

Universidade Federal de Juiz de Fora
Faculdade de Engenharia
Engenharia Elétrica – Habilitação em Sistemas Eletrônicos

Gabriel Eduardo de Lima Machado

**Implementação de técnicas para análise de inter-harmônicos variantes no
tempo**

Juiz de Fora

2019

Gabriel Eduardo de Lima Machado

Implementação de técnicas para análise de inter-harmônicos variantes no tempo

Trabalho de Conclusão de Curso apresentado à Faculdade de Engenharia da Universidade Federal de Juiz de Fora, como requisito para obtenção do grau de Engenheiro Eletricista.

Orientador: Marcelo Antônio Alves Lima

Juiz de Fora

2019

Ficha catalográfica elaborada através do Modelo Latex do CDC da
UFJF com os dados fornecidos pelo(a) autor(a)

Machado, Gabriel.

Implementação de técnicas para análise de inter-harmônicos variantes
no tempo / Gabriel Eduardo de Lima Machado. – 2019.
?? f. : il.

Orientador: Marcelo Antônio Alves Lima

Trabalho de Conclusão de Curso – Universidade Federal de Juiz de Fora,
Faculdade de Engenharia. Engenharia Elétrica – Habilitação em Sistemas
Eletrônicos, 2019.

1. Palavra-chave. 2. Palavra-chave. 3. Palavra-chave. I. Sobrenome,
Nome do orientador, orient. II. Título.

Gabriel Eduardo de Lima Machado

Implementação de técnicas para análise de inter-harmônicos variantes no tempo

Trabalho de Conclusão de Curso apresentado à Faculdade de Engenharia da Universidade Federal de Juiz de Fora, como requisito para obtenção do grau de Engenheiro Eletricista.

Aprovado em:

BANCA EXAMINADORA

Prof. Marcelo Antônio Alves Lima - Orientador
Universidade Federal de Juiz de Fora

Prof. Leandro Rodrigues Manso Silva
Universidade Federal de Juiz de Fora

Prof. Luciano Manhães de Andrade Filho
Universidade Federal de Juiz de Fora

ATA DE APRESENTAÇÃO DE TRABALHO DE CONCLUSÃO DE CURSO

DATA DA DEFESA: **05/12/2019**

CANDIDATO: **Gabriel Eduardo de Lima Machado**

ORIENTADOR: Prof.: **Marcelo Antônio Alves Lima**

BANCA EXAMINADORA: (NOME COMPLETO/INSTITUIÇÃO)

Prof.: **Marcelo Antônio Alves Lima** (Presidente/UFJF)

Prof.: **Leandro Rodrigues Manso Silva** (UFJF)

Prof.: **Luciano Manhães de Andrade Filho** (UFJF)

TÍTULO DO TRABALHO: **IMPLEMENTAÇÃO DE TÉCNICAS ADAPTATIVAS PARA ANÁLISE DE HARMÔNICOS E INTER-HARMÔNICOS VARIANTES NO TEMPO**

LOCAL: **Sala de Vídeo Conferência - PPEE – Faculdade de Engenharia - UFJF**

HORA DO INÍCIO: 10:00

Em sessão pública, após exposição de cerca de 23 minutos, o candidato foi arguido oralmente pelos membros da banca, tendo o mesmo sido:

☒ APROVADO.

☐ REPROVADO.

Na forma regulamentar foi lavrada a presente ata que é abaixo assinada pelos membros da banca na ordem acima determinada e pelo candidato.

Juiz de Fora, 05 de dezembro de 2019.

PRESIDENTE: Marcelo Antônio Alves Lima
Leandro Rodrigues Manso Silva
Luciano M. de A. Filho

CANDIDATO: Gabriel S. S. Machado

AGRADECIMENTOS

Agradeço principalmente a meus pais e minha família, que são os maiores responsáveis por minha formação em qualquer que seja o âmbito e o principal suporte que tive durante todos os anos de estudo.

Agradeço também a meus grandes amigos de faculdade, pelos quais tenho grande consideração, pois foram essenciais na jornada acadêmica além de torná-la muito mais divertida.

Agradeço a todos professores, principalmente ao meu Orientador, que são as pessoas que fazem trabalhos como este serem possíveis.

“Everything in this world is magic, except to the magician.”
Robert Ford.

RESUMO

A estimação espectral não é um novo problema matemático ou de engenharia, pois há alguns séculos já se tenta escrever sinais como uma composição de diferentes frequências. Entretanto, os sinais aparecem de formas distintas nos mais diversos setores da tecnologia. Pode-se citar sua importância em processamento de fala, área biomédica, na análise de eletroencefalogramas, identificação de sistemas, análise de vibrações e qualidade da energia elétrica. Este trabalho visa fundamentar teoricamente o problema em questão, implementar e avaliar técnicas já concebidas para sua solução, bem como apresentar variações e aplicações específicas. Serão utilizadas para tanto conhecidas técnicas de filtragem adaptativa tais com LMS e RLS, os chamados PLLs (*Phase-Locked-Loops*) aliados a estruturas multitaxa e as ferramentas padrão da teoria de análise de sinais discretos no tempo.

Palavras-chave: Estimação Espectral, Filtros Adaptativos, *Phase-Locked-Loop*, Processamento Multitaxa, Banco de Filtros.

ABSTRACT

Spectral estimation is not a new mathematical or engineering problem, as for centuries people have been trying to describe signals as a sum of sinusoidal frequencies. However, this problem still present in different contexts on technology. We can highlight its relevance in signal processing, image processing, biomedical signals, system identification and modeling, and power quality. This work aims to expose the fundamentals of the theory, to implement and evaluate well known techniques and present specific applications to this old problem. To reach this goal, it will be used adaptive filtering methods as LMS and RLS, the so called PLL (Phase-Locked-Loops) with multirate structures and standard tools of discrete time signals analysis.

Keywords: Spectral Analysis, Adaptive Filtering, Phase-Locked-Loop, Multirate Processing, Filter Banks.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de blocos de um filtro adaptativo.	25
Figura 2 – Downsampling para $f_s = 160Hz$, $f_0 = 10Hz$, $M_k = 5$	32
Figura 3 – Downsampling para $f_s = 160Hz$, $f_0 = 10Hz$, $M_k = 10$	33
Figura 4 – Convergência na presença do 3º e 5º harmônicos, além de ruído com SNR=10 dB	35
Figura 5 – Convergência na presença apenas de ruído, com SNR=10 dB	36
Figura 6 – Características do filtro com $w_0=0.25$	37
Figura 7 – Círculo de frequências	39
Figura 8 – Esquema multitaxa PLL	40
Figura 9 – Comparação entre o método de média (com 24 amostras) e o de alien- tamentação direta	41
Figura 10 – Comparação entre o método de média (com 24 amostras) e o geométrico ($\lambda = 0.9$)	41
Figura 11 – Convergência do 15º harmônico na presença de ruído	44
Figura 12 – Convergência do 15º harmônico com degrau em frequência de 0.5 Hz na fundamental	45
Figura 13 – Batimento observado na componente fundamental	46
Figura 14 – Convergência do inter-harmônico 6.4	46
Figura 15 – Convergência do RLS e NLMS na presença dos harmônicos 1, 3, 5 e 7; M=16	51
Figura 16 – Convergência dos coeficientes RLS e NLMS na presença dos harmônicos 1, 3, 5 e 7; M=16	52
Figura 17 – Convergência do RLS e NLMS vista na estimação das frequências, com um degrau de 10 Hz em 500 amostras	53
Figura 18 – Efeito de batimento em frequências muito próximas	54
Figura 19 – Imagem com estimação de amplitudes e frequências no tempo usando RLS	56
Figura 20 – Imagem com estimação de amplitudes e frequências no tempo usando NLMS	57
Figura 21 – Rastreio com teste do artigo [10] utilizando RLS	58
Figura 22 – Rastreio com teste do artigo utilizando RLS, componente fundamental	59
Figura 23 – Estimação de amplitudes com PLL-M	61
Figura 24 – Esquema híbrido	62
Figura 25 – Estágio de estimação de frequências e ordem do sistema	63
Figura 26 – Rastreio de amplitudes do PLL-M	64
Figura 27 – Rastreio de frequências do PLL-M	65
Figura 28 – Rastreio de amplitudes do PLL-M	66

LISTA DE TABELAS

Tabela 1	–	Tabela para a simulação sem ruído	42
Tabela 2	–	Tabela para a simulação com ruído ($\sigma^2=10$)	43
Tabela 3	–	Tabela para a simulação de inter-harmônicos	47
Tabela 4	–	Divisão do menor pelo maior autovalor da matriz de autocorrelação para diferentes pares de valores de frequência e ordem	50
Tabela 5	–	Erros para diferentes proximidades de frequências	52
Tabela 6	–	Simulação com amplitude variável, resultados em %	54
Tabela 7	–	Tabela com os erros para classificação de frequências próximas	55

LISTA DE ABREVIATURAS E SIGLAS

Adaline	Adaptive Linear Neuron
AR	Auto Regressive
ARMA	Auto Regressive Moving Average
DFT	Discrete Fourier Transform
DTFT	Discrete Time Fourier Transform
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
LMS	Least Mean Squares
NLMS	Normalized Least Mean Squares
PLL	Phase-Locked-Loop
PLL-M	Phase-Locked-Loop Multitaxa
PSD	Power Spectrum Density
QEE	Qualidade da Energia Elétrica
RLS	Recursive Least Squares
STFT	Short Time Fourier Transform
SNR	Signal to Noise Ratio
TF	Transformada de Fourier

LISTA DE SÍMBOLOS

\forall Para todo

\in Pertence

∇ Gradiente

SUMÁRIO

1	Introdução	14
1.1	Análise Espectral	14
1.2	Aplicações	14
1.3	Objetivos	15
1.4	Divisão do Trabalho	15
2	Revisão Bibliográfica e Fundamentação Teórica	17
2.1	Análise espectral	17
2.2	Densidade espectral de energia de um sinal determinístico	17
2.2.1	Densidade espectral de potência	18
2.2.2	Periodograma e correlograma	19
2.3	DFT	20
2.4	Análise de Prony	22
2.4.1	Solução do modelo de predição linear	23
2.4.1.1	Sistema linear	23
2.4.2	Adaline	24
2.4.3	Filtros adaptativos	24
2.4.4	Adaptação via gradiente descendente	25
2.4.5	O algoritmo LMS	27
2.4.6	O algoritmo NLMS	27
2.4.7	RLS	28
2.5	PLL	29
2.6	Processamento Multitaxa	31
2.6.1	<i>Downsampling</i>	31
3	Estrutura PLL-Multitaxa	34
3.1	Desempenho do PLL	34
3.2	Banco de filtros	34
3.3	Uso da estrutura multitaxa	36
3.4	Variação da frequência central do banco de filtros	38
3.5	Síntese da estrutura PLL Multitaxa	40
3.6	Simulações	41
3.6.1	Harmônicos ímpares	42
3.6.2	Inter-harmônicos	45
3.7	Esforço computacional	47
3.7.1	Banco de filtros	47

3.7.2	PLL	47
3.7.3	Atualização de frequência	47
3.7.4	Operações Totais	47
4	Estimação de frequências baseada em predição linear	49
4.1	Solução em predição linear	49
4.2	Solução em RLS e NLMS	50
4.3	Simulações	51
4.4	Resultados para o caso de presença de frequências próximas	54
4.5	Melhora da sensibilidade	55
4.6	Teste de rastreo de frequências	57
4.7	Complexidade computacional	59
4.8	Conclusões	59
5	Estrutura híbrida composta por predição linear e PLL-Multitaxa 61	
5.1	PLL-M em partículas	61
5.2	Estimação de Frequências	62
5.2.1	PLL-Multitaxa	62
5.2.2	Simulação e análise qualitativa	63
6	Conclusões e Trabalhos Futuros	67
6.1	Conclusões	67
6.2	Trabalhos futuros	68
	Bibliografia	69

1 Introdução

1.1 Análise Espectral

Um dos conceitos mais antigos em análise de sinais é o de frequência. As funções senoidais, ou as exponenciais complexas de modo geral, são fascinantes por possuírem diversas propriedades matemáticas interessantes, como por exemplo serem solução para diversas equações diferenciais e quando são entrada de um sistema linear e invariante no tempo, onde este sistema apresenta como saída um sinal da mesma classe da entrada, ou seja, outra exponencial complexa. O conceito de frequência sequer precisa estar ligado a séries temporais. Em imagens, ele também está presente na forma de frequência espacial. O próprio termo 'espectro' aparentemente foi introduzido por Newton enquanto estudava a decomposição da luz em diferentes cores. Tudo isso faz com que a decomposição de sinais em frequência seja um dos tópicos mais estudados em processamento de sinais [1, 2].

É quase impossível dissociar o tópico da famosa Transformada de Fourier (TF), mas olhando da perspectiva do que é basicamente uma TF há diversas outras bases sobre as quais se pode projetar um sinal qualquer de modo a extrair determinadas características do mesmo. Também é fato que hoje esta análise se dá na maioria dos casos por meios digitais, dado o poder computacional que se tem à disposição atualmente e a fundamentação teórica sólida que fornecem teoremas como o famoso Teorema da amostragem de Nyquist-Shannon [3, 4].

1.2 Aplicações

As aplicações da análise espectral são as mais diversas. Em processamento de imagens, tem-se seu uso para encontrar fronteiras e comprimir arquivos, reduzindo seu tamanho e economizando banda de transmissão e espaço de armazenamento [5].

Em processamento de fala, se pode utilizar análise espectral para extrair características de fonemas, de modo a posteriormente classificá-los [6].

Em biomedicina, se pode usar análise espectral para análise de eletrocardiogramas, e eletroencefalogramas na detecção de ondas alpha, beta, gamma e outras, que indicam determinadas atividades cerebrais e podem ser detectadas [7].

Além das demais áreas citadas, há uma área em especial que vem ganhando cada vez mais atenção e à qual se dedicam as simulações dos métodos que serão apresentados neste trabalho: a Qualidade da Energia Elétrica (QEE). Com o aumento constante do número de dispositivos não lineares conectados na rede elétrica, monitorar e filtrar harmônicos indesejados se torna indispensável. Atualmente existem diversas fontes de

energia e características de cargas. A grande interligação que existe entre os sistemas de potência também traz preocupação de grandes quedas no suprimento. Em resumo, o sistema de distribuição de energia elétrica está aumentando em complexidade, e desta maneira se fazem necessárias técnicas mais arrojadas para monitorá-lo e ajudar em seu controle [8].

Observando atentamente, o leitor se dará conta de que as simulações são em sua grande maioria implementadas considerando uma componente fundamental em 60 Hz e principalmente os harmônicos ímpares até o 15º, que são os que normalmente aparecem em sistemas de potência. O trabalho se baseia fortemente nos artigos [9] e [10], onde ambos os métodos propostos são concebidos com o intuito de analisar harmônicos e inter-harmônicos presentes comumente na rede elétrica.

O artigo [9] utiliza o algoritmo PLL desenvolvido em [11] para rastrear harmônicos e inter-harmônicos em conjunto com processamento multitaxa e bancos de filtros. O método final é capaz de rastrear de forma eficiente componentes espectrais desde que seja inicializado com valores de frequência próximos dos valores reais.

O artigo [10] utiliza uma estrutura de filtros adaptativos com dois estágios: um de estimação de frequência e outro de amplitude e fase. O primeiro estágio busca solucionar a equação de predição linear de um modelo AR utilizando o algoritmo de adaptação RLS, e assim estimar as frequências presentes no sinal. O segundo estágio utiliza novamente o RLS para encontrar os valores de amplitude de senoides e cossenoides geradas com a frequência do estágio anterior, obtendo desta maneira a amplitude e fase de cada componente.

1.3 Objetivos

O trabalho tem como objetivo discorrer sobre alguns métodos consolidados de análise espectral e selecionar os que tem a capacidade de rastreamento de inter-harmônicos no contexto de transmissão de energia elétrica. Também tentaremos propor um novo método, no capítulo 5, baseado nos citados anteriormente.

1.4 Divisão do Trabalho

O Capítulo 2 trata de uma fundamentação teórica e revisão bibliográfica traçando os fundamentos da estimação espectral e conceitos de processamento de sinais. Neste capítulo discorre-se sobre a teoria por trás dos métodos publicados em [9] e [10], os quais são a base deste trabalho, e também estão presentes algumas demonstrações relevantes para o entendimento dos mesmos.

O Capítulo 3 aborda especificamente o método Phase-Locked-Loop Multitaxa (PLL-M) [9], suas características, os problemas que este visa solucionar e porque foi concebido desta maneira. Além disso, estão incluídos resultados de simulações realizadas em MATLAB, na forma de figuras e tabelas.

O Capítulo 4 trata da etapa de estimação de frequências do método Adaptive Linear Neuron (Adaline) de dois estágios [10], e segue na mesma estrutura do capítulo 3: apresenta-se sua construção, pontos fortes, fracos e resultados de simulação.

O Capítulo 5 encerra a parte de resultados do trabalho com um método unindo de forma complementar os dois métodos anteriormente apresentados, utilizando o estágio de estimação de frequências presente no Capítulo 4 para inicializar o método PLL-M.

Finalmente, o Capítulo 6 apresenta conclusões a respeito do trabalho desenvolvido e proposições para trabalhos futuros.

2 Revisão Bibliográfica e Fundamentação Teórica

Neste capítulo serão abordadas as técnicas anteriormente mencionadas e que são o foco deste trabalho, a saber: PLL-M, análise de Prony, Adaline e Recursive Least Squares (RLS), bem como as convencionais Discrete Fourier Transform (DFT) e Short Time Fourier Transform (STFT). É importante ressaltar que esta será apenas uma abordagem superficial, não entrando, portanto, com profundidade nos assuntos tratados.

2.1 Análise espectral

Análise espectral considera o problema de encontrar a energia de um sinal, finito, distribuída em função da frequência, o que pode ser feito com métodos paramétricos ou não paramétricos. Métodos paramétricos assumem conhecimento do modelo com o qual se gerou o sinal em análise. Por exemplo, em alguns métodos, como o de análise de Prony, é assumido que o sinal é composto unicamente por um número conhecido de exponenciais complexas amortecidas. Para solucionar um problema de estimação paramétrica, o que se deve fazer é encontrar os parâmetros supostos. Por outro lado, métodos não paramétricos se baseiam unicamente em transformações do domínio temporal para o domínio da frequência, utilizando filtros que deixam passar a energia do sinal contida em uma determinada faixa do espectro, caso da DFT. Quando se tem uma boa estimação do modelo, os métodos paramétricos tendem a fornecer melhores resultados, enquanto que quando pouco se sabe sobre o sinal analisado, ou se o mesmo não está bem modelado, os métodos não paramétricos podem ser a melhor opção.

Algumas vezes é conveniente tratar os sinais de maneira determinística, e o será feito neste trabalho. Entretanto, como abordagem mais geral, pode-se tratá-los com enfoque probabilístico, isto é: admitir que não se pode de maneira alguma prever os valores assumidos por um sinal ao longo do tempo, mas pode-se estimar suas características por meios estatísticos.

É de se notar também que embora sinais reais sejam o caso mais comum na maioria das aplicações e no caso deste trabalho são praticamente regra, não necessariamente o tratamento de sinais complexos é muito mais complicado. Apenas por uma questão de praticidade e para deixar o trabalho mais conciso, não serão demonstrados os teoremas utilizados para entradas complexas quando não for necessário [1].

2.2 Densidade espectral de energia de um sinal determinístico

Seja $y_c(t)$ um sinal contínuo no tempo e $y[n]$, $n = 0, \pm 1, \pm 2, \dots$ um sinal discreto tal que $y[n] = y_c(nT_s)$, dado que T_s é o período de amostragem do sinal. Considere que $y[n]$

possui energia finita. Pode-se dizer então que a Discrete Time Fourier Transform (DTFT) de $y[n]$ está definida como:

$$Y(w) = \sum_{n=-\infty}^{+\infty} y[n]e^{-jwn} \quad (2.1)$$

onde w é denominada frequência angular e medida em radianos/segundo. A densidade espectral de energia do sinal $y(t)$ pode ser dada então por:

$$S(w) = |Y(w)|^2 \quad (2.2)$$

Pode-se interpretar este resultado de algumas formas, mas uma bem simples é lembrar que a DTFT de uma cossenoide é igual à soma dois impulsos refletidos no espectro de frequência:

$$\sum_{n=-\infty}^{+\infty} \cos(2\pi f n)e^{-jwn} = \pi\delta_w(2\pi f) + \pi\delta_w(-2\pi f), \quad w \in [0, 2\pi] \quad (2.3)$$

Uma relação muito parecida se estabelece no caso de uma senoide. Desta maneira, quando se imagina $y_c(t)$ composta unicamente de uma soma de senoides, seu espectro fica muito bem explicitado com a equação (2.1).

Uma outra relação importante se dá utilizando a autocorrelação de $y[n]$:

$$\rho_y[k] = \sum_{n=-\infty}^{+\infty} y[n]y^*[n-k] \quad (2.4)$$

$$\begin{aligned} \sum_{k=-\infty}^{+\infty} \rho_y[k]e^{-jwk} &= \sum_{n=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} y[n]y^*[n-k]e^{-jwk} = \\ &= \sum_{n=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} y[n]y^*[n-k]e^{-jwn}e^{jw(n-k)} = \\ &= \left[\sum_{n=-\infty}^{+\infty} y[n]e^{-jwn} \right] \left[\sum_{s=-\infty}^{+\infty} y[s]e^{-jws} \right]^* = Y(w)Y(w)^*, \end{aligned} \quad (2.5)$$

onde $s = n - k$.

Percebe-se então que a DTFT da autocorrelação de um sinal é igual à sua densidade espectral de energia.

2.2.1 Densidade espectral de potência

A maioria dos sinais com os quais se trabalha não têm um modelo puramente determinístico que os reproduza, não se sabe o valor exato que tomarão no futuro, e tam-

pouco se pode estender seu valor até o infinito, ou analisar infinitas amostras do mesmo. Para os casos reais, é mais conveniente lidar com sequências aleatórias ao longo do tempo, em que cada realização da sequência tem associada uma distribuição de probabilidade (até o momento de sua ocorrência, quando esta colapsa em algum valor). Para a maioria dos casos, serão considerados os sinais como processos estacionários em sentido amplo [12]. Para este trato aleatório, não é regra que os sinais a se tratar aqui possuam energia finita, mas é possível ao invés disso, estimar sua densidade de potência.

Assume-se agora que $y[n]$ é uma sequência de variáveis aleatórias, com o mesmo domínio da sequência anterior, e que $E[y[n]] = 0$ para todo n . Assim, todas as variáveis aleatórias possuem média nula. A função de autocorrelação de $y[n]$ está definida como:

$$r_y[k] = E[y[n] y[n-k]^*] \quad (2.6)$$

Define-se então a densidade espectral de potência como:

$$\phi(w) = \sum_{k=-\infty}^{+\infty} r_y[k] e^{-jwk} \quad (2.7)$$

Uma segunda definição para a Power Spectrum Density (PSD) pode ser dada como a DTFT do sinal com o número de amostras tendendo ao infinito da seguinte forma:

$$\phi(w) = \lim_{N \rightarrow \infty} \frac{1}{N} \left| \sum_{k=0}^{N-1} y[k] e^{-jwk} \right|^2 \quad (2.8)$$

Ambas definições são equivalentes. Será exposto mais adiante que uma das formas principais de se calcular a PSD é por meio de estimações da função de autocorrelação do sinal em questão.

2.2.2 Periodograma e correlograma

Uma forma bastante simples de estimar a PSD seria:

$$\hat{\phi}_p(w) = \frac{1}{N} \left| \sum_{k=0}^{N-1} y[k] e^{-jwk} \right|^2, \quad (2.9)$$

a qual é chamada periodograma. Outra forma poderia ser:

$$\hat{\phi}_c(w) = \sum_{k=-(N-1)}^{N-1} r_y[k] e^{-jwk}, \quad (2.10)$$

a qual é chamada correlograma. Dois estimadores para a autocorrelação serão apresentados abaixo, considerando $n = 0, 1, 2, \dots, N - 1$:

$$\hat{r}_y(k) = \frac{1}{N-k} \sum_{n=k}^{N-1} y[n] y^*[n-k] \quad (2.11)$$

$$\hat{r}_y(k) = \frac{1}{N} \sum_{n=k}^{N-1} y[n] y^*[n-k] \quad (2.12)$$

O primeiro estimador é chamado estimador padrão não polarizado, e o segundo é um estimador polarizado. É assim chamado (polarizado) um estimador cuja esperança não é o que visa estimar. Fato é também que $\hat{\phi}_c$ é igual a $\hat{\phi}_p$ se \hat{r}_y for estimada com estimador polarizado.

Muitas vezes, o estimador em (2.12) é preterido ao de (2.11), porque para muitos sinais a autocorrelação para valores grandes de k (valores considerados longe do 0) é muito baixa, e portanto não ajudaria fazer a correção do fator de normalização. Entretanto esse pode não ser bem o caso, porque os sinais muitas vezes são considerados senoides puras, as quais apresentam correlação em trechos periódicos. Em realidade, não serão discutidos com profundidade estimadores de correlação. A seguir serão analisados métodos relacionados à DFT. As vantagens e desvantagens destes métodos todos podem ser vistos à luz de processos estocásticos, ou pela análise das características do método em si. Tentar-se-á mostrar um pouco de ambos, os utilizando da maneira que for mais conveniente.

2.3 DFT

A DFT de uma sequência discreta $\{x[n]\}$ de tamanho N é definida como [3]:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{(-j2\pi kn/N)}, \quad k = 0, 1, 2, \dots, N-1 \quad (2.13)$$

Pode-se dizer que a sequência $X[k]$ é a DFT de $x[n]$. As duas sequências têm o mesmo tamanho, e $X[k]$ representa o mapeamento das frequências de $x[n]$ supondo estas estacionárias e com período igual ao analisado. Sendo T_s o período de amostragem da sequência $x[n]$, tem-se a seguinte relação:

$$R_s = \frac{1}{T_s N}, \quad (2.14)$$

onde R_s é igual à resolução de frequência de $X[k]$. O mapeamento de frequência é dado como:

$$f_k = R_s k, \quad k = 0, 1, 2, \dots, N/2 \quad (2.15)$$

Sendo N exatamente igual ao número de amostras compreendendo 1 ciclo fundamental do sinal $x[n]$, percebe-se que a DFT age como uma série de Fourier, onde valores absolutos de $X[k]$ são vistos como a amplitude de uma determinada frequência múltipla da componente fundamental f_1 , que é igual à resolução R_s . Para um sinal contendo apenas harmônicos de f_1 , e eventualmente algum valor médio, é possível extrair perfeitamente seus valores de amplitude e fase. Entretanto, para um sinal que possui componentes inter-harmônicos, não se pode fazer o mesmo mantendo-se o N igual ao número de amostras de 1 ciclo fundamental do sinal. Neste caso ocorre o chamado *espalhamento*. Já se pode notar algumas deficiências da DFT. Caso não seja amostrado um período exato do sinal analisado, ou um múltiplo inteiro de um período, a DFT interpreta, por exemplo, que existe um valor médio no sinal, além de outros conteúdos equivocados em frequências possivelmente inexistentes no sinal. Este fenômeno é denominado *spectral leakage* [3].

Como a DFT pressupõe um sinal estacionário, ela não é adequada para análise de sinais variantes no tempo. Para tanto, é possível utilizar de uma DFT de janela deslizante. Para uma janela de tamanho N :

$$X[k, m] = \sum_{n=0}^{N-1} x[m - n] e^{-j2\pi kn/N}, \quad k = 0, 1, 2, \dots, N - 1 \quad (2.16)$$

Desta forma, para cada $m > N$, tem-se uma janela de tamanho N onde será analisado o sinal. Existem algoritmos mais eficazes para efetuar este tipo de cálculo de forma recursiva, sem precisar calcular toda a DFT como se estivesse diante de uma janela completamente nova:

$$X[k, m] = C \{ X[k, m - 1] e^{j2\pi k/N} + (x[m] - x[m - N]) e^{j2\pi k/N} \}, \quad k = 1, 2, \dots, N/2, \quad (2.17)$$

sendo $C = 1/N$ para $k = N/2$ e igual a $2/N$ para os demais valores. O valor de k foi restringido levando-se em conta a simetria da DFT para sinais reais.

Pode-se usar diferentes tipos de janelas além da anterior, que é uma janela retangular, onde todos os termos da sequência têm igual peso no cálculo da DFT. O uso de outras janelas ajuda na amenização do *leakage*. Algumas janelas estão expostas abaixo.

Janela triangular:

$$w_{tri}(n) = 1 - \frac{|2n - N + 1|}{N - 1}, \quad 0 \leq n \leq N - 1 \quad (2.18)$$

Janela de Hamming:

$$w_{hm}(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N - 1}\right), \quad 0 \leq n \leq N - 1 \quad (2.19)$$

O uso deste tipo de janelamento atenua eventuais descontinuidades nas extremidades do sinal, auxiliando na medição dos parâmetros. Aplicar uma janela ao sinal significa

multiplicar ponto a ponto as sequências formadas por w e por $x[n]$. Desta forma, pode-se reescrever a equação (2.16) como:

$$X[k, m] = \sum_{n=0}^{N-1} x[m - n] * w(n) * e^{(-j2\pi kn/N)} , \quad k = 0, 1, 2, \dots, N - 1 \quad (2.20)$$

2.4 Análise de Prony

A análise de Prony foi desenvolvida em 1795 de modo a explicar a expansão de gases. Ela se assemelha em parte com a DFT, entretanto se propõe a ajustar uma soma de exponenciais complexas amortecidas a uma sequência de dados igualmente espaçados, ao passo de que a DFT apenas estima as exponenciais complexas e em subdivisões predefinidas da frequência de amostragem. A análise de Prony não é somente uma técnica de análise de sinais, mas também de identificação de sistemas amplamente utilizada em sistemas de potência, área biomédica, processamento de fala, decaimento radioativo, entre outras [2]. A análise de Prony é conhecida por não se comportar muito bem quando um sinal contém ruído, pois a técnica não faz distinção entre sinal e ruído e também ajusta as exponenciais às perturbações presentes. Seguindo o proposto em [10], tem-se o seguinte para um sinal $y[k]$ imaginando M exponenciais para se ajustarem a $2M$ amostras, respeitando o teorema da amostragem:

$$y[k] = \sum_{m=1}^M A_m e^{(\alpha_m + j2\pi f_m)(k-1)\Delta t + j\sigma_m} , \quad k = 1, 2, \dots, 2M, \quad (2.21)$$

onde f_m é a frequência das exponenciais, Δt é o intervalo de amostragem, α_m é o coeficiente de amortecimento, e σ_m é o ângulo de defasagem. Para o método proposto, se está interessado apenas em saber quais são as frequências presentes no sinal. Imaginando-se o sistema sendo Auto Regressive (AR) de ordem M , em qualquer instante k é possível prever o valor $y[k]$ considerando-se apenas as M amostras anteriores deste mesmo sinal. Pode-se então montar uma equação de diferenças, que neste caso também é um modelo de predição linear [6]:

$$y[k] = \sum_{m=1}^M a_m y[k - m] \quad (2.22)$$

É sabido que exponenciais complexas são soluções para tal modelo. Tendo em mãos os valores dos coeficientes $a_m, m = 1, \dots, M$, é possível montar o polinômio característico da equação, e suas raízes (exponenciais complexas) são soluções para o problema. De posse dos coeficientes a_m de-se então encontrar as raízes do polinômio seguinte:

$$P(z) = z^M - a_1 z^{M-1} - \dots - a_M z^0 \quad (2.23)$$

Como os coeficientes a_m são reais, as raízes complexas ocorrem em pares complexos conjugados. Desta maneira, cada par complexo representa uma possível senoide, já que $\cos(wt) = \frac{e^{j\omega t} + e^{-j\omega t}}{2}$:

$$f_m = (\text{Im}\{z_m\}/\text{Re}\{z_m\})2\pi f_s, \quad (2.24)$$

onde z_m é uma das M raízes e f_s é a frequência de amostragem.

As frequências f_m são possíveis soluções do sistema, não necessariamente elas estarão presentes. Qualquer combinação linear dessas senoides também é solução do sistema AR planteado. Para encontrar a forma da solução real, é necessário fazer uso das condições iniciais conhecidas do sistema. No caso, elas são as próprias amostras $y[k]$.

Os coeficientes de amortecimento α_m são os módulos das raízes complexas que foram encontradas. E se alguma raiz não é complexa, isto somente significa que uma exponencial é solução para a equação.

$$\alpha_m = |z_m| \quad (2.25)$$

2.4.1 Solução do modelo de predição linear

Existem diversas formas de solucionar um modelo deste tipo, sendo apresentadas algumas opções abaixo [12]:

2.4.1.1 Sistema linear

Conhecendo-se os valores $y[k]$, que são amostras do sistema analisado, pode-se montar um sistema linear considerando a equação anterior:

$$y[k] = \sum_{m=1}^M a_m y[k-m] \quad (2.26)$$

$$\begin{bmatrix} y[k-1] & y[k-2] & \dots & y[k-M] \\ y[k-2] & y[k-3] & \dots & y[k-M-1] \\ \vdots & & & \vdots \\ y[k-M] & y[k-M-1] & \dots & y[k-2M+1] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{bmatrix} = \begin{bmatrix} y[k] \\ y[k-1] \\ \vdots \\ y[k-M+1] \end{bmatrix} \quad (2.27)$$

Ou de forma simplificada:

$$\mathbf{Y}_k \mathbf{a} = \mathbf{y}_k \quad (2.28)$$

A forma mais simples de resolver este sistema é inverter a matriz \mathbf{Y}_k :

$$\mathbf{a} = \mathbf{Y}_k^{-1} \mathbf{y}_k \quad (2.29)$$

Um dos problemas com esta solução é que ela é computacionalmente custosa, já que não está implementada de maneira recursiva. Ainda pior que isso é o fato de que normalmente há ruído no sistema, e desta maneira se ajusta os valores de \mathbf{a} ao ruído também, o que em geral não é o desejado. Desta forma, cada vez que se calcula o vetor \mathbf{a} , o mesmo pode ser completamente diferente do anterior, levando a estimativas equivocadas. Modelando $y[k]$ como:

$$y[k] = \sum_{m=1}^M a_m y[k-m] + \xi_k, \quad (2.30)$$

em que ξ_k é ruído branco, tem-se um modelo estocástico do sinal, que possibilita pensar em soluções mais arrojadas. As equações que caracterizam a solução do modelo AR também são conhecidas como equações de Youle-Walker e têm como solução ótima o chamado algoritmo de Levinson-Durbin [6], o qual não será abordado neste trabalho. Entretanto, mesmo este modelo pode não representar corretamente o sinal. Se o sinal em questão é mais bem representado por $y(t) = \sum \sin(w_i t + \phi_i) + \xi(t)$, este modelo tem uma boa aproximação como AR para boas Signal to Noise Ratio (SNR). Caso contrário, ele pode não ser adequado.

2.4.2 Adaline

O elemento linear adaptativo, ou filtro Finite Impulse Response (FIR) adaptativo, também chamado de Adaline, é uma rede neural artificial composta de uma única camada e com função de ativação linear. Sendo \mathbf{x} as entradas, \mathbf{w} os pesos e b o valor de bias, o Adaline pode ser implementado matricialmente da seguinte maneira:

$$y = \begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_n \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} + b = \mathbf{x}^T \mathbf{w} + b \quad (2.31)$$

2.4.3 Filtros adaptativos

Filtros adaptativos são considerados sistemas não lineares, portanto a análise de seu comportamento é mais complexa que a de filtros com coeficientes fixos. Por outro lado, pelo fato de serem auto-projetados, por um ponto de vista mais prático, eles podem ser considerados menos complicados que os convencionais em termos de projeto [13].

O diagrama de blocos usual de um filtro adaptativo pode ser visto na Figura 1, onde k é o número da iteração, $x(k)$ é a entrada do filtro, $y(k)$ é o sinal de saída, $d(k)$ é o valor de referência e $e(k) = d(k) - y(k)$ é o valor de erro, necessário para o algoritmo de adaptação

do filtro. O algoritmo de adaptação é o processo usado para ajustar os coeficientes do filtro de modo a minimizar o erro de acordo com os critérios preestabelecidos. A escolha do algoritmo determina muitos aspectos do processo de adaptação, como a existência de soluções subótimas e complexidade computacional.

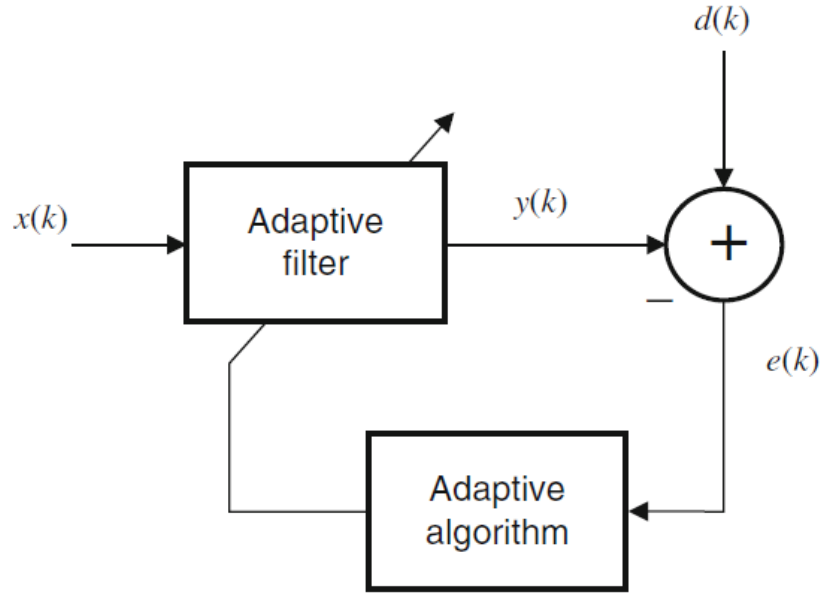


Figura 1 – Diagrama de blocos de um filtro adaptativo.

2.4.4 Adaptação via gradiente descendente

Uma das formas mais antigas e consagradas de otimizar uma função dentro dos métodos clássicos é seguir a direção do gradiente desta função, com relação às variáveis de interesse, para atualizar os valores destas variáveis de forma iterativa.

Seja um sistema descrito pela série temporal $u[n]$, $n = 0, 1, 2, \dots$, e um filtro de resposta impulsiva real $w_0, w_1 \dots w_{M-1}$:

$$y[n] = \sum_{k=0}^{M-1} u[n-k]w_k \quad (2.32)$$

Todo este desenvolvimento pode ser feito de maneira bastante similar considerando entradas complexas e filtros com coeficientes complexos, mas por questões de simplicidade será focado o caso em que ambos são reais. Supondo que se deseja que a saída deste filtro seja uma outra série temporal igualmente real $d[n]$, assim o erro seria:

$$e[n] = d[n] - y[n] \quad (2.33)$$

Considerando o caso de u estocástico, $e[n]$ é uma variável aleatória. Deseja-se então minimizar a esperança de $e^2[n]$. Seja a função custo:

$$J = E[e^2[n]] = E[(d[n] - y[n])^2] \quad (2.34)$$

Definindo o operador gradiente ∇_k :

$$\nabla_k J = \frac{\partial J}{\partial \mathbf{w}} = -2E[e[n]u[n-k]] \quad (2.35)$$

Para encontrar o ótimo com relação a esta função custo, $\nabla_k J$ deve ser igual a zero para todo k . Como a esperança do produto de duas variáveis aleatórias é sua correlação, $u[n-k]$ e $e[n]$ devem estar descorrelacionadas para todo k . Este resultado é chamado princípio da ortogonalidade. Um corolário deste princípio é que $y[n]$ também deve ser ortogonal, ou descorrelacionada com $e[n]$.

É mais conveniente trabalhar com a notação matricial destes resultados. Ao vetor coluna que contém as variáveis aleatórias $u[n-k]$, de $k = 0$ a $K = M - 1$, chama-se \mathbf{u}_n . O vetor de coeficientes w_k , também de $k = 0$ a $K = M - 1$, chama-se \mathbf{w}_n :

$$\mathbf{u}_n = \begin{bmatrix} u[n] & u[n-1] & \dots & u[n-M+1] \end{bmatrix}^T$$

$$\mathbf{w}_n = \begin{bmatrix} w_0 & w_1 & \dots & w_{M-1} \end{bmatrix}^T$$

Desenvolvendo a equação anterior, obtém-se:

$$\nabla J = -2E[e[n]\mathbf{u}_n] = 2E[\mathbf{u}_n \mathbf{u}_n^T \mathbf{w}_n - d[n]\mathbf{u}_n] \quad (2.36)$$

Desta última equação $E[d[n]\mathbf{u}_n]$ é o vetor de correlação cruzada entre $d[n]$ e as amostras atrasadas do sinal de entrada, ao qual se chama \mathbf{r}_{du} . Destaca-se também que $E[\mathbf{u}_n \mathbf{u}_n^T]$, a qual chama-se de \mathbf{R}_{uu} , é a matriz $M \times M$ de autocorrelação do mesmo sinal. Sendo assim:

$$\mathbf{R}_{uu} \mathbf{w}_n - \mathbf{r}_{du} = 0 \quad (2.37)$$

De onde se conclui que o vetor de coeficientes ótimos \mathbf{w}_{opt} é:

$$\mathbf{w}_{opt} = \mathbf{R}_{uu}^{-1} \mathbf{r}_{du} \quad (2.38)$$

Se u é um processo estacionário em sentido amplo, então é possível coletar amostras suficientes para fazer uma boa estimação das matrizes acima, encontrando assim um filtro muito mais apropriado para os propósitos deste trabalho. Desta maneira, se estima o conteúdo espectral de u com precisão. Repare que este método não é propriamente um gradiente descendente, já que se encontrou a solução de um passo apenas.

A solução acima tem sua utilidade. Porém, se a natureza do sinal se alterar, não haverá mais uma boa aproximação do mesmo. Neste caso, é melhor resolver estas equações de forma recursiva, de modo que o filtro possa se adaptar constantemente à mudanças

ocorridas. Para tanto, os algoritmos Least Mean Squares (LMS), Normalized Least Mean Squares (NLMS) ou RLS podem ser utilizados.

2.4.5 O algoritmo LMS

Para atualizar o vetor de pesos \mathbf{w}_n na direção oposta à do gradiente, define-se um coeficiente de aprendizagem μ . A cada iteração faz-se o seguinte:

$$\mathbf{w}_n = \mathbf{w}_{n-1} - \mu \nabla J$$

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu(\mathbf{r}_{du} - \mathbf{R}_{uu}\mathbf{w}_{n-1}) \quad (2.39)$$

O que diferencia o LMS de outros métodos de gradiente descendente é a forma de estimar as matrizes \mathbf{R}_{uu} e \mathbf{r}_{du} . Estas serão estimadas da seguinte forma:

$$\hat{\mathbf{R}}_{uu} = \mathbf{u}_n^T \mathbf{u}_n \quad (2.40)$$

$$\hat{\mathbf{r}}_{du} = d[n] \mathbf{u}_n \quad (2.41)$$

2.4.6 O algoritmo NLMS

Uma melhoria no algoritmo anterior pode ser proposta, multiplicando-se a parte que está entre parênteses na equação (2.39) pela inversa da matriz de autocorrelação dos dados de entrada, \mathbf{R}_{uu}^{-1} , da seguinte forma:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu \mathbf{R}_{uu}^{-1}(\mathbf{r}_{du} - \mathbf{R}_{uu}\mathbf{w}_{n-1}) = \mathbf{w}_{n-1} + \mu(\mathbf{w}_{opt} - \mathbf{I}\mathbf{w}_{n-1}) = \mathbf{w}_{opt} \quad (2.42)$$

Em teoria, este algoritmo convergiria em apenas uma iteração. Resta o empecilho de estimar a inversa da matriz de autocorrelação, \mathbf{R}_{uu}^{-1} . No entanto, pode-se fazer uma aproximação da forma $\hat{\mathbf{R}}_{uu} + \mathbf{I}\epsilon$, que é inversível e confere uma inversa válida, para valores pequenos de ϵ . E, após alguma álgebra [14], que pode ser encontrada em [12], chega-se ao seguinte:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu \frac{\mathbf{u}_n}{|\mathbf{u}_n|^2 + \epsilon} (d[n] - \mathbf{u}_n^T \mathbf{w}_{n-1}) = \mathbf{w}_{n-1} + \mu \frac{\mathbf{u}_n}{|\mathbf{u}_n|^2 + \epsilon} e[n] \quad (2.43)$$

A este algoritmo chama-se de LMS normalizado, ou NLMS. Há ainda o estudo de convergência para valores pequenos de μ , que também pode ser encontrado em [13] e [12],

tanto para o LMS quanto para o NLMS. Mas, de modo geral, o NLMS deve alcançar a sua convergência se respeitadas todas as condições estabelecidas anteriormente e escolhendo-se valores de $\mu < 1$.

2.4.7 RLS

O método RLS (*Recursive Least Squares*), ou mínimos quadrados recursivo, é um método de adaptação que visa a minimização da soma dos quadrados da diferença entre o sinal de referência e o sinal de saída do filtro em questão (o erro). O RLS pode ser obtido a partir do LMS, e é conhecido por possuir rápida convergência, tendo boa performance em sistemas variantes no tempo, como o caso do interesse neste trabalho. Isto vem ao custo de certa complexidade computacional aliada a problemas de estabilidade [13].

Voltando ao modelo do filtro adaptativo da seção anterior, havia sido feita uma estimação bastante simplória de \mathbf{R}_{uu} para o LMS. Agora se pode imaginar uma outra questão: nos casos anteriores, somente se aproveitava as M últimas amostras do sinal para cada iteração, mas se por exemplo faz-se uma média das $\mathbf{R}_{uu}^{(k)}$ aproxima-se mais da matriz verdadeira de autocorrelação. Pode-se escolher então um novo estimador para \mathbf{R}_{uu} :

$$\mathbf{S}_d^{-1} = \hat{\mathbf{R}}_{uu} = \frac{1}{k+1} \sum_{i=0}^k \lambda^{k-i} \mathbf{U}_i \mathbf{U}_i^* \quad (2.44)$$

Esta definição, em realidade, é uma média ponderada em uma série geométrica, e se $\lambda < 1$ as últimas amostras (mais recentes) contam mais do que as primeiras (mais antigas). Repare que se $\lambda = 1$, tem-se um estimador consistente da matriz. λ se chama fator de esquecimento.

Outra forma de se chegar ao RLS é por sua função objetivo, dada por:

$$\xi(k) = \sum_{i=0}^k \lambda^{k-i} \varepsilon^2(i) = \sum_{i=0}^k \lambda^{k-i} [d(i) - \mathbf{x}(k) \mathbf{w}(k)] \quad (2.45)$$

Onde ε é o erro a posteriori no instante i . É possível estimar a matriz \mathbf{R}_{uu} de forma recursiva, para evitar cálculos desnecessários, e melhor que isso, sua inversa também pode ser calculada de maneira recursiva. A inversa pode ser substituída na expressão obtida para o NLMS. A dedução no entanto não interessa tanto neste trabalho, e ela pode ser encontrada em [12] e [13]. Fica-se então com o seguinte resultado final:

$$\phi_k^{-1} = \lambda^{-1} \left[\phi_{k-1} - \frac{\phi_{k-1} \mathbf{u}_k^T \mathbf{u}_k \phi_{k-1}}{\lambda + \mathbf{u}_k \phi_{k-1} \mathbf{u}_k^T} \right] \quad (2.46)$$

Os pesos então podem ser atualizados da seguinte maneira:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \phi_n^{-1} e[n] \mathbf{u}_n \quad (2.47)$$

Uma recomendação de inicialização de ϕ_0 é $\phi_0 = \delta \mathbf{I}$, sendo δ o inverso da potência estimada do sinal.

2.5 PLL

Um Phase-Locked-Loop (PLL) digital, assim como sua versão analógica, visa determinar os parâmetros de um processo estocástico como sendo uma onda senoidal. Desta forma, o PLL tenta estimar parâmetros de Amplitude, Fase e Frequência. Tem-se diversas variantes digitais e analógicas [15], e a variante utilizada neste trabalho foi proposta por Ziarani em 2004 [11]. Uma breve demonstração será realizada abaixo.

Seja $u(t)$ uma função variante no tempo, e $y(t) = A \sin \phi(t)$ um sinal periódico senoidal com A sendo a amplitude do sinal e $\phi(t)$ a sua fase. Tendo este sinal uma frequência constante, pode-se escrever $\phi(t) = wt + \delta$, sendo w igual à frequência e δ um valor de fase constante, também chamada de fase inicial. Escrevendo de maneira mais geral:

$$u(t) = \sum_{i=0}^{\infty} A_i \sin \phi_i(t) + n(t), \quad (2.48)$$

onde $n(t)$ representa um distúrbio, como um ruído. Na realidade, como todos os parâmetros podem variar com o tempo, é conveniente escrever então um conjunto $\Psi(t) = [A(t), w(t), \delta(t)]$, o qual contém todos os parâmetros de um possível sinal $y(t)$.

Define-se a função d :

$$d^2(t, \Psi(t)) = [u(t) - y(t, \Psi(t))]^2 = e^2(t) \quad (2.49)$$

Assim sendo, o conjunto $\Psi(t)$ ótimo que minimiza a função d^2 será adotado como função custo $J(\Psi(t), t)$. Os parâmetros $\Psi(t)$ podem ser estimados por meio de gradiente descendente.

$$\frac{d\Psi(t)}{dt} = -\boldsymbol{\mu} \frac{\partial J(\Psi(t), t)}{\partial \Psi(t)}, \quad (2.50)$$

$$\boldsymbol{\mu} = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}, \quad (2.51)$$

onde $\frac{d\Psi(t)}{dt}$ denota o vetor na direção na qual são atualizados os valores de $\Psi(t)$ a cada iteração, e $\boldsymbol{\mu}$ é uma matriz diagonal contendo as constantes de atualização referentes a cada parâmetro. Como lida-se com estimativas, usa-se a notação $\hat{\Psi}(t)$:

$$\begin{bmatrix} \frac{d\hat{A}(t)}{dt} \\ \frac{d\hat{w}(t)}{dt} \\ \frac{d\hat{\delta}(t)}{dt} \end{bmatrix} = -\boldsymbol{\mu} \begin{bmatrix} \frac{\partial \epsilon^2(t)}{\partial \hat{A}} \\ \frac{\partial \epsilon^2(t)}{\partial \hat{w}} \\ \frac{\partial \epsilon^2(t)}{\partial \hat{\delta}} \end{bmatrix} \quad (2.52)$$

Obtém-se então o seguinte:

$$\frac{d\hat{A}(t)}{dt} = 2m_1 e(t) \sin\left(\int_0^t \hat{w}(\tau) d\tau + \hat{\delta}(t)\right) \quad (2.53)$$

$$\frac{d\hat{w}(t)}{dt} = 2m_2 e(t) \hat{A}(t) t \cos\left(\int_0^t \hat{w}(\tau) d\tau + \hat{\delta}(t)\right) \quad (2.54)$$

$$\frac{d\hat{\delta}(t)}{dt} = 2m_3 e(t) \hat{A}(t) \cos\left(\int_0^t \hat{w}(\tau) d\tau + \hat{\delta}(t)\right) \quad (2.55)$$

$$\frac{d\hat{\phi}(t)}{dt} = \hat{w}(t) + \frac{\hat{\delta}(t)}{dt} \quad (2.56)$$

Por conta do fator temporal t que aparece solto na equação referente a \hat{w} , esse sistema é variante no tempo, o que o torna instável. No entanto, uma solução heurística é substituir t por uma constante m_4 , que pode ser absorvida pela constante m_2 , tornando o sistema invariante no tempo e fazendo com que este sistema de equações seja útil na prática. Desta maneira, tem-se o seguinte:

$$\frac{d\hat{A}(t)}{dt} = 2\mu_1 e(t) \sin\left(\int_0^t \hat{w}(\tau) d\tau + \hat{\delta}(t)\right) \quad (2.57)$$

$$\frac{d\hat{w}(t)}{dt} = 2\mu_2 e(t) \hat{A}(t) \cos\left(\int_0^t \hat{w}(\tau) d\tau + \hat{\delta}(t)\right) \quad (2.58)$$

$$\frac{d\hat{\phi}(t)}{dt} = \hat{w}(t) + 2\mu_3 e(t) \hat{A}(t) \cos\left(\int_0^t \hat{w}(\tau) d\tau + \hat{\delta}(t)\right) \quad (2.59)$$

Usando o método *Euler Forward*:

$$A[n+1] = A[n] + \mu_1 T_s e[n] \sin(\phi(t)) \quad (2.60)$$

$$w[n+1] = w[n] + \mu_2 T_s e[n] \cos(\phi(t)) \quad (2.61)$$

$$\phi[n+1] = \phi[n] + T_s w[n] + \mu_3 T_s e[n] \cos(\phi(t)) \quad (2.62)$$

Com isto, está definido o conjunto de equações discretas que serão utilizadas para rastrear frequências no restante do trabalho.

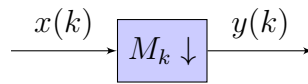
2.6 Processamento Multitaxa

Em determinadas situações, é desejável alterar a taxa de amostragem de um sinal para realizar certos tipos de processamento. Por exemplo, será visto em alguns resultados nas próximas seções que o algoritmo PLL descrito anteriormente é sensível à taxa de amostragem em relação à frequência rastreada. Também será visto que processar sinais na taxa necessária para abranger todo o espectro que se quer analisar é extremamente custoso computacionalmente e não é garantia de melhores resultados. Por isso, deve-se considerar o uso de uma estrutura multitaxa que diminua a frequência base.

Serão analisadas a seguir as implicações de um abaixamento na frequência de amostragem.

2.6.1 *Downsampling*

Uma estrutura *downsampler* tem a forma seguinte:



Representa-se a estrutura como um sistema que reproduz em sua saída apenas amostras onde k é múltiplo de M_k . Pode-se escrever $y[n] = x[nM_k]$ $n = 1, 2, 3, \dots$. Considerando que $x[k]$ foi amostrado com a frequência de Nyquist f_s , o efeito final é o mesmo que se tivesse amostrado $x[k]$ com f_s/M_k . Pode-se dizer que será observado *aliasing* no resultado final. Haverá portanto sobreposição no espectro de frequência referente a $X(w)$ quando se analisar $Y(w)$. Entretanto, é possível prever onde estarão localizadas as componentes espectrais presentes no sinal completo que se está subamostrando, e embora não se possa saber exatamente qual é o valor correspondente às frequências originais, pode-se saber no mínimo qual é sua soma.

Uma demonstração formal dos efeitos observados pode ser encontrada em [3]. Aqui será realizada uma abordagem mais intuitiva, dada uma componente espectral de $x[k]$ referente a w_0 . Supõe-se que é feita uma subamostragem neste sinal, e agora sua frequência de amostragem que era f_s se torna f_s/M_k . Agora, se $w_0 < f_s/(2M_k)$, as componentes dessa frequência não saem do lugar, elas continuam onde estavam anteriormente. Isso não significa que seu espectro não seja alterado nas frequências mais baixas que a nova taxa de amostragem, somente quer dizer que elas sofrem interferência no mesmo lugar onde estavam antes. Agora, se $w_0 > f_s/(2M_k)$, então ela não poderá de forma alguma

ser encontrada no mesmo local, pois simplesmente não há como visualizá-la com DFT ou DTFT. Mas então onde vão parar essas frequências? Bom, onde se situaria qualquer uma que sofre *aliasing*. Ela dá voltas no círculo de frequências. O círculo de frequências sempre vai de 0 a 2π , e as frequências sempre caminham nele no sentido anti-horário. Para cada vez que a frequência de interesse (a qual será chamada f_i) for maior que f_s , a frequência em questão dá uma volta no círculo. Um algoritmo simples será apresentado quando for exposto o método do PLL multitaxa, mas por ora pode-se dizer que o ângulo referente à nova frequência será o resto da divisão f_i/f_s multiplicado por 2π .

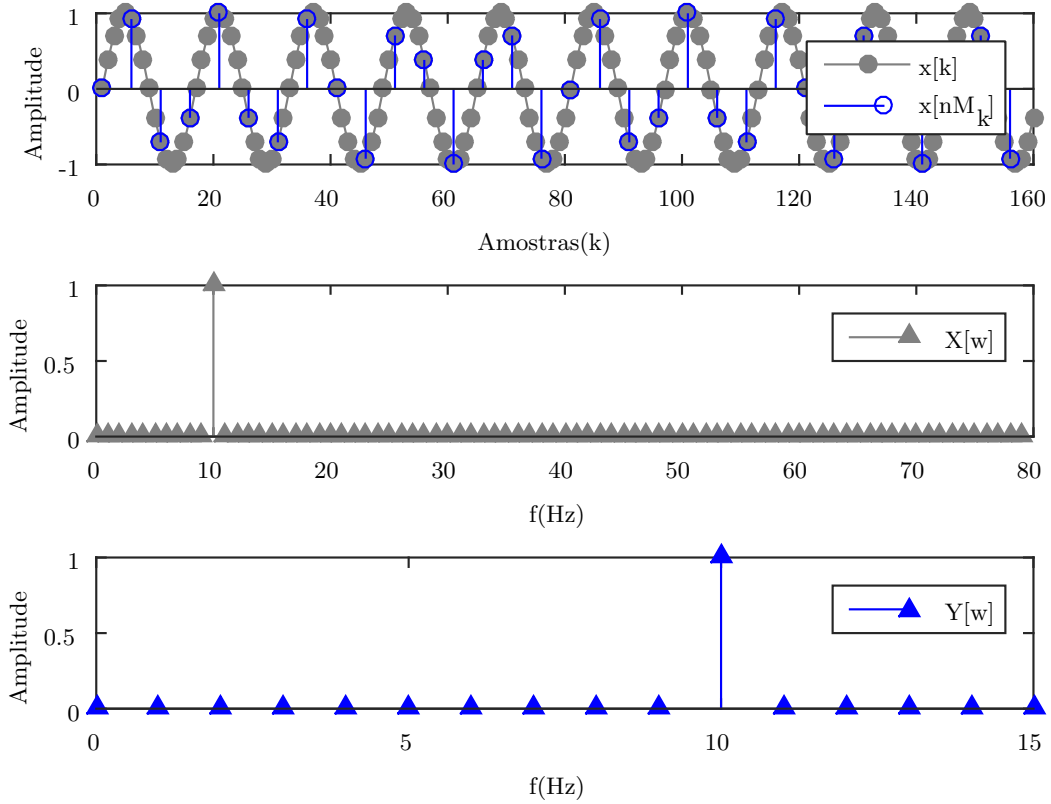


Figura 2 – Downsampling para $f_s = 160Hz$, $f_0 = 10Hz$, $M_k = 5$

No exemplo da Figura 2, tem-se $f_s = 160Hz$ e $M_k = 5$. Sendo a frequência de interesse $f_0 = 10Hz$, a frequência de amostragem depois da subamostragem ainda seria maior que $2f_i$. Desta forma, ela não se move no espectro de $Y(w)$, como pode ser observado na figura. Entretanto, no caso de $M_k = 10$, a situação se altera.

$$w_i = \text{rest}(f_i M_k, f_s)2\pi \quad (2.63)$$

O operador *rest* usado na equação 2.63 calcula a parte fracionária da divisão do

primeiro argumento pelo segundo. Usando esta equação para $M_k = 10$ tem-se $w_i = \frac{5}{4}\pi$, que é ligeiramente maior que π ; portanto já se pode saber que esta componente sofre *aliasing*, se pode calcular o valor em Hz da frequência final com $f_i = (2 - w_i/\pi)f_s/M_k$ encontramos $f_i = 6Hz$, como se nota na Figura 3.

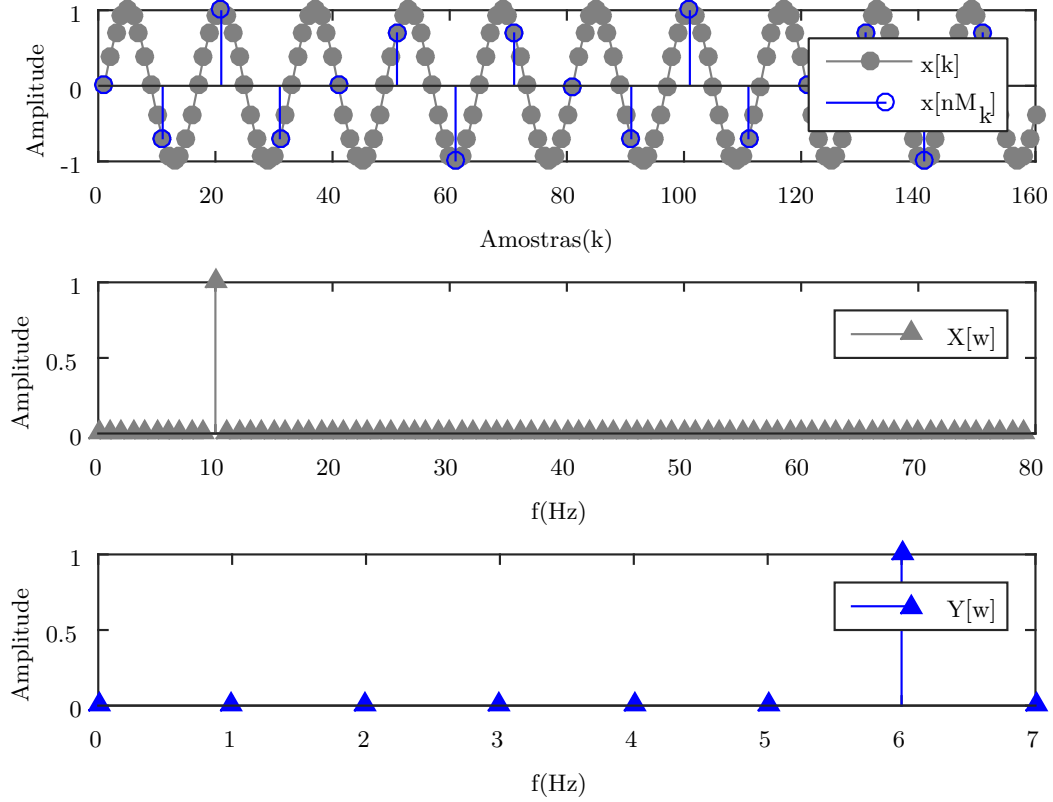


Figura 3 – Downsampling para $f_s = 160Hz$, $f_0 = 10Hz$, $M_k = 10$

3 Estrutura PLL-Multitaxa

Como visto no capítulo anterior na seção referente ao PLL, esse método é capaz de minimizar o erro quadrático médio entre um sinal $y(t)$ e uma componente senoidal para ao menos um mínimo local. Neste capítulo ganhará forma uma estrutura reunindo o PLL e o processamento multitaxa, com o objetivo de reduzir a complexidade computacional e conseguir um algoritmo mais versátil.

3.1 Desempenho do PLL

Nas figuras a seguir, poderá se observar como converge o algoritmo em diferentes situações, sendo que todas foram simuladas para um sinal cujo componente fundamental possui amplitude igual a 180 V, e as constantes do PLL são 300, 500, e 6, respectivamente. A amostragem foi realizada com frequência igual a 7680 Hz, partindo de condições iniciais $f_i = 60\text{Hz}$ e $A = 0$. Percebe-se pela simulação que o algoritmo converge rapidamente mesmo com ruído, entretanto, na presença de harmônicos com quantidade de energia considerável em relação à fundamental, a convergência já é bastante comprometida. Ainda assim, em valor médio, a estimação se mostra correta. Conclui-se que é possível estimar harmônicos diretamente com o algoritmo PLL obtido, entretanto é conveniente aliá-lo a outras técnicas para que se diminua o erro da estimação. Os resultados das simulações podem ser vistos nas Figuras 4 e 5.

3.2 Banco de filtros

A solução encontrada em [9] para parte dos problemas anteriormente citados é o uso de um pré-processamento com filtros passa-banda, para melhorar a relação sinal ruído, e posteriormente subamostragem, para diminuir a complexidade computacional, de modo também que não seja necessário mudar as constantes do algoritmo PLL.

O conjunto de filtros utilizado é uma cascata de dois filtros IIR com a seguinte função de transferência:

$$H_{bp}(z) = \frac{1 - \alpha}{2} \frac{1 - z^{-2}}{1 - \beta(1 - \alpha)z^{-1} + \alpha z^{-2}} \quad (3.1)$$

$$\beta = \cos(w_0) \quad (3.2)$$

O parâmetro α modifica a seletividade do filtro e se situa entre 0 e 1, para que este seja estável. Quanto mais próximo de 1, mais seletivo é o filtro. O parâmetro β modifica

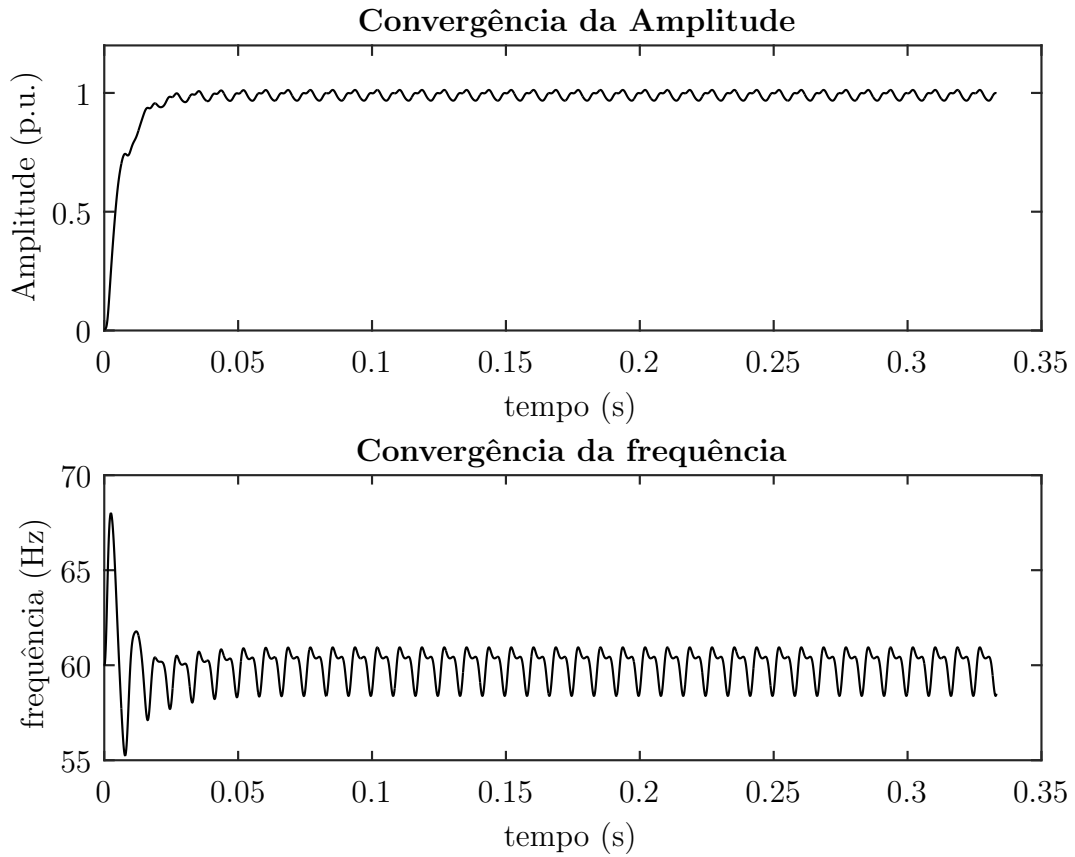


Figura 4 – Convergência na presença do 3º e 5º harmônicos, além de ruído com SNR=10 dB

a frequência central do filtro de acordo com a equação (3.5), onde w_0 é a frequência normalizada de acordo com a amostragem.

Este filtro é uma boa escolha por alguns motivos:

- Ele rechaça completamente a componente DC do sinal.
- Tem atraso de fase nulo na frequência central.
- É paramétrico, suas características dependem dos parâmetros α e β , os quais modificam propriedades muito específicas do filtro, sendo então muito fácil utilizá-lo e modificá-lo em tempo real.

Tem-se talvez como única desvantagem o atraso de grupo que é máximo para a frequência central, e quanto mais seletivo é o filtro, maior é esse atraso, como pode ser observado na Figura 6. Deve-se também ter atenção com a frequência de amostragem, pois quanto maior for, menos seletivo um mesmo α será. Se por exemplo, se mantém α e aumenta-se f_s , frequências que antes estavam normalizadas mais distantes da frequência central anterior, agora estariam mais próximas, por assim dizer. Dessa forma, quanto

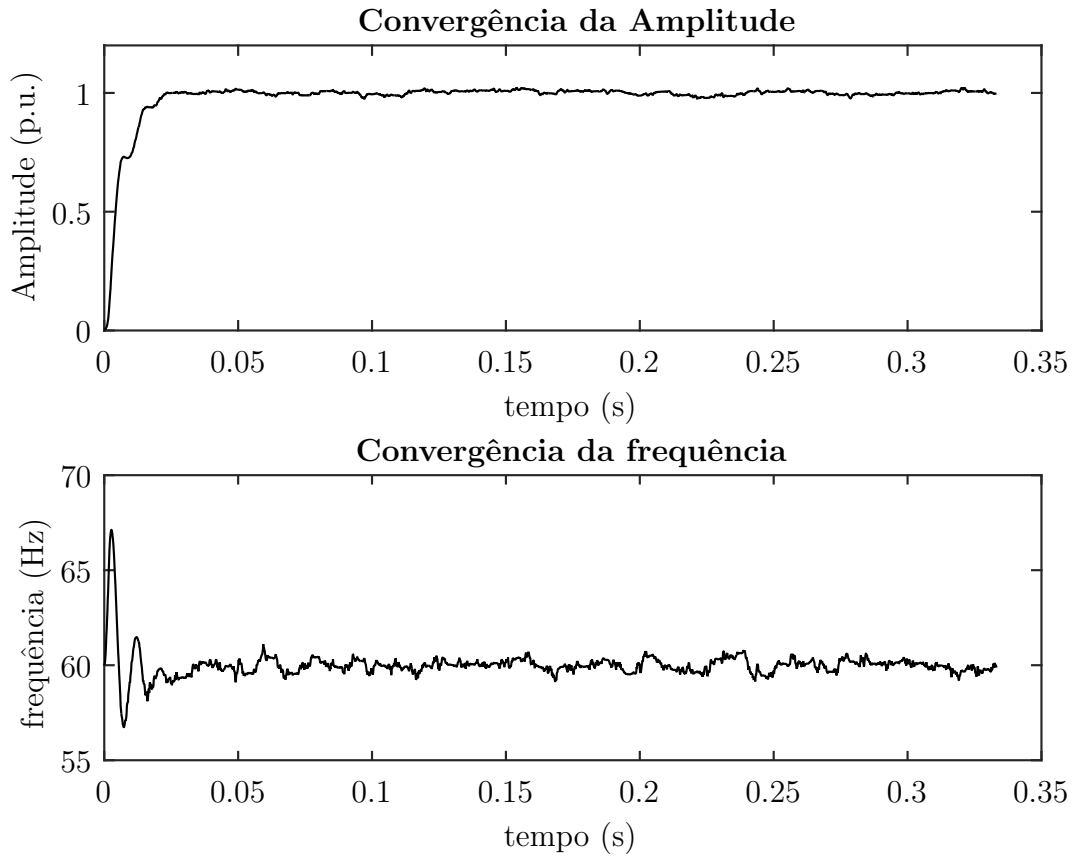


Figura 5 – Convergência na presença apenas de ruído, com SNR=10 dB

maior é f_s mais seletivo deve ser o filtro para a separação das mesmas frequências. Isso acaba se tornando um jogo de balanceamento destes parâmetros, pois como vimos anteriormente, um α maior também eleva o atraso de grupo, entretanto se tem mais amostras em menos tempo devido ao aumento em f_s . Todos estes efeitos estão muito bem descritos por J. Rodrigues em seu trabalho [16]. Ao final, será utilizado $\alpha = 0.975$ e dois filtros em cascata para o restante das simulações.

3.3 Uso da estrutura multitaxa

Depois de passado pelo filtro passa-banda, pode-se realizar a subamostragem do sinal, uma vez que em tese eliminou-se os harmônicos mais distantes quase que completamente, e estes não interferirão na estimação. Dentro da revisão bibliográfica foi discutido o perfil desta interferência e também como encontrar a posição de uma frequência depois de esta sofrer *aliasing*. Agora é realizada a exposição do algoritmo em linguagem C de uma função simples capaz de calcular esta frequência:

```
float freq_finder(const float Fs,
```

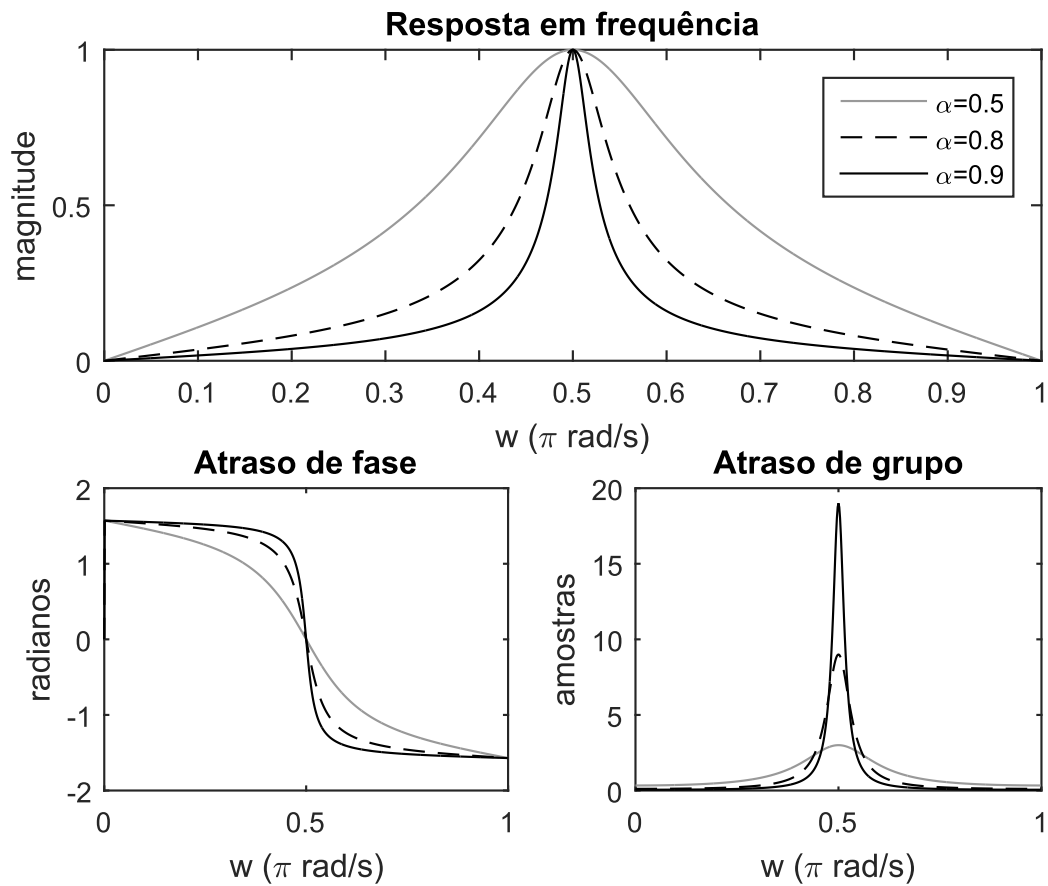


Figura 6 – Características do filtro com $w_0=0.25$

```

const float f_init , int *flag){

float f_aparente=f_init;
*flag=1; //valor padrao para a variavel

//verifica se a frequencia investigada sofre aliasing
if(f_init > (Fs/2)){
    //calcula o resto da divisao entre as frequencias
    f_aparente=fmod(f_init/Fs, 1);
    if(f_aparente <= 0.5){
        //se o resto e menor ou igual a 1/2, estamos
        //no semicirculo positivo
        f_aparente=f_aparente*Fs;
    }
    else{
        //caso contrario, estamos no semicirculo negativo
        f_aparente=(1-f_aparente)*Fs;
    }
}

```

```

        *flag=-1;
    }
}

return f_aparente;
}

```

A função recebe como argumentos a frequência de amostragem F_s , a frequência rastreada f_{init} , e o ponteiro para a variável $flag$, que indica se a frequência encontrada estava no semicírculo positivo, quando vale 1, ou negativo, quando vale -1 ($\theta < \pi$ ou $\theta > \pi$). É importante guardar esta informação porque ela será necessária para recuperar a frequência original estimada.

É possível observar na Figura 7 a localização de duas frequências f_1 e f_2 no círculo. Suponha que a frequência de amostragem é $f_s = 240 \text{ Hz}$, $f_1 = \frac{240}{2} \frac{3}{4} \text{ Hz} = 90 \text{ Hz}$ e $f_2 = \frac{240}{2} \frac{4}{3} \text{ Hz} = 160 \text{ Hz}$. Encontra-se então suas posições no círculo utilizando o algoritmo citado. Agora, repare que diferentes frequências serão mapeadas nas mesmas posições do círculo. Por exemplo, $f_1 = \frac{240}{2} \frac{11}{4} \text{ Hz} = 330 \text{ Hz}$ também é mapeada no mesmo ângulo. Então, não existe uma função inversa que devolva as frequências mapeadas para seus valores reais.

Também é importante se atentar ao fato de que quando se aumenta f_1 levemente, se está aumentando sua frequência aparente, mas quando se faz o mesmo com f_2 se está diminuindo sua frequência aparente. Por isso é importante guardar a variável 'flag'. Ela diz se acréscimos positivos em frequência aparente condizem a acréscimos ou decréscimos na frequência real, e uma vez que não existe uma função inversa que devolva a frequência real dada a aparente, deve-se utilizar da variação na estimação aparente e do valor inicial mapeado para recuperar a estimação real.

Uma coisa que se deve ter em mente é que algumas frequências se localizarão muito próximas a zero, e terão frequência aparente muito pequena. Isso dificulta muito a análise, e é desejável testar a localização antes de iniciar o algoritmo e fazer uma mudança no valor de subamostragem caso necessário.

$$\hat{f} = f_{init} + \Delta f_{aparente} flag \quad (3.3)$$

3.4 Variação da frequência central do banco de filtros

A estrutura multitaxa, apresentada na Figura 8, consiste em passar o sinal de entrada $x(n)$ pelo banco de filtros, sub-amostrar e passar este sinal para os respectivos PLLs. Como o filtro utilizado é bastante seletivo, a frequência estimada deve controlar

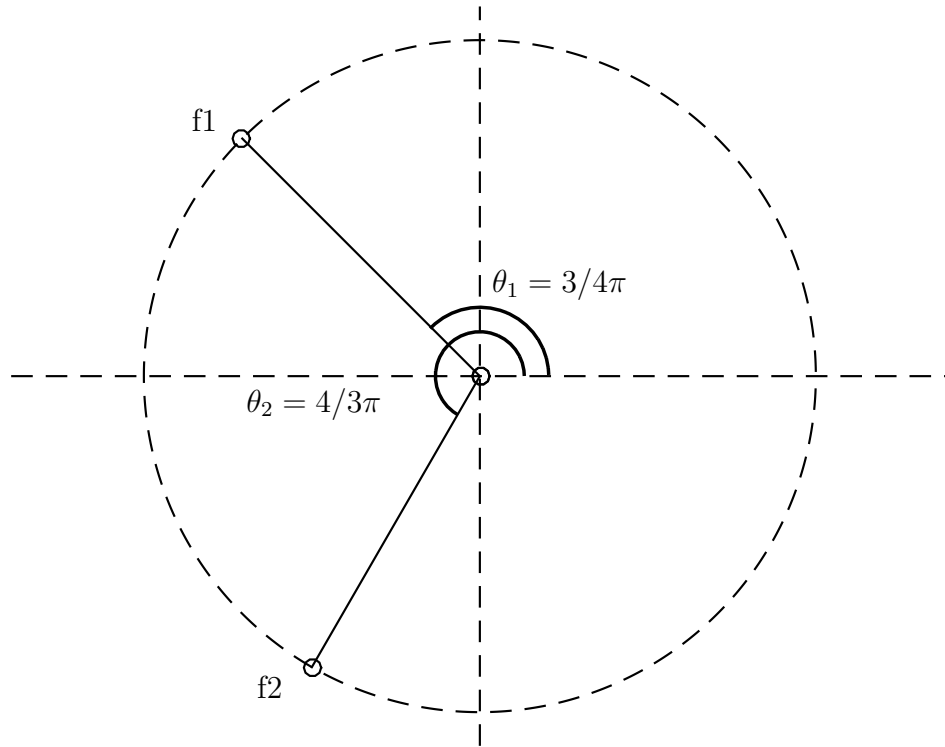


Figura 7 – Círculo de frequências

o banco de filtros centrando a frequência corretamente. A maneira mais básica de realizar este controle seria alimentar o banco diretamente com as frequências obtida no PLL, entretanto este método é inviável. Foi visto, na seção sobre o algoritmo PLL utilizado, que este é altamente não linear e complexo de se analisar a convergência. Fazer uma realimentação deste tipo muda o sistema e pode torná-lo instável, ou prejudica sua convergência. A alternativa encontrada por J. Rodrigues [16] é utilizar a média das últimas L estimações, assim o filtro passa-banda se move de maneira mais suave e não prejudica tanto o PLL. Desta maneira:

$$w_0[k] = \frac{1}{L} \sum_{i=k-L+1}^k \hat{w}[i] \quad (3.4)$$

Uma outra opção é fazer a estimação com uma série geométrica, ou soma convexa, que pode ser calculada de forma recursiva:

$$w_0[k] = w_0[k-1](1-\lambda) + \lambda \hat{w}[k], \quad (3.5)$$

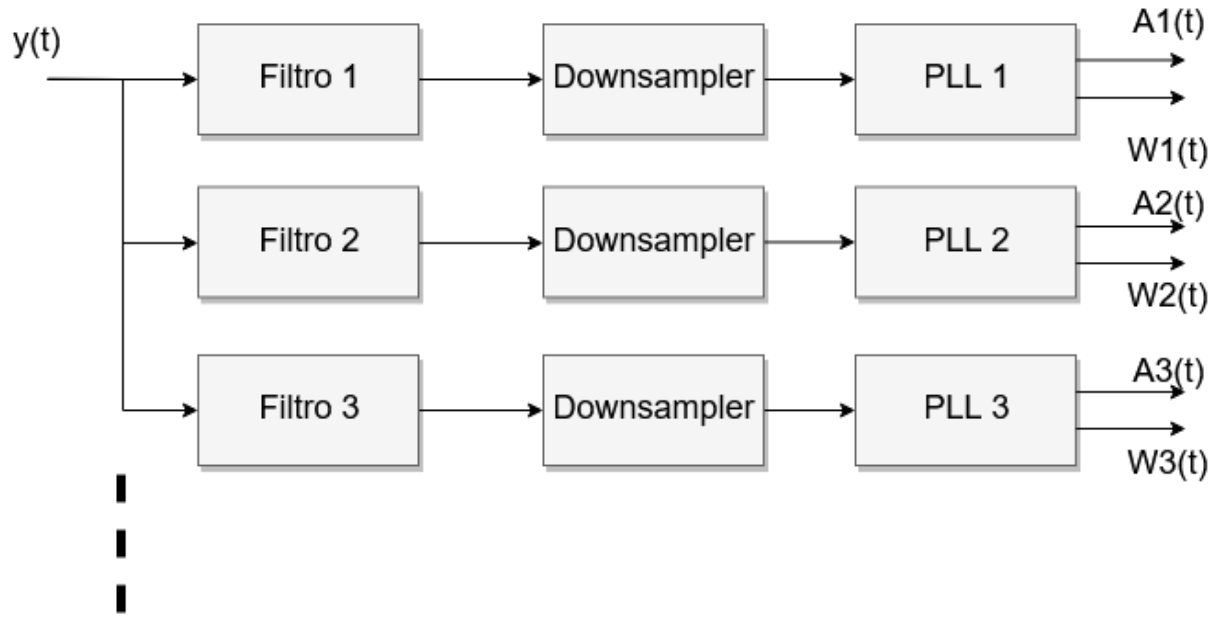


Figura 8 – Esquema multitaxa PLL

onde λ é uma constante entre 0 e 1.

Simulando as três opções, a alimentação direta da estimação realmente se mostra mais instável, e com maior tempo de convergência, enquanto que o método geométrico em geral é levemente mais estável e de mais rápida convergência que o de média, além de ser menos custoso computacionalmente. Todos foram testados para as mesmas constantes do exemplo anterior, amplitude de 180 V e 60 Hz iniciais, a constante de subamostragem escolhida foi $M_k = 16$. Depois de 1 segundo de simulação, é aplicado um degrau na frequência de -2 Hz. Também é passado um filtro média móvel de 16 amostras nos resultados finais de \hat{w} e \hat{A} . Os resultados dessas simulações são mostrados nas Figuras 9 e 10.

3.5 Síntese da estrutura PLL Multitaxa

1. Passar o sinal por um banco de filtros;
2. Abaixar a frequência por meio da subamostragem em M_k amostras;
3. Calcular a frequência aparente f_a da componente a rastrear;
4. Inicializar o PLL com esta f_a e a guardar para recompor a frequência original;
5. Realizar as estimações de w e A , usando a equação (3.5) e utilizar a série geométrica para realimentar o banco de filtros com a frequência calculada.

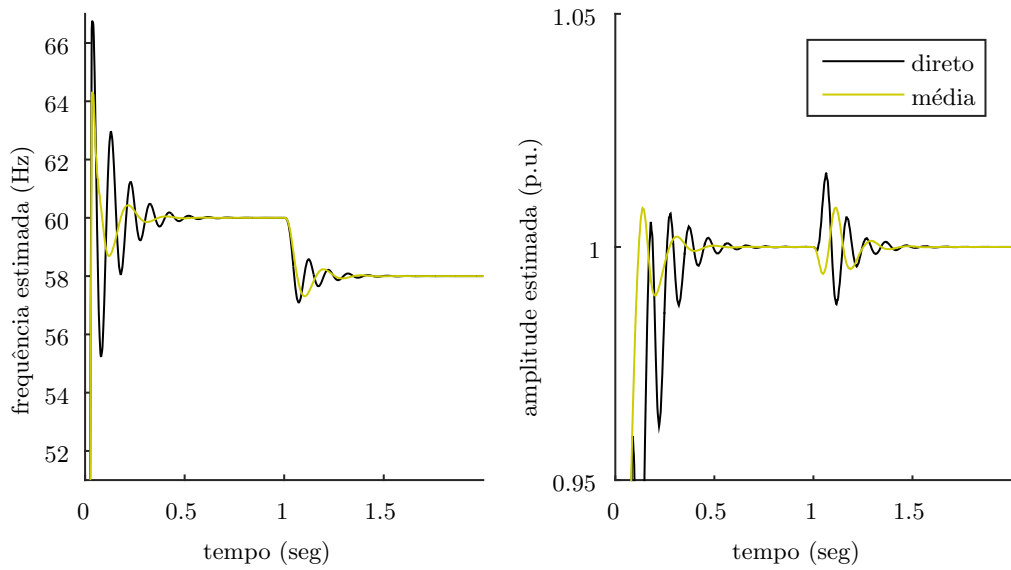


Figura 9 – Comparação entre o método de média (com 24 amostras) e o de alimentação direta

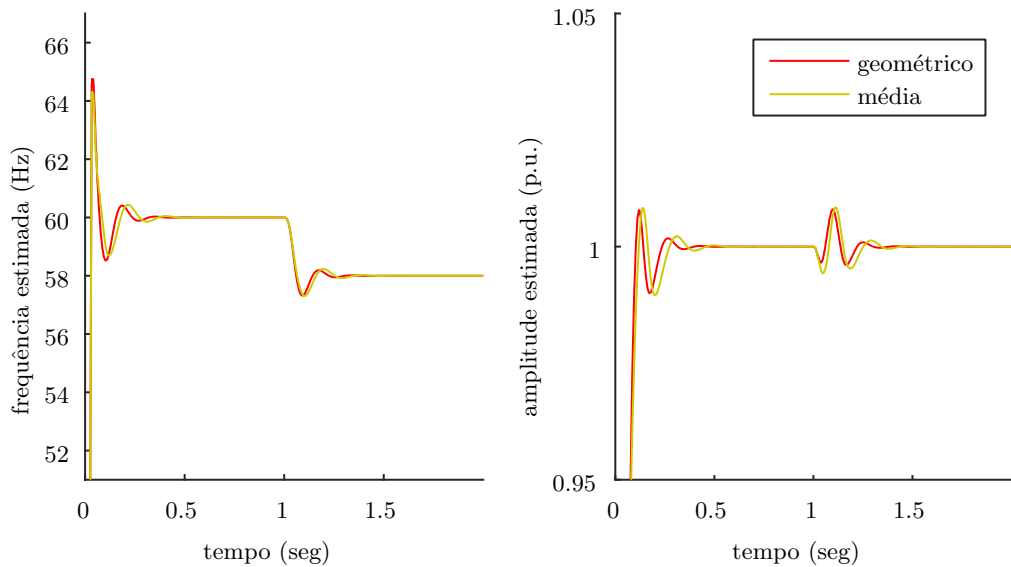


Figura 10 – Comparação entre o método de média (com 24 amostras) e o geométrico ($\lambda = 0.9$)

3.6 Simulações

Para as simulações computacionais foram seguidos determinados critérios:

- Convergência em amplitude: Foram consideradas as últimas 32 estimações. Assim que a média do erro quadrático destas fosse menor que $4e-4$ (que é um erro menor que 2% do sinal), considera-se que o algoritmo convergiu.

- O mesmo para a convergência em amplitude, mas com limiar de $1e-4$.
- Os cálculos de erro quadrático médio são feitos considerando o valor das estimações depois de 1 segundo, quando o algoritmo já convergiu.

Os parâmetros da simulação são $f_0 = 60Hz$, 128 pontos por ciclo de f_0 , $f_s = 7680 Hz$, $\alpha = 0.975$ e dois filtros em cascata no banco. A constante de subamostragem é $M_k = 16$, para todos os harmônicos e inter-harmônicos. A amplitude da componente fundamental é 180 V, e para todos os harmônicos suas amplitudes são a da fundamental dividido pela ordem do respectivo harmônico. Foi passado um filtro de média móvel de ordem 16 em todas as estimações de modo a suavizar a visualização dos resultados.

3.6.1 Harmônicos ímpares

Foram realizadas simulações com os harmônicos ímpares de 1 a 15 cada um com amplitude proporcional ao inverso de sua ordem. Observa-se na tabela 1 que na ausência de ruído, temos baixo erro e rápida convergência, principalmente para a estimação de frequência, enquanto que para amplitude, alguns harmônicos interferem nos outros devido a subamostragem.

Na tabela 2 observamos novamente erros baixos em frequência e valores pouco diferentes dos anteriores em amplitude. Como as medidas em qualquer instantes estão sujeitas a ruído, não foram considerados valores finais neste caso, mas sim o MSE.

Harmônico	Erro em frequência (%)	Erro em amplitude (%)	Tempo de convergência W (s)	tempo de convergência Amp. (s)
1	6,14E-07	1,344	0,144	0,098
3	2,08E-06	1,427	0,150	0,135
5	2,16E-06	1,338	0,035	0,088
7	3,05E-05	1,441	0,090	0,194
9	1,60E-05	0,913	0,035	0,090
11	4,93E-05	1,146	0,035	0,221
13	2,01E-04	0,136	0,035	0,090
15	4,57E-03	0,126	0,035	0,327

Tabela 1 – Tabela para a simulação sem ruído

Harmônico	MSE em frequência (%)	MSE em am- plitude (%)
1	4,21E-05	2,32E-04
3	6,05E-07	3,16E-04
5	4,23E-07	5,42E-04
7	3,70E-07	1,01E-03
9	5,56E-07	1,45E-03
11	3,26E-07	1,96E-03
13	7,37E-07	3,65E-03
15	7,76E-07	4,56E-03

Tabela 2 – Tabela para a simulação com ruído ($\sigma^2=10$)

Em resumo:

- o valor de offset para a componente fundamental é relativamente grande, e isto se explica principalmente pela subamostragem, uma vez que alguns harmônicos não são completamente atenuados, se sobrepondo à componente fundamental. Isso poderia ser melhorado com o aumento da seletividade do filtro, entretanto já foram discutidos os problemas que isto pode ocasionar. Seu uso ou não depende da natureza do problema em questão.
- o erro em frequência é em geral mais baixo que o de amplitude. E assim também é seu tempo de convergência.
- o tempo de convergência é difícil de prever, sendo que não é certo que os harmônicos de ordem mais elevada ou os de maior energia convergirão mais rapidamente que os demais.
- o algoritmo se comporta muito bem com ruído, principalmente na estimação de frequência.

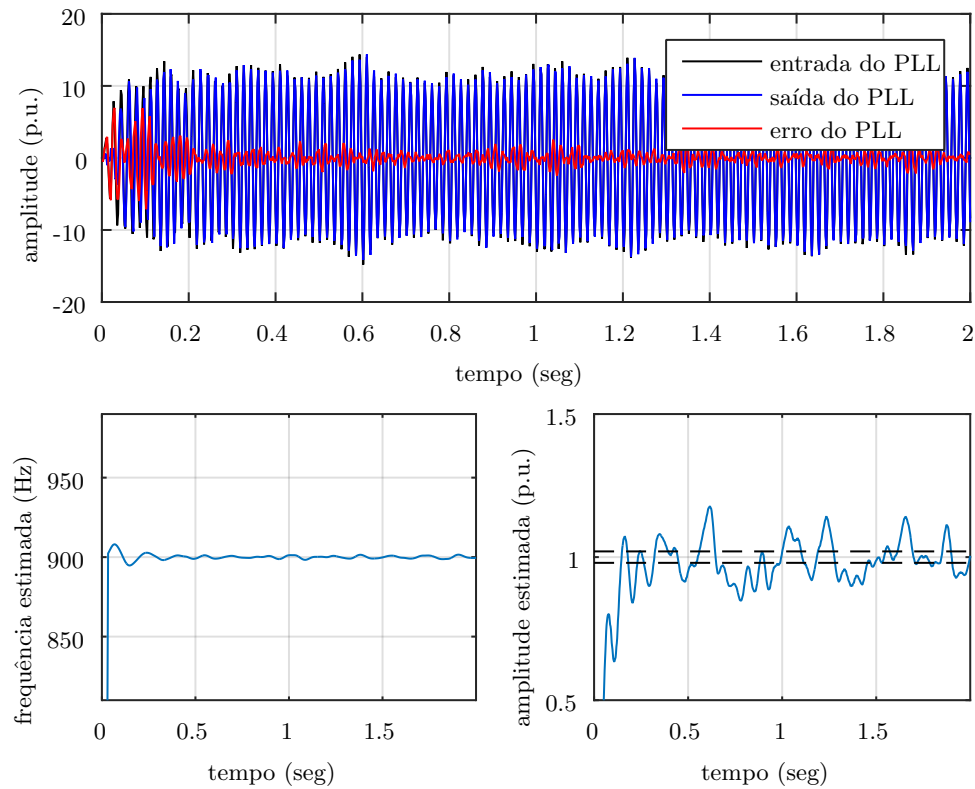


Figura 11 – Convergência do 15º harmônico na presença de ruído

É possível observar na figura 11 a convergência em amplitude e frequência da estrutura rastreado o 15º harmônico. Destaca-se na grande velocidade com que a frequência é rastreada, e a pouca variação presente em sua estimação.

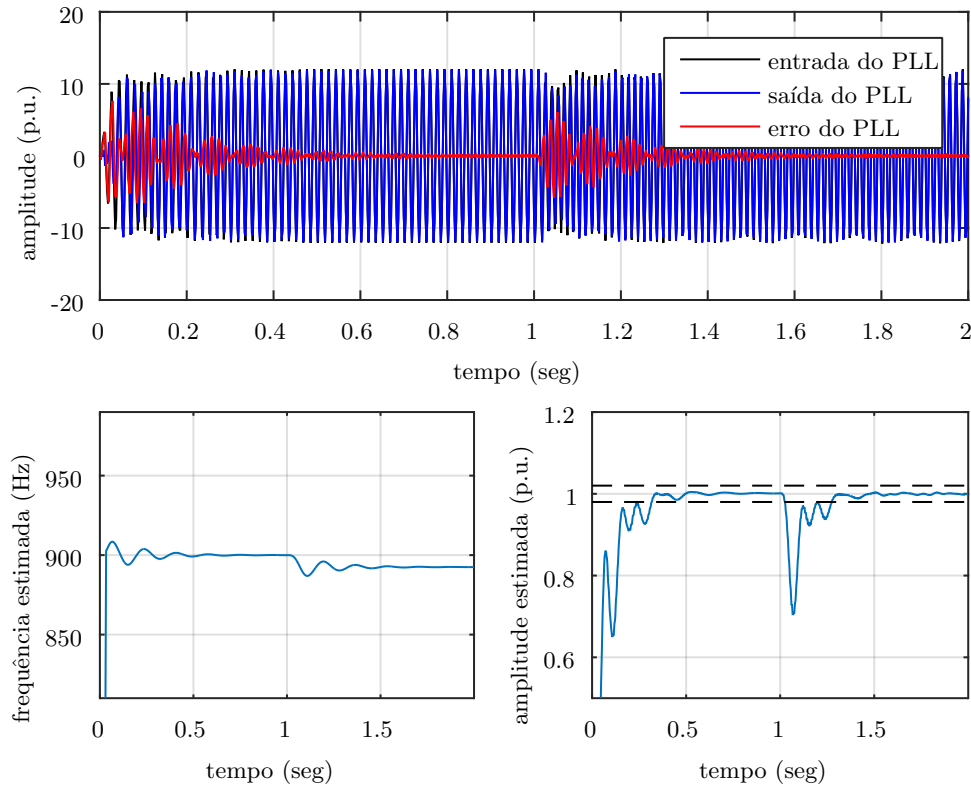


Figura 12 – Convergência do 15º harmônico com degrau em frequência de 0.5 Hz na fundamental

A figura 12 mostra o rastreamento da 15ª componente quando ocorre um degrau na frequência fundamental em 1 segundo, e mostra como o algoritmo se recupera rapidamente, voltando ao valor correto em menos de meio segundo.

3.6.2 Inter-harmônicos

Foram simuladas estimações com inter-harmônicos, seguindo as mesmas prescrições do caso anterior. Percebe-se que os tempos de convergência não são muito afetados como observamos na tabela 3, entretanto o erro quadrático médio para a componente fundamental aumenta um pouco. Isto se deve a um leve batimento que o harmônico 3.2 causa na componente fundamental, fazendo com que a estimação oscile um pouco. Contudo, se observar uma média longa o suficiente destas estimações, pode-se livrar do batimento, como observamos na figura 13.

A dinâmica da convergência do harmônico 6.4 pode ser observada na figura 14. Observa-se uma estimação até menos conturbada que a do 15º na seção anterior.

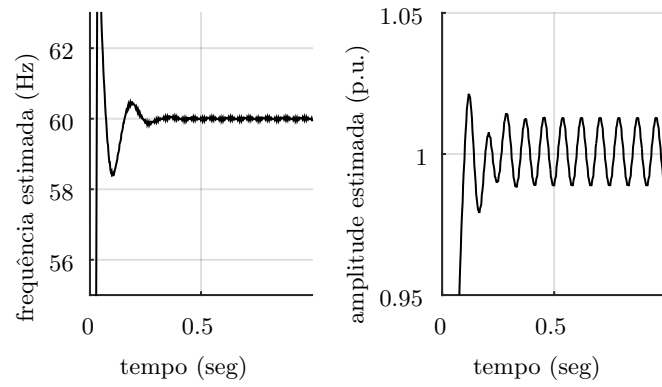


Figura 13 – Batimento observado na componente fundamental

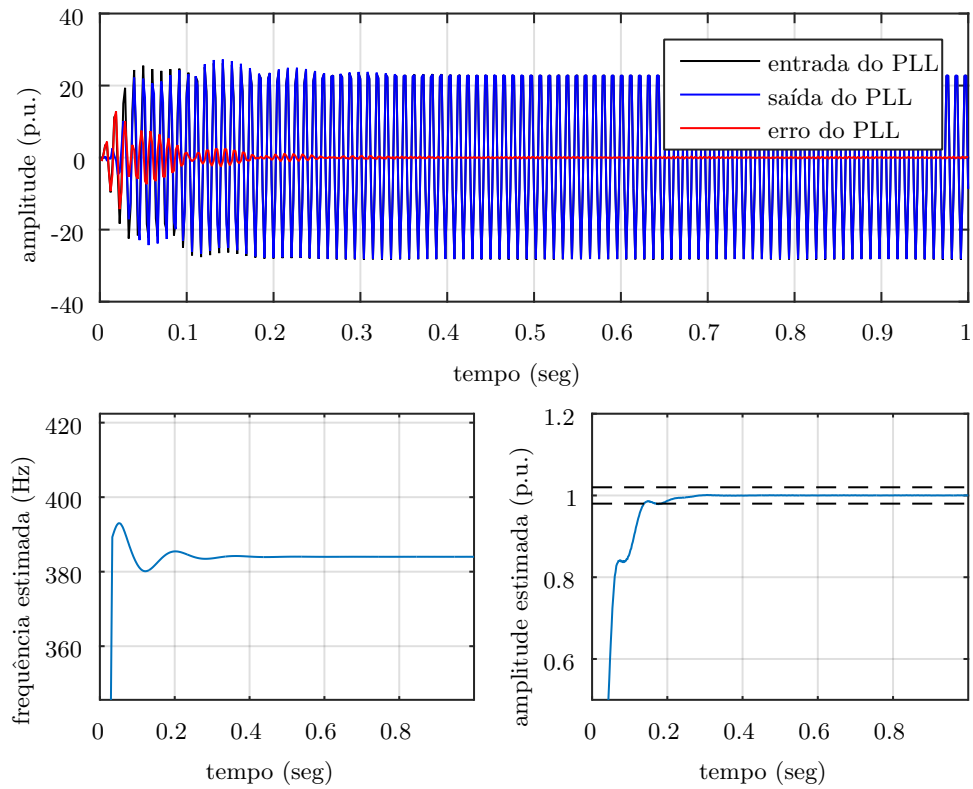


Figura 14 – Convergência do inter-harmônico 6.4

Harmônico	convergencia em freq. (s)	convergencia em amp (s)	MSE em frequência (%)	MSE em am- plitude (%)
1	0,177083	0,1375	2,91E-05	1,66E-04
3,2	0,179167	0,175	6,73E-08	1,06E-05
6,4	0,114583	0,19375	2,46E-09	3,65E-06
11,3	0,11875	0,352083	5,04E-08	3,59E-06

Tabela 3 – Tabela para a simulação de inter-harmônicos

3.7 Esforço computacional

Considerando um sinal amostrado em 7680 Hz e uma subamostragem de 16 amostras, será feita a estimativa do esforço computacional considerando operações realizadas a cada iteração de soma, multiplicação e funções trigonométricas, sem considerar deslocamentos de *buffers* e alocação de memória:

3.7.1 Banco de filtros

Cada estrutura correspondente ao filtro apresentado na equação (3.1) representa 5 multiplicações e 4 somas. Sendo dois filtros, totaliza 10 multiplicações e 8 somas por cada amostra.

3.7.2 PLL

Cada estrutura PLL, como apresentado nas equações do capítulo 2, representa 11 multiplicações, 5 somas e 4 operações trigonométricas, que são executadas a cada 16 amostras.

3.7.3 Atualização de frequência

Cada atualização de frequência custa 2 multiplicações, uma soma e uma operação trigonométrica.

3.7.4 Operações Totais

Abaixo tem-se uma tabela com todos os cálculos necessários para 1 segundo de sinal, por cada frequência rastreada. Percebe-se que mais de 90% dos cálculos são gastos pelo filtro digital.

Estrutura	Somas	Multiplicação	Trigonométricas
Banco	61440	76800	0
PLL	2400	5280	1920
Atualização	480	920	480
Total	64320	83040	2400

4 Estimação de frequências baseada em predição linear

Foi visto no Capítulo 3, que descreveu sobre o PLL-Multitaxa, que este é um método com grande eficácia no rastreo de frequências. Porém, é necessário saber onde essas frequências se encontram, e isto o método não informa. Seria possível, por exemplo, utilizá-lo como um filtro de partículas e alocar PLLs aleatoriamente pelo espectro do sinal. Entretanto, existem algumas formas de obter boas ideias de onde se encontram as componentes de frequência mais relevantes do sinal. Uma delas é a DFT ou a STFT com janelamento para reduzir o espalhamento pelo espectro. Desta forma, pode-se identificar regiões do espectro onde sabe-se que há energia relevante, sem saber no entanto a localização exata da componente, ou as componentes às quais pertence esta energia.

Outro método visto na revisão foi o de Prony [10], do qual se faz uso neste trabalho. Neste trabalho utilizou-se da otimização via RLS para resolver o problema da predição linear e calculou-se as raízes do polinômio característico do modelo Auto-Regressivo do sinal, sendo possível obter as componentes relevantes do sinal com certa precisão.

Neste mesmo trabalho, posteriormente se faz uso do RLS novamente para estimar a amplitude e a fase de cada uma das componentes, alimentando o RLS com sinais senoidais em quadratura de modo que os coeficientes dos filtros adaptativos estimados possuam informações das amplitudes e fases de senoides e cossenoides.

Na seção a seguir, será realizada a exposição de um método baseado em [10] com algumas modificações.

4.1 Solução em predição linear

O método planteado na seção de Análise de Prony visa encontrar os coeficientes w_m tais que:

$$u[k] = \sum_{m=1}^M w_m u[k - m], \quad (4.1)$$

que ao final é o equivalente a encontrar a matriz de autocorrelação e o vetor de correlação cruzada do sinal $u[k]$ na forma:

$$\mathbf{w}_{opt} = \mathbf{R}_{uu}^{-1} \mathbf{r}_{du} \quad (4.2)$$

Desta vez, o sinal de referência $d[k]$ ganha um caráter especial uma vez que ele é igual a $u[k + 1]$. Esta equação pode ser resolvida utilizando o método de Levinson–Durbin [17], para um determinado conjunto de dados. Mas também pode ser resolvida com os filtros adaptativos a cada iteração, fazendo um rastreo online destes parâmetros.

4.2 Solução em RLS e NLMS

Como visto na revisão no capítulo 2, o RLS faz uma estimação da matriz \mathbf{R}_{uu}^{-1} a cada iteração e geralmente é o método que vai propiciar o menor MSE em menor tempo. Entretanto, ele possui algumas deficiências:

- Se a matriz \mathbf{R}_{uu} é singular ou não é bem ajustada, pode haver problemas numéricos significativos, como por exemplo a ordem dos elementos da matriz ser muito desbalanceada ou muito elevada. Isso é bastante notável no problema em estudo neste trabalho, pois em geral não se sabe a ordem do sistema que se está analisando. Se este sistema possui uma ordem baixa e é estimado com um filtro de ordem elevada, a matriz \mathbf{R}_{uu} certamente será singular. Algo que pode ser feito para minimizar este problema é a adição de algum ruído;
- Em geral é mais lento para reagir a variações no sistema do que o LMS;
- Tem uma complexidade computacional consideravelmente maior;
- Sua convergência é de certa forma caótica, perturbando o cálculo das raízes.

Deve-se também destacar que tanto o RLS quanto o NLMS se beneficiam de ordens de filtro próximas à quantidade de amostras por ciclo da componente fundamental. Isto se deve ao fato de que quanto menores forem as variações dos sinais dentro do *buffer* \mathbf{U} , mais singular será a matriz de autocorrelação.

Realizando-se uma simulação com os harmônicos de ordem 1 até 7, com a amplitude igual ao inverso de sua ordem e presença de ruído gaussiano com $\sigma = 0.02$, foram comparados os resultados da divisão do menor pelo maior autovalor da matriz de autocorrelação estimada do sinal com diferentes valores de amostragem e ordem da matriz. Quanto menor o tamanho desta divisão, mais próxima de ser singular é a matriz estimada:

	M=16	M=64
$f_0 \times 16$	1.3e-03	2.1e-04
$f_0 \times 64$	2.7e-04	8.6e-05

Tabela 4 – Divisão do menor pelo maior autovalor da matriz de autocorrelação para diferentes pares de valores de frequência e ordem

A tabela 4 mostra o que já era esperado, e que pode ser confirmado com outras simulações mais adiante, de que se obtém melhores resultados com uma ordem de modelo M menor e com menores taxas de amostragem.

A figura 15 mostra como o RLS e o NLMS convergem na presença de alguns harmônicos ímpares. Destaque para a rápida convergência do RLS.

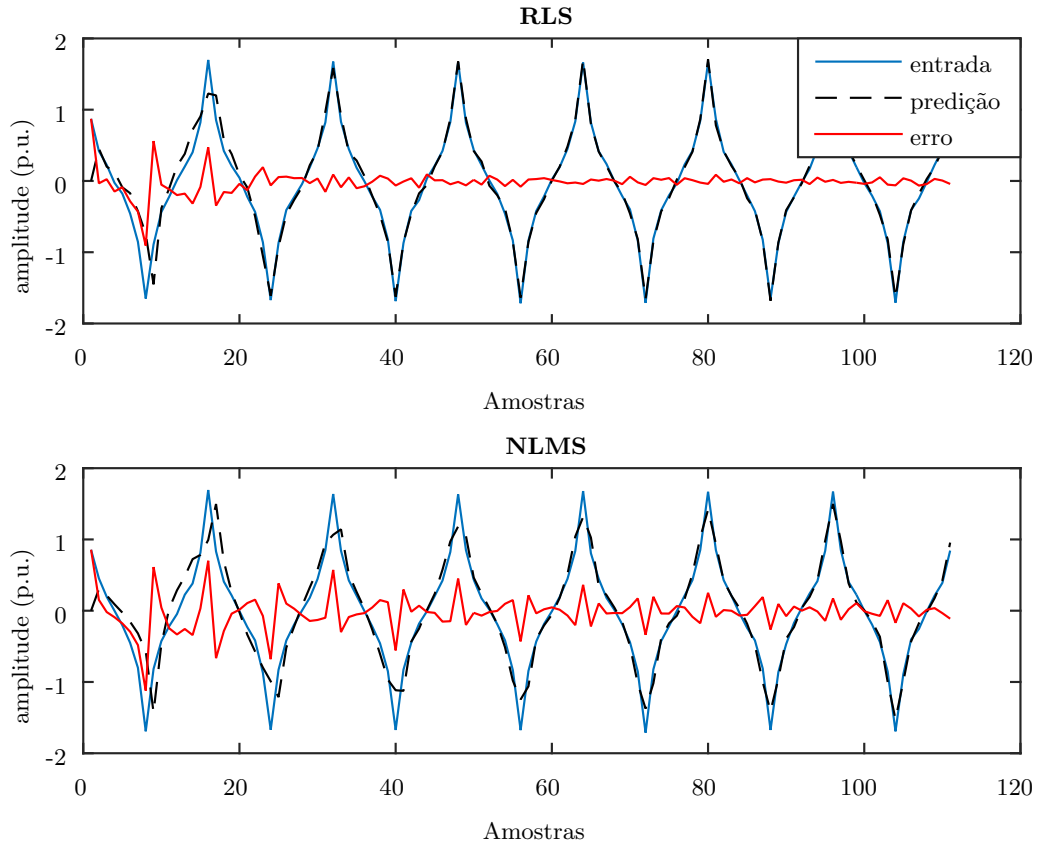


Figura 15 – Convergência do RLS e NLMS na presença dos harmônicos 1, 3, 5 e 7; $M=16$

A figura 16 mostra como convergem os coeficientes w_n com o passar do tempo para os dois algoritmos tratados. Nota-se como a estimação do NLMS é mais ordenada quando comparada com a do RLS. Ainda nesta linha de raciocínio, observando a figura 17 nota-se como a ordem na estimação dos coeficientes é refletida nas frequências.

4.3 Simulações

Foram realizadas simulações com os seguintes parâmetros: $f_s = 32f_0$, $M = 32$, harmônicos de 1 a 15, pares e ímpares e todos com amplitude unitária. É considerada uma variação aleatória nestas frequências, somando-lhes um valor aleatório de uma V.A. gaussiana com determinada amplitude σ_f (que varia na tabela). São apresentados na Tabela 5 os resultados de porcentagem de erro para o RLS e NLMS. Também foi agregado ao sinal um ruído gaussiano com $\sigma = 0.02$. Depois de 15 ciclos do sinal é tomado o resultado:

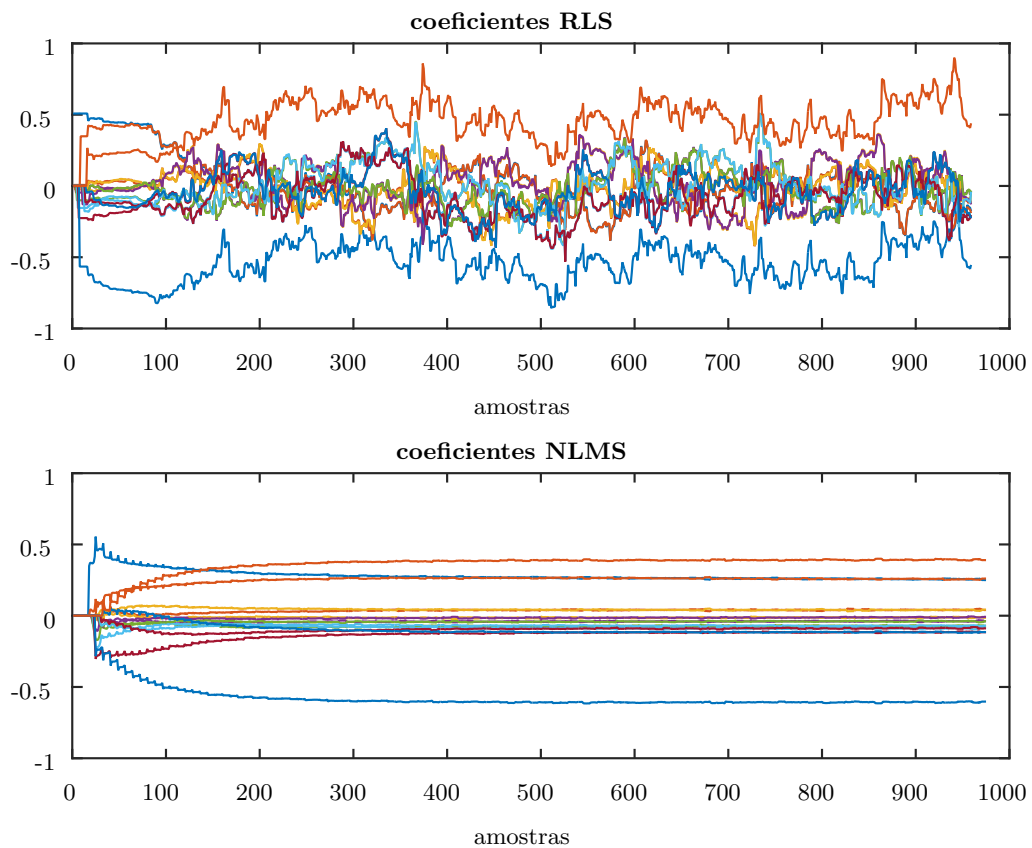


Figura 16 – Convergência dos coeficientes RLS e NLMS na presença dos harmônicos 1, 3, 5 e 7; $M=16$

	erros NLMS (%)	erros RLS (%)
$\sigma_f = 0.1$	0	0
$\sigma_f = 0.25$	1.9	2.3
$\sigma_f = 0.5$	18.4	3.4
$\sigma_f = 0.7$	22.8	6.8
$\sigma_f = 1.0$	28.3	10.2

Tabela 5 – Erros para diferentes proximidades de frequências

É considerado um erro quando não se encontra dentre os valores estimados nenhum correspondente com distância menor que 0.1 entre os valores corretos, de maneira também que uma vez pareada uma estimação com um valor real, este valor real não mais pode ser pareado com outra estimação. Em outras palavras, neste teste estamos medindo a especificidade do algoritmo, sua capacidade de identificar frequências que realmente se encontram no sinal, mas não estamos considerando os falsos positivos (sua sensibilidade). Assim pode-se ter uma boa ideia da capacidade do algoritmo de classificar frequências diferentes, mas que estão bastante próximas.

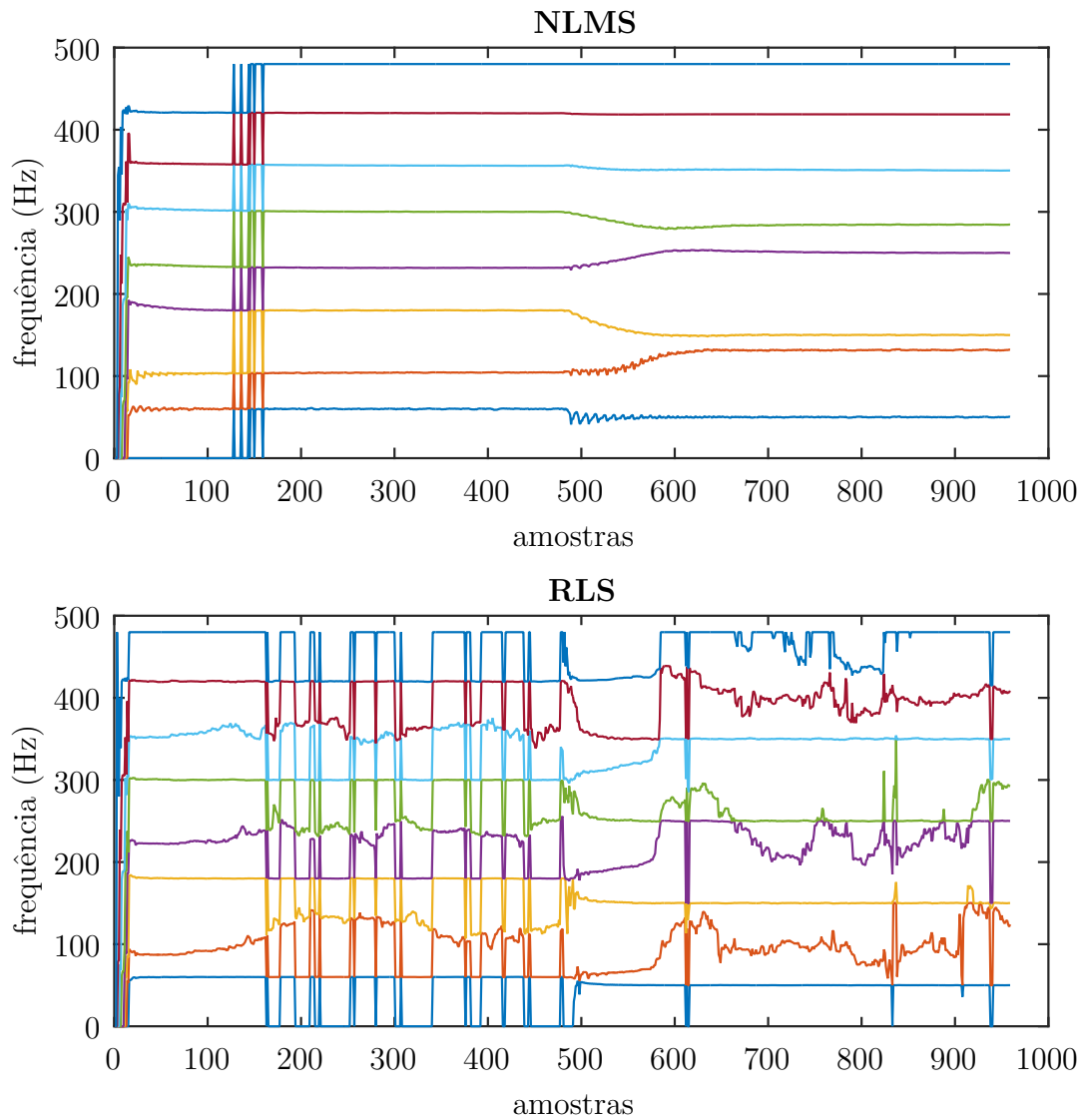


Figura 17 – Convergência do RLS e NLMS vista na estimação das frequências, com um degrau de 10 Hz em 500 amostras

Também foi realizado o mesmo teste, mas adicionando amplitude variável dos harmônicos, como uma V.A. uniforme de 0 a 1. Os resultados podem ser vistos na tabela 6. Percebe-se a dificuldade dos dois algoritmos quando as diferenças em amplitude são maiores.

	erros NLMS (%)	erros RLS (%)
$\sigma_f = 0.1$	3.4	0.6
$\sigma_f = 0.25$	14.6	2.0
$\sigma_f = 0.5$	28.4	8.5
$\sigma_f = 0.7$	31.3	12.9
$\sigma_f = 1$	34.5	16.6

Tabela 6 – Simulação com amplitude variável, resultados em %

4.4 Resultados para o caso de presença de frequências próximas

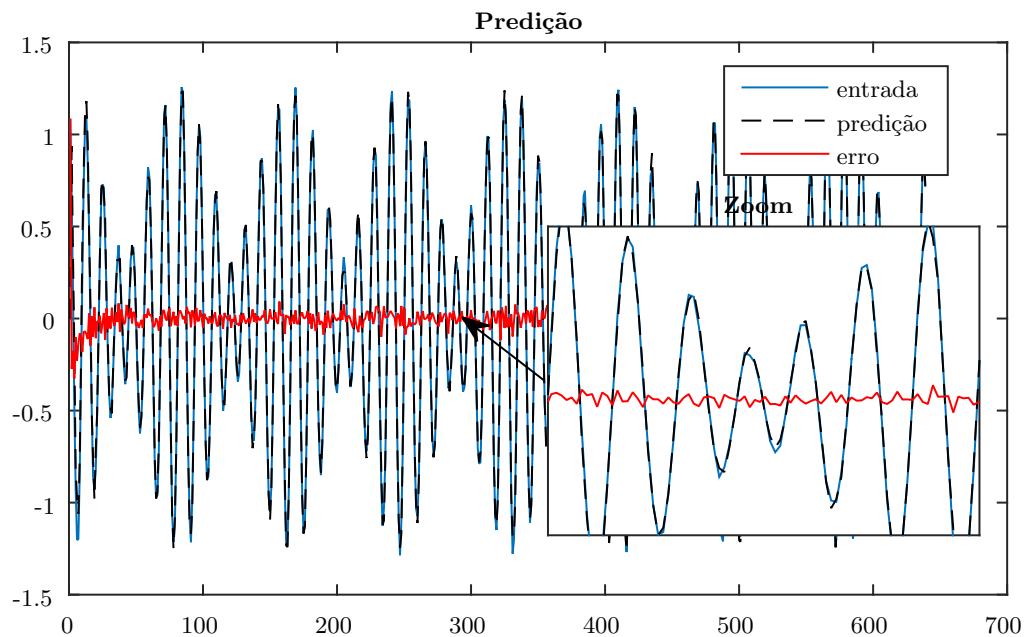


Figura 18 – Efeito de batimento em frequências muito próximas

Frequências próximas em geral são um problema e o algoritmo tem bastante dificuldade em identificá-las. Muitas vezes estas são classificadas como sendo apenas uma. Uma maneira de resolver este problema é fazendo subamostragem do sinal e dividindo-o em partes por faixa de frequência, analisando cada uma posteriormente.

O teste realizado seguiu este procedimento: foram sorteados harmônicos entre 0 e 15 com uma V.A. uniforme, desta frequência foram geradas outras duas, $f_1 = f_0 + U_{0,1}$ e $f_2 = f_0 + U_{0,1} + \delta$, sendo $U_{0,1}$ uma V.A. uniforme com amplitude 0.1. Os resultados podem ser vistos na tabela 7.

A tabela 7 mostra como é mais difícil para ambos os algoritmos o cenário onde existem duas frequências muito próximas. Quando δ é maior e as componentes estão mais

	RLS	LMS
δ	erros(%)	erros (%)
0.05	36,8	51,0
0.1	25,8	50,3
0.2	7,3	43,5
0.4	1,5	25,8
0.6	0,5	14,8

Tabela 7 – Tabela com os erros para classificação de frequências próximas

espaçadas, os erros são muito menores. A figura 18 exemplifica o problema relacionado, mostrando como duas frequências diferentes se confundem em um sinal.

4.5 Melhora da sensibilidade

Recuperando algumas equações do capítulo de revisão bibliográfica, foi apresentado que o modelo estocástico do sinal Auto-Regressivo era da forma:

$$u[k] = \sum_{m=1}^M w_m u[k-m] + \xi[n], \quad (4.3)$$

sendo ξ ruído branco. Desta maneira pode-se desenvolver a equação para obter a transformada Z de $u[n]$:

$$U(z) = \sum_{m=1}^M w_m U(z) z^{-m} + \Xi(z) \quad (4.4)$$

$$\frac{U(z)}{\Xi(z)} = \frac{1}{1 - \sum_{m=1}^M w_m z^{-m}} \quad (4.5)$$

A equação (4.5) mostra que a porção $Q(z) = 1 - \sum_{m=1}^M w_m z^{-m}$ determina as características espectrais de $U(w)$, uma vez que ξ é ruído branco e tem densidade de potência constante para todo o espectro. Calculando $Q(w)^{-1}$, pode-se obter uma estimativa da energia naquele ponto. Assim, se for calculado $Q(w_{root})$ para todas as raízes, é possível observar quais têm maior energia e quais têm menor. Desta forma, está permitido até mesmo descartar a análise mais profunda de certas componentes que serão acusadas na estimação de frequências. Este tipo de análise inclusive é usado em processamento de fala para identificar formantes de fonemas, e por si só já um método de estimação espectral paramétrico.

Pode-se observar nas Figuras 19 e 20 que a estimação feita desta forma para o RLS acusa muito mais frequências que não são reais e de maneira aparentemente aleatória no decorrer do tempo; entretanto, ele converge muito mais rápido que o NLMS, que apresenta

convergência suave e com pouco ruído. Em ambos os casos, tem-se a presença do quinto e terceiro harmônicos, que sofrem uma mudança brusca na frequência em $t=2s$.

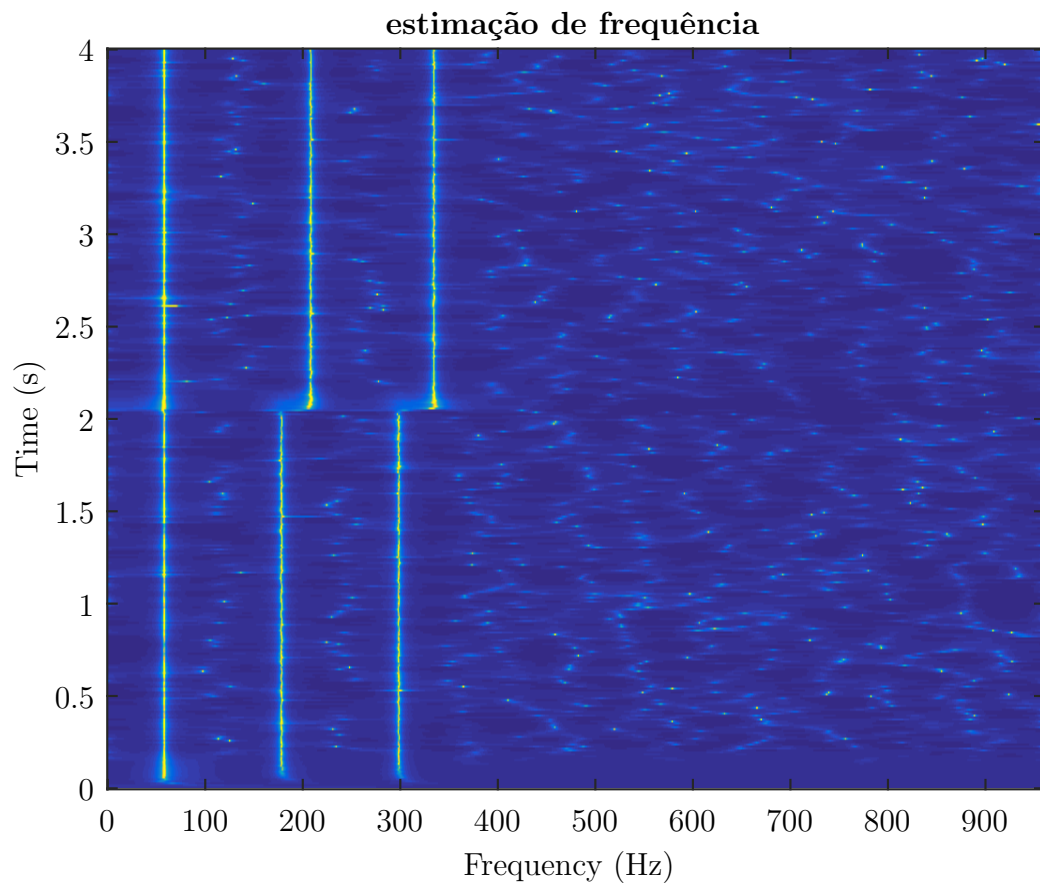


Figura 19 – Imagem com estimação de amplitudes e frequências no tempo usando RLS

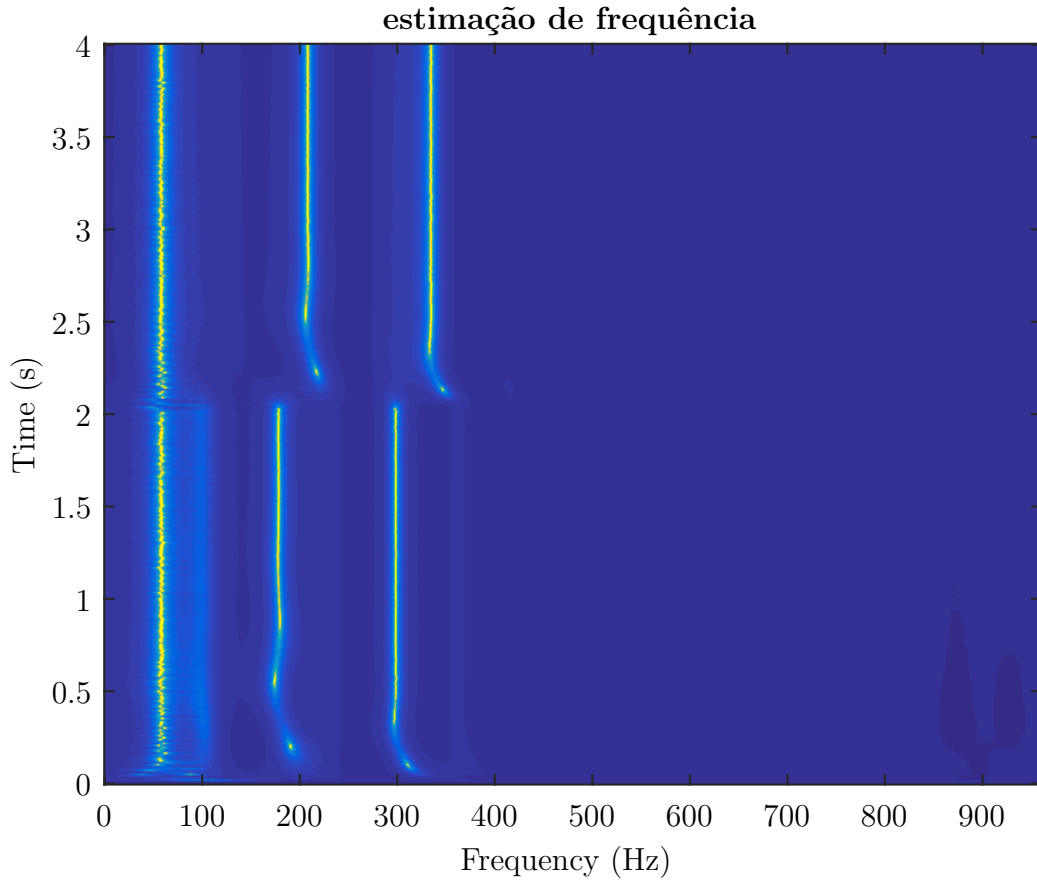


Figura 20 – Imagem com estimação de amplitudes e frequências no tempo usando NLMS

4.6 Teste de rastreo de frequências

Será realizado agora o mesmo teste de rastreo de frequências realizado no artigo [10]. Utiliza-se $M = 64$, e $f_s = 64f_0$. Este teste consiste em um sinal com cinco componentes senoidais, sendo três harmônicos da fundamental e dois inter-harmônicos, modelado de acordo com a seguinte equação:

$$y(t) = A_1 \sin(w_0 t + 30^\circ) + A_2 \sin(3w_0 t + 70^\circ) + A_3 \sin(5w_0 t + 40^\circ) + 0.15 \sin(w_1 t + 135^\circ) + 0.1 \sin(w_2 t + 175^\circ) + \xi(t), \quad (4.6)$$

onde $A_1 = 1$, $A_2 = 1/3$, $A_3 = 1/5$, $w_0 = 2\pi f_0$ e $f_0 = 60 \text{ Hz}$. Para os inter-harmônicos, $f_1 = 152 \text{ Hz}$ e $f_3 = 266 \text{ Hz}$. Além disso, $\xi(t) = \text{rand}(t)$.

- Em $t = 1.0 \text{ s}$ as amplitudes dos harmônicos são alteradas na forma de degrau para 0.8, 0.2 e 0.3, respectivamente;
- Em $t = 2.5 \text{ s}$, f_0 é alterada na forma de degrau para 59.8 Hz;

- em $t = 4.0$ s, f_0 é alterada para 60.3 Hz na forma de degrau, e os harmônicos voltam a 1.0, 1/3 e 1/5, respectivamente, também na forma de degrau;
- A simulação segue até $t = 5.5$ s.

Os resultados de rastreo de frequências podem ser vistos na figura 21 e em detalhe para a fundamental na figura 22:

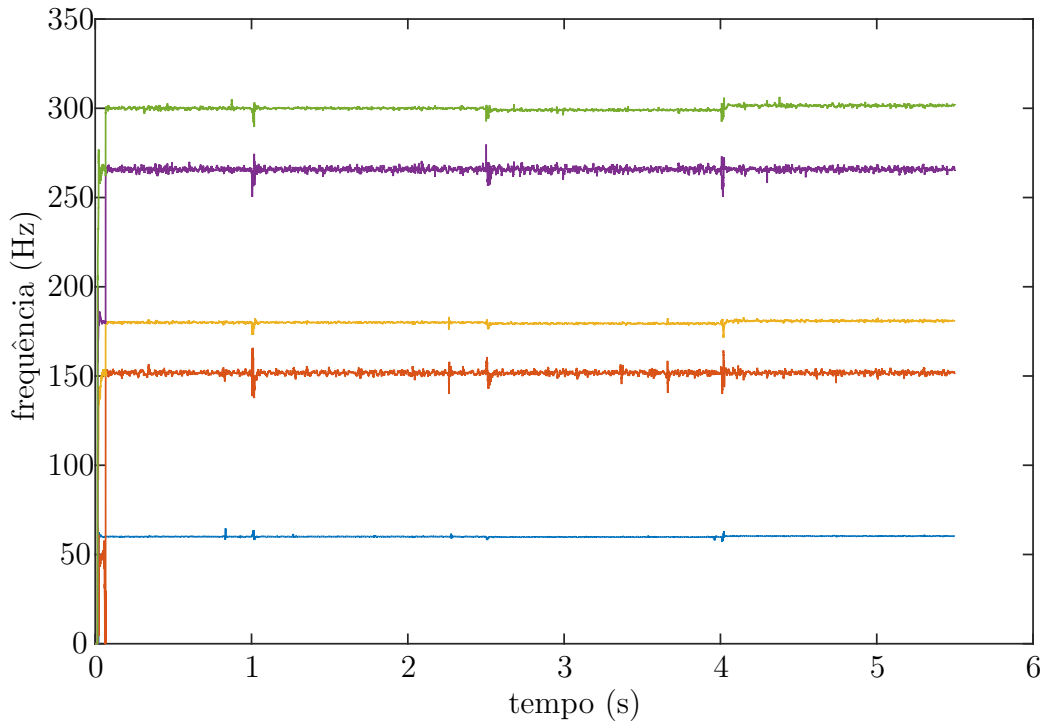


Figura 21 – Rastreo com teste do artigo [10] utilizando RLS

Para o cálculo das raízes foi utilizada a função *roots* do MATLAB. Esta função não permite usar valores anteriores como pontos iniciais de busca pelas novas raízes, e no caso do RLS já foi visto que este possui uma convergência bastante turbulenta, por assim dizer. Isso tudo faz com que muitas vezes uma mesma raiz seja calculada em ordens diferentes. Assim, é necessário plantear um pequeno classificador muito simples para que se possa fazer o rastreo.

Foi utilizado um classificador por distância, no qual utiliza-se os valores antigos presentes no mesmo para determinar qual das novas amostras pertence à posição 'n' do classificador. A cada iteração, analisa-se as novas estimações $\hat{f}_i[k]$ (frequência i da iteração k) e as coloca como uma nova evidência para a frequência mais próxima em valor absoluto no vetor de estimação global f_{global_n} (que é a estimação final do método). E para reduzir

o ruído de estimação, utiliza-se uma soma convexa das estimativas antigas com a mais recente, da mesma maneira feita com o PLL. O método está descrito na equação (4.7).

$$f_{global_n} = f_{global_{n-1}}(1 - \lambda) - [\operatorname{argmin}_{\hat{f}_i[k]} (|f_{global_{n-1}} - f_i[k]|)]\lambda \quad (4.7)$$

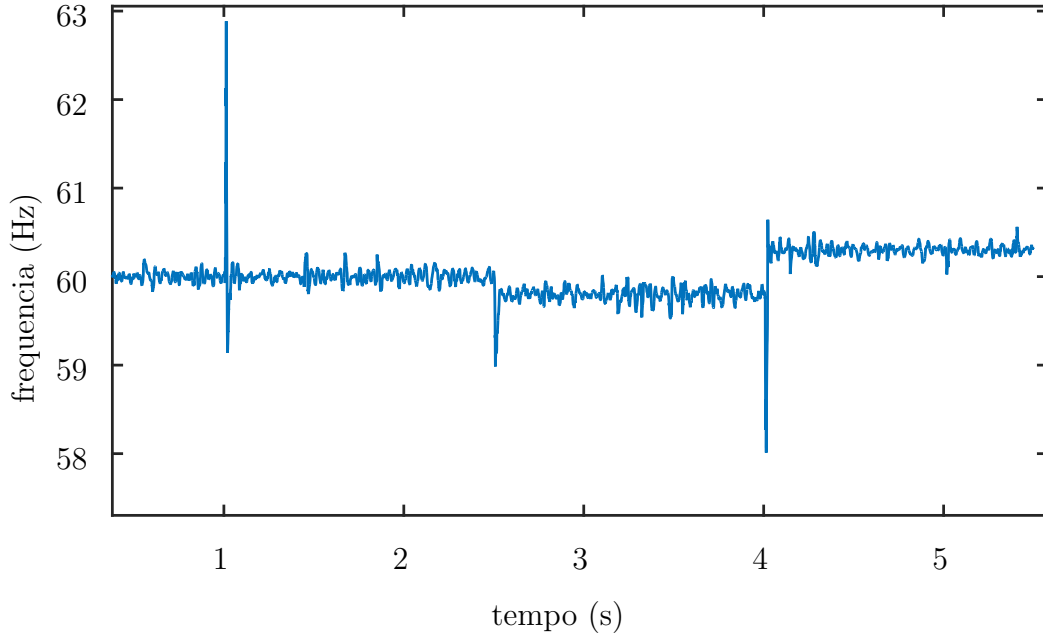


Figura 22 – Rastreamento com teste do artigo utilizando RLS, componente fundamental

4.7 Complexidade computacional

A complexidade computacional do RLS é de cerca de $3M^2 + 4M$ somas e multiplicações por iteração [10], enquanto que a do NLMS é de cerca de $4M$ somas e multiplicações. Apesar de a ordem de complexidade do NLMS ser bem menor, entre as operações do RLS há uma série de multiplicações e somas de matrizes, que podem ser otimizadas com computação em paralelo para determinados dispositivos. Então, é de se rever onde vai ser aplicado o algoritmo, e dependendo do *hardware* utilizado, o custo do RLS pode não ser tão mais alto que o do NLMS.

4.8 Conclusões

O algoritmo apresentado tem grande capacidade de estimação para determinados contextos. Por exemplo, com os harmônicos relativamente separados e com amplitude parecida, praticamente não há erros na estimação tanto no NLMS quanto no RLS. Claro

que este contexto não é o que se vai encontrar muitas das vezes, mas pode-se fazer ajustes de taxa de amostragem e eliminação de algumas componentes caso seja necessário, tudo depende da natureza do problema. Em situações mais críticas, com grande desnível dos harmônicos e proximidade dos mesmos, o RLS acaba levando grande vantagem. Escolher um método ou o outro depende do hardware disponível e também da proposta desenvolvida. Foi mostrado ainda que o NLMS faz uma estimação mais estável dos coeficientes e conseqüentemente das raízes e harmônicos, a qual pode ser aproveitado para uma implementação online.

5 Estrutura híbrida composta por predição linear e PLL-Multitaxa

Agora utilizando todos os métodos estudados até aqui, será desenvolvida uma estrutura capaz de estimar as frequências presentes no sinal e suas relevâncias, e em seguida rastrear estas componentes por tempo indeterminado. Para tanto, se fará uso de uma estrutura híbrida composta pela predição linear do capítulo anterior seguida pela estrutura PLL-Multitaxa.

5.1 PLL-M em partículas

Foi realizada, a cargo de comparação, uma simulação com sinal nas mesmas condições da presente no capítulo anterior. Para tentar extrair os parâmetros das senoides, se inicializa PLLs em diferentes frequências, correspondentes aos harmônicos de 1 a 10. Desta maneira obtém-se o resultado visto na figura 23.

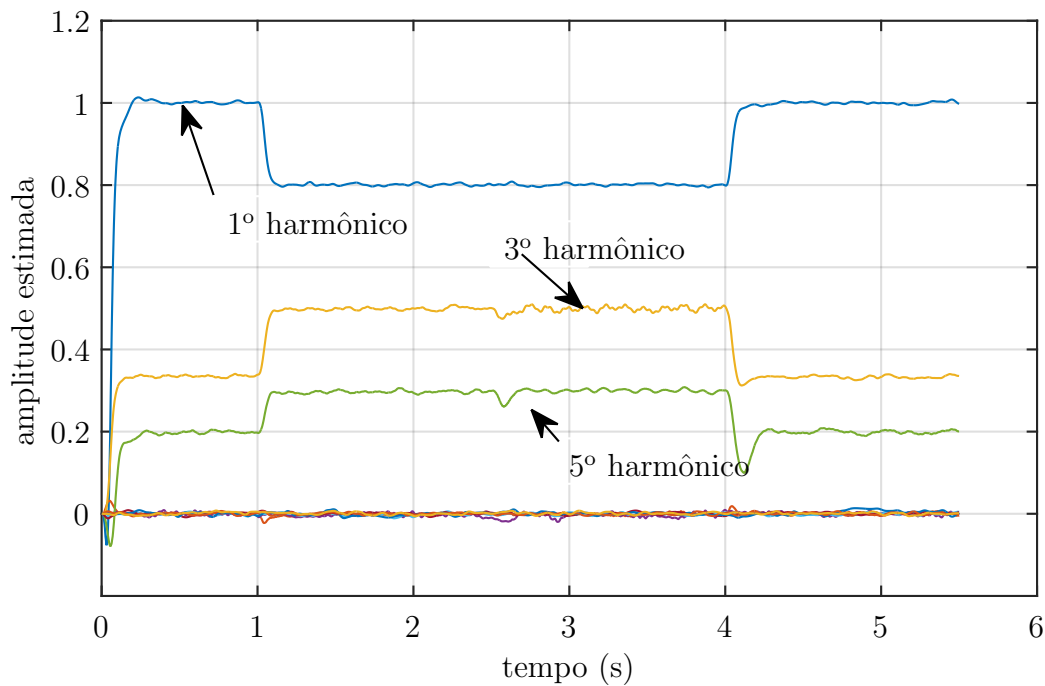


Figura 23 – Estimação de amplitudes com PLL-M

Nesta figura, observa-se que os harmônicos 1, 3 e 5 são corretamente rastreados, entretanto esta abordagem não é capaz de encontrar os inter-harmônicos em 152 Hz e 266 Hz. Também se percebe que ela é extremamente ineficaz uma vez que foram usados dez PLLs para rastrear cinco frequências e ainda assim não é possível encontrar duas delas.

5.2 Estimação de Frequências

Para solucionar o problema encontrado na seção anterior, será utilizado um método de estimação de frequências, assim será possível saber onde inicializar os PLLs, aproveitando recursos computacionais. O RLS será usado para calcular os coeficientes w_m de predição linear, onde será feita uma varredura da porção inicial do sinal aproveitando a grande velocidade de convergência. Para lidar com a desordem da estimação causada pelo RLS, será utilizado o classificador do capítulo anterior, com uma adição: vamos filtrá-las de acordo com a variância das últimas N amostras. Foi observado que as frequências corretas variam muito pouco dentro do classificador, diferente dos falsos positivos. Nos testes observamos diferença de pelo menos uma ordem de magnitude entre a variância de frequências corretas e falsas.

Passos para estimar as frequências:

1. Usar o RLS para estimar \mathbf{w} .
2. Calcular as raízes de $Q(z) = 1 - \sum_{m=1}^M w_m z^{-m}$
3. Calcular as frequências usando a equação (2.24);
4. Usar o classificador e os estimadores já mencionados para encontrar as frequências, calcular a variância destas frequências e filtrá-las, ficando com os menores valores.
5. Utilizar as frequências encontradas para inicializar o PLL.

Podemos ver o esquema completo do na figura 24:

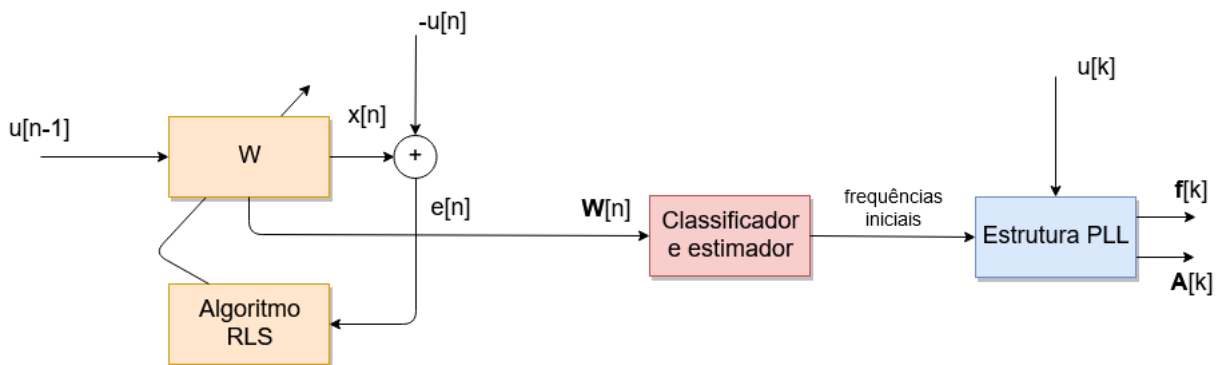


Figura 24 – Esquema híbrido

5.2.1 PLL-Multitaxa

A estrutura usada será idêntica à vista no final do capítulo 3. Tendo como única alteração as constantes, que serão $\mu = \{100, 5000, 400\}$.

5.2.2 Simulação e análise qualitativa

Será realizada a mesma simulação do artigo [10], entretanto com algumas alterações para forçar mais a estabilidade do algoritmo, em $t = 2.5\text{ s}$ se aplica um degrau de 1 Hz na frequência fundamental, ao invés de 0.2 Hz, e o ruído presente será de variância $\sigma = 0.05$ ao invés de 0.01. Tem-se $M=64$, e $fs = 128f_0$, sendo $f_0 = 60\text{ Hz}$. Entretanto, a frequência de amostragem será abaixada em 2 vezes para a estimação das frequências para diminuir o tempo de computação, uma vez que neste problema em geral não se investiga frequências acima do 15º harmônico. Se usará fator de esquecimento igual a 0.92 para o RLS. Também se rastrearão as componentes julgadas mais energéticas pelo estimador. Para o classificador será utilizado $\lambda = 0.1$ na classificação de frequências e umbral da variância igual a 5, ou seja, somente serão rastreadas as componentes cuja variância das últimas 100 amostras for menor que 5.

Se executa o RLS por meio segundo de sinal, onde devemos admitir que este mantém suas características estatísticas.

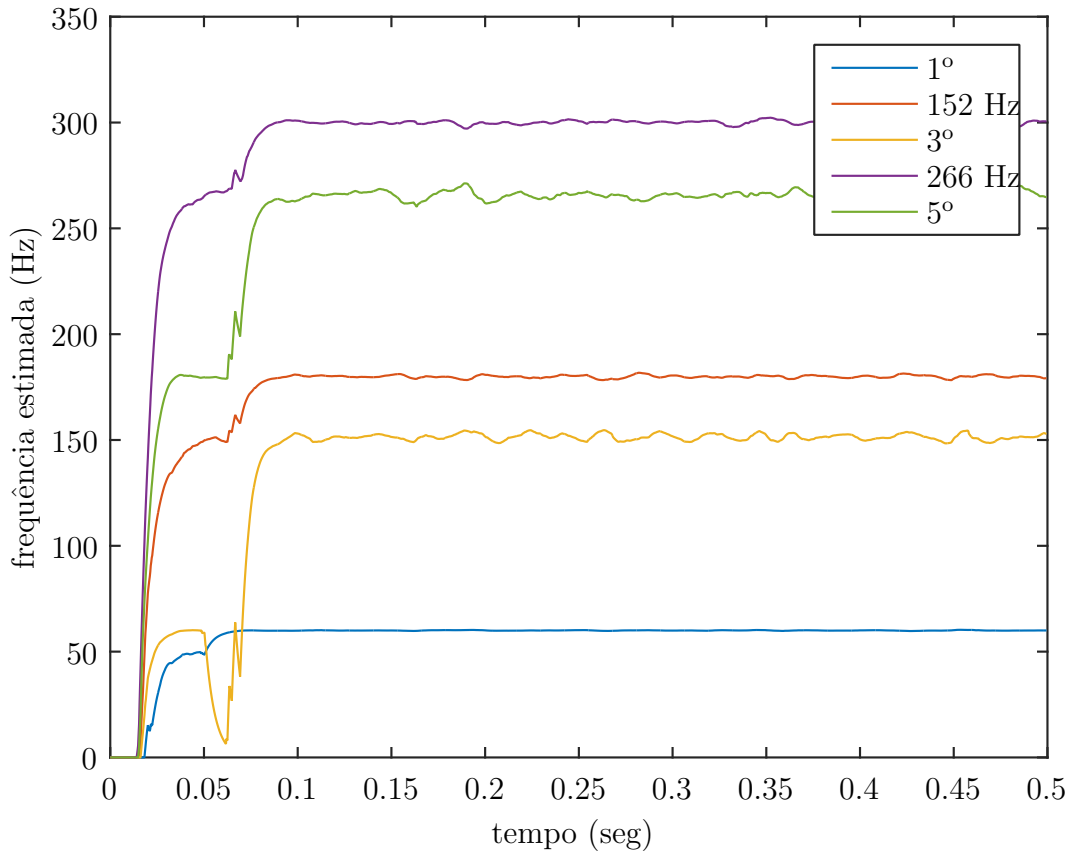


Figura 25 – Estágio de estimação de frequências e ordem do sistema

Observa-se na figura 25 o estágio de estimação de frequências e ordem do sistema,

identificando as cinco componentes desejadas. Observa-se também que ele converge para estas cinco componentes em aproximadamente 100 ms. A partir deste momento já temos os valores corretos estabilizados e adequados para passarmos ao próximo estágio, de rastreo de frequência e amplitude.

No rastreo de amplitudes da figura 26, e em detalhe na figura 28, vemos que o PLL segue as componentes de maneira rápida, eficaz e estável, apesar de mudanças na frequência fundamental, na amplitude dos harmônicos e quando os dois ocorrem ao mesmo tempo. Entretanto observa-se um batimento nas estimções da frequência errante de 152 Hz e também no 5º e 3º harmônicos depois do degrau de 1 Hz em 2.5 s. Isto se deve principalmente à interferências, uma vez que eles acabam ficando próximos, mas não leva a offset na estimação.

Percebe-se que a estimação fina de frequências, vista na figura 27, do PLL-M que é muito mais precisa e estável que a do primeiro estágio, embora esta não seja global. Se por exemplo, uma destas componentes deixa de existir, some do sinal, o PLL não tem condições de acusar isto, ou se uma nova aparece no lugar da antiga, o PLL também não poderia identificá-la.

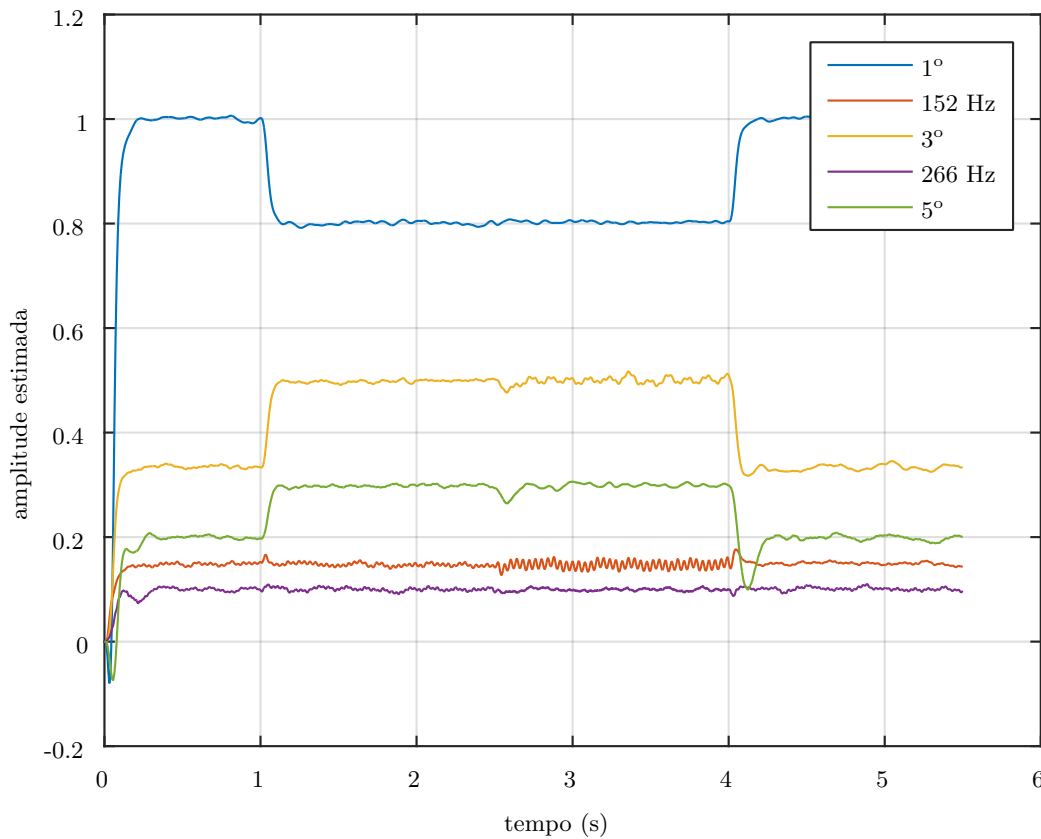


Figura 26 – Rastreo de amplitudes do PLL-M

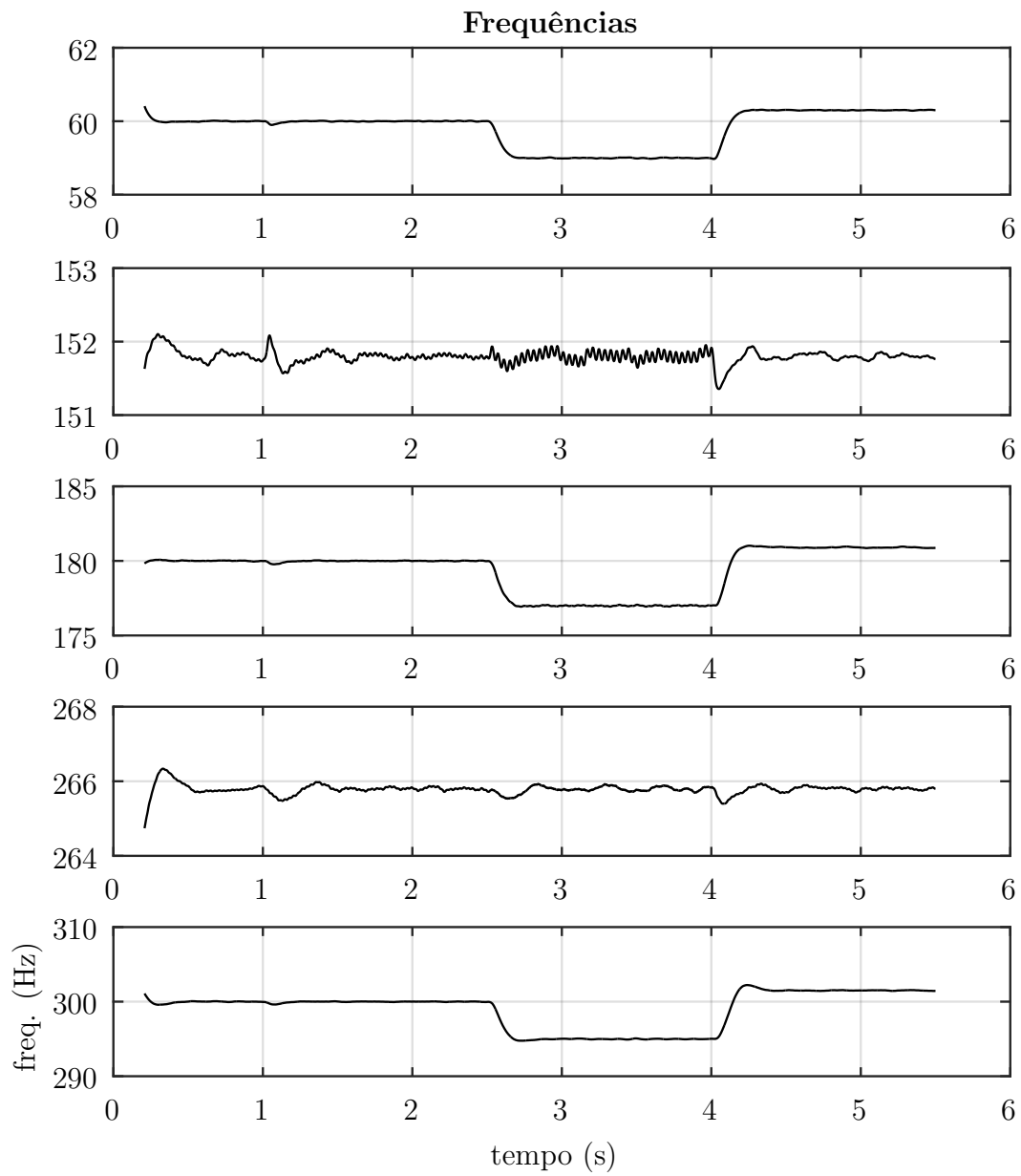


Figura 27 – Rastreo de frequências do PLL-M

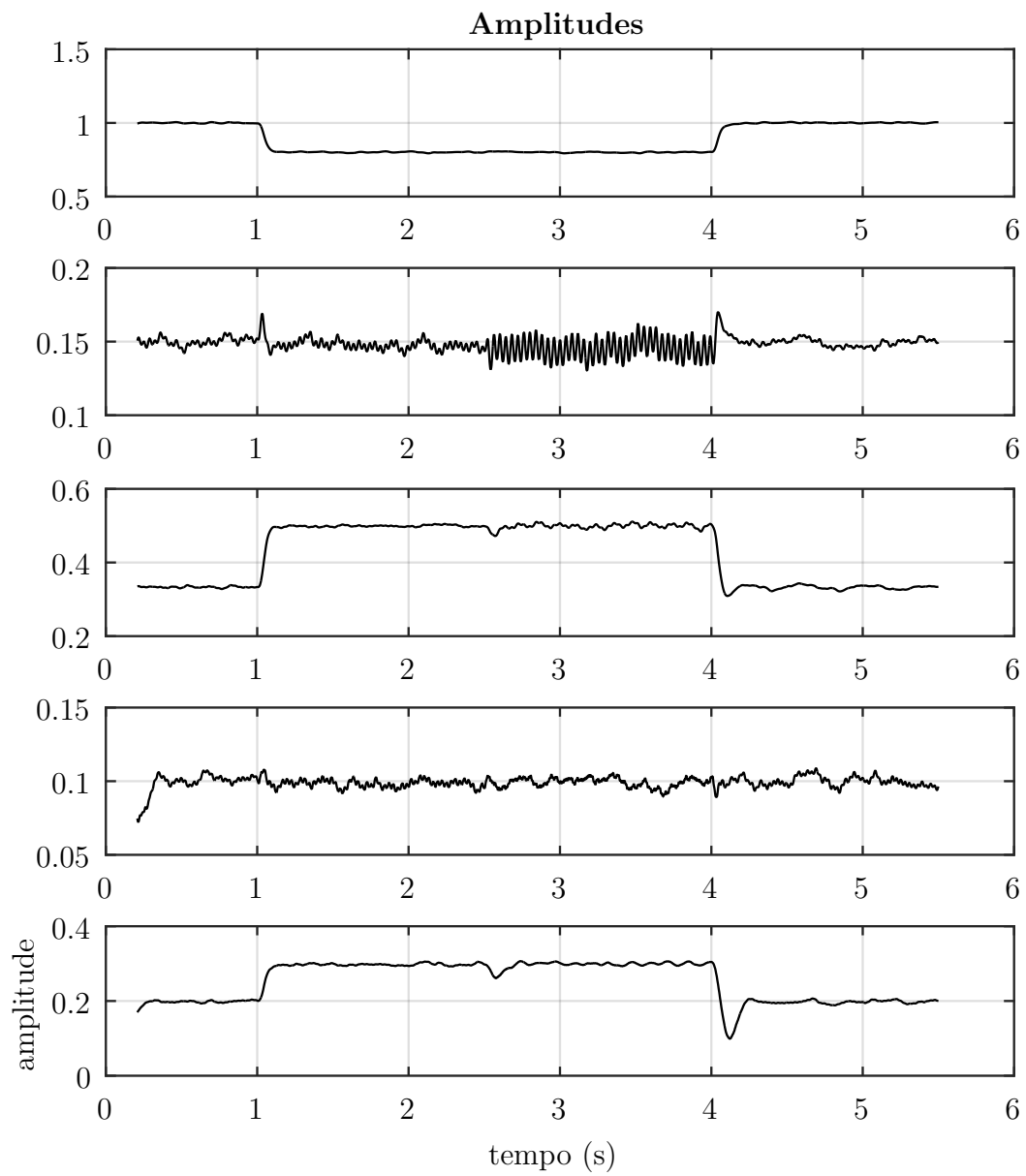


Figura 28 – Rastreo de amplitudes do PLL-M

6 Conclusões e Trabalhos Futuros

6.1 Conclusões

Neste trabalho foi apresentado o que é e o problema que representa hoje a estimação espectral, bem como a fundamentação teórica matemática por trás do tema. Discorreu-se sobre métodos paramétricos e não paramétricos. E Focou-se principalmente em simular e apresentar alternativas para duas metodologias relativamente recentes, publicadas nos últimos 10 anos.

Foram discutidas virtudes e pontos fracos de cada um dos métodos, ao final do trabalho é proposto um novo procedimento capaz de unir os pontos fortes de cada um. Aproveitando a robustez e simplicidade computacional no rastreamento do PLL-M e o poder de identificação global dos métodos baseados em modelos AR. Entretanto, mesmo dentro da metodologia aplicada ao capítulo final, há muito o que se desenvolver. Um método mais robusto seria capaz de identificar o surgimento de novas componentes e o desaparecimento de outras, fazendo uma administração inteligente dos recursos computacionais.

Mesmo a estimação baseada em modelos Auto Regressivo tem fortes limitações, como já foi discutido. Se consideramos um sinal determinístico que está altamente poluído por ruído, não é possível fazer uma boa estimação com este modelo.

De um modo geral se pode listar as boas condições de trabalho do método proposto:

- Soma de sinais senoidais finitos menores que M ;
- Relação sinal ruído maior que 20 dB;
- A ordem do modelo em si não é um agravante, mas a forma como estão distribuídas as senoides, sim pode ser. Os testes mostram que temos pouquíssimos erros para frequências separadas em no mínimo 0.04 radianos por segundo.

Por Parte do PLL temos as seguintes condições:

- O algoritmo é capaz de rastrear componentes mesmo em relações sinal ruído de 0 dB;
- Degraus em frequência por volta de 5 Hz em qualquer componente;
- Sinais dentro de certas amplitudes as quais as constantes setadas são capazes de seguir.

6.2 Trabalhos futuros

Para futuras melhorias estão:

- Determinar quando o primeiro estágio convergiu e já estimou corretamente as frequências e ordem do modelo. O que não é trivial, uma vez que pode ser difícil saber quando o algoritmo atingiu o erro mínimo. O RLS por exemplo é um método de mínima variância, capaz de convergir para o menor erro possível, que seria o de ruído. Entretanto como não se sabe o ruído, isto não é tão simples. E tampouco é trivial determinar as frequências corretas e as falsas;
- Estudar métodos paramétricos baseados em sinais determinísticos poluídos por ruído. Assim não teremos as limitações de uma aproximação por modelos AR ou ARMA;
- Encontrar um método sistemático de obtenção das constantes do PLL e outras formas de analisar sua convergência e dinâmica. Uma vez que é um método altamente não linear e complexo de se analisar. Nos trabalhos analisados foram usadas diversas soluções heurísticas que carecem de mais fundamento matemático, como a junção do PLL com um filtro IIR.

Bibliografia

- [1] Petre Stoica, Randolph L Moses et al. “Spectral analysis of signals”. Em: (2005).
- [2] Francis Castanié. *Digital spectral analysis: parametric, non-parametric and advanced methods*. John Wiley & Sons, 2013.
- [3] Sanjit Kumar Mitra e Yonghong Kuo. *Digital signal processing: a computer-based approach*. Vol. 2. McGraw-Hill New York, 2006.
- [4] TL Lago. “Digital sampling according to Nyquist and Shannon”. Em: *Sound and Vibration* 36.2 (2002), pp. 20–22.
- [5] Gregory A Baxes. *Digital image processing: principles and applications*. Wiley New York, 1994.
- [6] Xuedong Huang et al. *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice hall PTR, 2001.
- [7] Leif Sörnmo e Pablo Laguna. *Bioelectrical signal processing in cardiac and neurological applications*. Vol. 8. Academic Press, 2005.
- [8] Roger C Dugan, Mark F McGranaghan e H Wayne Beaty. “Electrical power systems quality”. Em: *New York, NY: McGraw-Hill,— c1996* (1996).
- [9] Janison Rodrigues de Carvalho et al. “A PLL-based multirate structure for time-varying power systems harmonic/interharmonic estimation”. Em: *IEEE Transactions on Power Delivery* 24.4 (2009), pp. 1789–1800.
- [10] Gary W Chang, Cheng-I Chen e Quan-Wei Liang. “A two-stage ADALINE for harmonics and interharmonics measurement”. Em: *IEEE Transactions on Industrial Electronics* 56.6 (2009), pp. 2220–2228.
- [11] Alireza K Ziarani e Adalbert Konrad. “A method of extraction of nonstationary sinusoids”. Em: *Signal Processing* 84.8 (2004), pp. 1323–1346.
- [12] Simon S Haykin. *Adaptive filter theory*. Pearson Education India, 2005.
- [13] Paulo SR Diniz. *Adaptive filtering*. Springer, 1997.
- [14] Lupércio F Bessegato et al. “On-Line Process Control using Attributes with Misclassification Errors: An Economical Design for Short-Run Production”. Em: *Communications in Statistics-Theory and Methods* 41.10 (2012), pp. 1813–1832.
- [15] Saleh R Al-Araji, Zahir M Hussain e Mahmoud A Al-Qutayri. *Digital Phase Lock Loops*. Springer, 2006.
- [16] Janison Rodrigues de Carvalho et al. “Estimação de harmônicos/interharmônicos: uma abordagem multitaxa”. Em: (2008).
- [17] Bruce Ronald Musicus. “Levinson and fast Choleski algorithms for Toeplitz and almost Toeplitz matrices”. Em: (1988).