

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

Tu Nombre

Nome do TCC

Juiz de Fora
2018

Tu Nombre

Nome do TCC

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 Lima

Cooorientadora: Nome do coorientador

Juiz de Fora

2018

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

Gabriel Eduardo, de Lima Machado.

Nome do TCC / Tu Nombre. – 2018.

?? f.

Orientador: Marcelo Lima

Coorientadora: Nome do coorientador

Trabalho de Conclusão de Curso – Universidade Federal de Juiz de Fora,
Faculdade de Engenharia. Engenharia Elétrica – Habilitação em Sistemas
Eletrônicos, 2018.

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

Tu Nombre

Nome do TCC

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. Dr. Marcelo Lima - Orientador
Universidade Federal de Juiz de Fora

Professor Dr. ???
Universidade Federal de Juiz de Fora

Professor Dr. ???
Universidade Federal de Juiz de Fora

AGRADECIMENTOS

De acordo com a Associação Brasileira de Normas Técnicas - 14724 (2011, p. 1) Agradecimentos é o “texto em que o autor faz agradecimentos dirigidos àqueles que contribuíram de maneira relevante à elaboração do trabalho.”

“Um orientador em paz não quer guerra com ninguém”
(Exuperry Barros Costa)

RESUMO

De acordo com a Associação Brasileira de Normas Técnicas - 6028 (2003, p. 2) “o resumo deve ressaltar o objetivo, método e as conclusões do documento (...) Deve ser composto de uma sequência de frases concisas, afirmativas e não de enumeração de tópicos. Recomenda-se o uso de parágrafo único.” O resumo deve ter de 150 a 500 palavras.

Palavras-chave: Palavra-chave. Palavra-chave. Palavra-chave.

ABSTRACT

...

Key-words: ...

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
UFJF	Universidade Federal de Juiz de Fora
IBGE	Instituto Brasileiro de Geografia e Estatística

LISTA DE SÍMBOLOS

\forall	Para todo
\in	Pertence
\odot	Sorrino
\ominus	Chorano
θ_1	Theta 1

SUMÁRIO

1 Introdução

Contextualizar o trabalho, apontando o problema que se deseja resolver. Descreva os desafios e o impacto do trabalho. Ao final do capítulo, faça uma breve descrição da organização do texto (melhor fazer no fim).

1.1 planejamento

2 REVISÃO BIBLIOGRÁFICA

[portuguese]babel

Neste capítulo abordaremos as técnicas anteriormente mencionadas, que são o foco deste trabalho, PLL (Phase-Locked Loop), análise de prony, ADALINE e RLS (Recursive Least Mean Squares), bem como as convencionais DFT e STFT. Ressaltamos que esta será apenas uma abordagem superficial, não entraremos, portanto, com profundidade nos assuntos tratados.

2.1 DFT

A transformada discreta de Fourier (DFT) de uma sequência discreta $x[n]$ de tamanho N é definida como[5]:

$$X[k] = \sum_{n=0}^{n=N-1} x[n] * e^{(-indentjnk\pi/N)} , \quad k = 0, 1, indent, ..., N - 1 \quad (2.1)$$

Podemos dizer que a sequência $X[k]$ é a DFT de $x[n]$. As duas sequências tem 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 tempo de amostragem da sequência $x[n]$, temos a seguinte relação:

$$Rs = \frac{1}{T_s * N} \quad (2.2)$$

Onde Rs é igual a resolução de $X[k]$. Temos o mapeamento dado como:

$$fk = Rs * k , \quad k = 0, 1, indent, ..., N/indent \quad (2.3)$$

Percebemos 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 a resolução Rs . Para um sinal contendo apenas harmônicos de f_1 , e eventualmente algum valor médio, podemos extrair perfeitamente seus valores de amplitude e fase. Entretanto, para um sinal que possui componentes inter-harmônicos, não é possível fazer o mesmo. Neste caso ocorre o chamado *espalhamento*. Já se pode notar algumas deficiências da DFT. Também temos problemas caso não seja amostrado um período exato do sinal analisado, ou um múltiplo inteiro de um período, podendo a DFT levar a crer, por exemplo, que existe um valor médio no sinal, e outros contúdos equivocados. Este fenômeno é denominado *spectral leakage*[5].

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

Para uma janela de tamanho N , temos:

$$X[k, m] = \sum_{n=0}^{n=N-1} x[m-n] * e^{(-indentjnk\pi/N)}, \quad k = 0, 1, indent, \dots, N-1 \quad (2.4)$$

Desta forma, para cada $m > N$, temos 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 estivéssemos diante de uma amostra completamente nova:

$$X[k, m] = C * \{X[k, m-1] * e^{jindent\pi k/N} + (x[m] - x[m-N]) * e^{jindent\pi k/N}\}, \quad k = 1, indent, \dots, N/indent \quad (2.5)$$

$C=1/N$ para $k=N/indent$ e igual a $indent/N$ para os demais valores. O valor de k foi restringido levando-se em conta a simetria dos sinais reais. [Não sei de onde veio essa equação, vou ver depois]

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

Janela triangular:

$$w_{tri}(n) = 1 - \frac{|indentn - N + 1|}{N - 1}, \quad 0 \leq n \leq N - 1 \quad (2.6)$$

Janela de Hamming:

$$w_{hm}(n) = 0.54 - 0.46 \cos\left(\frac{indent\pi n}{N - 1}\right), \quad 0 \leq n \leq N - 1 \quad (2.7)$$

O uso deste tipo de janelamento atenua eventuais descontinuidades nas extremidades do sinal, auxiliando na medição dos parâmetros. Aplicar uma janela ao sinal significa deslizar a sequência formada por w por $x[n]$, multiplicando termo a termo. Desta forma poderíamos reescrever a equação XXX como:

$$X[k, m] = \sum_{n=0}^{n=N-1} x[m-n] * w(n) * e^{(-indentjnk\pi/N)}, \quad k = 0, 1, indent, \dots, N-1 \quad (2.8)$$

podemos ver nas figuras o efeito da aplicação do janelamento.

2.2 Análise de Prony

A análise de Prony foi desenvolvida em 1795 de modo a explicar a expansão de gases. Ela se assemelha em parte a DFT, entretanto se propõe a ajustar uma soma de exponenciais complexas amortecidas a uma sequência de dados igualmente espaçados, ao passo de que

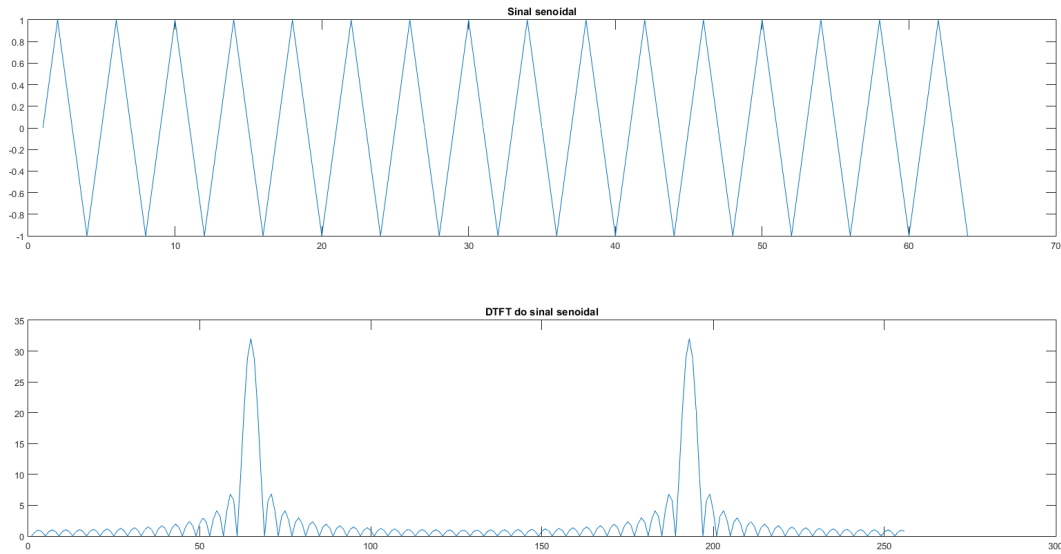


Figura 1 – legenda aqui

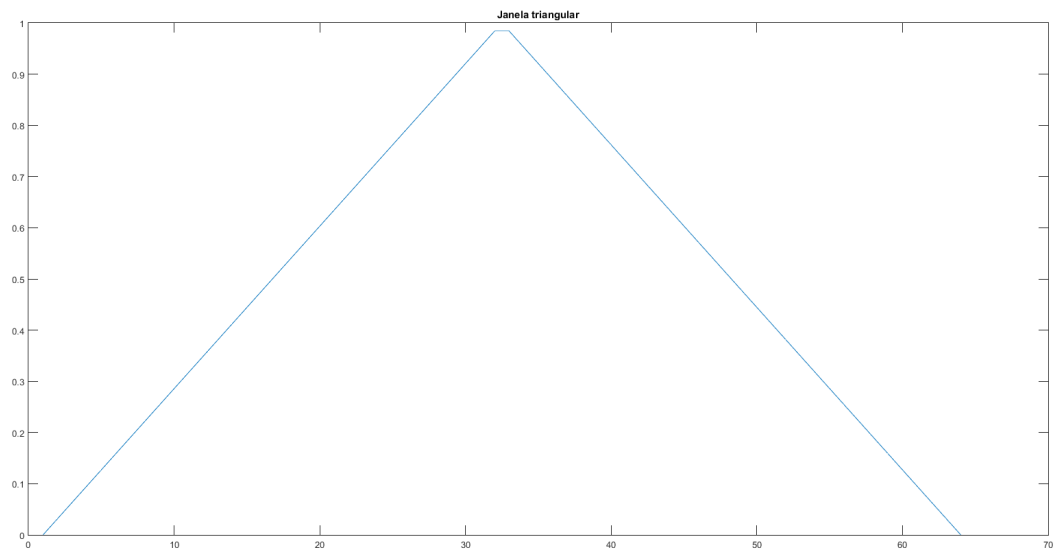


Figura 2 – legenda aqui

a DFT apenas estima as exponenciais complexas e em subdivisões predeterminadas da frequência de amostragem. A análise de prony não é somente uma técnica de análise de sinais, mas também de indentificação de sistemas amplamente utilizada em sistemas de potência, área biomédica, processamento de fala, decaimento radioativo entre outras. A análise de Prony é conhecida por não se comportar muito bem quando um sinal contém ruído, a técnica não faz distinção entre sinal e ruído, e também ajusta as exponenciais ao ruído presente. Seguindo o proposto em [3], temos o seguinte para um sinal $y[k]$ imaginando M exponenciais para se ajustarem a indentM amostras, respeitando o teorema

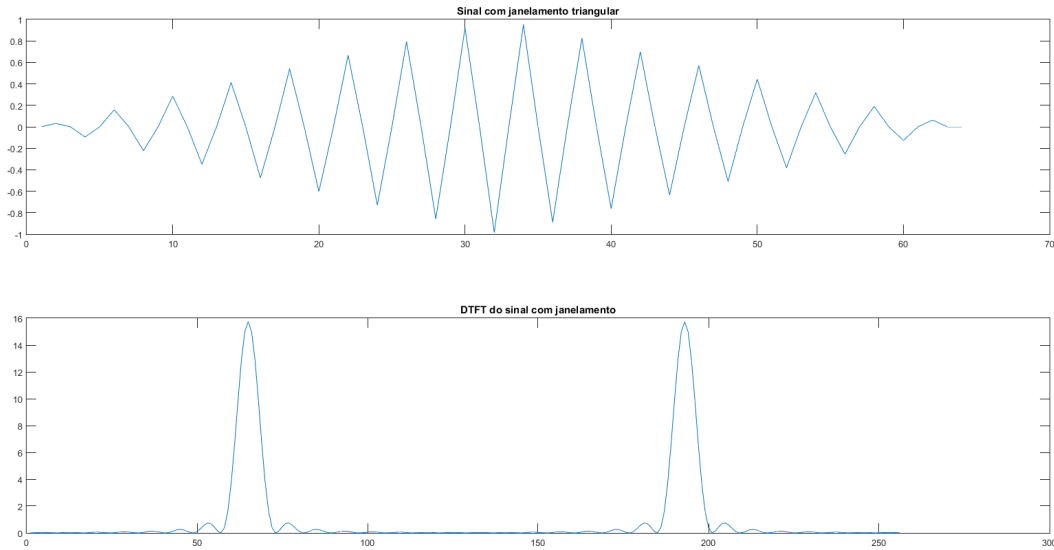


Figura 3 – legenda aqui

da amostragem:

$$y[k] = \sum_{m=1}^M A_m e^{(\alpha_m + j \text{indent} \pi f_m)(k-1)\Delta t + j\sigma_m}, \quad k = 1, \text{indent}, \dots, \text{indent}M \quad (2.9)$$

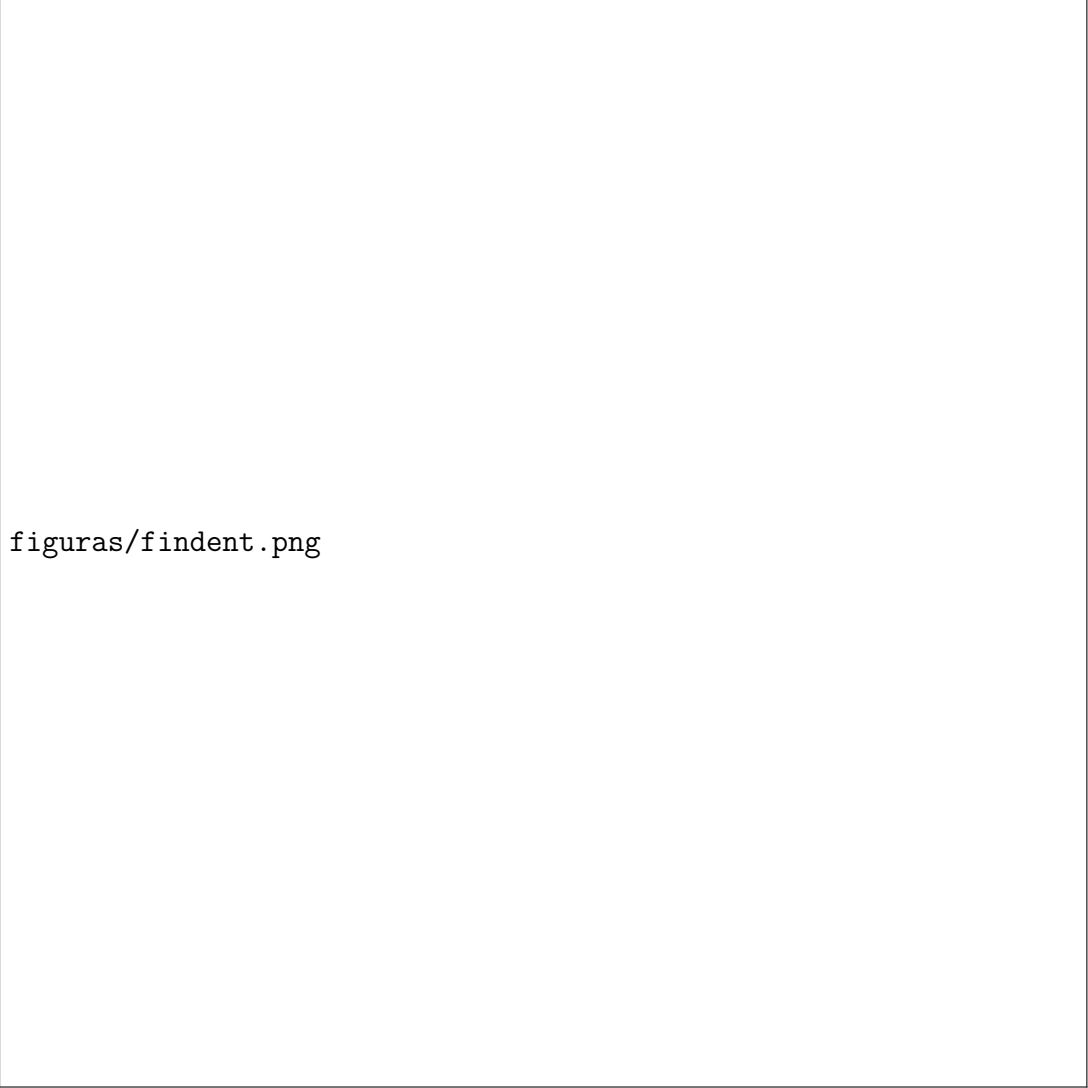
Onde f_m é a frequência das exponenciais, Δt é o tempo de amostragem, α_m é o coeficiente de amortecimento, e σ_m é o ângulo de defasagem. Para o método proposto, estamos apenas interessados em saber quais são as frequências presentes no sinal. Se imaginamos nosso sistema sendo auto regressivo (AR) de ordem M , temos que em qualquer instante de k é possível prever o valor $y[k]$ considerando apenas as M amostras anteriores deste mesmo sinal. Podemos então montar uma equação de diferenças, que neste caso também é um modelo de predição linear[4]:

$$y[k] = \sum_{m=1}^M a_m y[k - m] \quad (2.10)$$

É sabido que exponenciais complexas são soluções para tal modelo. Se tivermos em mãos os valores dos coeficientes a podemos montar o polinômio característico da equação e as raízes deste polinômio (exponenciais complexas) são soluções para nosso problema. De posse dos coeficientes a devemos então encontrar as raízes do polinômio seguinte:

$$P(z) = z^M - a_1 z^{M-1} - \dots - a_M z^0 \quad (2.11)$$

Como os coeficientes a são reais, as raízes complexas estão em pares complexos conjugados, desta maneira cada par complexo representa uma possível senóide já que



figuras/findent.png

Figura 4 – legenda aqui

$$\cos(wt) = \frac{e^{jwt} + e^{-jwt}}{indent}:$$

$$f_m = atan(Im\{z_m\}/Re\{z_m\})indent\pi fs \quad (2.12)$$

Onde z_m é uma das raízes e fs é a frequência de amostragem. Reparemos que as frequências f_m são frequências de possíveis soluções do sistema, não necessariamente elas estarão presentes. Qualquer combinação linear dessas seniíodes também é solução do sistema AR planteado. Para encontrar a forma da solução real, é necessário fazer uso das condições iniciais conhecidas de nosso sistema. No caso, as próprias amostras $y[k]$ as quais temos acesso.

Os coeficientes de amortecimento α_m são o módulo das raízes complexas que encontramos. E se alguma raiz não é complexa, isto somente significa que uma exponencial é solução para nossa equação.

$$\alpha_m = |z_m| \quad (2.13)$$

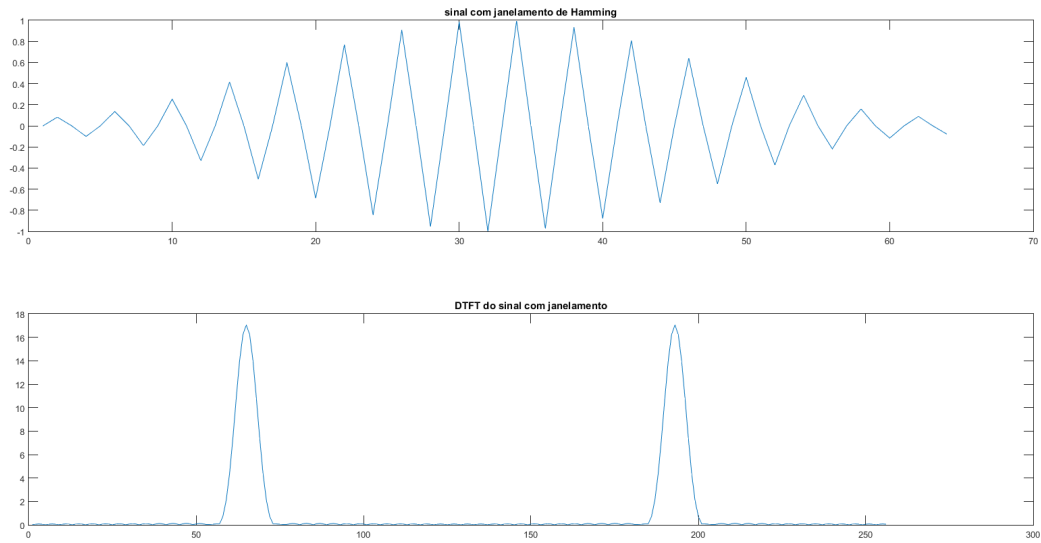


Figura 5 – legenda aqui

2.2.1 Solução do modelo de predição linear

Existem diversas formas de solucionar um modelo deste tipo, apresentaremos algumas opções abaixo[9]:

2.2.1.1 Sistema linear

Conhecendo os valores $y[k]$ que são amostras de nosso sistema analisado, podemos montar um sistema linear considerando a equação anterior:

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

$$(2.14) \quad \begin{bmatrix} y[k-1] & y[k-indent] & \dots & y[k-M] \\ y[k-indent] & y[k-3] & \dots & y[k-M-1] \\ \vdots & & & \vdots \\ y[k-M] & y[k-M-1] & \dots & y[k-indentM+1] \end{bmatrix} \begin{bmatrix} a_1 \\ a_{indent} \\ \vdots \\ a_M \end{bmatrix} = \begin{bmatrix} y[k] \\ y[k-1] \\ \vdots \\ y[k-M+1] \end{bmatrix}$$

Ou de forma simplificada:

$$\mathbf{Y}_k \mathbf{a} = \mathbf{y}_k \quad (2.15)$$

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

$$\mathbf{a} = \mathbf{Y}_k^{-1} \mathbf{y}_k \quad (2.16)$$

Um dos problemas com esta solução é que ela é computacionalmente custosa, já que não está implementada de maneira recursiva, mas pior que isso é o fato de que normalmente temos ruído em nosso sistema, e desta maneira estamos ajustando os valores de \mathbf{a} ao ruído também, o que em geral não é o desejado, e em nosso caso, certamente não é. Desta forma, cada vez que calculamos o vetor \mathbf{a} , ele pode sair completamente diferente do anterior, nos levando a estimações equivocadas. Modelando $y[k]$ como:

$$y[k] = \sum_{m=1}^M a_m y[k-m] + \xi_k \quad (2.17)$$

Em que ξ_k é ruído branco, temos um modelo estocástico de nosso sinal, que nos possibilita pensar em soluções mais arrojadas.

2.2.2 Filtros adaptativos

O Elemento linear adaptativo, ou filtro 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, b o valor de bias, o ADALINE pode ser implementado matricialmente da seguinte maneira:

$$y = \begin{bmatrix} x_1 & x_{indent} & x_3 & \dots & x_n \end{bmatrix} * \begin{bmatrix} w_1 \\ w_{indent} \\ \vdots \\ w_n \end{bmatrix} + b = \mathbf{x} * \mathbf{w} + b \quad (2.18)$$

Filtros adaptativos são considerados sistemas não lineares, portanto a análise de seu comportamento é mais complexa que de a 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[6].

O desenho usual de um filtro adaptativo pode ser visto na figura ??, 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.

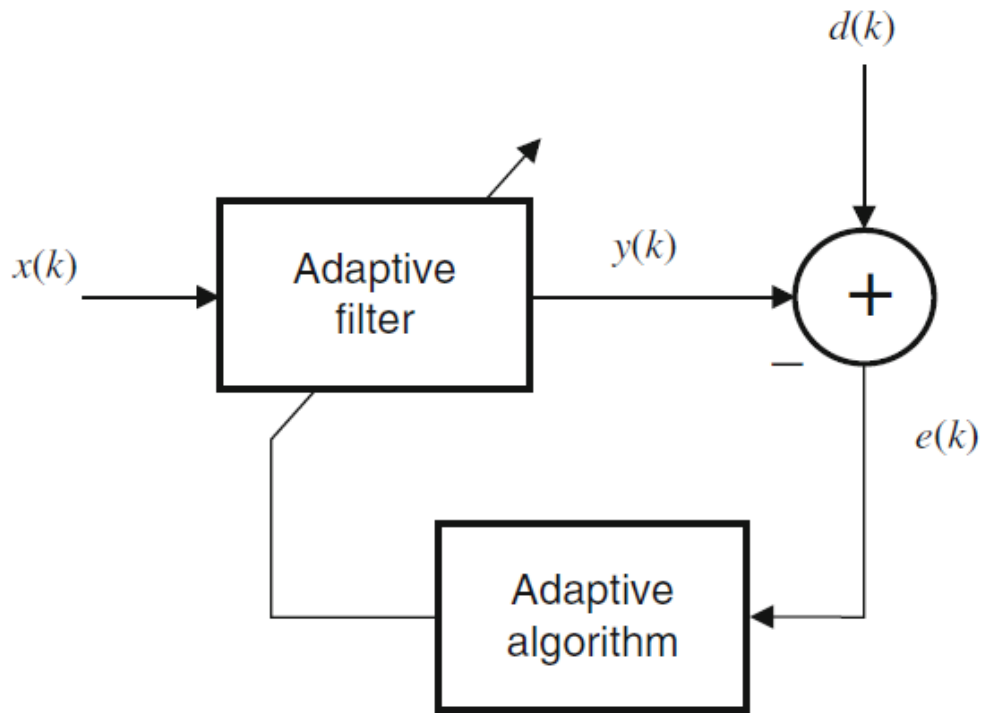


Figura 6 – legenda aqui

2.2.3 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 as variáveis de interesse, para atualizar os valores destas variáveis de forma iterativa.

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

$$y[n] = \sum_{k=0}^{M-1} u[n-k]w_k \quad (2.19)$$

Todo este desenvolvimento pode ser feito de maneira bastante similar considerando entradas complexas e filtros com coeficientes complexos, mas por questões de simplicidade estaremos focando no caso em que ambos são reais. Imaginemos que desejamos que a saída deste filtro seja uma outra série temporal igualmente real $d[n]$, assim nosso erro seria:

$$e[n] = d[n] - y[n] \quad (2.20)$$

Considerando o caso de u estocástico, $e[n]$ é uma variável aleatória. Gostaríamos então de minimizar a esperança de $e^i[n]$, definimos então nossa função de custo:

$$J = E[e^i[n]] = E[(d[n] - y[n])^i] \quad (2.21)$$

Definindo nosso operador gradiente ∇_k :

$$\nabla_k J = -E[e[n]u[n-k]] \quad (2.22)$$

Se queremos encontrar o ótimo com relação a esta função de 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 . A este resultados chamamos princípio da ortogonalidade. Um corolário deste princípio é que y também deve ser ortogonal, ou descorrelacionada com $e[n]$.

É mais conveniente para nós trabalhar com a notação matricial destes resultados. Ao vetor coluna que contém as variáveis aleatórias $u[n-k]$, de $k=0$ a $K=M-1$, lhe chamamos \mathbf{u}_n , ao vetor de coeficientes w , igualmente:

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

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

Desenvolvendo nossa equação anterior, podemos obter:

$$\nabla J = -\text{indent} E[e[n]\mathbf{u}_n] = \text{indent} E[\mathbf{u}_n \mathbf{U}_n^T \mathbf{W} - d[n]\mathbf{u}_n] \quad (2.23)$$

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

$$\mathbf{R}_{uu} \mathbf{w} - \mathbf{r}_{du} = 0 \quad (2.24)$$

De onde podemos concluir que o vetor coeficientes ótimos \mathbf{w}_{opt} é:

$$\mathbf{w}_{opt} = \mathbf{R}_{uu}^{-1} \mathbf{r}_{du} \quad (2.25)$$

Se u é um processo estacionário em sentido amplo, então podemos coletar amostras suficientes para fazer uma boa estimação das matrizes acima, encontrando assim um filtro muito mais apropriado para nossos propósitos. Desta maneira, é possível estimar o conteúdo espectral de u com precisão. Reparemos que este método não é propriamente um gradiente escendente já que encontramos a solução em um passo apenas.

A solução acima tem sua utilidade, entretanto, se a natureza do sinal mudar, não teremos 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 às mudanças ocorridas.

2.2.4 O algoritmo LMS

Para atualizar o vetor de pesos \mathbf{w}_n na direção oposta a do gradiente, definimos um coeficiente de aprendizagem μ . A cada iteração vamos fazer o seguinte:

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

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu(\mathbf{r}_{du} - \mathbf{R}_{uu}\mathbf{w}_{n+1}) \quad (2.26)$$

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} . Vamos estimá-las da seguinte forma:

$$\hat{\mathbf{R}}_{uu} = \mathbf{u}_n \mathbf{u}_n^T \quad (2.27)$$

$$\hat{\mathbf{r}}_{du} = \mathbf{u}_n d[n] \quad (2.28)$$

O que pode soar uma heresia, mas faz sentido se pensarmos que apenas temos M amostras de sinal.

2.2.5 O algoritmo NLMS

Uma melhoria no algoritmo anterior pode ser proposta. Se multiplicamos a parte que está entre parênteses na equação XX:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu \mathbf{R}_{uu}^{-1}(\mathbf{r}_{du} - \mathbf{R}_{uu}\mathbf{w}_{n+1}) = \mathbf{w}_{n-1} + \mu(\mathbf{w}_{opt} - \mathbf{I}\mathbf{w}_{n-1}) = \mathbf{w}_{opt} \quad (2.29)$$

Em teoria, nosso algoritmo convergiria em apenas uma iteração. Resta apenas o empecilho de que estimar tal matriz. Com estimador que tínhamos antes isto não é possível, porque a matriz estimada $\hat{\mathbf{R}}_{uu}$ não é inversível por ser o produto de dois vetores. Mas podemos fazer uma aproximação da forma $\hat{\mathbf{R}}_{uu} + \mathbf{I}\epsilon$. Que certamente é inversível e pode nos dar uma inversa útil, para valores pequenos de ϵ . E após alguma álgebra[10], que pode ser encontrada em [9], chegamos ao seguinte:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu \frac{\mathbf{u}_n}{|\mathbf{u}_n|^2 + \epsilon} (d[n] - \mathbf{u}_n \mathbf{w}_{n-1}) = \mathbf{w}_{n-1} + \mu \frac{\mathbf{u}_n}{|\mathbf{u}_n|^2 + \epsilon} e[n] \quad (2.30)$$

A este algoritmo chamamos LMS normalizado, ou NLMS. Há ainda o estudo de convergência para valores pequenos de μ , que também pode ser encontrado em [6][9], tanto para o LMS quanto para o NLMS. Mas de modo geral o NLMS deve convergir respeitada todas as condições estabelecidas anteriormente e escolhendo valores de $\mu < 1$.

2.3 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 apartir do LMS (*Least Mean Squares*). O RLS é conhecido por possuir rápida convergência, tendo boa performance em sistemas variantes no tempo, como o caso de nosso interesse. Isto vem ao custo de certa complexidade computacional aliada a problemas de estabilidade[6].

Voltando ao modelo do filtro adaptativo da seção anterior, vamos definir $\mathbf{x}(k)$ como o vetor contendo N amostras atrasadas de nosso sinal de entrada, ou seja $x(k)$. Assim:

$$\mathbf{x}(k) = [x(k-1) \quad x(k-indent) \quad \dots \quad x(k-N)] \quad (2.31)$$

Sendo N a ordem do filtro. Da mesma maneira, podemos definir $\mathbf{w}(k)$ como o vetor contendo os coeficientes do filtro na k ésima iteração. Assim:

$$\mathbf{w}(k) = \begin{bmatrix} w_1(k) \\ w_{indent}(k) \\ \vdots \\ w_N(k) \end{bmatrix}$$

(2.32)

A função objetivo para o RLS é determinística e dada por:

$$\xi(k) = \sum_{i=0}^k \lambda^{k-i} \varepsilon^i indent(i) = \sum_{i=0}^k \lambda^{k-i} [d(i) - \mathbf{x}(k) * \mathbf{w}(k)] \quad (2.33)$$

Onde ε é o erro a posteriori no instante i . O parâmetro λ é um fator de peso exponencial que deve ser escolhido entre 0 e 1, de modo que $0 < \lambda < 1$. λ representa o fator de esquecimento do erro, quanto menor seu valor, menor será a influência das amostras passadas sobre a atualização atual.

Após alguma álgebra que pode ser encontrada em [6], temos o seguinte:

$$\mathbf{w}(k) = \mathbf{S}_d(k) * \mathbf{p}_d(k) \quad (2.34)$$

$$\mathbf{S}_d = \lambda^{-1} \left[\mathbf{S}_d(k-1) - \frac{\mathbf{S}_d(k-1) \mathbf{x}^T(k) \mathbf{x}(k) \mathbf{S}_d}{\lambda + \mathbf{x}(k) \mathbf{S}_d(k) \mathbf{x}^T(k)} \right] \quad (2.35)$$

$$\mathbf{p}_d(k) = \lambda \mathbf{p}_d(k-1) + d(k) * \mathbf{x}(k) \quad (2.36)$$

Boas recomendações de inicialização de \mathbf{S}_d e \mathbf{p}_d são $\mathbf{S}_d = \delta \mathbf{I}$ e $\mathbf{p}_d = [0 \ 0 \ \dots \ 0]^T$. Nas quais δ pode ser o inverso da potência estimada do sinal.

2.4 PLL

Um Phase Locked Loop digital, assim como sua versão analógica, visa determinar os parâmetros de um processo estocástico como uma onda senoidal. Desta forma o PLL tenta extrair parâmetros de Amplitude, fase e frequência. Temos diversas variantes digitais e analógicas[7], a variante utilizada neste trabalho foi proposta por Ziarani em indent004[8]. 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 sendo A a amplitude do sinal e $\phi(t)$ sua fase, tendo este sinal uma frequência constante, podemos escrever $\phi(t) = wt + \delta$, sendo w igual a frequência e δ um valor de fase constante. Podemos escrever de maneira mais geral:

$$u(t) = \sum_{i=0}^{\infty} A_i \sin \phi_i(t) + n(t) \quad (2.37)$$

Onde $n(t)$ representa um distúrbio, como um ruído. Na realidade, 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 $y(t)$.

Definimos a função d :

$$d^{indent}(t, \Psi(t)) = [u(t) - y(t, \Psi(t))]^{indent} = e^{indent}(t) \quad (2.38)$$

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

$$\frac{d\Psi(t)}{dt} = -\boldsymbol{\mu} \frac{\partial J(\Psi(t), t)}{\partial \Psi(t)} \quad (2.39)$$

$$\boldsymbol{\mu} = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_{indent} & 0 \\ 0 & 0 & m_3 \end{bmatrix} \quad (2.40)$$

Com $\frac{d\Psi(t)}{dt}$ denotando o vetor na direção do qual são atualizados os valores de $\Psi(t)$ a cada iteração. E $\boldsymbol{\mu}$ a matriz diagonal contendo as constantes de atualização referentes a cada parâmetro. Como estaremos lidando com estimativas agora, usaremos a notação $\hat{\Psi}(t)$:

$$\begin{bmatrix} \frac{d\hat{A}(t)}{dt} \\ \frac{d\hat{w}(t)}{dt} \\ \frac{d\hat{\delta}(t)}{dt} \end{bmatrix} = -\boldsymbol{\mu} \begin{bmatrix} \frac{\partial e^{i\text{ndent}(t)}}{\partial \hat{A}} \\ \frac{\partial e^{i\text{ndent}(t)}}{\partial \hat{w}} \\ \frac{\partial e^{i\text{ndent}(t)}}{\partial \hat{\delta}} \end{bmatrix} \quad (2.41)$$

Podemos obter então o seguinte:

$$\frac{d\hat{A}(t)}{dt} = \text{indent}m_1e(t)\sin\left(\int_0^t \hat{w}(\tau)d\tau + \hat{\delta}(t)\right) \quad (2.42)$$

$$\frac{d\hat{w}(t)}{dt} = \text{indent}m_i\text{ndente}(t)\hat{A}(t)t\cos\left(\int_0^t \hat{w}(\tau)d\tau + \hat{\delta}(t)\right) \quad (2.43)$$

$$\frac{d\hat{\delta}(t)}{dt} = \text{indent}m_3e(t)\hat{A}(t)\cos\left(\int_0^t \hat{w}(\tau)d\tau + \hat{\delta}(t)\right) \quad (2.44)$$

$$\frac{d\hat{\phi}(t)}{dt} = \hat{w}(t) + \frac{\hat{\delta}(t)}{dt} \quad (2.45)$$

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 $m4$, que pode ser absorvida pela constante mindent , tornando o sistema invariante no tempo e fazendo com que este sistema de equações seja útil na prática. Desta maneira obtemos o seguinte:

$$\frac{d\hat{A}(t)}{dt} = \text{indent}\mu_1e(t)\sin\left(\int_0^t \hat{w}(\tau)d\tau + \hat{\delta}(t)\right) \quad (2.46)$$

$$\frac{d\hat{w}(t)}{dt} = \text{indent}\mu_i\text{ndente}(t)\hat{A}(t)t\cos\left(\int_0^t \hat{w}(\tau)d\tau + \hat{\delta}(t)\right) \quad (2.47)$$

$$\frac{d\hat{\phi}(t)}{dt} = \hat{w}(t) + \text{indent}\mu_3e(t)\hat{A}(t)\cos\left(\int_0^t \hat{w}(\tau)d\tau + \hat{\delta}(t)\right) \quad (2.48)$$

Usando o método Euler Forward:

$$A[n+1] = A[n] + \mu_1T_se[n]\sin(\phi(t)) \quad (2.49)$$

$$w[n+1] = w[n] + \mu_i\text{ndent}T_se[n]\cos(\phi(t)) \quad (2.50)$$

$$\phi[n+1] = \phi[n] + T_sw[n] + \mu_3T_se[n]\cos(\phi(t)) \quad (2.51)$$

Desta forma obtemos o conjunto de equações discretas que utilizaremos para rastrear frequências no restante do trabalho.