# IEE239 - Procesamiento de Señales e Imágenes Digitales
## Laboratorio 1 - Ejercicios Propuestos
### Segundo Semestre 2017

1. Se cuenta con 2 archivos de audio con sonidos facilmente distinguibles: `scream.wav` y `back.wav`. Adicionalmente, se cuenta con el archivo `lti01.mat`, el cual contiene los coeficientes $b_k$[1] de un **sistema LTI** en su forma **no recursiva**[2]:

$$y_1[n] \triangleq T_1\{x[n]\} = \sum_{k=-\infty}^{+\infty} b_k x[n-k].$$

La implementación del sistema es descrita en la Figura 1. Adicionalmente, se cuenta con el sistema **media acumulada**:

$$y_2[n] \triangleq T_2\{x[n]\} = \frac{1}{n+1}\sum_{k=0}^{n} x[k]; \quad n \geq 0.$$

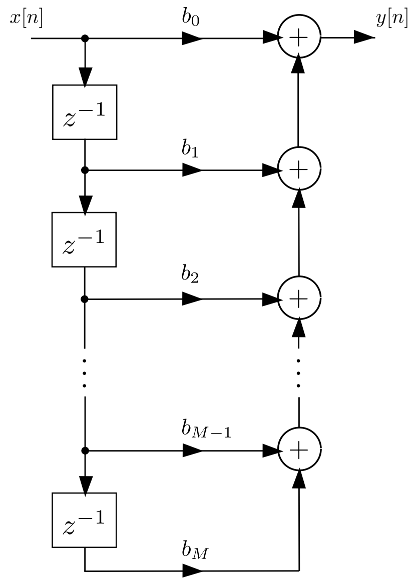



Figura 1: Sistema no recursivo en tiempo discreto $T_1$.

A partir de esta información, se requiere desarrollar el siguiente procedimiento:

a. Leer ambos audios y almacenarlos en las variables `scream_v` y `back_v`, respectivamente. Usar el comando **audioread()**. Dado que se trata de contenido stereo, conservar unicamente el primer canal de audio. Luego, recortar el vector de tal manera que conserve los

[1]La variable que los almacena es `b_v` y está ordenada de forma ascendente: b_v(1)= $b_0$, b_v(2)= $b_1$, etc.

[2]Los archivos están almacenados en la carpeta **/laboratorio/lab01/propuestos01/**.

primeros 4 segundos de información y reproducir cada audio por separado a partir de la función **audioplayer()**[3]. Finalmente, graficar ambas secuencias en una sola ventana. Usar las funciones **plot()**, **subplot()**, **figure()**, **xlabel()**, **ylabel()**, **title()**.

**Solución:**

```
% --- lectura y reproduccion de audio --- %
str01= './scream.wav';  % asumiendo que MATLAB apunta a la ...
    carpeta en la cual se encuentran los archivos WAV
str02= './back.wav';

[ scream_v, fs01]= audioread( str01);    % scream_v: secuencia ...
    de audio, fs01: frecuencia de muestreo a la cual fue ...
    adquirida la senal
length_v= 1: 4* fs01;   % numero de muestras 4 primeros ...
    segundos de audio para una frecuencia de muestreo fs01
scream_v= scream_v( length_v, 1);   % preservar unicamente los ...
    primeros 4 segundos de audio
player= audioplayer( scream_v, fs01);
play( player);  % reproducir audio

[ back_v, fs02]= audioread( str02);
back_v= back_v( length_v, 1);
player= audioplayer( back_v, fs02);
play( player);

% --- descripcion grafica --- %
fig01= figure;
subplot( 2, 1, 1); plot( scream_v); title( 'audio con sonidos ...
    agudos (alta frecuencia)');
xlabel( 'muestras'); ylabel( 'amplitud');
subplot( 2, 1, 2); plot( back_v); title( 'audio con sonidos ...
    graves (baja frecuencia)');
xlabel( 'muestras'); ylabel( 'amplitud');
```

b. Obtener la secuencia `super_v= scream_v +back_v`. Luego, reproducir el audio resultante usando la función **audioplayer()** y verificar que la mayoría del contenido no sufre distorsiones dado que se trata de un audio con sonidos graves y otro con sonidos agudos.

**Solución:**

```
% --- superposicion de audio --- %
super_v= scream_v+ back_v;
player= audioplayer( super_v, fs02);    % dado que fs01 y fs02 ...
    tienen el mismo valor, es posible realizar la suma sin ...
    distorsionar el audio.
play( player);
```

c. Se requiere recuperar las señales originales a partir de la versión superpuesta. Para ello, se hará uso del sistema $T_1$. Primero, demostrar que el sistema $T_1$ es lineal e invariante en el tiempo. Luego, determinar su respuesta al impulso y obtener la respuesta del sistema ante la entrada `super_v` a partir de la sumatoria de convolución usando la función **conv()**. Describir gráficamente el resultado y reproducirlo usando la función **audioplayer()**.

[3]Sugerencia, usar la información de frecuencia de muestreo para calcular el número de muestras de interés.

**Solución:**

I. <u>Linealidad</u>: se analiza el principio de superposición

$$
\begin{aligned}
T_1\{c\cdot x_1[n] + d\cdot x_2[n]\} &= \sum_{k=-\infty}^{+\infty} b_k\big(c\cdot x_1[n-k] + d\cdot x_2[n-k]\big)\\
&= \sum_{k=-\infty}^{+\infty} b_k\big(c\cdot x_1[n-k]\big) + \sum_{k=-\infty}^{+\infty} b_k\big(d\cdot x_2[n-k]\big)\\
&= c\cdot \sum_{k=-\infty}^{+\infty} b_k x_1[n-k] + d\cdot \sum_{k=-\infty}^{+\infty} b_k x_2[n-k]\\
&= c\cdot T_1\{x_1[n]\} + d\cdot T_1\{x_2[n]\}
\end{aligned}
$$

Por lo tanto, el sistema es lineal

II. <u>Invariante en el Tiempo</u>:

$$
y[n] = \sum_{k=-\infty}^{+\infty} b_k x[n-k]; \quad y[n-n_0] = \sum_{k=-\infty}^{+\infty} b_k x[n-n_0-k]
$$

$$
x_d \triangleq x[n-n_0]; \quad T\{x_d\} = \sum_{k=-\infty}^{+\infty} b_k x_d[n-k] = \sum_{k=-\infty}^{+\infty} b_k x[n-n_0-k]
$$

Por lo tanto, se demuestra que $T\{x[n-n_0]\} = y[n-n_0]$ para todo desplazamiento $n_0$ y toda secuencia de entrada $x[n]$. El sistema es invariante en el tiempo. Finalmente, $T_1$ es un sistema LTI. Luego, su frecuencia de muestreo corresponde a:

$$
\begin{aligned}
h[n] \triangleq T_1\{\delta[n]\} &= \sum_{k=-\infty}^{+\infty} b_k \delta[n-k]\\
&= \sum_{k=-\infty}^{+\infty} b_n \delta[n-k]\\
&= b_n \sum_{k=-\infty}^{+\infty} \delta[n-k] = b_n
\end{aligned}
$$

En consecuencia, los coeficientes del sistema no recursivo corresponden a la respuesta al impulso del sistema. A partir de ello, es posible obtener la respuesta del sistema ante cualquier entrada usando la sumatoria de convolución: $y[n] = x[n] * b_n$.

```
% --- respuesta de audio superpuesto a sistema LTI --- %
load ./lti01.mat     % asumiendo que MATLAB apunta a la ...
    carpeta en la cual se encuentran el archivo MAT
y01= conv( super_v, b_v, 'same');    % Primer argumento: ...
    secuencia de entrada. Segundo argumento, respuesta al ...
    impulso. Usar bandera 'same' para preservar el tamano de ...
    la secuencia de entrada.
e_y01= norm( y01- scream_v);
player= audioplayer( y01, fs02);     % reproducir resultado
play( player);
```

El audio resultante es muy similar a `scream_v`. Es decir, el sistema $T_1$ permite aislar las componentes agudas de la versión superpuesta. Para determinar de forma cuantitativa si es similar al audio original, se usa la norma $\ell_2$ como métrica de similaridad a partir de la función **norm()**:

$$e = \ell_2\{y, \text{scream\_v}\} = \sum_{n=-\infty}^{+\infty} |y_1[n] - \text{scream\_v}[n]|$$

```
>> e_y01
e_y01 =
0.4361
```

en comparación a los valores de ambas secuencias, el error calculado es mínimo, por lo que se puede afirmar que ambas secuencias son muy parecidas.

d. A partir del resultado del inciso anterior, proponer un sistema que permita preservar las componentes graves de *super_v*. Determinar su salida, reproducirla y verificar su similaridad con la secuencia `back_v`. Se trata de un sistema LTI? Justificar claramente su respuesta.

**Solución:** Una forma simple de estimar las componentes graves de la secuencia es restar la secuencia que conserva las componentes agudas. Por lo tanto, la secuencia estimada puede expresarse como:

$$\widehat{\text{back\_v}}[n] = \text{super\_v} - \text{super\_v} * b_n = \text{super\_v} * \big(\delta[n] - b_n\big).$$

Entonces, modelando un nuevo sistema cuya respuesta al impulso corresponde a $\hat{b}_n = \delta[n] - b_n$, podemos estimar las componentes graves de interés.

```
% --- sistema para preservar componentes graves en funcion a ...
    sistema LTI original --- %
b_hat_v= zeros( size( b_v));    % vector de 0s
b_hat_v( 1025)= 1;    % impulso unitario (asumiendo que la ...
    posicion 1025 del vector corresponde a n= 0) [fase lineal]
b_hat_v= b_hat_v- b_v;   % respuesta al impulso del nuevo sistema
z01= conv( super_v, b_hat_v, 'same');   % Primer argumento: ...
    secuencia de entrada. Segundo argumento, respuesta al ...
    impulso. Usar bandera 'same' para preservar el tamano de la ...
    secuencia de entrada.
e_z01= norm( z01- back_v);
player= audioplayer( z01, fs02);    % reproducir resultado
play( player);

>> e_z01
e_z01 =
0.4361
```

Nuevamente, el error demuestra que la secuencia estimada es muy similar a la secuencia `back_v` original.

e. Determinar si el sistema $T_2$ (media acumulada) es lineal o invariante en el tiempo. Luego, obtener la respuesta del sistema $T_2$ ante la entrada `super_v` de forma no recursiva y de forma recursiva. Calcular el tiempo de procesamiento requerido para cada implementación y argumentar acerca de cuál de ellas es más eficiente. Usar las funciones **tic()** y **toc()** para calcular tiempo de procesamiento. Finalmente, reproducir la secuencia resultante e indicar si lo obtenido es coherente con la naturaleza del audio de interés.

**Solución:**

I. Linealidad: se analiza el principio de superposición

$$\begin{aligned} T_2\{c \cdot x_1[n] + d \cdot x_2[n]\} &= \frac{1}{n+1}\sum_{k=0}^{n}\big(c \cdot x_1[k] + d \cdot x_2[k]\big) \\ &= \frac{1}{n+1}\sum_{k=0}^{n} c \cdot x_1[k] + \frac{1}{n+1}\sum_{k=0}^{n} d \cdot x_2[k] \\ &= \frac{c}{n+1}\sum_{k=0}^{n} x_1[k] + \frac{d}{n+1}\sum_{k=0}^{n} x_2[k] \\ &= c \cdot T_2\{x_1[n]\} + d \cdot T_2\{x_2[n]\} \end{aligned}$$

Por lo tanto, es lineal.

II. Invariante en el tiempo:

$$y[n] = \frac{1}{n+1}\sum_{k=0}^{n} x[k]; \quad y[n-n_0] = \frac{1}{n-n_0+1}\sum_{k=0}^{n-n_0} x[k]$$

$$x_d \triangleq x[n-n_0]; \quad T\{x_d\} = \frac{1}{n+1}\sum_{k=0}^{n} x_d[k] = \frac{1}{n+1}\sum_{k=0}^{n} x[k-n_0]$$

Cambio de variable: $\hat{k} = k - n_0$:

$$T\{x_d\} = \frac{1}{n+1}\sum_{\hat{k}=-n_0}^{n-n_0} x[\hat{k}] \neq \frac{1}{n-n_0+1}\sum_{k=0}^{n-n_0} x[k]$$

Por lo tanto, $T\{x[n-n_0]\} \neq [n-n_0]$. El sistema es variante en el tiempo. Finalmente, al no ser LTI, no está caracterizado por su respuesta al impulso y la salida no puede ser calculada a partir de convolución.

A continuación, se obtiene la salida del sistema de forma no recursiva usando directamente su expresión analítica, y de forma recursiva de acuerdo a la siguiente ecuación de diferencias:

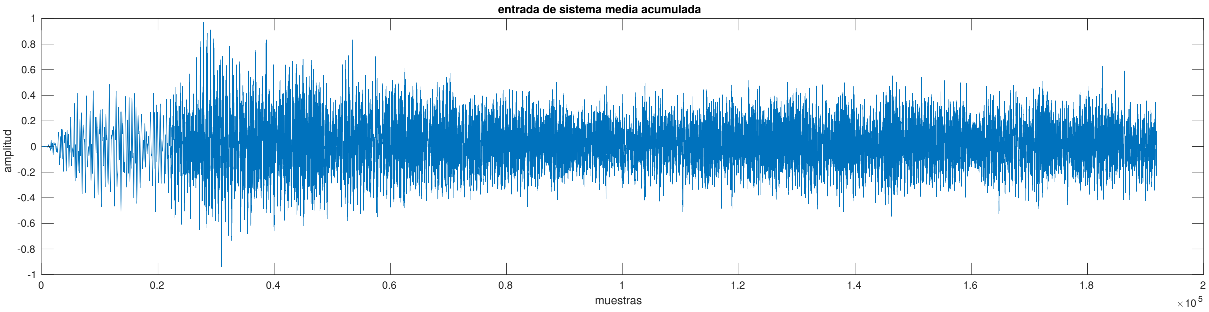
$$y[n] = \frac{n}{n+1}y[n-1] + \frac{1}{n+1}x[n]; \quad n \geq 0. \quad \text{[Forma recursiva de media acumulada]}$$

```
        % --- Media acumulada --- %
        y_v= zeros( size( super_v));    % vector de ceros (primera ...
            posicion corresponde a n= 0)
        iter_t= tic;    % iniciar cuenta de tiempo de procesamiento
        for i= 1: length( super_v)
            y_v( i)= 1/( i- 1+ 1)* sum( super_v( 1: i));    % media ...
                acumulada para cada valor de n calculada de forma no ...
                recursiva
```

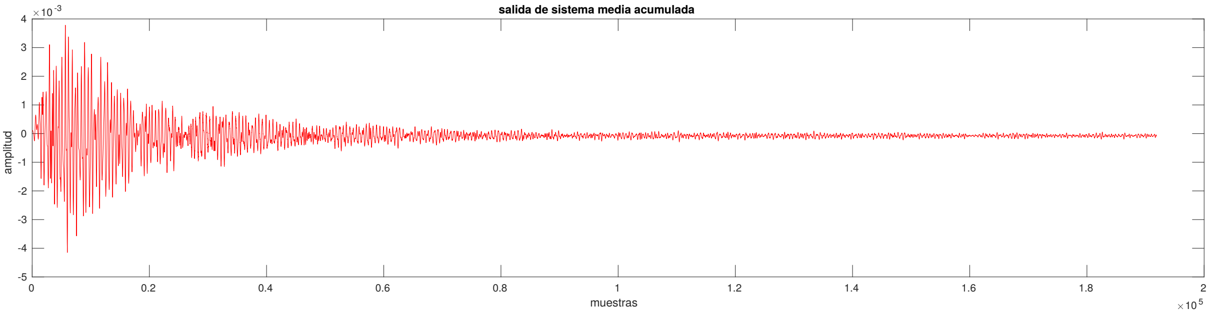
```
        end
        iter_time_v= toc( iter_t);  % terminar cuenta

        y_rec_v=  zeros( size( super_v));    % vector de ceros (primera ...
            posicion corresponde a n= 0)

        iter_rec_t= tic;    % iniciar cuenta de tiempo de procesamiento
        y_rec_v( 1)= super_v( 1);    % analizar salida cuando n=0: ...
            asumir sistema en reposo
        for i= 2: length( super_v)
            y_rec_v( i)= (i- 1)/( i- 1+ 1)* y_rec_v( i- 1)+ 1/( i- 1+ ...
                1)* super_v( i);    % media acumulada calculada de ...
                forma recursiva (depende de salida anterior)
        end
        iter_rec_time= toc( iter_rec_t);    % terminar cuenta

        e_acumulada= norm( y_v- y_rec_v);   % verificar si ambas ...
            secuencias son equivalentes

        % --- reproduccion y descripcion grafica --- %
        player= audioplayer( y_v, fs02);
        play( player);

        fig_acumulada= figure;
        subplot( 2, 1, 1); plot( super_v);
        title( 'entrada de sistema media acumulada');
        ylabel( 'amplitud'), xlabel( 'muestras');
        subplot( 2, 1, 2); plot( y_v, 'r');
        title( 'salida de sistema media acumulada');
        ylabel( 'amplitud'), xlabel( 'muestras');

        >> iter_time_v
        iter_time_v =
        24.2718

        >> iter_rec_time
        iter_rec_time =
        0.0157

        >> e_acumulada
        e_acumulada =
        1.0896e-15
```

Ambos procesos obtienen la misma secuencia. Sin embargo, la forma recursiva toma aproximadamente 24.272 segundos, mientras que la forma recursiva toma 0.02 segundos. La reproducción del audio y las gráficas obtenidas muestran que la salida tiende a 0. Esto es coherente ya que se tratan de señales de audio, por lo cual su valor promedio es muy cercano a 0, así que a mayor número de muestras promediadas, el sistema obtiene valores de salida mas cercanos al valor promedio de toda la secuencia. La Figura 2 muestra la entrada y salida de interés.

2. Las señal $x[n] = 50 \cdot \cos(\frac{\pi}{50} \cdot n) \quad n \in \{-200, 199, \ldots, 200\}$ es transmitida por un sistema de radar hacia 2 objetos cercanos. Por una falla en el sistema de adquisición de datos, las señales son reflejadas hacia el sistema con un alto nivel de distorsión, por lo que se recuperan muestras ruidosas de cada secuencia. Las señales recuperadas están definidas de la siguiente forma:




(a)




(b)

Figura 2: Sistema media acumulada. (a) Entrada al sistema: `super_v`. (b) Salida del sistema: `y_v`.

$$\tilde{x}_1[n] = x[n] + \eta; \quad \eta \sim (0, 10^2);$$

$$\tilde{x}_2[n] = x_{\mathrm{sp}}[n];$$

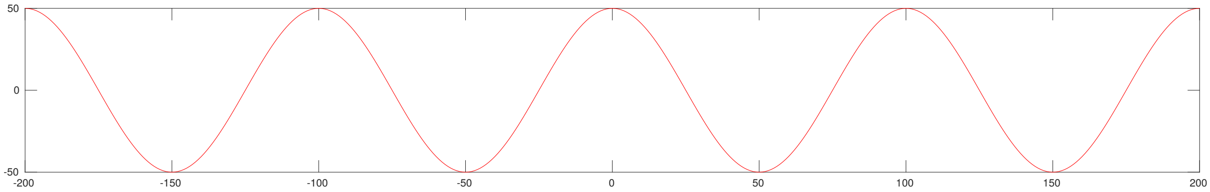
$$x_{\mathrm{sp}}[n] = \begin{cases} x[n] & \text{con probabilidad} \quad 0{,}85, \\ x_{\max} & \text{con probabilidad} \quad 0{,}075, \\ x_{\min} & \text{con probabilidad} \quad 0{,}075, \end{cases}$$

En consecuencia, se requiere implementar un método de procesamiento que permita recuperar ambas señales de manera eficaz para luego obtener sus desfases respecto a la señal original $x[n]$. Para la primera tarea, se cuenta con dos sistemas: un **filtro promedio** y un **filtro mediano**. Para la segunda tarea, se plantea usar **correlación cruzada**. La Figura 3 muestra la secuencia libre de ruido, así como ambas secuencias afectadas por ruido.

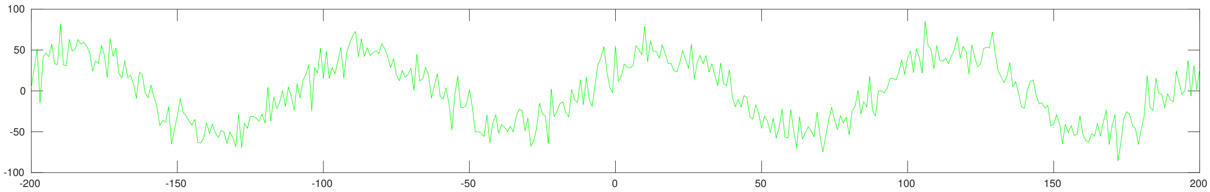
a) Agregar al workspace las variables `xhat01` ($\tilde{x}_1[n]$) y `xhat02` ($\tilde{x}_2[n]$) almacenadas en el archivo **radar.mat** y describirlas graficamente junto a la señal libre de ruido $x[n]$. Para esta última secuencia, crear la variable `xtrue` de acuerdo a la expresión de $x[n]$.

b) Hallar la respuesta de ambos sistemas ante la entrada $\tilde{x}_1[n]$ para un tamaño de filtro 10. Describir gráficamente el resultado y explicar detalladamente qué filtro obtuvo una señal resultante más fiel a la original. Usar las funciones **figure()**, **plot()**, **subplot()**, **title()**, **xlabel()**, **ylabel()**.

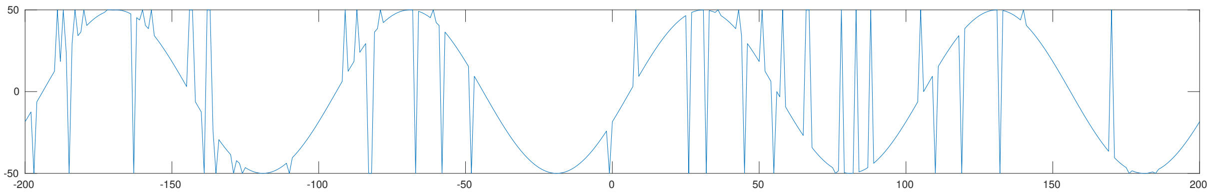
c) Hallar la respuesta de ambos sistemas ante la entrada $\tilde{x}_2[n]$ para un tamaño de filtro 10. Describir gráficamente el resultado y explicar detalladamente qué filtro obtuvo una señal resultante más fiel a la original.




(a)




(b)




(c)

Figura 3: Sistema de radar basado en eliminación de ruido y correlación. (a) Señal libre de ruido. (b) Señal afectada por ruido aditivo Gaussiano. (c) Señal afectada por ruido sal y pimienta.

d) Con los resultados más fieles de cada caso, calcular el desfase (en radianes) respecto a $x[n]$ aplicando **correlación cruzada** y definir cuál de los objetos está más alejado.

3. *(3 puntos)* **Dado el sistema de la Figura 4:**

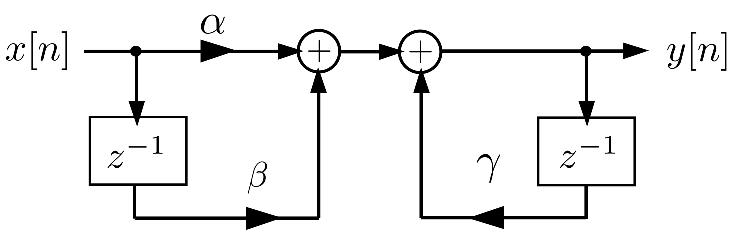



Figura 4: Sistema discreto en forma directa I.

a. Hallar la expresión de la función de transferencia, $H(z)$, del sistema causal, presente en la Figura 4. Luego, haciendo uso de las tablas y las propiedades de la transformada, calcular la respuesta al impulso $h[n]$, incluir su respuesta en los comentarios. De manera alternativa, generar la secuencia de las primeras 20 muestras de $h[n]$ con la función **impz()**. Finalmente graficar la respuesta al impulso calculada de forma analítica, y la calculada por medio de la función de matlab, verificar la similitud de ambas señales.

**Solución:**

I. Calculando de forma analítica $H(z)$

Simplificando en base a los valores de la función de transferencia:

$$\begin{aligned} Y(z) &= \alpha \cdot X(z) + \beta \cdot X(z) \cdot Z^{-1} - \gamma \cdot Y(z) \cdot Z^{-1} \\ Y(z)(1+\gamma \cdot Z^{-1}) &= X(z)(\alpha + \beta \cdot Z^{-1}) \\ \frac{Y(z)}{X(z)} &= \frac{(\alpha + \beta \cdot Z^{-1})}{(1+\gamma \cdot Z^{-1})} \\ H(z) &= \frac{(\alpha + \beta \cdot Z^{-1})}{(1+\gamma \cdot Z^{-1})} \end{aligned}$$

Separando las ecuaciones:

$$H(z) = \frac{\alpha}{(1+\gamma \cdot Z^{-1})} + \frac{\beta}{(1+\gamma \cdot Z^{-1})} \cdot Z^{-1}$$

II. Calculando de forma analítica $h[n]$
Usando tablas y la propiedad de desplazamiento temporal:

$$h[n] = \alpha \cdot \gamma^n u[n] + \beta \cdot \gamma^{n-1} u[n-1]$$

Reemplazando valores y simplificando:

$$h[n] = 0{,}14 \cdot (0{,}73)^n u[n] + 0{,}14 \cdot (0{,}73)^{n-1} u[n-1]$$

III. Calculando $h[n]$ con la función **impz()**:

```

% Pregunta 3.
% --- Calculando la respuesta al impulso con los coeficientes de ...
    H(z)--- %

b1=[0.14, 0.14];
a1=[1, -0.73];
h1=impz(b1,a1,20);
```

IV. Calculando $h[n]$ por medio :

```


% --- Calculando la respuesta al impulso con h[n] analitico--- %

n=(0:19)';   % se generan 20 muestras de forma vertical
h2= 0.14*((0.73).^(n)+ (n-1≥0).*(0.73).^(n-1) );
%Se considera (n-1≥0) para considerar el retrazo en tiempo de u[n-1]
```

V. Graficando ambas señales :

```
% ---- descripcion grafica ---- %
fig01= figure;
subplot( 2, 1, 1); stem(n,h1); title( 'Respuesta al impulso ...
    calculada con Impz');
xlabel( 'muestras'); ylabel( 'amplitud');
subplot( 2, 1, 2); stem(n,h2 ); title( 'Respuesta al impulso ...
    calculada de forma analitica');
xlabel( 'muestras'); ylabel( 'amplitud');
```

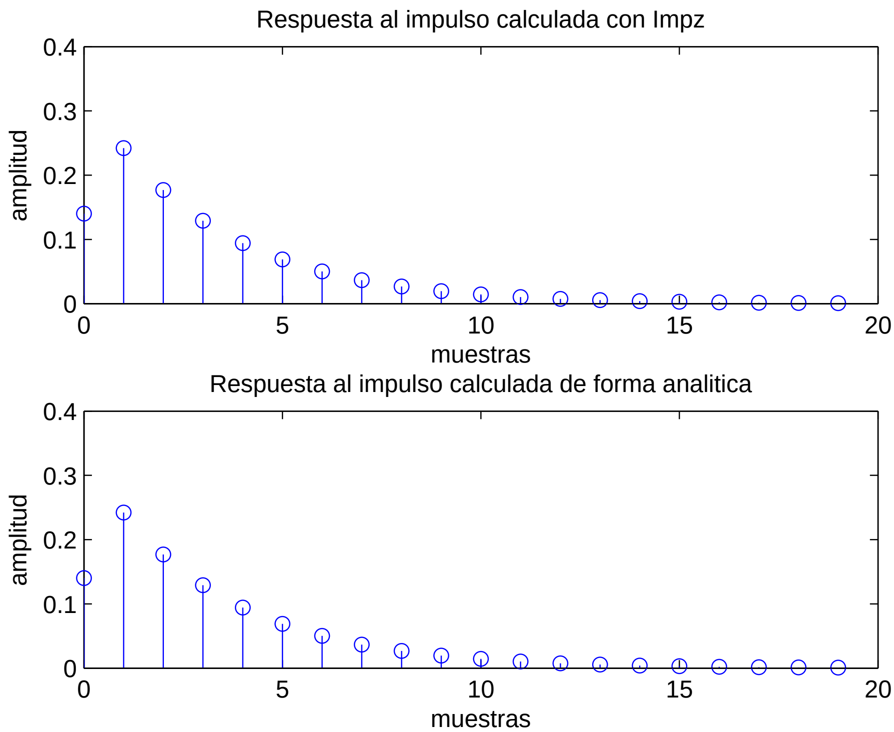



Figura 5: Respuesta al impulso por ambos métodos.

```
error=norm(h1-h2); % calculo de similitud
```

b. De forma analítica, calcular polos y zeros, verificar estos, usando **zplane()**. Determinar todas las regiones de convergencia e incluirlas en sus comentarios.

**Solución:**

I. Calculando de forma analítica polos y ceros:

- Calculando ceros

$$\begin{aligned} \alpha + \beta/z &= 0 \\ z &= -1 \end{aligned}$$

- Calculando polos

$$\gamma/z = 1$$
$$z = 0{,}73$$

II. Calculando diagrama de polos y ceros con **zplane()**:

```
% Pregunta 2:

% ——Calculando polos y ceros —— %
zplane(b1,a1)
```

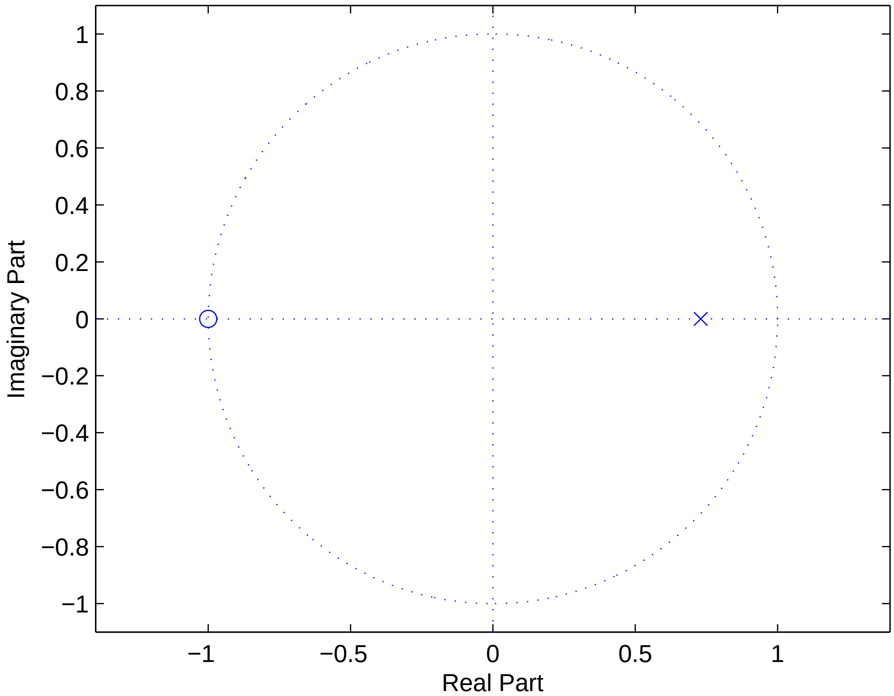



Figura 6: Diagrama de polos y ceros.

c. Considerar la señal en tiempo contínuo $x(t) = \sin(2\pi \cdot 100t) + \sin(2\pi \cdot 300t)$ . Hallar su versión discreta para un periodo de muestreo de $T = \frac{1}{1000}$ s, $n \in \{0, 1, \ldots, 199\}$. Describir gráficamente la secuencia resultante.
**Solución:**

```
% ——Calculando la senal discreta x[n] —— %

n=0:199;   % Vector de espacio de muestras.
T=1/2000;  % Frecuencia de muestreo
t=n*T;     % Vector de Tiempo discreto
```

```
x= sin(2*pi*10*t)+sin(2*pi*500*t);

plot(n,f); title('Senal discreta x[n]');
xlabel( 'muestras'); ylabel( 'amplitud');
```

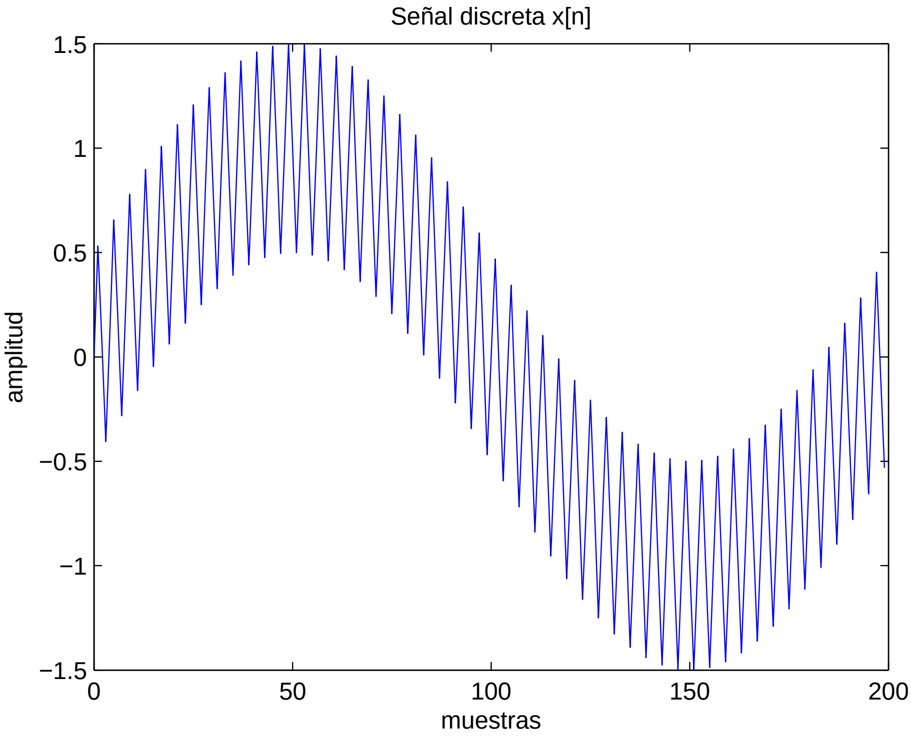



Figura 7: Señal discreta $x[n]$

d. Asumiendo un sistema causal, hallar la respuesta del sistema a las secuencias $x[n]$, usar la funcion **conv()**. Describir gráficamente la secuencia resultante.
**Solución:**

```

% −−−Calculando la senal discreta y[n] −−− %

y=conv(x,h1);
y=y(1:end−20+1); %el Vector resultante es recortado a la misma longitud ...
    de x[n]
fig01= figure;
subplot( 2, 1, 1); plot(n,y); title( 'Respuesta del sistema y[n]');
xlabel( 'muestras'); ylabel( 'amplitud');
subplot( 2, 1, 2); plot(n,x ); title( 'Senal de entrada x[n]');
xlabel( 'muestras'); ylabel( 'amplitud');
```

4. *(3 puntos)* **Se cuenta con un sistema LTI cuya función de transferencia es:**

$$H(z) = \frac{1}{1 - a_0 \cdot z^{-1}}$$

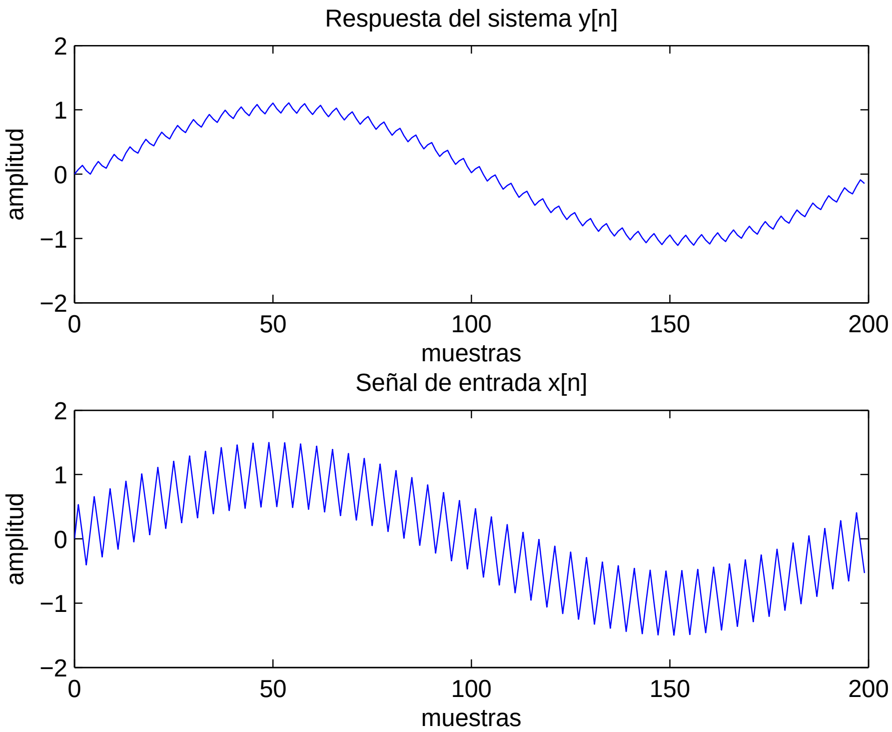



Figura 8: Señal discreta $x[n]$

a. Calcular la respuesta al impulso de $H(z)$ usando la función **impz()**, para 100 muestras. Considerando $a_0 = 1/8$. Describir graficamente. Se trata de un sistema BIBO estable? Se trata de un sistema FIR? Justificar claramente su respuesta.

b. Dada la secuencia $x[n] = (1/10)^n \cdot \sin(2 \cdot \frac{\pi}{10} n)$. Describir graficamente, considerar $n \in \{0, 1, \ldots, 99\}$.

c. Hallar la respuesta del sistema a la secuencia $x[n]$. Usar **conv()**. Describir gráficamente el resultado.

d. Calcular nuevamente la respuesta del sistema a $x[n]$, pero para $a_0 = 1/2$. Describir gráficamente el resultado. Qué se puede afirmar con respecto al nuevo resultado del sistema?