# Princípios de Comunicação

2014/02

Profs. André Noll Barreto / Judson Braga

Universidade de Brasília

# Exercício de Simulação No 1

# Instruções

* A simulação poderá ser feita em Matlab (recomendado), Octave, Scilab, Python ou C++.
* A simulação deve ser entregue **sob a forma de relatório em formato pdf**, no qual conste:
  + descrição do que foi feito e observado, conforme as especificações;
  + gráficos ou tabelas com resultados;
  + deve ser enviado também arquivo com código fonte em anexo.
* Os relatórios podem ser entregues até o dia 10/11/2014
* O trabalho pode ser feito individualmente ou em grupos de até três alunos.
* Cópias de trabalhos entre grupos diferentes não serão toleradas.
* **Tomem cuidado com o português e o estilo do texto.** Estes aspectos também serão avaliados.

# Parte 1 – Compreendendo a DFT

Em processamento digital de sinais trabalhamos com sinais amostrados em tempo discreto e com duração finita. Neste caso, a transformada de Fourier é aproximada pela Transformada de Fourier Discreta (DFT), ou, pela sua implementação mais eficiente, a FFT (Fast Fourier Transform). Leia a Seção 3.9 do Lathi, 4ª Edição, na qual é explicada a DFT.

Algumas características importantes da DFT

* Considerando um sinal com duração *T*0 e intervalo de amostragem *Ta*, temos amostras. No domínio da frequência teremos as mesmas *N* amostras, com um intervalo entre as frequências.
* A DFT é periódica tanto no tempo quanto na frequência, com período *N*.
* Uma convolução circular no tempo representa uma multiplicação no domínio da frequência.

Com isto em mente, realize o seguinte exercício.

1. Gere um sinal , com Hz e ,5Hz. Como no computador se trata de um sinal em tempo discreto, considere uma taxa de amostragem de 20 Hz e represente o sinal em intervalo de 5s. Plote o sinal no tempo.
2. Calcule sua DFT [[1]](#footnote-1) e plote sua amplitude e fase (lembre-se que a DFT é complexa), tomando cuidado em utilizar a referência de frequências correta. No que o gráfico da DFT difere do esperado para a Transformada de Fourier? Por quê?
3. Repita os itens 1 e 2, representando agora o sinal em um intervalo de 20s. O que mudou na DFT e por quê?
4. Mantendo o intervalo de 20s, repita os itens 1 e 2 com taxas de amostragem de 5 e 2Hz. O que mudou na DFT e por quê?

Podemos realizar filtros tanto no domínio do tempo quanto no domínio da frequência. Vamos aplicar uma filtragem passa-baixa no sinal acima, com frequência de corte de 1,2Hz. Considere os parâmetros do item 3 acima.

1. Realize a filtragem no domínio da frequência. Multiplique o sinal no domínio da frequência pela resposta do filtro e faça a DFT inversa. Plote o sinal filtrado no tempo, assim como seu espectro.
2. Realize a filtragem no domínio no tempo, fazendo a convolução do sinal com a resposta impulsional desejada[[2]](#footnote-2). Plote novamente o sinal filtrado no tempo, assim como seu espectro. O que podemos notar no sinal com as duas abordagens?

# Parte 2 - Densidade Espectral de Potência e Filtros

1. Importe no Matlab uma imagem qualquer em formato bmp, e a converta num vetor[[3]](#footnote-3). Supondo uma taxa de transmissão de 1000 pixel/s, plote o sinal resultante como função do tempo.
2. Estime e plote sua autocorrelação (com 512 amostras)[[4]](#footnote-4), atentando para a referência de tempo correta.
3. A partir da autocorrelação, e usando a FFT estime e plote sua D.E.P. (em dB e linear). Novamente, atente para a referência certa de frequência.
4. Estime a largura de banda essencial do sinal (de 95% da potência)
5. Filtre o sinal em um passa-baixa de frequência de corte igual à sua largura de banda essencial, e mostre a figura correspondente ao sinal filtrado[[5]](#footnote-5).
6. Acrescente ruído branco Gaussiano com RSR de 3 e 10 dB[[6]](#footnote-6), plote a D.E.P. com ruído e visualize a diferença em ambos os casos.
7. Para o sinal ruidoso, Implemente um filtro de Wiener, e visualize de novo o resultado filtrado, plote a resposta na frequência do filtro e a D.E.P. na saída do filtro.
   1. o filtro de Wiener é dado por . Com base na D.E.P do sinal e do ruído, ache H(f), e faça a filtragem na frequência ou no tempo.

# Parte 3 – Modulação Analógica

1. Utilizando o sinal gerado no item (1) da Parte 2, gere um sinal modulado em DSB-SC[[7]](#footnote-7). Utilize uma frequência de portadora igual a 4 vezes a banda essencial *Be* do sinal.
   1. Explique como foi gerado o sinal modulado.
   2. Plote o espectro do sinal modulado.
   3. Demodule o sinal e mostre a imagem resultante.
2. Agora, utilizando uma outra imagem de mesmo tamanho, multiplexe-as na frequência, considerando uma delas enviada em uma portadora na frequência 4 *Be* e a outra na frequência 6 *Be* , ambas com DSB-SC
   1. Plote o espectro do sinal multiplexado.
   2. Demultiplexe os sinais e mostre as imagens resultantes.
3. Repita o item (2), considerando frequências de 4 *Be* e a outra na frequência 8 *Be.* O que podemos observar e por quê?

1. No Matlab use a função fft [↑](#footnote-ref-1)
2. Lembre-se que para que o filtro seja causal devemos aplicar um atraso à resposta impulsional e truncá-la em t<0. [↑](#footnote-ref-2)
3. Uma imagem em bitmap contém 3 matrizes, cada uma representando os pixels em uma cor (RGB), ou seja, temos um tensor de 3 dimensões, que pode ser convertido em um vetor com a função reshape. [↑](#footnote-ref-3)
4. Pode ser usada a função xcorr. [↑](#footnote-ref-4)
5. Ao se filtrar o sinal, cuidado que o filtro pode introduzir atrasos ou mudar a amplitude do sinal, o que pode impossibilitar a visualização. Estes efeitos têm que ser compensados. Além disso a imagem bmp exige valores inteiros entre 0 e 255 para cada pixel. [↑](#footnote-ref-5)
6. A função randn gera um vetor gaussiano com média 0 e variância 1. Multiplicando-se por , temos um ruído de variância (e potência) igual a 2. [↑](#footnote-ref-6)
7. Será necessário super-amostrar o sinal, lembrando que a frequência de amostragem deve satisfazer a taxa de amostragem de Nyquist, com uma margem de segurança (considere 50% a mais). [↑](#footnote-ref-7)