P4

**Ejercicio 1**

function [t, x]= sinus2(M, A, f0, Ts)

t = [0:Ts:M\*Ts];

x = A\*sin(2\*pi\*f0.\*t);

end

[t,x] = sinus2(100,1,0.2,0.1)

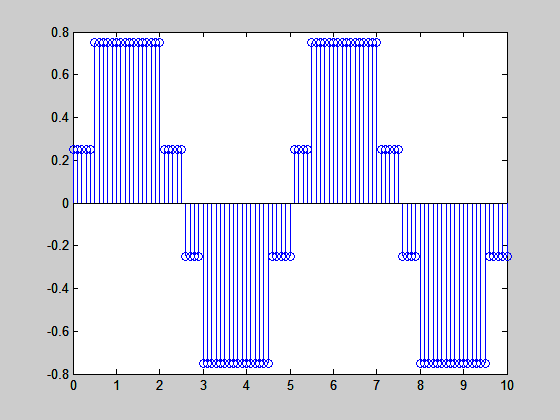
stem(t,x)

C:\Users\alumno\Desktop\P4\im1.emf

**Ejercicio2**

>> xq = uniform(2,x)

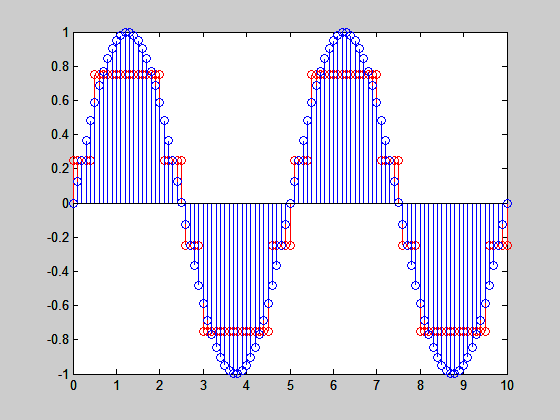
>> stem (t,xq)



>> stem (t,xq,'r')

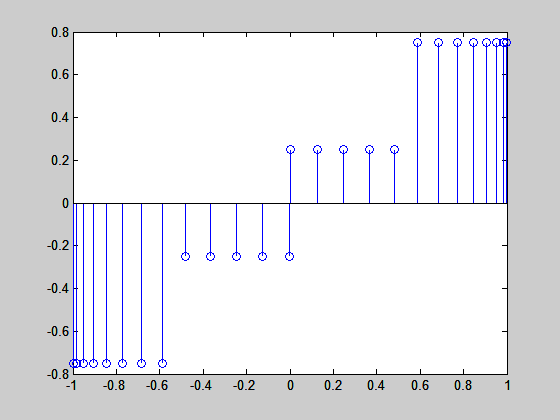
>> hold on

>> stem(t,x)



**Función de transferencia:**

>> stem(x,xq)



**Error máximo de cuantificación:**

0.5/2 = 0,25

**Ejercicio3**

**Potencia del ruido de cuantificación:**

Fórmula:

>> M=100

>> er=x-xq

>>P=(1/M)\*sum(er.^2)

P =

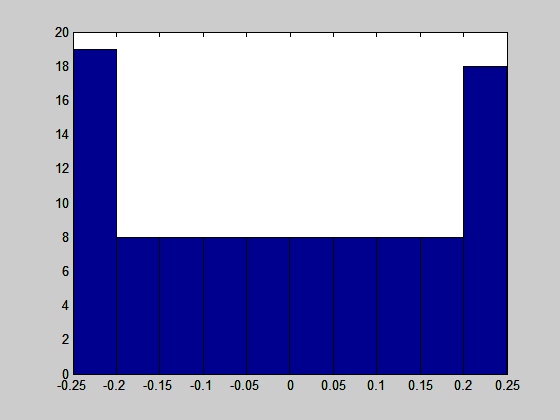
**0.0274**

Fórmula teórica:

Podemos observar que los resultados son bastante similares

**Función de densidad de probabilidad del error cuantificación:**

>> hist(er)



**Ejercicio 4**

>> xpcm\_par = bin\_enc(xq, 2)

xpcm\_par =

**1 0**

**1 0**

**1 0**

**1 0**

**1 0**

**1 1**

**1 1**

**1 1**

**1 1**

**1 1**

**1 1**

**1 1**

**1 1**

**1 1**

**1 1**

**1 1**

**1 1**

**1 1**

**1 1**

**1 1**

**1 1**

**1 0**

**1 0**

**1 0**

**1 0**

**1 0**

**0 1**

**0 1**

**0 1**

**0 1**

**0 0**

**0 0**

**0 0**

1. **0**

**.**

**.**

**.**

**.**

**.**

>>xpcm\_ser = par2ser(xpcm\_par)

**Ejercicio 5**

>> xpcm\_dec\_par = ser2par(xpcm\_ser, 2)

>> xq\_dec = bin\_dec(xpcm\_dec\_par)

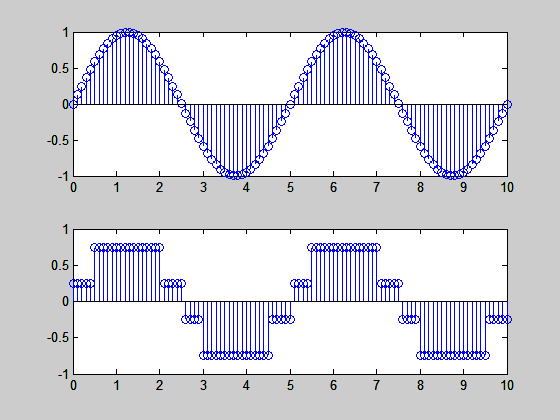
Reperesentación:

>> subplot(2,1,1)

>> stem(t,x)

>> subplot(2,1,2)

>> stem(t,xq\_dec)



**Ejercicio 6**

>> Pe=0.1

>> xpcm\_can\_ser = chan\_bin(xpcm\_ser, Pe)

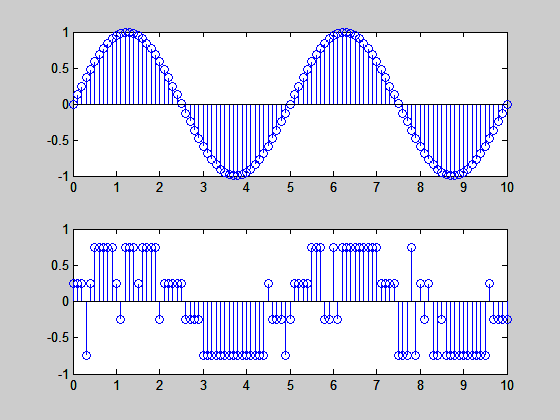
>> xpcm\_dec\_par1 = ser2par(xpcm\_can\_ser, 2)

>> xq\_dec1 = bin\_dec(xpcm\_dec\_par1)

Representación:

>> subplot(2,1,2)

>> stem(t,xq\_dec1)



**Ejercicio7**

>> Pe=0

>> xpcm\_can\_ser = chan\_bin(xpcm\_ser, Pe)

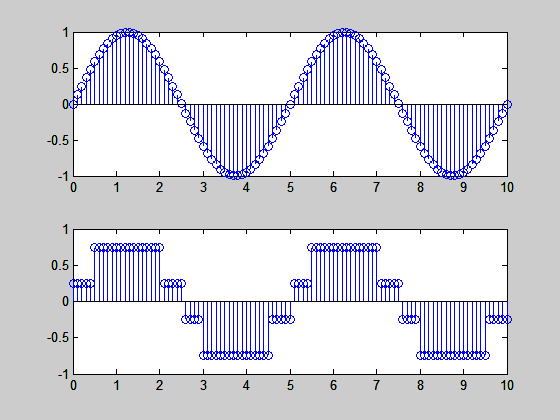
>> xpcm\_dec\_par1 = ser2par(xpcm\_can\_ser, 2)

>> xq\_dec1 = bin\_dec(xpcm\_dec\_par1)

Representación:

>> subplot(2,1,2)

>> stem(t,xq\_dec1)



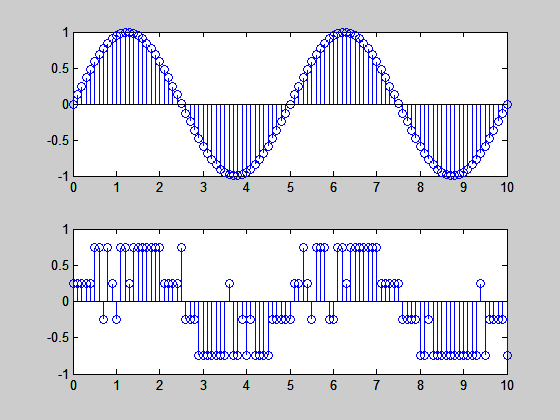
>> er=x-xq\_dec1

>> P=(1/M)\*sum(er.^2)

**P =**

**0.0274**

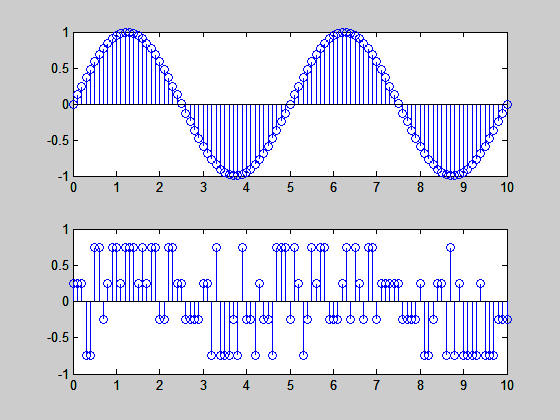
**Pe=0.1**



P =

0.1461

>> Pe=0.2



P =

0.4223