Ejercicios Respuesta al Impulso y Correlación

PROCESAMIENTO DIGITAL DE SEÑALES

Respuesta al Impulso

Obtén la respuesta al impulse de la siguiente ecuación de diferencias. Las condiciones iniciales son $y[n] = 0, n < 0$

$$y[n] - 0.8y[n-1] + 0.15y[n-2] = x[n]$$

El polinomio característico de la ecuación es:

$$p(\lambda) = \lambda^{n-2}(\lambda^2 - 0.8\lambda + 0.15)$$

Encuentra las raíces del polinomio.

numpy.roots

```
import numpy as np
from matplotlib import pyplot as plt

# define los coeficientes
coeff = [1,-0.8,0.15]
lamda = np.roots(coeff);
print(lamda)
```

```
[0.5 0.3]
```

Resolviendo el Sistema de Ecuaciones

```
A=np.array([[1,1],[0.5,0.3]])
b = np.array([[1, 0.8]]).T
C=np.linalg.solve(A,b)
print(C)
```

```
A=np.array([[1,1],[0.5,0.3]])
b = np.array([[1, 0.8]]).T
C = np.dot(np.linalg.inv(A),b)
print(C)
```

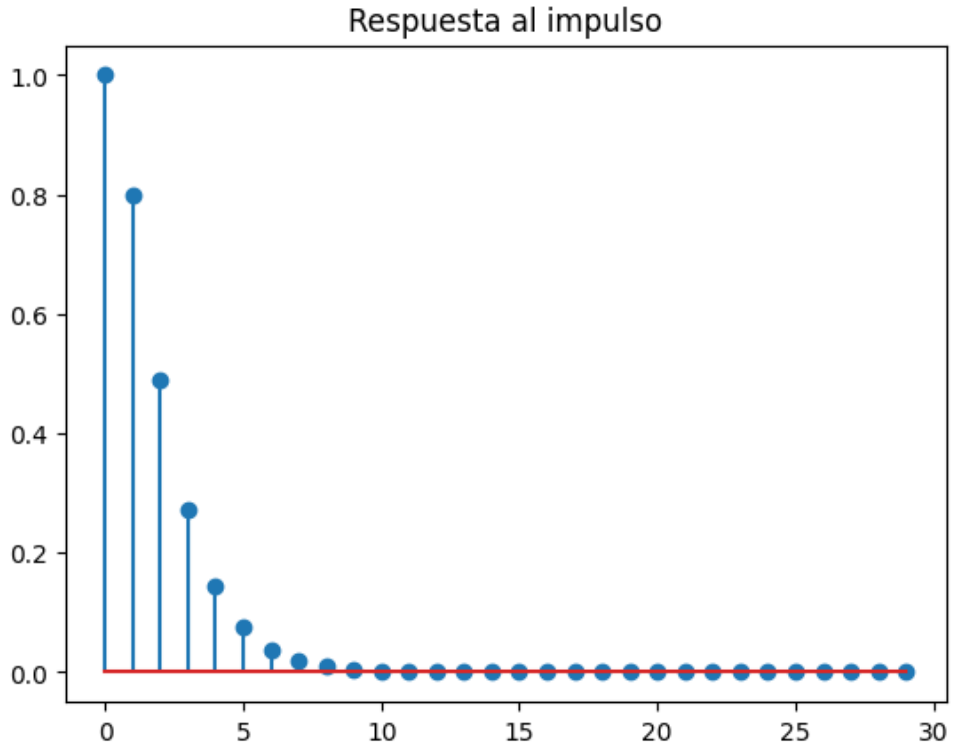
[[2.5] [-1.5]]

$C_1 = 2.5$ $C_2 = -1.5$

Respuesta al Impulso

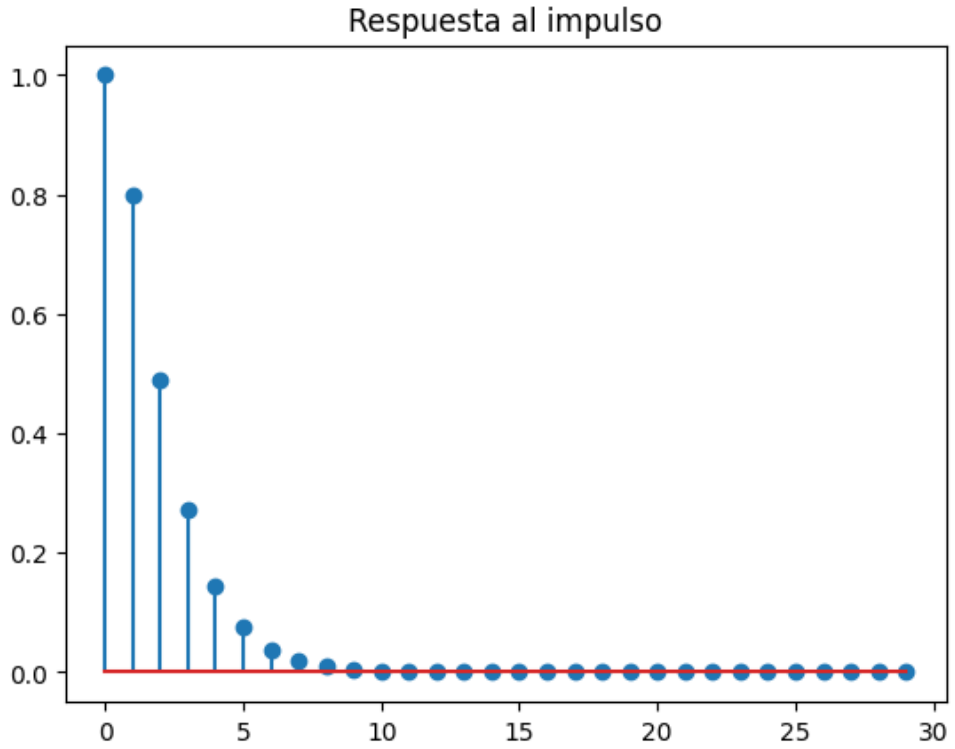
$h[n] = 2.5(0.5^n) - 1.5(0.3^n), n \geq 0$

```
n=np.arange(30)

h=2.5*(0.5**n)-1.5*(0.3**n)

plt.stem(n,h)

plt.title('Respuesta al impulso')
```



Respuesta al impulso

```
def sistema(x):
    y = np.zeros(np.size(x))
    for n in range(np.size(x)):
        y[n]=0.8*y[n-1]-0.15*y[n-2]+x[n]
    return y

n=np.arange(30)
delta = n==0
y = sistema(delta)
plt.stem(n,y)
plt.title('Respuesta al impulso')
```



Ejercicio

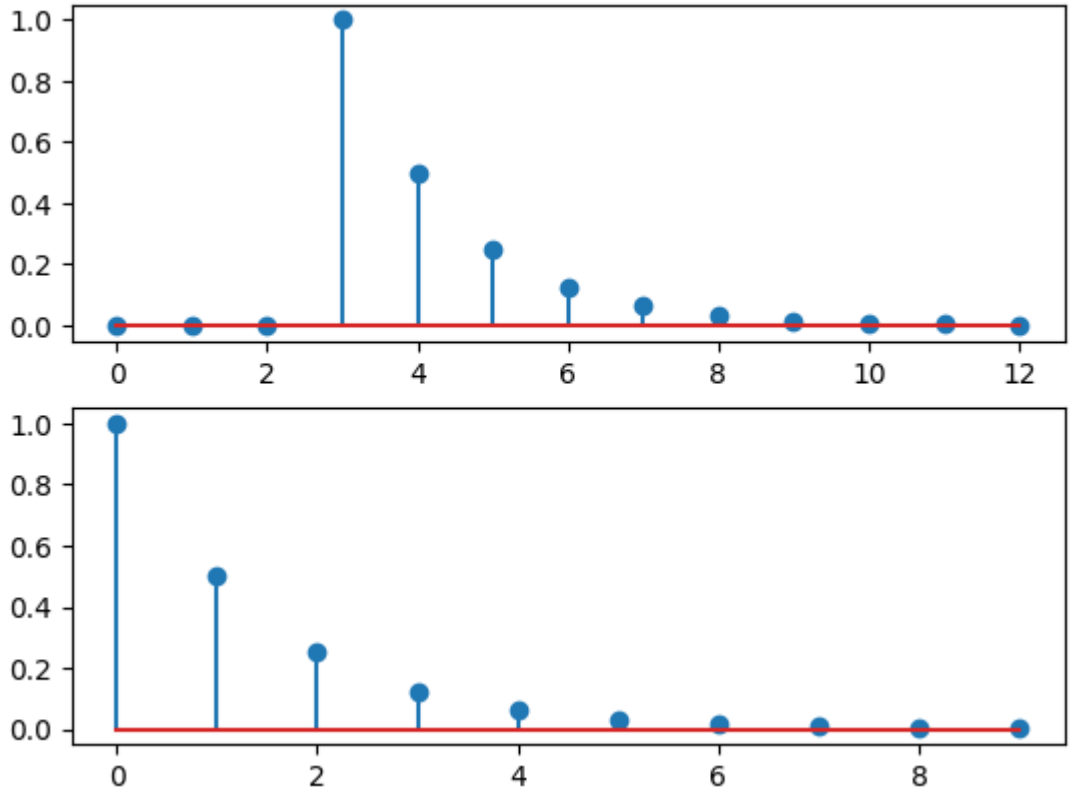
Obtén la respuesta al impulse de la siguiente ecuación de diferencias. Las condiciones iniciales son $y[n] = 0, n < 0$

$$y[n] - 2.1y[n-1] + 0.18y[n-2] + 0.04y[n-3] = x[n]$$

1. Encuentra el polinomio característico
2. Obtén sus raíces
3. Resuelve el Sistema de ecuaciones
4. Grafica la respuesta al impulso

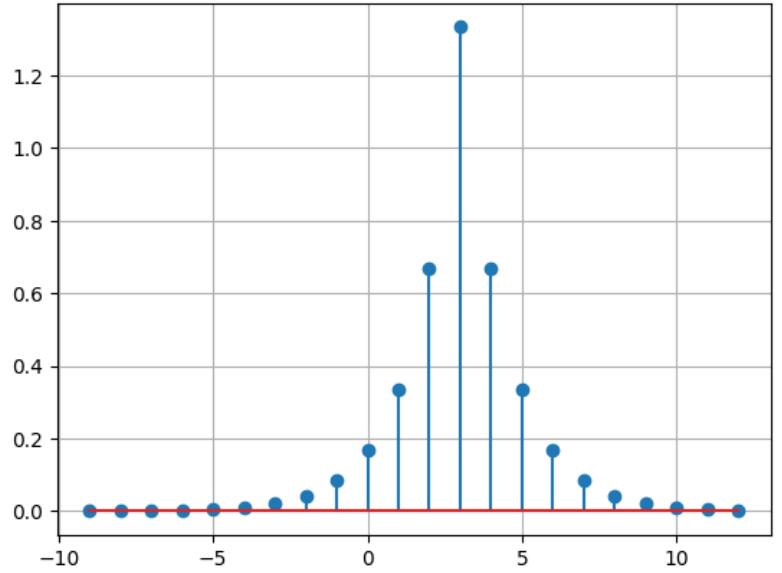
Correlación

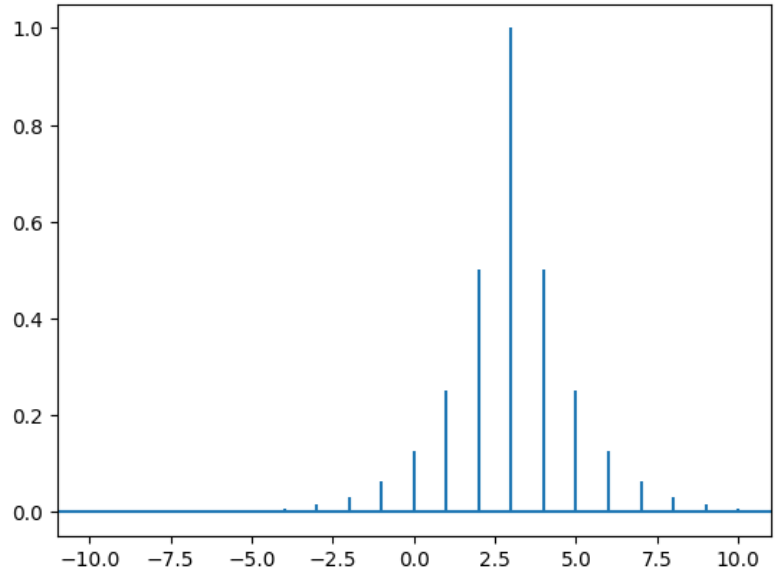
```
import numpy as np
import matplotlib.pyplot as plt

def expSeq(a,low,high):
    n = np.arange(low,high)
    x = a**n;
    return n,x

def retardo(x,n,k):
  y = np.zeros(np.size(x)+k)
  y[k:np.size(x)+k] = x;
  m = np.arange(n[0],n[-1]+(k+1)*(n[1]-n[0]),n[1]-n[0]);
  return m,y

def corr(h,m,x,n):
  y=np.convolve(h,np.flip(x,0))
  t=np.arange(-n[-1]+m[0],n[0]+m[-1]+1)
  return t,y
```



Correlación

```
n,x=expSeq(0.5, 0, 10)
m,y=retardo(x,n,3);
n1,r=corr(y,m,x,n);
plot1=plt.figure(1)
plt.stem(n1,r)
plt.grid()
plot1=plt.figure(2)
n1,x1=expSeq(0.5, 0, 13)
plt.xcorr(y,np.append(x,np.zeros(3)))
plot1=plt.figure(3)
plt.subplot(2,1,1)
plt.stem(m,y)
plt.subplot(2,1,2)
plt.stem(n,x)
```





Ejercicio

Realice un programa en Python que le permita calcular la autocorrelación de la señal $x[n]$ corrompida por ruido gaussiano con media cero y desviación estándar 0.1, grafique el resultado.

$$x[n] = 1\delta[n] + 2\delta[n-1] + 3\,\delta[n-2] + 2\delta[n-3] + \delta[n-4]$$

El ruido se puede modelar mediante:

```
ruido = np.random.normal(0,0.1,x.shape)
s = ruido+x;
```

¿Qué nota de la secuencia resultante?, ¿Qué sucede si aumenta la desviación standard a 0.9 ?

Ejercicio

Genere la señal exponencial:

$$x[n] = 0.8^n$$

Para n=0,…, 30; Agregue ruido Gaussiano con media 0, y desviación estándar 0.1, para obtener la secuencia

$$y[n] = x[n] + w[n]$$

Obtenga la correlación cruzada de $x[n]$ y $y[n]$. Grafique la señal $x[n]$, $y[n]$ y su autocorrelación. Ahora, cambie el valor de la desviación estándar a 0.5 y 0.9, ¿Qué observa?