

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia Eletrônica

**Análise do algoritmo Iterative Interpolated
Discrete Fourier Transform para aplicação em
Unidades de Medição Fasorial**

Autor: Lucas Gonçalves Campos
Orientador: Dr. Anderson Jose Costa Sena

Brasília, DF
2023



Lucas Gonçalves Campos

**Análise do algoritmo Iterative Interpolated Discrete
Fourier Transform para aplicação em Unidades de
Medição Fasorial**

Monografia submetida ao curso de graduação em (Engenharia Eletrônica) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia Eletrônica).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Dr. Anderson Jose Costa Sena

Brasília, DF

2023

Lucas Gonçalves Campos

Análise do algoritmo Iterative Interpolated Discrete Fourier Transform para aplicação em Unidades de Medição Fasorial/ Lucas Gonçalves Campos. – Brasília, DF, 2023-

81 p. : il. (algumas color.) ; 30 cm.

Orientador: Dr. Anderson Jose Costa Sena

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2023.

1. i-IPDFT. 2. Phasor measurement unit. I. Dr. Anderson Jose Costa Sena. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Análise do algoritmo Iterative Interpolated Discrete Fourier Transform para aplicação em Unidades de Medição Fasorial

CDU 02:141:005.6

Lucas Gonçalves Campos

Análise do algoritmo Iterative Interpolated Discrete Fourier Transform para aplicação em Unidades de Medição Fasorial

Monografia submetida ao curso de graduação em (Engenharia Eletrônica) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia Eletrônica).

Trabalho aprovado. Brasília, DF, 01 de junho de 2013:

Dr. Anderson Jose Costa Sena
Orientador

Dr. Daniel Mauricio Muñoz Arboleda
Convidado 1

Dr. Cristiano Jacques Miosso
Convidado 2

Brasília, DF
2023

Agradecimentos

Agradeço a todos os que permitiram chegar até aqui. À minha família, que me deu todo o apoio quanto possível, e à qual um dia desejo devolver em dobro todo suporte e carinho que me foi dado. À meus amigos, os quais tornaram minha vida mais tranquila apenas através dos momentos que compartilhamos. A meus professores, sem os quais seria impossível ter obtido o conhecimento atual, através do trabalho digno que é lecionar, e em destaque a meu professor orientador, professor Anderson Costa, que me ajudou a lapidar o atual trabalho. E por fim, a todos aqueles que passaram por minha vida e contribuíram na construção do que sou hoje.

*“Nossas lealdades estão com as espécies e com o planeta.
Nós falamos pela Terra. Nossa obrigação de sobreviver e florescer pertence,
não só a nós mesmos, mas também ao Cosmos,
antigo e vasto, do qual surgimos.”
(Carl Sagan)*

Resumo

Este trabalho visa investigar o desempenho e comportamento do algoritmo i-IPDFT, tanto para sinais simples, com apenas um tom, quanto para sinais com ruído. Para tal, foram realizadas 4 simulações no total, com os algoritmos programados em Python e utilizando-se de Orientação a Objeto, de forma a permitir a realização do algoritmo em sinais artificiais com quatro parâmetros distintos: a frequência do tom principal f_0 , a frequência do tom ruidoso f_r , a amplitude do tom ruidoso A_r e a amplitude de ruído branco A_{wn} . Foram ajustados 4 parâmetros do algoritmo - iterações P , iterações Q , índice da frequência central K Central e frequência de amostragem f_r , buscando assim avaliar as melhores combinações de parâmetros quanto à medição dos indicadores de Erro de Frequência, Erro de Magnitude e Taxa de Acertos.

Os resultados destas simulações apontam um desempenho significativamente impactado pela frequência do tom principal do sinal, enquanto os parâmetros de iterações do algoritmo não influenciaram significativamente, ao menos dentro da faixa de valores estudada, nos resultados tanto com quanto sem ruído. Também foi exposto pelas simulações que o aumento da resolução da DFT ou aumento da frequência de amostragem não traduzem em melhoria significativa para eficácia ou precisão do algoritmo, enquanto a amplitude do tom ruidoso, assim como a proximidade de sua frequência em relação ao tom principal impactam significativamente na eficácia e precisão do algoritmo.

Os resultados deste trabalho também permitiram o ensejo de novas correntes e áreas a serem estudadas acerca do algoritmo, tal como o motivo e fundamentação do impacto da frequência de amostragem e resolução da Transformada de Fourier Discreta no algoritmo. Além disso, foi levantada a possibilidade de estudos acerca da interpolação do algoritmo Interpolated Discrete Fourier Transform, identificada no trabalho como fonte de erro no algoritmo e passível de otimização.

Palavras-chaves: *phasor measurement unit*. redes inteligentes. monitoramento de longa distância. *iterative interpolated DFT*.

Abstract

This work aims to investigate the performance and behavior of the I-IPDFT algorithm, both for simple signs, with only one tone and for noise signs. To this end, 4 simulations were performed in total, with the algorithms programmed in Python and using object orientation, in order to allow the algorithm in artificial signals with four distinct parameters: the frequency of the main tone F_0 , the frequency of the noisy tone F_R , the breadth of the noisy tone a_r and the white noise amplitude a_{wn} . Four parameters of the algorithm - iterations p , iterations q , central frequency index K central and sampling frequency f_r , were evaluated, thus seeking to evaluate the best combinations of parameters regarding the measurement of frequency error indicators, error of magnitude and rate of hits.

The results of these simulations point to a significantly impacted performance by the frequency of the signal tone, while the algorithm iterations parameters did not influence significantly, at least within the values studied, in both results with no noise. It has also been exposed by simulations that increased DFT resolution or increased sampling frequency do not translate into significant improvement for the efficacy or accuracy of the algorithm, while the breadth of the noisy, as well as the proximity of its frequency in relation to the main tone impacts significantly in the efficacy and accuracy of the algorithm.

The results of this work also allowed the rise of new currents and areas to be studied about the algorithm, such as the reason and reasoning of the impact of sampling frequency and resolution of the discrete Fourier transform in the algorithm. In addition, the possibility of studies on the interpolated interpolated discrete Fourier transform was raised, identified at work as a source of error in the algorithm and liable to optimization.

Key-words: phasor measurement unit. smart grid. wide area monitoring. iterative interpolated DFT.

Lista de ilustrações

Figura 1 – Diagrama de Blocos de uma PMU	32
Figura 2 – FS do sinal em estado estático fora da frequência nominal amostrado	35
Figura 3 – Evolução de EF Médio para F_0	54
Figura 4 – Evolução de EM Médio para F_0	54
Figura 5 – Evolução de Taxa de Acertos para F_0	55
Figura 6 – EF Médio para f_s	55
Figura 7 – EM Médio para f_s	56
Figura 8 – Taxa de Acertos para f_s	56
Figura 9 – Evolução de EF Médio para P	57
Figura 10 – Evolução de EM Médio para P	57
Figura 11 – Evolução de Taxa de Acertos para P	58
Figura 12 – Evolução de EF Médio para Q	58
Figura 13 – Evolução de EM Médio para Q	59
Figura 14 – Evolução de Taxa de Acertos para Q	59
Figura 15 – Evolução de EF Médio para K Central - 2ª Simulação	60
Figura 16 – Evolução de EM Médio para K Central - 2ª Simulação	60
Figura 17 – Evolução de Taxa de Acertos para K Central - 2ª Simulação	61
Figura 18 – Evolução de EF Médio para F_0 - 2ª Simulação	61
Figura 19 – Evolução de EM Médio para F_0 - 2ª Simulação	62
Figura 20 – Evolução de Taxa de Acertos para F_0 - 2ª Simulação	62
Figura 21 – Evolução de EF Médio para F_s - 2ª Simulação	63
Figura 22 – Evolução de EM Médio para F_s - 2ª Simulação	63
Figura 23 – Evolução de Taxa de Acertos para F_s - 2ª Simulação	64
Figura 24 – Evolução de EF Médio para P - 3ª Simulação	65
Figura 25 – Evolução de EM Médio para P - 3ª Simulação	65
Figura 26 – Evolução de Taxa de Acertos para P - 3ª Simulação	66
Figura 27 – Evolução de EF Médio para Q - 3ª Simulação	66
Figura 28 – Evolução de EM Médio para Q - 3ª Simulação	67
Figura 29 – Evolução de Taxa de Acertos para Q - 3ª Simulação	67
Figura 30 – Evolução de EM Médio para F_r - 3ª Simulação	68
Figura 31 – Evolução de EF Médio para F_r - 3ª Simulação	68
Figura 32 – Evolução de Taxa de Acertos para F_r - 3ª Simulação	69
Figura 33 – Evolução de EF Médio para A_r - 3ª Simulação	69
Figura 34 – Evolução de EM Médio para A_r - 3ª Simulação	70
Figura 35 – Evolução de Taxa de Acertos para A_r - 3ª Simulação	70
Figura 36 – Evolução de EF Médio para F_0 - 3ª Simulação	71

Figura 37 – Evolução de EM Médio para F_0 - 3ª Simulação	71
Figura 38 – Evolução de Taxa de Acertos para F_0 - 3ª Simulação	72
Figura 39 – Evolução de Taxa de Acertos para A_{wn} - 4ª Simulação	73
Figura 40 – Evolução de Taxa de Acertos para P - 4ª Simulação	74
Figura 41 – Evolução de Taxa de Acertos para Q - 4ª Simulação	74
Figura 42 – Evolução de EF Médio para A_{wn} - 4ª Simulação	75
Figura 43 – Evolução de EM Médio para A_{wn} - 4ª Simulação	75
Figura 44 – Evolução de EF Médio para P - 4ª Simulação	76
Figura 45 – Evolução de EM Médio para P - 4ª Simulação	76
Figura 46 – Evolução de EF Médio para Q - 4ª Simulação	77
Figura 47 – Evolução de EM Médio para Q - 4ª Simulação	77

Lista de tabelas

Tabela 1 – Tabela de janelas para filtros FIR	44
Tabela 2 – Tabela dos melhores resultados por parâmetros simulados para 1ª Simulação	53
Tabela 3 – Tabela dos melhores resultados por parâmetros simulados para 2ª Simulação	60
Tabela 4 – Tabela dos melhores resultados por parâmetros simulados para 3ª Simulação	64
Tabela 5 – Tabela dos melhores resultados por parâmetros simulados para 4ª Simulação	73

Lista de abreviaturas e siglas

SG	<i>Smart Grid</i>
FERC	<i>Federal Energy Regulatory Commission</i>
CTTCR	Controle de Tensão e de Tensão-Corrente Reativa
DFIR	Detecção de falha, isolamento e restauração
RD	Resposta à Demanda
RED	Recursos Energéticos Distribuídos
MLDCP	Monitoramento de Longa Distância, Controle e Proteção
DEI	Dispositivos Eletrônicos Inteligentes
SCADA	<i>Supervisory Control and Data Acquisition</i>
PMU	<i>Phasor Measurement Unit</i>
CPS	Centro de Proteção do Sistema
SAMS	Sistema de Automação e Monitoramento da Subestação
LAN	<i>local area network</i>
SMFS	Sistema de Medição Fasorial Sincronizada
FS	Fasor Sincronizado
UTC	<i>Coordinated Universal Time</i>
TVE	<i>Total Vector Error</i>
MTFMF	Medição de Frequência e Taxa de Mudança de Frequência
TMF	Taxa de Mudança de Frequência
FE	<i>Frequency Error</i>
ME	<i>Magnitude Error</i>
EF	Erro de Frequência
EM	Erro de Magnitude
RFE	<i>Rate of change of Frequency Error</i>

Lista de símbolos

ϕ	Ângulo de fase de Fasorial Sincronizado
θ	Ângulo de um sinal
σ	Fator de correção da frequência
A_0	Amplitude do tom principal
A_r	Amplitude do tom ruidoso
A_{wn}	Variância do ruído branco
f_0	Frequência do tom principal
f_r	Frequência do tom ruidoso
F_s	Frequência de amostragem do sinal
λ	Margem de ativação do algoritmo i-IPDFT
k	Índice da DFT
Q	Iterações do algoritmo i-IPDFT
P	Iterações do algoritmo e-IPDFT
Δf	Resolução da DFT

Sumário

	Introdução	23
I	REVISÃO BIBLIOGRÁFICA	27
1	PHASOR MEASUREMENT UNIT	29
1.1	Padrões IEEE	29
1.1.1	IEEE C37.118-2005	29
1.1.2	IEEE C37.118.1-2011	30
1.1.3	IEEE 60255-118-1-2018	31
1.2	Princípios de funcionamento	31
1.2.1	Frequência, TMF e SMFS segundo IEEE 60255-118-1-2018	32
1.2.2	Funcionamento em estado dinâmico e estático	34
1.2.2.1	Estado estático em frequência nominal	34
1.2.2.2	Estado estático em frequência não nominal	34
1.2.2.3	Estado dinâmico em fase e amplitude	35
1.2.2.4	TMF constante e diferente de zero	36
1.3	Classes de funcionamento	36
2	ALGORITMOS DE ANÁLISE DE SINAL	37
2.1	A transformada DFT	37
2.2	O algoritmo msDFT	38
2.3	O algoritmo lpDFT	39
2.4	O algoritmo e-lpDFT	40
2.5	O algoritmo i-lpDFT	41
3	FILTRO FIR	43
II	DESENVOLVIMENTO, DISCUSSÃO E RESULTADOS	45
4	METODOLOGIA DO TRABALHO	47
5	RESULTADOS	51
5.1	Execução das Simulações	51
5.1.1	Simulação 1	51
5.1.2	Simulação 2	51
5.1.3	Simulação 3	52

5.1.4	Simulação 4	52
5.2	Discussão dos Resultados	53
5.2.1	Simulação 1	53
5.2.2	Simulação 2	59
5.2.3	Simulação 3	64
5.2.4	Simulação 4	72
6	CONCLUSÃO	79
6.1	Conclusões sobre os resultados	79
6.2	Trabalhos Futuros	80
	REFERÊNCIAS	81

Introdução

Contexto

A Smart Grid (SG) é um conceito abrangente e que inclui uma série de tecnologias e funcionalidades para atender a seus mais diversos requisitos. Essencialmente, se trata de uma rede elétrica autônoma e com fornecimento de energia descentralizada, que virá a substituir o sistema tradicional de energia elétrica, permitindo um controle avançado da rede e uma série de novos serviços e otimizações, com a criação de novos mercados de energia.

Segundo [Fang et al. \(2012\)](#), a SG cria um fluxo duplo de informações e energia e, ao coletar, mesclar e processar essas informações, permite a distribuição eficiente de eletricidade e a capacidade de responder a várias circunstâncias e eventos, reunindo diversas funcionalidades.

Para [Berger e Iniewski \(2015\)](#), há cinco grandes áreas de desenvolvimento da Smart Grid, sendo elas o Monitoramento de Longa Distância, Controle e Proteção (MLDCP), o Controle de Tensão e de Tensão-Corrente Reativa (CTTR), a Gestão de resposta à demanda (RD), a Detecção de falha, isolamento e restauração (DFIR) e os recursos energéticos distribuídos (RED). Em destaque, o MLDCP é responsável pela coleta de informações de diversos parâmetros de estabilidade da rede e manutenção da mesma. Dentre os módulos necessários para a implementação do sistema, a *Phasor management unit* (PMU) é uma unidade indispensável.

A PMU é um dispositivo de medida baseado no Sistema de Medição Fasorial Sincronizado (SMFS), o qual é responsável por aferir a fase atual dos sinais encontrados na rede em relação a um sinal de referência na frequência nominal. A partir desses dados, é possível coletar informações como o deslocamento de fase da rede, a frequência e a Taxa de Mudança de Frequência (TMF), ou seja, a variação da frequência na rede. Esses dados são todos sincronizados e atrelados ao tempo em UTC em que foram medidos, de forma a sincronizar as medidas das PMUs entre si.

Os padrões IEEE, tais como o IEEE C37.118-2005, o IEEE C37.118.1-2011 e o IEEE 60255-118-1-2018, reúnem uma série de especificações, testes e definições acerca do desenvolvimento de PMUs. Neles, é definida a necessidade da pré-filtragem do sinal da rede e, em virtude disso, é proposto um filtro do tipo FIR para tal tarefa. O sinal então é convertido inicialmente por um conversor AD, filtrado pelo filtro para segregação da frequência útil do ruído e, em seguida, convertido para o SMFS, sendo então enviado para os computadores de monitoramento.

Entretanto, é proposto em [Derviškadić e Paolone \(2020\)](#) o algoritmo i-IpDFT (*iterative interpolated DFT*), o qual realizaria com eficiência a identificação da amplitude, frequência e fase do tom principal da rede com significativa precisão. Este algoritmo se baseia na Transformada Discreta de Fourier (DFT) e busca dirimir o efeito de distorção espectral causado pela transformação de uma frequência fora da resolução da transformada, além de oferecer mais precisão e robustez contra interferência espectral que outros algoritmos do tipo, como o IpDFT (*interpolated DFT*) ou o e-IpDFT (*enhanced interpolated DFT*).

Motivação

O principal motivo para o desenvolvimento do projeto se trata de sua vital importância no desenvolvimento de uma rede com MLDCP, que por sua vez é parte integrante de uma SG completa. A PMU é um componente essencial para implementação desse sistema, ao permitir a coleta de dados de frequência, TMF e fase da rede, dados estes necessários para a correta manutenção da estabilidade da mesma.

As PMUs conseguem coletar de maneira síncrona dados de diversas áreas da rede, que serão consolidadas pelas PDCs e enviadas para o centro de gerenciamento da rede. Com essa capacidade, é possível realizar um controle na ordem de segundos ([BERGER; INIEWSKI, 2015](#)) sobre a rede. Isso permite um avanço no quesito de confiabilidade, segurança, seletividade e robustez do sistema como um todo.

Dessa forma a aplicação do algoritmo i-IpDFT proporcionará uma alternativa eficiente e eficaz para PMUs quanto à aquisição dos dados de fase, amplitude e frequência. Além disso, o algoritmo permite robustez suficiente para atuar tanto em estado de monitoramento quanto de proteção, sem a necessidade de grandes modificações de *software* ou *hardware* do equipamento.

Entretanto, *a priori* de sua aplicação, é necessária a compreensão do funcionamento do algoritmo e do efeito da alteração de seus parâmetros em função de diferentes sinais de entrada. Tendo isso em vista, a pergunta que o trabalho busca solucionar é: qual o comportamento\desempenho do algoritmo i-IpDFT quanto à mudança de seus parâmetros internos a sinais mono-tonais e ruidosos?

Objetivos

O objetivo principal do trabalho é implementar o algoritmo i-IpDFT, assim como um ambiente de simulação para o teste deste, e aferir o desempenho deste algoritmo quanto à precisão das medições realizadas com um sinal mono-tonal e com ruído embutido. É

importante frisar que, para este trabalho, foram considerados apenas sinais em regime estático, não sendo de interesse buscar o comportamento em regime dinâmico.

Como objetivos específicos, é necessário:

1. Implementar ferramentas que permitam a produção de sinais de teste artificiais;
2. Implementar o algoritmo de maneira a permitir a mudança de seus parâmetros internos;
3. Desenvolver um ambiente de simulação que teste o algoritmo com parâmetros diferentes, em vista do melhor ajuste destes, e colete os resultados em vista dos indicadores propostos;

Divisão do Trabalho

O trabalho é dividido em 2 partes e 6 capítulos. Os capítulos 1, 2 e 3 tratam-se da primeira parte do trabalho, a Revisão Bibliográfica, a qual busca reunir informações teóricas sobre os conceitos envolvidos no atual trabalho. Já os capítulos 4, 5 e 6 tratam-se da segunda parte do trabalho, o Desenvolvimento, Discussão e Resultados, a qual apresenta a metodologia e os resultados obtidos da execução do trabalho proposto, além da discussão destes últimos.

Quanto à primeira parte do trabalho, o capítulo 1 expõe informações sobre a PMU, seu funcionamento e padrões os quais fundamentam seus requisitos. O capítulo 2 reúne a fundamentação teórica dos algoritmos de análise de sinal, relevantes para o entendimento do i -IpDFT. E por fim, o capítulo 3 agrega informações sobre o filtro FIR, assunto o qual é relevante para o entendimento da PMU padrão fornecida pelo IEEE.

Quanto à segunda parte do trabalho, o capítulo 4 expõe a metodologia utilizada para seu desenvolvimento, detalhando indicadores e especificidades das simulações. O capítulo 5 apresenta os resultados em detalhes de cada simulação realizada, assim como discussão acerca dessas informações obtidas. E por fim, o capítulo 6 compila os resultados e conclusões obtidos no capítulo 5, além de apresentar uma reflexão sobre possíveis melhorias e trabalho futuros possíveis de serem realizados dando seguimento ao assunto do atual trabalho.

Para concluir, como adendo ao texto deste documento, o Apêndice reúne todos os gráficos gerados pelas simulações realizadas, como forma de organizar melhor a estrutura do trabalho e complementar as informações expostas nos resultados.

Parte I

Revisão Bibliográfica

1 Phasor Measurement Unit (PMU)

A PMU, como citado anteriormente, trata-se do componente o qual afere as magnitudes e fases de corrente e tensão em determinados pontos da rede. Especificamente, segundo [Hojabri et al. \(2019\)](#), são dispositivos os quais trabalham com base na tecnologia de Sistema de Medição Fasorial Sincronizada (SMFS) - correspondente ao *synchrophasor* citado no artigo. As PMUs são capazes de estimar os fasores de maneira sincronizada, a frequência, e a taxa de mudança de frequência dos sinais de tensão e/ou corrente, com referência ao Tempo Universal Coordenado (UTC, do inglês *Coordinated Universal Time*). Para tal, usa-se a tecnologia do GPS (do inglês, *Global Positioning System*), o qual possui alta disponibilidade, alcance global e uma alta acurácia de tempo (na ordem de μs).

1.1 Padrões IEEE

Neste trabalho, serão resumidos três padrões IEEE em sequência histórica: IEEE C37.118-2005, IEEE C37.118.1-2011 e o padrão IEEE 60255-118-1-2018. O padrão IEEE 1344 é de 1995 e, embora não citado especificamente, ele é referenciado no subtópico referente ao padrão C37.118-2005.

1.1.1 IEEE C37.118-2005

De acordo com [Martin et al. \(2008\)](#), o padrão IEEE C37.118-2005 objetiva definir convenções de medida, prover um método de determinação da precisão das medidas, e propor requisitos para medição da performance sobre condições controladas referentes à SMFS. O padrão também introduz o conceito de PMU, assim como seus requisitos.

A PMU é significada neste documento como um dispositivo que produz, comunica e potencialmente armazena as medidas de Fasor Sincronizada (FS). As medidas de FS são caracterizadas como as representações de números complexos em uma frequência fundamental da tensão ou corrente, baseado em uma referência de tempo absoluta - a qual é fornecida por sinais de tempo sincronizados com o UTC por GPS. Assim, uma PMU cria uma representação, em determinado instante, de fasor de um sinal senoidal atrelado ao tempo UTC de captura desse sinal, determinando a fase entre o estado atual e a referência utilizada baseado na frequência base. Individualmente, essa representação não é significativa, mas quando há um conjunto de várias PMUs com a mesma referência de tempo, as diferenças de ângulos de fase são comparáveis e de maior acurácia. A PMU também pode, segundo o padrão, fornecer estimativas de frequência e taxa de mudança de frequência (df/dt).

O padrão também define uma medida de acurácia TVE (do inglês, *total vector error*) na auditoria das PMUs, representada pela Equação 1.1. Na Equação, $\vec{X}_{MEAS(k)}$ é o vetor aferido e \vec{X}_{IDEAL} é o vetor esperado, ambos considerados no instante k .

$$TVE_{(k)} = 100\% \cdot \frac{|\vec{X}_{MEAS(k)} - \vec{X}_{IDEAL(k)}|}{|\vec{X}_{IDEAL(k)}|} \quad (1.1)$$

O TVE pressupõe três fontes de erro: magnitude, ângulo e tempo. O padrão permite também dois níveis de auditoria para acurácia, os níveis 0 e 1. Ambos exigem TVE abaixo de 1%, mas o Nível 1 é mais restrito quanto as condições de teste. Entretanto, o padrão não especifica métodos de medida ou de outros fatores como amostragem, algoritmos ou sincronização. Quanto à frequência, não há no padrão medidas de acurácia para ela.

O padrão define um protocolo simples, com poucos tipos de mensagens e comandos, para a comunicação entre a PMU e o PDC. Se a PMU suporta comunicação em tempo real, é previsto 5 *frames* - tipos de mensagem, padrões: de dados, dois tipos de configuração, cabeçalho e um tipo de comando. As saídas das PMUs são formatadas como um inteiro de 16 bits ou um ponto flutuante de 32 bits.

O novo padrão possui diferenças quanto ao padrão IEEE 1344-1995. Embora possua os mesmos elementos básicos, no IEEE 1344 a descrição de FS é ambígua, fato corrigido no padrão IEEE C37.118, no qual a descreve de maneira mais completa. O novo padrão também apresenta mais informações e adiciona mais anexos para promover clareza e orientação quanto ao tópico.

1.1.2 IEEE C37.118.1-2011

Este padrão (IEEE . . . , 2011) veio por incrementar o padrão IEEE C37.118.1-2005. Ele é apenas uma das subdivisões ao qual o padrão anterior fora submetido, tratando sobre as medições e métricas, enquanto a outra subdivisão (IEEE C37.118.2) trata sobre comunicação de dados.

Neste padrão, foi adicionado mais informações acerca de fasores e FS. Os conceitos de TVE e testes de conformidade foram mantidos e expandidos, além de serem inseridos testes sobre variação de temperatura e performance dinâmica. Também foram inclusos limites e características sobre a Medição de Frequência e Taxa de Mudança de Frequência (MFTMF, tradução literal de ROCOF). Seu anexo C fornece um modelo para verificar a implementação das medidas de performance requeridas.

O padrão estabelece, além do TVE, medidas de acurácia nomeadas *frequency error* (FE) - erro de frequência, e a *rate of change of frequency error* (RFE) - erro da taxa de

mudança da frequência, assim como seus limites de tolerância. Tanto a FE quanto a RFE são definidas pelas Equações 1.2 e 1.3.

$$FE = |f_{true} - f_{meas}| \quad (1.2)$$

$$RFE = \left| \left(\frac{df}{dt} \right)_{true} - \left(\frac{df}{dt} \right)_{meas} \right| \quad (1.3)$$

Na Equação 1.2, f_{true} representa a frequência real, enquanto f_{meas} representa a frequência medida. Já na Equação 1.3, $(df/dt)_{true}$ representa a Taxa de Mudança de Frequência (TMF) real, enquanto $(df/dt)_{meas}$ representa TMF medida.

Nesse padrão são considerados dois níveis de conformidade: classe P e classe M. A primeira classe serve para aplicações que não necessitam de respostas rápidas, enquanto a segunda não necessita de uma resposta rápida e pode ser afetada por sinais adversos.

1.1.3 IEEE 60255-118-1-2018

Este padrão (IEEE/IEC..., 2018) foi o mais recente acerca deste tópico. Ele continua a série de padrões C37.118, fornecendo mais detalhes acerca dos parâmetros de conformidade e detalhes técnicos das PMUs, enquanto mantém as determinações dos padrões anteriores.

Um aditivo deste padrão é a inclusão da definição de alguns comportamentos transitórios, como *overshot* - extrapolação do limite superior da tolerância devido a comportamento transitório antes e depois de reposta ao degrau (valor de pico), e *undershot* - extrapolação do limite inferior da tolerância devido a comportamento transitório antes e depois de reposta ao degrau. O padrão busca oferecer uma análise mais completa das PMUs sobre esses efeitos.

Além de fornecer parâmetros de conformidade para a resposta transitória, o padrão também oferece parâmetros para conformidade do comportamento dinâmico, como largura de banda, resposta à rampa em frequência, reposta ao degrau em fase e magnitude e latência do reporte (latência do envio de dados ao PDC).

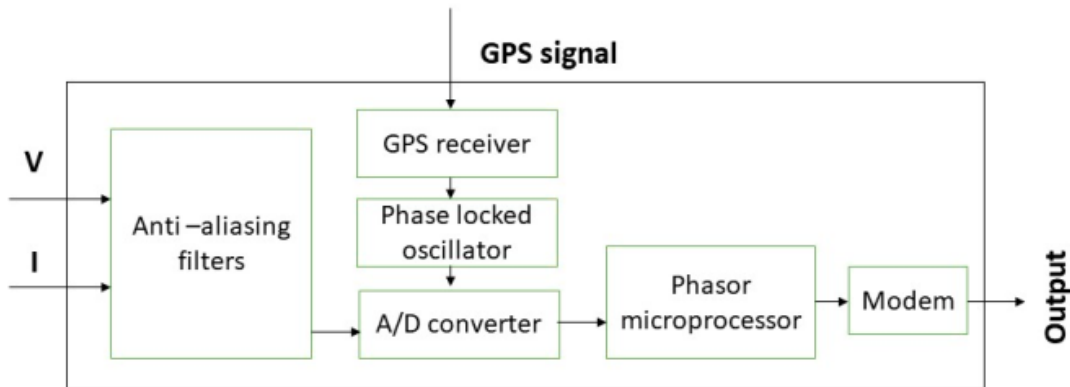
Por fim, este padrão fornece uma lista com as informações necessárias para toda documentação sobre PMUs, para comprovação da conformidade em relação ao padrão IEEE 60255-118-1-2018.

1.2 Princípios de funcionamento

Como já definido anteriormente, a PMU coleta dados da rede perante o uso de SMFS e reporta esses dados à PDC, resumidamente. A Figura 1 apresenta um diagrama

para as funções da PMUs a partir de uma perspectiva de maior abstração.

Figura 1 – Diagrama de Blocos de uma PMU



Fonte: (HOJABRI et al., 2019)

Os filtros Anti-aliasing (*Anti-aliasing filters*) são filtros cuja função é evitar o fenômeno de *aliasing* consoante à filtragem de frequências que estão fora do escopo da análise das PMUs e que possam ultrapassar a frequência de Nyquist. O sinal do GPS fornece o tempo UTC para adicionar a *time-tag* ao *Phase locked oscillator* embutido no sistema, o qual é responsável pela criação da frequência de referência para a criação dos FSs. Após a conversão para o domínio digital no conversor analógico-digital *A/D converter*, ambos os sinais, o sinal capturado e o sinal de referência, são enviados para o *Phasor microprocessor*, o qual será responsável por produzir os FSs e transferi-los para o modem, este último sendo o que enviará o sinal para o PDC.

1.2.1 Frequência, TMF e SMFS segundo IEEE 60255-118-1-2018

O padrão IEEE 60255-118-1-2018 (IEEE/IEC..., 2018) fornece uma ótima base acerca da matemática que descreve a frequência, a TMF e o SMFS. A Equação 1.4 modela um sinal de tensão ou corrente genérico na rede.

$$x(t) = X_m(t) \cos[\theta(t)] + D(t) \quad (1.4)$$

Nesta equação, $X_m(t)$ é a magnitude de pico do sinal, ω representa a posição angular do sinal em radianos, t é o tempo em segundos - o qual coincide com o UTC, e $D(t)$ é o sinal de ruído, o qual advém de várias possíveis fontes.

O ângulo de fase do FS é definido pela Equação 1.5.

$$\phi(t) = \theta(t) - 2\pi f_0 t \quad (1.5)$$

Na equação, f_0 representa a frequência nominal do sistema, ou seja, 50 Hz ou 60 Hz. Dessa forma, percebe-se que o ângulo de fase do FS se trata da diferença entre a

frequência medida e a frequência nominal do sistema. O ângulo de fase de FS é medido em radianos (rad).

Um fator interessante é que, embora a frequência do sinal seja de 60 Hz, essa forma produz um sinal que tende a zero quando próximo à frequência nominal. O resultado desta propriedade é a necessidade de uma menor taxa de amostragem, cerca de 10 Hz - desde que se tenha um filtro anti *aliasing* adequado. Essa taxa de amostragem é suficiente, por exemplo, para recuperar o sinal de uma rede com frequências entre 55 e 65 Hz com 60 Hz de frequência nominal.

Quanto ao FS, por ser um número complexo, pode ser representado em forma polar (Equação 1.6) ou em forma retangular (Equação 1.7).

$$X_P(t) = \left(\frac{X_m(t)}{\sqrt{2}}, \phi(t) \right) \quad (1.6)$$

$$X_r(t) = \left(X_r(t), X_i(t) \right) \quad (1.7)$$

Na Equação 1.7, X_r e X_i representam, respectivamente, a parte real e imaginária de FS. As Equações 1.8 e 1.9 apresentam seus respectivos valores.

$$X_r(t) = \frac{X_m(t)}{\sqrt{2}} \cos[\phi(t)] \quad (1.8)$$

$$X_i(t) = \frac{X_m(t)}{\sqrt{2}} \sin[\phi(t)] \quad (1.9)$$

A frequência é definida pela Equação 1.10.

$$f(t) = \frac{1}{2\pi} \frac{d\theta(t)}{dT M t} = f_0 + \frac{1}{2\pi} \frac{d\phi(t)}{dt} \quad (1.10)$$

Da Equação 1.10, percebe-se que a frequência é definida como a taxa de variação em relação ao tempo do ângulo de fase do FS somado à frequência nominal. A unidade de medida da frequência é o Hertz (Hz).

Quanto a TMF, esta pode ser obtida conforme a expressão da Equação 1.11.

$$\text{TMF}(t) = \frac{df(t)}{dt} = \frac{1}{d\pi} \frac{d^2\theta(t)}{dt^2} = \frac{1}{2\pi} \frac{d^2\phi(t)}{dt^2} \quad (1.11)$$

Da Equação 1.11, percebe-se que a TMF é definida como a taxa de variação da frequência, ou seja, a derivada segunda do ângulo de fase do FS. A unidade de medida da TMF é o Hertz por segundo (Hz/s).

1.2.2 Funcionamento em estado dinâmico e estático

O padrão (IEEE/IEC..., 2018) determinou quatro regimes de funcionamento para PMU e a saída em FS para cada um deles: estado estático na frequência nominal, estado estático em frequência não-nominal, estado dinâmico em fase e amplitude e TMF constante e diferente de zero. Para exemplificação desses regimes, foi utilizado o sinal da fórmula descrita na Equação 1.4. A frequência nominal está definida como f_0 .

A reconstrução do sinal em qualquer estado pode ser feito a través da Equação 1.12.

$$x(t) = X_m(t) \cos[2\pi f_0 t + \phi(t)] = \sqrt{2}X_r(t) \cos(2\pi f_0 t) - \sqrt{2}X_i(t) \sin(2\pi f_0 t) \quad (1.12)$$

1.2.2.1 Estado estático em frequência nominal

Para o estado estático em f_0 , todos os parâmetros são constantes, com $f(t) = f_0$, $X_m(t) = \frac{X_m}{\sqrt{2}}$, $\phi(t) = \phi$, $\text{TMF}(t) = 0$ e $D(t) = 0$. Nesse caso, o FS será representado pela Equação 1.13.

$$X(t) = \frac{X_m}{\sqrt{2}} \cdot e^{j\phi} \quad (1.13)$$

Na equação, ϕ é a diferença de fase inicial entre o sinal aferido e o sinal de referência. Ou seja, o FS é constante em estado estático na frequência nominal.

1.2.2.2 Estado estático em frequência não nominal

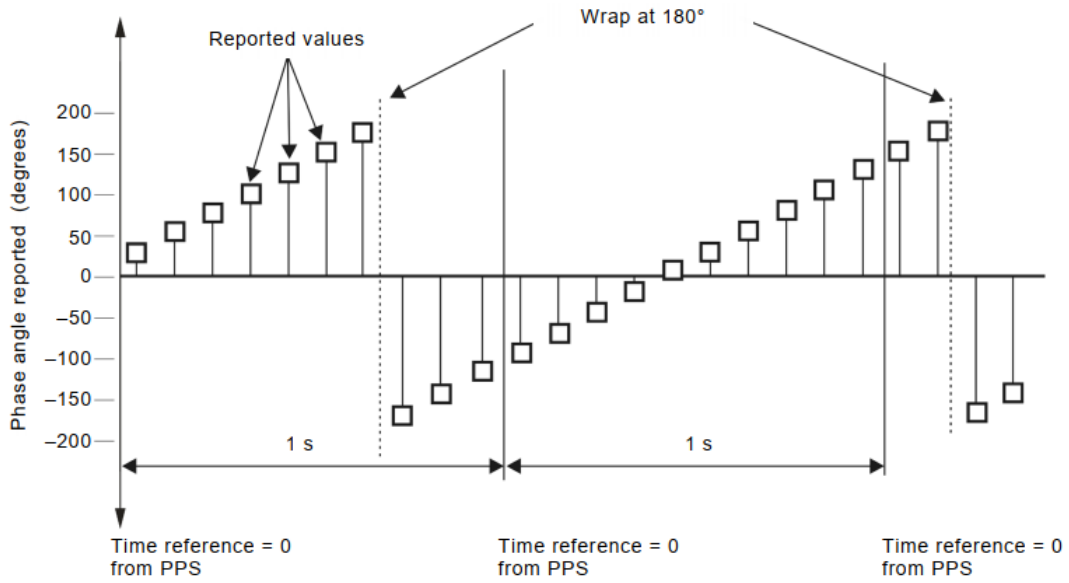
Nesse caso, os parâmetros são constantes, com $f(t) = f_0 + \Delta f$, $X_m(t) = \frac{X_m}{\sqrt{2}}$, $\phi(t) = \phi$, $\text{TMF}(t) = 0$ e $D(t) = 0$. Para esse caso, o FS possui representação expressa na Equação 1.14.

$$X(t) = \frac{X_m}{\sqrt{2}} \cdot e^{j(\phi\pi\Delta f t + \phi_0)} \quad (1.14)$$

O componente $\phi\pi\Delta f t + \phi_0$ representa $\theta(t)$, obtido a partir da integração $\int_0^t 2\pi f(t) dt$. Na equação, ϕ_0 representa a fase inicial do sinal, enquanto Δf representa a diferença entre a frequência do sinal e a frequência nominal.

Percebe-se, para este caso, que o o FS medido variará em torno de ϕ_0 numa frequência equivalente à diferença entre a frequência nominal e a frequência aferida. Isso justifica taxas menores de amostragem, como citado na seção anterior. A Figura 2 demonstra o comportamento do FS amostrado nesse estado.

Figura 2 – FS do sinal em estado estático fora da frequência nominal amostrado



Fonte: (IEEE/IEC... , 2018)

1.2.2.3 Estado dinâmico em fase e amplitude

O sinal considerado para o exemplo é expresso pela Equação 1.15, tratando-se de um sinal modulado em amplitude e fase.

$$x(t) = X_m[1 + k_x \cos(2\pi f_m t)] \cdot \cos[2\pi f_0 t + k_a \cos(2\pi f_m t - \pi)] \quad (1.15)$$

Nesse caso, f_m é a frequência de modulação em Hz, k_x é o índice de modulação da amplitude e k_a é o índice de modulação de fase. Para esse caso, $x_m(t) = X_m[1 + k_x \cos(2\pi f_m t)]$, $\phi(t) = k_a \cos(2\pi f_m t - \pi)$ e $D(t) = 0$. A representação do FS é dado pela Equação 1.16.

$$X(t) = \frac{X_m}{\sqrt{2}} [1 + k_x \cos(2\pi f_m t)] e^{j[k_a \cos(2\pi f_m t - \pi)]} \quad (1.16)$$

As Equações 1.17 e 1.18 representam, respectivamente, a frequência e o MTF nesse estado do sinal.

$$f(t) = f_0 + \frac{1}{2\pi} \frac{d[k_a \cos(2\pi f_m t - \pi)]}{dt} = f_0 - k_a f_m \sin(2\pi f_m t - \pi) \quad (1.17)$$

$$\text{TMF}(t) = \frac{df(t)}{dt} = -k_a 2\pi f_m^2 \cos(2\pi f_m t - \pi) \quad (1.18)$$

Neste estado, tanto a frequência quanto a TMF oscilam de acordo com a frequência de modulação f_m , enquanto o FS oscila de acordo com a magnitude e fase. Caso $k_a = 0$,

a frequência e TMF serão constantes, ou seja, elas oscilam apenas para $k_a \neq 0$. Para este exemplo, pressupõe-se $f_m < f_0$, $k_x < 1$ e $k_a < 0$, de forma a evitar soluções estranhas e melhor representar o comportamento real nas aplicações de PMUs.

1.2.2.4 TMF constante e diferente de zero

Nesse estado de funcionamento, $\text{TMF}(t) = R_f$, sendo R_f contante e diferente de zero, e $X_m(t) = \frac{X_m}{\sqrt{2}}$. A Equação 1.19 formula a frequência nesse estado.

$$f(t) = \int_0^t R_f d\tau = R_f t + f_a \quad (1.19)$$

A frequência f_a é a frequência em $t = 0$. Percebe-se um comportamento linear da frequência para este caso. O ângulo de fase FS nestas circunstâncias é expresso pela Equação 1.20.

$$\phi(t) = 2\pi \int_0^t (R_f \tau + f_a) d\tau - 2\pi f_0 t = \pi R_f t^2 + 2\pi(f_a - f_0)t + \phi_0 \quad (1.20)$$

Neste caso, ϕ_0 representa o ângulo de fase inicial do FS. Dessa forma, o FS desse estado é representado pela Equação 1.21.

$$X(t) = \frac{X_m}{\sqrt{2}} e^{j[\pi R_f t^2 + 2\pi(f_a - f_0)t + \phi_0]} \quad (1.21)$$

Percebe-se que há um incremento parabólico na frequência, a uma taxa de $\pi R_f t^2 + 2\pi(f_a - f_0)t$. Quanto a frequência, esta é uma rampa linear.

1.3 Classes de funcionamento

O padrão IEEE 60255-118-1-2018, tal como o padrão IEEE C37.118.1-2011, prevê dois níveis de conformidade distintos, as classes P e M. A Classe P é requerida para aplicações de resposta rápida, enquanto a Classe M é útil em aplicações as quais podem ser afetadas por sinais fora de banda, mas requerem menores tempos de resposta.

2 Análise de sinal

2.1 A transformada DFT

Uma transformada trata-se da representação de um sinal pela somatória de um conjunto de funções, o qual nomeia-se base, de maneira ponderada. No caso da DFT, esta trata-se de uma transformada de sinais discretos para outro domínio discreto, cujas funções da base são exponenciais imaginárias. Se colocarmos a base como \mathbf{B} , então temos:

$$\mathbf{B} := \left\{ \exp\left(\frac{j2\pi kn}{N}\right) \right\}, \forall k \in [0, 1, \dots, N-1] \quad (2.1)$$

Na formulação dada, N representa o total de amostras do sinal. Com a base definida, a obtenção da transformada de um sinal para a respectiva base é realizada conforme o produto interno desse sinal com a base. Essa operação permite a obtenção dos coeficientes para cada elemento da base. Assim, dado um coeficiente \mathbf{C}_k da base e um sinal genérico $x(n)$ discreto, têm-se:

$$\mathbf{C}_k = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \cdot \exp\left(-\frac{j2\pi kn}{N}\right) \quad (2.2)$$

A exponencial torna-se negativa pois trata-se do conjugado do elemento da base em questão. Quanto ao valor $\frac{1}{N}$, trata-se do fator de normalização do sinal. Com os respectivos coeficientes da base, é possível visualizar o gráfico em uma dimensão transformada, que no caso da DFT nomeia-se dimensão da frequência. Também é possível recuperar o sinal consoante o processo de transformada inversa, o qual se trata da simples multiplicação da base pelos seus coeficientes:

$$x[n] = \sum_{k=0}^{N-1} \mathbf{C}_k \cdot \exp\left(\frac{j2\pi kn}{N}\right) \quad (2.3)$$

Convenciona-se que o coeficiente \mathbf{C}_k seja representado por $X[k]$, sendo então X a DFT do sinal $x[n]$, formado por todos os índices $X[k] = \mathbf{C}_k$.

É importante citar algumas características da DFT que serão utilizadas neste trabalho:

- O trecho da DFT de $\frac{N}{2} < k < N$ equivale à parte negativa da transformada;
- Em razão do item anterior, o trecho da segunda metade equivale ao conjugado do primeiro trecho para números reais;

- A DFT de um sinal se relaciona com sua DTFT (*Discrete Time Fourier Transform* - em tradução literal, Transformada Discreta de Fourier em Tempo Discreto) conforme a fórmula $X_{DFT}[k] = X_{DTFT}(\frac{k}{N})$;
- O número de amostras da transformada depende do período (T) da amostra do sinal e de sua frequência de amostragem (f_s), segundo a relação $N = F_s \cdot T$;
- O período de amostragem do sinal reflete na resolução da transformada (Δf): quanto maior o tempo de amostragem, menor o intervalo entre as frequências representadas pelos índices k da transformada, ou seja, maior sua resolução;
- O cálculo de Δf é dado por $\Delta f = \frac{F_s}{N} = \frac{1}{T}$;
- Para a primeira metade da transformada, a frequência representada pelo índice k é igual ao índice vezes a resolução da transformada ($f_k = \Delta f \cdot k$, $\forall k \in [0, 1, \dots, \frac{N}{2}]$);
- Para a segunda metade da transformada, a frequência representada pelo índice k é igual ao índice menos N vezes a resolução da transformada ($f_k = \Delta f \cdot (k - N)$, $\forall k \in [\frac{N}{2}, \frac{N}{2} + 1, \dots, N]$);
- Caso a frequência do sinal amostrado não seja múltipla de Δf , o sinal sofrerá da chamada distorção espectral - o sinal será representado como sendo composto por mais de uma frequência, mesmo tendo frequência única.

2.2 O algoritmo msDFT

O algoritmo msDFT trata-se de um método, segundo [Romano e Paolone \(2014b\)](#), o qual visa atualizar os valores de cada impulso da transformada em tempo real. Este método trata-se de um algoritmo recursivo, o qual se destaca dos demais por garantir estabilidade sem sacrificar acurácia e por ser resiliente ao acúmulo de erros numéricos provenientes do arredondamento existente na representação digital.

Para a descrição do algoritmo, pressuponha um sinal cujo componente principal ocorre em torno de uma frequência f_0 e cujo período de amostragem é $T_s = 1/F_s$. A descrição matemática deste sinal é dada abaixo:

$$x[m] = A_0 \cos(2\pi f_0 m T_s + \phi_0), \quad m \in [0, 1, \dots, M - 1] \quad (2.4)$$

Neste sinal, A_0 é a amplitude, f_0 é a frequência principal, T_s é o intervalo amostral, ϕ_0 é a fase e M é o número de amostras do sinal janelado. Para este sinal, pressupõe-se que a frequência é constante dentro de uma janela de observação M . Para cada passo de tempo m , a DFT do sinal pode ser atualizada para um novo conjunto de amostras na janela:

$$X_k[n] = \sum_{m=0}^{M-1} x[q+m] \cdot W_M^{-km} \quad (2.5)$$

Nesta Equação, têm-se que $q = n - M + 1$ é o índice da amostra mais antiga dentro da janela avaliada, m é o índice das amostras janeladas e $W_M^{-km} = \exp(-j2\pi km/M)$. É possível reformular a Equação 2.5 na forma da Equação 2.6 descrita abaixo:

$$X_k[n] = W_M^k \cdot (X_k[n-1] - x[n-M] + x[n]) \quad (2.6)$$

A fórmula recursiva proposta acima é potencialmente instável e propensa ao acúmulo de erro gerado pelo arredondamento numérico. Entretanto, é notar que, quando $k = 0$, a formulação não envolve a função de base (W_M^k) e, portanto, é por definição estável:

$$X_0[n] = X_0[n-1] - x[n-M] + x[n] \quad (2.7)$$

Se aproveitando dessa propriedade e utilizando-se da propriedade de modulação da DFT, é possível alcançar a fórmula da Equação 2.8 - retirada de [Orallo et al. \(2012\)](#), na qual qualquer índice k pode ser deslocado para $k = 0$ multiplicando o sinal de entrada pela função de base:

$$X_k[n] = X_k[n-1] + W_M^{-km} \cdot (-x[n-M] + x[n]) \quad (2.8)$$

Dessa forma, a realização iterativa da Equação 2.8 junta à atualização do índice m a cada iteração é o que caracteriza o algoritmo msDFT. Com isso, é possível realizar a atualização de um único índice k do espectro DFT, sendo necessário a repetição desse algoritmo para cada índice da DFT incluso na janela avaliada.

2.3 O algoritmo IpDFT

Segundo [Derviškadić e Paolone \(2020\)](#), algoritmo IpDFT trata-se de um algoritmo o qual estima parâmetros de uma onda senoidal, no caso sua amplitude, frequência e fase. Ele se propõe, além disso, a mitigar os efeitos de uma amostragem incoerente - quando a frequência principal não é múltipla da frequência de amostragem.

O algoritmo executa a DFT do sinal e em seguida realiza o janelamento do sinal para a frequência em análise. No caso de uma amostragem incoerente, o pico da DTFT da frequência principal é localizada entre duas amostras da DFT, com a representação em frequência sendo expressa por:

$$f_0 = (k_m + \sigma)\Delta f \quad (2.9)$$

Na Equação 2.9 f_0 é a frequência principal, k_m é o índice da maior amostra, $-0,5 \leq \sigma < 0,5$ é o termo de correção e $\Delta f = 1/T$, ou seja, a frequência de resolução do sinal.

O cálculo do termo de correção é feito em razão do tipo e tamanho do janelamento adotado. No caso deste trabalho, usar-se-á o janelamento de Hanning e uma janela de três amostras, o que permite a computação do termo σ pela interpolação dessas três amostras adotadas. A expressão abaixo expressa o cálculo de σ , adaptada do artigo de [Derviškadić e Paolone \(2020\)](#):

$$\sigma = 2 \cdot \frac{|X[k_m + 1]| - |X[k_m - 1]|}{|X[k_m - 1]| + 2|X[k_m]| + |X[k_m + 1]|} \quad (2.10)$$

A partir do valor de σ , é possível alcançar os valores da amplitude (A_0) e fase (ϕ_0) segundo as seguintes fórmulas:

$$A_0 = |X[k_m]| \left| \frac{\pi\sigma}{\sin(\pi\sigma)} \right| |\sigma^2 - 1| \quad (2.11)$$

$$\phi_0 = \angle X[k_m] - \pi\sigma \quad (2.12)$$

SMFSs baseados em IpDFT possuem taxas de amostragem na ordem de dezenas de kiloHertz e janelas de observação pequenas. Entretanto, o processo está sujeito a vários problemas de processamento de sinais, tal como a distorção amostral (*spectral leakage*) e a interferência espectral, as quais são o motivo do desenvolvimento de métodos mais robustos quanto a estes problemas, sendo eles o e-*IpDFT* e o i-*IpDFT*.

2.4 O algoritmo e-*IpDFT*

O algoritmo e-*IpDFT* possui o mesmo objetivo do algoritmo *IpDFT*, entretanto consegue reduzir a interferência espectral - interferência cruzada existente entre as várias ondas (tons) compondo o espectro de frequência, neste caso causada pela imagem negativa do sinal em análise.

Segundo [Derviškadić e Paolone \(2020\)](#) e [Romano e Paolone \(2014a\)](#), o algoritmo aplica preliminarmente o *IpDFT* ao sinal, obtendo assim os parâmetros do tom principal. Com estes parâmetros, permite-se a separação do índice $X(k)$ em uma componente positiva e outra negativa, como descrito abaixo:

$$X[k] = X_0^+[k] + X_0^-[k] = A_0 \exp(+j\phi_0)W[k - f_0/\Delta f] + A_0 \exp(-j\phi_0)W[k + f_0/\Delta f] \quad (2.13)$$

Na Equação 2.13, f_0 , A_0 e ϕ_0 são respectivamente a frequência, amplitude e fase do tom principal, obtidos pela aplicação do IpDFT no sinal. $W(k)$ trata-se da transformada de Fourier da janela de Hanning, expresso por:

$$W[k] = -0,25 \cdot D_N[k - 1] + 0,5 \cdot D_N[k] - 0,25 \cdot D_N[k + 1] \quad (2.14)$$

Nesta Equação, $D_N[k]$ se trata da transformada de Fourier de uma janela retangular (Núcleo de Dirichlet):

$$D_N[k] = \exp(-j\pi k(N - 1)/N) \cdot \frac{\sin(\pi k)}{\sin(\pi k/N)} \quad (2.15)$$

Diante do exposto, define-se que a componente negativa é dada por:

$$X_0^-[k] = A_0 \exp(-j\phi_0)W[k + f_0/\Delta f] \quad (2.16)$$

Assim, basta a subtração do X_0^- da DFT original para se obter a componente positiva do sinal:

$$X_0^+[k] = X[k] - X_0^-[k] \quad (2.17)$$

Por fim, basta a aplicação do algoritmo IpDFT novamente para se obter os parâmetros do sinal sem a interferência da imagem negativa. Caso necessário, esse processo pode ser iterado com os novos parâmetros obtidos de forma a diminuir ainda mais a influência da componente negativa.

2.5 O algoritmo i-IpDFT

O algoritmo e-IpDFT não funciona para minimizar interferência espectral causada por tons diferentes do principal. Para este caso, o algoritmo i-IpDFT foi especificado para tratar de interferência de ruído próximo ao tom principal, além da interferência causada pela imagem negativa do tom principal.

O algoritmo funciona comparando a energia espectral do sinal reconstruído a partir dos parâmetros obtidos da aplicação do e-IpDFT com a energia espectral do sinal janelado original. Se a razão entre eles for superior a um valor λ , então o algoritmo busca obter uma aproximação dos parâmetros do ruído próximo. Isso é feito conforme a reaplicação

do e-*IpDFT* à subtração do sinal janelado original pela função do tom principal, o que permite obter os parâmetros do tom ruidoso, que é então reconstruído e subtraído do sinal janelado original, para então ser reaplicado o e-*IpDFT* para obtenção dos parâmetros do tom ruidoso. No final, é feita a subtração do tom ruidoso do sinal janelado original e, novamente, é aplicado o e-*IpDFT* para obtenção do tom principal sem o tom ruidoso. O processo é iterado uma certa quantidade de vezes para dirimir ao máximo a interferência do tom ruidoso.

A função que reconstrói o tom original se trata da Equação 2.13. Usando os parâmetros obtidos pela aplicação do e-*IpDFT*, essa equação é usada para reconstruir o tom original e validar a existência de um sinal ruidoso no sinal janelado original.

O pseudocódigo abaixo, obtido a partir de [Derviškadić e Paolone \(2020\)](#), define o funcionamento do algoritmo:

```

1   $X[k] = \text{DFT}(x[n])$ 
2   $X[k] = \text{Hanning}(X[k])$ 
3   $\{f_0^0, A_0^0, \phi_0^0\} = \text{e-}IpDFT(X[k])$ 
4   $X_0^0 = \text{Rec}(\{f_0^0, A_0^0, \phi_0^0\})$ 
5  if  $\sum |X[k] - X_0^0[k]|^2 > \lambda \cdot \sum |X[k]|^2$ 
6      for  $q \rightarrow Q$ 
7           $\{f_r^q, A_r^q, \phi_r^q\} = \text{e-}IpDFT(X[k] - X_0^{q-1}[k])$ 
8           $X_r^q[k] = \text{Rec}(\{f_r^q, A_r^q, \phi_r^q\})$ 
9           $\{f_0^q, A_0^q, \phi_0^q\} = \text{e-}IpDFT(X[k] - X_r^{q-1}[k])$ 
10          $X_0^q = \text{Rec}(\{f_0^q, A_0^q, \phi_0^q\})$ 
11     end for
12 else
13     break
14 end if

```

No pseudocódigo, a função *Rec* se trata da fórmula da Equação 2.13, Q é o número de iterações programadas para o algoritmo e X_r se trata do sinal ruidoso o qual se deseja dirimir.

3 Filtro FIR

A funcionalidade de um filtro de sinais é segregar informações úteis para determinada aplicação das informações consideradas inúteis para ela, estas últimas sendo chamadas de ruído. Os filtros podem ser discretos ou contínuos, e geralmente realizam sua filtragem no domínio da frequência.

O *Finite Impulse Response* (FIR) é um filtro de sinais discretos, o qual é capaz de segregar informações contidas em uma faixa de frequência específica de um sinal do ruído. Este filtro, como o próprio nome diz, trabalha com um número finito de elementos do sinal, necessitando então de um janelamento adequado do sinal para diminuir os efeitos de distorção (*ripple*) gerados pela "recorte" do sinal. Suas principais características são:

1. Possui como produto do projeto a resposta ao impulso do filtro;
2. Não possui polos em sua função de transferência;
3. Possui fase linear e, portanto, não possui distorção de fase;
4. Produz um *ripple* máximo dependente exclusivamente do tipo de janelamento utilizado.

Dada as características acima, o filtro FIR possui como principal vantagem a menor distorção do sinal, além de poder ser configurado para não possuir distorção de fase. Entretanto, sua faixa de transição entre a banda passante - faixa em frequência que deseja-se preservar, e banda de rejeição - faixa de frequência a qual deseja-se desprezar, são mais lentas, tornando o filtro inadequado para aplicações em tempo real.

O projeto de filtros FIR possui 5 passos de especificação:

1. Cálculo da resposta ideal;
2. Escolha da janela;
3. Escolha do tamanho da janela;
4. Cálculo da resposta janelada do filtro;
5. Atraso da resposta do filtro.

O primeiro passo é realizado em consonância com a banda passante e banda de rejeição requisitada. Ela é calculada consoante o uso da DTFT inversa da resposta ideal,

usando geralmente a função sinc no tempo, a qual é uma função retangular na frequência. No segundo passo, escolhe-se uma janela específica, baseado nos requisitos de atenuação na banda de rejeição e tamanho da banda de transição (intervalo de banda entre a banda de rejeição e banda passante). No terceiro passo é escolhido o tamanho da janela, sendo necessário um compromisso entre este (e subsequente tempo de processamento), o tamanho da banda de transição e a atenuação desejada na banda de rejeição. No quarto passo, é calculada a resposta janelada do filtro segundo a multiplicação entre a janela e a resposta ideal. Por fim, basta atrasar a resposta em metade do tamanho da janela, para tornar o filtro causal (em função apenas dos valores passados do sinal).

Há diversas janelas já calculadas para a especificação do sinal. Na Tabela 1 são expostas algumas das janelas mais utilizadas para a especificação de filtros FIR, traduzida de Mitra (2010).

Tabela 1 – Tabela de janelas para filtros FIR

Tipo de janela	Largura do lóbulo principal	Nível relativo do lóbulo lateral	Atenuação mínima da banda de rejeição	Tamanho da banda de transição
Retangular	$\frac{4\pi}{2M+1}$	13,3 dB	20,9 dB	$0,92 \frac{\pi}{M}$
Hann	$\frac{8\pi}{2M+1}$	31,5 dB	43,9 dB	$3,11 \frac{\pi}{M}$
Hamming	$\frac{8\pi}{2M+1}$	42,7 dB	54,5 dB	$3,32 \frac{\pi}{M}$
Blackman	$\frac{12\pi}{2M+1}$	58,1 dB	75,3 dB	$5,56 \frac{\pi}{M}$

Fonte:(MITRA, 2010)

Parte II

Desenvolvimento, Discussão e Resultados

4 Metodologia do Trabalho

A linguagem de programação utilizada para este trabalho se trata da linguagem Python. Optou-se por esta devido a disponibilidade de bibliotecas para tratamento e processamento de dados, tais como *Pandas* e *NumPy*, as quais facilitaram e agilizaram o desenvolvimento dos algoritmos e simulações. Também foi diferencial a possibilidade do uso do método de programação de orientação a objeto, o qual foi utilizado neste trabalho. O método de orientação a objeto foi escolhido dada a necessidade de reaproveitamento de trechos de código entre os algoritmos, outrossim objetivando uma melhor organização e modularização dos códigos.

Foi utilizada a plataforma *Jupyter Notebook* para o desenvolvimento dos códigos. A plataforma foi usada em vista da possibilidade do teste de trechos de códigos, além de permitir a visualização embutida dos gráficos e *prints* logo abaixo do compartimento de escrita destes trechos.

As simulações foram executadas visando o ajuste de quatro parâmetros: a quantidade de iterações do algoritmo e-IPDFT (P), a quantidade de iterações do algoritmo i-IPDFT (Q), a frequência de amostragem do sinal (F_s) e por fim a resolução do sinal na forma do índice k , o qual representa a frequência central ($K_{\text{Central}} = \frac{50}{\Delta f}$) do algoritmo. A proposta era encontrar a melhor combinação entre esses parâmetros, tendo em vista três indicadores de performance: EF (Erro de Frequência) Médio, EM (Erro de Magnitude) Médio e a Taxa de Acertos. Também foram considerados outros quatro parâmetros para as simulações, estes referentes às características do sinal de entrada: a frequência do tom principal (f_0), a frequência do tom ruidoso (f_r), a amplitude do tom ruidoso (A_r) e a amplitude\variância do ruído branco (A_{wn}). Estes últimos alteravam as condições de funcionamento do algoritmo, não tendo objetivo de ajuste, mas sim de comparação dos resultados.

As simulações foram realizadas com sinais de entrada artificiais. Os sinais mono-tonais foram produzidos utilizando-se da Equação 4.1. Nas simulações, ϕ_0 foi configurado como 0 para todas as simulações. Na equação $T_s = \frac{1}{f_s}$, e m são os valores do eixo do tempo, sendo $t = m \cdot T_s$.

$$A_0 \cos(2\pi f_0 m T_s + \phi_0) \quad (4.1)$$

Quanto aos sinais bi-tonais, estes foram produzidos conforme a Equação 4.2. Os sinais bi-tonais são basicamente a soma de dois sinais mono-tonais, um sinal dito ser o principal e outro sinal de ruído. Nas simulações ϕ_{i_0} e ϕ_{i_r} foram configurados como 0.

$$A_0 \cos(2\pi f_0 m T_s + \phi_0) + A_r \cos(2\pi f_r m T_s + \phi_r) \quad (4.2)$$

Por fim, os sinais com ruído branco foram construídos somando-se um sinal mono-tonal com um sinal aleatório uniforme, o ruído branco. A produção deste ruído foi realizada consoante a criação de um vetor aleatório uniforme, adaptado pela função $X_{wn}(n) = A_{wn} \cdot (U(n) - 0.5)$. No caso, U_n era o vetor uniforme produzido pela função *rand*, da biblioteca *Random* do Numpy. Como a função $U(n)$ produz um vetor uniforme de média 0.5 (gera valores aleatórios entre 0 e 1) e variância $\frac{1}{12}$, foi subtraído 0.5 para reduzir a média a zero e, em seguida, elevou-se a variância conforme a multiplicação por A_{wn} . A multiplicação não apenas eleva a variância, como também aumenta os limites da amostra para $[-0.5 \cdot A_{wn}, 0.5 \cdot A_{wn}]$. Após isso, o vetor foi tratado de forma a se excluir a influência da frequência de 50 Hz deste, permitindo apenas a existência do tom principal neste índice e assim facilitando o cálculo do erro. Para isso, foi realizada a FFT (*Fast Fourier Transform*) do ruído, para então zerar-se os índices correspondentes à frequência do sinal mono-tonal, no caso 50 Hz e -50 Hz, em seguida realizando-se a FFT Inversa para produzir um ruído branco sem influência desta frequência. Em seguida, o sinal é somado ao tom principal. Todo este procedimento é realizado pela função *addWhiteTone*.

Quanto aos indicadores, o indicador EF Médio trata-se da média dos erros de frequência ocorridos ao longo da execução do algoritmo. O indicador EM Médio trata-se da média dos erros da magnitude (amplitude) do sinal durante a execução do algoritmo. Por fim, a Taxa de Acertos trata-se do percentual de resultados que alcançaram um EF e um EM menor que 0.5% do valor ideal.

Os indicadores propostos para este trabalho foram adotados por mensurarem a eficácia (indicadores EF Médio e EM Médio) e a eficiência do algoritmo (Taxa de Acertos). Em adendo, o indicador EF Médio trata-se de indicador proposto nos padrões IEEE, enquanto o EM Médio é basicamente o vetor do TVE com fase e tempo iguais a zero, o qual também é proposto pelo IEEE. Quanto à Taxa de Acertos, o valor de 0,5% adotado como margem de acerto foi escolhido por ser metade da margem do TVE.

Foram realizadas quatro simulações em estado estático, tendo cada uma destas se baseando nos resultados das anteriores para limitar o escopo dos parâmetros - a simulação seguinte se aproveita dos melhores resultados da anterior para eliminar iterações, por exemplo. A primeira simulação foi realizada com um sinal mono-tonal utilizando-se dos parâmetros f_s , f_0 , P e Q (K Central foi configurado como equivalente a 3). A segunda foi feita novamente com um sinal mono-tonal, tendo os parâmetros P e Q sido retirados (configurados como $P = 1$ e $Q = 11$), dado que pouco impactaram nos resultados, e substituídos pelo parâmetro K Central. A terceira simulação foi feita com um sinal bi-tonal - um sinal principal e outro de ruído, utilizando-se dos parâmetros f_0 , f_r , A_r , P e Q - K Central e f_s foram configurados respectivamente como 3 e 10 kHz. Por fim, a quarta

simulação foi realizada com sinal mono-tonal somado a um ruído branco, utilizando-se apenas dos parâmetros A_{wn} , P e Q - os demais foram configurados para os melhores resultados das simulações anteriores, pois o objetivo era avaliar apenas o efeito do ruído branco.

Todas as simulações rodaram sobre um sinal com amostras equivalentes a 2 segundos de duração, com a frequência de ativação do algoritmo configurada a 50 fps. A cada iteração da simulação, os resultados dos indicadores da simulação eram armazenados em um *Dataframe* da biblioteca *Pandas*. No final de cada simulação, foram gerados três gráficos a partir dos resultados armazenados para cada um dos parâmetros: um gráfico para o indicador de EF Médio, um para o EM Médio e um terceiro para a Taxa de Acertos. Cada um desses gráficos foram feitos selecionando uma combinação dos demais parâmetros, produzindo três linhas no gráfico: uma para o menor resultado do indicador, outra para a mediana dos resultados e, por fim, uma última para o melhor resultado do indicador. Optou-se por este método de agregação com o objetivo de identificar o impacto da evolução do parâmetro na melhor e na pior circunstância, além de produzir, com uso da mediana, uma noção inicial da distribuição amostral dos resultados dada a configuração da simulação. Por fim, os melhores resultados de cada simulação foram filtrados e armazenados em um arquivo txt para consulta e análise posterior.

5 Resultados

5.1 Execução das Simulações

5.1.1 Simulação 1

A primeira simulação mono-tonal foi realizada com os seguintes parâmetros e suas respectivas faixas de valores:

- **Frequência do tom principal (F_0):** intervalo da simulação de 45 a 55 Hz, equivalendo a 11 valores de teste;
- **Frequência de Amostragem (F_s)** intervalo da simulação de 22 a 60 kHz, em passos de 2 kHz (equivalente a 20 valores de teste);
- **Iterações de P :** intervalo da simulação de 1 a 4 iterações;
- **Iterações Q :** intervalo da simulação de 11 a 30 iterações, totalizando 20 valores de teste.

O número total de testes executados é obtido pela multiplicação entre as quantidades de testes de cada parâmetro, o que totaliza 17.600 combinações dos valores dos parâmetros simulados.

5.1.2 Simulação 2

A segunda simulação mono-tonal foi realizada os parâmetros $P=1$ e $Q=11$. Isso se deve ao fato de não ter sido aferido nenhuma interferência desses parâmetros nos resultados da simulação anterior, logo, por questão de velocidade, optou-se por realizá-la com os menores valores possíveis. Foram avaliados os seguintes parâmetros nesta simulação:

- **Frequência do tom principal (F_0):** intervalo da simulação de 47 a 53 Hz, equivalendo a 7 valores de teste;
- **Frequência de Amostragem (F_s)** intervalo da simulação de 10 a 30 kHz, em passos de 2 kHz (equivalente a 11 valores de teste);
- **Índice na frequência de F_0 (K central):** intervalo da simulação de 3 a 6, ou seja, 4 valores de teste.

Esta simulação teve por objetivo testar o impacto do aumento da resolução da frequência do sinal junto à frequência de amostragem. O próprio algoritmo realiza os cálculos para ajustar o tempo de amostragem de acordo com o índice k que será usado como centro da janela. O número total de testes executados para esta simulação foi de 588 valores de parâmetros.

5.1.3 Simulação 3

A terceira simulação foi bi-tonal, ou seja, o sinal utilizado possui o tom principal e um tom de ruído. Dado que não houve muita evolução nas simulações mono-tonais no que tange a frequência de amostragem e a resolução da DFT, a terceira simulação manteve a frequência de amostragem constante em $F_S=10$ kHz e K Central constante em 3, em virtude da diminuição no tempo de simulação. Os parâmetros avaliados nesta simulação foram:

- **Frequência do tom principal (F_0):** intervalo da simulação de 47 a 53 Hz, equivalendo a 7 valores de teste;
- **Frequência do tom ruidoso (F_r):** intervalo da simulação de 60 a 80 Hz, totalizando 21 valores de teste;
- **Amplitude do tom ruidoso (A_r):** intervalo variando de $0.05 \times A_0$ a $0.2 \times A_0$, em passos de $0.05 \times A_0$, o que totaliza 4 valores de teste;
- **Iterações de P :** intervalo da simulação de 1 a 4 iterações;
- **Iterações Q :** intervalo da simulação de 11 a 30 iterações, totalizando 20 valores de teste.

A simulação iterou por 47.040 combinações de valores de teste ao todo.

5.1.4 Simulação 4

A quarta e última simulação foi mono-tonal somada a ruído branco. O ruído branco trata-se de um ruído de natureza estocástico, cuja média é zero e cuja variância foi parametrizada como $\frac{A_{wn}}{12}$. Parâmetros fixos foram configurados de acordo com os melhores resultados das últimas simulações, a saber: $f_0 = 50$ Hz, $F_s=10$ kHz e K Central=3. Nessa simulação, os intervalos dos parâmetros foram então:

- **Amplitude do ruído branco (A_{wn}):** intervalo variando de 2 a 22, em passos de 2, o que totaliza 11 valores de teste;
- **Iterações de P :** intervalo da simulação de 1 a 4 iterações;

- **Iterações Q :** intervalo da simulação de 11 a 30 iterações, totalizando 20 valores de teste.

O objetivo desta simulação foi avaliar o comportamento do algoritmo dada uma interferência do tipo ruído branco. O número total de testes para esta simulação foi de 880 combinações.

5.2 Discussão dos Resultados

5.2.1 Simulação 1

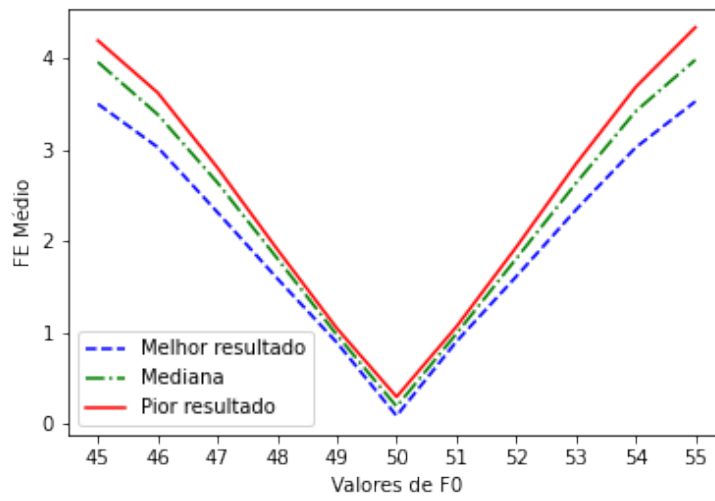
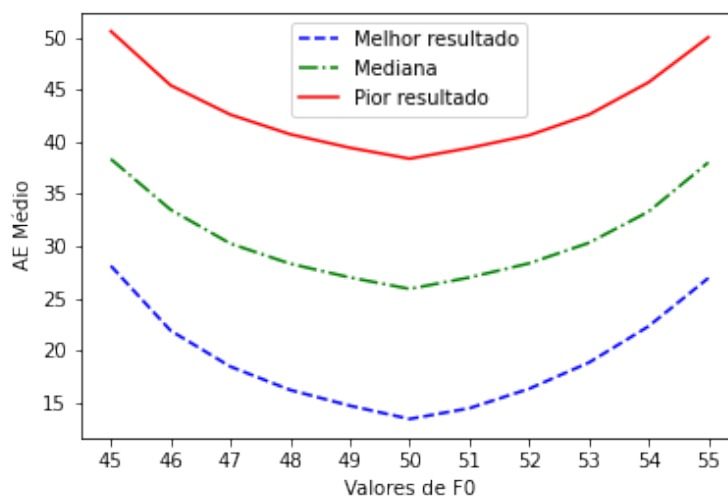
Iniciando pela primeira simulação, a Tabela 2 expõe os resultados obtidos em sua execução.

Tabela 2 – Tabela dos melhores resultados por parâmetros simulados para 1ª Simulação

Indicadores de Desempenho	Parâmetros Simulados			
	F_s (Hz)	F_0 (Hz)	Q ad.	P ad.
Taxa de Acerto	22.000	50	11-30	1-4
EF Médio	22.000	50	11-30	1
EM Médio	22.000	50	11-30	1

Fonte: Autoria própria

Como explicitado, os melhores resultados adquiridos como resultado desta simulação para os três indicadores mensurados são todos referentes à combinação de 22 kHz de F_s e 50 Hz de F_0 . A frequência de 50 Hz para F_0 era esperada devido ao próprio erro envolvido na interpolação linear dos resultados, já que a relação entre a posição da frequência central e as magnitudes dos índices da DFT não é linear. Outrossim, como sigma se torna exatamente zero nesta frequência por ser a frequência central do algoritmo, neste ponto o algoritmo é exato. As figuras 3, 4 e 5 expõem os resultados obtidos para este parâmetro. Nela, percebe-se o significativo decréscimo da eficácia e eficiência do algoritmo em proporção ao afastamento de sua frequência central.

Figura 3 – Evolução de EF Médio para F_0 Figura 4 – Evolução de EM Médio para F_0

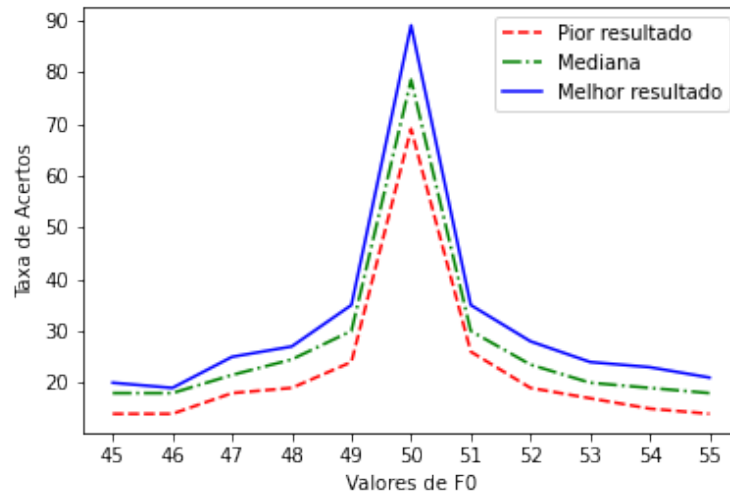


Figura 5 – Evolução de Taxa de Acertos para F_0

Quanto à frequência de amostragem, percebe-se pelos resultados expostos pelas Figuras 6, 7 e 8 que há uma perda de eficácia e eficiência com a evolução de F_s . O EM Médio apresenta um crescimento aproximadamente linear com o crescimento de F_s . Quanto ao EF Médio, este se mantém relativamente estável, com um leve crescimento diante do aumento de F_s , isto para os melhores resultados - para a mediana e piores resultados, ele oscila em torno de uma média. A diminuição na Taxa de Acertos se deve principalmente à diminuição da precisão tanto da amplitude quanto da frequência. Há de se dizer que um fato que pode diminuir o erro aferido é a exclusão dos primeiros resultados: os primeiros resultados impactam no erro ao não estarem estáveis ainda, pois o algoritmo msDFT requer um certo período para estabilizar os índices calculados.

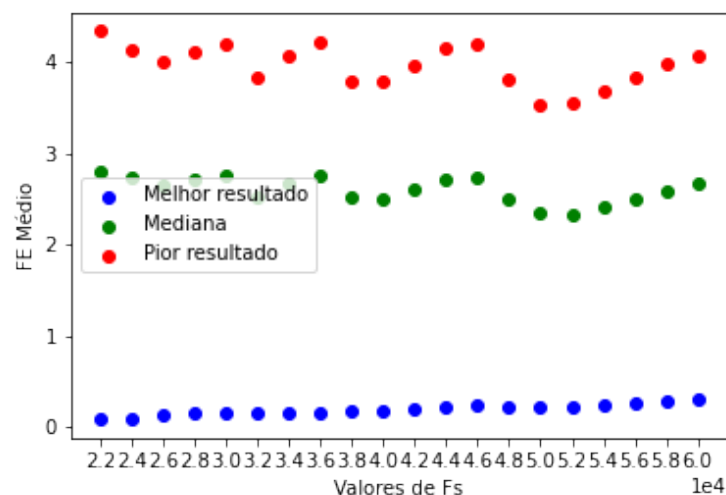
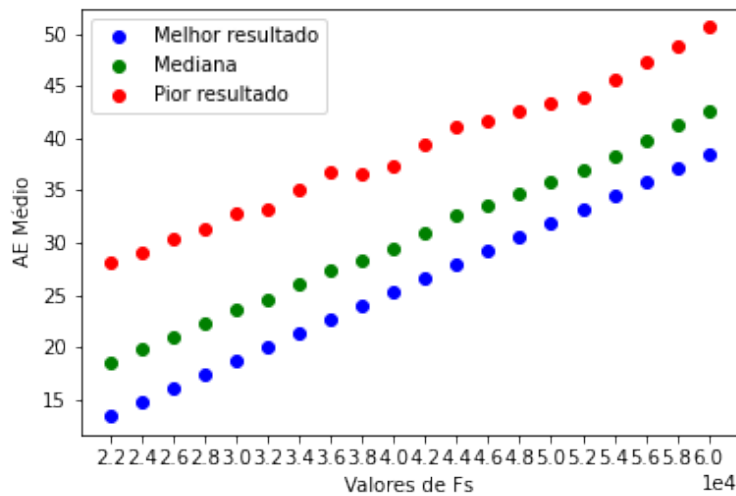
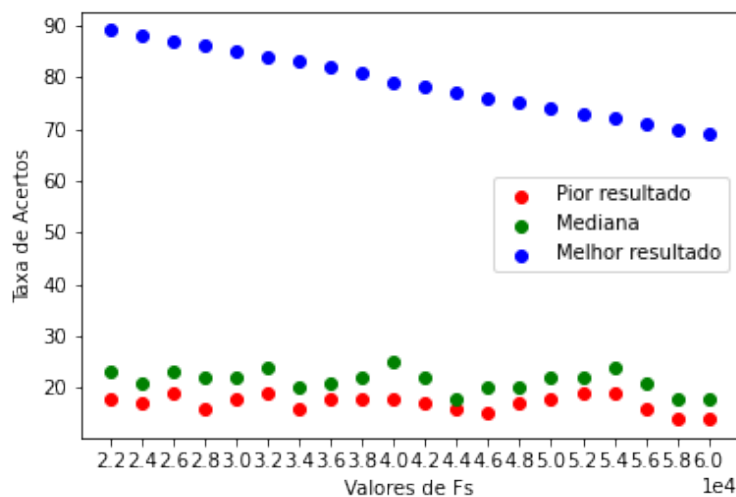


Figura 6 – EF Médio para f_s

Figura 7 – EM Médio para f_s Figura 8 – Taxa de Acertos para f_s

Além disso, testes anteriores à simulação apontaram uma certa periodicidade do algoritmo quando a frequência do tom principal está fora da resolução, com este apresentando um valor próximo ao estimado como mínimo, crescendo e retornando a este valor após algumas iterações. Embora este último comportamento tenha sido identificado, também não é escopo do trabalho o estudo e interpretação deste comportamento.

Quanto aos parâmetros P e Q , cujos resultados são expostos pelas Figuras de 9 a 14, constata-se a irrelevância destes na melhoria dos indicadores devido à constância da reta para os resultados. Isto também é percebido ao se averiguar que os resultados, na Tabela 2, são os mesmos para todas as iterações de Q e P . O comportamento era teoricamente esperado: como o sinal é mono-tonal, o critério de ativação do algoritmo i-

IpDFT não seria ativado em nenhuma circunstância, além de não possuir uma interferência significativa da imagem espectral negativa, tendendo a zero rapidamente em proporção às iterações do algoritmo e-IpDFT - neste caso com apenas uma iteração.

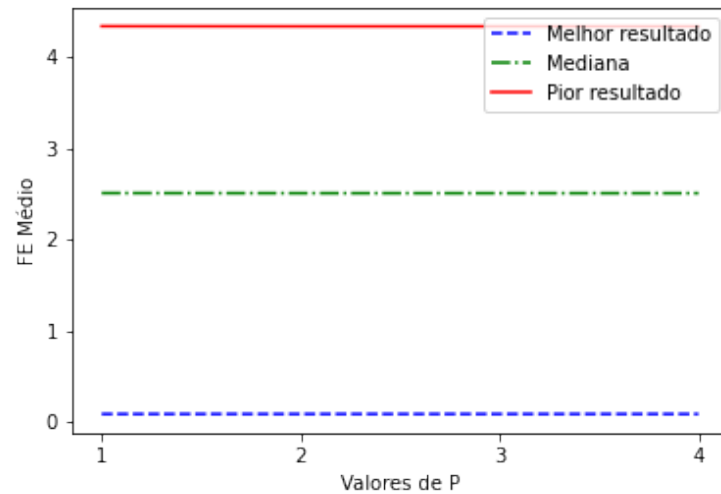


Figura 9 – Evolução de EF Médio para P

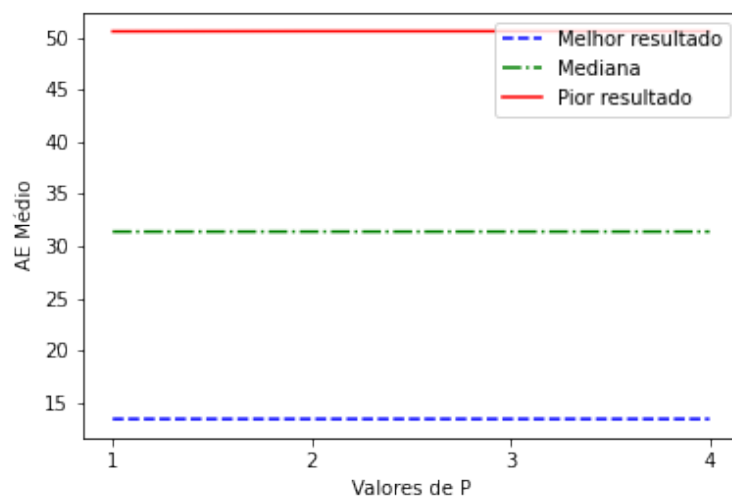
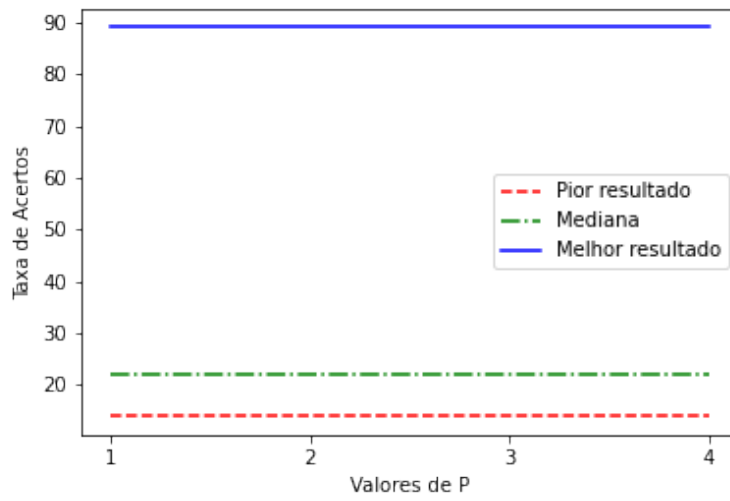
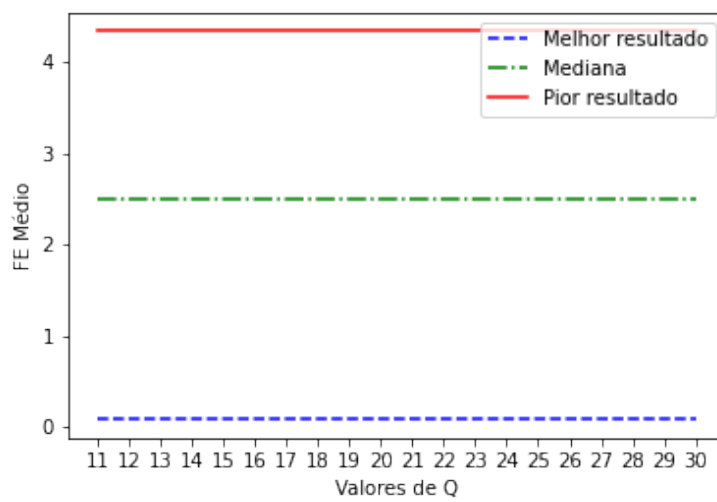
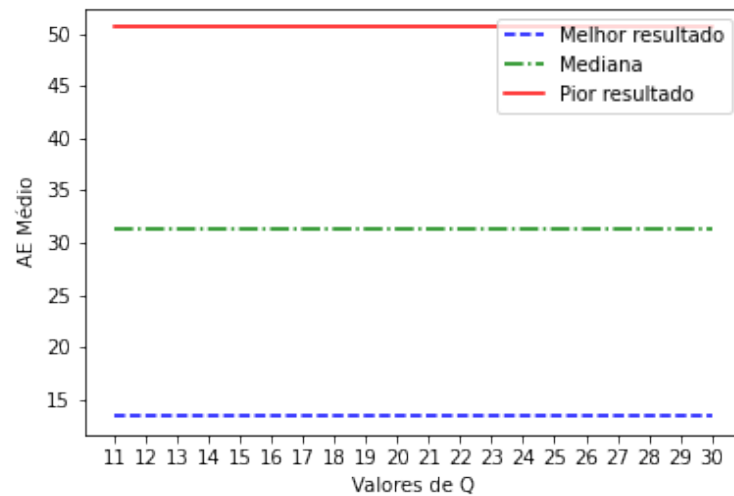
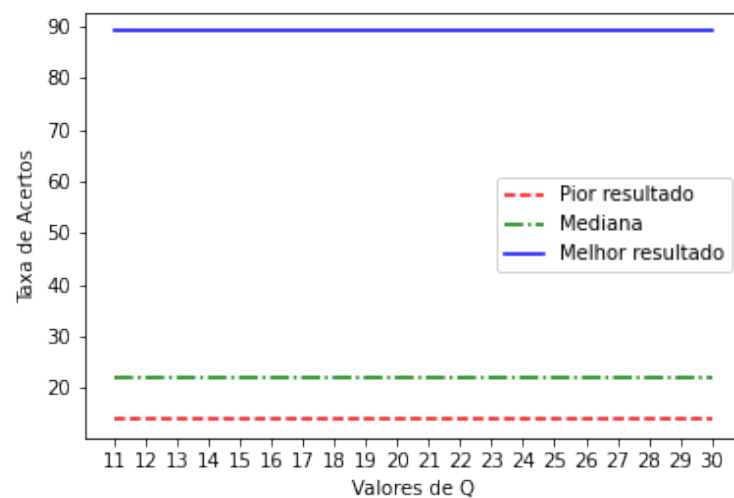


Figura 10 – Evolução de EM Médio para P

Figura 11 – Evolução de Taxa de Acertos para P Figura 12 – Evolução de EF Médio para Q

Figura 13 – Evolução de EM Médio para Q Figura 14 – Evolução de Taxa de Acertos para Q

5.2.2 Simulação 2

Na segunda simulação, o intento era testar o impacto do aumento da resolução da transformada do sinal e sua interação com a frequência de amostragem. Foram definidos que $Q=11$ e $P=1$, dado os resultados da simulação anterior e o com o objetivo desta configuração ser mais rápida, permitindo um menor tempo de execução da simulação. Tal como na simulação anterior, segue os melhores resultados obtidos, explícitos na Tabela 3.

Percebe-se uma relação inversa entre a resolução do sinal (K Central) e o EM Médio do sinal. Isto pode ser visualizado pelas Figuras 15, 16 e 17. Percebe-se no gráfico um aumento no EM Médio para o aumento da índice k da frequência central. Entretanto,

Tabela 3 – Tabela dos melhores resultados por parâmetros simulados para 2ª Simulação

Indicadores de Desempenho	Parâmetros Simulados		
	F_s (Hz)	F_0 (Hz)	K_c ad.
Taxa de Acerto	10.000	50	3
EF Médio	12.000	50	6
EM Médio	10.000	50	3

Fonte: Autoria própria

para o EF Médio, este resultado é inverso, ou constante no melhor resultado. Por fim, a taxa de acerto possui um leve decrescimento com a melhoria da resolução.

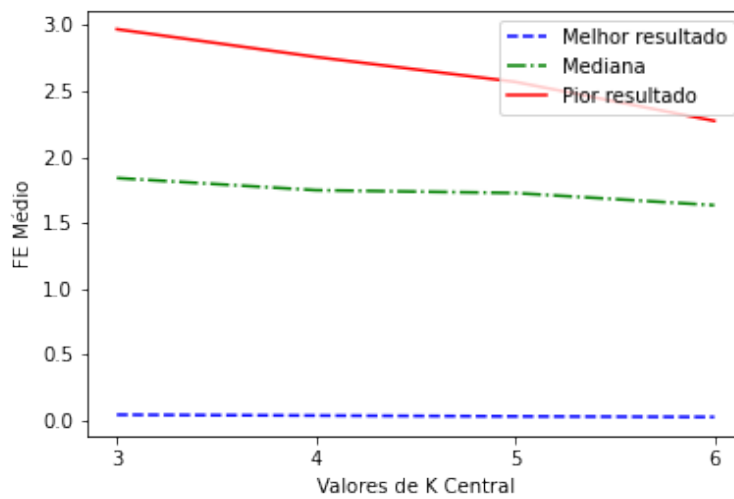


Figura 15 – Evolução de EF Médio para K Central - 2ª Simulação

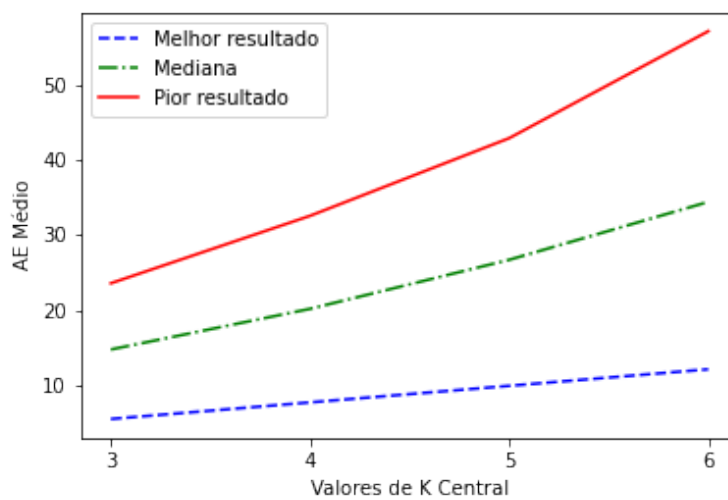
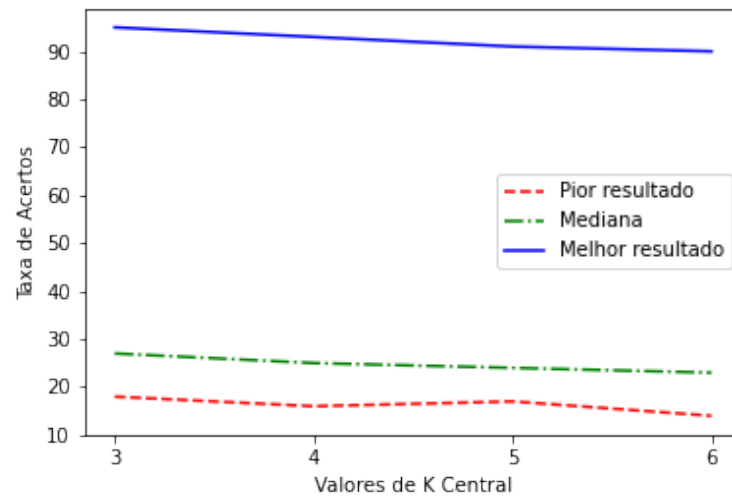
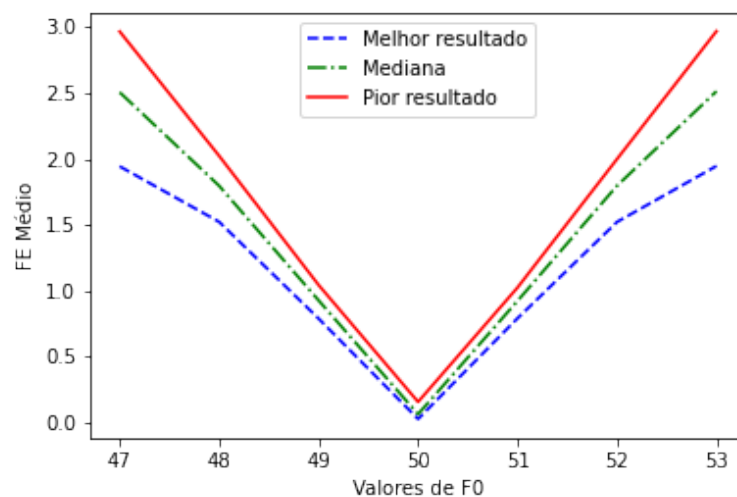
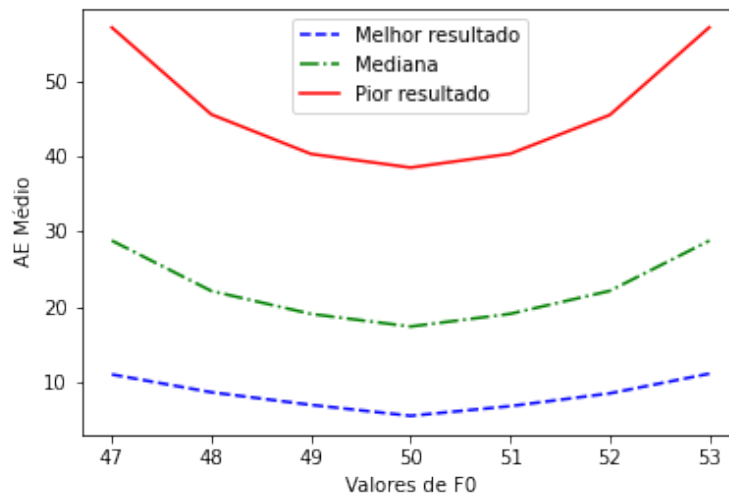
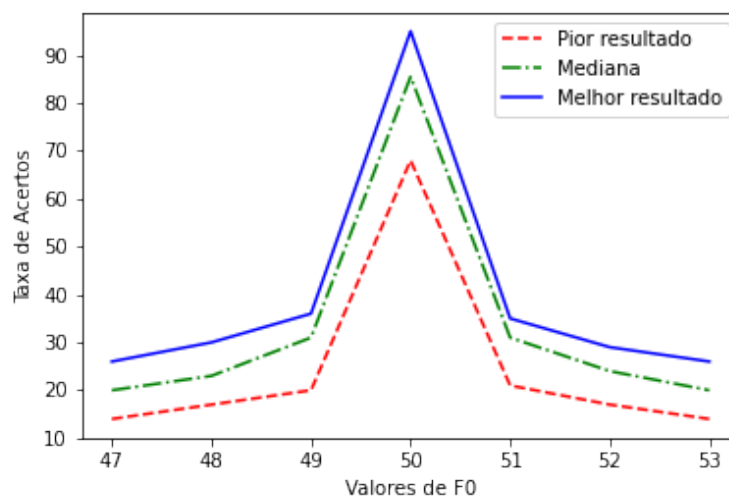


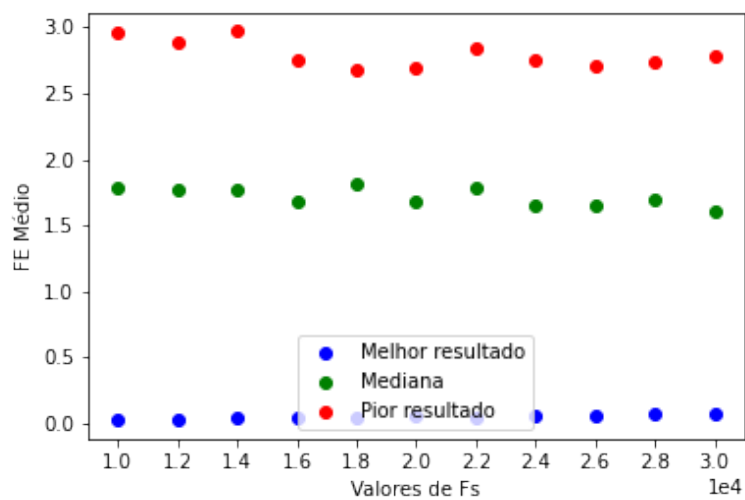
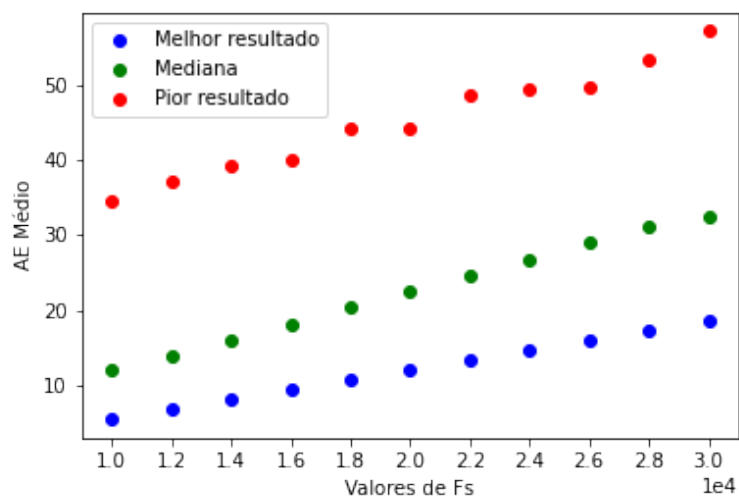
Figura 16 – Evolução de EM Médio para K Central - 2ª Simulação

Figura 17 – Evolução de Taxa de Acertos para K Central - 2ª Simulação

Quanto aos resultados de f_s e f_0 , visualizados pelas Figuras de 18 a 23. Percebe-se com elas que não ocorreu nenhuma mudança quanto aos resultados da Simulação 1.

Figura 18 – Evolução de EF Médio para F_0 - 2ª Simulação

Figura 19 – Evolução de EM Médio para F_0 - 2ª SimulaçãoFigura 20 – Evolução de Taxa de Acertos para F_0 - 2ª Simulação

Figura 21 – Evolução de EF Médio para F_s - 2ª SimulaçãoFigura 22 – Evolução de EM Médio para F_s - 2ª Simulação

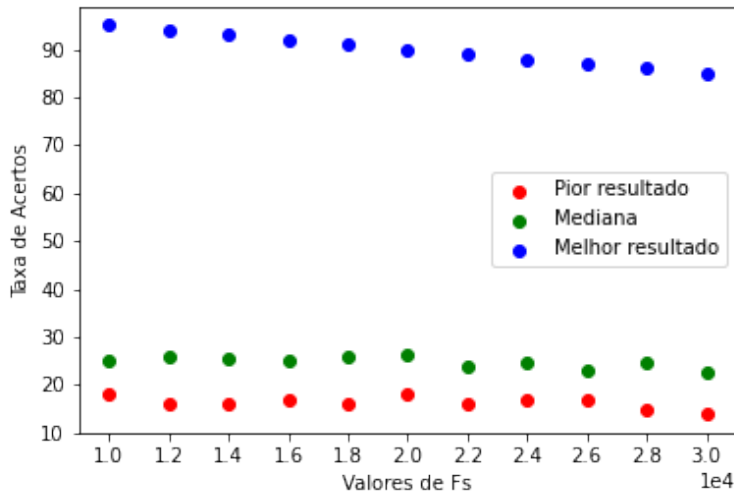


Figura 23 – Evolução de Taxa de Acertos para F_s - 2ª Simulação

Para os resultados serem mais definitivos, talvez um período maior do sinal e a exclusão de alguns *outliers* permitiria uma avaliação mais apurada dos resultados, mas, ao que indicam os resultados, não há nenhum ganho nos indicadores diante do aumento da resolução do sinal ou da frequência de amostragem do sinal. Fica como proposta de melhoria e trabalhos futuros a exclusão dos *outliers* e expansão do tempo de amostragem, assim como a análise do tempo para a estabilização o sinal.

5.2.3 Simulação 3

Na terceira simulação, objetivou-se testar o impacto da interferência de um tom ruidoso no desempenho do algoritmo. Dado o desempenho da simulação anterior, foi configurado $F_s=10$ kHz e $K_c=3$. O cálculo dos indicadores também desconsiderou os dois primeiros e os dois últimos resultados, dado que estes não eram precisos já que o algoritmo msDFT não é estável neles. Segue os melhores resultados obtidos, explícitos na Tabela 4.

Tabela 4 – Tabela dos melhores resultados por parâmetros simulados para 3ª Simulação

Indicadores de Desempenho	Parâmetros Simulados				
	F_0 (Hz)	F_r (Hz)	A_s (V)	Q ad.	P ad.
Taxa de Acerto	50	75	11 e 22	11-30	1-4
EF Médio	50	75	11	11-30	1
EM Médio	50	80	11	11-30	4

Fonte: Autoria própria

Como visto na tabela e nas Figuras de 24 a 29, a quantidade de iterações Q e P não impactaram na Taxa de Acertos do algoritmo. Entretanto, um ponto a se destacar trata-se da diferença nos resultados ótimos para melhor EF Médio e melhor EM Médio

quanto à evolução de P . No caso do EF Médio, percebe-se que seu desempenho é melhor quando são feitas poucas iterações P , enquanto para o EM Médio há uma melhora com o aumento das iterações. Essa diferença no desempenho, entretanto, é baixa - na ordem de 10^{-4} tanto para o EF quanto para o EM nos melhores resultados.

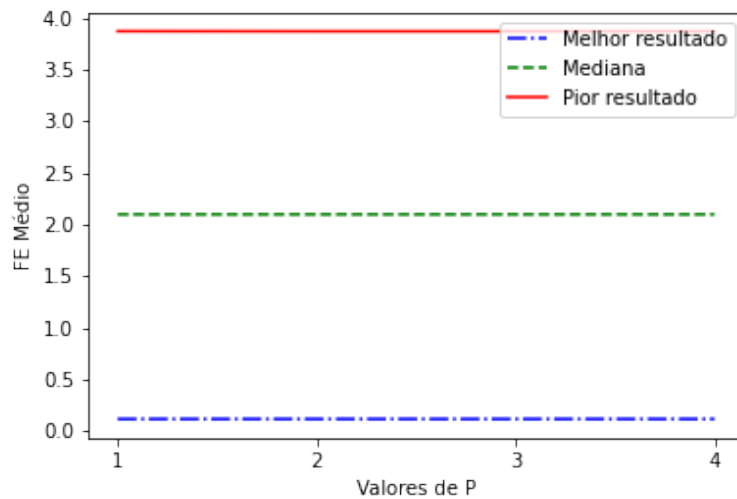


Figura 24 – Evolução de EF Médio para P - 3ª Simulação

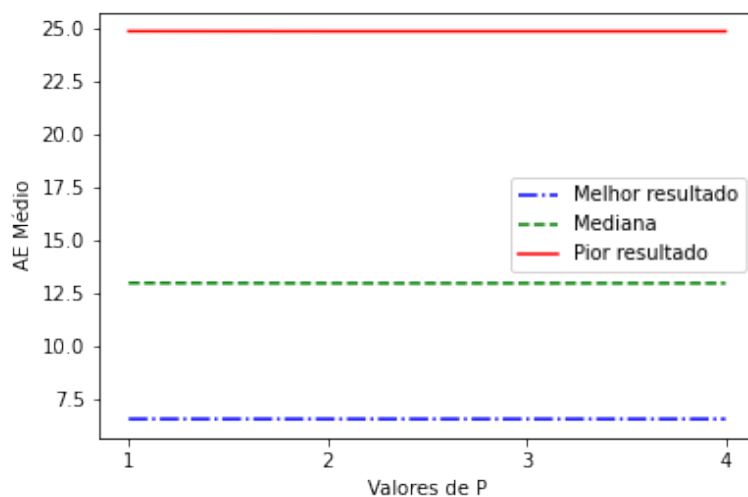
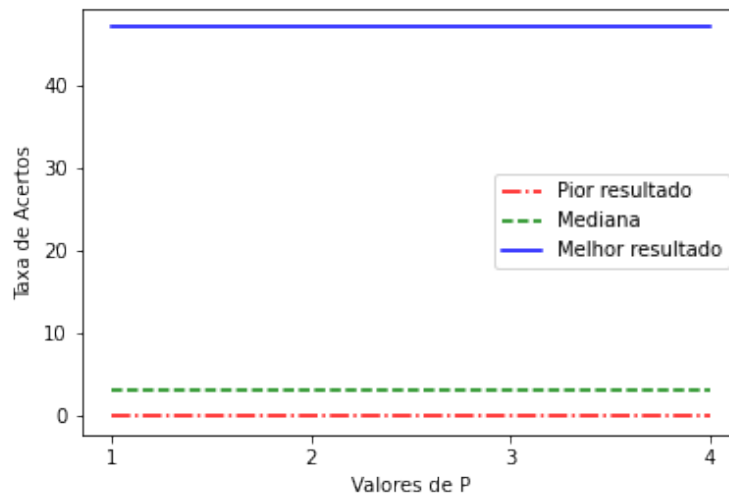
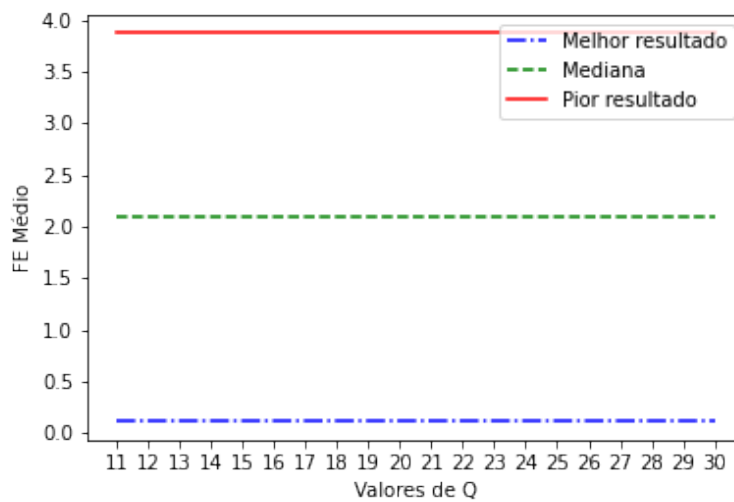
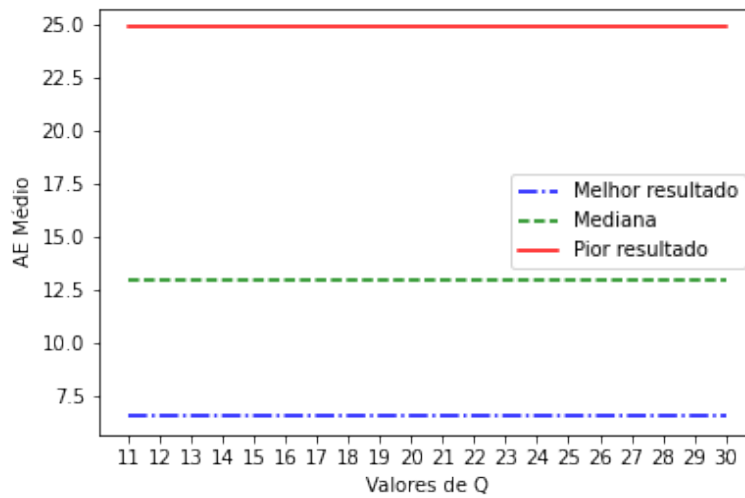
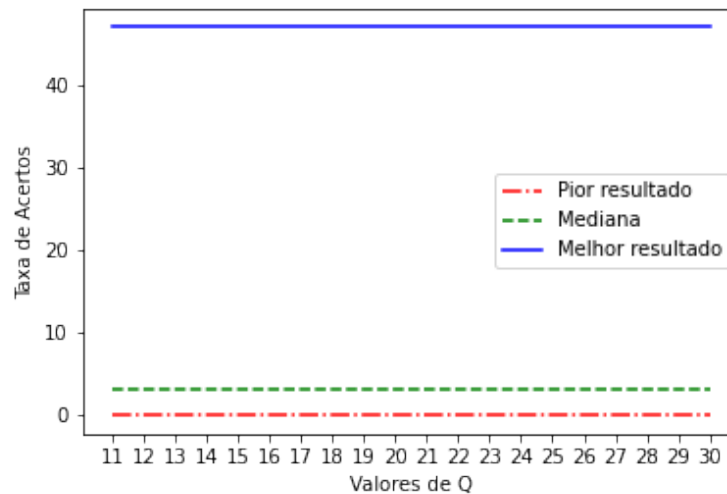


Figura 25 – Evolução de EM Médio para P - 3ª Simulação

Figura 26 – Evolução de Taxa de Acertos para P - 3ª SimulaçãoFigura 27 – Evolução de EF Médio para Q - 3ª Simulação

Figura 28 – Evolução de EM Médio para Q - 3ª SimulaçãoFigura 29 – Evolução de Taxa de Acertos para Q - 3ª Simulação

Para o parâmetro F_r , os indicadores desempenharam melhor na frequência de 75 Hz, ou seja, 150% da frequência F_0 em questão (50 Hz). Também vê-se o seguinte comportamento quando avaliado os melhores resultados: para F_r o EF Médio se mantém próximo à constância, com uma melhoria abrupta em $F_r=75$ e retornando ao nível próximo ao anterior. Avaliando agora o indicador de EM Médio, este demonstrou um desempenho visivelmente melhor com o aumento da frequência de amostragem e consequentemente distanciamento do tom principal. Vê-se uma diminuição significativa do erro conforme se aumenta a frequência F_r . Quanto à Taxa de Acertos, não há um padrão de desempenho quando avaliado a progressão do indicador, entretanto é identificado uma elevação do indicador conforme este se aproxima de 75 Hz, seu máximo local, seguida com o decréscimo do indicador. Além disso, a 78 Hz o indicador possui um mínimo

local e retorna a crescer, continuando em crescimento até o fim do intervalo de simulação. Tudo isto pode ser visualizada nas Figuras 31, 30 e 32.

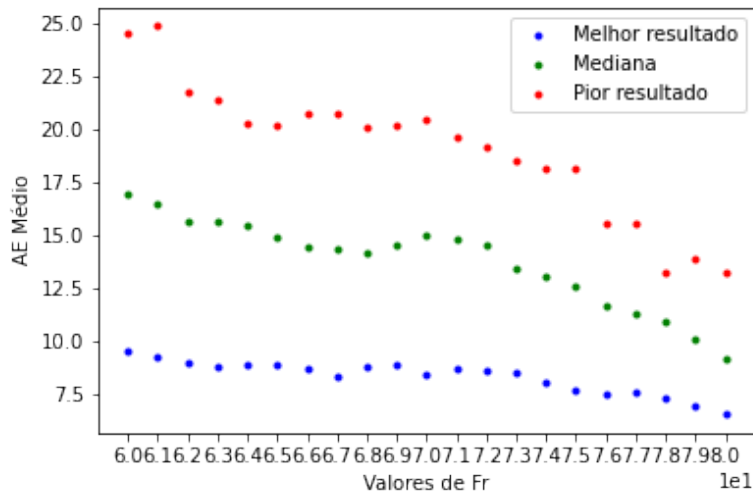


Figura 30 – Evolução de EM Médio para F_r - 3ª Simulação

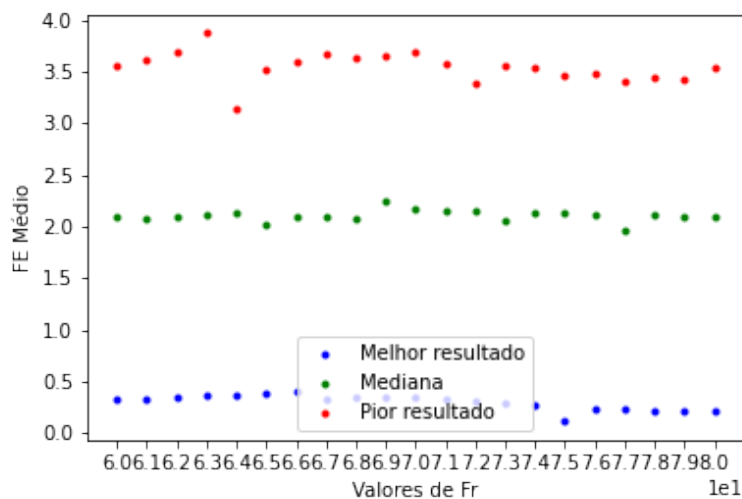


Figura 31 – Evolução de EF Médio para F_r - 3ª Simulação

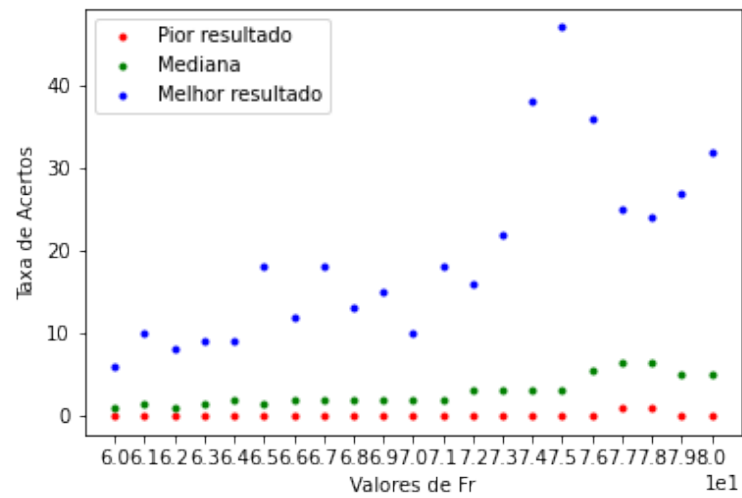


Figura 32 – Evolução de Taxa de Acertos para F_r - 3ª Simulação

Quanto a A_r , cujos resultados são visíveis nas figuras 33, 34 e 35, o comportamento é de decréscimo da Taxa de Acertos com a progressão dos valores do intervalo, com uma queda de quase 50% do desempenho comparado ao primeiro valor na curva de melhor resultado. Para EF Médio e EM Médio a piora do indicador também ocorre com a progressão dos valores, mas é mais significativa para o indicador EM Médio.

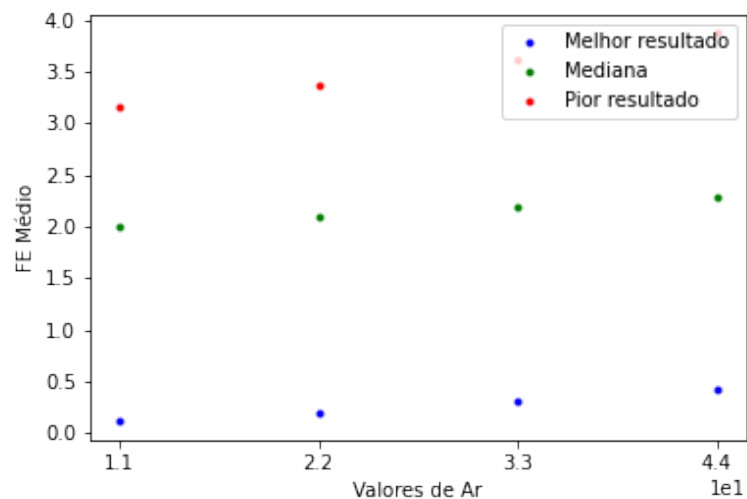


Figura 33 – Evolução de EF Médio para A_r - 3ª Simulação

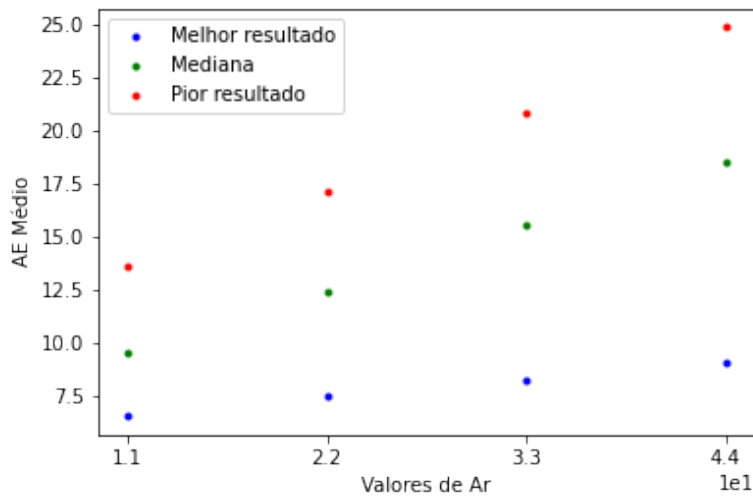


Figura 34 – Evolução de EM Médio para A_r - 3ª Simulação

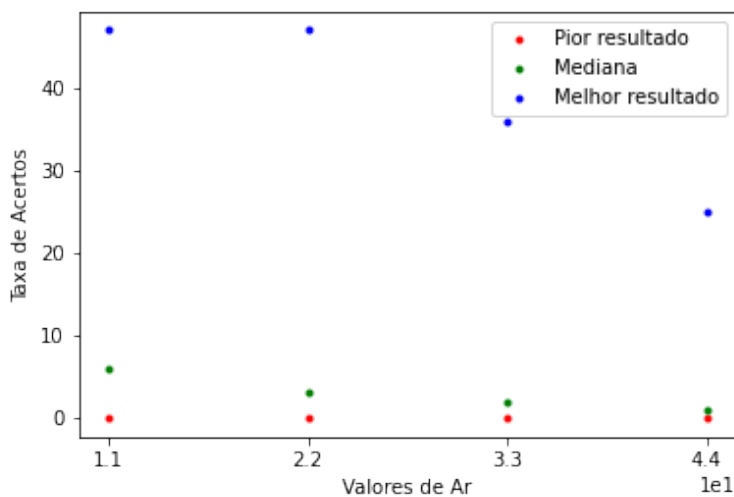
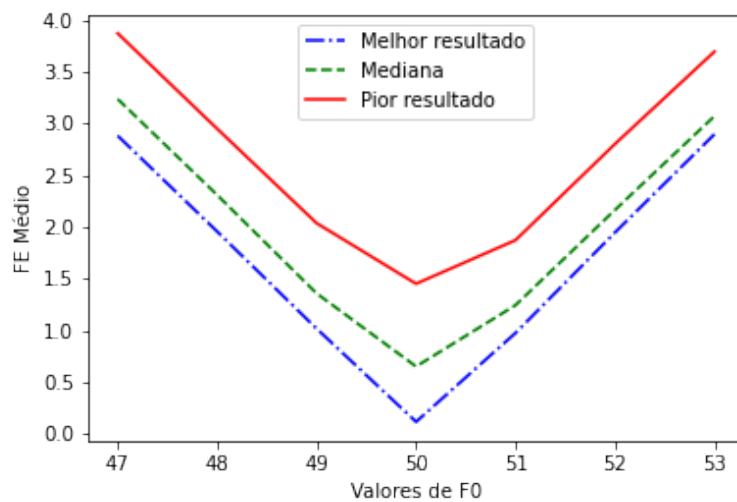
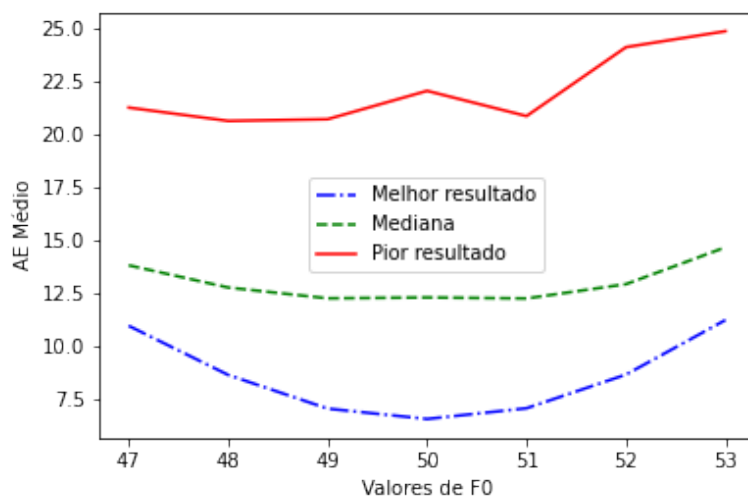


Figura 35 – Evolução de Taxa de Acertos para A_r - 3ª Simulação

Quanto à evolução da f_0 , como visto nas Figuras 36, 37 e 38, não houve significativa alteração no comportamento visto em simulações anteriores, exceto no pior resultado, no qual ocorreu uma relevante distorção. Entende-se a partir disto que a interferência do ruído no pior resultado impactou significativamente no funcionamento do algoritmo.

Figura 36 – Evolução de EF Médio para F_0 - 3ª SimulaçãoFigura 37 – Evolução de EM Médio para F_0 - 3ª Simulação

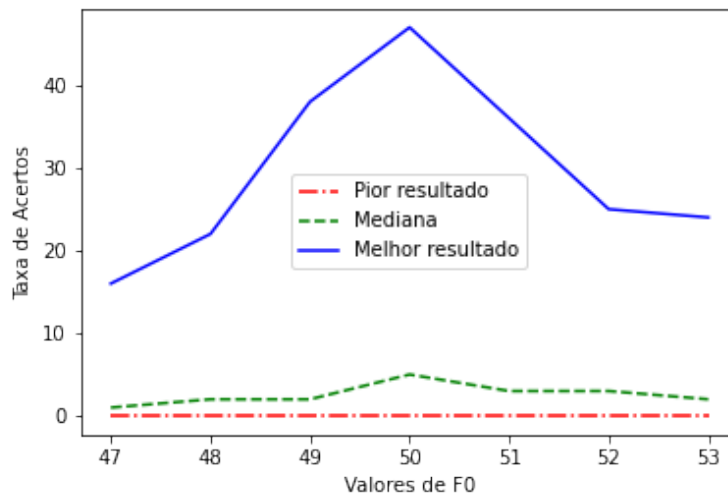


Figura 38 – Evolução de Taxa de Acertos para F_0 - 3ª Simulação

Por fim, um ponto de destaque desta simulação é que seus piores resultados, com 0% de Taxa de Acertos, representam 23.09% do total de parâmetros simulados. Uma análise mais detalhada aponta que esses resultados concentram-se mais nas frequências mais distantes da frequência central, como era de se esperar. Entretanto, a frequência de 53 Hz possui um resultado melhor quanto à quantidade de falhas totais que 52, 49 e 48 Hz, sendo um ponto interessante de destaque. Quanto a F_r , como era de se esperar, os a faixa [70,80] concentram menos falhas totais que a faixa [60,69], com exceção da frequência 66, que possui menos falhas totais que a frequência 71 e igual à frequência 70. Para A_r , também houve previsibilidade no resultado: quanto menor a amplitude, menos falhas totais ocorreram.

Como diagnóstico final, percebe-se que o algoritmo possui seu melhor desempenho conforme a frequência do sinal ruidoso se afasta da frequência do tom principal e também quanto menor é sua amplitude. Em destaque, a frequência do sinal ruidoso de 75 Hz se trata de um máximo local para os melhores resultados. Para uma análise mais profunda, é interessante investigar essas relações, assim como outras faixas de frequência.

5.2.4 Simulação 4

Por fim, nesta última simulação, objetivou-se mensurar a eficiência do algoritmo quando sob a influência de ruído branco. Segue na Tabela 5 os melhores resultados obtidos.

Tabela 5 – Tabela dos melhores resultados por parâmetros simulados para 4ª Simulação

Indicadores de Desempenho	Parâmetros Simulados		
	A_{wn} (V)	Q ad.	P ad.
Taxa de Acerto	2- 22	11-30	1-4
FE Médio	2	11-30	1
EM Médio	2	11-30	4

Fonte: Autoria própria

Como é possível visualizar na tabela, não houve mudança significativa quanto à taxa de acertos nesta simulação, como visto nas Figuras de 39 a 41. Isso se deve ao fato deste indicador aceitar uma margem de até 0.5% de erro para considerar o acerto. O erro se manteve abaixo dessa faixa em aproximadamente 98% dos casos para os três indicadores, oscilando abaixo da margem de acerto.

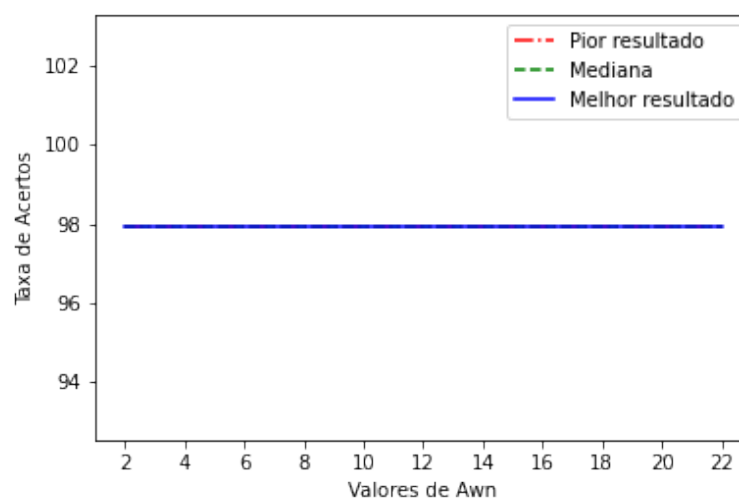
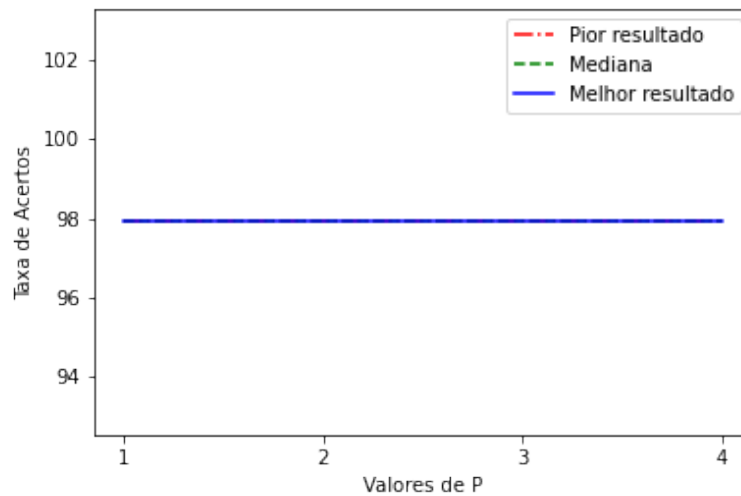
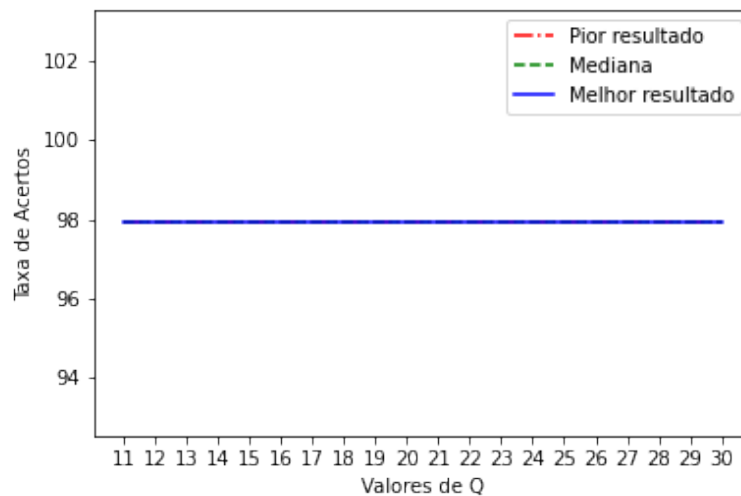


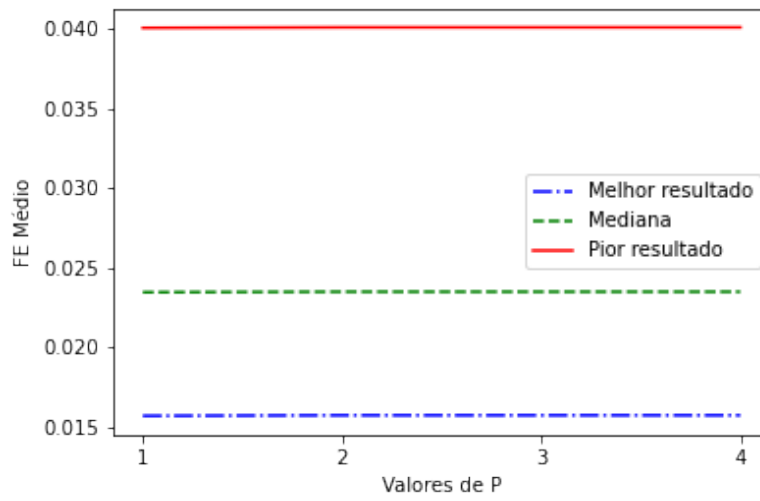
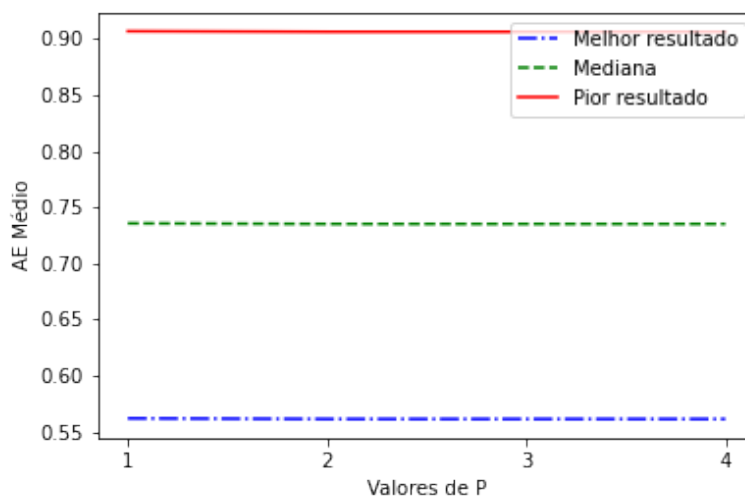
Figura 39 – Evolução de Taxa de Acertos para A_{wn} - 4ª Simulação

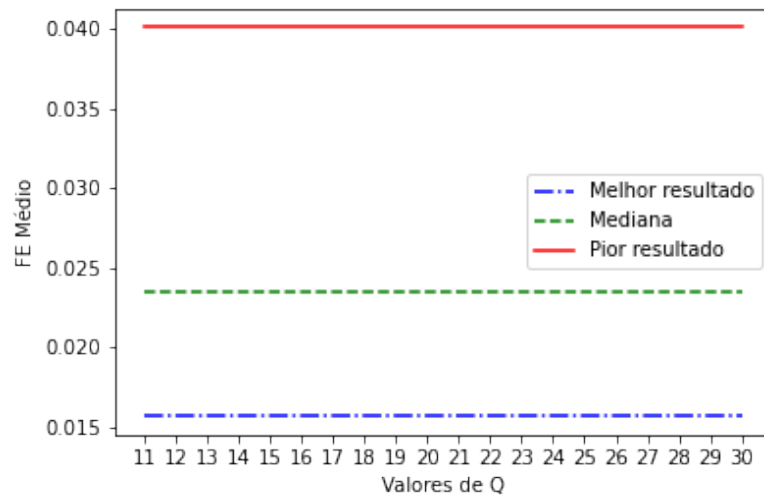
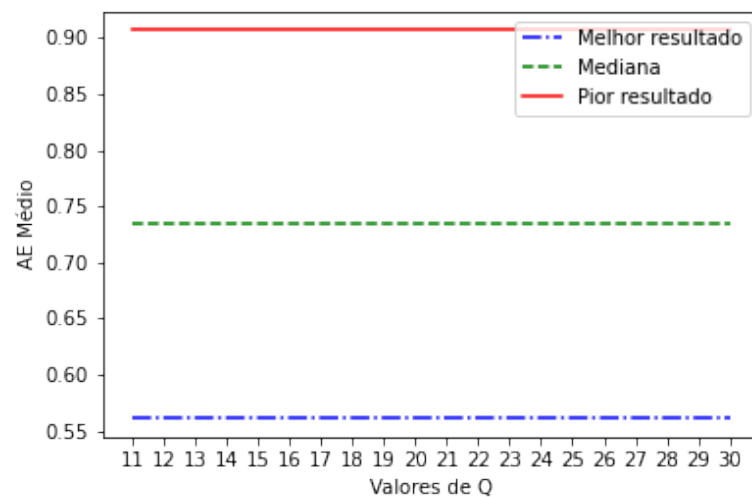
Figura 40 – Evolução de Taxa de Acertos para P - 4ª SimulaçãoFigura 41 – Evolução de Taxa de Acertos para Q - 4ª Simulação

Assim, para uma melhor avaliação do comportamento do algoritmo, é necessário avaliar os indicadores de EM e EF Médios pelos gráficos das Figuras 42 e 42. Com eles, é possível perceber uma tendência de aumento em ambos os indicadores proporcionalmente ao aumento de A_{wn} . O aumento não segue um padrão pela própria característica estocástica do ruído, mas a tendência é claramente perceptível. Esse comportamento era esperado tendo em vista os resultados das simulações anteriores, que apontam o mesmo padrão de aumento da interferência com o aumento da magnitude do ruído no sinal, assim como com a diminuição do intervalo de frequência entre eles.

Figura 42 – Evolução de EF Médio para A_{wn} - 4ª SimulaçãoFigura 43 – Evolução de EM Médio para A_{wn} - 4ª Simulação

Quanto à progressão de Q e P , avaliados nos gráficos das figuras de 44 a 47, novamente percebe-se pouquíssima influência em seu aumento nos resultados. Entretanto, vê-se que, embora mínima, ainda há uma relação inversa entre EM e EF para o aumento de P , embora pouco significativa.

Figura 44 – Evolução de EF Médio para P - 4ª SimulaçãoFigura 45 – Evolução de EM Médio para P - 4ª Simulação

Figura 46 – Evolução de EF Médio para Q - 4ª SimulaçãoFigura 47 – Evolução de EM Médio para Q - 4ª Simulação

Assim, a avaliação dos resultados aponta um desempenho regressivo quanto ao aumento da variância do ruído branco, além da pouca significância dos parâmetros P e Q nos indicadores para a faixa estabelecida, tal como visto nas simulações anteriores. Para o aprofundamento dessa análise, seria interessante avaliar outras formas de ruído, de maneira a elucidar melhor os fatores mais significativos no erro de mensuração.

6 Conclusão

Neste capítulo, são resumidas as conclusões referentes às discussões levantados no capítulo de resultados. Outrossim tais conclusões são conectadas ao problema gerador deste trabalho, de forma a concluir assim os objetivos iniciais.

6.1 Conclusões sobre os resultados

O trabalho possibilitou a visualização de novas áreas de estudos, como a questão da interpolação do sinal e a relação entre os indicadores propostos e o parâmetro P , que permitem a produção de pesquisas específicas, assim como expôs o comportamento do algoritmo e seu desempenho perante parâmetros internos e a determinados tipos de sinal. Dito isto, a pergunta inicial do trabalho, o entendimento do comportamento/desempenho do algoritmo em função da mudança de seus parâmetros internos e diante de diferentes sinais mono-tonais e ruidosos de entrada, foi respondida com sucesso.

A partir do discutido na seção 6.2, é visível que o algoritmo possui desempenho máximo em torno da frequência central configurada do algoritmo. Tal característica é atribuída ao erro envolvido na interpolação linear realizada em vista de estimar o fator de correção do índice da DFT (σ). Isso leva à conclusão que o algoritmo é adequado em situações com uma faixa de variação de frequência relativamente curto, para a qual não há um erro na estimativa.

Também percebeu-se que, para faixa utilizada, o incremento de frequência amostral ou resolução de frequência não produziram um impacto significativo, e até mesmo apresentaram resultados opostos ao desejado, na melhoria dos índices avaliados. Não foi escopo do trabalho se aprofundar nesses resultados e sua fundamentação, logo conclui-se que o algoritmo não possui melhoria significativa com o incremento desses parâmetros.

Quanto às simulações com ruído, percebeu-se que há dois fatores impactantes na eficácia do algoritmo: 1 - a proximidade da frequência ruidosa da frequência central; e 2 - a magnitude da frequência ruidosa. Esses resultados eram esperados, já que ambos impactam nos índices utilizados para a realização da interpolação linear do algoritmo. O mesmo comportamento foi aferido quando realizada a simulação com ruído branco: o aumento da variância do ruído produziu uma tendência de aumento nos indicadores de erro, embora não o suficiente para extrapolar a margem de erro para a aceitação do resultado como um acerto (não influenciou a taxa de acertos).

Por fim, quanto aos parâmetros Q e P do algoritmo estudado, ambos não apresen-

taram significativa influência nos resultados. Os gráficos destes parâmetros apresentaram retas quase constantes ao longo de sua faixa de valores. Entretanto, embora pouquíssimo significativo (diferença na ordem 10^{-4}), foi percebido uma relação inversa entre o indicador EM e EF com o aumento do parâmetro P - o crescimento do indicador leva ao aumento FE e à diminuição do EM, no contexto das simulações que envolvem ruído.

Diante dessas observações, fica claro as circunstâncias de ideal funcionamento do algoritmo: um sinal o mais limpo possível, sem ruído atrelado. Quanto aos parâmetros envolvidos, percebe-se que Q e P não impactam significativamente na melhoria do desempenho do algoritmo, ao menos não para os valores testados nas simulações.

6.2 Trabalhos Futuros

O atual trabalho elucida as circunstâncias ideais de funcionamento e expõe a tendência de melhoria ou piora da eficácia do algoritmo em vista de certos fatores, mas para além disso ele também enseja o estudo e fundamentação de alguns dos resultados obtidos. Por exemplo, este não se aprofunda quando à características dos algoritmos componentes do i-IpDFT. Assim, além da exploração de fatores e ruídos que ficaram de fora do trabalho, um ponto a ser investigado seria o impacto do uso de outras formas de interpolação no algoritmo IpDFT, ou mesmo uma comparação do desempenho do algoritmo e-IpDFT com o algoritmo i-IpDFT em diferentes circunstâncias, para averiguar a necessidade de aplicação de um ou outro.

Seria um foco de estudo interessante, também, a análise mais aprofundada do impacto da frequência de amostragem e da resolução do sinal no algoritmo, as quais não foram significativamente exploradas. A perda de eficácia exposta com o crescimento desses parâmetros ainda carece de fundamentação, sendo um âmbito a ser explorado em trabalhos futuros.

Outro ponto importante a ser averiguado seria qual é a melhor tecnologia para aplicação do algoritmo. Nos artigos utilizados como referência para o trabalho, os algoritmos foram implementados em *hardware*, por uma placa FPGA. Entretanto, não há uma análise empírica e comparativa do desempenho com a implementação via *software* ou por microcontrolador. Portanto, a implementação nessas plataformas permitiria justificar ou dispensar a necessidade de implementação em *hardware*.

O atual trabalho limitou-se a análises comportamentais, portanto realizou programação via orientação a objeto na linguagem Python, a qual torna a execução extremamente lenta. Para a elaboração de uma biblioteca voltada para aplicação em tempo real, e por conseguinte adequada para a realização do estudo proposto no parágrafo anterior, seria necessário o desenvolvimento da aplicação em uma linguagem mais rápida, como C, por exemplo, e de maneira mais otimizada para tal funcionalidade.

Referências

- BERGER, L. T.; INIEWSKI, K. *Redes Elétricas Inteligentes - Aplicações, Comunicação e Segurança*. Rio de Janeiro, RJ: Grupo GEN, 2015. Citado 2 vezes nas páginas 23 e 24.
- DERVIŠKADIĆ, A.; PAOLONE, M. Design and experimental validation of an fpga-based pmu simultaneously compliant with p and m performance classes. *Electric power systems research*, Elsevier B.V, Amsterdam, v. 189, p. 106650, 2020. ISSN 0378-7796. Citado 4 vezes nas páginas 24, 39, 40 e 42.
- FANG, X. et al. Smart grid - the new and improved power grid: A survey. *IEEE*, v. 14, n. 4, p. 944–980, 2012. ISSN 1553-877X. Citado na página 23.
- HOJABRI, M. et al. A comprehensive survey on phasor measurement unit applications in distribution systems. v. 12, n. 23, p. 4552, 2019. ISSN 1996-1073. Citado 2 vezes nas páginas 29 e 32.
- IEEE Standard for Synchrophasor Measurements for Power Systems. *IEEE Std C37.118.1-2011 (Revision of IEEE Std C37.118-2005)*, p. 1–61, 2011. Citado na página 30.
- IEEE/IEC International Standard - Measuring relays and protection equipment - Part 118-1: Synchrophasor for power systems - Measurements. *IEC/IEEE 60255-118-1:2018*, p. 1–78, 2018. Citado 4 vezes nas páginas 31, 32, 34 e 35.
- MARTIN, K. et al. Exploring the iee standard c37.118-2005 synchrophasors for power systems. *IEEE*, New York, NY, v. 23, n. 4, 2008. ISSN 0885-8977. Citado na página 29.
- MITRA, S. K. *Digital Signal Processing - A Computer Based Approach*. [S.l.]: McGraw-Hill, 2010. Citado na página 44.
- ORALLO, C. et al. Harmonics measurement with a modulated sliding dft algorithm on an fpga platform. In: *2012 Argentine School of Micro-Nanoelectronics, Technology and Applications (EAMTA)*. [S.l.]: IEEE, 2012. p. 40–46. ISBN 9781467326964. Citado na página 39.
- ROMANO, P.; PAOLONE, M. Enhanced interpolated-dft for synchrophasor estimation in fpgas: Theory, implementation, and validation of a pmu prototype. *IEEE transactions on instrumentation and measurement*, IEEE, PISCATAWAY, v. 63, n. 12, p. 2824–2836, 2014. ISSN 0018-9456. Citado na página 40.
- ROMANO, P.; PAOLONE, M. An enhanced interpolated-modulated sliding dft for high reporting rate pmus. In: *2014 IEEE International Workshop on Applied Measurements for Power Systems Proceedings (AMPS)*. [S.l.: s.n.], 2014. p. 1–6. Citado na página 38.