma onde se desenvolve o algoritmo. Pode-se pensar que a existência de um estrutura generalista que realize tais tarefas, permitindo essa compatibilização e apresentação dos resultados do algoritmo, permitiria uma certa agilidade em seu desenvolvimento, pois o usuário realizaria apenas modificações a fim de aproximar aquele código generalista do objetivo proposto. Tais considerações são positivas tendo em vista de que essas técnicas capazes de avaliar índices de qualidade de Energia Elétrica (QEE) submetidas a eventos na distribuição e transmissão

e com isso realizar a prevenção de uma série de problemas originados na presença de eventos de QEE (GRANADOS-LIEBERMAN et al., 2011). As técnicas que fazem o monitoramento de energia podem fornecer índices de qualidade, realizar detecções temporais no caso de falhas, prevenindo o Sistema Elétrico de Potência (SEP).

A importância da QEE é indiscutível para os meios de estudo e para a qualidade de vida de um país. E a preocupação torna-se cada vez mais evidente com a demanda das indústrias que incorporam cada vez mais equipamentos eletronicamente controlados, que são sensíveis quanto a flutuação de frequência e tensão (MELO, 2008). Além de atender a qualidade da energia fornecida, tal sincronização entre a produção de energia e o consumo em tempo real é um dos grandes desafios na sociedade atual (INSTITUTO ASCENDE BRASIL, 2014). Por isso, Agência Nacional de Energia Elétrica (ANEEL), responsável pelo monitoramento do sistema elétrico brasileiro, visa padronizar os parâmetros de QEE. Sendo eles: tensão em regime permanente, fator de potência, nível de distorção harmônica, desequilíbrio de tensão, variações de curta duração e variação de frequência, publicado no documento Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional (PRODIST), que estabelece os procedimentos relativos a QEE (ELÉTRICA, 2013). Esses então, são a base do estudo de qualidade de energia.

A QEE, traduzida do inglês *Power Quality* (PQ) é analisada por meio da combinação da qualidade da corrente e da tensão, e essas individualmente são relacionadas com o desvio de seus valores ideais de frequência, magnitude e forma de onda. Valores elevados dessas variações podem trazer sérios problemas para a rede elétrica e para os usuários (GARCÍA, 2012). Como exemplos de eventos de QEE têm-se: afundamento de tensão, que é a redução de 10 % a 90 % de seu valor eficaz, classificado como momentâneo caso sua duração seja superior ou igual a um ciclo e inferior a 3 segundos, e caso sua duração seja maior do que 3 segundos e inferior a 3 minutos, passa a ser classificada como afundamento de tensão temporária; elevação de tensão, representada pelo aumento do valor eficaz acima de 1,1 p.u, podendo ser momentânea caso sua duração seja superior ou igual a um ciclo e inferior ou igual a 3 segundos, e classificada como temporária caso seja superior a 3 segundos e inferior a 3 minutos; ruído, definido como uma perturbação aleatória superposta ao sinal da grandeza elétrica e geralmente com frequência entre 0 a 2 MHz; distorções harmônicas, periódicas e originadas de combinações entre as ondas fundamentais e seus múltiplos inteiros; interrupções momentâneas, caracterizada pela perda de potência total de duração menor que 2 segundos; entre outros (HADDAD, 2004)(MANIKANDAN; SAMANTARAY; KAMWA, 2015). O estudo da QEE tenta identificar e classificar esses problemas. Por isso, as companhias de energia elétrica precisam estar aptas e atender as novas tecnologias no mercado atual, prolongando assim a vida útil de equipamentos sensíveis ligados ao sistema elétrico (OLIVEIRA, 2007).

Para poder avaliar a QEE, técnicas e metodologias capazes de diagnosticar eventos na distribuição e transmissão de energia têm se desenvolvido por setores da área de pesquisa de concentração. Com isso pode-se realizar a prevenção de uma série de problemas originados na presença de eventos de QEE (GRANADOS-LIEBERMAN et al., 2011), auxiliando no planejamento estratégico das demandas futuras por equipamentos e sistemas ligados ao SEP.

1.1 OBJETIVO GERAL

O trabalho consiste no projeto e desenvolvimento de um *framework* no *software* Matlab que auxilie e acelere o processo de desenvolvimento de algoritmos propícios na obtenção de índices e resultados em eventos de QEE, de sinais provenientes de aparelhos medidores de QEE e de simuladores com os formatos COMTRADE, CSV e Simulink, respectivamente. E posteriormente apresentar esses resultados por meio de gráficos ajustáveis por parâmetros de interesse do usuário.

1.2 OBJETIVOS ESPECIFICOS

- Caracterizar e apresentar os formatos de arquivos COMTRADE, CSV e Simulink, passíveis de serem lidos pelo *framework*.
- Apresentar a estrutura geral e os módulos que compõem o *framework*, dissertando o processo que ocorre em cada etapa.
- Apresentar matematicamente a definição da *Wavelet Transform* (WT)
- Desenvolver um *template* utilizando algoritmo desenvolvido com teoria da WT.
- Realizar as simulações e avaliar os resultados de cada um dos arquivos analisados pelo *template* desenvolvido.
- Apresentar conclusões e indicativos de implementações que impulsionem a viabilidade desse *framework* para a área de processamento de sinais.

1.3 ORGANIZAÇÃO DO TRABALHO

A organização deste trabalho foi realizada da seguinte forma. O Capítulo 1 apresentou uma introdução aos *frameworks*, exemplificando com uma aplicação em análises de ultrassom e na reconstrução de imagens, também apresentou de forma sucinta os problemas envolvidos na área de QEE, e os benefícios que os *frameworks* podem trazer para a área de processamento de sinais. No Capítulo 2, é feita a fundamentação teórica deste trabalho, apresentando teorias que envolvem os *frameworks*,

além da apresentação dos arquivos de leitura, sua estruturação e considerações para correta utilização neste trabalho. Nesse mesmo capítulo, apresenta-se de modo sucinto a teoria da WT, mais especificamente a análise multirresolucional (MRA). Em seguida, no Capítulo 3, é apresentada a metodologia adotada neste trabalho, demonstrando o estudo levantado a fim de elaborar a melhor estrutura do *framework*. No Capítulo 4, são apresentados os resultados do *framework* na análise dos três tipos de arquivos com o *template* da WT. Por fim, o Capítulo 5 apresenta as principais conclusões e considerações que proporcionem a melhor usabilidade e aplicabilidade do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar a base teórica deste trabalho. Primeiro, é apresentado o conceito de *frameworks*, suas características, dificuldades e classificação. Em seguida, são apresentados os três formatos de arquivos provenientes das fontes passíveis de serem interpretadas por este *framework*, e uma breve introdução a transformada *Wavelet* voltada para análise de sinais com eventos de QEE. Por último são apresentadas as considerações finais com as principais conclusões deste capítulo.

2.1 FRAMEWORKS

Sabe-se que o processo de construção de *softwares* continua demandando complexidade e é um processo relativamente oneroso (LEITE, 2006). Nesse sentido, projetistas de *softwares* mais experientes sabem que nem todo problema deve ser resolvido a partir de uma nova criação, mas reutilizar soluções já implementadas e que alcançaram resultados positivos (GAMMA et al., 1995). No âmbito da computação, a *Engenharia de Software* tem buscado estabelecer critérios para definir e projetar soluções que possam ser reutilizadas em diferentes momentos (PREE et al., 1995). Nesse sentido, surge a compreensão de *frameworks*, onde não se busca reutilizar simples componentes de *software*, mas *subsistemas*, que contribui para um grau de aumento da reutilização e para a melhor qualidade do produto.

A definição de *framework* deve ser entendida como a de uma aplicação semi-completa, reutilizável que pode ser especializada para produzir novas aplicações (JOHNSON; FOOTE, 1988). Com isso, o usuário pode adaptar ao programa suas necessidades, poupar tempo e diminuir a probabilidade de erros no desenvolvimento de programas do zero. Com uma estrutura inicial pronta, de uma aplicação generalista, ele pode ser customizado por um desenvolvedor para tomar características para aplicações mais específicas. Eliminando o oneroso tempo de construção, não se preocupando com etapas preliminares como pré-processamento, adaptação ou manipulação de dados. Para exemplificar, suponha o caso de um usuário que realizou experimentos em um dado simulador e deseje apenas visualizar os dados por meio de gráficos e ilustrações. Caso existisse um *framework* capaz de realizar toda aquisição de dados, pré-processar e posteriormente realizar a plotagem, certamente evitaria com que o usuário se preocupasse em desenvolver toda essa etapa novamente, muito provavelmente seria necessário realizar algumas modificações ou adaptações, entretanto, com um trabalho e risco consideravelmente menor. Mostrando, novamente o poder e utilidade dos *frameworks*.

A complexidade da estrutura dos *frameworks* é notória devido ao grande número de funções interligadas internamente ao programa. Por isso, além do conhecimento aprofundado por parte do desenvolvedor que irá utilizar o *framework* tanto sobre a linguagem quanto a estrutura, é crucial para que outros usuários possam compreender e utilizá-lo em suas pesquisas. Por isso, a documentação por parte do desenvolvedor do *framework* é um passo necessário, e deve detalhar o máximo possível dos processos considerados no desenvolvimento.

2.2 ARQUIVOS DE LEITURA DO FRAMEWORK

Esta seção apresenta os formatos de arquivos que o *framework* proposto neste trabalho estará habilitado para realizar a leitura, sendo eles o formato COMTRADE, que é proveniente de analisadores qualidade de energia, e os formatos CSV e Simulink, que são dados provenientes de simuladores. Esta seção tem o intuito de trazer de maneira sucinta, considerações e explicações sobre os formatos desses arquivos. Como nota, alguns caracteres possuem fontes diferentes no sentido de destacar as estruturas de arquivos dedicados como por exemplo dos arquivos `Header`, `Configuration`, `Date` e `Information` do COMTRADE, entre outros.

2.2.1 ARQUIVOS COMTRADE

A norma COMTRADE (IEEE *Standard Common Format for Transient Data Exchange*) regida pela norma IEEE C37.111-1991 (SOCIETY, 1999) define um formato padrão para os arquivos de dados e demais arquivos necessários para representar as faltas no SEP. Esse padrão é largamente empregado por fabricantes de registradores digitais de sinais, relés e aparelhos de subestações. Isso possibilita compatibilizar arquivos de registros oscilográficos gerados por elementos na rede do sistema elétrico (ACCIOLY, 2016). Diferentemente da maneira com que alguns *softwares* de simulações ou outros registradores apresentam seus resultados em uma única planilha ou documento, o formato COMTRADE possui quatro categorias de documento de leitura, sendo eles:

1. `Header`: com extensão `.HDR` é um texto ASCII para armazenamento de informações textuais, para que o usuário possa compreender melhor as condições de registro. Na pasta `Header` pode conter a descrição do SEP antes do distúrbio, nome da estação, identificação da linha, transformador, reator, entre outros elementos que classificam todo o SEP em que está inserido.
2. `Configuration`: é a pasta a ser lida por um programa específico de computador, ela tem informações em que esse programa irá ordenar a pasta `Date`. Ela também está no formato ASCII e contém o número, nome e tipos de canais, as unidades

de medida, frequência da linha, data e tempo do início e do fim em que ocorre o registro.

3. *Date*: com a extensão *.DAT*, essa pasta contém o valor de cada amostra do canal medido. O ASCII do arquivo *DATE* são divididos em linhas e colunas, onde as linhas correspondem a um canal específico e as colunas, separadas por vírgulas, insere os valores analógicos mensurados.
4. *Information*: é uma pasta opcional contendo informações extras, usado para anotações e lembretes de fácil acesso pelos usuários.

Na Figura 1 é apresentada a estrutura do arquivo *Configuration* no formato *.CFG* que é o arquivo a ser analisado por programas de computador específicos para leitura de arquivos.

```

Condie,518,1999
12,6A,6D
1,Popular Va-g,,kV, 0.14462,0.0000000000,0,-2048,2047,2000,1,P
2,Popular Vc-g,,kV, 0.14462,0.0000000000,0,-2048,2047,2000,1,P
3,Popular Vb-g,,kV, 0.14462,0.0000000000,0,-2048,2047,2000,1,P
4,Popular Ia,,A,11.5093049423,0.0000000000,0,-2048,2047,1200,5,P
5,Popular Ib,,A,11.5093049423,0.0000000000,0,-2048,2047,1200,5,P
6,Popular Ic,,A,11.5093049423,0.0000000000,0,-2048,2047,1200,5,P
1,Va over,,,0
2,Vb over,,,0
3,Vc over,,,0
4,Ia over,,,0
5,Ib over,,,0
6,Ic over,,,0
60
1
8000.000,885
11/07/1995,17:38:26.663700
11/07/1995,17:38:26.687500
ASCII
1

```

Figura 1 – Exemplo de estrutura da categoria CFG do arquivo COMTRADE.

Fonte: Adptado de SOCIETY,1999.

Pela Figura 1, a primeira linha apresenta o nome e número da estação de leitura, e a versão do arquivo (1999). Na segunda linha pode-se observar a indicação de 12 canais existentes no arquivo. Sendo 6 canais analógicos (6A), e outros seis digitais (6D). Nas linhas que seguem, são apresentados os canais analógicos, sendo que os três primeiros canais (linhas) classificam a grandeza como tensão, por meio da unidade de grandeza, indicada por kilovolts (kV), e as outras três linhas como sendo corrente, representada pela grandeza Ampere (A). Os seis canais restantes, são os canais digitais, que não possuem registro de medida. De maneira geral, grande parte

dos equipamentos ainda realiza as leituras por valores analógicos, sendo poucas as aplicações de leituras digitais.

2.2.2 ARQUIVOS CSV

O arquivo *Comma-Separated Values* (CSV) é uma implementação particular de arquivos separados por um delimitador, que usa a vírgula e a quebra de linha para separar os valores (SHAFRANOVICH, 2005). Esse é um dos formatos que o programa ATP DRAW (*Alternative Transient Program*) tem como extensão para salvar seus dados. O *software* ATP é dedicado para estudos eletromagnéticos. Ele permite uma modelagem que reproduz o mais fiel possível os eventos esperados na prática. Nele, pode-se realizar a simulação de redes polifásicas através da matriz de admitância de barras. Os resultados do ATP são calculados a intervalos de tempos discretos. O programa permite a representação de não linearidades, elementos com parâmetros concentrados, elementos com parâmetros distribuídos, chaves, transformadores, reatores, entre outros (BERNARDON ROBERTO RECH, 2005).

De maneira formal, o CSV é um formato de dados em que as suas colunas são separadas por caracteres de vírgulas, e as linhas que são separadas por quebras de linha. Como consequência, caracteres especiais, como vírgulas, entre outros deve ser inserido entre aspas. A fim de ilustrar essas conclusões, abaixo, na Tabela 1 é apresentado o modelo mais comum, onde os valores são separados por linhas e colunas. Essa mesma tabela pode ser representada no formato CSV, conforme é apresentado na Figura 2.

Tabela 1 – Exemplo de arquivo semelhante ao ATP.

Time	Voltage	Voltage	Current
Second	X0001A	X0001B X0002A	X0001C
0.000	10000	-10000	150
1.000	9500	-9500	102

A Tabela 1 pode ser representada em CSV conforme mostra a Figura 2.

```
Time,Voltage,Voltage,Current
Second,X0001A,X0001B,X0001C
,,X0002A,
0.000,10000,-10000,150
1.000,9500,-9500,102
```

Figura 2 – Exemplo de arquivo CSV.

Fonte: Autoria própria.

2.2.3 ARQUIVOS SIMULINK

O Simulink é uma ferramenta de simulação integrada do *software* Matlab, que permite modelar, simular e analisar sistemas dinâmicos. Permitindo que uma grande variedade de sistemas possam ser projetados e testados, seja na área de sistemas de controle, processamento de sinais, sistemas de comunicação, entre muitos outros. Por meio da interface gráfica do Simulink, os modelos de análise são criados sob a forma de diagramas de blocos que podem representar elementos que compõem o SEP. Muitos dos blocos conhecidos já estão disponibilizados para utilização, como motores, transformadores, geradores, entre outros.

Os resultados de simulações no Simulink podem ser exportados facilmente para o *workspace* do Matlab, por meio de blocos como o `to workspace`, ou por ajustes no bloco `Scope`, entre outros. O formato dos resultados dos arquivos do Simulink são matrizes conhecidas, onde cada coluna representa um canal do arquivo. Essa matriz pode ser interpretada diretamente pelo Matlab, dando maior facilidade quanto ao processo de leitura destes dados.

2.3 TRANSFORMADA WAVELET

Atualmente, existem diversas técnicas matemáticas capazes de fornecer índices e coeficientes na análise de diferentes sinais. A mais comumente conhecida é a *Fourier Transform* (FT) que faz a análise no domínio da frequência. Ela é bem adequada para análise de sinais estacionários, pois em sua definição, um sinal periódico qualquer pode ser escrito como sendo a somatória de senos e cossenos com diferentes frequências. Entretanto, ela torna-se ineficaz para sinais com transitórios e picos de curta duração, que estão presentes na grande maioria dos sinais elétricos contendo problemas de QEE. Para poder superar essa deficiência, são usadas ferramentas matemáticas alternativas. Algumas dessas são a *Short Time Fourier Transform* (STFT), *Wavelet Transform* (WT), entre outras. A STFT divide o sinal analisado em janelas (pedaços menores) de tamanhos

fixos, fazendo uma análise local em cada janela, considerando que o sinal é estacionário dentro da região de análise. Dessa forma, volta-se a uma análise pela FT, só que agora com uma maior resolução na obtenção dos harmônicos (HAYKIN; VEEN, 2001).

A análise de sinais pela WT representa o próximo passo após a FT: a técnica de variação do tamanho da janela ao longo do sinal de análise. A análise pela *wavelet* permite variações no intervalo de tempo da duração da janela (MISITI et al., 1996). Para obter informações mais precisas de baixas frequências, a janela de tempo fica maior, enquanto que para frequências mais elevadas, o intervalo de tempo torna-se reduzido. O plano *escala por tempo* da transformada *Wavelet* é demonstrado na Figura 3.

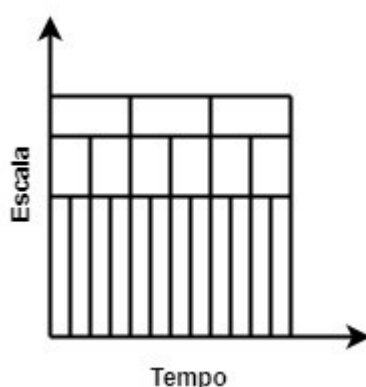


Figura 3 – Plano Escala x Tempo da análise por *Wavelet*.

Fonte: Autoria própria.

A análise pela WT se divide em Transformada Wavelet Contínua (CWT do original *Continuous Wavelet Transformer*) e Transformada Wavelet Discreta (DWT do original *Discrete Wavelet Transformer*). A CWT, similarmente a transformada de Fourier, é aplicada como sendo o somatório do sinal multiplicado por versões escalonadas e deslocadas da *wavelet*-mãe $\psi(t)$. A *wavelet*-mãe é uma forma de onda de curta duração que tem um valor médio nulo (energia finita) (REINALDO et al., 2003). A DWT permite que seja analisado um subconjunto de escalas e posições para os cálculos, diferentemente da CWT que realiza todas as escalas possíveis, ocasionando uma quantidade razoável de trabalho e gerando muitos dados, sendo essa um das principais diferenças entre as duas.

A DWT tem sido muito utilizada atualmente na análise de sinais com problemas de QEE (OLIVEIRA, 2007). Uma maneira eficiente desenvolvida por Mallat com a *wavelet* discreta (MALLAT, 1989), que ficou conhecida como análise Multirresolucional (MRA). Essa análise permite interpretar a WT como sendo um banco de filtros. A aplicação dessa transformada consiste na separação do sinal analisado, em dois sinais com faixas de frequências diferentes. O primeiro é chamado de *aproximação*, que

contém as baixas frequências. E o outro é chamado de *detalhe*, que contém as altas frequências. Essa decomposição pode ser feita sucessivamente com o novo sinal de aproximação gerado até uma escala j desejada. Visto que a maioria dos transitórios, ruídos, picos de tensão e outros defeitos são de alta frequência, o sinal de detalhe contém os eventos a serem analisados, enquanto que a aproximação é uma versão do sinal original, livre dos elementos de alta frequência. De modo analítico, pode-se descrever os coeficientes de aproximação e detalhe da escala $j + 1$ conforme as equações (1) e (2):

$$c_{j+1}(k) = \sum_n h(n - 2k)c_j(n) \quad (1)$$

$$d_{j+1}(k) = \sum_n g(n - 2k)c_j(n) \quad (2)$$

em que c_j representa os coeficientes de aproximação, d_j os coeficientes de detalhe, $h(\cdot)$ é a resposta ao impulso de um filtro passa-baixa, $g(\cdot)$ a resposta ao impulso de um filtro passa-alta e j é a escala de decomposição. Como se pode observar, os coeficientes de aproximação c_{j+1} tem um filtro $h(\cdot)$ do tipo passa-baixa, enquanto que os coeficientes de detalhe d_{j+1} possuem na sua equação o filtro passa-alta $g(\cdot)$ que realizam a convolução ao longo dos coeficientes de aproximação da escala inferior c_j . Os termos $-2k$ representam uma subamostragem por um fator de dois, no processamento de sinais digitais. Isso significa que, com o aumento dos níveis de decomposição, os coeficientes do sinal original são reduzidos pela metade a cada novo nível de decomposição.

A cada nível de decomposição ocorre o fenômeno de subamostragem por dois, chamado de *downsampling*. Esse fenômeno reduz do número de coeficientes das aproximações e detalhes pela metade a cada novo nível de decomposição. Essa redução dos coeficientes é justificável, visto que a somatória dos coeficientes de detalhes e aproximações devem resultar no mesmo número de coeficientes do sinal de análise (MISITI et al., 1996). Dessa forma, na análise final, obtém-se vários sinais de detalhe e apenas um sinal de aproximação, conforme mostrado na Figura 4. Para obter o sinal original a partir dos sinais da aproximação e dos detalhes, é aplicado o *upsampling* que é o inverso do *downsampling* mencionado anteriormente. Agora, os coeficientes começam a ser duplicados, de modo que o sinal obtido tenha a mesma quantia de coeficientes que o sinal em análise.

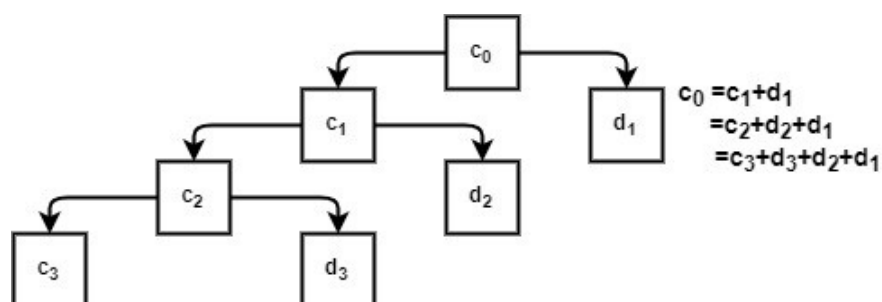


Figura 4 – Níveis de decomposição da Transformada *Wavelet*.

Fonte: Autoria própria.

2.4 CONSIDERAÇÕES FINAIS

Este capítulo apresentou os recursos e a fundamentação teórica dos principais temas a serem utilizados ao longo deste trabalho. Foram apresentadas as diversas classificações que envolvem os *frameworks*, foram expostos os três formatos de arquivo das fontes geradoras dos dados de análise de QEE deste trabalho, e por último apresentou-se teoria matemática da WT. É um capítulo de fundamental importância, para que seja compreendida a relação dos arquivos de leitura com o *framework* utilizando o *template* da WT.

Dentre as várias classificações existentes para a teoria de *frameworks*, neste trabalho ela foi limitada, para que sua definição seja aquela classificada como uma estrutura generalista, tendo todo seu desenvolvimento documentado e passado de forma clara, a fim de permitir melhorias contínuas, fazendo com que as características alcancem funções mais sofisticadas e recursos que potencializem ainda mais esse *framework* para análise de QEE. As estruturas dos três tipos de arquivos possíveis de serem lidos por esse *framework* estão nos formatos COMTRADE, CSV e Simulink. Os dois primeiros foram ilustrados por figuras e exemplos, a fim de dar ao trabalho maior aderência e entendimento por parte do leitor.

A Seção 2.3, que apresentou a WT, expôs de modo mais incisivo a teoria desenvolvida por Mallat, conhecida como análise multirresolucional (MRA), que decompõe o sinal de análise em dois outros sinais, chamados de *detalhes* e *aproximações*, cada um contendo as altas e baixas frequências, respectivamente. Vale ressaltar, que este trabalho desenvolveu um *template* utilizando a WT, mas não tem como objetivo apresentar de modo profundo a teoria dessa ferramenta matemática.

Objetivando alcançar as considerações teóricas, a revisão bibliográfica e a pesquisa de novos temas foram aprofundados, a fim de proporcionar ao trabalho resultados significativos que contribuam para a área de processamento de sinais, ramo de tamanha importância para a Engenharia Elétrica.

3 METODOLOGIA

De maneira geral, o desenvolvimento do *framework* consiste no processamento de arquivos e na criação, manipulação e operação com variáveis, vetores e matrizes. Além do desenvolvimento de algoritmos de técnicas matemáticas e operações na apresentação de gráficos. Para atender todos esses requisitos, será utilizado um dos *softwares* mais conhecidos e bem difundido nos meios acadêmicos que é o Matlab. Seus recursos e ferramentas atendem as demandas do trabalho. Por isso, esse será o *software* usado para o desenvolvimento do *framework*.

A grande totalidade do *framework* foi desenvolvida em linhas de comando, não havendo interfaces gráficas, com exceção de uma única janela que recolhe parâmetros de informação a fim de exibir a comparação entre os resultados do algoritmo *wavelet* com o sinal de análise. A opção pela inserção dessa interface é devido a facilidade quando comparada ao caso de recolher essas informações usando funções ou linhas de comando. Todo o restante do trabalho realiza a leitura dos parâmetros de entrada por funções, que recolhem os dados e realizam o processamento interno, retornando ao usuário apenas os resultados de interesse.

Uma visão geral do *framework* é apresentado na Figura 5. Os quatro blocos contidos nessa estrutura foram separados com o intuito de estabelecer e limitar as funções de cada etapa, dando ao usuário mais clareza e facilidade na operação.

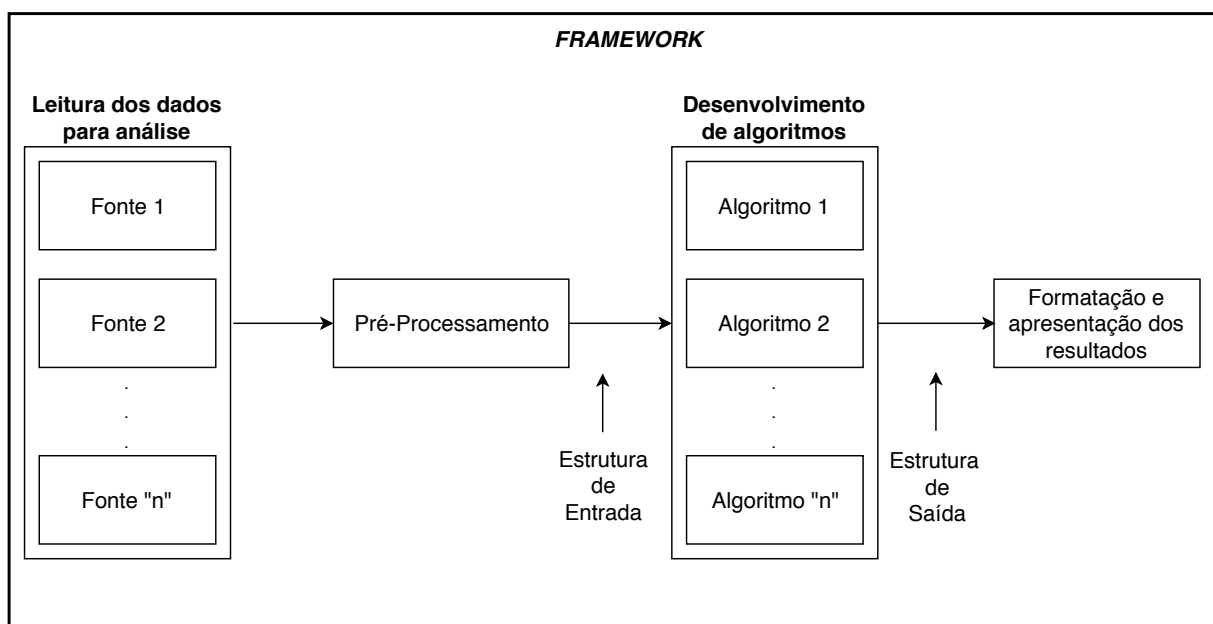


Figura 5 – Estrutura do *framework*.

Fonte: Autoria própria

Por meio da Figura 5 pode-se observar a capacidade de leitura de dados provenientes de n fontes diferentes, que passam por um mesmo pré-processamento e tratamento em seus arquivos, resultando em uma Estrutura de Entrada capaz de fornecer ao algoritmo um padrão nos arquivos de dados passíveis de implementação. Em seguida, é apresentada a etapa de *Desenvolvimento de Algoritmos*, na qual o usuário, por meio de necessidade e pesquisa, implementa o algoritmo de interesse. A saída desse algoritmo resulta em uma Estrutura de Saída, que será lida pela etapa de *Formatação e Apresentação dos Resultados*, responsável por exibir gráficos de interesse de acordo com os parâmetros informados. Esses blocos são apresentados nas Seções 3.1, 3.2, 3.3 e 3.4 apresentando as ferramentas, parâmetros e função de cada etapa no trabalho.

3.1 LEITURA DOS DADOS PARA ANÁLISE

O processo de leitura dos dados consiste em realizar e examinar o arquivo, desde legendas, títulos até os dados. Adaptando para um formato compatível com o *software* Matlab. Como já mencionado e apresentado, os formatos de arquivos desse trabalho são do COMTRADE, CSV e Simulink. Cada arquivo possui um processo de leitura específico, visto a diferença apresentada entre eles. Assim, como apresentado na Figura 5, é possível realizar a inserção de novas fontes além das três mencionadas, bastando que seja realizado o desenvolvimento do algoritmo de leitura específico dessa nova fonte.

Essa etapa é importante pois é feita a coleta de informações referentes aos sinais, bem como número de canais, tamanho dos canais, *labels*, validação do tipo de grandeza como corrente e tensão, entre outros. Todas as informações contidas nos arquivos são obtidas nessa etapa, sendo que esses parâmetros serão usados como entradas na etapa de pré-processamento, algoritmos de análise e apresentação dos resultados. A seguir são apresentadas as funções responsáveis pela leitura dos três arquivos.

A função de leitura para o arquivo COMTRADE é exibida abaixo. Para a sua leitura basta informar o nome do arquivo CFG apresentado entre parênteses na função abaixo, que deve estar localizado no mesmo diretório do *framework*.

```
read_Comtrade = read_comtrade(NomedoArquivo);
```

Para a função de leitura dos arquivos CSV deve-se informar o nome do arquivo e o número de canais que se deseja analisar. Conforme é apresentado abaixo na função de leitura para o arquivo CSV.

```
outCSV = readCSV('NomedoArquivo', 'Nchannels', 3);
```

A última função de leitura refere-se aos arquivos Simulink, ela requer como parâmetros o nome do arquivo (previamente salvo no *workspace* do Matlab), além da informação da grandeza que está sendo analisada (corrente ou tensão). A função de leitura para o arquivo Simulink é exibida abaixo.

```
outSimu = ReadSimu(Teste_Exemplo, 'Measure', 'Tensao');
```

Após a finalização do processo de leitura, o *framework* retira todas as informações e realiza a compatibilização para sua utilização pelo Matlab.

3.2 PRÉ-PROCESSAMENTO

Após a importação e leitura dos dados, realiza-se o segundo bloco de operação do *framework*, o pré-processamento. Essa etapa realiza a aplicação de várias técnicas para organização, tratamento e preparação dos dados. É uma etapa de fundamental relevância no processo, que compreende desde a interpretação de dados errados até a formatação dos dados a serem processados pelos algoritmos. Algumas das principais funções podem ser: a seleção de atributos, que é a escolha de dados que irão ou não ser utilizados; a limpeza de dados o que permite não comprometer a qualidade dos modelos de conhecimento a serem extraídos ao final do processo; além da construção de novos atributos; transformação de variáveis; entre outros.

No processo interno da função de pré-processamento são calculados valores estatísticos, como ponto máximo e mínimo de cada canal, que ficam salvos na estrutura da função. Também é calculado o valor do sinal por unidade (pu), com exceção do formato COMTRADE. Para realizar a transformação para pu, o *framework* identifica o número de fases (canais) que influenciam diretamente na equação, de acordo com o número de fases. A função COMTRADE, como mencionado, não realiza a conversão para pu pois há uma grande complexidade envolvendo essa operação, e as manipulações necessárias tornarem-se demasiadamente mais trabalhosos do que aqueles que dos arquivos CFG, Simulink e do escopo deste trabalho. Observou-se também, que grande parte da literatura do COMTRADE apresenta os resultados dos canais nas grandezas informadas pelo arquivo, apenas. Consequentemente a isso, não há necessidade de se inserir parâmetros nesta etapa. A função de pré-processamento, para o COMTRADE é representada conforme a seguir.

```
pre_Com = PreCom(read_Comtrade);
```

A função de pré-processamento dos arquivos CSV tem como entrada os arquivos da etapa de leitura, além de informações por parte do usuário, relativas aos valores de

tensão e potência de base dos canais. Esses valores são inseridos em um vetor-linha, com a mesma dimensão do parâmetro `Nchannels` da etapa de leitura.

```
outPreP = OutCsv(OutCSV, 3, 'VectorVbase', [Vb1 Vb2 Vb3], 'VectorSbase',  
                [Sb1 Sb2 Sb3]);
```

Para o último, a função de pré-processamento do Simulink tem como entrada a função de leitura e toda sua estrutura de dados. Nessa função, o usuário deve inserir o valor de tensão e potência de base que são atribuídos a todos os canais.

```
outpreSimu = OutPreSimu(outSimu, 'Vbase', Vb, 'Sbase', Sb);
```

Ao final da etapa de pré-processamento, os dados resultantes são estruturados em duas matrizes: `MatrizDados` e `MatrizLegenda`. A primeira contém os dados a serem analisados pelo algoritmo. A segunda contém arquivos de leitura, *labels*, títulos e inúmeras outras variáveis necessárias em etapas posteriores.

3.3 DESENVOLVIMENTO DE ALGORITMOS

A etapa de desenvolvimento dos algoritmos é uma das principais finalidades do *framework*. Após a realização das etapas anteriores, almeja-se disponibilizar todos os dados e recursos para o desenvolvimento do algoritmo baseado em diferentes técnicas matemáticas, favorecendo a melhor análise do sinal. A manipulação dos dados do sinal de análise proveniente de uma das três fontes, nesse momento se resume a análise de uma matriz de estrutura compatível com o programa (`MatrizDados`).

O algoritmo desenvolvido nesse trabalho para servir como *template*, conforme mencionado, é fundamentado na teoria *Wavelet Transform*. Para isso, uma função na parte principal do programa dispõe de posições definidas, onde o usuário deve informar alguns parâmetros conforme apresenta a função a seguir. É possível selecionar o nível de decomposição da WT e ainda informar qual a *Wavelet*-mãe que será usada para fazer a análise.

```
outAlgoritmo = AlgWave(outRead, outpre, NivelDecomposicao,  
                       'WaveletM', 'ψ')
```

A função `AlgWave` é a mesma para analisar todos os tipos de arquivos, visto que agora, estão em uma mesma *Estrutura de Entrada*. Esta função tem como parâmetro os resultados das etapas de leitura e pré-processamento. Em seguida o usuário informa o nível de decomposição da WT desejado, e por último a variável `WaveletM` que recebe o tipo de *wavelet*-mãe adotada.

Todos os resultados calculados pelo algoritmo, por meio da função AlgWave, retornam uma variável outAlgoritmo, que armazena a Estrutura de saída do algoritmo *Wavelet* apresentado como *template*. A variável outAlgoritmo consiste em duas matrizes: MatrizApp e MatrizDet. Essas matrizes contém as *aproximações* e *detalhes*, respectivamente. As dimensões dessas matrizes podem ser calculadas de acordo com a Eq.(3):

$$N_{col} = N_{channels} \times N_{decomp} , \quad (3)$$

em que N_{col} é o número de colunas de MatrizApp e MatrizDet, $N_{channels}$ é o número de canais do sinal analisado, podendo ser compreendido também pelo número de fases; e N_{decomp} é nível de decomposição informado pelo usuário na função AlgWave. Vale ressaltar que o número de linhas dessa matriz é variável, ela não possui o mesmo número de linhas em todas as colunas, mas o número de linhas cai pela metade a cada número de colunas correspondente à N_{decomp} . Pois a WT faz o *downsampling*, que reduz o número de coeficientes a metade em cada novo nível de decomposição (MISITI et al., 1996).

3.4 FORMATAÇÃO E APRESENTAÇÃO DOS RESULTADOS

A última etapa que compreende o trabalho é o tratamento e apresentação dos resultados no *framework*. Nessa etapa, uma série de opções foram desenvolvidas com o intuito de dar maior autonomia e liberdade ao usuário, permitindo impor suas preferências ao visualizar os resultados.

A função de apresentação dos resultados é a mesma para todos os tipos de arquivos, tendo cada um de seus parâmetros descritos a seguir.

```
showResults(outread, outpre, outAlgoritmo, 'Language', 'ENG',
'ShowAllapp', true, 'ShowAlldet', false, 'Colored', false, 'Tmin', 0.0,
'Tmax', 0.10, 'Nshow', true, 'histapp', false, 'histdet', false,
'histsignal', false, 'Comparative', true);
```

em que:

- outread: Todas as variáveis salvas na estrutura de leitura de dados;
- outpre: Variáveis e dados salvas na estrutura de pré-processamento;
- outAlgoritmo: Resultados do algoritmo salvas na estrutura do algoritmo;
- Language: Parâmetro que permite a seleção do idioma de apresentação dos resultados em Português (PT) ou Inglês (ENG). Quaisquer idiomas diferentes desses retornam um erro;

- `ShowAllApp`: É uma condição lógica (binária), onde no caso `'true'` apresenta todos os níveis de aproximações do canal. Caso contrário, (condição lógica `'false'`), apresenta apenas o último nível;
- `ShowAllDet`: É uma condição lógica (binária), onde no caso `'true'` apresenta todos os níveis de detalhes do canal. Caso contrário (condição lógica `'false'`), apresenta apenas o último nível;
- `Colored`: É uma condição lógica (binária). Com essa opção ativada (`'true'`), gráficos, histogramas e comparações serão coloridos durante a plotagem. Caso contrário, todos os resultados serão em preto e branco.
- `Tmin`: Opção que ajusta o ponto inicial do intervalo do gráfico dos resultados;
- `Tmax`: Opção que ajusta o ponto final do intervalo do gráfico dos resultados;
- `Nshow`: É uma condição lógica (binária). Com essa opção ativada (`'true'`), os resultados serão exibidos em valores reais de grandeza. Caso desativada (`'false'`), os resultados serão apresentados em pu;
- `histapp`: É uma condição lógica (binária). Com essa opção ativada (`'true'`) são exibidos os histogramas de todos os canais das aproximações. Caso desativada (`'false'`), os histogramas não são exibidos;
- `histdet`: É uma condição lógica (binária). Com esta opção ativada (`'true'`) são exibidos os histogramas de todos os canais dos detalhes. Caso desativada (`'false'`), os histogramas não são exibidos;
- `histsignal`: É uma condição lógica (binária). Com essa opção ativada (`'true'`) são exibidos os histogramas de todos os canais do sinal de análise. Caso desativada (`'false'`), os histogramas não são exibidos;
- `Comparative`: Ativada por uma condição lógica (binária). É uma janela especial que permite realizar a análise mútua do sinal original, detalhe e aproximação em uma única janela. A seleção do canal de análise, seleção do nível de detalhe e de aproximação são feitas por meio de uma interface gráfica simples, onde o usuário precisa apenas informar os parâmetros de interesse. Essa janela é exibida se o parâmetro `Comparative` estiver `'true'`. Caso contrário (`'false'`), esta janela não é apresentada.

3.5 CONSIDERAÇÕES FINAIS

O Capítulo 3 apresentou de forma contínua e detalhada como o *framework* realiza a leitura, pré-processamento, análise via algoritmo e apresentação dos resultados. Foi

possível observar em cada uma das seções quais são as principais funções e os parâmetros existentes a serem preenchidos pelo usuário. Esse capítulo teve como objetivo documentar detalhadamente o trabalho, para que futuros usuários realizem modificações e melhorias contínuas, especificando e impulsionando cada vez mais esse *framework*, a fim de ganhar aperfeiçoamento para pesquisas na área de QEE.

O primeiro bloco do *framework*, referente a Seção 3.1, apresentou as funções que realizam a leitura dos arquivos, sendo um total de três funções, uma para cada tipo de arquivo. Nessa seção, foi descrito o procedimento para inserção de parâmetros para processamento da leitura e quais eram as operações internas realizadas.

O segundo bloco, discutido na Seção 3.2, apresentou a etapa de pré-processamento, onde foram exibidas três funções, novamente, uma para cada formato de arquivo. Tais funções fazem a aquisição dos valores de base de tensão e potência, e realizam processamentos internos, cálculos matemáticos, estatístico e gerando uma conformidade dos dados de cada tipo de arquivo, resultando em uma Estrutura de Entrada.

Essa Estrutura de Entrada que é o formato dos dados interpretados pelo algoritmo apresentado na Seção 3.3 estão em padrão comum, não havendo mais diferenciação de acordo com a fonte geradora do sinal. Por isso, a função dos algoritmo desenvolvido é a mesma para análise de todos os arquivos, referindo-se a função *AlgWave*, onde seus parâmetros foram detalhados, permitindo que o usuário alcance os resultados de interesse, como o nível de decomposição e escolha do tipo de *wavelet*-mãe. Esse algoritmo teve como intuito servir de *template*, exemplificando uma apresentando uma aplicação prática do *framework*.

A etapa de apresentação, último bloco do *framework*, apresentado na Seção 3.4, apresentou as principais ferramentas e recursos de apresentação do trabalho. Esses recursos dão grande valorização ao trabalho, como exemplo, permitem limitar a faixa de tempo na análise do resultado, idioma, forma de exibição dos Detalhes e Aproximações, entre outros. Ainda há uma infinidade de recursos que podem ser inseridos, extraíndo ao máximo as informações referentes ao algoritmos que foram implementados, que vão surgindo de acordo conforme especialidades.

Por fim, vale ressaltar que toda documentação e detalhamento, foram feitos para incentivar e facilitar alterações que resultem em melhorias, seja na adesão de novas fontes para leitura dos dados, ou ainda alcançar novas manipulações na etapa de pré-processamento surgindo de acordo com a necessidade de cada algoritmo implementado. Todas as operações convergem para um único intuito: facilitar o desenvolvimento de algoritmos para análise de sinais, retirando todo trabalho do desenvolvedor de realizar atividades anteriores que não sejam exclusivamente no desenvolvimento do algoritmo e análise dos dados de interesse.

4 RESULTADOS

Este capítulo apresenta os principais resultados obtidos com o *framework*. Essa apresentação é acompanhada da descrição das escolhas dos parâmetros em cada função. Optou-se por diversificar o valores dos parâmetros nas funções para cada tipo de arquivo no intuito de apresentar uma maior pluralidade de resultados. A sequência de apresentação segue a ordem: arquivos COMTRADE, CSV e Simulink.

4.1 ARQUIVOS COMTRADE

Para a etapa de apresentação dos resultados, o arquivo COMTRADE à ser analisado é apresentado na Figura 6. O arquivo apresentado mostra a existência de 15 canais, sendo 5 analógicos e 10 digitais.

```

UTFPR,H30,1999
15,5A,10D
1,IL1,,,A,0.742585,0,0,-99999,99998,1,1,P
2,IL2,,,A,0.738278,0,0,-99999,99998,1,1,P
3,IL3,,,A,0.735624,0,0,-99999,99998,1,1,P
4,Io1,,,A,0.0030079,0,0,-99999,99998,1,1,P
5,Uo,,,%,0.0246874,0,0,-99999,99998,1,1,P
1,DI1,,,0
2,DI2,,,0
3,F1,,,0
4,F2,,,0
5,VI1,,,0
6,VI2,,,0
7,VI3,,,0
8,VI4,,,0
9,T1,,,0
10,T2,,,0
60
1
1919.39,7104
15/01/2014,13:32:53.364704
15/01/2014,13:32:54.290000
ASCII
1|

```

Figura 6 – Arquivo COMTRADE analisado pelo *framework*.

Fonte: Autoria própria

Conforme foi apresentado na Seção 3.1, é necessário que as funções de leitura, pré-processamento, algoritmo e apresentação dos resultados sejam ajustadas pelo

usuário. A seguir, todas essas funções são apresentadas e descritas nessa sequência. As duas primeiras funções apresentadas, realizam a leitura do arquivo com o nome `Exemplo_TCC` e fazem seu pré-processamento.

```
read_Comtrade = read_comtrade(Exemplo_TCC);  
  
pre_Com = PreCom(read_Comtrade);
```

A terceira função é responsável pelos parâmetros de análise do algoritmo *wavelet*. O nível de decomposição selecionado foi o terceiro, e a *Wavelet*-mãe escolhida foi a *Daubechies* do tipo 1 (db1) conforme apresenta a função `AlgWave`, que armazena na variável `outAlgoritmo` todos os resultados e variáveis de interesse obtidos pela função, conforme a seguir.

```
outAlgoritmo = AlgWave(outRead, outpre, 3, 'WaveletM', 'db1')
```

A função `showResults` recebe os parâmetros de escolha para apresentação dos resultados do *framework* com o *template* da WT. Nessa função, foram selecionados os parâmetros de idioma para Português (Brasil); selecionada a opção para exibir todos os níveis de aproximações e detalhes; os gráficos serão apresentados coloridos; a faixa de análise de tempo vai de 0,00 até 0,4202 s; e deve ser realizada a apresentação da janela *Comparative*. Em termos de código, a função que realiza esses objetivos é apresentada a seguir:

```
showResults(read_Comtrade, pre_Com, outAlgoritmo, 'Language', 'PT',  
'ShowAllapp', true, 'ShowAlldet', true, 'Colored', true, 'Tmin', 0.0,  
'Tmax', 0.4202, 'histapp', false, 'histdet', false, 'histsignal', false,  
'Comparative', true);
```

Como o arquivo que está sendo analisado possui 5 canais analógicos, a simulação pelo *framework* resultará em onze figuras. São duas figuras para cada canal (uma das aproximações e uma dos detalhes) e a última figura resulta da opção *Comparative*. Devido a esse número extenso de figuras, e com a finalidade de apresentar o funcionamento do *framework* de maneira eficiente, é feita a apresentação dos resultados de apenas um canal, poupando demasiadas figuras que atendem ao mesmo propósito do ponto de vista do funcionamento do *framework*. O canal escolhido para observação da resposta do *framework* é o canal analógico 5.

Finalizada as configurações das funções, inicia-se o processo de simulação. Como mencionado na Seção 3.4, existe uma única interface gráfica de interação com o usuário que será exibida pelo *framework*. Essa interface está associada ao parâmetro

Comparative. Exibida na Figura 7, essa interface gráfica recolhe informações do usuário referentes ao canal do arquivo a ser exibido, seguido do nível de decomposição do detalhe e da aproximação. Conforme mencionado, o canal de análise foi o canal analógico 5, o nível de detalhe foi 1, e o da aproximação foi 2.

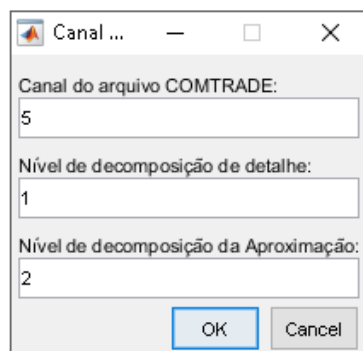


Figura 7 – Caixa de diálogo do parâmetro Comparative.

Fonte: Autoria própria

A Figura 8 apresenta o resultado da janela Comparative.

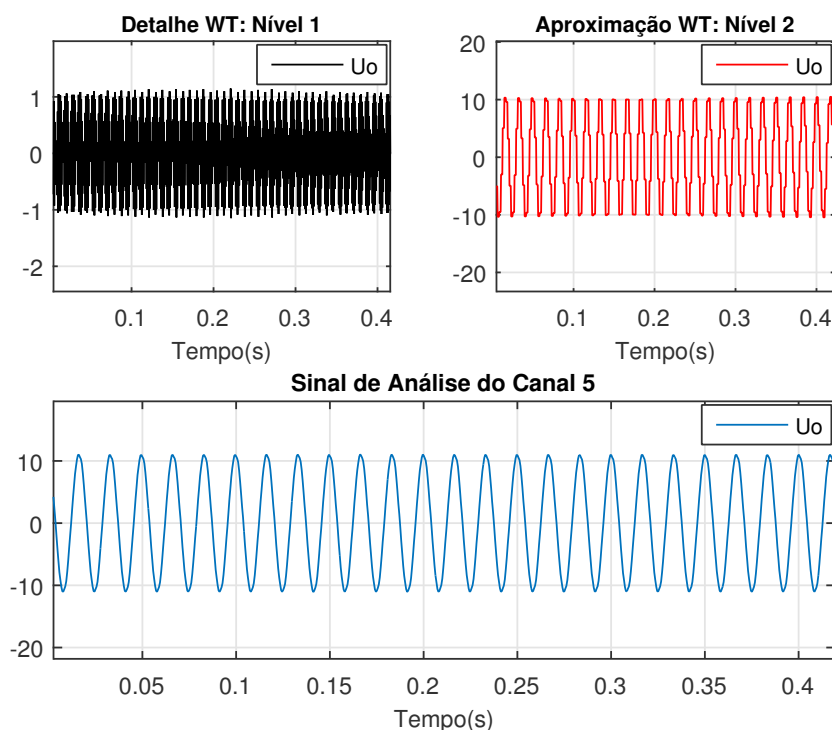


Figura 8 – Resultados da janela Comparative.

Fonte: Autoria própria

Pode-se observar na Figura 8 a apresentação do canal 5 de análise (em azul). Esse canal tem a unidade de medida em % e é representado pela legenda U_o , podendo ser observado no arquivo original da Figura 6. Essa legenda é lida para todos os canais pelo *framework* e posteriormente usada para ser apresentada nos resultados, a fim de situar o usuário na apresentação dos resultados, sobre qual a grandeza de análise. O gráfico em vermelho da Figura 8 apresenta o sinal de aproximação do segundo nível da WT, e o gráfico na cor preta apresenta o primeiro nível dos detalhes do sinal original. O arquivo utilizado não continha eventos de QEE, sendo utilizado apenas para demonstrar o funcionamento do *framework* para esse tipo de arquivo. Vale reafirmar que a janela *Comparative* funciona como um adicional do trabalho, e não obedece aos parâmetros de idioma e cor. Seus únicos parâmetros de ajuste são provenientes de interface apresentada na Figura 7.

Os resultados do algoritmo *wavelet* são apresentados nas Figuras 9 e 10. Como não houveram eventos de alta frequência, a transformada *wavelet* localiza a maior frequência como sendo a própria frequência do sinal, retirando coeficientes do próprio sinal de baixa frequência.

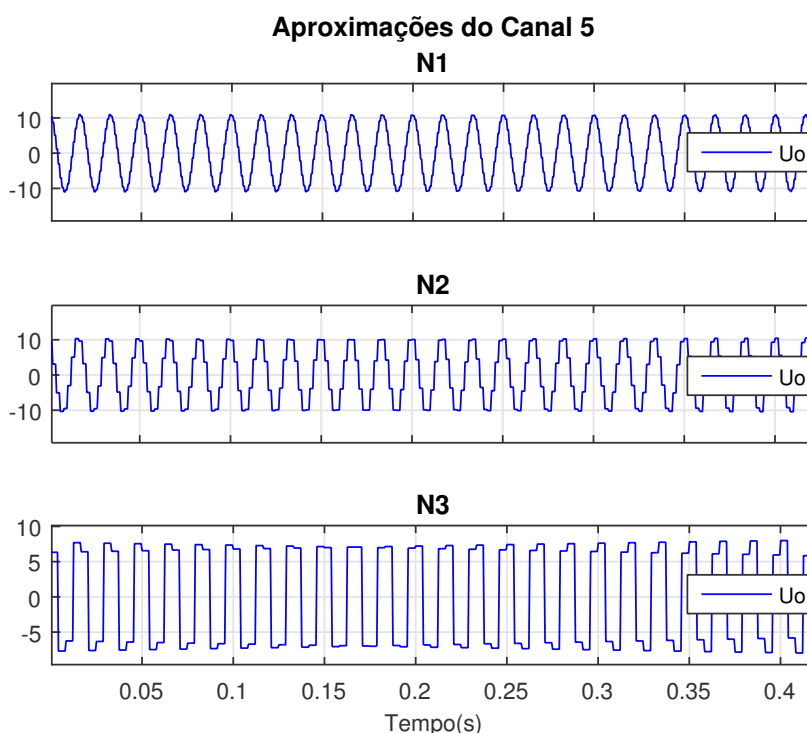


Figura 9 – Coeficientes de detalhes da WT do canal 5.

Fonte: Autoria própria

Como mostra a Figura 9, os coeficientes de aproximação são interpretados como coeficientes de detalhes, causando a deformação na forma de onda original. Isso

demonstra como a WT atua na análise MRA, semelhantemente a um banco de filtros, onde, ele localiza a maior frequência e a interpreta como coeficientes de detalhes.

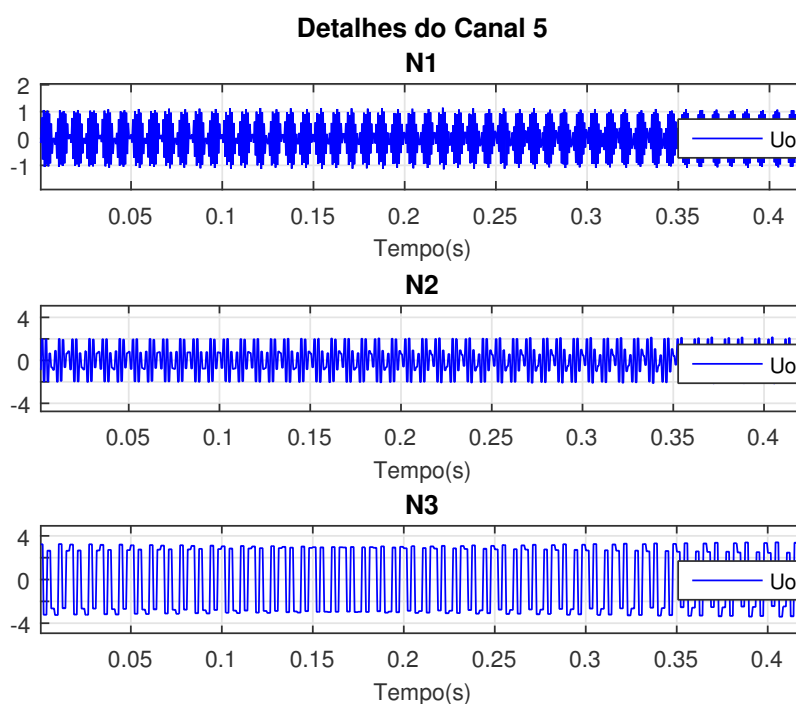


Figura 10 – Coeficientes de aproximação da WT do canal 5.

Fonte: Autoria própria

Foi possível perceber que a resposta do *framework* atendeu aos requisitos impostos na escolha dos parâmetros para o algoritmo e para apresentação dos resultados. Os canais 1,2,3 e 4 que não foram apresentados acompanham a mesma lógica dos resultados apresentados para o canal 5. As aproximações e os detalhes são decompostos até o nível 3, com idioma em português e apresentação em colorido, conforme foi declarado na função de apresentação dos resultados.

4.2 ARQUIVOS CSV

A segunda seção deste Capítulo consiste na apresentação dos resultados do *framework* para análise de arquivos no formato CSV. Esse tipo de arquivos, como mencionado, contém matrizes onde os delimitadores são vírgulas e quebras de linha. A Figura 11 apresenta um recorte do arquivo que foi analisado (o arquivo completo possui dimensões 150004 linhas \times 10 colunas).

Time	, Voltage	, Voltage	, Voltage	, Voltage
Second	, X0001A	, X0001B	, X0001C	,
	,	,	,	,
0.000000e+000,	9.890413e+003,	-1.263244e+003,		
2.000000e-006,	9.887205e+003,	-1.255183e+003,		
4.000000e-006,	9.883991e+003,	-1.247122e+003,		
6.000000e-006,	9.880771e+003,	-1.239059e+003,		
8.000000e-006,	9.877547e+003,	-1.230995e+003,		
1.000000e-005,	9.874315e+003,	-1.222932e+003,		
1.200000e-005,	9.871079e+003,	-1.214867e+003,		
1.400000e-005,	9.867837e+003,	-1.206801e+003,		
1.600000e-005,	9.864590e+003,	-1.198735e+003,		
1.800000e-005,	9.861337e+003,	-1.190668e+003,		
2.000000e-005,	9.858078e+003,	-1.182601e+003,		
2.200000e-005,	9.854813e+003,	-1.174533e+003,		
2.400000e-005,	9.851543e+003,	-1.166464e+003,		
2.600000e-005,	9.848268e+003,	-1.158395e+003,		
2.800000e-005,	9.844986e+003,	-1.150324e+003,		
3.000000e-005,	9.841699e+003,	-1.142254e+003,		
3.200000e-005,	9.838407e+003,	-1.134182e+003,		
3.400000e-005,	9.835109e+003,	-1.126110e+003,		
3.600000e-005,	9.831806e+003,	-1.118037e+003,		
3.800000e-005,	9.828496e+003,	-1.109964e+003,		

Figura 11 – Arquivo .CSV de análise do *framework*.

Fonte: Autoria própria

Conforme a segunda linha do arquivo na Figura 11, são apresentados os nomes dos canais. Esses *labels* são utilizados pelo processamento para classificação dos canais. A terceira linha, que está vazia, pode ser usada como um segundo nível de classificação, passível de ser editada pelo usuário e preenchida com a legenda de interesse.

Independente do tamanho do arquivo, o *framework* permite a análise completa de todos os canais, ou ainda, de uma faixa de específica de interesse, dando maior agilidade no processamento, ficando mais direcionado ao interesse do usuário. A função do *framework* que permite a escolha da faixa de canais para arquivos nesse formato é o parâmetro `Nchannels` apresentado na função de leitura `readCSV` a seguir.

```
outCSV = readCSV('NomedoArquivo', 'Nchannels', 3);
```

Com o parâmetro `Nchannels` da função `readCSV` preenchido com o valor 3 a análise do arquivo fica limitada apenas para as três primeiras colunas, ao invés da análise de todo o arquivo. O próximo passo consiste em informar os parâmetros de pré-processamento. Para que a função `outPreP` apresente os resultados em `pu`, deve-se

informar os valores de tensão e potência de base, por meio de um vetor. Esse vetor deve conter a mesma dimensão do número de canais informados no parâmetro `Nchannels` da função `readCSV`. Caso contrário, o programa retornará um erro, e o usuário deve inserir os dados corretamente. Caso não haja interesse por parte do usuário para análise dos resultados em pu, deve-se completar o vetor com valores unitários e inserir `true` no parâmetro `Nshow` da função `ShowResults`. Dessa forma, a função fica disposta conforme a seguir, onde nesse caso, foram inseridos valores aleatórios de base para fins de análise da resposta do *framework*.

```
outPreP = OutCsv(outCSV, 3, 'VectorVbase', [11000 15000 19000],  
                'VectorSbase', [1 2 3]);
```

O terceiro passo do *framework* na análise CSV é o ajuste dos parâmetros da função do algoritmo. O primeiro parâmetro a ser analisado é o parâmetro informando o nível de decomposição. Para esse *template*, o valor máximo de decomposição foi 7. A *wavelet*-mãe escolhida foi a *db1*, conforme é apresentado na função `AlgWave` a seguir.

```
outAlgoritmo = AlgWave(outCSV, outPreP, 7, 'WaveletM', 'db1')
```

A última função que deve ser configurada consiste na apresentação dos resultados do algoritmo. Para apresentar outras configurações daquelas já vistas na Seção 4.1, nessa apresentação os gráficos dos detalhes e aproximação são em preto e branco (parâmetro `Colored` recebe `false`), e ao invés de exibir todos os níveis de decomposição das aproximações, foi exibido apenas o último nível (parâmetro `ShowAllapp` recebe `false`). Em maneira prática, para atender esses interesses, a função `showResults` fica como a seguir.

```
ShowResults(outCSV, outAlgoritmo, 'Language', 'PT', 'ShowAllapp', false,  
            'ShowAlldet', true, 'Colored', false, 'Tmin', 0.0, 'Tmax', 0.3,  
            'histapp', false, 'histdet', false, 'histsignal', false,  
            'Comparative', true);
```

Após isso, é iniciado o processo de exibição dos resultados. Inicialmente, como a opção `Comparative` foi selecionada, é mostrada a interface da Figura 6. O canal escolhido foi o primeiro (X0001A), conforme foi mostrado na Figura 11. O nível de *decomposição* de detalhe foi 2 e nível de *decomposição* de *aproximação* foi 7. Após isso, o programa inicia o processo de simulação. A Figura 12 apresenta a janela `Comparative` exibindo o canal original (azul), a aproximação (vermelho) e o detalhe (preto) do canal X0001A.

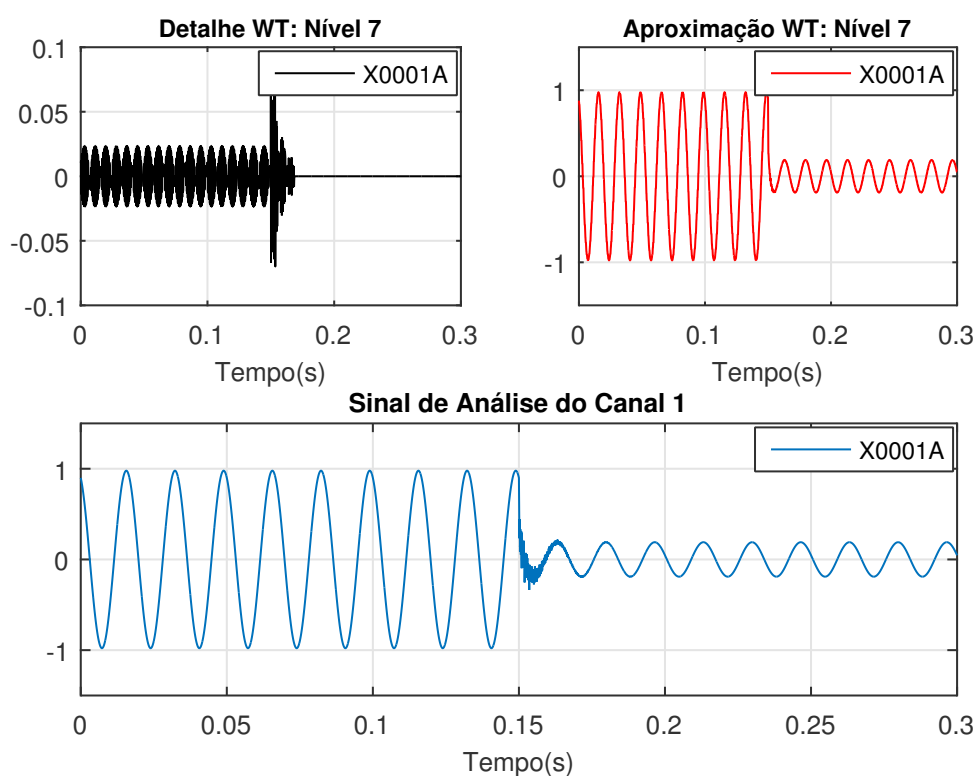


Figura 12 – Janela Comparativa do arquivo CSV.

Fonte: Autoria própria

É possível observar, por meio da Figura 12, como atua o algoritmo *wavelet*. Fazendo a separação do transitório de alta frequência, sendo exibidos nos coeficientes de detalhes de Nível 7 (em preto), enquanto que a parte suave do sinal (coeficientes de aproximação) está livre desses transitórios (gráfico vermelho), em comparação com o sinal original (azul). No instante de tempo 0,15 s ocorre um evento QEE. Há uma redução de aproximadamente 80 % da amplitude de tensão, passando de 1 pu para 0,2 pu em regime estacionário. Ainda, antes que o sistema entrasse em regime, pode-se observar variações de alta frequência na transição em torno do tempo de 0,15 s. O detalhe de nível 7 do algoritmo *wavelet* consegue detectar a alta frequência contida nesse sinal.

Para poder observar melhor quais são os níveis de detalhes que conseguem localizar essas altas frequências, a Figura 13 apresenta a decomposição contendo todos esses níveis. A Figura 13 é apresentada como consequência do parâmetro `ShowAllDet` receber `true`.

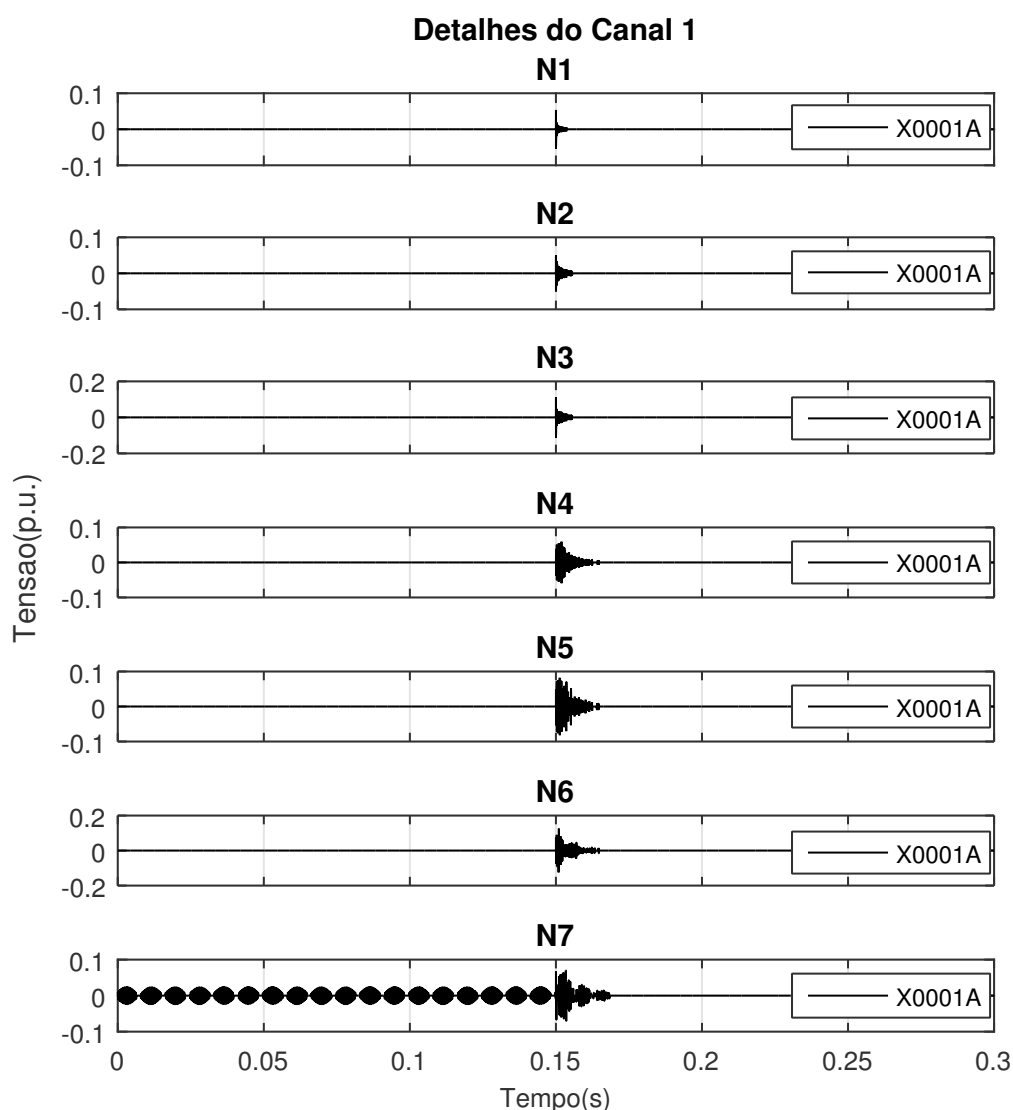


Figura 13 – Detalhes do canal 1 do arquivo CSV com nível de decomposição *wavelet* 7.

Fonte: Autoria própria

Pela Figura 13 é possível observar que a detecção foi maior no nível 7 de decomposição, identificando até a frequência de um sinal periódico de amplitude muito baixa antes do evento de QEE que ocorreu em 0,15 s.

O último resultado a ser apresentado pelo *framework* refere-se as *aproximações* do sinal. Como o parâmetro `ShowAllDet` foi ajustado para `false`, o função de apresentação dos resultados irá exibir apenas o último nível de decomposição da transformada *wavelet*, e como o nível de decomposição escolhido foi o 7, este é o nível de aproximação a ser graficamente exibido ao usuário.

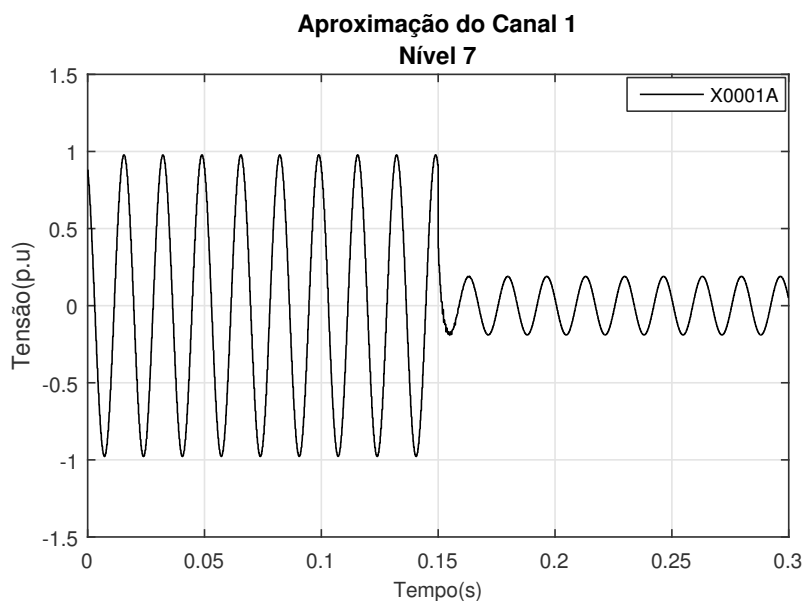


Figura 14 – Aproximação nível 7 do canal X0001A.

Fonte: Autoria própria

Conforme a Figura 14 pode-se observar a versão suavizada do sinal, deixando mais claro o afundamento de tensão de aproximadamente 80 %, como discutido anteriormente.

4.3 ARQUIVOS SIMULINK

O último formato de arquivo a ser analisado pelo *framework* são arquivos do Simulink. Após feita a simulação dentro do próprio Simulink, realiza-se a exportação dos dados, a fim de poderem ser manipulados no Matlab ou em outras plataformas. Se a análise for ser realizada no próprio Matlab, os dados devem ser salvos no *workspace*. Essa exportação pode ser realizada de diversas formas e todas atendem a um mesmo objetivo, elas inserem duas matrizes no *workspace*, uma relativa ao tempo de simulação, e uma outra matriz, correspondendo aos dados de análise. O *framework* desenvolvido não restringe ou impõe um modo específico para exportar os arquivos, a única consideração é que esses dados devem ser salvos como uma variável no *workspace* para que o *framework* faça a leitura e processamento destes dados.

Para apresentar uma análise desse tipo de arquivo pelo *framework*, é apresentado na Figura 15 um exemplo de simulação simples realizada via Simulink. Um sistema trifásico, em que a fase A sofre interferência por alta frequência. A escolha por esse tipo de sinal é para demonstrar como a *wavelet* consegue atuando segundo a teoria MRA, tentando realizar a visualização do sinal livre de interferência de QEE.

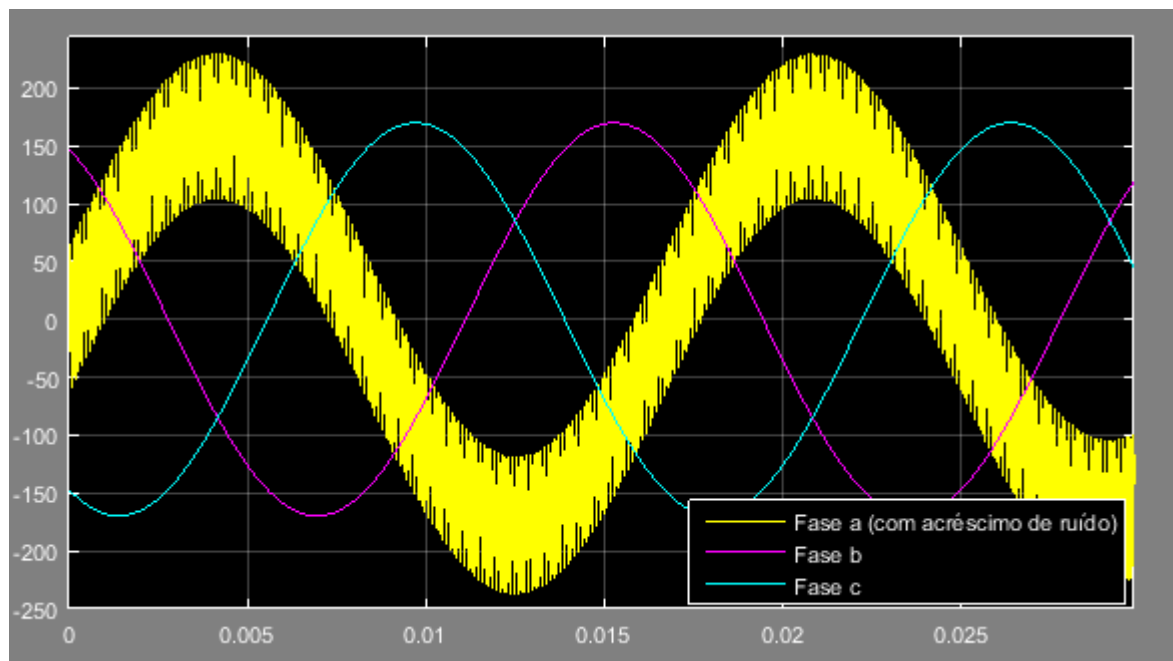


Figura 15 – Sistema de análise do arquivo Simulink.

Fonte: Autoria própria

Os procedimentos a seguir descrevem as funções e os parâmetros responsáveis para análise dos arquivos Simulink. A primeira função, responsável pela leitura do arquivo, é a `ReadSimu`, que conta com dois parâmetros que devem ser informados pelo usuário. O primeiro refere-se ao nome da variável de análise (previamente exportado para o *workspace*). O segundo parâmetro recebe a grandeza que está sendo lida (tensão ou corrente). Essa função está lendo o sinal de tensão do sistema trifásico da Figura 15, e os resultados obtidos com essa função são armazenados na variável `outSimu`. Dessa forma, a informação dos parâmetros ficam conforme a seguir.

```
outSimu = ReadSimu(Teste_Exemplo, 'Measure', 'Tensao');
```

O *framework* possui uma biblioteca que interpreta os nomes das grandezas inseridas acima. Por exemplo, a grandeza `Corrente` poderia ser escrita com letra minúscula, ou até mesmo em inglês, que sua interpretação pela função seria correta.

A segunda função, responsável por realizar o pré-processamento funciona de maneira semelhante a função de pré-processamento dos arquivos CSV. A função `OutPreSimu` recebe os dados obtidos pela etapa de leitura, além dos valores de base de tensão e corrente que devem ser informados pelo usuário, conforme exemplificado a seguir.

```
outpreSimu = OutPreSimu(outSimu, 'Vbase', 1000, 'Sbase', 20000);
```

Na sequência, a função `AlgWave` recebe os parâmetros referentes ao algoritmo *wavelet*. Semelhantemente as já preenchidas nas funções nas Seções 4.1 e 4.2. Os parâmetros inseridos nessa função referem-se ao nível de decomposição e a *wavelet*-mãe selecionada. O nível de decomposição escolhido foi 5 e a *wavelet*-mãe foi `db1`.

```
outAlgoritmo = AlgWave(outSimu, outpreSimu, 5, 'WaveletM', 'db1')
```

Por fim, realizam-se os ajustes na função de apresentação dos resultados. A função que atende aos interesses é apresentada em sequência. Nessa seção, todos os níveis de decomposições dos detalhes e aproximações foram apresentados, e a unidade de medida foi ajustada conforme o Sistema Internacional (S.I).

```
showResults(outSimu, outpreSimu, outAlgoritmo, 'Language', 'ENG',  
'ShowAllapp', true, 'ShowAlldet', true, 'Colored', true, 'Tmin', 0.0,  
'Tmax', 0.05, 'histapp', false, 'Nshow', true, 'histdet', false,  
'histsignal', false, 'Comparative', true);
```

Dessa forma, é iniciado o processo de simulação. A janela da opção `Comparative` é preenchida com nível de decomposição 5 para detalhes e aproximações do canal 1 (fase "a"). O resultado da janela `Comparative` é apresentada na Figura 16.

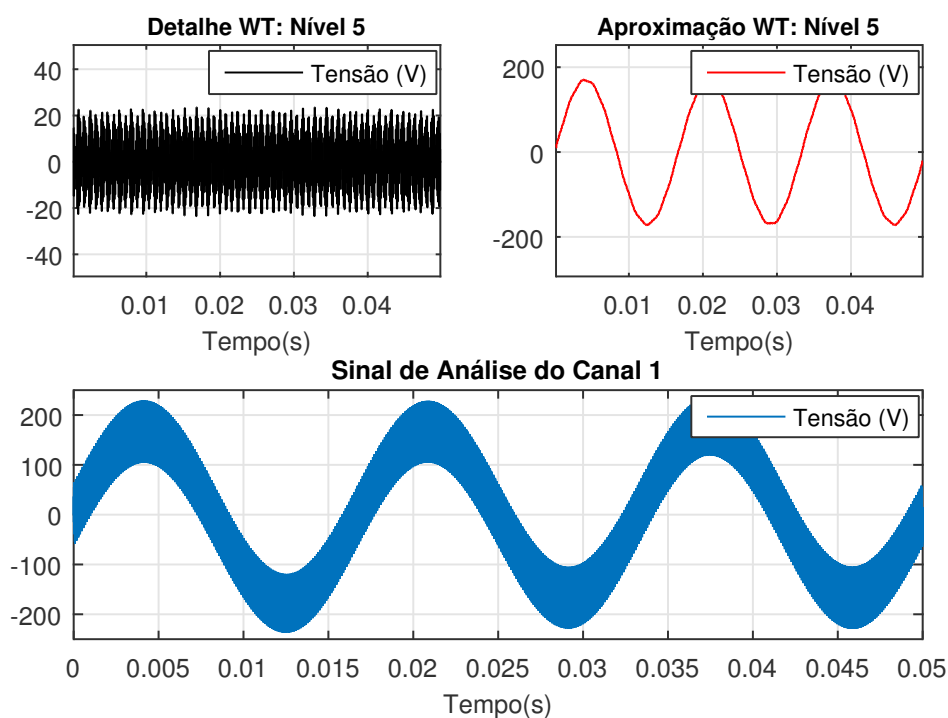


Figura 16 – Janela `Comparative` do *framework* para os arquivos do Simulink.

Fonte: Autoria própria

O valor de pico da onda da Figura 16 é calculada por meio da Eq. (4):

$$V_{pico} = V_{rms} \sqrt{2}, \quad (4)$$

em que, para esse caso, o valor de V_{rms} é de 120 V, resultando em um valor de pico (V_{pico}) de aproximadamente 180 V. Esse valor de pico é visualizado no sinal de aproximação da WT de nível 5 (vermelho). O sinal de análise (azul) da Figura 16 possui uma amplitude maior devido a sobreposição ao sinal original. Na Figura 17 é apresentado os coeficientes de detalhes, dos cinco níveis de decomposição.

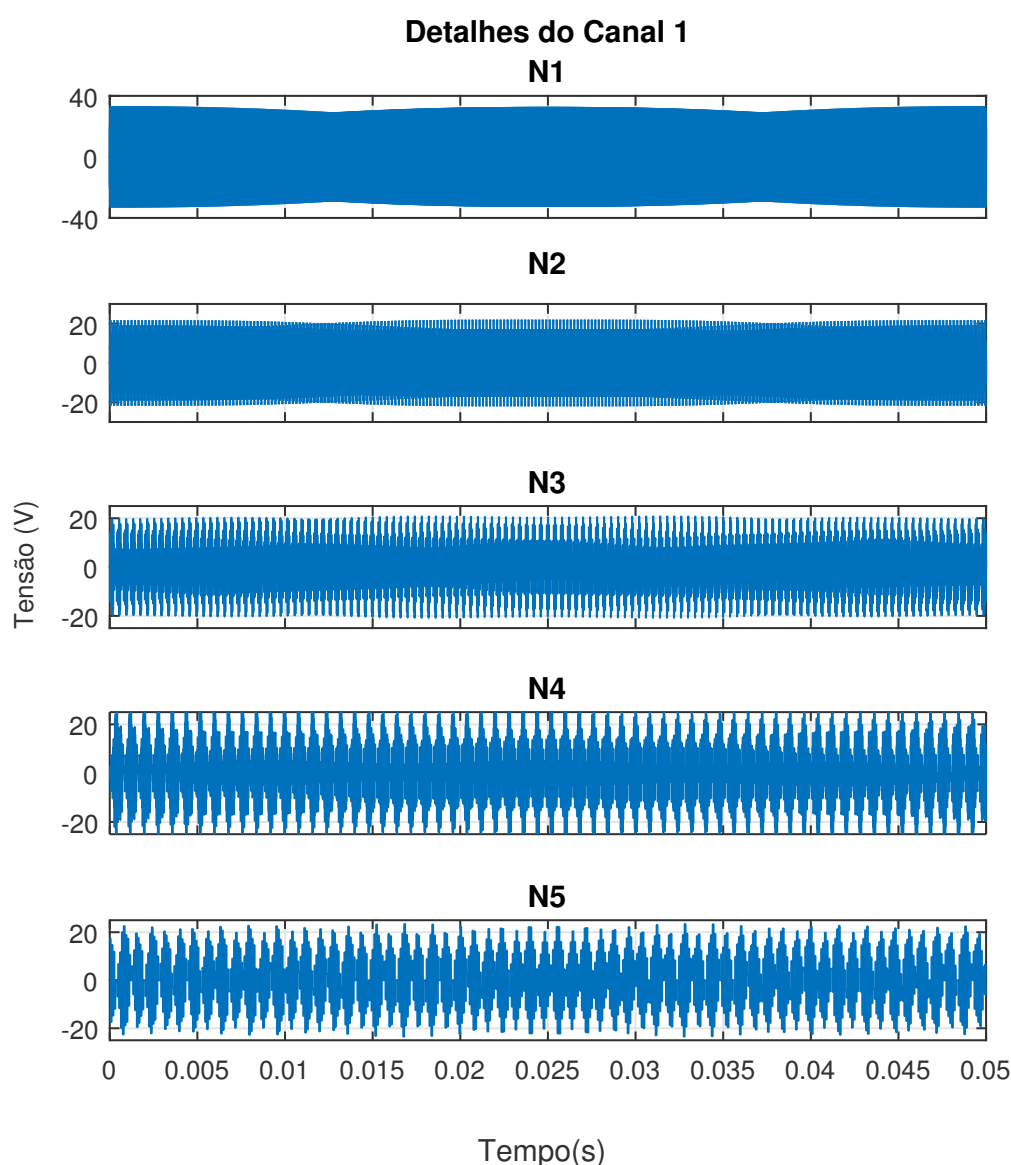


Figura 17 – Coeficientes de detalhes do canal 1 para arquivos do Simulink.

Fonte: Autoria própria

Os coeficientes mostram que grande parte da alta frequência se entravam nos níveis iniciais de decomposição, perdurando até o nível 5 (N5), tendo uma amplitude

significativa. Por meio dessa detecção da alta frequência, o sinal resultante da parte suavizada. Isso é, as aproximações, tendem cada vez mais para a forma de onda do sinal original, livre de eventos de QEE. Para que seja feita a observação do impacto dessas decomposições nível a nível. Na Figura 18 são apresentadas as aproximações do sinal.

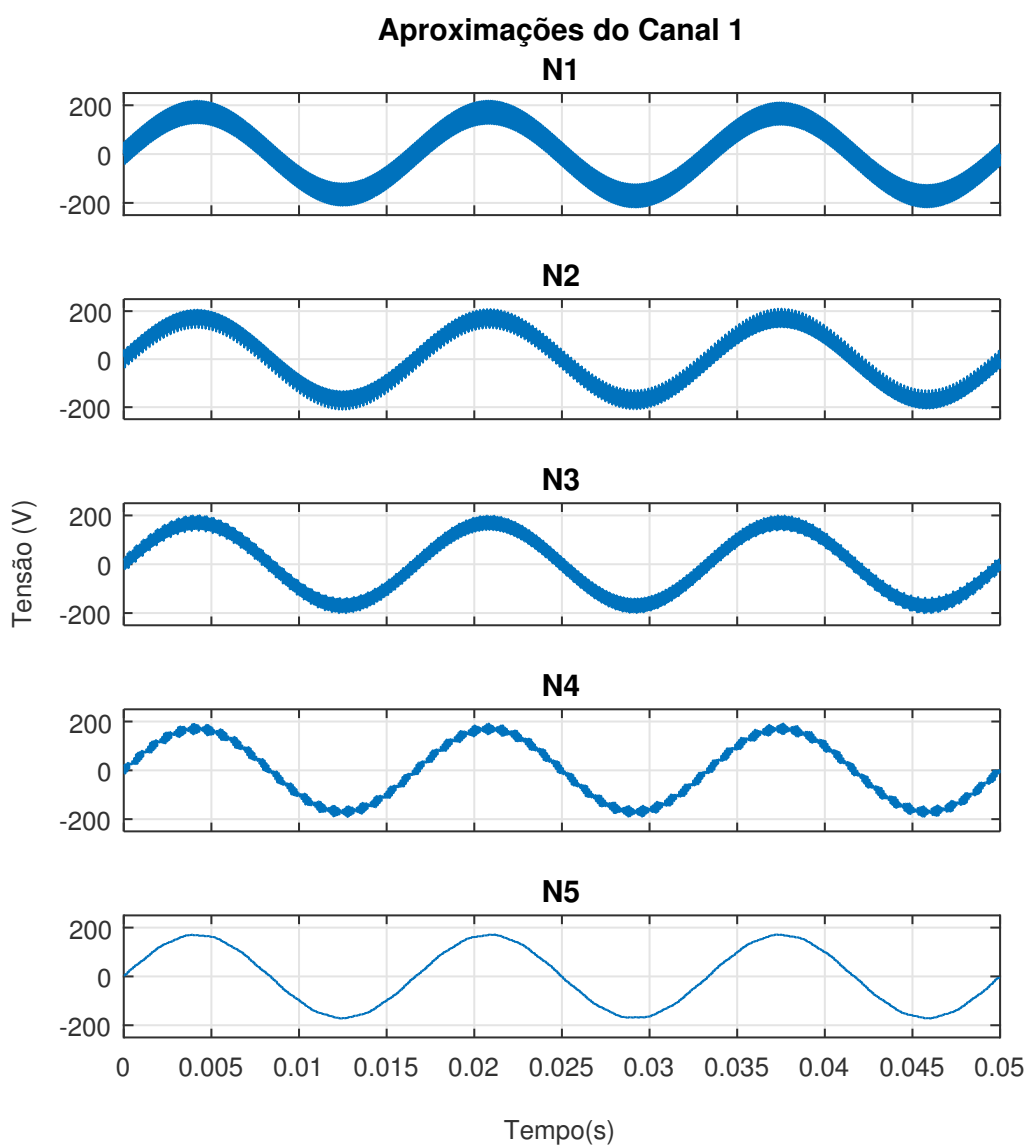


Figura 18 – Sistema de análise do arquivo Simulink.

Fonte: Autoria própria

A escolha do nível de decomposição, a fim de obter o sinal apresentado no nível 5 (N5) foi alcançado por meio de observação e testes de níveis superiores. Para os valores de decomposição superiores ao nível 5, a redução de coeficientes devido ao efeito do *downsampling* faz com que a forma de onda sofra alterações que caracterizariam uma

análise errada. Pois, o número reduzidos de coeficientes, faz com que a reconstrução se distancie muito do sinal original.

Para tanto, o *framework* permite a alteração do nível de decomposição a cada nova simulação. Recomenda-se que o usuário decomponha até um nível mais elevado, e por meio da observação, encontre o nível que obteve a maior aproximação do sinal original, livre dos eventos sem perder as características originais. Caso nenhum dos níveis seja atingido, pode-se alterar o modelo da *wavelet*-mãe. Nesse exemplo, todas as funções usaram a *db1*, mas o *framework* encontra-se habilitado para utilização de outras *wavelets*-mãe conhecidas.

Do ponto de vista de funcionamento do *framework*, os parâmetros informados em cada uma das funções corroboraram para que fossem apresentados todos os níveis de decomposição tanto dos detalhes, quanto das aproximações. Os gráficos foram apresentados em colorido, o idioma em português e a faixa de tempo da plotagem respeitou os parâmetros de tempo inicial e final.

4.4 CONSIDERAÇÕES FINAIS

Nesse capítulo foram apresentados os principais resultados para cada um dos arquivos, diversificando a escolha dos parâmetros para cada arquivo a fim de observar os recursos desenvolvidos do *framework*. Um dos pontos chaves enfatizados foi na facilidade que se tornou a análise desses arquivos antes incompatíveis com a plataforma do Matlab, com exceção dos arquivos Simulink. Resumindo toda a análise em 4 funções por arquivo, com parâmetros simples e documentados. Livrando o oneroso trabalho no desenvolvimento de funções avançadas, trabalho de compatibilização e algoritmos de apresentação. Resumindo-se a uma atividade direcionado ao objetivo principal de pesquisadores da área.

Mesmo que ainda com uma estrutura generalista, os resultados observados neste capítulo mostraram sofisticados parâmetros que permitem uma fácil leitura e interpretação dos resultados destes arquivos. Essa estrutura generalista, assim é descrita pois tem como intuito um trabalho continuado, uma análise mais aprofundada que potencialize ainda mais os recursos deste *framework*, a fim de tomar características mais específicas de acordo com a necessidade de cada pesquisador.

O algoritmo *wavelet* foi apresentado como exemplo de implementação para outros algoritmos que podem ser desenvolvidos por futuros pesquisadores. Esse *template* apresentou recursos de seleção do nível de decomposição do sinal via análise MRA, e também permitiu a escolha do tipo da *wavelet*-Mãe. Para ilustrar uma próxima função, suponha outro algoritmo, baseado em séries de Fourier, a função relativa a esta técnica matemática pode ter como variáveis de entrada, o número de harmônicas,

por exemplo. E assim desenvolver todo o processo interno do algoritmo, retornando a Estrutura de Saída para a etapa de apresentação dos resultados.

A última função, responsável pela apresentação gráfica dos resultados, expõe recursos relevantes para impulsionar a observação dos resultados do algoritmo *wavelet*, como por exemplo o parâmetro que permite visualizar todos os níveis de decomposição dos detalhes e aproximações, o idioma de apresentação, opção de visualização em preto e branco ou colorido, seleção da faixa de tempo de análise do sinal, entre outros. Novos algoritmos, com novas características, terão ferramentas de apresentação diferentes e irão surgir conforme demanda de cada algoritmo, proporcionando visualizar os principais resultados obtidos.

Com relação a interpretação dos resultados, como a classificação e discussão aprofundada dos eventos de QEE não é o foco principal deste trabalho, mas sim fornecer aos pesquisadores a ferramenta que de condições e possibilite realizar tal análise sobre estes resultados. Por isso, foram apresentadas todas as funções, detalhando as etapas, a fim de proporcionar plena compreensão pelos usuários, para que esses consigam realizar conclusões importantes em suas análises.

5 CONCLUSÕES

O trabalho apresentou o desenvolvimento de um *framework* capaz de unir funções que facilitem a compreensão de dados provenientes de fontes externas para o processo de desenvolvimento de algoritmos com a finalidade de realizar a análise de eventos do QEE.

Esse trabalho, teve intencionalmente o objetivo de dividir por blocos de programas, a fim de facilitar a modificação e a realização de adaptações, não apenas no desenvolvimento do algoritmo, como em todo o *framework*. Pois é levado em conta que esse deve estar em constante melhoria e adaptação. Todas essas etapas, foram apresentadas no Capítulo 3 deste trabalho.

No âmbito da metodologia do processo, as quatro etapas do *framework* (leitura, pré-processamento, *template* baseado na WT e apresentação dos resultados) foram implementadas e alcançaram um consenso positivo pelos autores do trabalho e por meio de avaliações e estudos.

A primeira fonte de dados apresentada pelo trabalho refere-se aos arquivos COMTRADE, adquiridos por leitura direta de analisadores de qualidade de energia do SEP. Esse não é um tipo de arquivo com grande difusão nos meios de ensino, e pouco é encontrado sobre ele em literaturas ou acervos. Esse tipo de estrutura de arquivo apresentam pontos positivos, pois permitem a leitura de diversos canais, informações temporais e locais da ocorrência, podendo ser abertos por *softwares* simples de computadores, além de permitir a manipulação de seus dados, criando, excluindo ou inserindo variáveis de interesse. Apesar de pouco conhecido, esse tipo de arquivo não possui grande complexidade quanto a compreensão de sua estrutura. O trabalho apresentou sua estrutura de forma sucinta, no segundo e terceiro capítulo.

O *framework* permitiu que o usuário realizasse a análise comparativa por meio da janela *Comparative* entre o sinal original e os resultados do algoritmo *wavelet*, deixando livre a escolha do canal e nível de decomposição. Além disso, foram exibidos separadamente os níveis de decomposição das aproximações e dos detalhes. O exemplo do arquivo COMTRADE apresentado não continha eventos em QEE, era um sinal simples, com unidade de medida em percentual de tensão conforme informada pelo arquivo. Por meio desse canal, foi possível observar a atuação prática da *wavelet*. Como não haviam eventos de alta frequência, a maior frequência detectada foi a própria frequência do sinal, e esse passou a ser compreendido como sinal de detalhe.

Conforme apresentado, a etapa de apresentação dos resultados permite a inserção de diversos parâmetros, como a informação do idioma, cor, faixa de tempo para análise, resultados apresentados em pu ou em grandezas do SI, entre outros.

O segundo formato de arquivo analisado foi o CSV, alcançando os resultados esperados, por meio dos parâmetros de interesse e das informações passadas para o *framework* por meio das funções de cada etapa. Por ser um formato de arquivo proveniente do *software* ATP, de grande utilização na Engenharia, esses resultados favorecem grandemente a área de pesquisa, permitindo que seus dados sejam manipulados no Matlab. Sua análise pelo *framework* permite a seleção de uma faixa de canais específica, não exigindo do usuário a leitura e compreensão dos arquivos CSV, pois após a etapa de pré-processamento, esses dados agora correspondem a estrutura padrão simples (matriz).

O terceiro arquivo apresentado foi a plataforma integrada do Matlab, o Simulink. Após apresentadas as funções e os parâmetros que regem o objetivo de apresentação desse tipo de arquivo, os resultados para este tipo de arquivo alcançaram os resultados esperados. Foi possível separar e analisar o canal de interesse, realizar a decomposição e obter o resultado do algoritmo para a fase que continha o evento de QEE. Todos os parâmetros informados corroboraram para atingir o tipo de apresentação semelhantemente aos outros dois arquivos anteriores.

De maneira geral, o trabalho alcançou resultados positivos, apresentando os passos para utilização do *framework*, além do *template* baseado na WT. Dando aos pesquisadores da área, um início de projeto que atue como suporte suas pesquisas para aqueles que trabalham com desenvolvimento de algoritmos para análise de QEE.

Para este trabalho, uma série de fatores merecem maiores investigações. Os principais pontos identificados são:

- A ampliação da biblioteca de funcionalidades da etapa de pré-processamento, que venham a surgir de acordo com a demanda do arquivo por algoritmos diferentes, como por exemplo valores estatísticos avançados.
- Desenvolvimento de uma interface gráfica que permita uma maior interação visual com o usuário. Uma das ferramentas viáveis de realizar essa tarefa é o GUIDE, uma ferramenta que se encontra diretamente na plataforma do Matlab. Essa interface facilitaria ainda mais o processo de análise por pesquisadores.
- A expansão no número de fontes de leitura. Como foi apresentado neste trabalho, existem três fontes de leitura, e a ampliação do número dessas fontes pode dar ainda maior utilidade para esse *framework*.

- Ampliação dos recursos de apresentação dos resultados. Como todo algoritmo tem uma apresentação própria, a junção de ferramentas de apresentação mais generalistas podem impulsionar a melhoria dos recursos de apresentação.

-

REFERÊNCIAS

- ACCIOLY, A. G. H. Ferramenta computacional para compatibilizar registros de ocorrências com caixas de testes de relés. UFPE, n. 1, p. 5, 2016.
- BERNARDON ROBERTO RECH, F. D. V. L. T. d. L. M. D. P. Modelagem e aplicação do programa atp para estudos de paralelismo nas redes de distribuição atendidas por subestações de diferentes fontes. XVII Seminário Nacional de distribuição de Energia Elétrica Belo Horizonte/MG, p. 9, 2005.
- ELÉTRICA, P. d. D. de E. no sistema elétrico nacional—prodíst. *Agência Nacional de Energia Elétrica*, 2013.
- GAMMA, E. et al. Elements of reusable object-oriented software. *Reading: Addison-Wesley*, 1995.
- GARCÍA, V. V. *Caracterização de eventos transitórios da qualidade da energia elétrica utilizando sistemas inteligentes e processamento de sinais*. Tese (Doutorado) — Universidade de São Paulo, 2012.
- GIMENES, I. M. d. S.; HUZITA, E. H. M. Desenvolvimento baseado em componentes: conceitos e técnicas. *Rio de Janeiro: Ciência Moderna*, 2005.
- GRANADOS-LIEBERMAN, D. et al. Techniques and methodologies for power quality analysis and disturbances classification in power systems: a review. *IET Generation, Transmission & Distribution*, IET, v. 5, n. 4, p. 519–529, 2011.
- HADDAD, J. Energia elétrica: Conceitos, qualidade e tarifação. *Programa Nacional de Conservação de Energia Elétrica—PROCEL*, Rio de Janeiro, 2004.
- HAYKIN, S. S.; VEEN, B. V. *Sinais e sistemas*. [S.l.]: Bookman, 2001.
- HOPP, T. et al. A matlab gui for the analysis and reconstruction of signal and image data of a saft-based 3d ultrasound computer tomograph. *International Journal on Advances in Software*, Citeseer, v. 2, p. 11, 2009.
- INSTITUTO ASCENDE BRASIL. Qualidade no fornecimento de energia elétrica: Confiabilidade, conformidade e prestreza. São Paulo, p. 36, 2014.
- JOHNSON, R. E.; FOOTE, B. Designing reusable classes. *Journal of object-oriented programming*, v. 1, n. 2, p. 22–35, 1988.
- LEITE, A. F. Framework e padrões de projeto. Janeiro 2006. Disponível em: <<https://www.devmedia.com.br/frameworks-e-padrees-de-projeto/1111>>.
- MALLAT, S. G. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, leee, v. 11, n. 7, p. 674–693, 1989.

MANIKANDAN, M. S.; SAMANTARAY, S.; KAMWA, I. Detection and classification of power quality disturbances using sparse signal decomposition on hybrid dictionaries. *IEEE Transactions on Instrumentation and Measurement*, IEEE, v. 64, n. 1, p. 27–38, 2015.

MELO, M. O. B. C. de. Avaliação do impacto da qualidade de energia elétrica na produção industrial: proposta de metodologia. *Produto & Produção*, v. 9, n. 3, 2008.

MISITI, M. et al. Wavelet toolbox: for use with matlab. The Math Works, Inc., 1996.

OLIVEIRA, H. M. D. *Análise de sinais para engenheiros*. [S.l.]: Brasport, 2007.

PREE, W. et al. Active guidance of framework development. *Software-Concepts and tools*, Berlin: Springer International, c1994-c2000., v. 16, n. 3, p. 136, 1995.

REINALDO, F. A. F. et al. Definição e aplicação de um framework para desenvolvimento de redes neurais modulares e heterogêneas. Florianópolis, SC, 2003.

SHAFRANOVICH, Y. Common format and mime type for comma-separated values (csv) files. Copyright (C) The Internet Society (2005), p. 8, 2005.

SOCIETY, I. P. E. Standard commom format for transient data exchange. March 1999. Disponível em: <www.ieeexplore.ieee.org/document/159102/?reload=true>.