Laboratorio de Bioseñales y Sistemas

Aplicación 2
Señales y Sistemas discretos

Representación de señales discretas

Para la representación de una secuencia discreta en Python, se utiliza un vector fila el cual contiene los valores correspondientes. Sin embargo, dicho vector fila no incluye ninguna información acerca de cuál es la muestra correspondiente a cada valor del vector. Es por esto que para una correcta representación de una señal discreta *x(n)*, es necesario crear un vector para cada *x* y otro para cada *n*. Por ejemplo, la secuencia $x(n) = \{5,6,\underset{\wedge}{7},5,4,3\}$ se representaría como:

```
>> n=[-2,-1,0,1,2,3]; x=[5,6,7,5,4,3];
```

Donde el sombrero indica cuál es la muestra correspondiente al índice $n = 0$. En caso de que se trate de una señal discreta en la que no se requiera información de la posición, o en la que la información de la posición sea trivial (es decir que la señal inicie en *n = 0*) se puede omitir el vector correspondiente a las muestras.

Implemente las siguientes funciones, tanto las que se presentan como las que se propone implementar:

Secuencias Discretas Elementales

Impulso unitario

El impulso unitario se define como:

$$\delta(n - n_0) = \begin{cases} 1 & si\ n = n_0 \\ 0 & si\ n \neq n_0 \end{cases}$$

En Python, implementamos el impulso unitario mediante la siguiente función:

```
def impseq(n0,n1,n2):
#   Genera x(n) = delta(n-n0); n1 <= n <= n2
# ---------------------------------------------
    n = np.arange(n1,n2+1); # Se crea el vector de muestras
    x = (n-n0) == 0;
    return [x,n]
```

Escalón unitario

El escalón unitario se define como:

$$u(n - n_0) = \begin{cases} 1 & si\ n \geq n_0 \\ 0 & si\ n < n_0 \end{cases}$$

1. Implemente una función en Python que permita generar una función escalón unitario, definida en un intervalo $n_1 \leq n_0 \leq n_2$

Rampa

La función rampa se define como:

$$r(n-n_0)=\begin{cases}0 & si\ n<n_0\\ n & si\ n\geq n_0\end{cases}$$

2. Implemente una función en Python que permita generar una función rampa, definida en un intervalo $n_1 \leq n_0 \leq n_2$

Exponenciales complejas

Las exponenciales complejas son de la forma:

$$x(n)=e^{(\sigma+j\omega 0)n}, \forall\ n$$

En la que σ produce una atenuación si es negativo o una amplificación si es positivo. En Python se hace uso del comando *np.exp*.

Sinusoides

Estas secuencias son de la forma:

$$x(n)=\sin(\omega_0+\theta_0)$$

3. Genere las siguientes secuencias usando las funciones básicas de Python que se han presentado. Grafique los resultados.

   a. $x_1(n)=\ 5\delta(n+3)+4\delta(n+2)+3\delta(n+1)+2\delta(n)+\ \delta(n)$. ¿Cómo debe ser el vector de muestras?
   b. $x_2(n)=\ -2u(n+5)+4u(n), -5\leq n\leq 5$
   c. $x_3(n)=e^{0.05n}\sin(0.1\pi n), 0\leq n\leq 100$. Comente acerca de la forma de onda.
   d. $x_4(n)=\ r(n+2)-r(n-2)-4u(n-5), -5\leq n\leq 10$

4. Sea $x(n)=\{-4,-3,-2,-1,0,\hat{1},2,3,4\}$. Genere las siguientes secuencias y grafique los resultados.

   a. $x_5(n)=2x(n-4)+x(n)$
   b. $x_6(n)=2e^{0.5n}x(n)+sin(0.2\pi n)\,x(n+2), -20\leq n\leq 20$