
```

rng(467)

L = randi(500)+100; % Longitud de la señal.
I = fix((rand*6+2)*10)/10;
Omega = pi/I;
% K determina el número de sinusoides. Se escoge al azar entre 15 y
35.
K = randi(20)+15;

% Diseño de la señal.
x = PassBandSig_4_DSP(L,Omega,K); % Diseño de la señal.

%Representación de la señal x[n] en el dom del tiempo
figure(1)
n = 0:L-1;
subplot(311), stem(n,x)
xlabel('n')
ylabel('x[n]')
title(['Señal en el dominio del tiempo: L = '...
num2str(L) ' y K=' num2str(K) '.'])
xlim([n(1) n(end)])
subplot(312), plot(n,x)
xlabel('n')
ylabel('x[n]')
title('x[n] representada con plot.')
xlim([n(1) n(end)])

% calculo y representación del espectro de la señal.

N = min([2^ceil(log2(L)+1) 2048]); % Resolución para el cálculo de la
DFT.
X_w = abs(fftshift(fft(x,N))).^2/L; % Cálculo del periodograma.
F = linspace(-1,1,size(X_w,2));
figure(1)
subplot(313), plot(F,X_w)
xlim([F(1) F(end)]), grid on
xlabel('Frecuencia normalizada ($F = \Omega/\pi$)',...
'Interpreter','latex','FontSize',13)
ylabel('$\left\vert X\right\vert(e^{j\Omega}) \right\vert$',...
'Interpreter','latex','FontSize',13)
title(['Representación del espectro de x[n]: F_0 = '...
num2str(Omega/pi,'%2f') '.'])

function [x] = PassBandSig_4_DSP(L,Omega_0,K)
% This function designs a signal as a combination of K+1
% sinusoids. The spectral content gathers a set of pure
% frequencies inside the interval around the fundamental
% frequency (Omega_0). The bunch of sinusoids are defined
% to be random frequencies in a small range very close to
% the fundamental one.

```

```

%Dominio de definición
ak=randn*0.02;
n=0:(L-1);
n=0:L-1;
x=cos(Omega_0*n);
alphaK=(0.02)*randn(1,1);

for k=1:K
    sigma_k=(1+(k/25))^-1;

    A1=sigma_k*randn(1,1);
    A2=sigma_k*randn(1,1);

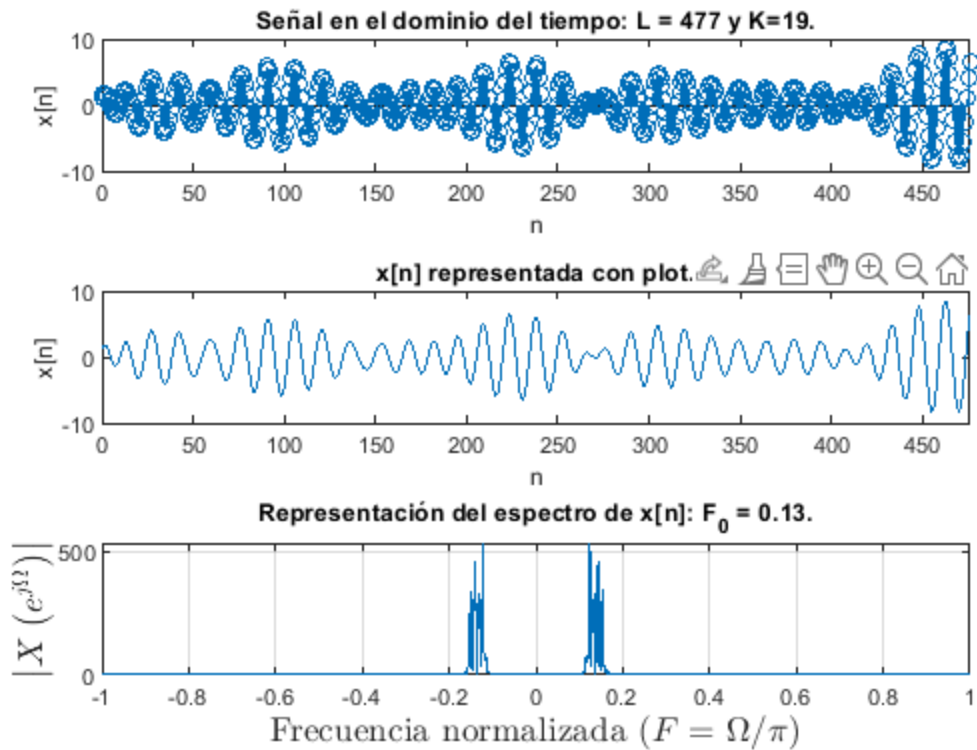
    Phi1=rand*(2*pi-pi);
    Phi2=rand*(2*pi-pi);

    Omega_K=Omega_0*(1+(alphaK*k));

    x1=cos((Omega_K*n)+Phi1);
    x2=cos((((Omega_0)^2)/Omega_K)*n)+Phi2);

    x=x+A1*x1+A2*x2;
end
end

```



Published with MATLAB® R2020b