INSTITUTO FEDERAL DO ESPIRITO SANTO

Curso de Engenharia Elétrica

ANÁLISE DE SINAIS E SISTEMAS - 2023-1

PRÁTICA 4

1 – Traçar sinais discretos para n= 0 até 10. Use o comando <stem> para plotar a função.

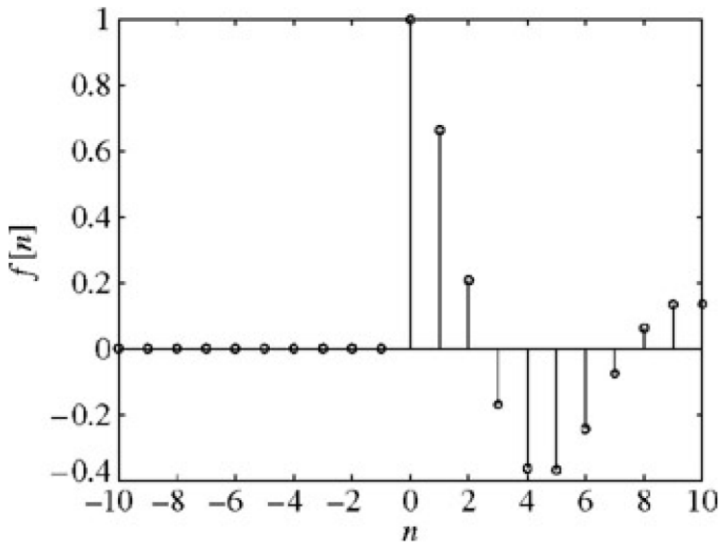
$$xa[n] = \cos\left(\frac{n\pi}{12} + \frac{\pi}{4}\right)$$

2 – O exemplo abaixo (C3.3 do Lathi), que implementa uma equação das diferenças utilizando Interatividade, viabiliza o cálculo dos valores de Saída de (y[n]) de uma equação das diferenças. Utilize esse exemplo como referência e implemente a equação $y[n] = x[n-1] - 2y[n-1]$ para a entrada x[n]=u[n] e y[0]=1.

NOTA: A equação citada acima é a MESMA que figura no arquivo: **"ASS_2023_1 Teoria 11 - Resposta do Sistema Discreto - Exercício Resolvido - Vrs01.pdf"** postado no AVA (Moodle).

```
n = (0:10)';
y=[4;13;zeros(length(n)-2,1)];
x = (3*n+5).*(n>=0);
for k = 1:length(n)-2
    y(k+2) = 5*y(k+1) - 6*y(k) + x(k+1) - 5*x(k);
end;
clf;
stem(n,y,'k');
xlabel('n');
ylabel('y[n]');
disp(' n        y');
disp([num2str([n,y])]);
```

3 – Utilize o comando <**inline**> e implemente a função discreta $f[n] = e^{-n/5}\cos\left(\frac{n\pi}{5}\right)u[n]$ afim de obter EXATAMENTE o gráfico abaixo:



4 – Utilizando o comando <residuez> (semelhante ao comando <residue> já ensinado) e faça a Expansão em Frações Parciais da Função: $H(z) = \left(\frac{1 - 2z^{-1}}{1 + 3z^{-1} + 2z^{-2}} \right)$.

5 – Os sinais discretos podem ser transformados para o "domínio da frequência", pela Transformada de Fourier no Tempo Discreto (DT FT)

Logo: a transformada de tempo contínuo

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-i2\pi ft}\, dt.$$ se transforma em

$$X_T(f) \stackrel{\text{def}}{=} \sum_{k=-\infty}^{\infty} X\left(f - kf_s\right) \equiv T \sum_{n=-\infty}^{\infty} x(nT)\, e^{-i2\pi fTn}.$$

Onde:

$$\omega = 2\pi fT = 2\pi \left(\frac{f}{f_s} \right).$$

Mas... NÃO É POSSÍVEL se obter, em um software, um **somatório infinito**, então a Transformada de Fourier no Tempo Discreto (DTFT) **foi modificada** para ser a Transformada Discreta de Fourier (DFT) que é finita.

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-i2\pi \frac{k}{N} n}$$

No Matlab essa função é implementada pela Transformada Rápida de Fourier (FFT) que é uma versão que substitui o tradicional, e lento, Loop por um algoritmo chamado de "Butterfly".

A função <fft>, então, é utilizada para fazer a Transformada de Fourier computacional de um vetor temporal de dados como pode ser visto no exemplo abaixo:

```
clc; clear all; close all
%% Senoides e aliasing
% Taxa de amostragem
fs=1000; Ts=1/fs;
% Frequencias, em Hz, do sinal
f1=20; DigFreq1=2*pi*f1/fs;
f2=30; DigFreq2=2*pi*f2/fs;
f3=40; DigFreq3=2*pi*f3/fs;
%Plot
N=1500;
```

```
n = 0:1:N-1;
t_sample = [0 : Ts : (N-1)*Ts];
x=3.*cos(DigFreq1.*n)+cos(DigFreq2.*n)+2.*cos(DigFreq3.*n);
figure(1);
plot(t_sample,x)


%% FFT do sinal
X=fft(x);
% os valores de x são valores reais, enquanto os valores de X são
% complexos. Veja alguns exemplos:
x(2:6)
X(30:34)
%X são valores complexos porque representam magnitude e fase!!!!
%Magnitude
X_mag=abs(X);
X_mag(30:34)
figure(2)
plot(X_mag)
```

Porém, o ESPECTRO DE FOURIER aparece na faixa de 0 a N [Hz], e essa é uma "aparência" complicada de se analisar, logo utiliza-se a função `<fftshift>` para colocar a frequência ZERO no centro do Gráfico, conforme mostra o algoritmo abaixo:

```
 clear all; close all; clc
%%
Tmax=0.5; % Intervalo de duração de cada onda
fs=200; % Frequência de amostragem
t=[0:(1/fs):Tmax+2]; % Amostragem no tempo
L=length(t);
% Pulso retangular
T0=0; % Instante de início do pulso retangular
T=Tmax; % Duração do pulso retangular
% Definição do início e final da janela
L_ini=length([0:(1/fs):T0]);
L_pulse=length([0:(1/fs):T]);
L_fin=L-L_ini-L_pulse;
win = rectwin(L_pulse);
wRect1 = [zeros(L_ini,1); win; zeros(L_fin,1)]';
figure(1)
subplot(311)
plot(t,wRect1,'LineWidth',1)
grid on
xlabel('Tempo(s)');
ylabel('Amplitude');
X=fft(wRect1);
subplot(312)
plot(t,abs(X))
```

```
grid on
xlabel('bins');
ylabel('Amplitude');
Y=fftshift(X);
subplot(313)
plot(t,abs(Y))
ylim([0 100])
grid on
xlabel('bins');
ylabel('Amplitude');
```

Porém, o EIXO da frequência não condiz com os valores reais de frequência amostrados, logo o algoritmo abaixo demonstra como ajustar o eixo da frequência para mostrar a frequência real do espectro:

```
Fs = 1000;                    % Sampling frequency
T = 1/Fs;                     % Sampling period
L = 1000;                     % Length of signal
t = (0:L-1)*T;                % Time vector
%%Create a matrix where each row represents a cosine wave with scaled
frequency. The result, X, is a 3-by-1000 matrix. The first row has a wave
frequency of 50, the second row has a wave frequency of 150, and the third
row has a wave frequency of 300.

x1 = cos(2*pi*50*t);          % First row wave
x2 = cos(2*pi*150*t);         % Second row wave
x3 = cos(2*pi*300*t);         % Third row wave

X = [x1; x2; x3];
%%Plot the first 100 entries from each row of X in a single figure in order
and compare their frequencies.
figure(1)
for i = 1:3
    subplot(3,1,i)
    plot(t(1:100),X(i,1:100))
end


dim = 2;
%%Compute the Fourier transform of the signals.

Y = fft(X,L,dim);
%%Calculate the double-sided spectrum and single-sided spectrum of each
signal.

P2 = abs(Y/L);
```

```
P1 = P2(:,1:L/2+1);
P1(:,2:end-1) = 2*P1(:,2:end-1);

%%In the frequency domain, plot the single-sided amplitude spectrum for each
row in a single figure.
figure(2)
for i=1:3
    subplot(3,1,i)
    plot(0:(Fs/L):(Fs/2-Fs/L),P1(i,1:L/2))
end

Fs = 100;
t = 0:1/Fs:1-1/Fs;
f1=15; % Hz
f2=40; % Hz
x1 = cos(2*pi*f1*t - pi/4);
x2 = cos(2*pi*f2*t + pi/2);
x = x1+x2;

%%Compute the Fourier transform of the signal. Plot the magnitude of the
transform as a function of frequency.
y = fft(x);
z = fftshift(y);
figure(3)
ly = length(y);
f = (-ly/2:ly/2-1)/ly*Fs;
subplot(2,1,1)
plot(t,x)
subplot(2,1,2)
stem(f,abs(z))
title('Double-Sided Amplitude Spectrum of x(t)')
xlabel('Frequency (Hz)')
ylabel('|y|')
Grid
```

De posse de todos esses dados, some as frequências 60, 120 e 180 Hz e mostre, fazendo uso do algoritmo anterior, o espectro de frequência do sinal final.