ELT 355 - Laboratório de sinais e sistemas

Aula prática 6: Filtragem no domínio da frequência

Departamento de Engenharia Elétrica
Centro de Ciência Exatas e Tecnológicas, Universidade Federal de Viçosa

1 Introdução

Nesta prática realizaremos a filtragem de sinais usando o domínio da frequência, i.e., multiplicando-se a transformada de Fourier da entrada com a resposta em frequência do sistema linear invariante no tempo. Especificamente, serão processados sinais de audio através de filtragem passa-baixa e passa-alta.

2 Comandos úteis

- `help`, ajuda para usar um comando.
- `fft`, transformada de Fourier.
- `ifft`, transformada de Fourier inversa.
- `audioread`, lê um arquivo WAV.
- `audiowrite`, escreve um arquivo WAV.
- `fftshift`, ajusta o resultado da `fft`.

3 Tempo↔Frequência

A faixa de frequência do espectro de um sinal amostrado é relacionada com sua frequência de amostragem. No exemplo abaixo, está representada a transformada de Fourier de uma soma de senoides com ruído, onde a frequência de amostragem é $F_s = 8000$. Neste caso, a largura do espectro está na faixa `[-4000,4000]` Hz. Tanto o sinal no domínio do tempo quanto no domínio da frequência tem o mesmo número de pontos e a resolução em frequência (ou Δf) é inversamente proporcional a este número. A transformada de Fourier é obtida através do comando `fft`, o qual implementa o algoritmo "Fast Fourier Transform".

```
Fs = 8000;
L = 8000;
t = 0:1/Fs:(L-1)/Fs;
x = 0.7*sin(2*pi*500*t)+sin(2*pi*2000*t)+2*randn(1,L);
plot(t,x)
X = fft(x);
```

O resultado do comando `fft` é uma transformada não-simétrica em relação ao eixo (por questões de processamento numérico). Para um resultado mais fidedigno à teoria é necessário usar o comando `fftshift`, o qual centraliza o resultado da `fft`. Todavia, apenas a primeira metade dos pontos é inédita, sendo necessário plotar somente esta primeira metade para se ter informação sobre o espectro do sinal. O código abaixo obtêm o espectro sem centralizar, centralizado e apenas da parte positiva da frequência que corresponde a metade inédita.

```
freq = [-(L/2-1):L/2]*Fs/L;
pfreq = [0:L/2]*Fs/L;
subplot(3,1,1), plot(freq,abs(X))
subplot(3,1,2), plot(freq,abs(fftshift(X)))
subplot(3,1,3), plot(pfreq,abs(X(1:L/2+1)))
```

Experimente verificar o espectro de outros sinais. Note que o sinal simulado possui número par de pontos, portanto $L/2$ é inteiro. Se o tamanho do sinal for ímpar, então é melhor omitir a última amostra do sinal, por exemplo `x=x(1:length(x)-1);`, para evitar erros de indexação.

Para retornar um espectro para o domínio do tempo usa-se o comando `ifft`, como mostrado abaixo.

```
xnew = real(ifft(X));
```

3.1 Filtragem passa-baixa

Um filtro passa-baixa ideal elimina completamente altas frequências:

$$H_L^{ideal}(\omega) = \begin{cases} 1, & |\omega| \leq \alpha \\ 0, & |\omega| > \alpha \end{cases}$$

Porém, um filtro real possui $H_L(\omega)$ muito baixa para altas frequências, além de uma mudança gradual da magnitude entre as regiões de baixa e alta frequências. Um filtro passa-baixas simples pode ser implementado a partir da seguinte função de transferência:

$$H_L(\omega) = \frac{\alpha}{\alpha + j\omega},$$

onde α é a frequência de corte.

Este filtro pode ser implementado em Matlab usando a teoria da transformada de Fourier, i.e. multiplicação no domínio da frequência é igual a convolução no domínio do tempo.

$$y(t) = x(t) * h(t)$$
$$Y(\omega) = X(\omega)H(\omega)$$

O código abaixo implementa uma filtragem passa-baixa no sinal $x(t)$, criado no exemplo anterior. O filtro é definido pela equação de $H_L(f)$ (observe que a frequência agora está em Hz) acima e a frequência de corte é definida pela constante `a`. Por definição, o filtro é centralizado em relação à frequência $f = 0$ ou $\omega = 0$.

```
a = 1000;
H = a./(a+j*freq);
plot(freq,abs(H))
```

A figura mostra a resposta em frequência do filtro de acordo com a teoria. Porém é necessário descentralizá-la antes de multiplicá-la com a transformada de $x(t)$.

```
Hshift = fftshift(H);
Y = X.* Hshift';
plot(freq,abs(fftshift(Y)))
```

O sinal filtrado está no domínio da frequência e deve ser trazido de volta ao domínio do tempo usando o comando `ifft`.

```
y = real(ifft(Y));
```

3.2 Filtragem passa-alta

Um filtro passa-alta ideal elimina completamente baixas frequências:

$$H_H^{ideal}(\omega) = \begin{cases} 0, & |\omega| \leq \alpha \\ 1, & |\omega| > \alpha \end{cases}$$

Porém, um filtro real possui $H_H(\omega)$ muito baixa para baixas frequências, além de uma mudança gradual da magnitude entre as regiões de baixa e alta frequências. Um filtro passa-alta simples pode ser implementado a partir da seguinte função de transferência:

$$H_H(\omega) = 1 - \frac{\alpha}{\alpha + j\omega},$$

onde α é a frequência de corte.

Este filtro pode ser implementado de forma semelhante à do filtro passa-baixa. Experimente atenuar a frequência de 500 Hz de $x(t)$.

4 Roteiro

4.1 Aplicação dos filtros passa-baixa e passa-alta

1. Acesse a página da disciplina no PVAnet, baixe o arquivo *castanets44m.wav* e carregue-o.

2. Efetue filtragens passa-baixa e passa-alta, iniciando com `a = 500 Hz`. Tente também mais alguns valores de frequência de corte.

3. Ouça o sinal original e as versões filtradas do som. Plote também seus conteúdos espectrais e avalie os efeitos.

4.2 Eliminação de instrumentos musicais

A empresa X gravou uma sessão de trompete e bateria para seu novo lançamento. Porém, o chefe não agradou da bateria de fundo e decidiu que ela deveria ser retirada. Infelizmente, a mesa de mixagem foi programada erradamente e, ao invés de ter uma trilha diferente para cada instrumento, os sons foram misturados em apenas uma faixa. Um amigo seu, técnico de mesa de mixagem, pediu-lhe ajuda para eliminar a bateria da trilha, pois não havia tempo suficiente para trazer de volta o trompetista ao estúdio para a regravação. Use seus conhecimentos de processamento de sinais para salvar o emprego de seu amigo distraído.

1. Acesse a página da disciplina e baixe o arquivo *mixed.wav*. Este som foi criado a partir de *bassdrum.wav*, *hatclosed.wav* e *shake.mat*.

2. Use filtragens passa-alta e passa-baixa (lembre-se que um filtro passa-faixa é um passa-baixa e um passa-alta em cascata e um rejeita-faixa é um passa-baixa em paralelo com um passa-alta) e avalie o quanto consegue aproximar-se do som do trompete. Talvez seja interessante dar uma olhada no conteúdo espectral dos sons, mas não é permitido usar os sinais individuais na solução (por exemplo, subtraí-los no tempo ou frequência). Sinta-se a vontade para testar outras abordagens.

3. Tente agora eliminar o som do trompete, preservando a bateria.