

0.1 Identificação de frequências

Vimos na seção sobre o PLL-Multitaxa que este é um método com grande eficácia no rastreamento de frequências, entretanto precisamos saber onde elas estão, e isto o método não nos diz. Seria possível por exemplo utilizá-lo como um filtro de partículas e largar PLLs aleatoriamente pelo espectro do sinal, entretanto temos algumas formas de obter boas ideias de onde estão as componentes mais relevantes do sinal. Uma delas é a DFT ou a STFT com janelamento para reduzir o espalhamento pelo espectro, desta forma podemos identificar regiões do espectro onde sabemos que temos energia relevante, sem saber no entanto a localização exata da componente, ou as componentes as quais pertence esta energia. Outro método visto na revisão foi o de Prony, do qual se fez uso no trabalho [3]. Neste trabalho utilizaram da otimização via RLS para resolver o problema da predição linear e calcularam as raízes do polinômio característico do modelo Auto-Regressivo do sinal, assim puderam obter as componentes relevantes com certa precisão.

Neste mesmo trabalho, posteriormente se faz uso do RLS novamente para estimar a amplitude e fase de cada uma das componentes, alimentando o RLS com sinais senoidais em quadratura de modo que o filtro estimado fosse a amplitude de senoides e cossenoides [obs: Vou colocar uma diagrama disso depois].

Nesta seção faremos a exposição de um método baseado no trabalho [3] com algumas modificações.

0.1.1 Solução em predição linear

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

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

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

$$\mathbf{w}_{opt} = \mathbf{R}_{uu}^{-1} \mathbf{r}_{du} \quad (2)$$

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

0.1.2 Solução em RLS e NLMS

Como vimos anteriormente, o RLS faz uma estimação da matriz \mathbf{R}_{uu}^{-1} a cada iteração e geralmente é o método que nos vai dar menor MSE e em menos tempo, entretanto ele tem algumas deficiências:

- Se a matriz \mathbf{R}_{uu} é singular ou não é bem ajustada, podemos ter problemas numéricos significativos, como a ordem dos elementos da matriz ser muito desbalanceada ou grande demais. Isso é bastante notável em nosso problema de estudo, pois em geral não sabemos a ordem do sistema que estamos analisando, se este sistema tem uma ordem pequena e o estimamos com um filtro grande, a matriz \mathbf{R}_{uu} certamente será singular. Algo que podemos fazer para minimizar este problema é a adição de ruído.
- Em geral é mais lento para reagir a variações no sistema que o LMS e NLMS.
- Tem uma complexidade computacional consideravelmente maior.
- Sua convergência é de certa forma caótica, perturbando o cálculo das raízes.

Devemos também destacar que tanto o RLS quanto o NLMS se beneficiam de ordens de filtro próximas a quantidade de amostras por ciclo da fundamental. Isto se deve ao fato de que quanto menor são as variações dos sinais dentro do buffer \mathbf{U} mais singular é a matriz de autocorrelação.

Fazendo uma simulação com os harmônicos de 1 até 7, com a amplitude igual ao inverso de sua ordem e ruído gaussiano com $\sigma = 0.02$, comparamos 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:

	M=16	M=64
f0x16	1.3e-03	2.1e-04
f0x64	2.7e-04	8.6e-05

A tabela mostra o que já esperávamos, e o que pode ser confirmado com outras simulações mais adiante, obtemos resultados melhores com uma ordem de modelo menor e com menores taxas de amostragem.

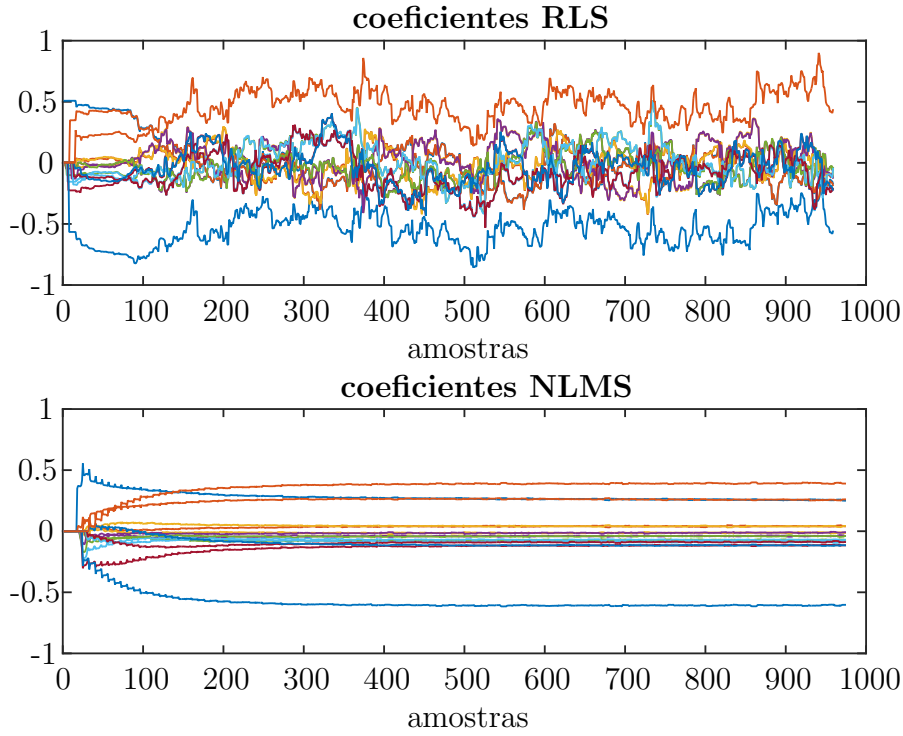


Figura 1: Convergência dos coeficientes RLS e NLMS na presença dos harmônicos 1, 3, 5 e 7; $M=16$

0.1.3 Simulações

Foram realizadas simulações com os seguintes parâmetros: $f_s = 32f_0$, $M = 32$, harmônicos de 1 a 15, pares e ímpares todos com amplitude unitária. E é considerada uma variação aleatória nestas frequências somando-lhes um valor aleatório de uma V.A gaussiana com determinada amplitude. Apresentamos na tabela abaixo 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:

	NLMS	RLS
G(0, 0.1)	0	0
G(0, 0.25)	1.9	2.3
G(0, 0.5)	18.4	3.4
G(0, 0.7)	22.8	6.8
G(0, 1)	28.3	10.2

Tabela 1: resultados em %

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. Assim podemos ter uma boa ideia da capacidade do algoritmo de classificar frequências diferentes, mas que estão bastante próximas.

Também foram realizadas simulações para o caso de amplitude variável dos harmônicos, como uma V.A uniforme de 0 a 1.

	NLMS	RLS
G(0, 0.1)	3.4	0.6
G(0, 0.25)	14.6	2.0
G(0, 0.5)	28.4	8.5
G(0, 0.7)	31.3	12.9
G(0, 1)	34.5	16.6

Tabela 2: Simulação com amplitude variável, resultados em %

frequências 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 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 3.

	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 3: Tabela com os erros para classificação de frequências próximas

0.1.4 complexidade computacional

A complexidade computacional do RLS é de cerca de $3M^2 + 4M$ somas e multiplicações por iteração[14], 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, dependendo do hardware usado, o custo do RLS pode não ser tão mais alto que o do NLMS.

0.1.5 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. Mostramos que o NLMS faz uma estimação mais estável dos coeficientes e conseqüentemente das raízes e harmônicos, para uma implementação online.

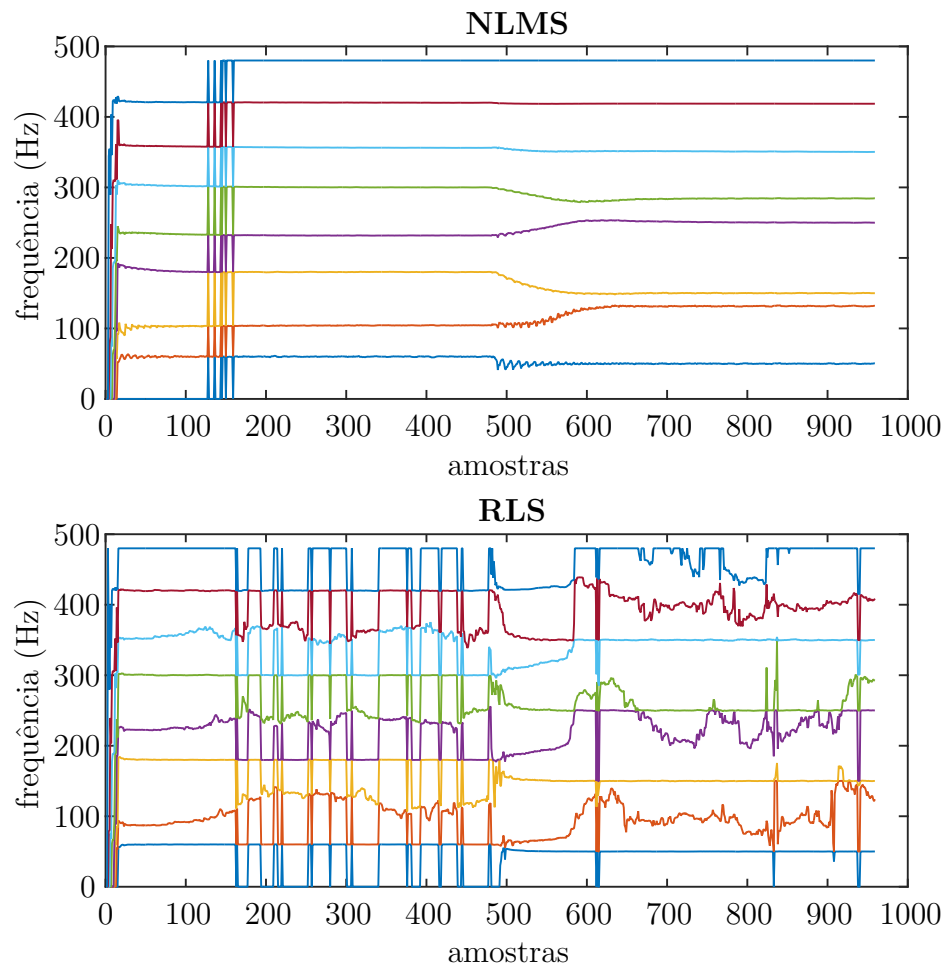


Figura 2: Convergência do RLS e NLMS vista na estimação das frequências, com um degrau de 10 Hz em 500 amostras

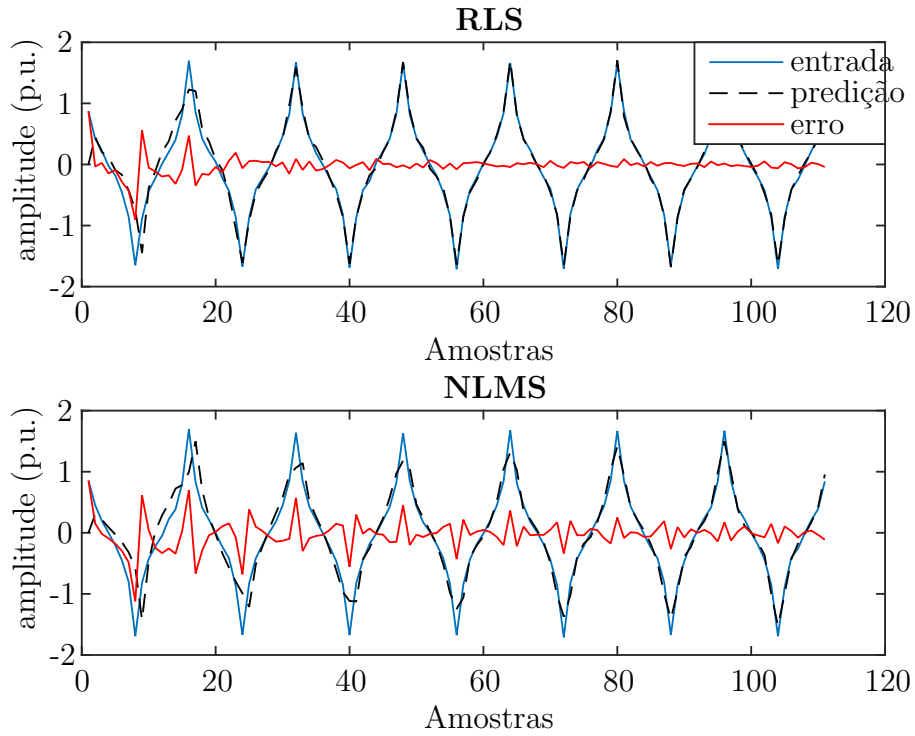


Figura 3: Convergência do RLS e NLMS na presença dos harmônicos 1, 3, 5 e 7; $M=16$

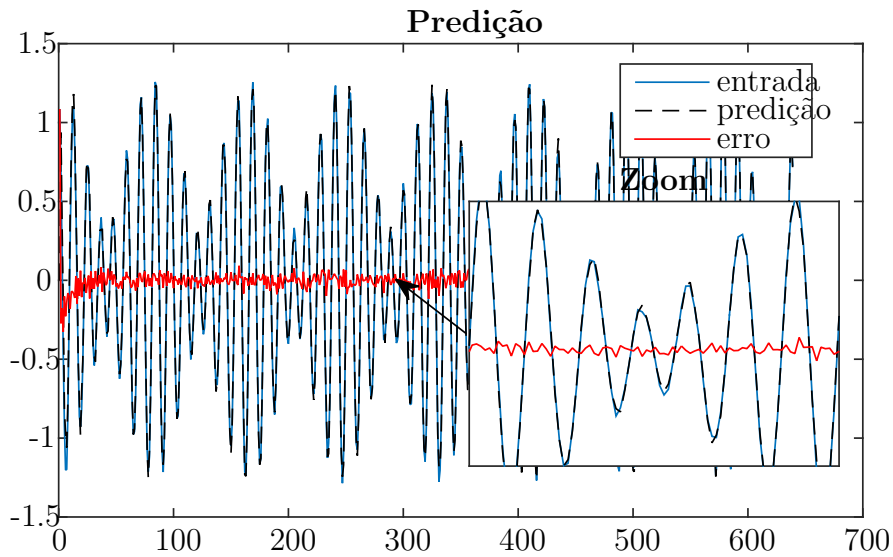


Figura 4: Efeito de batimento em frequências muito próximas