0.1 Estrutura PLL Multitaxa

Como vimos 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. Nas figuras seguintes podemos ver como converge o algoritmo em diferentes situações, todas foram simuladas para um sinal de 180 V de amplitude e constantes 300, 500, e 6, com frequência de amostragem igual a 7680 Hz, partindo de condições inicias $f_i = 60Hz$ e $A = 0$. Podemos perceber pela simulação que o algoritmo converge rapidamente mesmo com ruído, entretanto, na presença de harmônicos com a mesma quantidade de energia, a convergência já é bastante comprometida. Ainda assim, em valor médio, a estimação se mostra correta. Podemos concluir 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.

0.1.1 banco de filtros

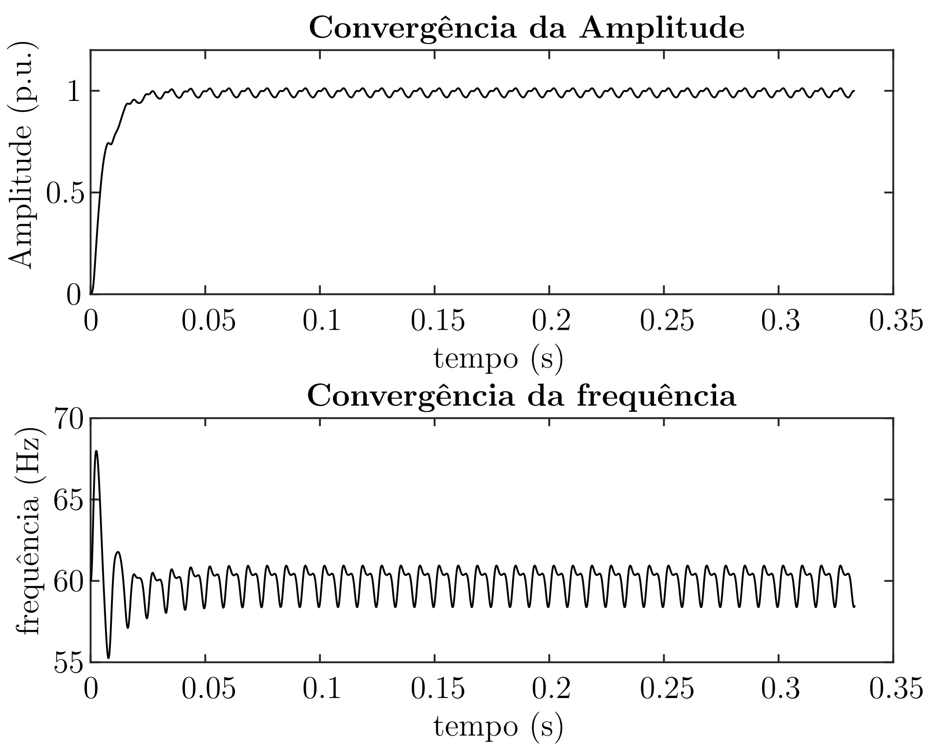

Figura 1: Convergência na presença do 3º e 5º harmônicos; SNR=10 dB

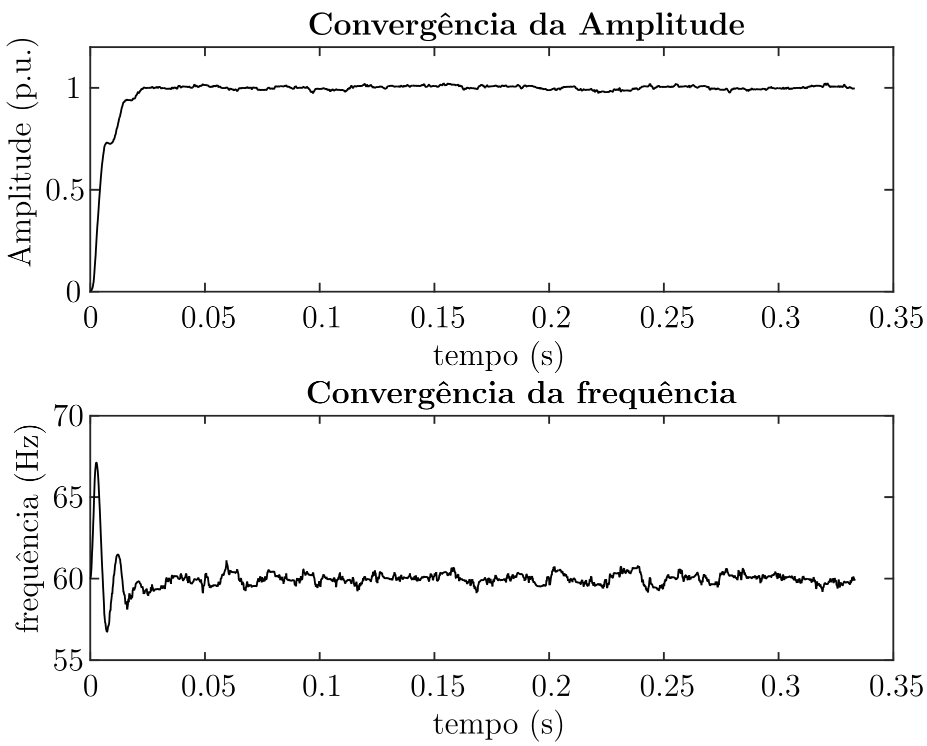

Figura 2: Convergência na presença de ruído; SNR=10 dB

A solução encontrada em [2] é o uso de um preprocessamento 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.

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

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

$$\beta = cos(w_0) \tag{2}$$

O parâmetro α modifica a seletividade do filtro e está entre 0 e 1, para que este seja estável. Quanto mais próximo de 1, mais seletivo é o filtro. O parâmetro β modifica a frequência central do filtro de acordo com a equação 2, 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 entAõ muito fácil utilizá-lo e modificá-lo em tempo real.

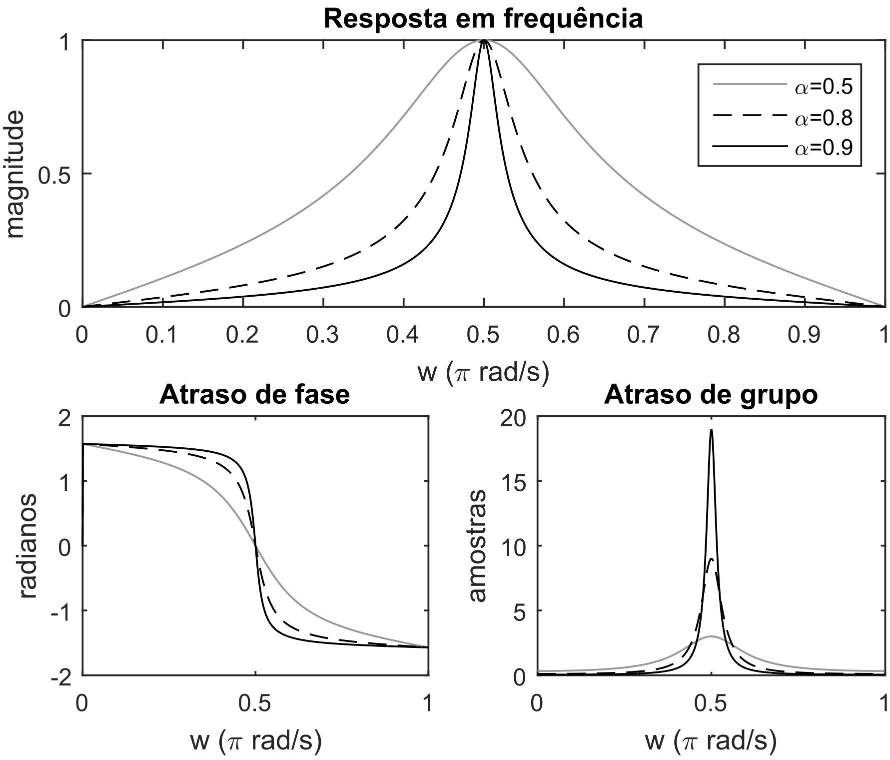

Figura 3: Características do filtro com w_0=0.25

Temos 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. Devemos também ter atenção com a frequência de amostragem, pois quanto maior, menos seletivo um mesmo α seria. Se por exemplo, mantemos um α e aumentamos f_s, frequências que antes estavam normalizadas mais longe de nossa frequência central anterior, agora estariam mais próximas, por assim dizer. Dessa forma quanto 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 temos mais amostras em menos tempo devido ao aumento em f_s. Todos estes efeitos estão muito bem descritos por J. Rodrigues em seu trabalho [13].

0.1.2 Uso da estrutura multitaxa

Depois de passado pelo filtro passa-banda, podemos realizar a subamostragem do sinal, uma vez que em tese eliminamos os harmônicos mais distantes quase que completamente, e estes não interferirão em nossa estimação. Na revisão bibliográfica foi discutido o perfil desta interferência e também como encontrar a posição de uma frequência depois de esta sofrer aliasing. Agora faremos a exposição do algoritmo em C de uma função simples capaz de calcular esta frequência, que pode ser até mais explicativo que o texto anterior:

```c
float freq_finder(const float Fs,
                  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 Fs, 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 necessitamos dela para recuperar a frequência original estimada.

Podemos observar na figura 4 a localização de duas frequências f1 e f2

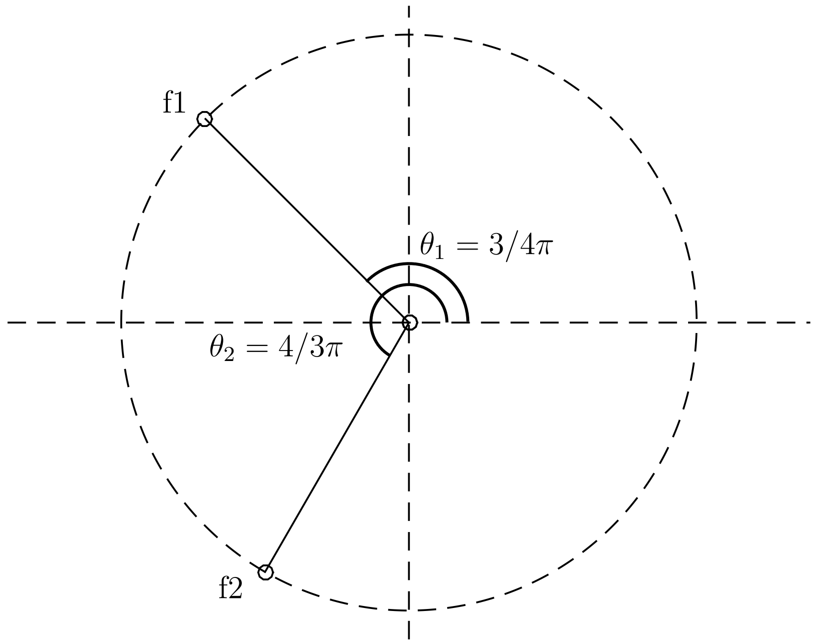

Figura 4: círculo de frequências

no círculo. Suponhamos que a frequência de amostragem é $f_s = 240\ Hz$, $f_2 = \frac{240}{2}\frac{3}{4}\ Hz = 90\ Hz$ e $f_2 = \frac{240}{2}\frac{4}{3}\ Hz = 160\ Hz$, encontramos facilmente suas colocações no círculo utilizando o algoritmo citado. Agora reparemos que diferentes frequências serão mapeadas nas mesmas posições do círculo, por exemplo $f_1 = \frac{240}{2}\frac{11}{4}\, Hz = 330\, Hz$ também é mapeada no mesmo ângulo, então não temos uma função inversa que devolva as frequências mapeadas para seus valores reais.

Também nos atentemos para o fato de que se aumentamos f_1 levemente, estaremos aumentando sua frequência aparente, mas se fazemos o mesmo com f_2 estaremos diminuindo sua frequência aparente. Por isso é importante guardar a variável 'flag', ela nos diz se acréscimos positivos em frequência aparente condizem à acréscimos ou decréscimos na frequência real, e uma vez que não temos uma função inversa que nos devolva a frequência real dada a aparente, temos que utilizar da variação na estimação aparente e o valor inicial que mapeamos para recuperar a estimação real.

$$\hat{f} = f_{init} + \Delta f_{aparente}\, flag \tag{3}$$

0.1.3 variação da frequência central do banco de filtros

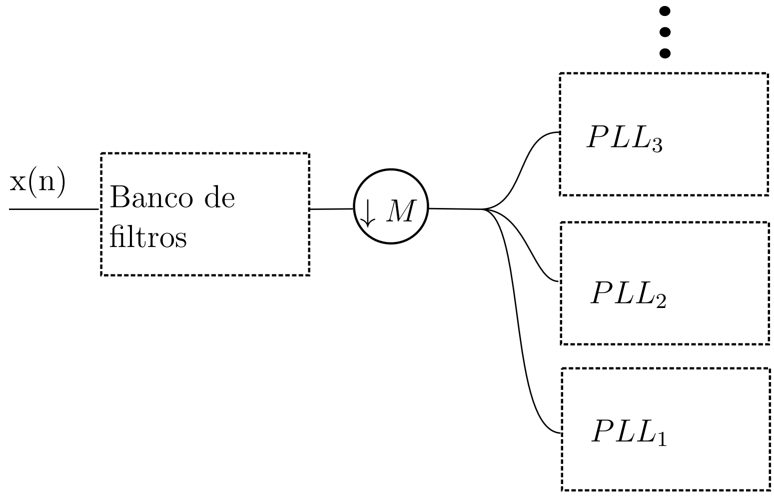

Figura 5: esquema multitaxa PLL

A estrutura multitaxa consiste em passar o sinal de entrada $x(t)$ 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 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. Vimos 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 prejudicar sua convergência. A alternativa encontrada por J. Rodrigues [13] 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:

$$w_0[k] = \frac{1}{L} \sum_{i=k-L+1}^{k} \hat{w}[i] \tag{4}$$

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

$$w_{rec}[k] = w_{rec}[k-1]\lambda + \hat{w}[k] \tag{5}$$

$$w_0[k] = \frac{1}{1-\lambda} w_{rec}[k] \tag{6}$$