Universidad Nacional Mayor de San Marcos

Facultad de Ciencias Físicas

Procesamiento de Datos Digitales

Apuntes del Curso de Teoría

Prof.: Mg. César Jiménez

Sumilla del Curso

Introducción al lenguaje de programación Matlab. Señales y Sistemas, Sistemas lineales invariantes en el tiempo. Convolución. Sistemas de adquisición de datos. Teorema del Muestreo. Transformada rápida de Fourier y espectros de frecuencia. Filtros analógicos y digitales. Aplicaciones: audio, geofísica, tratamiento de imágenes. Introducción a los wavelets.

2015-II

Prólogo

Esta publicación está conformada por los apuntes del curso de Procesamiento de datos Digitales, dictado en la Facultad de Ciencias Físicas de la UNMSM, durante los semestres 2010-II, 2011-II, 2012-II, 2013-I y 2014-II.

Actualmente, con el desarrollo de las computadoras personales de alta velocidad y rendimiento, se realiza el procesamiento de los datos en forma digital mediante las herramientas computacionales del procesamiento digital de señales. Es importante que un estudiante de física tenga dominio de estas herramientas computacionales con el objetivo de resolver un problema físico específico.

Se hace un amplio uso de herramientas computacionales como el lenguaje Matlab, para el procesamiento de los datos digitales y para su visualización. Es necesario que todo estudiante de Física tenga un dominio de estas herramientas como Matlab y Fortran.

La información contenida en este trabajo es complementaria y referencial, es susceptible de ser revisada, corregida y aumentada. Se invita al estudiante a revisar la bibliografía para obtener un mejor conocimiento de los tópicos correspondientes.

Lima, agosto de 2015

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE CIENCIAS FÍSICAS
SILABO

1. INFORMACION GENERAL

Asignatura	: Procesamiento de Datos Digitales
Carácter	: Electivo
Pre-requisito	: Física Matemática II, Programación.
Créditos	: 04
Horas semanales	: 03 Teoría / 02 Laboratorio
Ciclo Académico	: 2015-II
Duración	: 17 Semanas
Profesor	: Mg. César Jiménez (cjimenezt@unmsm.edu.pe)

2. SUMILLA

Introducción al lenguaje de programación Matlab. Señales y Sistemas, Sistemas lineales invariantes en el tiempo. Convolución. Sistemas de adquisición de datos. Teorema del Muestreo. Transformada discreta de Fourier y espectros de frecuencia. Filtros analógicos y digitales. Aplicaciones: audio, geofísica, tratamiento de imágenes. La transformada wavelet.

3. OBJETIVOS

Estudiar las leyes físicas que gobiernan la dinámica de las señales y sistemas.
Estudiar y aplicar las técnicas del procesamiento digital de señales.
Guiar al estudiante en el tratamiento de datos experimentales y elaboración de informes científicos.
Aprender el manejo del Toolbox de procesamiento de señales de Matlab.
Aplicación del procesamiento de señales: audio y sonido, geofísica, imágenes.

4. METODOLOGÍA

Exposiciones de clases magistrales utilizando pizarra y los medios audiovisuales
Discusión de problemas. Realización de laboratorios.
Manejo de software especializado.

5. CRITERIOS DE EVALUACION

La evaluación del rendimiento de los alumnos es objetiva, en base al promedio de:
PL : Promedio de Laboratorios.
E1 : Nota del primer examen parcial
E2 : Nota del segundo examen parcial

El Promedio Final se calculará de la siguiente forma: $PF = \frac{E1 + E2 + PL}{3} > 10.5$

El alumno podrá rendir un examen sustitutorio, el que será único y abarcará toda la asignatura, cuya nota remplazará a la nota más baja de los exámenes parciales (E1 o E2).

6. CONTENIDO ANALITICO SEMANAL

SEMANA 01: EL LENGUAJE DE PROGRAMACION MATLAB

Tipos de datos. Vectores y matrices. Sentencias de asignación. Iteraciones y bucles. Sentencias de toma de decisiones. Manejo de archivos: lectura y escritura.
Laboratorio 1.

SEMANA 02: SEÑALES Y SISTEMAS

Definición y características de las Señales. Tipos de señales: continuas y discretas, señales periódicas. Energía y potencia de una señal. Funciones singulares. Definición y características de los Sistemas.

SEMANA 03: SISTEMAS LINEALES INVARIANTES EN EL TIEMPO

Características de los sistemas lineales e invariantes en el tiempo (SLIT). Suma de convolución. Integral de convolución. SLIT descrito por ecuaciones diferenciales y de diferencias.
Laboratorio 2.

SEMANA 04: SISTEMAS DE ADQUISICION DE DATOS

Sistemas de adquisición de datos. Teorema del Muestreo. Reconstrucción de una señal a partir de sus muestras. Efecto del submuestreo: Traslape (aliasing).

SEMANA 05: TRANSFORMADA DE LAPLACE

Definición y características de la Transformada de Laplace. Función de transferencia de un sistema en el dominio de la frecuencia. Polos y ceros. Aplicaciones.

SEMANA 06: TRANSFORMADA Z

La Transformada Z. La transformada Z inversa. Propiedades de la Transformada Z. Ecuaciones de diferencias lineales. Representación en diagrama de bloques. Aplicaciones.
Laboratorio 3.

SEMANA 07: SERIE Y TRANSFORMADA DISCRETA DE FOURIER

Serie de Fourier. Definición y características de la Transformada de Fourier. Transformada rápida de Fourier. Aplicaciones.

SEMANA 08: CALIFICACIÓN: Primer Examen Parcial

SEMANA 09: FILTROS ANALOGICOS Y DIGITALES

Definición de filtro. Tipos de filtros: filtro pasa-bajo, filtro pasa-alto, filtro elimina-banda, filtro pasa-banda. Manejo del Toolbox de Procesamiento de Señales de Matlab. Polos y ceros. Filtro mediano. Filtro Butterworth. Aplicaciones.
Laboratorio 4.

SEMANA 10: SEÑALES EN EL DOMINIO DEL TIEMPO Y DE LA FRECUENCIA

Análisis en el dominio del tiempo y de la frecuencia. Espectro de frecuencias de una señal. Cálculo de periodos de retorno. Aplicaciones.

SEMANA 11: PROCESAMIENTO DIGITAL DE SEÑALES DE AUDIO
Software para adquisición de señales de voz y audio. Reconocimiento de locutor.

SEMANA 12: PROCESAMIENTO DIGITAL DE SEÑALES DE AUDIO
Fundamentos de acústica. Diseño e implementación de un sonómetro de bajo costo.

SEMANA 13: PROCESAMIENTO DIGITAL DE SEÑALES GEOFÍSICAS
Señales sísmicas. Algoritmos para determinar las fases sísmicas P y S. Algoritmos para determinar la amplitud y frecuencia de una señal.

SEMANA 14: PROCESAMIENTO DIGITAL DE SEÑALES GEOFÍSICAS
Señales mareográficas. Interfaz gráfica de usuario en Matlab. Series de tiempo de temperatura: cálculo de parámetros.
Laboratorio 5.

SEMANA 15: PROCESAMIENTO DIGITAL DE IMÁGENES
Definición de una imagen. El pixel. Formatos de imágenes. Lectura y escritura de imágenes en Matlab. Transformaciones lineales: traslación, rotación, zoom. Ecualización de histograma. Filtros. Transformada de Fourier 2D.

SEMANA 16: TRANSFORMADA WAVELET
Definición de wavelets. Tipos de wavelets. Manejo de la transformada wavelet con Matlab. Aplicaciones al análisis de señales.

SEMANA 17: EVALUACIÓN: Examen Final. Examen Sustitutorio

LABORATORIOS
1. Programación y manejo de Matlab.
2. Sistemas lineales invariantes en el tiempo.
3. Transformadas. Señales en el dominio de la frecuencia.
4. Filtros digitales.
5. Procesamiento digital de señales geofísicas.

BIBLIOGRAFÍA

Oppenheim, A. Análisis de Señales y Sistemas.

Proakis – Manolakis. Tratamiento Digital de Señales.

Nava, A. Procesamiento de Series de Tiempo.

Ogata. Ingeniería de Control Moderna.

Página web con información del curso: http://fenlab.9k.com/pds

INTRODUCCIÓN A MATLAB

Matlab es un acrónimo que proviene de Matrix Laboratory. Matlab es una herramienta de cálculo técnico, simulación y visualización. Es un lenguaje interpretado de alto funcionamiento para computación técnica. Este integra computación, visualización y programación en un entorno fácil de usar, donde los problemas y las soluciones son expresados en la más familiar notación matemática. Los usos más familiares de Matlab son:

- Matemática y Computación
- Desarrollo de algoritmos
- Modelamiento, simulación
- Análisis de datos, exploración y visualización
- Gráficos científicos e ingenieriles
- Desarrollo de aplicaciones, incluyendo construcción de interfaces graficas de usuario

Matlab presenta una familia de soluciones a aplicaciones específicas de acoplamiento rápido llamadas ToolBoxes. Los toolboxes son colecciones muy comprensibles de funciones Matlab, o archivos de Matlab (M-files) que extienden el entorno de Matlab para resolver clases particulares de problemas. Algunas áreas, entre otras, en las cuales existen toolboxes disponibles son:

- Procesamiento de señales
- Sistemas de control
- Redes neuronales
- Lógica difusa
- Wavelets
- Simulación

Lo importante en esta presentación es el procesamiento de datos digitales, por ende, utilizaremos ampliamente una aplicación de Matlab: el toolbox de procesamiento digital de señales. Sin embargo, se debe conocer algunas sentencias y comandos básicos, que permitirá seguir paso a paso nuestro objetivo:
Líneas de comandos (prompt): >>

Sentencias básicas

dir	% listado de archivos
pwd	% ver directorio actual
cd	% cambiar de directorio
clc	% borrar pantalla
clear	% borra el espacio de trabajo
whos	% visualiza las variables que están en el espacio de trabajo que hay en Matlab
cd ..	% retrocede al directorio raíz
close	% cierra ventanas y ploteos
help	% proporciona ayuda o información sobre algún comando.

% comentario: este símbolo antepuesto a cualquier escrito o comando es reconocido por Matlab como un comentario.

Sentencia de asignación
>> a=5;
>> X = [1 5 3 4 6];   % vector fila
>> Y = [1; 5; 3; 4; 6];   % vector columna
Ejemplo:

Para obtener la matriz: $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ se tiene que aplicar la siguiente sentencia:

>>A = [1 2 3; 4 5 6; 7 8 9];

Nota: Si al finalizar la sentencia, no se coloca el punto y coma (;), Matlab mostrará en la línea de comandos el contenido de la variable del espacio de trabajo, es decir, en nuestro caso mostrará la forma de la matriz. Por lo tanto, es recomendable en la mayoría de los casos colocar un punto y coma (;) después de cada sentencia.

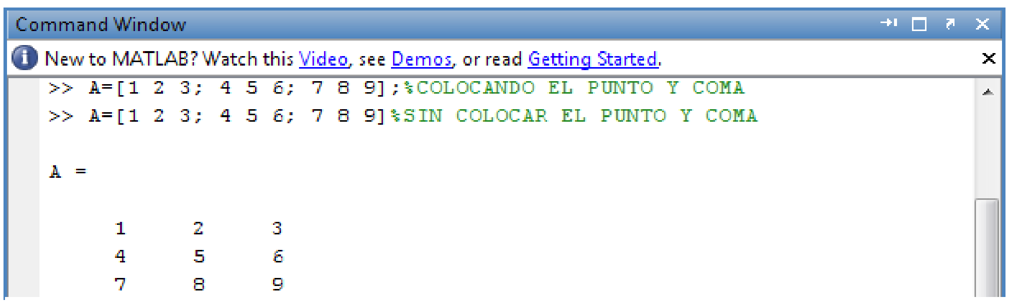


Fig. 1.1 Matriz cuadrada A

Constantes en Matlab
>> pi   % 3.141592
>> i   % $\sqrt{-1}$
>> j   % $\sqrt{-1}$
>> eps   % 2.2204e-016

Funciones matemáticas incorporadas
>> sqrt(4)   % raíz cuadrada
>> log   % logaritmo natural
>> log10   % logaritmo decimal
>> exp   % exponencial
>> sin   % función seno trigonométrico
>> cos   % función coseno trigonométrico
>> tan   % función tangente trigonométrica

En la Fig. 1.2 se muestra un programa sencillo para observar la sintaxis de Matlab.

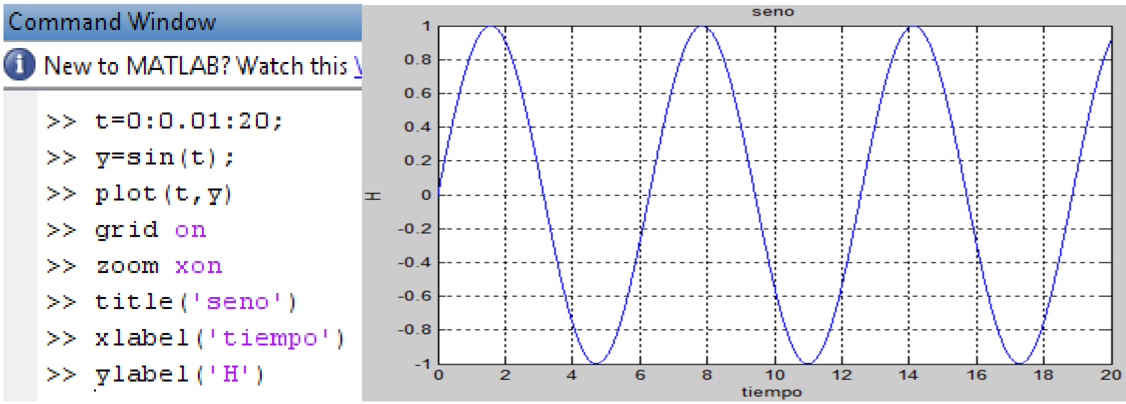


Fig. 1.2 Ejemplo de un programa simple en Matlab.

LECTURA Y ESCRITURA DE ARCHIVOS DE DATOS

Sea el archivo de datos: “datos.txt”, en formato ASCII o de texto:

1.0	2.0	3.0
4.0	5.0	6.0
7.0	8.0	9.0

Para leer dicho archivo de datos en Matlab:
>> load datos.txt; % la matriz de datos se asigna a la variable “datos”
Otra forma de sintaxis es:
>> A = load(‘datos.txt’);   % la matriz de datos se asigna a la variable “A”

Para escribir las variables “t” y “y” en el archivo “salida.txt” se escribe el comando:
>> save salida.txt t y –ascii   % en formato Ascii o texto
>> save salida.mat t y –mat   % en formato binario de Matlab

Ejemplo: Sea un ejemplo aritmético sencillo, la suma de dos números enteros: a+b=c
El ejercicio es muy simple, pero complicado para ordenarle a la computadora que lo resuelva, si no tenemos un lenguaje común con ella. Para resolver el problema se debe realizar una serie de instrucciones que se pueden traducir en un programa en MATLAB que sume los números a y b, de la siguiente manera:

```
%  Pide que se le proporcione un número y lo almacena en a
                a = input (‘Variable a = ‘)
%  pide que se le proporcione otro número y lo almacena en b
                b = input (‘Variable b = ‘)
%  sumar a + b y lo almacena en c.
                c = a + b;
%  muestra al usuario el resultado de la operación
               fprintf ('%4.2f  %4.2f %4.2f \n',a,b,c);
               disp (c)
% graba los datos en un archivo de texto
```

```
        save salida.txt a b c -ascii
%  Visualiza el contenido del archivo salida.txt:
        type salida.txt
```

PROGRAMACIÓN EN MATLAB

SENTENCIAS SELECTIVAS

Sentencia IF

```
if <condición>
    orden 1
    else
    orden 2
end
```

Ejemplo: ecuación de 2do grado para soluciones reales.

```
a = input ('Valor de a = ');
b = input ('Valor de b = ');
c = input ('Valor de c = ');
if (a == 0)
    disp ('ecuacion de 1er grado');
    r1 = -b/c
end
if (a ~= 0)
    d = b*b-4*a*c;
    p = -b/(2*a);
    q = sqrt(d)/(2*a);

    if d > 0
        disp ('raices reales')
        r1 = p+q;
        fprintf ('%s %8.6f \n','r1 = ',r1);
        r2 = p-q
    end
end
```

Nota: se debe evitar los bucles infinitos al usar el comando IF.

SENTENCIAS ITERATIVAS

Sentencia FOR

Permite repetir las declaraciones un número finito de veces:

```
for (variable)=expresión
    orden 1
    orden 2
end
```

Ejemplo

```
disp ('Sentencia For')
for k = 1:1:10
   p = k*k;
   fprintf ('%4.0f %6.2f \n', k, p);
end
```

Nota: No es aconsejable utilizar el bucle FOR para un número grande de iteraciones, debido a que Matlab es un lenguaje interpretado, es decir, el programa lee el comando: lo interpreta y lo ejecuta. Entonces para iteraciones muy grandes con el bucle FOR, el programa puede ejecutarse demasiado lento.

Sentencia WHILE

Mientras la expresión es verdadera siempre se ejecuta las órdenes, esto termina, si la expresión es falsa, ademas no se sabe cuantas veces se ejecuta el bucle.

while <condición>
    orden 1
    orden 2
end

Ejemplo:

```
disp ('Sentencia While')
k = 1;
while (k<11)
   p = k*k;
   fprintf ('%4.0f %6.2f \n', k, p);
   k = k+1;
end
```

Sentencia BREAK

Termina la ejecución del bucle For o While

Archivo .m

Son archivos de texto o “scripts” ejecutables desde Matlab. Contienen un conjunto de comandos o sentencias. Para ejecutarlos simplemente se escribe el nombre del archivo en la línea de comandos de Matlab.

>> archivo <Enter>

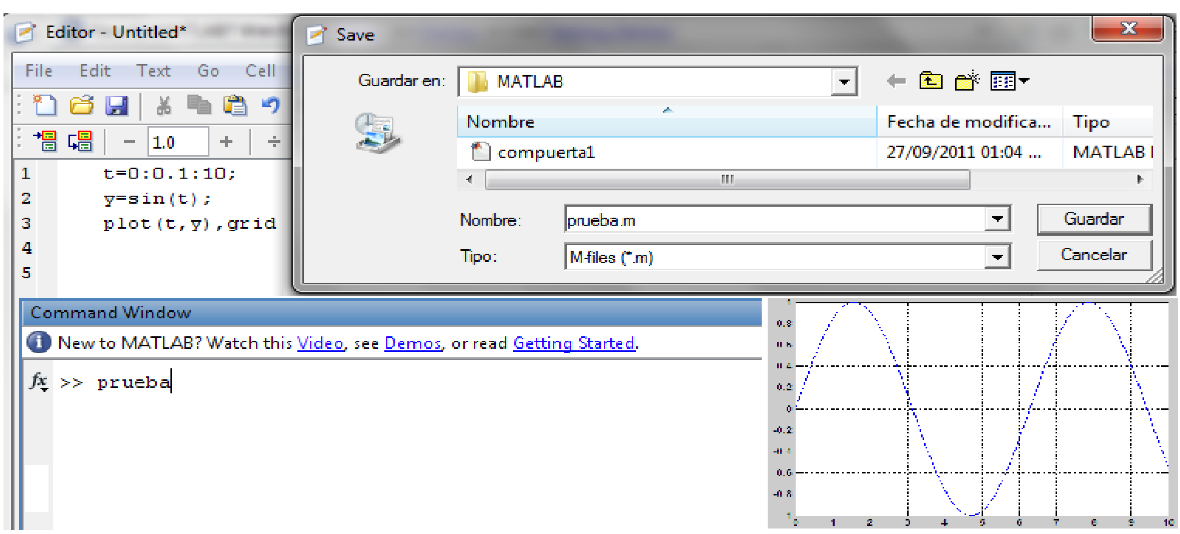


Fig. 1.3 Modo de grabar un archivo m

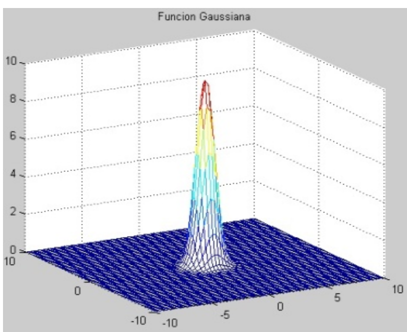
Función en Matlab

Editor - D:\Fernando\Documents\MATLAB\p.m

```
function p=p(n,t)
switch n
    case 1 %este será para el impulso unitario
        p=1.*(t>-eps)-1.*(t>eps);
    case 2 %Este será para el escalon unitario
        p=1.*(t>=0);
    case 3 %Este será la función rampa
        p=t.*(t>=0);
end
```

Gráficas en 3D

[X Y] =meshgrid(x,y); % crea una grilla o matriz donde x,y son vectores previamente definidos
mesh(X, Y, Z) % crea la grafica en 3D




```
%GRAFICA EN 3D DE LA FUNCION GAUSSIANA
clear,clc
A=input('ingrese la ampitud de la curva =');
[x,y]=meshgrid(-10:0.3:10);
z=A*exp(-x.^2-y.^2);
mesh(x,y,z)
title('Funcion Gaussiana')
```

Fig. 1.4 Ejemplo de grafica en 3D

SEÑALES Y SISTEMAS

Sistema de adquisición de datos (DSP)

Es el proceso de adquirir señales de fenómenos del mundo real, este proceso se puede esquematizar de la siguiente manera:

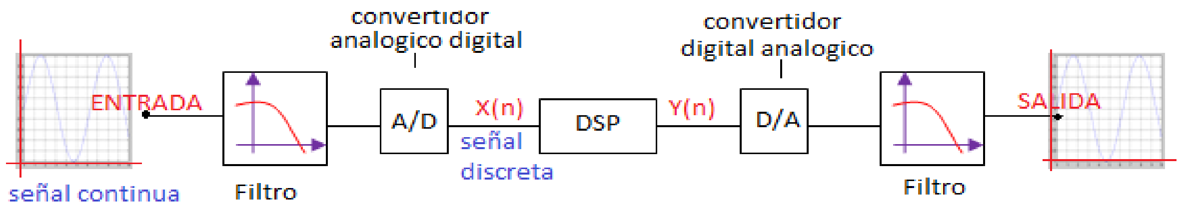


Fig. 2.1 Diagrama de bloques de un sistema de adquisición de datos

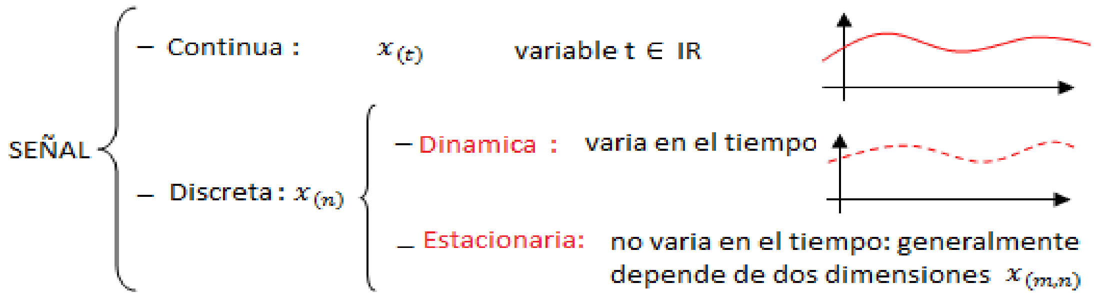
Como primer paso se da el acondicionamiento de la señal, una consideración importante para medidas analógicas es conocer las características del sensor y la señal con el fin de conocer si es necesario el acondicionamiento para medir la señal correctamente, dentro de esta etapa se encuentra los filtros, ya sea pasa baja o pasa alta, para eliminar todo tipo de ruidos que podrían interferir con nuestra medida, siguiendo este proceso es necesario pasar la señal analógica a digital; el procesamiento digital tiene muchas ventajas respecto del procesamiento analógico, por ende, luego de la conversión, la señal pasaría por el DSP (Digital Signal Processing) que viene a ser el procesamiento de la señal digital. Como es sabido, la señal de salida (procesada) es una señal discreta, por ello, tiene que pasar nuevamente por un convertidor; en este caso será convertidor digital-analógico, y para evitar que la señal de salida tenga ruido, nuevamente se pasa por un filtro. Es así, como obtenemos una buena medida y por ende una mejor interpretación de los fenómenos del mundo real.

¿Por qué usar DSP? (Digital Signal Processing)

PARAMETRO	Procesamiento analógico	Procesamiento digital
Precisión	1% a 10%	2^{-16} 2^{-14}
Efectos	Temperatura, humedad, ruido	Redondeo o truncamiento
Cota o tamaño	Elevado	Bajo / mediano
Compatibilidad	Media	----------------------
S/N	50 a 60 dB	> 100 dB
Consumo de potencia	Manual	No necesario o digital
Señales en 2D ó 3D	No procesable	Procesable

Tabla 2.1

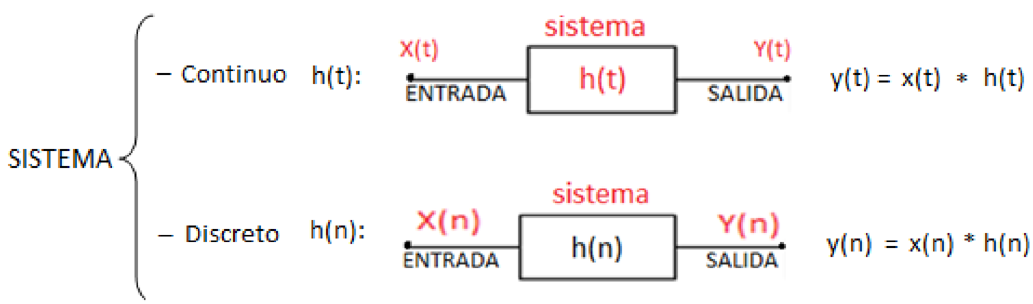
SEÑAL.- Una señal es una magnitud física susceptible de ser medida, es un conjunto de datos que representa una variable física. Una señal contiene información.



RUIDO.- Es la parte de la señal que no lleva información relevante. Puede ser de pequeña o gran amplitud y de periodo corto o periodo largo.

Relación Señal/Ruido: S/N

SISTEMA: Es un conjunto de procesos que realiza una acción sobre una o más señales.

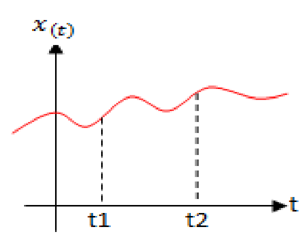


Donde: *h(t)* o *h(n)* representa la función de transferencia del sistema. Además el símbolo asterisco * representa la operación de convolución en el dominio del tiempo.

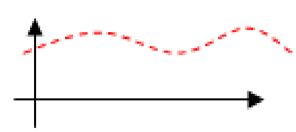
CONTENIDO ENERGÉTICO Y POTENCIA DE UNA SEÑAL

ENERGIA DE LA SEÑAL

Para la señal continua: $E = \int_{t1}^{t2} [x(t)]^2 dt$



Para una señal discreta: $E = \sum_{N1}^{N2} [x(n)]^2$



Nota 1: Las unidades de energía y potencia de una señal no necesariamente serán las correspondientes a Joule y Vatio.

Nota 2: En Matlab la energía de una señal se calcula de la forma siguiente:

```
>> x = [-1 0 1 2 5 -2];
>> E = sum(x.*x)
>> E = 35
```

Ejemplo: para una señal sísmica es importante conocer la magnitud del terremoto entonces a partir del contenido energético de la señal sísmica es posible calcular la magnitud del terremoto.

APLICACIÓN

Cálculo de la magnitud de un sismo a partir del contenido energético de la señal sísmica.

$M = f(E, D, H)$: La magnitud M es una función de la energía (contenido energético de la señal), de la distancia epicentral D (distancia del centro hasta el lugar donde está el sismógrafo) y también de la profundidad H, como sigue en la siguiente ecuación

$$M = a\log(E) + b\log(D) + c\log(H) + d$$

Entonces, si se calcula cada parámetro teniendo en cuenta que a, b c y d deben ser tabulados mediante una regresión multilineal, podremos calcular la magnitud de la señal sísmica.

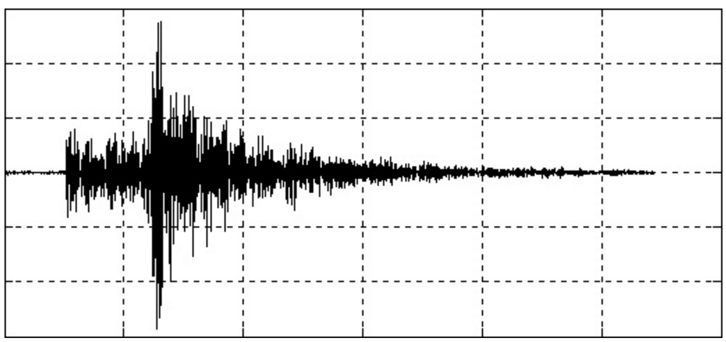


Fig. 2.2 Ejemplo de señal sísmica: amplitud vs. tiempo.

POTENCIA DE LA SEÑAL

Una señal es una serie de tiempo que viene a ser un conjunto de valores que determinan como varía una señal, entonces si medimos el contenido energético de una señal por intervalo de tiempo, este resultado se definirá como la potencia:

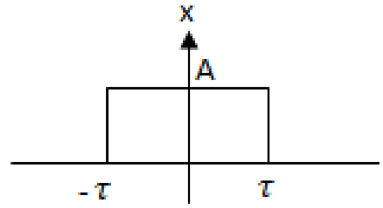
a) Señal continua: $$P = \frac{1}{t_2 - t_1}\int_{t_1}^{t_2} x^2(t)dt$$

b) Señal discreta: $$P = \frac{1}{N_2 - N_1 + 1}\sum_{n=N_1}^{N_2} x^2(n)$$

Nota: en particular, una señal de energía finita, tiene potencia cero. Mientras que una señal de potencia finita, tiene energía infinita.

Ejemplo:

a) Hallar la energía total del pulso rectangular:



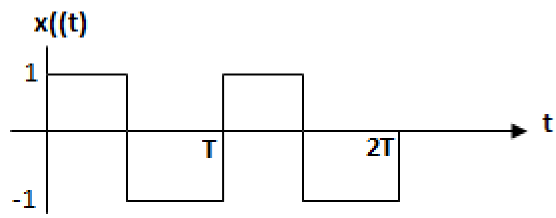
$$x(t) = \begin{cases} A, & -\tau \le t \le \tau \\ 0, & |t| > \tau \end{cases}$$

$$E = \int_{-\tau}^{\tau} A^2 dt = 2A^2\tau$$

$$P = \frac{2A^2\tau}{2\tau} = A^2$$

b) Hallar la potencia promedio de la onda cuadrada:

- para un periodo



$$E = \int_0^T x^2(t)dt = \int_0^{T/2} (1)^2 dt + \int_{T/2}^{T} (-1)^2 .dt$$

$$E = \frac{T}{2} + \frac{T}{2} = T$$

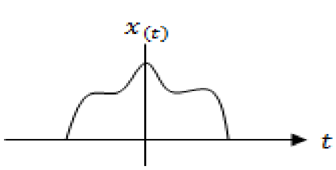
$$P = \frac{1}{T}\int_0^T x^2(t)dt = \frac{T}{T} = 1$$

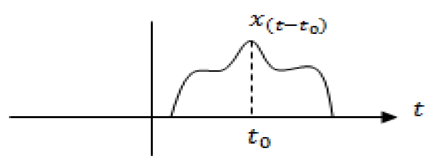
TRANSFORMACION DE LA VARIABLE INDEPENDIENTE

Caso continuo: variable independiente "t"

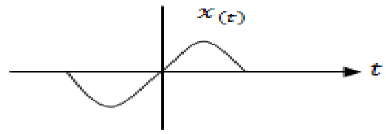
Caso discreto: variable independiente "n"

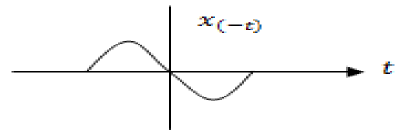
a) Desplazamiento en el tiempo: $t \rightarrow t - t_0$



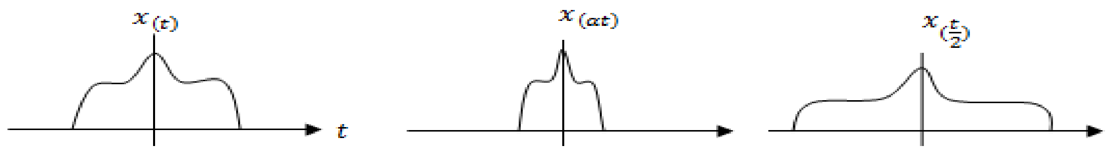


b) Inversión de tiempo: $t \rightarrow -t$





c) Escalamiento en el tiempo: $t \rightarrow \alpha t$



Si α > 1 entonces la señal se comprime
Si α < 1 entonces la señal se expande

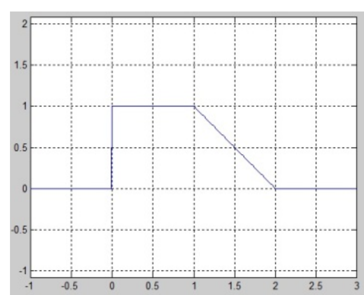
d) Transformación lineal: $t \rightarrow \alpha t + \beta$

Es un caso general que considera a los casos anteriores, pero hay que tener en cuenta, que para utilizarlo correctamente lo que se debe hacer primero es factorizar la expresión, luego hacer un desplazamiento en el tiempo, para después expandir (hacer un escalamiento).

$$\alpha t + \beta = \alpha(t + \frac{\beta}{\alpha})$$

Ejemplo:

Sea *x(t)* la función mostrada en el siguiente grafico:

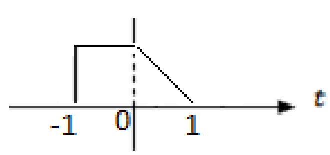


Realizar las siguientes transformaciones y graficar:

a) x(t+1)
b) x(-t+1)
c) $x(\frac{3}{2}t)$

Solución a)

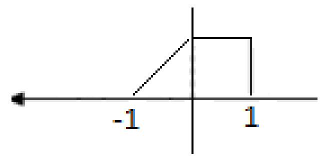
En este caso la función solo está desplazada en el tiempo una unidad hacia la izquierda.



Solución b)

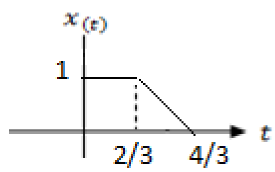
Teniendo en cuenta el caso general, procedemos primero con la factorización

d) $x(-t+1) = x(-(t-1))$



Solución c)

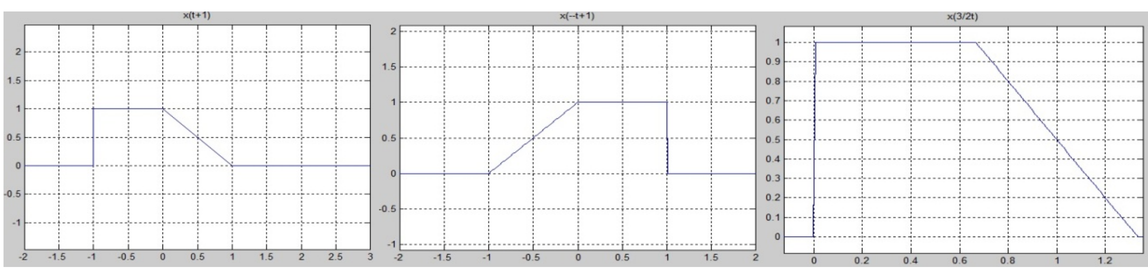
En este caso: $\alpha = \dfrac{3}{2} > 1$, entonces la función se comprime.



Nota: todas estas transformaciones también se pueden realizar en Matlab.

Por ejemplo:

a) b) c)




```
function x=x(t)
x=1.*(t>=0&t<1)-(t-2).*(t>=1&t<2);
end
```

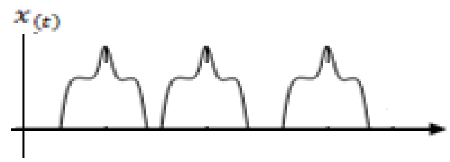
```
>> t=-2:0.01:3;
>> y=x(t+1);
>> plot(t,y),grid on,axis equal
>> title('x(t+1)')
```

```
>> t=-2:0.01:2;
>> y=x(-t+1);
>> plot(t,y),grid on,axis equal
>> title('x(-t+1)')
```

```
>> t=-0.5:0.01:1.5;
>> y=x(1.5.*t);
>> plot(t,y),grid on
>> axis equal
```

SEÑALES PERIÓDICAS

$x(t) \rightarrow x(t+mT)$, donde T = periodo de la señal y m es un número entero.



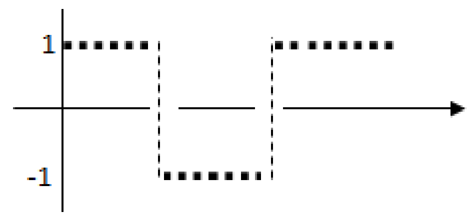
Las funciones senos y cosenos así como sus combinaciones lineales son periódicas.

$$x(t) = Asen(\omega t) + B\cos(\omega t)$$

$$x(t) = X_0 sen(\omega t + \alpha)$$

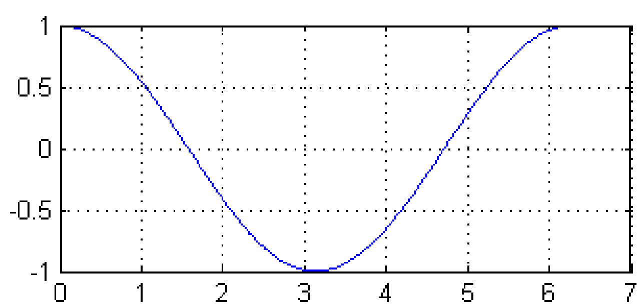
En el caso discreto:

$x(n) = x(n + N_0)$, donde $\mathbf{N_0}$ = periodo fundamental

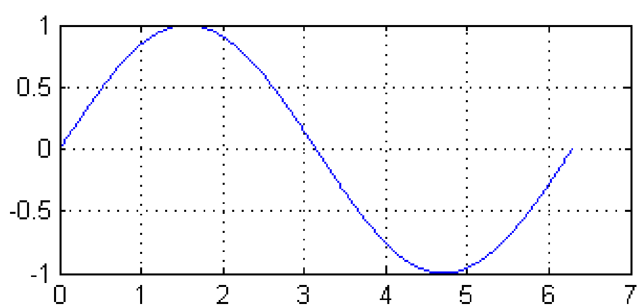


SEÑALES PAR E IMPAR

Una señal es par si: *x(t)= x(-t)*. Ejemplo: la función coseno.



Una señal es impar si: *x(t) = -x(-t)*. Ejemplo: la función seno.



NOTA 1: Una señal impar no necesariamente pasa por el origen de coordenadas.

NOTA 2: Una señal impar es invariante ante una reflexión especular en torno al eje vertical.

NOTA 3: Una señal cualquiera se puede descomponer en su parte PAR e IMPAR:

- $\text{Par}\{x(t)\} = \frac{1}{2}[x(t) + x(-t)]$
- $\text{Impar}\{x(t)\} = \frac{1}{2}[x(t) - x(-t)]$

Sumando ambas expresiones tenemos: $x(t) = Par\{x(t)\} + Impar\{x(t)\}$

SEÑAL EXPONENCIAL COMPLEJA

$x(t) = Ae^{\alpha t}$, donde $\alpha = a + bj$, remplazando α en la función tenemos:

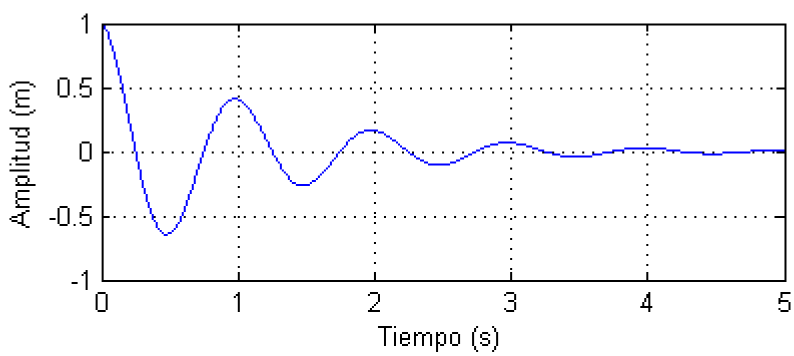
$x(t) = Ae^{(a+jb)t} = Ae^{at}e^{jbt}$

$x(t) = Ae^{at}[\cos(bt) + j\,sen(bt)]$

Ejemplo: un movimiento oscilatorio amortiguado

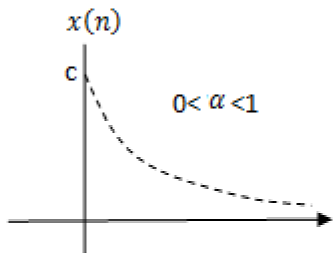
Si a<0 entonces hay amortiguación

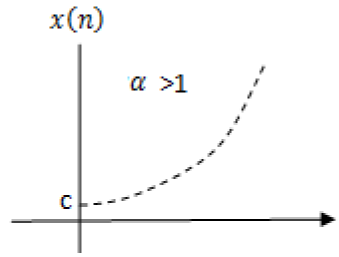
b es la frecuencia angular de oscilación



SEÑAL EXPONENCIAL DISCRETA

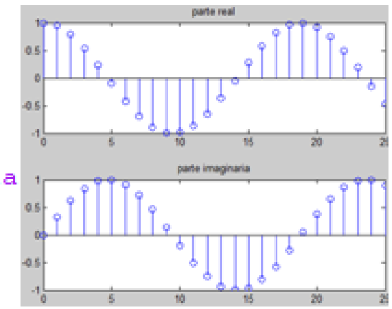
En el caso discreto: $x(n) = ca^n$





En Matlab

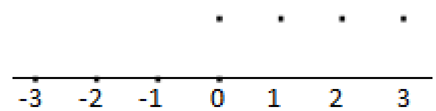
```
nn=[0:25];
xx=exp(j.*(nn/3));
subplot(2,1,1),stem(nn,real(xx)),title('parte real')
subplot(2,1,2),stem(nn,imag(xx)),title('parte imaginaria
```



FUNCIONES SINGULARES

1. Impulso unitario discreto: $\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$

2. Escalón unitario discreto: $u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$



NOTA: El impulso unitario discreto es la diferencia del escalón unitario discreto:

$$\delta(n) = u(n) - u(n-1)$$

NOTA: el escalón unitario discreto es la sumatoria de la muestra o impulso unitario:

$$u(n) = \sum_{m=0}^{\infty} \delta(m)$$

De manera equivalente: $u(n) = \sum_{k=n}^{\infty} \delta(n-k)$

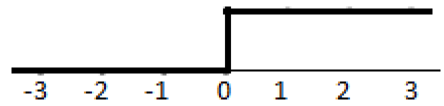
Que sería una superposición de impulsos unitarios retrasados en el tiempo.

Propiedad del muestreo: $x(n)\delta(n) = x(0)\delta(n) = x(0)$

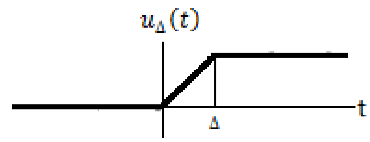
En general: $x(n)\delta(n-n_0) = x(n_0)\delta(n-n_0) = x(n_0)$

3. Escalón unitario e impulso unitario continuo:

La función escalón unitario: $u(t)=\begin{cases}1, & t\geq 0\\ 0, & t<0\end{cases}$



La relación entre el impulso $\delta(t)$ y el escalón unitario $u(t)$ es:



$$\delta(t)=\frac{du(t)}{dt} \quad => \quad u(t)=\int_{-\infty}^{\infty}\delta(t)dt$$

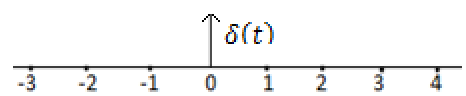
Aproximación del escalón unitario:

$$\delta_\Delta(t)=\frac{du_\Delta(t)}{dt} \qquad\qquad u_\Delta(t)=\lim_{\Delta\to 0} u_\Delta(t)$$

NOTA: La función impulso unitario $\delta_\Delta(t)$ es un pulso corto de duración **Δ** y con un área unitaria para cualquier valor de delta. En el límite $\Delta\to 0$

$$\delta(t)=\lim_{\Delta\to 0}\delta_\Delta(t)$$

Un impulso escalado $k\delta(t)$ tendrá un área *k*:



$$\int_{-\infty}^{t}k\,\delta(t)dt=ku(t) \quad => \quad u(t)=\int_{-\infty}^{t}\delta(t)dt \quad => \quad u(t)=\int_{0}^{\infty}\delta(t-\tau)d\tau$$

Propiedades:

1) Propiedad de muestreo: $x(t)\delta(t)=x(0)\delta(t)=x(0)$

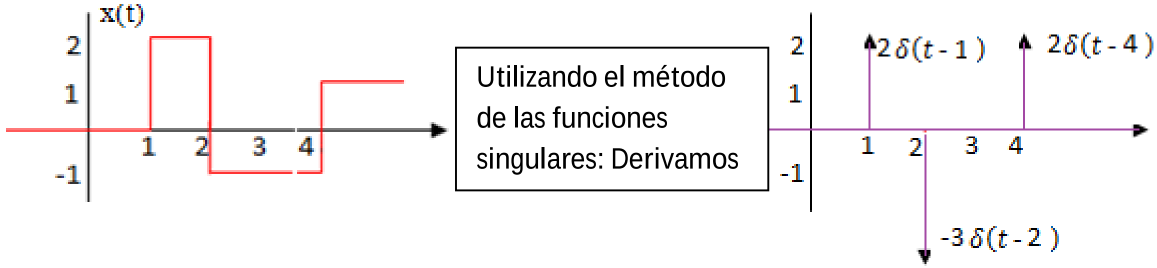
   En general: $x(t)\delta(t-t_0)=x(t_0)$

2) $\int_{-\infty}^{\infty}x(t)\delta(t)dt=x(0)$

3) $\int_{-\infty}^{\infty}x(t)\delta(t-t_0)\,dt=x(t_0)$

4) $\delta(at)=\frac{1}{|a|}\delta(t)$

Ejemplo 1: Sea *x(t)* la función mostrada en la grafica. Hallar la derivada de *x(t)* usando funciones singulares.



Por lo tanto:

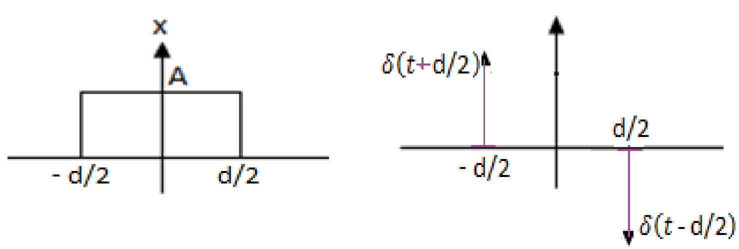
$$x'(t) = 2\delta(t-1) - 3\delta(t-2) + 2\delta(t-4)$$

Ejemplo 2: Función compuerta unitaria (función Gate):

$$G(t) = \begin{cases} 1, & t \leq |d/2| \\ 0, & t > |d/2| \end{cases}$$

Hallar la derivada de la función compuerta unitaria $G'(t)$ en función de impulsos unitarios:

Solución: como en el caso anterior utilizaremos el método de las funciones singulares:



$$G'(t) = \delta(t + d/2) - \delta(t - d/2)$$

SISTEMAS CONTÍNUOS Y DISCRETOS

Los sistemas físicos son una interconexión de componentes, dispositivos y subsistemas. Un sistema puede considerarse como un proceso en el cual las señales de entrada son transformadas por el sistema, o provocan que este, responda de alguna forma, lo que da como resultado otras señales como salida.

Donde: podemos identificar a las siguientes variables:

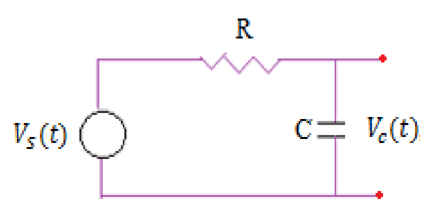
x: Señal de entrada (input)
y: Señal de salida (output)
h: Función de transferencia del sistema, conocido también como función de Green, que es el conjunto de parámetros que caracterizan al sistema considerado. También se le conoce como respuesta al impulso.

Por ejemplo: para un sistema eléctrico.

Tenemos un circuito eléctrico que consta de una resistencia (*R*) y un condensador (*C*), además de una fuente de tensión alterna $v_s(t)$ como muestra la figura. Vemos también que entre los terminales del condensador hay una diferencia de potencial $v_c(t)$, que no es otra cosa que el voltaje en el condensador y será también la señal de salida. Entonces podemos calcular la función de transferencia del sistema, mediante la ecuación diferencial que relaciona al voltaje de entrada y de salida.

Entonces, para obtener la función de transferencia nos referiremos a una ecuación diferencial de la ley que gobierna al sistema (por ejemplo, la ley de Kirchhoff, la ley de mallas, etc.).

$$i(t) = \frac{v_s(t) - v_c(t)}{R} \qquad (1)$$



Para el capacitor:

$$i(t) = C\frac{dv_c(t)}{dt} \qquad (2)$$

Entonces igualando (1) = (2):

$$\frac{dv_c(t)}{dt} + \frac{1}{RC}v_c(t) = \frac{1}{RC}v_s(t)$$

Esta es la ecuación diferencial que gobierna a dicho sistema eléctrico, de allí se puede obtener la función de transferencia del sistema; por el momento no resolveremos esta ecuación, pero si debemos tener las siguientes consideraciones:

1. Para obtener la función de transferencia de cualquier sistema, primero se debe encontrar la ecuación diferencial que relacione la señal de entrada y la señal de salida.

2. Luego se debe aplicar la transformada de Laplace con condiciones iniciales nulas, es decir, llevar del dominio del tiempo al dominio de la frecuencia aplicando la transformada de Laplace.

3. La función de transferencia del sistema será la razón de la salida sobre la entrada, en el dominio de Laplace.

NOTAS
1. La función de transferencia del sistema en el dominio de la frecuencia es:

$$H(s) = \frac{Y(s)}{X(s)}$$

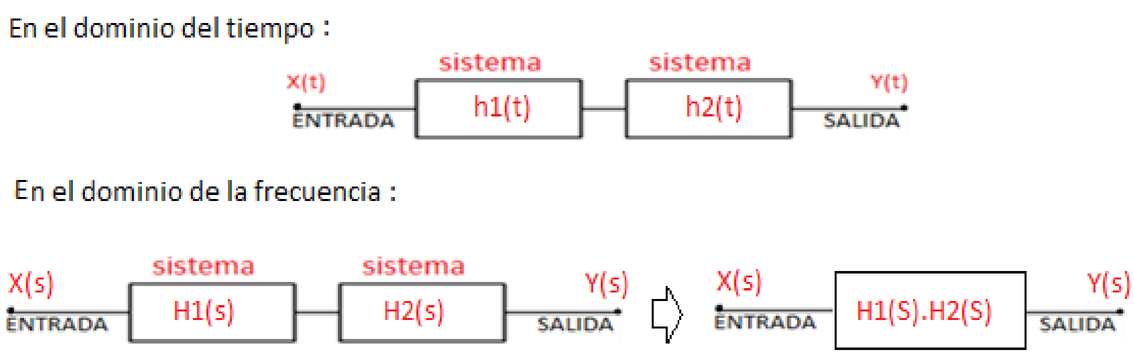
Donde s es la variable de la transformada de Laplace, en general es un número complejo.

2. En general, un sistema puede ser de tipo mecánico, eléctrico, nuclear, químico, económico, financiero, social, etc. Lo importante es conocer la función de transferencia de dicho sistema.

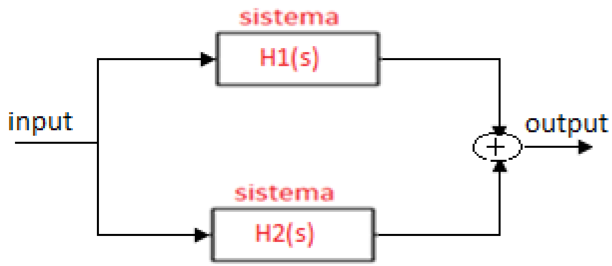
3. Los sistemas tienen propiedades y estructuras que se pueden explotar para tener información sobre su comportamiento y desarrollar herramientas efectivas para su análisis.

INTERCONEXION DE SISTEMAS

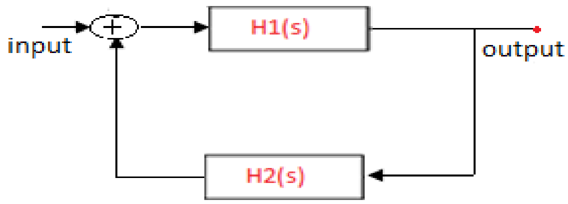
a) Serie o cascada:



b) En paralelo:



c) Interconexión con retroalimentación:



PROPIEDADES BÁSICAS DE LOS SISTEMAS

1) Sistemas con y sin memoria: un sistema es "sin memoria" si su salida para cada valor de la variable independiente en un tiempo dado, solo depende de la entrada en ese mismo instante.

Sistemas sin memoria:
a) $y(n) = (2x(n) - x^2(n))^2$ sin memoria
b) Un resistor: $y(t) = Rx(t)$ sin memoria
c) Sistema identidad: $y(n) = x(n)$ sin memoria

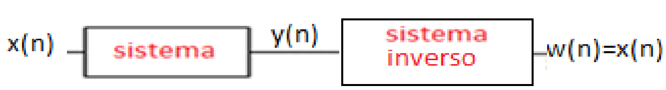
Sistema con memoria:

a) Un capacitor: $y(t) = \frac{1}{C}\int_{-\infty}^{t} x(t)dt$

b) Un acumulador o sumador: $y(n) = \sum_{k=-\infty}^{n} x(k)$

c) Un retraso (delay): $y(n) = x(n-1)$

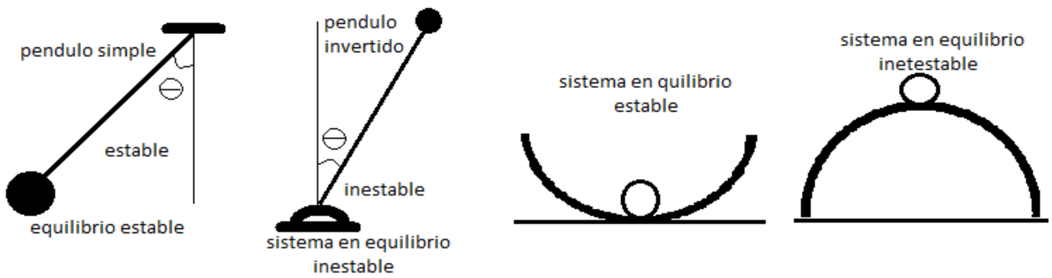
2) Invertibilidad y sistemas inversos:



Ejemplo:
a) Modem = modulador o demodulador
b) Códec = codificador o decodificador

3) Causalidad: un sistema es causal si su salida en cualquier instante de tiempo depende solo de los valores de la entrada en el momento presente y en el pasado (sistema no anticipativo). Ejemplo: Circuito RC, el movimiento de un automóvil.

4) Estabilidad: un sistema estable es aquel en el que entradas pequeñas conducen a respuestas que no divergen.

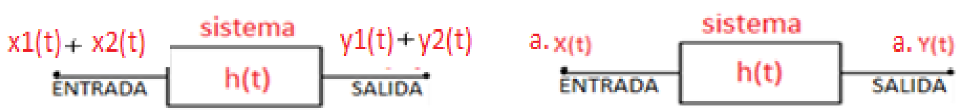


5) Invariancia en el tiempo: un sistema es invariante en el tiempo, si el comportamiento o propiedades de dicho sistema no cambien o están fijos en el tiempo.

$$x(n) \rightarrow y(n)$$

$$x(n-n_0) \rightarrow y(n-n_0)$$

6) Linealidad: un sistema es lineal si puede aplicarse la importante propiedad de la superposición:



Nota: para sistemas lineales, si la entrada es cero entonces la salida debe ser necesariamente cero.

Ejemplo: El sistema descrito por la ecuación $y(t) = 3x(t) + 2$ no es lineal.

SISTEMAS LINEALES E INVARIANTES EN EL TIEMPO (SLIT)

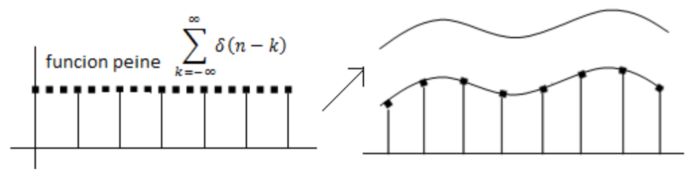
Un sistema es SLIT, si además de cumplir la importante propiedad de la superposición, su comportamiento y propiedades no cambian o están fijos en el tiempo.

Una de las características más importantes del impulso unitario es que las señales en general, pueden representarse como una combinación lineal de impulsos unitarios.

$$x(n) = \sum_{k=0}^{N} x(n)\delta(n-k)$$

La suma de convolución.- Es la representación de señales discretas en función de impulsos unitarios:

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k)$$



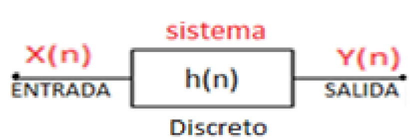
Comentario: esto corresponde a la representación de una secuencia arbitraria como una combinación lineal de impulsos unitarios desplazados $\delta(n-k)$ donde los pesos ponderados son los $x(k)$. La ecuación anterior representa la PROPIEDAD DE SELECCIÓN del impulso unitario discreto.

Para el escalón unitario: $x(n) = u(n)$: $u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$

$$u(n) = \sum_{k=-\infty}^{\infty} u(k)\delta(n-k) = \sum_{k=0}^{\infty} \delta(n-k) \quad \rightarrow \quad u(n) = \sum_{k=0}^{\infty} \delta(n-k)$$

FUNCION DE TRANSFERENCIA DEL SISTEMA

Viene a ser la respuesta del sistema cuando la entrada es un impulso unitario. También se le conoce como función de Green.



En general: $y(n) = x(n) * h(n)$

Si la entrada es el impulso unitario: $x(n) = \delta(n)$

Entonces: $y(n) = \delta(n) * h(n) \rightarrow y(n) = h(n)$

Donde:

x(n): Entrada (input).

y(n): Salida (output).
h(n): Función de transferencia del sistema o función de Green.

Si *x(n)* es una señal arbitraria, entonces: $y(n) = \sum_{k=-\infty}^{\infty} x(k) h_k(n)$

Donde: $h_k(n)$ representa la respuesta del sistema al impulso unitario desplazado $\delta(n-k)$.

Por ejemplo, como señal de entrada podemos tener el movimiento de la tierra, entonces la señal de salida (en desplazamiento, velocidad o aceleración) se mostrará en un sismograma. Matemáticamente, la señal de salida será la convolución de la señal de entrada con la función de transferencia del sistema:

$$y(n) = x(n) * h(n) \quad \rightarrow \quad Y(s) = X(s)H(s) \quad \rightarrow \quad H(s) = \frac{Y(s)}{X(s)}$$

Para un sistema SLIT: $h_k(n) = h_0(n-k) \quad \rightarrow \quad h(n) = h_0(n)$ esto es, *h(n)* es la salida del sistema SLIT cuando la entrada es un impulso unitario *δ(n)*.

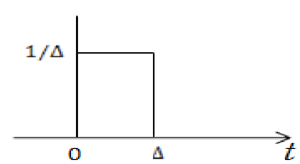
Luego:

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$ es la suma de convolución.

NOTA: la respuesta debido a la entrada *x(k)* aplicada en el tiempo *k* es entonces *x(k).h(n-k)*, es decir, es una versión desplazada y escalada (un eco) de la función de transferencia *h(n)*. La salida real será la superposición de todas estas respuestas.

Integral de convolución: se aplica para señales continuas en el tiempo *x(t)*.

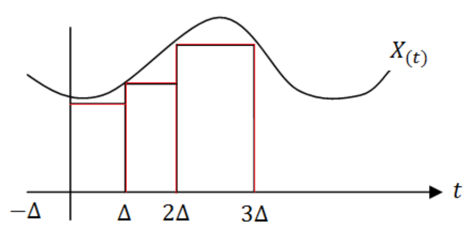
Sea: $$\delta_\Delta(t) = \begin{cases} 1/\Delta, & 0 \le t \le \Delta \\ 0, & otro\,valor \end{cases}$$



Luego $\Delta\delta_\Delta(t)$ siempre tiene amplitud unitaria y el área

de $\delta_\Delta(t)$ siempre es unitario.

$$\hat{x}(t) = \sum_{k=-\infty}^{\infty} x(k\Delta)\delta_\Delta(t-k\Delta)\Delta\,, \quad \text{si } \Delta \rightarrow 0$$



$$x(t) = \lim_{\Delta \to 0} \hat{x}(t) = \lim_{\Delta \to 0} \sum_{k=-\infty}^{\infty} x(k\Delta)\delta_{\Delta}(t - k\Delta)\Delta$$

$$x(t) = \int_{-\infty}^{\infty} x(t)\delta(t-\tau)d\tau$$

Esta ecuación representa la propiedad de selección del impulso unitario contínuo.

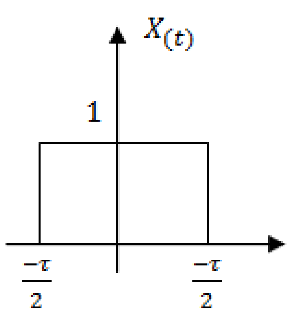
Si $x(t) = u(t)$, entonces:

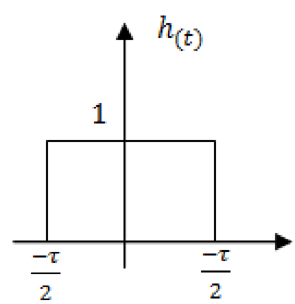
$$x(t) = \int_{-\infty}^{\infty} u(t)\,\delta(t-\tau)d\tau \quad \text{y} \quad u(t) = \int_{0}^{\infty} \delta(t-\tau)d\tau$$

La integral de convolución será: $y(t) = \int_{-\infty}^{\infty} x(t)h(t-\tau)d\tau$

En forma compacta: $y(t) = x(t) * h(t)$

Problema.- Hallar la convolución de *x(t)* con *h(t)*, es decir la convolución de la función compuerta unitaria consigo misma.

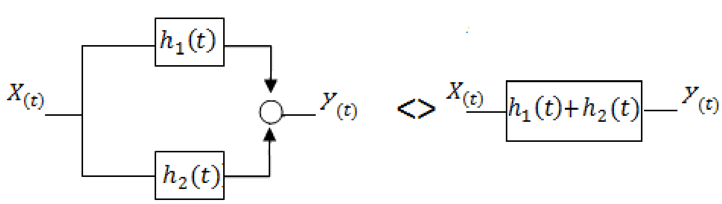




Propiedades de los SLIT:

1) Propiedad conmutativa: $x(t) * h(t) = h(t) * x(t)$

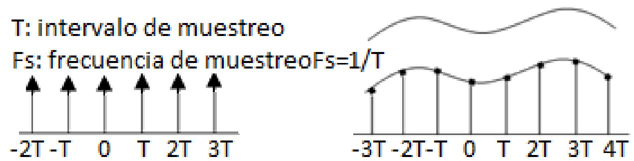
2) Propiedad distributiva: $x(t) * [h_1(t) + h_2(t)] = x(t) * h_1(t) + x(t) * h_2(t)$



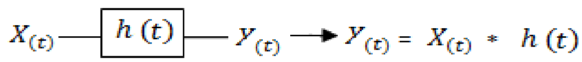
3) Propiedad asociativa: $x(t)*\left[h_1(t)*h_2(t)\right]=\left[x(t)*h_1(t)\right]*h_2(t)$

Propiedad: $x(t)*\delta(t)=x(t)$

Función peine: $\varphi(t)=\sum_{k=-\infty}^{\infty}\delta(t-kT)$



Respuesta del sistema al impulso unitario:



Si la entrada es un impulso unitario: $x(t)=\delta(t)\rightarrow y(t)=\delta(t)*h(t)\rightarrow y(t)=h(t)$

Si la entrada es un escalón unitario:

$$x(t)=u(t)\rightarrow y(t)=u(t)*h(t)=s(t)=\int_{-\infty}^{t}h(t)dt\rightarrow h(t)=\frac{ds(t)}{dt}$$

La función *s(t)* es la respuesta del sistema cuando la entrada es un escalón unitario *u(t)*.

En el tiempo discreto:

$$s(n)=\sum_{k=-\infty}^{n}h(k)\rightarrow h(n)=s(n)-s(n-1)$$

Descripción del SLIT mediante ecuaciones diferenciales o ecuaciones de diferencias:

Caso continuo: $\frac{dy(t)}{dt}+ay(t)=bx(t)$

Donde: *x(t)*: señal de entrada, *y(t)*: señal de salida. Se cumple que la solución es, en general:

y(t) = y(homogéneo) + y(particular)

Con la condición inicial: $y(0)=k$

Nota: También se puede resolver la ecuación diferencial utilizando la transformada de Laplace.

Caso discreto:

$$y(n)+ay(n-1)=bx(n)$$

donde: *x(n)*: señal de entrada o dato, *y(0)=k*: condición inicial, *y(n)*: señal de salida.

La solución consiste en obtener una ley de formación para *y(n)* mediante recursividad.

Por ejemplo: sea la ecuación $y(n)-\frac{1}{2}y(n-1)=x(n)$, donde: $x(n)=\delta(n)$ y $y(-1)=0$

Solución:

$$n=0 \rightarrow y(0)-\frac{1}{2}y(-1)=\delta(0) \rightarrow y(0)=1$$

$$n=1 \rightarrow y(1)-\frac{1}{2}y(0)=\delta(1) \rightarrow y(1)=\frac{1}{2}$$

$$n=2 \rightarrow y(2)-\frac{1}{2}y(1)=\delta(2) \rightarrow y(2)=(\frac{1}{2})^2$$

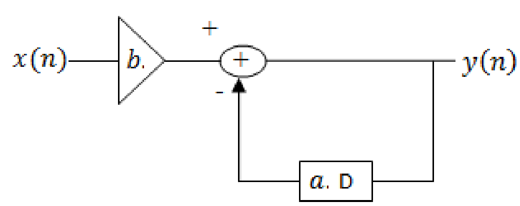
$$n=3 \rightarrow y(3)-\frac{1}{2}y(2)=\delta(3) \rightarrow y(3)=(\frac{1}{2})^3$$

Para *n*=*k*, tenemos: $y(k)=\left(\frac{1}{2}\right)^k$, generalizando: $y(n)=\left(\frac{1}{2}\right)^n u(n)$

Al agregar la función escalón unitario *u(n)* estamos indicando que la función *y(n)* es causal en el tiempo.

Nota: También se puede aplicar la transformada discreta de Fourier o la transformada Z.

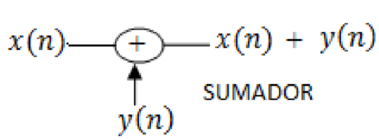
Diagrama de bloques

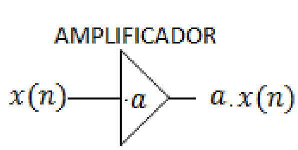


$$y(n)+ay(n-1)=bx(n)$$

$$y(n)=-ay(n-1)+bx(n)$$

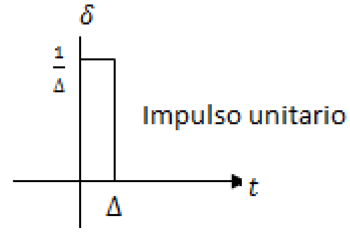
Bloques básicos:



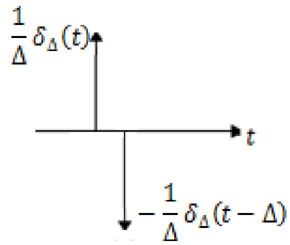


MÁS DE FUNCIONES SINGULARES:

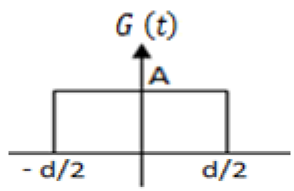
$$\frac{d\delta_\Delta(t)}{dt} = \frac{1}{\Delta}\delta_\Delta(t) - \frac{1}{\Delta}\delta_\Delta(t-\Delta) \rightarrow \frac{d\delta_\Delta(t)}{dt} = \lim_{\Delta\to 0}\frac{\delta_\Delta(t)-\delta_\Delta(t-\Delta)}{\Delta}$$

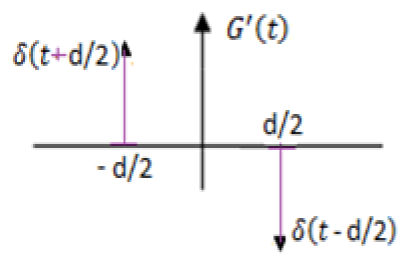


1) Dipolo o doblete unitario



2) Compuerta unitaria: $G(t) = u(t+d/2) - u(t-d/2)$

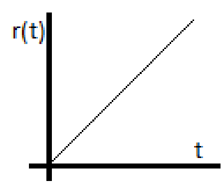




La derivada de la función compuerta unitaria: $G'(t) = \delta(t+d/2) - \delta(t-d/2)$

3) Función rampa unitaria:

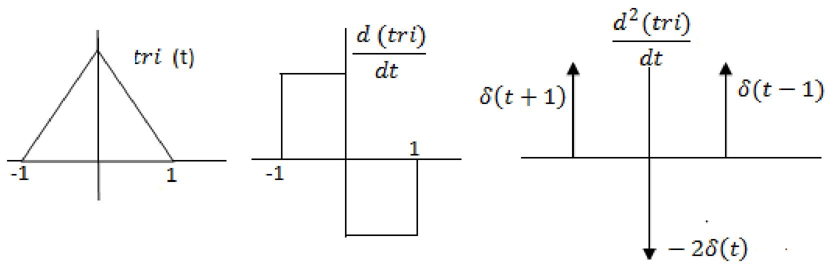
$$r(t) = \begin{cases} t & t \geq 0 \\ 0 & t < 0 \end{cases} \qquad r(t) = t\,u(t)$$



4) Función triangulo unitario:

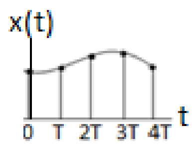
$$tri(t) = \begin{cases} t+1 & -1 \leq t < 0 \\ -t+1 & 0 \leq t < 1 \\ 0 & |t| > 1 \end{cases}$$

Derivando dos veces: $$\frac{d^2 tri(t)}{dt} = \delta(t+1) - 2\delta(t) + \delta(t-1)$$



MUESTREO

1) Representación de una señal mediante sus muestras:

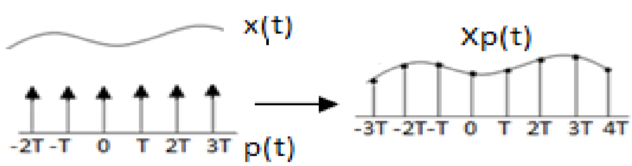


x(t): es una función continua, cada T seg. se toma una muestra.

El muestreo se realiza mediante la convolución de la función contínua *x(t)* con un tren de impulsos unitarios:

$x_p(t) = x(t) * p(t)$, donde la función: $p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$, es el tren de impulsos.

T: intervalo de muestreo,   Fs=1/T: frecuencia de muestreo.



NOTA: $X_p(t)$ es un tren de impulsos en el cual las amplitudes de las mismas son iguales a las muestras de *x(t)* en intervalos espaciados por *T*.

Teorema del Muestreo (Teorema de Shannon)

Sea *x(t)* una señal de banda limitada con $X(j\omega) = 0$, para $\omega > \omega_{\max}$. Entonces, *x(t)* se determina unívocamente mediante sus muestras $x(nT)$, $n = 0, \pm1, \pm2, \pm3\ldots$ si:

$$\omega_s > 2\omega_{\max}$$

$$F_s > 2f_{\max}$$

Donde: $\omega_s = 2\pi F_s = \dfrac{2\pi}{T}$, es la frecuencia angular de muestreo.

$\omega_{\max}$: máxima frecuencia angular presente en la señal.

Una señal en el dominio del tiempo contínuo se representa mediante la suma de Fourier:

$$x(t) = \sum_{n=0}^{\infty} A_n \, sen(\omega_n t + \alpha_n)$$

Por ejemplo: el oído humano puede captar señales de audio en el rango de frecuencias de 20 Hz < *f* < 20 kHz, entonces la frecuencia de muestreo será: Fs>2(20 kHz) => Fs > 40 kHz. Comercialmente se ha elegido el valor: Fs = 44.1 kHz para la grabación de señales de audio de alta fidelidad.

Nota: El formato Mp3 comprime la señal de audio (el archivo digital), para lo cual filtra las componentes de alta frecuencia y también disminuye la resolución.

Nota: dadas estas muestras, podemos recuperar la función *x(t)* generando un tren de impulsos periódicos en el cual los impulsos sucesivos tengan amplitudes que correspondan a valores de muestras sucesivas.

2) Reconstrucción de una señal a partir de sus muestras usando interpolación.

a) Interpolación lineal: es la más simple y la menos eficaz. Se utilizan rectas interpoladoras entre puntos consecutivos.
b) Interpolación polinomial: para *N* muestras se puede obtener un polinomio interpolador de grado *N-1*:

$$x(t) = \sum_{n=0}^{N-1} a_n t^n$$

Ojo: el algoritmo de la interpolación cúbica en Matlab es muy eficaz.

c) Interpolación con splines.
d) Sea el filtro pasa bajo de la señal:

Esta última ecuación representa una fórmula de interpolación.

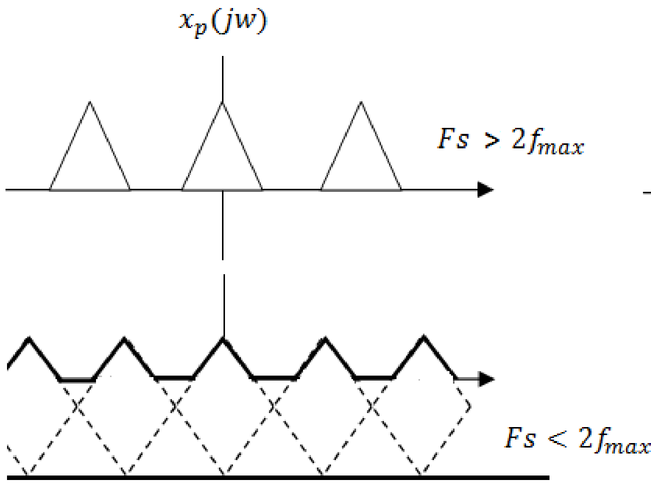
Nota.- La interpolación es útil para completar datos en una serie de tiempo incompleta.

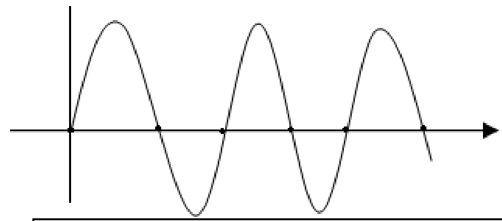
Añadir un ejemplo de interpolación de datos en Matlab

3) Efecto del submuestreo: traslape o aliasing:

Si la frecuencia de muestreo $Fs > 2f_{max}$, entonces el espectro de la señal muestreada consiste en réplicas escaladas del espectro de *x(t)* y esto forma la base del teorema del muestreo.

Pero, cuando la frecuencia de muestreo $Fs < 2f_{max}$, el espectro de la señal *x(t)* ya no está reproducida en $X_p(j\omega)$ y por consiguiente la señal no es recuperable.





El teorema del muestreo requiere explícitamente que la frecuencia de muestreo sea mayor que el doble de la frecuencia más alta de la señal, en vez de ser mayor o igual al doble de la frecuencia más alta.

TRANSFORMADA DE LAPLACE

Para un sistema lineal invariante en el tiempo (SLIT) caracterizado por *h(t)* causal, donde: *h(t)=h(t)u(t)*, se define la transformada de Laplace como:

$$H(s) = \int_0^{\infty} h(t)e^{-st}\,dt \qquad \text{donde:} \qquad s = \sigma + j\omega$$

H(s) describe las propiedades del sistema en el dominio de la frecuencia, es la función de transferencia del sistema.

Ejemplo: sea $x(t) = e^{-at}u(t)$. Hallar su transformada de Laplace.

$$X(s) = \int_0^{\infty} e^{-at}e^{-st}\,dt = \int_0^{\infty} e^{-at-st}\,dt = \frac{1}{s+a}$$

Nota.- La transformada de Laplace se utiliza para estudiar una señal en régimen transitorio.

Tabla de transformadas:

$x(t)$	$X(s)$
$u(t)$	$\frac{1}{s}$
e^{-at}	$\frac{1}{s+a}$
$\delta(t)$	1
$\cos(\omega_0 t)$	$\frac{s}{s^2+\omega_0^2}$
$sen(\omega_0 t)$	$\frac{\omega}{s^2+\omega_0^2}$
$\delta(t-\tau)$	$e^{-s\tau}$

Propiedades:

1) Linealidad:

$$L\{x_1(t)+x_2(t)\} \to X_1(s)+X_2(s)$$
$$L\{ax(t)\} \to aX(s)$$

2) Desplazamiento en el dominio de la frecuencia “s”:

$$e^{s_0\tau}x(t) \to X(s-s_0)$$

3) Escalamiento en el tiempo: $x(at) \to \frac{1}{|a|} X(\frac{s}{a})$

4) Propiedad de convolución: $L\{x_1(t) * x_2(t)\} = X_1(s) X_2(s)$

5) Derivada en el dominio del tiempo:

$$x(t) \to X(s)$$

$$\frac{dx(t)}{dt} \to sX(s) - X(0)$$

$$\frac{d^2 x(t)}{dt^2} \to s^2 X(s) - sX(0) - X'(0)$$

Ecuaciones diferenciales: Al aplicar la transformada de Laplace, la ecuación diferencial se convierte en una ecuación algebraica en "s":

$$L\left\{\frac{d^2 f(t)}{dt^2} + b\frac{df(t)}{dt} + cf(t) = x(t)\right\}$$

$$s^2 F(s) - sF(0) - F'(0) + bsF(s) - bF(0) + cF(s) = X(s)$$

Si las condiciones iniciales son: $F(0) = F'(0) = 0$, entonces:

$$F(s)\left[s^2 + bs + c\right] = X(s) \quad => \quad F(s) = \frac{X(s)}{s^2 + bs + c}$$

Nota: *f(t)* se obtiene mediante la transformada inversa por el método de fracciones parciales.

Nota: La transformada de Laplace transforma ecuaciones diferenciales en ecuaciones algebraicas.

Aplicación en Sistemas Lineales e Invariantes en el Tiempo

En el dominio del tiempo: $y(t) = x(t) * h(t)$

En el dominio de la frecuencia: $Y(s) = X(s).H(s) \quad => \quad H(s) = \frac{Y(s)}{X(s)}$

donde: $H(s)$ es la función de transferencia del sistema en el dominio de la frecuencia.

Obtención de la función de transferencia de un sistema físico

Para obtener la función de transferencia de un sistema se realiza los siguientes pasos:

1. Aplicar la ley gobernante del sistema.
2. Obtener una ecuación diferencial en el dominio del tiempo.
3. Aplicar la transformada de Laplace con condiciones iniciales iguales a cero.
4. Finalmente la función de transferencia del sistema será: $H(s) = \dfrac{Y(s)}{X(s)}$

Si resolvemos la ecuación $Y(s) = 0$, se obtienen los "ceros".

Si resolvemos la ecuación $X(s) = 0$, se obtienen los "polos".

De manera que la función de transferencia del sistema se puede expresar en función de los "polos" y ceros" y de la ganancia (factor multiplicador) del sistema:

$$H(\omega) = \frac{Y(s)}{X(s)} = k\frac{N(s)}{D(s)} = k\frac{(s-c_1)(s-c_2)...(s-c_m)}{(s-p_1)(s-p_2)...(s-p_n)}$$

Por ejemplo para un sismógrafo tenemos que:

La ganancia: k=1000; ceros: c_1=3; polos: p_1=1, p_2=4; hallar *H(s)* y *h(t)*.

$$H(s) = k\frac{N(s)}{D(s)} = k\frac{(s-c_1)}{(s-p_1)(s-p_1)} = 1000\frac{(s-3)}{(s-1)(s-4)} = 1000(\frac{A}{(s-1)} + \frac{B}{(s-4)})$$

$A(s-4) + B(s-1) = s-3$, luego: $A = \dfrac{2}{3}$ y $B = \dfrac{1}{3}$

Por lo tanto: $H(s) = 1000\,(\dfrac{2}{3(s-1)} + \dfrac{1}{3(s-4)})$

En el dominio del tiempo: $h(t) = 1000\,(\dfrac{2}{3}e^{-t} + \dfrac{1}{3}e^{-4t})$

TRANSFORMADA CONTÍNUA DE FOURIER

Sea *f(t)* una función en el dominio del tiempo continuo, la transformada de Fourier de *f(t)* es:

$$F(j\omega) = \Im\left\{f(t)\right\} = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt$$

Los requisitos para que *f(t)* tenga una Transformada de Fourier son:

a) *f(t)* debe ser continua por tramos.
b) El número de discontinuidades debe ser finito.
c) Debe converger a cero si: $t \to \infty \;\Rightarrow\; f(t) \to 0$

d) El área bajo la curva debe ser finito: $\int_{-\infty}^{\infty} f(t)dt = \;finito$

e) La función *f(t)* debe ser cuadrado integrable: $\int_{-\infty}^{\infty} \left[f(t)\right]^2 dt = \;finito$

Luego existe la transformada continua de Fourier: $F(j\omega) = \Im\left\{f(t)\right\}$

Nota: $f(t) = \sum_{n=0}^{\infty} a_n sen(\omega_n t) + b_n \cos(\omega_n t)$

Propiedades:

1) Linealidad (principio de superposición):

$$y(t) = af_1(t) + bf_2(t) \;=>\; Y(\omega) = aF_1(\omega) + bF_2(\omega)$$

2) Desplazamiento en el tiempo:

$$\Im\left\{f(t)\right\} = F(\omega) \;=>\; \Im\left\{f(t-t_0)\right\} = e^{-j\omega t_0}F(\omega)$$

3) Conjugada compleja: $F^*(\omega) = F(-\omega)$

4) Derivación

$$\Im\left\{f'(t)\right\} = j\omega F(\omega)$$

$$\Im\left\{f''(t)\right\} = (j\omega)^2 F(\omega)$$

$$\Im\left\{f^n(t)\right\} = (j\omega)^n F(\omega)$$

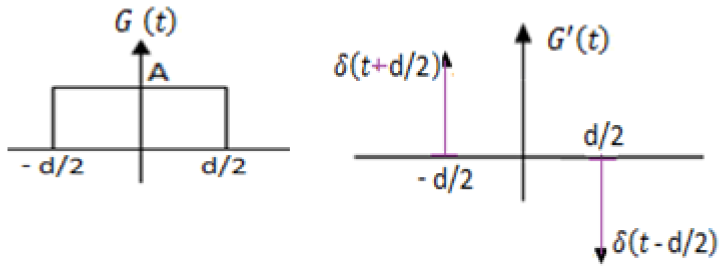
Ejemplo: Hallar la transformada de Fourier de la función compuerta unitaria *G(t)* por dos métodos: a) Definición y b) Propiedad

Solución:

a) Usando la definición tenemos:

$$F(j\omega)=\int_{-1/2}^{1/2}1\cdot e^{-j\omega t}dt=\left[-\frac{1}{j\omega}e^{-j\omega t}\right]_{-1/2}^{1/2}=\frac{2}{\omega}\left[\frac{e^{j\omega/2}-e^{-j\omega/2}}{2j}\right]=\frac{sen(\omega/2)}{\omega/2}=\sin c(\frac{\omega}{2})$$

b) Por propiedades: primero se debe derivar *G(t)* hasta obtener impulsos unitarios:



$$G(t)=u(t+1/2)-u(t-1/2)$$

$$G'(t)=\delta(t+1/2)-\delta(t-1/2)$$

Aplicando la Transformada a la derivada: $\Im\{G'(t)\}=j\omega G(\omega)$

$$\int_{-\infty}^{\infty}[\delta(t+1/2)-\delta(t-1/2)]e^{-j\omega t}dt=j\omega G(\omega)$$

$$e^{j\omega/2}-e^{-j\omega/2}=j\omega G(\omega)$$

$$\omega G(\omega)=2\left[\frac{e^{j\omega/2}-e^{-j\omega/2}}{2j}\right]=2sen(\omega/2)$$

$$G(\omega)=\frac{sen(\omega/2)}{\omega/2}=\sin c(\frac{\omega}{2})$$

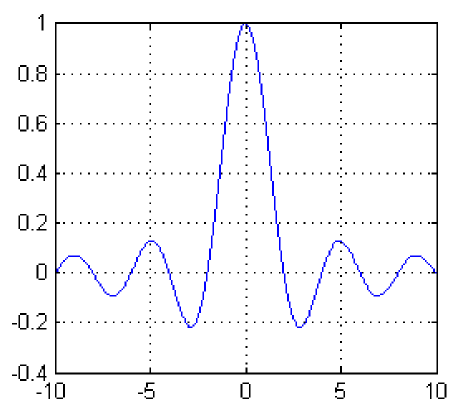


Fig. xx Función *sinc(ω/2)*. Amplitud vs frecuencia *ω*

Ejemplo: transformada de Fourier del impulso unitario.

$$F(j\omega) = \int_{-\infty}^{\infty} \delta(t) e^{-j\omega t} dt = e^{-j\omega 0} = 1$$

Ejemplo: transformada de Fourier de una función exponencial decreciente causal: $e^{-at}u(t)$

$$F(j\omega) = \int_{-\infty}^{\infty} e^{-at} u(t) e^{-j\omega t} dt ,$$ donde *f(t)* es una función causal.

$$F(j\omega) = \int_{0}^{\infty} e^{-at} e^{-j\omega t} dt = -\frac{1}{a + j\omega} e^{-(a+j\omega)t}\Big|_0^{\infty} = \frac{1}{a + j\omega}$$

Transformada de una función periódica.

Sea f(t) una función periódica de periodo T, entonces la función puede ser escrito como:

$$f(t) = \sum_{-\infty}^{\infty} c_n e^{jnw_0 t} \quad (2)$$

Tomando transformada en ambos miembros de la ecuación se obtiene:

$$\mathcal{F}[f(t)] = F(w) = \mathcal{F}\left[\sum_{-\infty}^{\infty} c_n e^{jnw_0 t}\right] = \sum_{-\infty}^{\infty} c_n \mathcal{F}[e^{jnw_0 t}] \quad (3)$$

La transformada de la ecuación 3 se puede resolver de la manera siguiente:

$$\mathcal{F}[f(t)] = F(w) = \mathcal{F}\left[e^{jnw_0 t}\right] = \mathcal{F}\left[p(t).e^{jnw_0 t}\right] \quad (4)$$

donde se ha considerado que p(t)=1 (función constante) de la cual se conoce su transformada como:

$$\mathcal{F}[p(t)] = \mathcal{F}[1] = 2\pi\delta(w) \quad (5)$$

De acuerdo a la propiedad de desplazamiento en la frecuencia aplicada a la ecuación 5, se tiene:

$$\mathcal{F}\left[p(t).e^{jnw_0 t}\right] = P(w - nw_0) = 2\pi\delta(w - nw_0) \quad (6)$$

Reemplazando la ecuación 6 en la 3 se tiene finalmente:

$$F(w) = 2\pi \sum_{n=-\infty}^{\infty} c_n \delta(w - nw_0) \quad (7)$$

con $\omega_0 = \dfrac{2\pi}{T}$

Según el resultado de la ecuación 7 se establece que la transformada de Fourier de una función periódica, consta de una sucesión de impulsos equidistantes localizados en las frecuencias armónicas de la función.

Consideremos ahora como señal periódica, un tren de impulsos unitarios equidistantes, la transformada de Fourier del tren de impulsos unitarios δT (t) donde $\delta(t)=\sum_{n=-\infty}^{\infty}\delta(t-nT)$ es una sucesión de impulsos equidistantes, se puede determinar considerando que la serie de Fourier de δT (t) está dada por:

$$\delta_T(t)=\frac{1}{T}\sum_{n=-\infty}^{\infty}e^{jnw_0t}$$

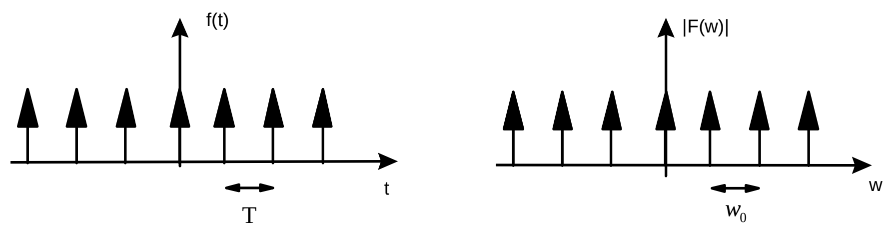
Entonces:

$$\mathrm{F}[\delta_T(t)]=\frac{1}{T}\sum_{n=-\infty}^{\infty}\mathrm{F}[e^{jnw_0t}] \qquad (8)$$

$$\mathrm{F}[\delta_T(t)]=\frac{2\pi}{T}\sum_{n=-\infty}^{\infty}\delta(w-nw_0)=w_0\sum_{n=-\infty}^{\infty}\delta(w-nw_0)$$

$$\mathrm{F}[\sum_{n=-\infty}^{\infty}\delta(t-nT)]=w_0\sum_{n=-\infty}^{\infty}\delta(w-nw_0) \qquad (9)$$

De acuerdo al resultado de la ecuación 9 se establece que la transformada de Fourier de un tren de impulsos es también un tren de impulsos equidistantes en ω_0. La figura #6 muestra la función f(t) de impulsos equidistantes y su correspondiente transformada.

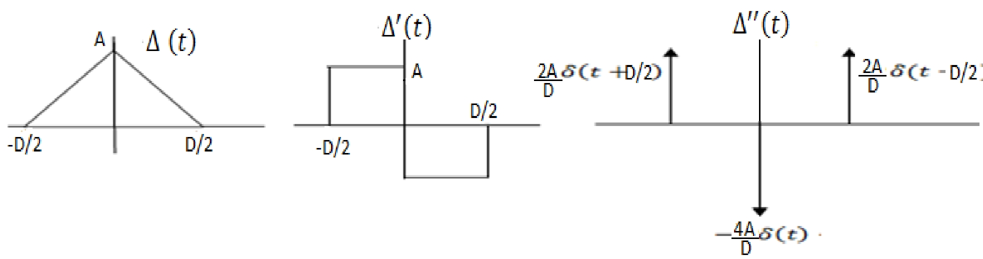


Representación gráfica de la función impulso periódica de periodo T y su transformada.

Ejemplo: para una función periódica (tren de impulsos unitarios).

$$F(\omega)=\frac{2\pi}{T}\sum_{k=-\infty}^{\infty}\delta(\omega-\frac{2\pi k}{T})$$ es otro tren de impulsos

Ejemplo: triangulo de ancho "D" y amplitud "A".



$$\Delta''(t) = f''(t) = \frac{2A}{D}\delta(t+\frac{D}{2}) - \frac{4A}{D}\delta(t) + \frac{2A}{D}\delta(t-\frac{D}{2})$$

Aplicando la transformada de Fourier y sus propiedades a *f''(t)*:

$$(j\omega)^2 F(\omega) = \frac{2A}{D}\left[e^{j\omega D/2} - 2e^{j\omega 0} + e^{-j\omega D/2}\right] = \frac{2A}{D}\left[2\frac{e^{j\omega D/2} + e^{-j\omega D/2}}{2} - 2\right]$$

$$F(\omega) = \frac{4A}{D\omega^2}\left[\cos(\frac{\omega D}{2}) - 1\right]$$

Caso particular: triangulo unitario A=1 y D=1: $\Im\{\Delta_1(t)\} = \frac{4}{\omega^2}\left[\cos(\frac{\omega}{2}) - 1\right]$

Más propiedades:

1) Integrales: $\Im\left\{\int_{-\infty}^{t} f(\tau)d\tau\right\} = \frac{1}{j\omega}F(\omega) + \pi F(0)\delta(\omega)$

2) Escalamiento en el tiempo:

$\Im\{f(at)\} = \frac{1}{|a|}F(\frac{\omega}{a})$ si a = -1 entonces: $\Im\{f(-t)\} = F(-\omega)$

3) Convolución: $\Im\{x_1(t) * x_2(t)\} = X_1(\omega)X_2(\omega)$

4) Transformada inversa: $f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{j\omega t}d\omega$

5) Relación de Parseval: $\int_{-\infty}^{\infty}[f(t)]^2 dt = \frac{1}{2\pi}\int_{-\infty}^{\infty}[F(\omega)]^2 d\omega$

Ecuaciones diferenciales con coeficientes constantes

$$\sum_{n=0}^{N} a_n \frac{d^n y(t)}{dt^n} = \sum_{n=0}^{N} b_n \frac{d^n x(t)}{dt^n}$$

Si se aplica la transformada de Fourier a la ecuación diferencial en el dominio del tiempo, se transforma en una ecuación algebraica en el dominio de la frecuencia.

$$Y(\omega) = X(\omega)H(\omega)$$

Donde $H(\omega)=\dfrac{Y(\omega)}{X(\omega)}$ es la función de transferencia del sistema

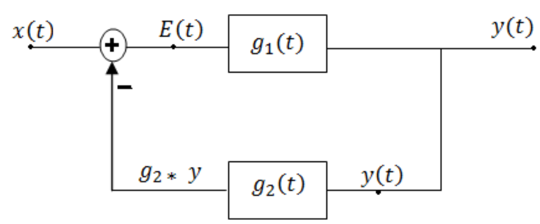
Ejemplo1: $\dfrac{dy(t)}{dt}+ay(t)=x(t)$, condición inicial: y(0) = 0

$$\Im\left\{\frac{dy(t)}{dt}+ay(t)\right\}=\Im\{x(t)\}$$

$$j\omega Y(\omega)+aY(\omega)=X(\omega)\Rightarrow Y(\omega)[j\omega+a]=X(\omega)\Rightarrow H(\omega)=\frac{Y(\omega)}{X(\omega)}=\frac{1}{j\omega+a}$$

Si aplicamos la transformada inversa de Fourier obtenemos: $h(t)=u(t)e^{-at}$

Ejemplo 2: Diagrama de bloques:



a) Deducir la ecuación de diferencias del sistema.
b) Hallar la función de transferencia *H(ω)*.

$$E(t)=x(t)-g_2(t)*y(t)$$
$$y(t)=E(t)*g_1(t)\rightarrow$$
$$y(t)=[x(t)-g_2(t)*y(t)]*g_1(t)$$

En el dominio de la frecuencia seria:

$$Y(\omega)=[X(\omega)-G_2(\omega)Y(\omega)]G_1(\omega)$$
$$Y(\omega)=X(\omega)G_1(\omega)-G_1(\omega)G_2(\omega)Y(\omega)$$
$$Y(\omega)[1+G_1(\omega)G_2(\omega)]=X(\omega)G_1(\omega)\rightarrow$$
$$H(\omega)=\frac{Y(\omega)}{X(\omega)}=\frac{G_1(\omega)}{1+G_1(\omega)G_2(\omega)}$$

TRANSFORMADA DISCRETA DE FOURIER

Sea *f(n)* una función de variable discreta, donde: $n = \ldots -2, -1, 0, 1, 2, \ldots$

La transformada de Fourier de *f(n)* es: $F(j\omega) = \sum_{n=-\infty}^{\infty} f(n) e^{-j\omega n}$

Si *T*: es el intervalo de muestreo; *Fs= 1/T* : frecuencia de muestreo, entonces:

$$F(j\omega) = \sum_{k=-\infty}^{\infty} f(kT) e^{-j\omega kT} .T$$

Ejemplo: sea $f(n) = \delta(n+1) + 2\delta(n) + \delta(n-1)$, hallar la transformada discreta de Fourier.

$$F(j\omega) = \sum_{n=-1}^{1} f(n) e^{-j\omega n} = f(-1)e^{j\omega} + f(0)e^{j\omega 0} + f(1)e^{-j\omega} = 2(\frac{e^{j\omega} + e^{-j\omega}}{2}) + 2$$

Por lo tanto: $F(j\omega) = 2(1 + \cos\omega)$

Función "trf" en Matlab:

```
function trf(y, Fs)
% Calcula la Transformada Discreta de Fourier de una funcion
% Sintaxis:   trf(y,Fs)
% y = funcion en el dominio del tiempo discreto
% Fs = frecuencia de muestreo: Fs = 1/T

subplot (3,1,1), plot(y), grid, zoom xon
title ('Señal en el dominio del tiempo')
% xlabel ('Tiempo (s)')
ylabel ('Altura (m)')

N = length(y);   %512    % N = numero de puntos
Y = fft(y,N);
Pyy = Y.*conj(Y) / N; % espectro de frecuencias
f = Fs*(0:N/2-1)/ N;
subplot(3,1,2), plot(f,Pyy(1:length(f))), grid, zoom xon
title('Espectro de frecuencias de la señal')
xlabel('Frecuencia (Hz)')
subplot(3,1,3), loglog(f,Pyy(1:length(f))), grid, zoom xon
% espectro de frecuencia logaritmico
```

Ejemplo de corrida de la función:
\>> T = 0.01; Fs = 1/T;
\>> t = -10:0.01:10;
\>> y = sin(t) + 0.25*sin(20*t);
\>> trf (y,Fs)

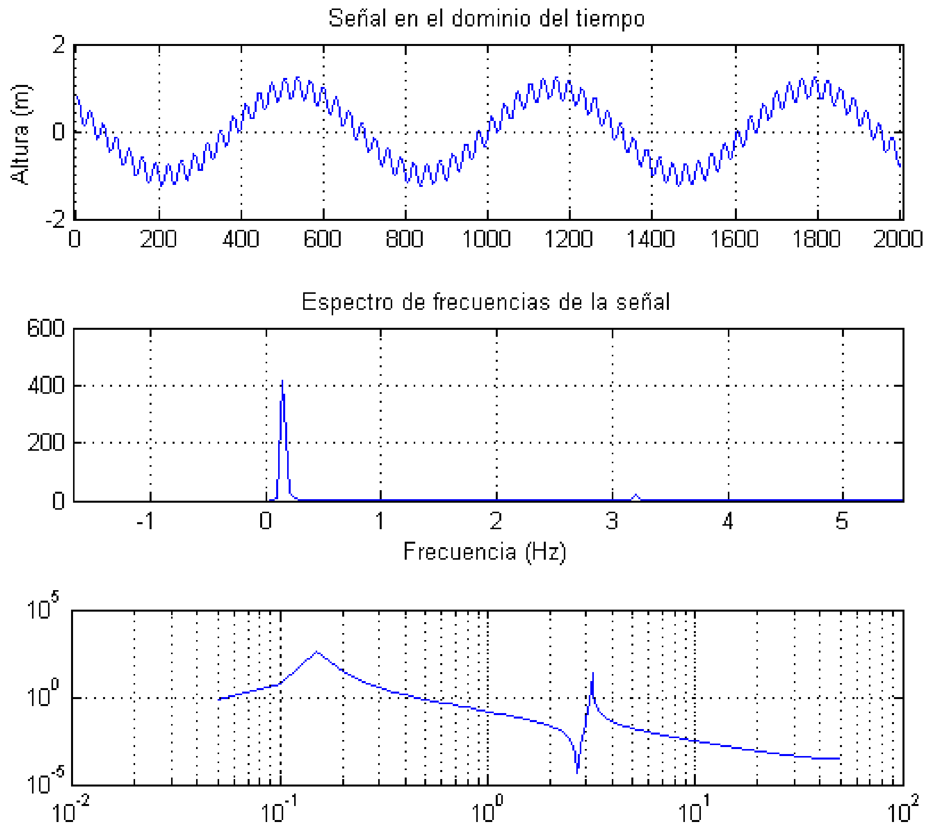
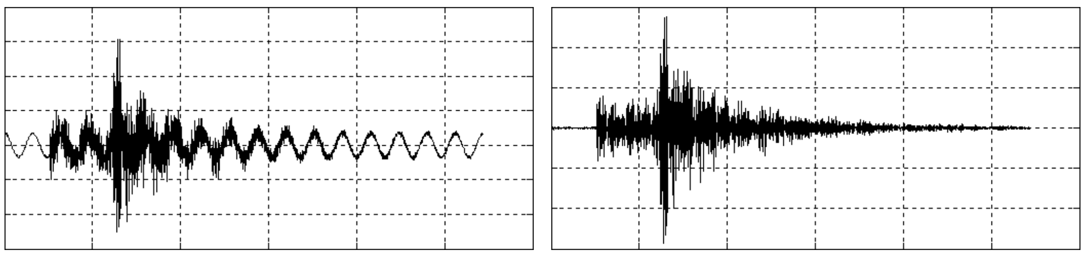


Fig. xx Corrida de la función trf.m

FILTROS ANALOGICOS Y DIGITALES

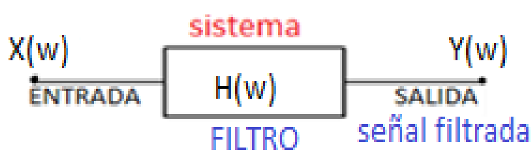
Un filtro es un sistema lineal e invariante en el tiempo (SLIT) que aplicado sobre una señal modifica, elimina o atenúa las componentes frecuenciales seleccionadas. Entonces, el proceso de filtrado consiste en eliminar, atenuar o amplificar ciertas componentes frecuenciales de una señal. Por ejemplo, la Figura x(a) muestra una señal sísmica contaminada con ruido de baja frecuencia, la cual debe ser tratada con un filtro pasa-alta para eliminar dicha componente frecuencial. El resultado es la señal de la Figura x(b).



Entonces, el filtro se puede considerar como un sistema que tiene una determinada función de transferencia que matemáticamente será expresada como:

$$H(\omega) = \frac{Y(\omega)}{X(\omega)}$$

donde $H(\omega)$ es la función de transferencia del filtro

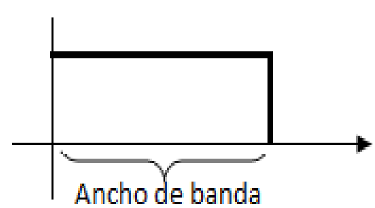


En el dominio del tiempo: la señal original o de entrada se representa en función de senos y cosenos de diferente amplitud y de diferente frecuencia:

$$x(t) = \sum_{n=0}^{\infty} a_n \cos(n\omega t) + b_n sen(n\omega t)$$

Entonces el proceso de filtrado consiste en modificar los parámetros a_n y b_n, es decir, al filtrar la señal $x(t)$, se modifican las amplitudes a_n y b_n.

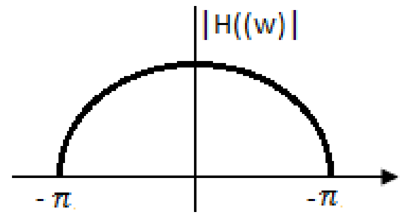
En el dominio de la frecuencia: el filtro tiene un ancho de banda determinado por la frecuencia de corte. Al filtrar la señal se estaría reduciendo el ancho de banda de su espectro de frecuencias.



Filtro mediano

Se utiliza el filtro mediano para eliminar el ruido impulsivo o variaciones rápidas o de alta frecuencia (glitch = pico).

$$y(n)=\frac{1}{2}\left[x(n)+x(n-1)\right]$$



Aplicamos la transformada discreta de Fourier:

$$Y(\omega)=\frac{1}{2}\left[X(\omega)+e^{-j\omega}X(\omega)\right]$$

$$H(\omega)=\frac{Y(\omega)}{X(\omega)}=\frac{1}{2}\left[1+e^{-j\omega}\right]=e^{-j\omega}\cos(\omega/2)$$

En el dominio del tiempo: $h(n)=\frac{1}{2}\left[\delta(n)+\delta(n-1)\right]$

Nota: El filtro mediano se comportaría como un filtro pasa bajo.

Implementación en Matlab: y = medfilt1(x,n). Donde: *x* representa la serie de tiempo a filtrar, *y* es la serie de tiempo filtrada y *n* es el orden que por defecto es igual a 3.

Ejemplo: Sea $y(n)=\frac{1}{2}\left[x(n)-x(n-1)\right]$, hallar la función de transferencia del filtro.

Aplicando la transformada discreta de Fourier: $Y(\omega)=\frac{1}{2}\left[X(\omega)-e^{-j\omega}X(\omega)\right]$

La función de transferencia del filtro: $H(\omega)=\frac{Y(\omega)}{X(\omega)}=\frac{1}{2}\left[1-e^{-j\omega}\right]$

En el dominio del tiempo: $h(n)=\frac{1}{2}\left[\delta(n)-\delta(n-1)\right]$

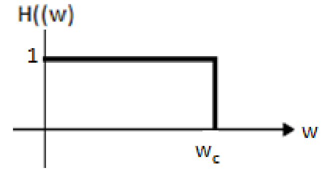
Nota: en este caso este filtro seria un filtro pasa alto.

TIPOS DE FILTROS (Filtros selectivos en frecuencia)

Según su aplicación, un filtro puede ser clasificado como pasa bajo, pasa alto, pasa banda o elimina banda.

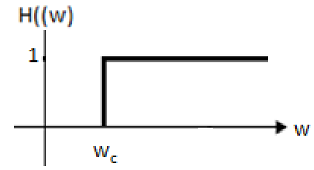
1) Filtro pasa bajo: deja pasar las componentes de bajas frecuencias y elimina las altas frecuencias, con respecto a una frecuencia de corte ω_c.

$$H(\omega)=\begin{cases}1 & \omega<\omega_c\\ 0 & \omega>\omega_c\end{cases}$$ donde ω_c es la frecuencia de corte



2) Filtro pasa alto: deja pasar las componentes de altas frecuencias y elimina las bajas frecuencias.

$$H(\omega)=\begin{cases}0 & \omega<\omega_c\\ 1 & \omega>\omega_c\end{cases}$$ donde ω_c es la frecuencia de corte



Nota: Cuando se trabaje con Matlab, será necesario introducir parámetros, tales como la frecuencia de corte del filtro.

Para una señal real (por ejemplo, una señal mareográfica), el filtro que se utilice dependerá del fenómeno físico que se desee estudiar. Como se sabe, en general una señal mareográfica está compuesta por la superposición de varias componentes frecuenciales:

Señal mareográfica = marea + olas + maremoto

Donde cada término representa un fenómeno físico en particular. En general, la marea es un fenómeno producido por las fuerzas gravitacionales de la luna y del sol, además tiene un periodo largo, en el orden de las horas, mientras que el oleaje (olas) son oscilaciones de periodo corto en el orden de los segundos (de 12 a 18 segundos). Un maremoto es un evento excepcional producido por un terremoto, tiene periodo largo, en el orden de los minutos; entonces, si se desea estudiar la marea, solo sería conveniente tener señales con frecuencias bajas, por lo tanto se aplicaría un filtro pasa bajo, pero es importante tener la frecuencia de corte. ¿Y cómo se calcula la frecuencia de corte? Para esto se aplica la transformada discreta de Fourier, es decir se transforma la serie de tiempo al dominio de la frecuencia y es en este dominio, en el que se va a identificar los picos principales que corresponden a la marea, oleaje o maremoto, luego se puede identificar la frecuencia de corte. El gráfico que se muestra a continuación, representa una señal mareográfica: las pequeñas ondas de color rojo representaría al oleaje y la onda de color negra representaría a la marea.

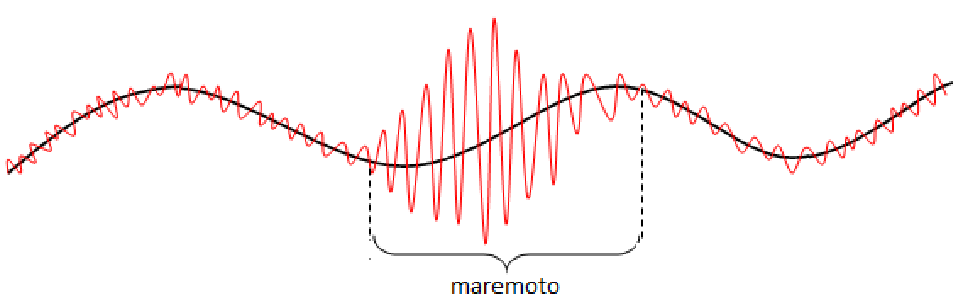


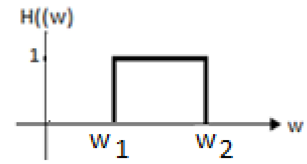
Fig. xx Ejemplo de una señal mareográfica.

Que se desea investigar	Filtro a aplicarse
Olas	Pasa alto
Marea	Pasa bajo
Maremoto	Pasa banda

3) Filtro pasa banda: solo deja pasar las componentes dentro de una banda de frecuencia.

$$H(\omega) = \begin{cases} 1 & \omega_1 < \omega < \omega_2 \\ 0 & otro \end{cases}$$

donde ω_1 y ω_2 son las frecuencias de corte



Ejemplos de filtros descritos por ecuaciones diferenciales:

a) En la primera parte de este libro (página 22) se puso como ejemplo a un sistema eléctrico que desde ya se puede decir que éste, es un filtro descrito por una ecuación diferencial que proviene de una ley que gobierna a ese sistema (ley de Kirchhoff),

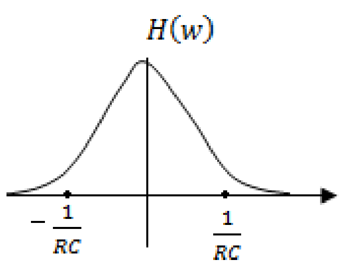
$$\frac{dv_c(t)}{dt} + \frac{1}{RC} v_c(t) = \frac{1}{RC} v_s(t)$$

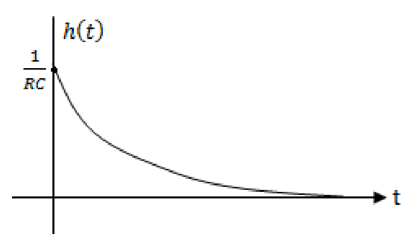
Entonces aplicando la transformada de Fourier a dicha ecuación se obtiene:

$$j\omega V_c(\omega) + \frac{1}{RC} V_c(\omega) = \frac{1}{RC} V_s(\omega)$$

$$V_c(\omega)\left[j\omega + \frac{1}{RC} \right] = \frac{1}{RC} V_s(\omega)$$

Donde la función de transferencia en el dominio de la frecuencia seria:



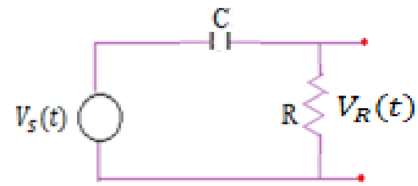


$$H(\omega) = \frac{V_c(\omega)}{V_s(\omega)} = \frac{\frac{1}{RC}}{\frac{1}{RC} + j\omega}$$

$$h(t) = \frac{1}{RC} e^{-t/RC} u(t)$$

La última expresión viene a ser la función de transferencia del filtro en el dominio del tiempo.

b) Ahora analicemos el voltaje de salida en la resistencia $v_R(t)$. Filtro pasa alta.



$$v_R(t) + v_C(t) = v_S(t)$$

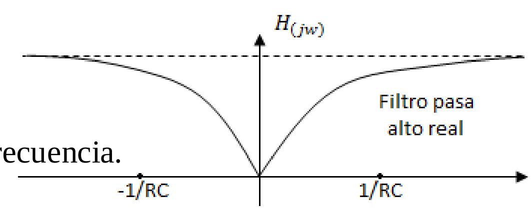
$$RC\frac{dv_R(t)}{dt} + v_R(t) = RC\frac{dv_S(t)}{dt}$$

Aplicando la transformada de Fourier:

$$RC(j\omega)V_R(j\omega) + V_R(j\omega) = RC(j\omega)V_S(j\omega)$$

$$V_R(j\omega)[1 + RC(j\omega)] + = RC(j\omega)V_S(j\omega)$$

$$H(j\omega) = \frac{V_R(\omega)}{V_S(\omega)} = \frac{RC}{1 + RCj\omega},$$



H es la función de transferencia en el dominio de la frecuencia.

Si se aplica la trasformada inversa de Fourier se tiene:

$h(t) = e^{-t/RC}u(t)$,   función de transferencia en el dominio del tiempo.

FILTROS DISCRETOS (Ecuaciones de diferencias)

a) Filtros recursivos: con respuesta al impulso infinito (I I R).

b) Filtro no recursivo: con respuesta al impulso finito (F I R).

Filtro recursivo de primer orden:

$$y(n) - ay(n-1) = x(n)$$

Aplicando la transformada discreta de Fourier: $Y(\omega) - ae^{-j\omega}Y(\omega) = X(\omega)$

$$Y(\omega)[1 - ae^{-j\omega}] = X(\omega)$$

La función de transferencia del filtro es: $H(\omega) = \dfrac{Y(\omega)}{X(\omega)} = \dfrac{1}{1 - ae^{-j\omega}}$

En el dominio del tiempo discreto: $h(n) = a^n u(n)$ -1<*a*<*1*

Si la entrada es un escalón unitario *u(n)* entonces: *y(n)=h(n)*x(n)*

$$y(n) = s(n) = \left(\frac{1 - a^{n+1}}{1 - a}\right) u(n) \qquad |a| < 1$$

Filtro no recursivo discreto: la forma general es $y(n) = \sum_{k=0}^{N} b_k x(n-k)$

En forma particular y(n) puede ser: $y(n) = \dfrac{1}{3}[x(n+1) + x(n) + x(n-1)]$

Aplicando la transformada discreta de Fourier:

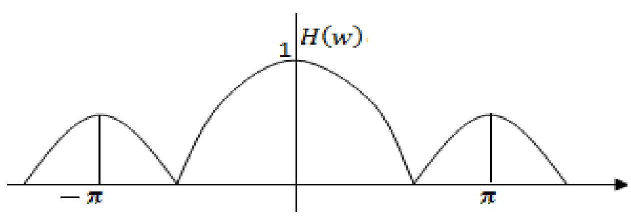
$$Y(\omega) = \frac{1}{3}\left[e^{j\omega} X(\omega) + X(\omega) + e^{-j\omega} X(\omega)\right]$$

Factorizando *X(ω)* y pasando al otro miembro de la igualdad:

$$H(\omega) = \frac{Y(\omega)}{X(\omega)} = \frac{1}{3}\left[e^{j\omega} + 1 + e^{-j\omega}\right] = \frac{1}{3}[2\cos\omega + 1]$$

Donde *H(ω)* es la función de transferencia del filtro en el dominio de la frecuencia.

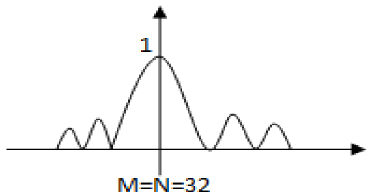
La transformada inversa seria: $h(n) = \dfrac{1}{3}[\delta(n+1) + \delta(n) + \delta(n-1)]$

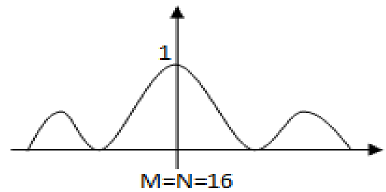


Otro filtro: no recursivo discreto.

$$y(n) = \frac{1}{N + M + 1} \sum_{k=-M}^{N} x(n-k)$$

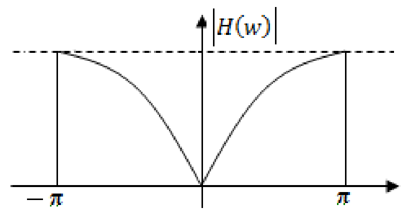
La ventaja es que se puede manejar el ancho de banda del filtro (frecuencia de corte).





Filtro recursivo pasa alta:

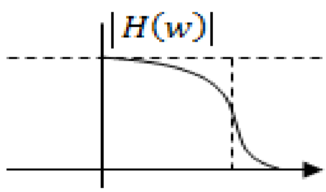
$$y(n) = \frac{1}{2}\left[x(n) - x(n-1)\right]$$



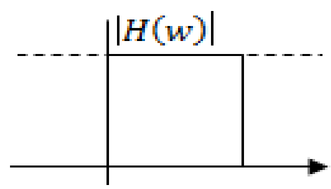
Aplicamos la transformada discreta de Fourier:

$$Y(\omega) = \frac{1}{2}\left[X(\omega) - e^{-j\omega}X(\omega)\right] \qquad => \qquad H(\omega) = \frac{Y(\omega)}{X(\omega)} = \frac{1}{2}\left[1 - e^{-j\omega}\right]$$

Filtro pasa bajo real



Filtro pasa bajo ideal



La forma general de un filtro es:

$$H(\omega) = \frac{Y(\omega)}{X(\omega)} = k\frac{N(\omega)}{D(\omega)} = k\frac{b_0 + b_1\omega + b_2\omega + ... + b_{n-1}\omega^{n-1}}{a_0 + a_1\omega + a_2\omega + ... + a_n\omega^n}$$

Donde: k es la ganancia del filtro. Las raíces de N(ω) vienen a ser los “ceros” y las raíces de D(ω) vienen a ser los “polos”. Otra forma de expresar la función de transferencia es:

$$H(\omega) = k\frac{(\omega - \omega_1)(\omega - \omega_2)...(\omega - \omega_m)}{(\omega - p_1)(\omega - p_2)...(\omega - p_n)}$$

Donde ω_j representa a los ceros y p_j representa a los polos.

Filtro Butterworth.- Es el filtro más utilizado en aplicaciones geofísicas, debido a su pequeña banda de transición y a la ausencia de ripple (ondulaciones) en la banda de paso. Por ejemplo, el rango de frecuencia del ruido sísmico de baja frecuencia para sismos locales está comprendido entre 0.1 a 1 Hz.

Implementación de un filtro “pasa alto” en Matlab:

```
fcorte=1;  % elimina debajo de 1Hz
[b, a] = butter (5,fcorte/(Fs/2),'high');
yf = filtfilt (b,a,y);
plot (t,yf,'red'), grid on, zoom xon
```

Implementación de un filtro “pasa bajo” en Matlab:

```
fcorte=1;  % elimina arriba de 1Hz
[b, a] = butter (5,fcorte/(Fs/2));  % Filtro pasabajo
yf = filtfilt (b,a,y);
plot (t,yf,'red'), grid on, zoom xon
```

PROCESAMIENTO DIGITAL DE IMÁGENES CON MATLAB

Existen muchos paquetes o software para realizar el procesamiento digital de imágenes, pero casi todos estos programas son "cajas negras" donde no se puede modificar nada y se esta supeditado a hacer lo que el programa puede realizar, puesto que no se tiene el código fuente. Sin embargo, con el toolbox de procesamiento de imágenes del lenguaje Matlab y su entorno de programación gráfica se puede construir un software de acuerdo a las necesidades o gustos particulares del usuario. En el presente trabajo se desarrolla un software con una interfaz gráfica de usuario interactiva para el tratamiento de imágenes digitales, en el cual se han implementado algunas funciones como rotación, selección de partes, zoom, ecualización de histograma, lectura-escritura de archivos, filtros, negativo, transformada de Fourier, imagen especular, etc. Una aplicación práctica es el manejo y mejoramiento de imágenes fotográficas, imágenes biomédicas, etc. La ventaja del presente software es que se tiene el código fuente, el cual puede ser compartido como software libre y puede ser modificado y mejorado por parte del usuario.

1. INTRODUCCION

Una imagen analógica es una función bidimensional de variables de posición (x,y) cuyo valor representa la intensidad o tonalidad de color.

Una imagen digital es un arreglo matricial cuyos elementos representan la intensidad o tonalidad de color. En este sentido, una imagen en tonos de gris (o blanco y negro) está representada por una sola matriz. Mientras que una imagen a color está representada por 3 matrices, tiene 3 componentes, como por ejemplo el formato RGB (red, blue, green).
Desde el punto de vista de señales y sistemas, una imagen es una señal estacionaria invariante en el tiempo.

Un pixel (picture element) es la unidad básica de una imagen digital. Es el elemento de la matriz, representado por su posición (i,j) y por su valor o tonalidad de color.

El procesamiento digital de imágenes se refiere a procesar imágenes digitales por medio de una computadora digital.

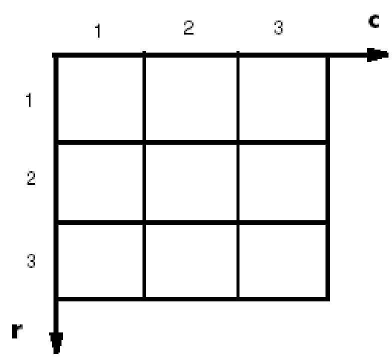


Fig. 1: Sistema de coordenadas en Matlab.

2. EL LENGUAJE DE PROGRAMACION MATLAB

Matlab (Matrix Laboratory) es un lenguaje de alto rendimiento para computación técnica y científica. Integra el cálculo, visualización y programación en un entorno fácil de usar donde los problemas y soluciones son expresados en una notación matemática familiar. Los usos típicos incluyen lo siguiente:

- Matemáticas y cálculo.
- Desarrollo de algoritmos.
- Adquisición de datos.
- Simulación y modelaje de sistemas.
- Análisis de datos, exploración y visualización.
- Gráficos en ciencia e ingeniería.
- Desarrollo de aplicaciones, incluyendo interfaces gráficas de usuario.

Matlab es un sistema interactivo cuyo elemento básico de datos es un arreglo vectorial que no requiere dimensionar.

¿Por qué utilizar Matlab? Hay varias razones para ello:

Matlab trabaja con vectores y matrices.

Es fácil de utilizar debido a su “caja de herramientas” (toolboxes).

Se puede trabajar en programación estructurada u orientada a objetos.

Es casi un estándar en ciencias e ingeniería.

Es apropiado en procesamiento digital de imágenes, donde las matrices son del orden de 1000×1000.

Ejemplo de programa en Matlab:

```
clear                         % Borra la memoria en el espacio de trabajo.
clc                           % Limpia la pantalla.
f = imread (‘venus.jpg’);     % Lee el archivo de imagen y le asigna a la matriz “f”
D = size (f);                 % Asigna a D las dimensiones de la matriz “f”
imshow (f)                    % Muestra un gráfico de la imagen.
title (‘Imagen’)              % Le coloca un título al gráfico.
imwrite (f, ‘otro.jpg’);      % Copia la matriz “f” en el archivo “otro.jpg”
```

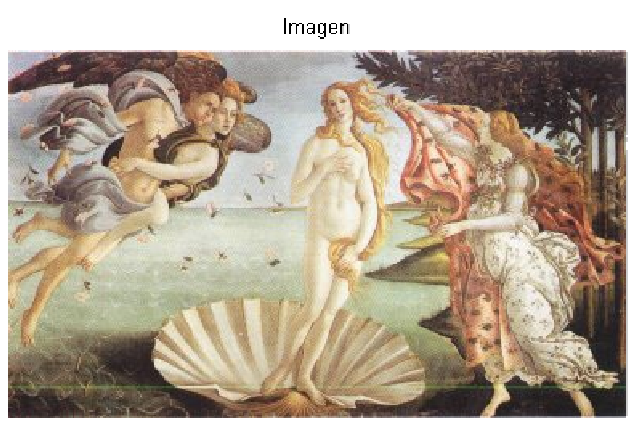


Fig. 2: El nacimiento de Venus.

3. PROCESAMIENTO DIGITAL DE IMAGENES

Desde el punto de vista del álgebra lineal, se puede considerar a las imágenes como matrices. Entonces, el procesamiento digital de imágenes será un proceso de operación de matrices. A continuación se describen algunos de estos procesos u operaciones:

3.1 Ecualización del histograma

Esta operación consiste en la distribución uniforme de los niveles de intensidad en una imagen, de esta manera, se mejora el contraste de la imagen. La ecualización del histograma se implementa en el Toolbox mediante la función "histeq", con la siguiente sintaxis:

g = histeq (f, nlev)

donde f es la imagen de entrada y nlev es el número de niveles de intensidad especificado para la imagen de salida.

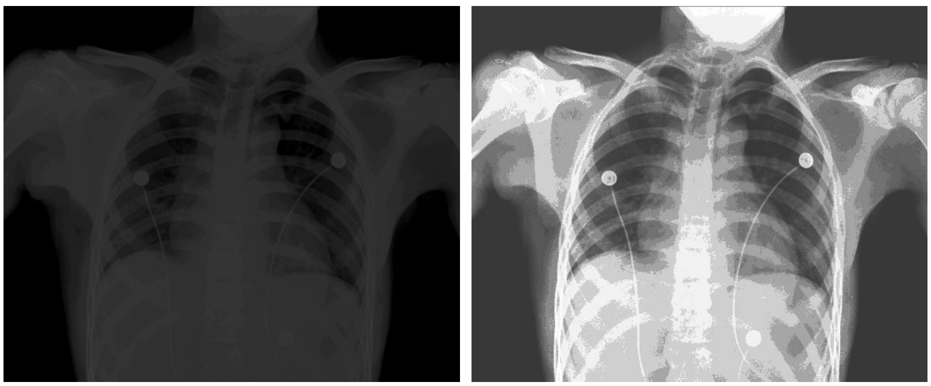


Fig. 3. Radiografía del pecho y su imagen ecualizada.

```
subplot(1,2,1), imshow(f)
title('Imagen original')
ylim('auto')
g = histeq(f,256);
subplot(1,2,2), imshow(g)
title('Imagen equalizada')
```

3.2 Rotación de imágenes

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

```
f = imread('archivo.jpg');
g = imrotate(f,25,'bilinear');
imshow(f), figure, imshow(g)
```

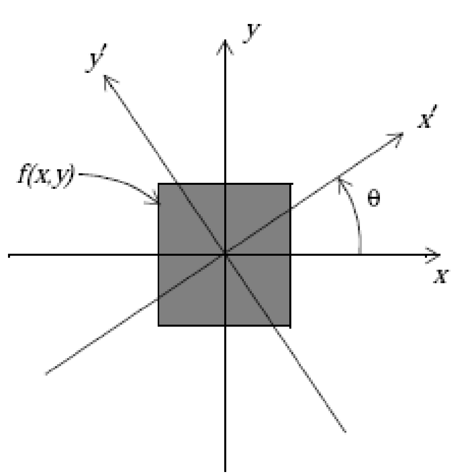


Fig. 4: Transformación de coordenadas por rotación.

3.3 Selección de áreas

```
imshow pavo.jpg
g = imcrop;
figure, imshow(g)
```

3.4 Variación del brillo

```
  if (f1(i,j) > 120 & f1(i,j) <= 192)
     brillo = brillo/8;
  end
  if f1(i,j) > 192
     brillo = 0;
  end
     g1(i,j) = uint8(double(f1(i,j)) + brillo);
```

3.5 Transformada de Fourier

Es una herramienta matemática que transforma una señal del dominio de la posición al dominio de la frecuencia.

$$F(\omega_1,\omega_2) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} f(m,n)e^{-j\omega_1 m}e^{-j\omega_2 n}$$

3.6 Filtrado de imágenes

El proceso del filtrado consiste en eliminar, atenuar o amplificar ciertas componentes frecuenciales de una señal. Por ejemplo, el filtro “mediano” es un filtro pasabajo que elimina el llamado ruido “sal y pimienta”.

```
I = imread('lena.jgp');
J = imnoise(I,'salt & pepper',0.02);
K = medfilt2(J);
figure, imshow(J), figure, imshow(K)
```

Fig. 5: Aplicación del filtro mediano: elimina el ruido “sal y pimienta”.

3.7 Conversión de imagen a color en escala de grises.

El comando “rgb2gray” convierte una imagen RGB a una imagen en escala de grises. Esta operación lo realiza eliminando la información del tono (hue) y la saturación (saturation), mientras que retiene la información de iluminación (luminance). La sintaxis es:

```
I = rgb2gray (RGB)
```

donde RGB es la matriz de la imagen a colores e I representa la matriz de la imagen en escala de grises. Si la entrada es una imagen RGB, puede ser de tipo entero "uint8", "uint16" o tipo real "double"; la imagen de salida I será del mismo tipo que la imagen de entrada.

3.8 Negativo de una imagen

```
g = imcomplement(f);
imshow(f), figure, imshow(g)
```

3.9 Imagen especular.- En este caso solo se permutan las columnas: la primera por la última, la segunda por la penúltima y así sucesivamente:

```
for j = 1:n
  k = n-j+1;
  g(:,j) = f(:,k);
end
```

4. CORRELACIÓN ENTRE DOS IMÁGENES

El coeficiente de correlación bidimensional (2D) entre dos imágenes se obtiene con el comando "corr2", mediante la siguiente sintaxis:

```
R = corr2(A,B)
```

donde A y B son matrices o vectores del mismo tamaño. Ejemplo:

```
I = imread('pout.tif');
J = medfilt2(I);
R = corr2(I,J)   % R coeficiente de correlación en 2D
```

5. INTERFAZ GRAFICA DE USUARIO

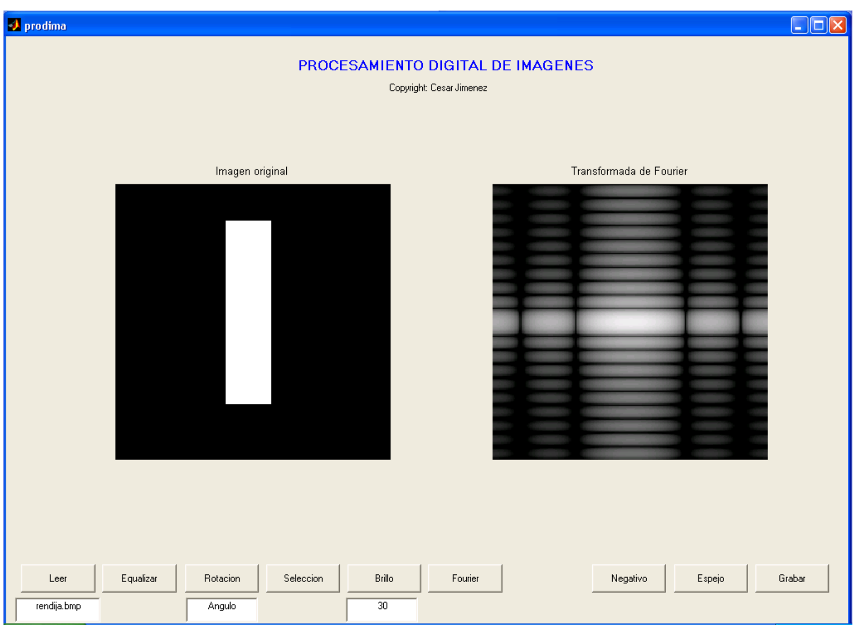


Fig. 6: Interfaz Gráfica de Usuario.

6. APLICACIONES

Procesamiento digital de imágenes fotográficas.
Procesamiento digital de imágenes biomédicas.
Simulación de patrones de difracción.
Procesamiento de imágenes de satélite.

7. CONCLUSIONES

Se ha diseñado e implementado un software para procesar imágenes digitales utilizando las herramientas del Matlab.

El software es factible de ser mejorado o adecuado a la necesidad del usuario. Puede ser compartido como software libre.

Se pueden añadir mas funciones del Toolbox de Procesamiento de Imágenes. Por ejemplo, es necesario mejorar la función de “Brillo”.

El presente software es ideal para realizar trabajos de procesamiento digital de imágenes fotográficas e imágenes biomédicas.

BIBLIOGRAFIA

1. R.C. González, Digital Image Processing using Matlab, Prentice Hall, New Jersey 2004.
2. H. Morales, Matlab 7, métodos numéricos, Editorial Megabyte, Lima 2005.
3. Building GUIs with Matlab, The Mathworks Inc., June 1997.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE CIENCIAS FÍSICAS
PROCESAMIENTO DE DATOS DIGITALES **–** Lab. 1

Implementar un programa en Matlab para resolver lo siguiente:

1) Hacer un programa que genere una matriz “cuadrado mágico” de nxn elementos y que la guarde en un archivo de datos “magico_n.txt”. Modificar el programa para que lea dicho archivo y calcule el valor máximo de la matriz y la posición correspondiente.

2) Hacer un programa para resolver la ecuación de 2do grado: $ax^2+bx+c = 0$. Los parámetros a, b y c serán introducidos desde el teclado. Debe tener en cuenta las raíces reales y complejas. Las raíces deben aparecer en la pantalla con 6 decimales.

3) Hacer un programa para resolver un sistema de ecuaciones lineales: **A X** = **B**, donde **A** es una matriz cuadrada, **B** es un vector columna y **X** es un vector columna. Los datos serán leídos desde un archivo. Las incógnitas deben aparecer en la pantalla con 4 decimales. Debe grabar las incógnitas en un archivo solucion.txt.

4) Hacer un programa para calcular la distancia entre dos puntos geográficos de latitud y longitud determinados. Considerar que la Tierra tiene una forma esférica y que la distancia NO es una línea recta, sino una longitud de arco esférica. Sugerencia: $L = R\ \theta$, donde θ es el ángulo formado por los vectores que van del centro a los puntos geográficos. Hallar la distancia entre Lima y Nueva York (en km) y compararla con la distancia obtenida a través de Google Earth.

5) El día “juliano” es el número de orden que le corresponde a una fecha dada; por ejemplo, el 01 de enero sería el día juliano 1 y el 31 de diciembre sería el día juliano 365. Hacer un programa para convertir de día juliano a fecha. ¿A que fecha corresponde el día juliano 220? Variar el programa para tener en cuenta los años bisiestos: múltiplos de 4, excepto los que terminen en 00, como el año 2000.

6) Método de Montecarlo. Se tiene un cuadrado de lado L y una circunferencia inscrita en el cuadrado. Supongamos que lanzamos pequeños dardos a gran distancia. Muchos caerán dentro y otros caerán fuera de la circunferencia. Sea: *n* = dardos que caen dentro del circulo
N = dardos que caen dentro del cuadrado.
La razón de estas dos cantidades será proporcional a la razón de las áreas del cuadrado y de la circunferencia. Hallar una aproximación de “pi” en función de *n* y *N*. Hacer un programa para hallar el valor de “pi” para un valor de *N* introducido por el usuario.

7) Hacer una gráfica en 3 dimensiones de la curva gaussiana: $z = Ae^{-(x^2+y^2)}$, donde A = 10 es la amplitud de la curva. Utilice una grilla para el dominio: -10 < x < 10 -10 < y < 10
a. Considere que la dimensión de la grilla es unitaria.
b. Considere que la dimensión de la grilla es 0.2
c. Modifique el programa para visualizar las curvas de nivel.

Prof.: Lic. César Jiménez

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE CIENCIAS FÍSICAS
PROCESAMIENTO DE DATOS DIGITALES – Lab. 2

1. Utilizando Matlab, haga un programa (function) que evalúe las funciones singulares: impulso unitario, escalón unitario y función rampa.

2. Haga un programa para visualizar la función compuerta unitaria.
a) Utilizar los comandos "zeros" y "ones".
b) Utilizar la función desarrollada en el problema 1.

3. Desarrollar un conjunto de comandos Matlab para aproximar las siguientes señales periódicas en tiempo continuo, dibujando 5 ciclos de cada una:
a) Onda Cuadrada, de amplitud 5 Volts, frecuencia fundamental 20 Hz.
b) Señal diente de sierra, amplitud 5 Volts y frecuencia fundamental 20Hz

4. La solución a una ecuación diferencial está dada por: $x(t) = 10\,e^{-t} - 5e^{-0.5t}$
Usando Matlab, grafique la solución de la ecuación en el siguiente intervalo [0,5] con una frecuencia de muestreo de 100 Hz

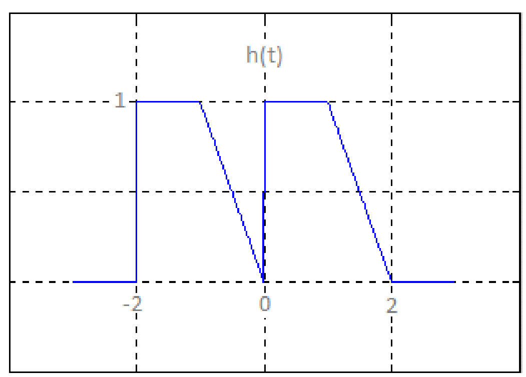
5. Repita el problema anterior para la siguiente expresión: $x(t) = 10\,e^{-t} + 5e^{-0.5t}$

6. Una señal sinusoidal con amortiguación exponencial está definida por la siguiente expresión:

$$x(t) = e^{-at}\cos(2\pi ft)$$

donde f = 1 Hz y el parámetro a es variable y toma valores sobre el siguiente conjunto: 1, 5, 20. Usando Matlab, investigar el efecto de variar dicho parámetro en la señal en el intervalo [0, 5]. Utilice una frecuencia de muestreo de 20 Hz. Calcule el valor de a para el caso de amortiguamiento crítico. Haga una gráfica.

7. Para la gráfica mostrada, haga una función en Matlab que visualice *h(t)*. Use el comando "function" Graficar:



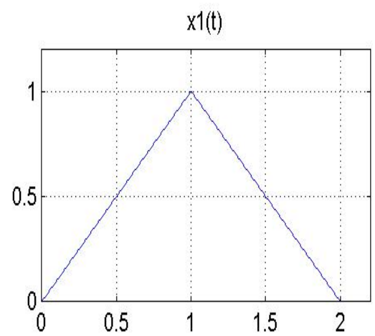
a) h(t+1)
b) h(t/2-2)
c) h(1-2t)
d) 4*h(t/4)
e) 0.5*h(t)*u(t)+h(-t)*u(t)
f) h(t/2)*δ(t+1)
g) h(t)*(u(t+1)-u(t-1))

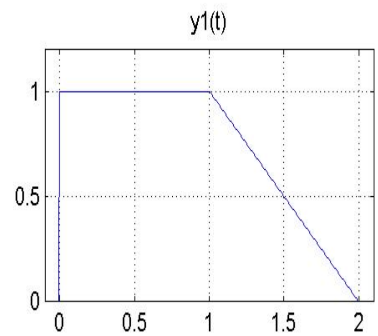
Prof. César Jiménez
http://cjimenez.741.com/pds

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE CIENCIAS FÍSICAS
PROCESAMIENTO DE DATOS DIGITALES – Lab. 3

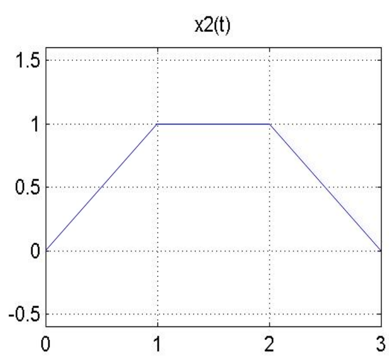
1. Sea: $x(n) = \delta(n) + 2\delta(n-1) - \delta(n-3)$ y $h(n) = 2\delta(n+1) + 2\delta(n-1)$
Calcule y haga la gráfica (usar el comando “stem”) de cada una de las siguientes convoluciones:
   a) $y1(n) = x(n)*h(n)$
   b) $y2(n) = x(n+2)*h(n)$
   c) $y3(n) = x(n)*h(n+2)$

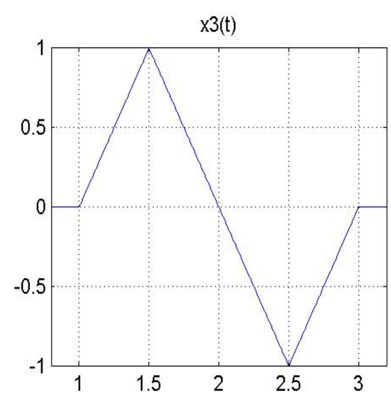
2. Considere un sistema LIT cuya respuesta a la señal x1(t) es y1(t)





Hallar las respuestas del sistema anterior a las siguientes excitaciones:

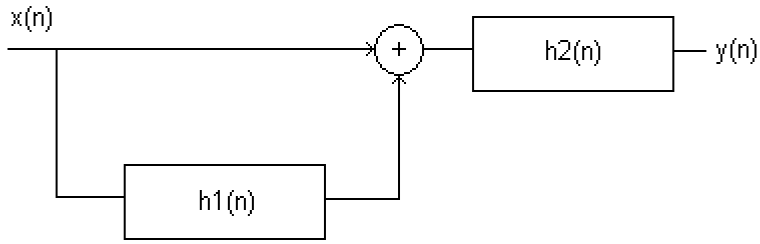




3. Calcular la convolución entre el siguiente par de señales:
a) $x(n) = (0.5)^n u(n-4)$ $h(n) = 4^n u(2-n)$
b) $x(n) = u(-n) - u(-n-2)$ $h(n) = u(n-1) – u(n-4)$
c) $x(n) = u(n)$ $h(n) = (1/2)^{-n} u(-n)$
d) $x(t) = \exp(-at) u(t)$ $h(t) = \exp(-at) u(t)$

Verificar en Matlab las convoluciones obtenidas en los incisos anteriores.

4. a) Escribir la ecuación en diferencias que relaciona la entrada con la salida para el sistema de la figura mostrada, donde:
$h1(n) = \beta\, \delta(n-1)$
$h2(n) = \exp(\alpha)\, \delta(n)$



b) Hallar α y β, de tal forma que la salida sea el promedio entre la entrada en el instante n y la entrada en el instante n-1.

5. Dada la siguiente ecuación en diferencias: $y(n) = -a\, y(n-1) + b\, x(n) + c\, x(n-1)$, realizar una representación en diagrama de bloques.

6. Realizar en Matlab la convolución del siguiente par de señales:

$x(n) = (-1)^n\, (u(n) - u(-n-8))$

$h(n) = u(n) - u(n-8)$

Graficar la señal resultante: $y(n) = x(n) * h(n)$. Usar el comando “stem”

7. Considere un sistema lineal e invariante en el tiempo, causal, cuya entrada x(n) y salida y(n) estén relacionadas por la ecuación de diferencias:

$$y(n) = 0.25\, y(n-1) + x(n)$$

Determine y(n) si $x(n) = \delta(n-1)$. Grafique en Matlab la salida y(n), use el comando “stem”

Lic. César Jiménez T.

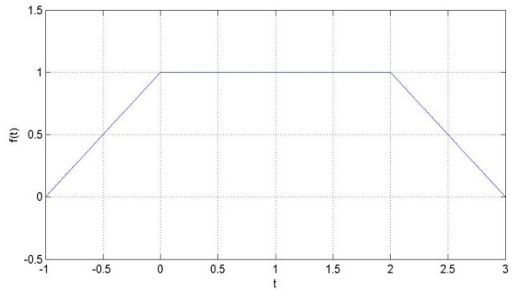
UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE CIENCIAS FÍSICAS
PROCESAMIENTO DE DATOS DIGITALES – Lab. 4

Hallar la transformada de Fourier de las siguientes señales y graficar en Matlab:

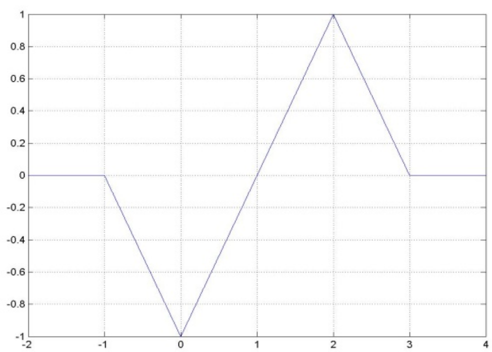
a) f(t) = G(t/2) – G(t)
   G : función compuerta

b) f(t) = 5G(t-1) – G(t+1)

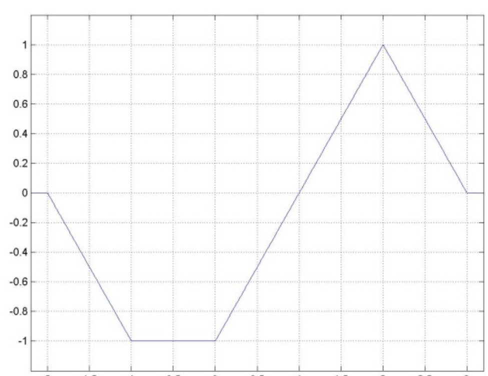
c)



d)

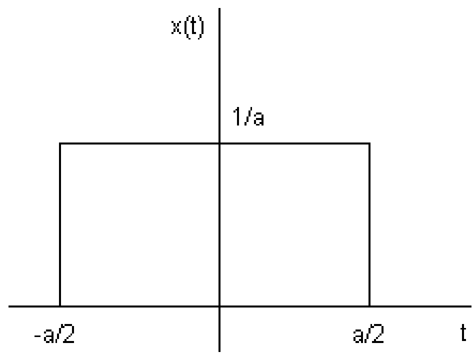


2. Hallar la transformada de Fourier de x(t) y graficar en Matlab:



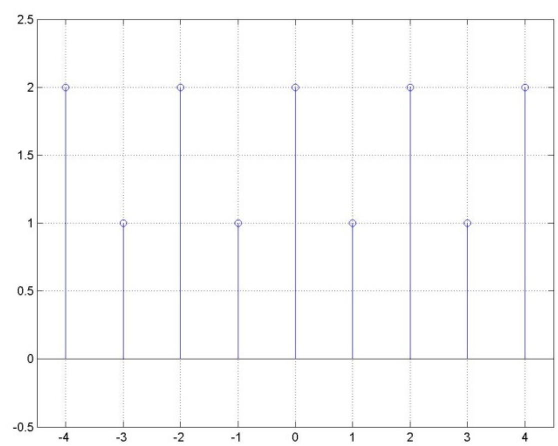
Si *x(t)* pasa a través del bloque de la figura, calcule la transformada de Fourier de *y(t)*.

3. Halle y grafique la transformada de Fourier de las siguientes señales:
a)



b) La función periódica mostrada en la figura: donde: $-\infty < n < \infty$

x(n) = 2 si n: par
x(n) = 1 si n: impar



c) $f(t) = t^2 \exp(-t/\tau)$

4) Dada la señal en el dominio del tiempo:

y(t) = sin(t) + 0.25sin(10t)

a) Hacer un programa para graficar la señal para 4 periodos, con una frecuencia de muestreo de 100 Hz.
b) Hacer un programa para graficar el espectro de frecuencias de la señal.
c) ¿Cual es la amplitud y la frecuencia correspondiente a cada pico?

5) Baje el archivo "datos.txt" de la web, que representa una señal de audio:

http://cjimenez.741.com/pds/datos.txt

La frecuencia de muestreo es de Fs = 8000 Hz. Hacer un programa en Matlab para que realice lo siguiente:

a) Hallar el número de datos N.
b) Hallar la duración de la señal.
c) Hallar el valor medio de la señal.
d) Graficar la señal x(t)
e) Graficar el espectro de frecuencias.

6) a) Calcule en forma analítica y grafique la transformada de Fourier de la función triángulo: x(t) = triang(t/2)

b) La integral que define la TF puede calcularse numéricamente, para cada valor de frecuencia, utilizando la suma de Riemman. Para subdominios de T se tiene:

$$X(f)=\sum_{N=-\infty}^{\infty}\Lambda(nT/2)\exp(-j2\pi f nT)T$$

Calcular para T=0.8 y para el rango de frecuencia de 0 a 2 con intervalos de 0.125 ejecutando las siguientes sentencias en Matlab:

```
T = 0.8;
n = [-2:2];
f = [0:0.125:2];
X = zeros(size(f)) ;
for i = 1: length(f)
X(i) = sum(T*triang(n*T/2).*
        exp(-j*2*pi*f(i)*n*T);
end
```

c) Repetir para T 10 veces menor.

Nota: La función "triang" es la función triángulo unitario Λ(t) con amplitud 1 y ancho 1 que Ud. debe implementar en Matlab.

Lic. César Jiménez T.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE CIENCIAS FÍSICAS
PROCESAMIENTO DE DATOS DIGITALES – Lab. 5

Dada la señal en el dominio del tiempo:

$y(t) = \sin(t) + 0.25\sin(10t)$

a) En la gráfica del espectro de potencias, hallar la amplitud, la frecuencia y el periodo correspondiente a cada pico.
b) Diseñe un filtro pasabajo y grafique la señal filtrada. Cuál es la frecuencia de corte?
c) Diseñe un filtro pasa-alto y grafique la señal filtrada.

2. Reconocimiento de locutor.- Con el programa de adquisición de datos "adqsound.m", realice la grabación durante 3 s de las voces de 2 personas diferentes.
a) Aplique un filtro pasa-alto con frecuencia de corte de 100 Hz, grabe el archivo obtenido.
b) Calcule la Transformada de Fourier discreta de las dos señales. Grafique.
c) Calcule el coeficiente de correlación entre las TRF de ambas señales. ¿Qué puede concluir?

3. Baje el archivo "pardat.txt" que contiene una señal mareográfica real medida en la bahía de Paracas con un sensor de nivel ultrasónico:
http://cjimenez.741.com/pds/lab4/pardat.txt

a) Aplique un filtro mediano (medfilt1.m) para eliminar los picos impulsivos. Grafique la señal antes y después de aplicar dicho filtro.
b) Representar la señal en el dominio de la frecuencia. Identificar los picos principales y periodos de retorno.
c) Aplicar un filtro adecuado para estudiar las mareas: ¿cuál es el periodo y amplitud de la marea?
d) Aplicar un filtro adecuado para estudiar las olas: ¿cuál es el periodo y amplitud de las olas?

4. Se tiene una señal sísmica de 3 componentes (en el archivo "nana.mat"):
http://cjimenez.741.com/pds/lab4/nana.mat

La ubicación del epicentro fue:

Lat = -15.36°
Lon = -70.90°
Prof = 180 km

a) Graficar la señal para las 3 componentes: Vertical (V), Norte (N), Este (E) en función del tiempo t.
b) Hallar el tiempo de arribo de la fase P y S.
c) Hallar el contenido energético promedio de la señal.
d) Hallar la magnitud del sismo. Ver artículo:
http://cjimenez.741.com/pds/magnitud.pdf

5. Buscar los datos del precio del dólar desde el 01 Ene hasta el 30 Nov del presente año.
a) Completar la serie de tiempo para los sábados, domingos y feriados mediante interpolación simple. Graficar.
b) Representar la serie en el dominio de la frecuencia. Identificar los picos principales y periodos de retorno.
c) Filtre las fluctuaciones de alta frecuencia y grafique. ¿Cuál es la tendencia del precio del dólar?
d) Pronostique el precio del dólar para el 31 Dic del presente año, en base a interpolación (en este caso extrapolación) cúbica. Sugerencia:

```
yi = interp1(t,y,ti,'cubic');
```

Lic. César Jiménez T.