



Práctica 1 de Señales y Sistemas

Generación de señales discretas y transformaciones de la variable independiente

Grado en Ciencia e Ingeniería de Datos

Profesores:

Alejandro Díaz, Jorge Larrey y Antonio Lozano

Prólogo

El objetivo de esta práctica es que el alumno se familiarice con la generación de distintos tipos de señales discretas, así como que conozca las principales transformaciones que puede sufrir la variable independiente.

Las prácticas de la asignatura Señales y Sistemas se llevan a cabo en el entorno MATLAB. Para obtener ayuda sobre cualquier comando en MATLAB, introduzca **help nombre del comando**.

Funciones útiles para la realización de la práctica

abs, angle, cos, ceil, exp, figure, find, floor, imag, length, log, mod, ones, plot, rand, randn, real, round, semilogx, sin, size, subplot, sum, stem, tan, zeros

1. Generación de señales discretas

Ejercicio

- Genere la señal $x[n]$
`x=[0 0 -0.4 -0.75 -0.2 0 0.5 0.8 1.2 0.8 0.4 0 -0.3 0 0];`
- Genere el eje temporal
`N=floor(length(x)/2);`
`n=[-N:N];`
- Dibuje la señal $x[n]$
`stem(n,x,'.')`
- Muestre el valor de la señal para el instante $n=-3$
`x(find(n==-3))`



- Modifique el valor de la señal, $x[-3] = -0.6$
`x(find(n==-3))=-0.6;`

Cuestiones

Utilizando las variables creadas en el ejercicio anterior

- Genere la señal $y[n] = (-1.1)^n$
- Genere la señal $x2[n] = y[n] \cdot x[n]$
- Genere $z[n] = 0.75 \cdot (x[n] + x2[n])$
- Genere la señal $E[n] = x[n] \cdot x[n]$ o $E[n] = x^2[n]$
- Calcule la energía y la potencia de $x[n]$

2. Transformaciones en la variable independiente

Desplazamiento temporal

Se muestra el código una función que aplica un desplazamiento temporal sobre una secuencia discreta como ejemplo de programación de una función en MATLAB, concretamente $y[n] = x[n - n_0]$.

```
function y=desplazamiento(x,n0)
long=length(x);                               % Longitud de x
if(n0>=0)                                     % Si n0 es =>0
    y=[zeros(1,n0) x(1:long-n0)];             % desplazamiento a la derecha
else                                           % Si n0 es <0
    n0=-n0;
    y=[x(1+n0:long) zeros(1,n0)];             % desplazamiento a la izquierda
end
```

La función se denomina **desplazamiento** y recibe dos argumentos, **x** (el vector a desplazar) y **n0** (un escalar que indica el número de muestras a desplazar). La función proporciona un solo resultado, la secuencia desplazada, **y**. El vector resultante tendrá las mismas dimensiones que el original, de esta manera se eliminarán parte de los valores una vez realizado el desplazamiento hasta que los tamaños coincidan y así se facilitará la representación. Por otra parte se añaden ceros por la derecha o por la izquierda según sea necesario.

Cree el fichero **desplazamiento.m** con el editor de MATLAB: **File-> New-> M-file**. Copie el código anterior y guárdelo.

- Compruebe el correcto funcionamiento de dicha función ejecutando las siguientes instrucciones:

```
x = rand(1,15);
n = -7:7;
y = desplazamiento(x,1);
subplot(2,1,1)
```



```
stem(n,x,'.')  
subplot(2,1,2)  
stem(n,y,'.')  

```

- Ejecute retardos de 2 (2 muestras hacia la derecha) y -3 (3 muestras hacia la izquierda) y represente las gráficas correspondientes comprobando el correcto funcionamiento de la función.

Inversión temporal

Otra función que implementaremos es la función *inversion.m*, que realiza una inversión en el tiempo del vector **x**, de nuevo, el resultado se proporciona en **y**, y en este caso $y[n]=x[-n]$. Guarde de nuevo la función tal y como se ha realizado en el caso anterior:

```
function [y,n2]=inversion(x,n)  
ind = length(x):-1:1;  
y = x(ind); % Cambiamos el orden del vector x  
n2 = -n(ind); % calculamos los índices de n  

```

- Emplee las secuencias **x1**=[1 1 2 3 5 8 13 21 34]; (definida entre n=0 y n=8) y **x2**=[1 -2 3 -4 5 -4 3 -2 1]; (definida entre n=-4 y n=4) para calcular sus correspondientes inversiones en el tiempo y1 e y2, respectivamente.
- Compruebe si la inversión se ha realizado correctamente consultando las gráficas generadas. ¿Qué puede comentar sobre y2?

Compresión

Seguidamente trabajaremos con la función *compresion.m*, que realiza una compresión temporal por un factor **a** de la secuencia **x**, proporcionando el resultado en **y**, es decir $y[n]=x[a \cdot n]$. Guarde la función para poder ejecutarla en Matlab:

```
function [y,n2]=compresion(x,n,a)  
ind = find(mod(n,a)==0); % Valores de n múltiplo de a  
y = x(ind); % Valores de x para n múltiplo de a  
n2 = n(ind)/a; % Variable independiente  

```

- Empleando las secuencias del apartado anterior calcule su compresión en el tiempo y1 e y2, respectivamente. Emplee varios valores del factor **a**.
- ¿Se ha realizado la compresión correctamente? ¿Qué ocurre con las dimensiones del vector **y** con respecto a **x**?



3. Señales periódicas

Ejercicio

- Genere la señal $x[n]$ y el eje temporal

```
x=randn(1,9);
x=[x x x x x];
N=floor(length(x)/2);
n=[-N:N];
```
- Representela en la parte superior de la figura empleando el comando **subplot**

```
subplot(2,1,1)
stem(n,x,'.')
```
- Represente la señal en la ventana $[-20,20]$ (en la parte inferior de la figura empleando **subplot**)

```
n2=[-20:20];
i0=find(n==n2(1));
subplot(2,1,2)
stem(n2,x(i0:i0+length(n2)-1))
```

Cuestiones

- Genere la señal $x[n]=\sin(\pi/5 \cdot n)$ en el intervalo $[-100:100]$.
- Indique el periodo N de la señal
- Muestre en la parte superior de la figura la señal en el intervalo $[-20:20]$
- Muestre en la parte inferior de la figura la señal $x[n-N]$ en el intervalo $[-20:20]$
- Genere la señal $y[n]=\sin(0.6 \cdot n)$ entre $[-100:100]$.
- Muestre la señal $y[n]$ en el intervalo $[-20:20]$ en la parte inferior de la figura. ¿Es similar a la anterior?
- Cree $z[n]=y[n]-y[n-N]$ y muestre en el intervalo $[-20:20]$ ¿Es $y[n]$ periódica?

4. Exponenciales complejas

Ejercicio

- Cree los números complejos $z_1=1.2+j0.75$ y $z_2=0.5+j0.8$

```
z1=1.2+0.75*j;
```



$z2=0.5+0.8*j;$

- Fije los ejes de la figura entre -2 y 2 para cada eje del plano complejo

`cla reset`

`axis([-2 2 -2 2])`

`hold on`

- Dibuje los dos números complejos $z1$ y $z2$, y su producto

`plot(z1,'b.')`

`plot(z2,'g.')`

`plot(z1*z2,'r.')`

- Muestre el módulo y la fase de $z1$

`abs(z1)`

`angle(z1)`

- Genere la señal $x[n]=e^{j0.01\pi n}$ entre [-30,30]

`n=[-30:30];`

`x=exp(j*0.01*pi*n);`

- Muestre en la pantalla la señal compleja $x[n]$

`plot(x,'.')`

- Muestre en la parte superior de la figura la parte real de $x[n]$ y en la inferior la parte imaginaria.

`hold off`

`subplot(2,1,1)`

`stem(n,real(x),'.')`

`subplot(2,1,2)`

`stem(n,imag(x),'.')`

Cuestiones

Para la siguiente exponencial compleja represente

$x[n]=e^{-n/10+jn/4}$ entre 0 y 40

- La parte real e imaginaria en función de n
- El módulo y la fase en función de n