Aula 13 e 14

Ana Gonçalves e Hélio Neto
Universidade Federal de Minas Gerais

I. Questão 1

A DFT (Transformada Discreta de Fourier) é implementada no código abaixo
As ondas senoidais e cossenoidais usadas na DFT são comumentemente chamadas de funções base da DFT. A saída da DFT é um conjunto de números que representam as amplitudes destes senos e cossenos.

```
clc; clear; close all
% gera senoide
for n=0:20
    x(n+1) = 2*sin(n*pi/2) + 0.5*sin(n*pi/4);
end

% calcula dft
N = length(x);
for k=0:N-1
    for n = 0:N-1
        X(n+1) = x(n+1)*exp(-j*2*pi*k*n/N);
    end
    Xk(k+1)=sum(X);
end

%sinal
subplot(2,1,1)
stem(0:N-1, x);
legend('|x_n|');
xlabel('tempo, n = 0 ... N');
ylim([-3 3]);

% magnitude do espectro
subplot(2,1,2)
mag =  abs(Xk);
stem(0:N-1, mag);
legend ('|X_k|');
xlabel('\omega, k = 0 ... N');
```

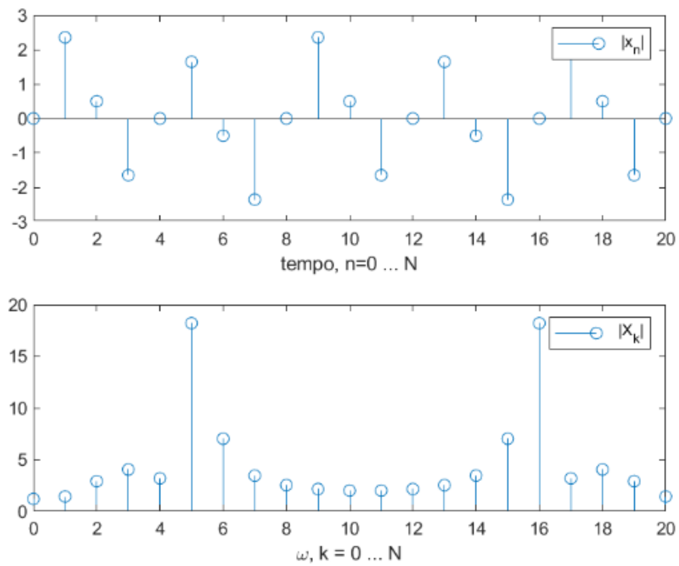

Figura 1: *Gráfico*

a) Qual a equação executada nas linhas 9-14? Observe o cálculo com variável complexa na forma exponencial (polar).
A equação designada entre as linhas 9 e 14, referem-se à transformada de fourier:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N} \tag{1}$$

para k = 0,...,N-1
Por sua vez, a transformada de Fourier descreve apropriadamente sinais periódicos ou não periódicos. Note que os sinais não periódicos podem ser vistos como sinais periódicos de período infinito.

b) Na linha 9, qual a exigência para que o sinal gerado (linhas 3-4) possa ser recuperado a partir do espectro?
O sinal x[n] deve ser absolutamente integrável; ou seja, deve ser tal que o somatório da equação 1 não seja divergente.

II. Questão 2

O código do próximo slide é uma versão mais elaborada do código usado na parte 1. São usados algoritmos fft() do Matlab para implementar a DFT. Também é possível ouvir o som gerado e salvar arquivos .wav (para isso, retire os comentários nas linhas comentadas 17 e 18).

```
1  clc; clear; close all
2  %________________________________
3  % Criacao de sinal e arquivo .wav
4
5  fs = 22050;
6  T = 1;
7  A1 = 1;
8  A2 = 0.5;
9  f0 = 440;
10 Δ_f = 800;
11
12 N = fs*T;
```

```
Ts = 1/fs;

t = 0:Ts:T;
x = A1*cos(2*pi*f0*t)+A2*cos(2*pi*f0*t)+A2*cos(2*pi*(f0+Δ_f)*t);
sound(x1, fs);
audiowrite('seno.wav', x1, fs, 'BitsPerSample', 16);

%________________________________________
%janela o sinal em L pontos no tempo

L = 512;
x1 = x(1:L);
subplot(4,1,1);
n = 0:L-1;
t = n*Ts*1000;

%________________________________________
%janela para suavizar as bordas
h = hamming(L)';

x1h = h .* x1;
plot(t,x1,'r',t,x1h,'b');
xlabel('t, ms');
title(['L=', num2str(L), 'amostras, ', ...
'\Delta_f=', num2str(Δ_f), 'Hz,', ...
'f0=',num2str(f0),'Hz']);
legend('original', 'janelado');
%________________________________________
%Modulo de espectro (par)

W = 1024;
%comparar com o espectro do sinal nao janelado
%ver a atenuacao das franjas as custas de um
%alargamento dos lobulos nas frequencias de interesse
Xmod = abs(fft(x1, W));

Xmod1 = Xmod(1:W/2);

subplot(4, 1, 2);
k = linspace(0, fs/2, W/2);
plot(k, Xmod1/max(Xmod1));
ylabel('linear, ua');
%observe as franjas devido ao janelamento
%aumentando a janela temporal L, as franjas diminuem

subplot(4, 1, 3);
semilogx(k, 20*log10(Xmod1/max(Xmod1)));
title(['X1, Amostragem do espectro fs/W=', num2str(fs/W), 'Hz']);
ylabel('dB ref FS');
xlim([10e1 10e3]); ylim([-60 10]);

subplot(4, 1, 4);
Xmod = abs(fft(x1h, W));
Xmod1 = Xmod(1:W/2);

semilogx(k, 20*log10(Xmod1/max(Xmod1)));
ylabel('db ref FS');
title(['X1h (janelado)']);
xlabel('frequencia, Hz');
xlim([10e1 10e3]); ylim([-60 10]);

%observe a separacao das duas senoides
%dado uma janela temporal L, aumentar W nao melhora
%a resolucao espectral
```

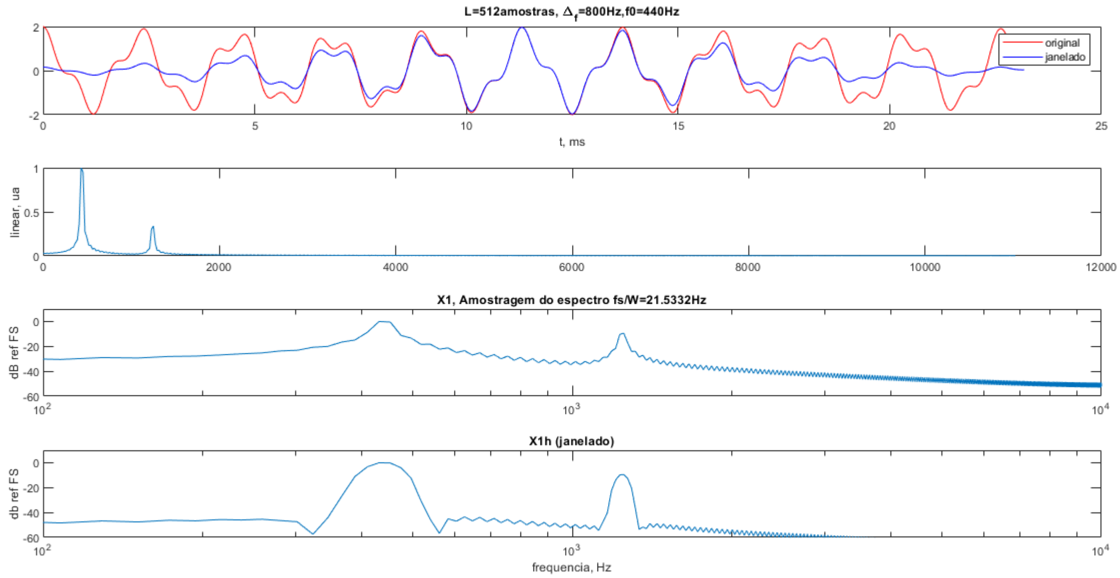

Figura 2: *Gráfico 2*

a) O sinal a ser analisado é ponderado por uma "janela de Hamming" que não é retangular, mas tem as bordas suavizadas. Explique a diferença nos gráficos do subplot(4,1,1);
Observa-se que, com o janelamento do sinal em questão, seu formato no domínio do tempo foi alterado em virtude da convolução com a janela de Hamming. Assim, as amostras que correspondem à faixa de frequências indesejadas tiveram amplitude atenuada.
Percebe-se que, dependendo do processamento, chega-se a diferentes valores de frequências e níveis de distorção devido as harmônicas de ordem superior.

b) Os espectros gerados podem estar em escalas lineares [subplot(4,1,2)] e log-log [subplot(4,1,3) e subplot(4,1,4)]. Execute o código, comparando duas durações L da janela na linha 23 (L = 124 e L = 1024), explicando o que ocorre nos espectros.
Com o menor comprimento da janela, mais amostras do sinal são atenuadas em relação à frequência de interesse. Assim, observa-se, por exemplo, pelo gráfico do espectro normalizado (4,1,2), a atenuação das franjas quando L = 124. Fica explícito, portanto, que quanto maior o comprimento da janela, maior a resolução espectral do sinal tratado, o que faz a largura do lóbulo principal diminuir enquanto a dos lóbulos secundários aumenta.

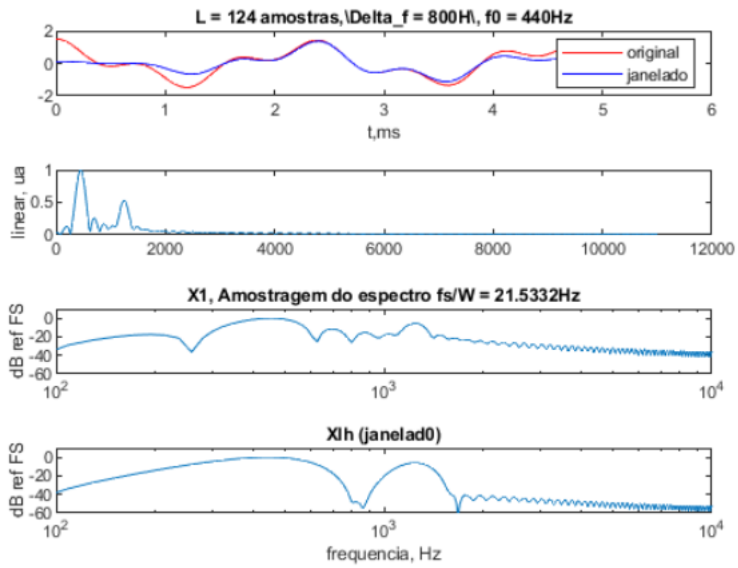

Figura 3: *Gráfico*

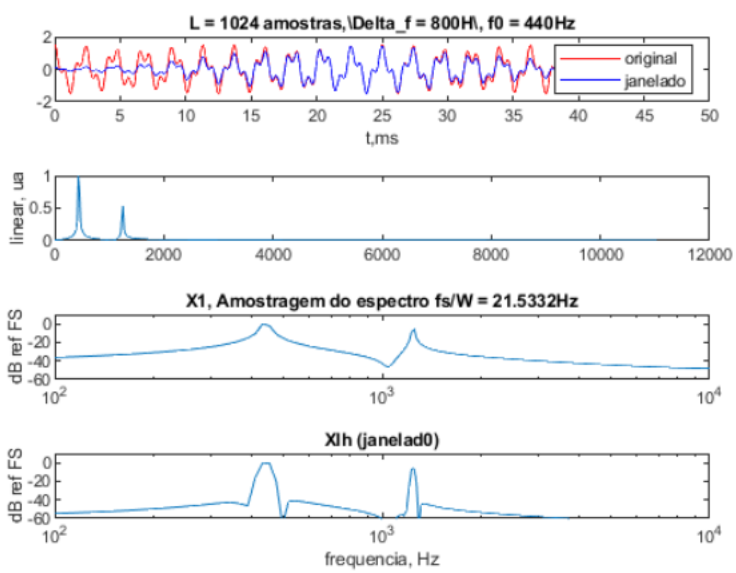

Figura 4: *Gráfico*

c) Com L = 1024, qual a capacidade de resolução em frequência?

Como, com o janelamento, o valor das amostras se reduz, a relação para resolução torna-se:

$$\Delta f = \frac{f_s}{L} = \frac{22050}{1024} = 21,5Hz \tag{2}$$

d) Justifique (e verifique experimentalmente) a afirmativa nas linhas 74-76.

A resolução espectral caracteriza a capacidade do sensor em operar em várias e estreitas bandas espectrais.

Para melhorar a resolução espectral, o que se pode fazer é aumentar o número de ciclos para uma mesma taxa de amostragem. Sendo assim, o valor do módulo do espectro (W) não interfere na resolução em frequência. Isso é facilmente percebido sob uma abordagem matemática, uma vez que a equação $\Delta f = \frac{f_s}{L}$ independe de W.

e) Qual a maior frequência que pode ser sintetizada no código?
À partir do gráfico normalizado do espectro de frequências, observa-se que a maior frequência sintetizada para o código com o sinal em questão é de 441,862Hz; ou, aproximadamente 442 Hz.

III. Questão 3

Impulso. Explique o que é feito no código ao lado, que é fragmento de um código maior. Nas linha 15 (e 23) são utilizadas as funções ifft (e fft) para realizar o cálculo rápido da IDFT (e DFT).

```
1  clc; clear; close all
2  %***********************************
3  %espectro no plano, fase 0
4  % impulso no tempo
5  N = 1024;    % tamanho do sinal e da DFT/FFT
6  
7  for k = 1:N
8      Xmod(k) = 1; % 0 ... 2pi
9      Xfas(k) = 0  % 0 ... 2pi
10     X(k) = Xmod(k)*exp(j*Xfas(k));
11 end
12 
13 subplot(4,2,1);
14 x = ifft(X);
15 plot(x);     % impulso no tempo
16 xlabel('n');
17 xlim([-20 N/2]);
18 ylim([0 1.2]);
19 title('impulso unit[U+FFFD]o');
20 
21 subplot(4,2,2);
22 Xmod = 20*log10(abs(fft(x)));
23 plot(Xmod);
24 title('|X_k|dB, branco');
25 xlabel('k'); xlim([0 N/2]); ylim([-60 5]);
26 
27 %espectro plano, fase aleatoria
28 %sinal aleatorio no tempo
29 N = 1024;
30 
31 for k=1:N/2
32     Xmod1(k)=1;
33     Xfase1(k)=2*pi*random('Norm',0,1)-pi;
34 end
35 
36 %sintaxe compacta, com recursos do Matlab
37 %    0...pi    pi...2pi
38 Xmod=[Xmod1 Xmod1(end-1:-1:2)];
39 %    0...pi    pi...2pi
40 Xfase=[Xfase1 -Xfase(end-1:-1:2)];
41 
42 
43 X = Xmod.*exp(j*Xfase);
44 
45 subplot(4,2,3);
```

```
46 x=real(ifft(X));
47 %imprecisoes numericas => x_im nao nulo (pequeno)
48
49 x=x/max(abs(x))*0.98;
50
51 plot(x);
52 xlim([0 N/2]); ylim([-1.2 1.2]); xlabel('n');
53 title('ruido branco');
54
55 subplot(4,2,4);
56 Xmod=abs(fft(x));Xmod=20*log10(Xmod/max(Xmod));
57
58 plot(Xmod);title('|X_k|db branco');
59 xlabel('k');xlim([0 N/2]);ylim([-60 5]);
```

a) Quais as condições para usar algoritmos de transformada rápida?

A Transformada Rápida de Fourier (FFT) exige que o número N de amostras consideradas seja uma potência de 2. Além disso, o sinal de entrada deve ser definido como uma variável complexa.

FFT é simplesmente uma forma mais rápida de calcular a DFT: A FFT utiliza alguns algoritmos que permitem reduzir o número de operações para Nlog_2N. Para utilizar a FFT, é necessário que o número de amostras seja uma potência de 2 – a FFT é executada mais rapidamente com um vetor cujo comprimento é uma potência de 2. Para N = 1000, DFT = 1 000 000, FFT = 10 000 operações.

A FFT no Matlab: Matlab permite o cálculo fácil da DFT via FFT. Se tivermos um vetor A, de n elementos,

$$FFTdeA = fft(A);$$

b) O que é realizado nas linhas 7 -11?

As linhas 7-11 descrevem o sinal constante no domínio da frequência, que X(jw) = 1, corresponde ao impulso unitário no domínio da frequência.

c) Se que a frequência de amostragem for 1 kHz, em que intervalo o espectro é amostrado?

Com N = 1024 sendo o número de amostras, para 1kHz como frequência de amostragem, o intervalo de amostragem do espectro é tido por:

$$\Delta f = \frac{fs}{N} = \frac{1000}{1024} \approx 0{,}977Hz \tag{3}$$

d) Neste exemplo específico, como a linha 10 poderia ser simplificada?

Como X_{mod} é sempre igual à um e X_{fas} é sempre um valor nulo, então, a exponencial $X_{mod} \times e^{X_{fas} \times j}$ será igual a $1 \times e^0$. Assim, a linha 10 poderia ser simplificada escrevendo-se apenas: X(k) = 1;

e) Expresse, matematicamente, o que é feito na linha 22.

$$X_{mod} = 20 \times log(|X(j\omega)|) \tag{4}$$

IV. Questão 4

4) Ruído branco. O uso de impulsos pode ser impraticável (em acústica, danificaria alto-falantes, por exemplo). Isto leva ao uso de outros sinais para caracterização de sistemas (acústica de sala, por exemplo). O código ao lado gera um sinal com espectro plano mas com fase aleatória. O resultado, é conhecido como ruído branco. Digite o código e ilustre o resultado [pode ser necessário limpar as variáveis no início do código: clear all]

a) O que é feito nas linhas 32-34? Utilize o comando help para entender a função random();

É descrito um sinal aleatório com igual intensidade para diferentes frequências, a aleatoriedade é definida pela função random() que utiliza distribuição normal.

b) Nas linhas 39 e 41 é usada uma notação compacta do Matlab. Explique o é feito nas linhas 37-42 e justifique, em termos de simetrias espectrais par ou ímpar.

A linha 39 serve para concatenar os valores de X_{mod} de 0 a com o seu valor oposto(ou seja, de π a 0) gerando um sinal entre 0 e 2π com simetria par; o mesmo é feito para na linha 41, porém com os valores negativos, de 0 a π, e seus valores opostos (ou seja, de π a 0), o que cria um sinal entre 0 e 2π com simetria ímpar.

c) A linha 48 alerta para imprecisões numéricas. Teoricamente, qual seria o valor da parte imaginária de x (linha 47)?

$\text{sen}(\Gamma^{-}1X_{mas}) = 0.0016$, para o programa executado (varia de caso a caso pois é randômico), é o valor da parte imaginária de X.

d) O que é feito nas linhas 56-60?

As linhas 56-57 calculam a amplitude e a defasagem do sinal normalizado.

As linhas 59-60 plotam o gráfico, delimitam ele e definem título.

e) Como este sinal (ruído branco) pode ser usado na estimativa da resposta impulsiva de uma sala, por exemplo?

Em amplificadores no geral, é interessante que a relação-sinal-ruído seja a melhor possível. Assim, a medição da resposta ao impulso do ruído-branco permite conhecer seu valor com mais precisão, de modo que a frequência do sinal amplificado não torne-se a mesma que a do ruído em módulo.

V. Questão 5

5) Varredura de senos. Este é um sinal variante no tempo, onde a frequência cresce linearmente como ilustrado ao lado. São apresentados dois conceitos novos:

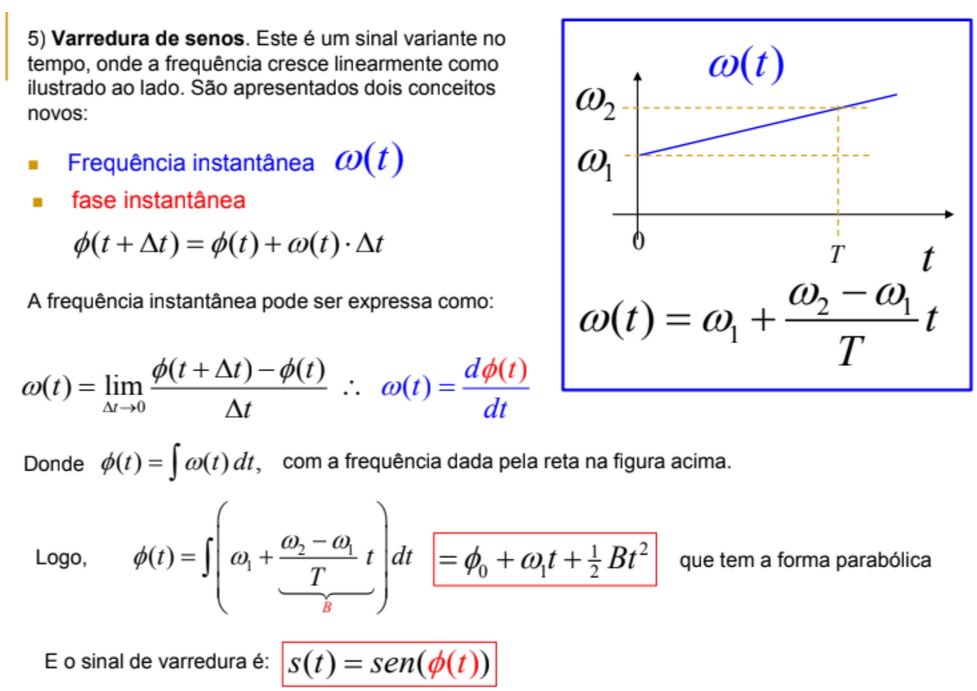

Figura 5: *Explicação*

O código ao lado sintetiza um sinal de varredura de senos com uma taxa de amostragem fs (linha 10). Você poderá

ouvir o som gerado (linha 34) e também salvar um arquivo .wav (linha 35).

```
1 % Varredura linear de senos - sintese no tempo
2 % w(t) = w1 + ((w2-w1)/T)*t
3 % B = (w2-w1)/T
4 % T = duracao do sinal
5 % W1 - frequencia inicial
6 % phi(t) = phi0 + w1*t + (B/2)*t^2
7 clc; clear; close all
8
9 fs = 22050;
10 phi0 = 0;
11 T = 2;
12 w1 = 2*pi*20;
13 w2 = 2*pi*fs/2;
14 B = (w2-w1)/T;
15 N = T*fs;
16 Ts = 1/fs;
17
18 %laco 'for' pode ser escrito de forma compacta
19 for n=0:(N-1)
20     phi(n+1) = phi0 + w1*(Ts*n) + (B/2)*(Ts*n)^2;
21     s(n+1) = sin(phi(n+1));
22 end
23
24 subplot(2,1,1);
25 t = linspace(0,T,N); plot(t,s); title('s(t)');
26 xlim([0, 0.2]);
27 xlabel('tempo(s) (truncado para visualizacao)');
28
29 subplot(2,1,2);
30 spectrogram(s,128,120,128,fs,'yaxis');
31 title('Varredura linear de senos (TSP)');
32
33 sound(s,fs);
34 audiowrite('TSP.wav',s,fs,'BitsPerSample',16);
```

a) Quais as frequências (em Hz) inicial e final do sinal gerado?
Frequência inicial: 20 Hz ou 125.663706 rad/s
Frequência final: 11.025 Hz ou 69272.117932 rad/s

b) complete as linhas 21 e 22 para executar o código. Por que o índice (n+1) é usado?
Usa-se o índice n+1 uma vez que o Matlab indexa valores à partir do 1 e não do 0, como a maioria das linguagens de programação.

c) as saídas a serem obtidas estão ilustradas no próximo slide.

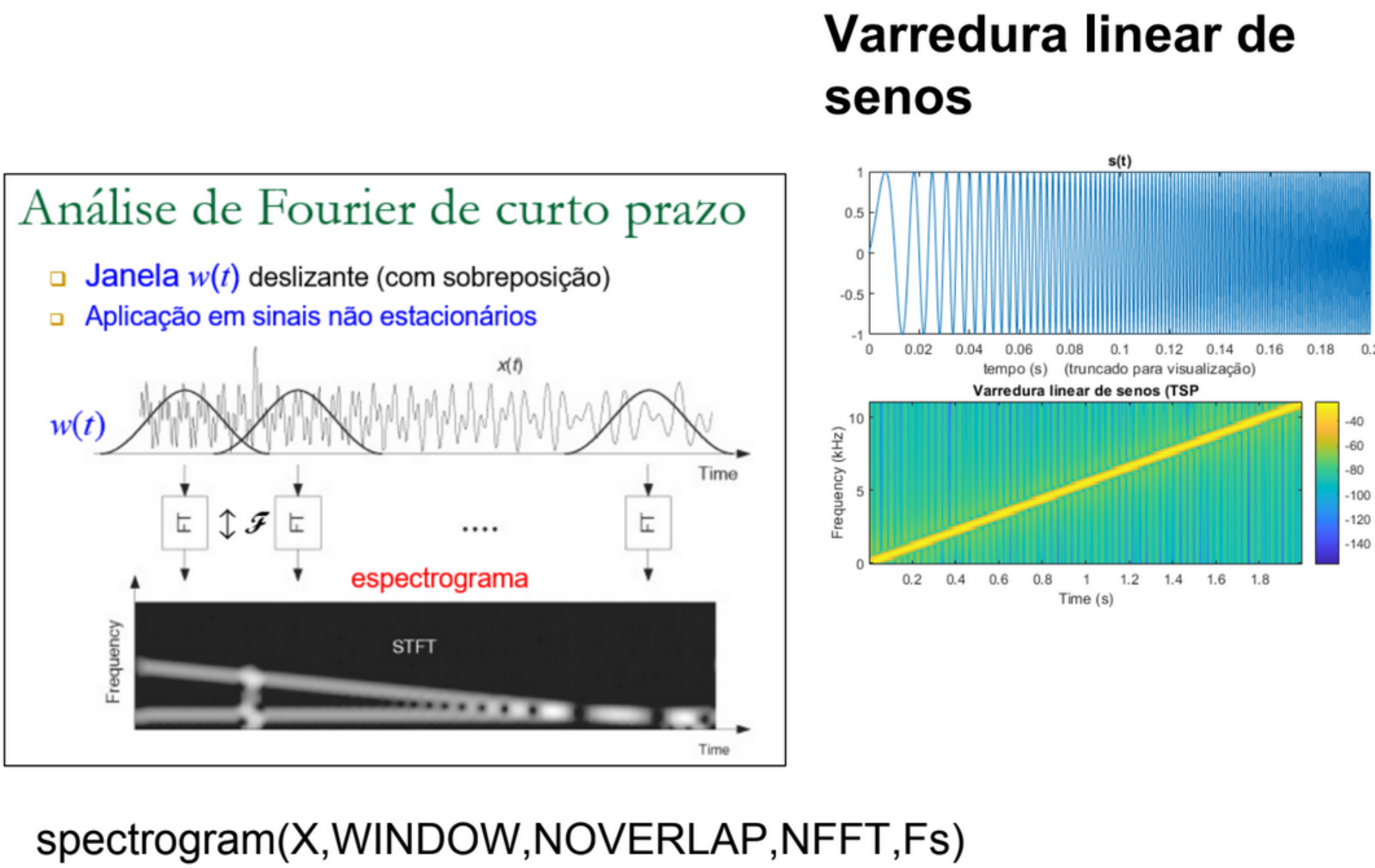

Figura 6: *Explicação*

d) O subplot(2,1,2) é um espectrograma. Como o sinal é variante no tempo, o espectrograma é gerado fazendo-se análises de Fourier em blocos (janelas) relativamente pequenas, como ilustrado no slide seguinte. Com a ajuda do [help], explique os parâmetros usados na linha 31, que tem a forma: spectrogram(X,WINDOW,NOVERLAP,NFFT,Fs)

X:sinal de entrada;
Window: tamanho da janela de Hamming;
Noverlap: número de amostras sobrescritas;
nfft: número de pontos da transformada discreta de Fourier;
fs: taxa de amostragem