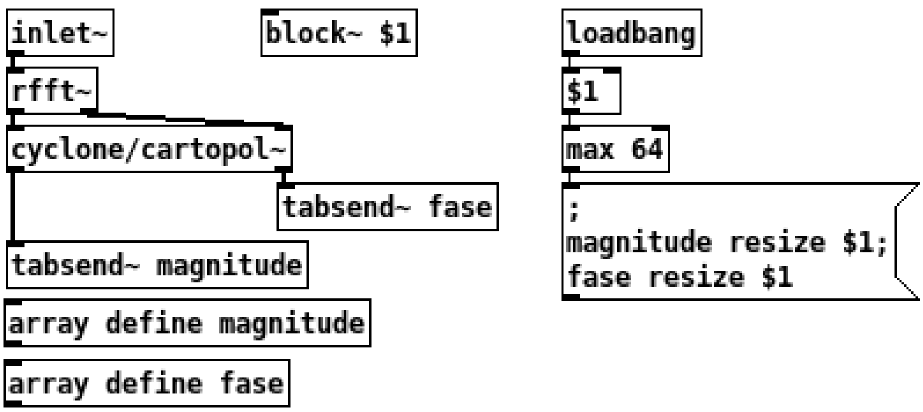
CompMus 2020 - Roteiro para a tarefa prática 8 - 18/12/2020

Estimação de frequência fundamental

1) Abra o Puredata, crie um novo patch, acrescente nome e número USP como comentários e salve-o. Lembre-se de fazer entregas parciais a cada etapa da implementação, submetendo o trabalho no link Oitava Tarefa Prática (TP8).

2) Neste trabalho nós implementaremos e compararemos 3 técnicas elementares de estimação de frequência fundamental a partir do máximo pico de amplitude no espectro de magnitude: a própria frequência do pico, a interpolação quadrática, e a diferença de fase do pico entre janelas sucessivas. Além disso usaremos o nosso patch para mensurar os erros cometidos pelas 3 estimativas aplicadas a sinais senoidais em diferentes faixas de frequências. Especificamente, devemos escrever uma abstração `rastreiaF0~.pd` com um argumento N representando o tamanho do bloco de análise, um `[inlet~]` para o sinal de entrada e três `[outlet]` para os três valores estimados de frequência. Use um `[block~ $1]` no próprio `rastreiaF0~.pd` (caso $1 não esteja definido, o Pd manterá o valor default $N = 64$ para o tamanho do bloco de análise). A análise envolve os espectros de magnitude e de fase, que devem ser armazenados em tabelas compatíveis com o tamanho de bloco de análise. Use a imagem abaixo para começar o seu patch:

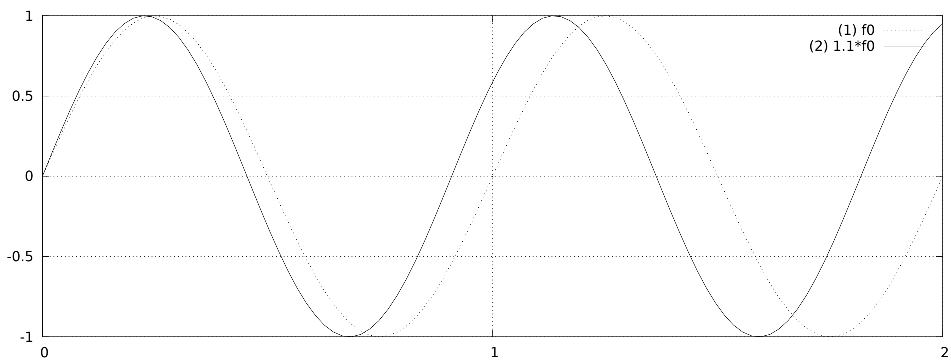


Um índice $k \in \left[0 \ldots \frac{N}{2}\right]$ correspondente ao máximo do espectro de magnitude deve ser computado a cada bloco DSP. Você pode usar `[bang~]`$\longrightarrow$`[array max magnitude]` para isso. A partir deste índice k computaremos as três estimativas de frequência a seguir.

3) **Frequência de pico:** esta estimativa corresponde à expressão $\frac{kR}{N}$, onde $R = 44100$ é a taxa de amostragem. No caso de uma entrada senoidal, o erro máximo cometido por essa estimativa é de $\pm\frac{1}{2}\frac{R}{N}$, de onde se pode observar que a precisão aumenta com o tamanho de bloco.

4) **Interpolação quadrática:** sendo $F[k-1]$, $F[k]$ e $F[k+1]$ os valores do espectro de magnitude nestes 3 índices, o polinômio quadrático interpolador é dado pela expressão $F(x) = A(x-k)^2 + B(x-k) + C$ onde $A = (F[k+1] - 2 * F[k] + F[k-1])/2$, $B = (F[k+1] - F[k-1])/2$ e $C = F[k]$, de onde o valor de $\bar{x}$ correspondente ao máximo de $F(x)$ será dado por $\bar{x} = k - \frac{B}{2A}$. A estimativa produzida no segundo outlet deve ser então a frequência correspondente $\frac{\bar{x}R}{N}$. **Observe** que esse método só pode ser aplicado se $0 < k < \frac{N}{2}$, do contrário não é possível obter uma interpolação quadrática. Você pode usar `[ofelia]` ou objetos Pd puros (é possível acessar o espectro de magnitude por exemplo com `[expr magnitude[$f1]]`).

5) Diferença de fase: para entender a relação entre a variação de fase inicial de um certo sinal e o uso dessa informação para re-estimar a frequência fundamental, considere o sinal senoidal da figura abaixo, com frequência $1.1f_0$, próxima da frequência de análise f_0. Na primeira janela, a fase inicial é 0, enquanto que na segunda janela a fase inicial é $\frac{2\pi}{10}$, pois o sinal já percorreu 10% da sua segunda volta pelo ciclo trigonométrico na janela anterior. Este acúmulo de fase caracteriza a diferença entre a frequência real do sinal $(1.1f_0)$ e a frequência de análise usada pela FFT (f_0).



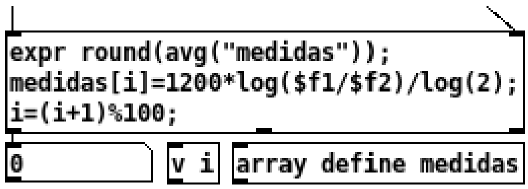
Desta forma, é possível re-estimar a frequência real do sinal próxima do k-ésimo bin da FFT a partir da variação de fase inicial $\Delta = \varphi^{(n+1)}(k) - \varphi^{n}(k)$ entre duas janelas sucessivas. Considerando que a distância entre os inícios de duas janelas sucessivas é de $\frac{N}{R}$ seg, a frequência real em relação ao k-ésimo bin da FFT será

$$f(k) = \left(k + \frac{\Delta}{2\pi}\right)\frac{R}{N},$$

No exemplo da figura, temos $k = 1$ (associado à frequência da janela) e $\Delta = \frac{2\pi}{10}$, de onde $f(k) = 1.1\frac{R}{N}$.

Para computar Δ seu código deve **lembrar** do valor do espectro de fase no índice k computado na última janela. Para simplificar o trabalho, considere que o índice k não muda de uma janela para a outra, e portanto basta armazenar o valor de fase encontrado no índice k para uso como "fase anterior" no próximo bloco. Se quiser usar `[ofelia d]`, basta criar um atributo/variável fora das funções; se preferir Pd puro, pode-se armazenar o valor no inlet frio de um objeto `[f ]`, para ser recuperado na próximo bloco com um *bang*. Converta Δ para o ângulo equivalente no intervalo $[-\pi, +\pi]$ com o código `if` $\Delta > \pi$ `then` $\Delta = \Delta - 2\pi$ `elseif` $\Delta < -\pi$ `then` $\Delta = \Delta + 2\pi$ `end;`

6) Para testar o seu patch e obter algumas medidas de erro dessas 3 estratégias, use a construção abaixo para medir os erros das estimativas em cents usando a média de 100 observações consecutivas:



Aqui o primeiro inlet recebe uma estimativa $\tilde{f}_0$ e o segundo inlet recebe o valor real de f_0 usado no oscilador. Alimente sua abstração `[rastreiaF0~ N]` com um oscilador senoidal de frequência f_0, e teste o seu patch com os parâmetros $N = 512, 1024, 2048$ e com as frequências $f_0 = 100, 1000, 10000$ Hz, anotando no patch os 9 valores de erros médios em cents correspondentes às combinações de valores de N e f_0, para cada método de estimação. **Dica:** sempre que alterar os valores de f_0 ou N, aguarde um pouco (100 blocos) para a leitura do erro se estabilizar.

7) Entregue o trabalho no e-disciplinas.