ETSI Telecomunicación

# Laboratorio de Señales y Sistemas

Apuntes de Pak

**Laboratorio de Señales y Sistemas**

Apuntes de Pak (Francisco José Rodríguez Fortuño)
ETSI Telecomunicación. Universidad Politécnica de Valencia.
Primer cuatrimestre de $3^{er}$ curso
Curso 2005/2006

Contenido:

- Referencia rápida
- Examen resuelto
- Guiones de las prácticas

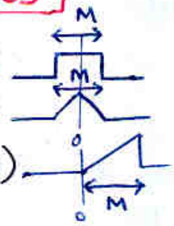
**Fecha de última actualización:** 23 Agosto 2007

# Lab SyS – Referencia Rápida

## Prac 1. Señales y Sistemas Continuos

escalon(t)
delta(t)

pulso(t, M)
trian(t, M)
rampa(t, M)



$xn = x(-n)$ se genera a mano

par: $x_p = \frac{x + xn}{2}$

impar: $x_i = \frac{x_i - x_n}{2}$

dibut(x) convol(x1, x2)

cero(t)
uno(t)

men = mensaje [t, [0 1 1 0])
trxs trxn

- DSF señales periódicas:

X = abs(especsf(x))
dibuf(X)

## Prac 2. Señales y sistemas discretos

$$x(n) = \sum_{k=0}^{N-1} a_k e^{j\frac{2\pi}{N}kn} \longrightarrow \boxed{\begin{matrix} h(t) \\ H(f) \end{matrix}} \longrightarrow y(n) = \sum_{k=0}^{N-1} \overbrace{H\left(\tfrac{2\pi}{N}k\right)a_k}^{b_k} e^{j\frac{2\pi}{N}kn}$$

a = coef_DSF(x)

si $\left.\begin{matrix} h(n) = \left(\frac{1}{a}\right)^n u(n) \\ H(\Omega) = \frac{1}{1 - \frac{1}{a}e^{-j\Omega}} \end{matrix}\right\}$ $H\left(\frac{2\pi}{N}k\right)$ = pract_H(a, N)

$\Downarrow$

$b_k = a_k$ .* pract_H(a, N)

y = x_DSF(b)

- Pract conv

$x \in [nx_{ini}, nx_{fin}]$
$y \in [ny_{ini}, ny_{fin}]$
$(x * y) \in [nx_{ini} + ny_{ini}, nx_{fin} + ny_{fin}]$
length(x) = $nx_{fin} - nx_{ini} + 1$

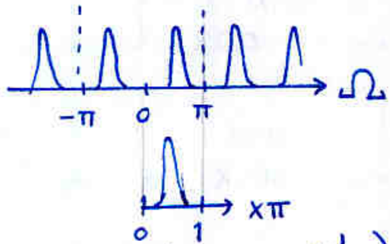
NOTA
$\cos(2\pi \cdot \frac{m}{N} \cdot n)$
$\downarrow$
periodo

## Prac 3. Muestreo

$x_a(t)$

$x(n) = x_a(nT_s) \longrightarrow$

$X(\Omega)$:
- $2\pi$ - periódica (puede implicar aliasing)
- Amplitud multiplicada por $f_s$
- Frecuencia normalizada




$\Omega = \pi \cdot \left(\frac{f}{f_s/2}\right)$

- Sinusoide frec $f_0$ muestreo $f_s$

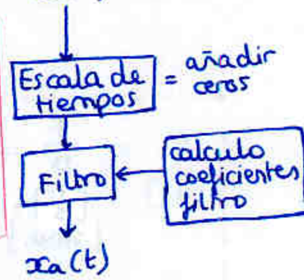
n = 0 : 1 : tend · fs
$x = \text{sen}(2\pi f_0 \cdot nT_s)$

- Otra forma de muestrear:
Tomar un subconjunto de valores
conversor_AD(x, fsim, fs)

- Conversión DA

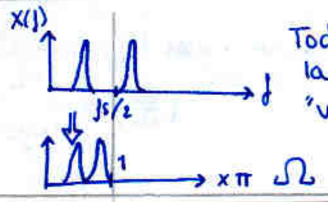
x_con_ceros: añadir $\frac{f_{sim}}{f_s} - 1$ ceros entre cada muestra
[B, A] = cheby2 (orden, atenuación, $\boxed{\frac{f_s/2}{f_{sim}/2}}$)
xa_recuperada = filter(B, A, x_con_ceros)

frec corte ($f_s/2$) normalizada a $f_{sim}/2$



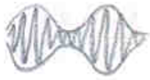

Para ver el aspecto del filtro: resp_freq_filtro (orden, aten, fs, fsim)

Aliasing

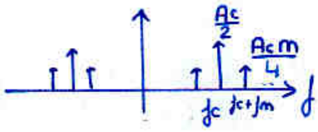



Todo lo que "cruza" la frec maxima fs/2 "vuelve hacia atras"

# Practica 4. Modulaciones Analógicas



**AM:** $y(t) = A_c\,[1 + m\,\underbrace{x(t)}_{\cos(2\pi f_m t)}]\cos(2\pi f_c t)$




DBL igual pero sin $f_c$

**FM y PM:**

PM → fase instantánea $\psi(t) = w_c t + \phi_d\, x(t)$

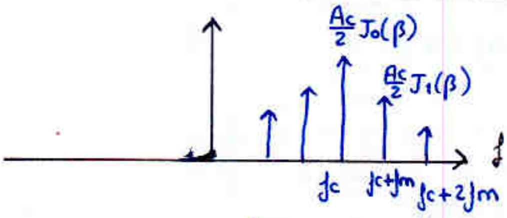
FM → frec instantánea $f(t) = f_c + f_d\, x(t)$

$f(t) = \frac{1}{2\pi}\frac{d\psi(t)}{dt}$

si $x(t) = \cos(2\pi f_m t)$ ⇒

$\beta = \frac{\Delta f}{f_m} = \frac{f_d \cdot A_m}{f_m} = \frac{f_d}{f_m}$ ($A_m = 1$)

maxima desviación de frecuencia = desviación frecuencia




Carson: $\boxed{B = 2(\beta+1)\cdot f_m}$

$(\beta+1)$: nº de rayas espectrales -1

$f_m$: ancho entre rayas = frec moduladora

$\boxed{\beta = \frac{f_d}{f_m}}$

$\beta$ = indice de modulación = máxima desviación de fase (por eso está $f_m$ dividiendo, por haber hecho la integral)

$\psi(t) = 2\pi \int f(t)\,dt$

NOTA: el $2\pi$ se cancela arriba y abajo

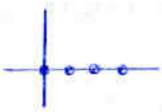
NOTA acerca de Tsim

valores recomendados:

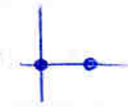
| | Practica Mod. Analogicas | Practica Mod. Digitales |
|---|---|---|
| visualización temporal | $T_{sim} = \frac{1}{15 f_c}$ | $T_{sim} = \frac{D}{150}$ |
| visualización espectral | $T_{sim} = \frac{1}{4 f_c}$ | $T_{sim} = \frac{D}{40}$ |

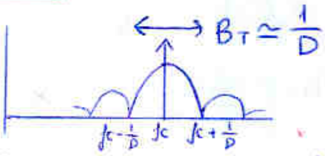
# Practica 5. Modulaciones Digitales

**ASK**



M=2 OOK

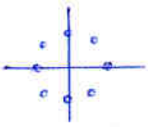




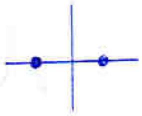

mitad del lobulo principal

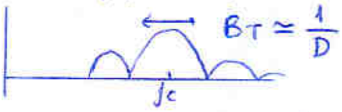
D: intervalo de simbolo

**PSK**



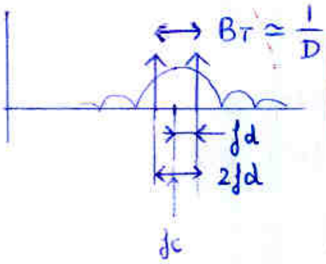
M=2 BPSK





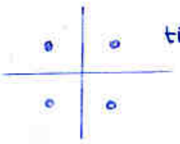

**FSK** $f_1, f_2, f_3, f_4, \dots$

M=2 BFSK $\boxed{\begin{aligned} f_1 &= f_c + f_d \\ f_2 &= f_c - f_d \end{aligned}}$




en BFSK la separacion entre las 2 deltas es $2f_d$. No confundir con la FM donde la separación entre $\infty$ deltas es $f_m$ ($f_d$ no se ve involucrado en el eje de frecuencias de FM, sólo en la $\beta$ q. determinará amplitud de las deltas) (en BFSK $f_d$ es determinante)

**M-QAM**



tipico M=4

$$\boxed{\underset{[Hz]}{B_T} = \frac{1}{\underset{[s/simb]}{D}} = \underset{[simb/s] = [Bd]}{V_{modulacion}}}$$

$$\boxed{\underset{[bit/s]}{V_{binaria}} = \underset{[bit/simb]}{\log_2 M} \cdot \underset{[simb/s]}{V_{mod}}}$$

$\frac{N_0}{2} = \sigma^2$

$N_0$: $[\frac{W}{Hz}]$ del ruido; $\sigma^2$: varianza del ruido

Nota

$$\left(\frac{S}{N}\right) = \frac{2\cdot\left(\frac{A_c^2}{4}\right)}{2\cdot\left(\frac{N_0}{2}\cdot B_T\right)} = \frac{\frac{A_c^2}{4}}{\frac{N_0}{2}\cdot\frac{1}{D}} = \frac{\frac{A_c^2}{2}D}{N_0} = \frac{E_b \log_2 M}{N_0} = \left(\frac{E_b}{N_0}\right)\cdot \log_2 M$$

$\left(\frac{E_b}{N_0}\right)$: se llama a veces $\left(\frac{S}{N}\right)$ normalizada a bit

$$\left(\frac{E_b}{N_0}\right) = \frac{(S/N)}{\log_2 M}$$

# Laboratorio de Señales y Sistemas

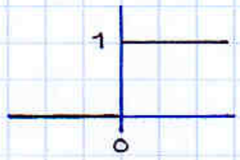
## PRACTICA 1

tren(t, T)

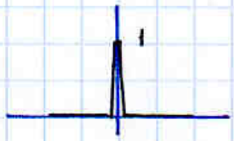

### Señales Básicas

dibut(...)

escalon(t)

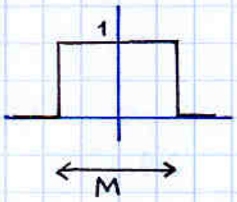



delta(t)

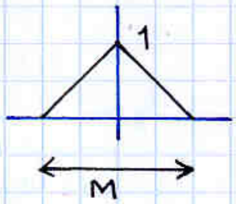



exp(t)

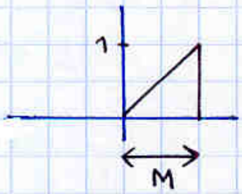

pulso (t, M)




trian(t, M)




rampa (t, M)




### Transformaciones

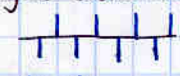
$a(bt + c)$

**simetria par o impar**

Si tenemos $x$, no sirve $xn = x(-t)$
hay que generar $xn$ manualmente

Parte par: $x_p = (x + xn)/2$ Parte Impar: $x_i = (x - xn)/2$

Tren pulsos signo alterno:



tren1 = tren(t,1)
tren 2 = tren(t-1, 2)
tren = tren1 - 2· tren 2

### convolución

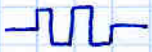
convol (x1, x2)

### Correlación

entorno1
ejet = axc;

cero(t)

uno(t)



filtro adaptado $h(t) = s^*(-t) \overset{s \text{ real}}{=} s(-t)$

i.e. $h0 =$ cero $(-t)$
$h1 =$ uno $(-t)$

```
men = mensaje(t, [0 1 1 0])
rs  = trxs(men)   % retardo
rn  = trxn(men)   % retardo y ruido

convol(rn, h0)
convol(rn, h1)
```

entorno 2

### Desarrollo en serie de Fourier

: devuelve deltas de altura correspondiente

DSF = especsf(x)

abs (DSF) real(DSF) dibuj(...)
angle (DSF) imag(DSF)

### Transformada de Fourier

: las deltas llegan a infinito

TF = espectf(x)

Diferenciacion e Integración: deriva(x), integra(x)
Modulación: producto ⟷ convolución
Retardo e inversión: retardo ⟷ $e^{jt\cdot}$ (cambia fase)

Señales y Sistemas

# Práctica 2. Señales y sistemas discretas

**Señales básicas:** sen, cos, delta(n) — son periódicas si $sen(2\pi \frac{m}{N} \cdot n)$ ($N$ → periodo)

NOTA: subplot(mnp) — m: filas, n: columnas

- cualquier señal es sumatoria de deltas.

[x,n] = pract_seno (A, pulsacion, fase, inicio, fin)

[x,n] = pract_escalon (inicio, fin)

[x,n] = pract_exp (a, inicio, fin) $\quad x(n) = a^n$

[x,n] = pract_suma_exp (a,L) $\quad x = \sum_{n=0}^{L} a^n = \frac{1-a^L}{1-a}$

[x,n] = pract_exp_compl (a, pulsación, fase, inicio, fin) $\quad x(n) = A \cdot e^{j(\omega n + \theta)}$

## Convolución

[z,n] = pract_conv (x, nx, y, ny)

- Identidad $\quad x(n) * \delta(n) = x(n)$
- conmutativa $\quad x_1 * x_2 = x_2 * x_1$
- asociativa $\quad x_1 * x_2 * x_3 = (x_1 * x_2) * x_3$
- distributiva

## Análisis Fourier

a = coef_DSF(x) ← pasarle sólo un periodo

a: vector contiene coef 0 → N−1

x = x_DSF (a)

cuidado al representar, usar
stem (real (a))
stem (real (x))

- **señal a la salida de un sistema**

$$b_k \longrightarrow \boxed{H(\Omega)} \longrightarrow c_k = H\left(\frac{2\pi}{N}k\right) \cdot b_k$$

$\frac{2\pi}{N}$: pulsacion correspondiente a $a_k$

H = pract_H (a, N) $\quad$ calcula $H(\frac{2\pi}{N} \cdot k)$ desde $k=0$ a $k = N-1$

N: periodo de la señal

siendo $H(\Omega) = \frac{1}{1 - \frac{1}{a} e^{-j\Omega}}$

$h(n) = \left[\frac{1}{a}\right]^n \cdot u(n)$

Luego hay que hacer:

c = b .* H

recuerda
cos (2 * pi * $\frac{m}{N}$ * n) — N es el periodo

ej: $\cos(\frac{8}{9}\pi n)$ periodo 9

en la practica se pone el número de valores de la entrada

i.e.
```
n = 0 : 1 : 20
b = cos (8 * pi * n / 9);
h = pract_H (a, 21)
```
21 → 0 : 20

# Lab. Señales y Sistemas. Practica 3 MUESTREO

## MUESTREO EN EL DOMINIO TEMPORAL

$x_a(t) \rightarrow x(n) = x_a(nT) \qquad T = 1/f_s$

- sinusoide frec $f_0$, muestreo $f_s$: $n = 0:1:f_s \cdot t_{end}$
  $x = \text{sen}(2\pi f_0 \cdot n \cdot T_s)$ se puede recuperar si $f_s \geq 2 \cdot f_0$

- Chirp $f_i(t) = f_0 + \mu \cdot t$
  $c_a(t) = \cos(2\pi f_0 t + \pi \mu t^2)$ $\quad f(t) = \frac{1}{2\pi}\frac{d\varphi}{dt}$
  $c(n) = c_a(\boxed{n/f_s})$ con $n = 0:1:t\_end$
  hay solapamiento a partir de $f_i(n) = f_0 + \mu(n/f_s) > f_s/2$
  despejar $n$

## MUESTREO EN EL DOMINIO DE LA FRECUENCIA

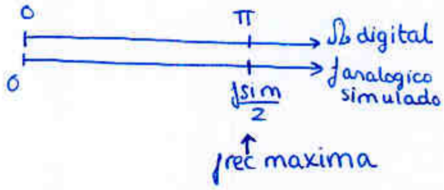
'Simulamos' las señales analógicas:
- $f_{sim} = 100 \cdot 10^3$ Hz
- $t = 0 : T_{sim} : t_{end}$
- $x(t)$ nota: $t_{end} = (\text{length}(x) - 1) \cdot T_{sim}$

dibuja $(x, T_{sim})$
dibuja_tf $(x, T_{sim})$

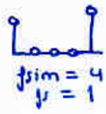
- CONVERSION AD: tomar un subconjunto de las muestras de la analogica
  conversor_AD($x$, $f_{sim}$, $f_s$)
  dibuja_tf_senal_muestreada($x$) → usa funcion de matlab $X = \text{freqz}(x)$

- FILTRO RECONSTRUCTOR:
  Inicio de la banda atenuada analogica $f_s/2$
  " " " " " digital equivalente $W_n = \dfrac{(f_s/2)}{(f_{sim}/2)}$ ← ¿porque no hay $\pi$?

  resp_freq_filtro (orden, aten, fs, fsim):
  [B, A] = cheby2 (orden, atenuacion, $(f_s/2)/(f_{sim}/2)$)
  [H, W] = freqz (B, A)
  (H: resp frec; W: Ω digital)
  eje_frec = $W \cdot \dfrac{(f_{sim}/2)}{\pi}$ plot(eje_frec, $10 \log H$)




- CONVERSION DA:
  x_con_ceros = x añadiendo $\frac{f_{sim}}{f_s} - 1$ ceros entre cada muestra
  xa_recuperada = filter(B, A, x_con_ceros) $\cdot \left(\frac{f_{sim}}{f_s}\right)$? ← ajuste de amplitud
  conversor_DA ($x$, $f_{sim}$, $f_s$, orden $N$, atenuacion $R$)




- PROCESO COMPLETO
  xa
  x ← conversor AD
  x con zeros ← conversor DA
  xa_recuperada
  → [x, x_zeros, xa_rec] = simula muestreo ($x_a$, $f_{sim}$, $f_s$, $N$, $R$)

Laboratorio Señales y Sistemas

Práctica 4 : Modulaciones Analógicas

>> load config4

**NOTA**

$T_{sim} = \frac{1}{4 f_c}$ para espectro

$T_{sim} = \frac{1}{15 f_c}$ para temporal

Simulink:

- Simulation Parameters: start time, stop time
- Cajas mas usadas:
  - Simulink > Sources (Signal Generator)
  - Simulink > sinks (scope)(to workspace)
  - Simulink Extras > Additional Sinks (d.e.p.)
  - Simulink > Math Operations (gain)
  - DSP Blocksets > Filtering > Filter designs (butter/ filter)
  - Communications > Modulation (AM, FM, ...)

· Ruido Filtrado

White noise — Filter — Averaging power Spectral density

Butterworth

Edge frequency (0→1) : $\boxed{f_c \cdot 2 \cdot T_{sim}}$ (ya que la frec máxima es $\frac{f_{sim}}{2}$ y esto daría 1)

↑ frec corte que queramos

· Lectura y escritura de datos

para ver una señal: → $\boxed{\text{to\_workspace}}$ x

y luego

```
dibuja(x,Tsim)
dibuja_tf_bi(x,Tsim)
dibuja_tf_bi_dB(x,Tsim)
```

} a veces puede ser buena idea coger solo unas muestras de x. ej 1024

· Modulación AM

La modulación AM que nosotros estudiamos $y(t) = (1 + m x(t)) \cos(w_c t)$ se consigue: → DSB AM → ▷ m

$\left(\frac{1}{m} + x(t)\right)\cos(w_c t)$

↑ input signal offset

variando índice modulación podemos experimentar (amplitud deltas, sobremodulación, ...)

· Modulacion DBL

· Modulación en frecuencia:

Ancho Banda = $2(D+1)W$
Carson

↑ para un tono $D = \beta = \frac{f_d \cdot A_m}{f_m}$

$f_d$ → constante de modulación

$f_m$ → frecuencia de la moduladora

Vamos variando $f_d$ → FM

modulation constant: $f_d$

Relación señal a ruido de pre y postdetección

Para calcular la potencia de una señal: To workspace a una variable $x$ y haces: pot = (mean(x)).^2 + var(x)

Calcular potencias de ruido:
- desconectar entrada
- fijamos $N_0$
- medimos potencia predeteccion y postdeteccion

Calcular potencia señal:
- $N_0 = 0$

Lab Señales y Sistemas

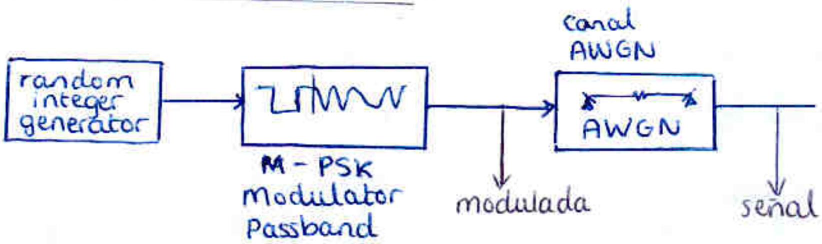
# Practica 5. Modulaciones Digitales

NOTA

$T_{sim} = \frac{D}{40}$ para ver el espectro

$T_{sim} = \frac{D}{150}$ para ver temporal

- Modulación Simulada:




Procesos aleatorios: hay que visualizar la d.e.p (no la tf).
Función de Matlab

$>> psd(\text{señal}, 1024, \frac{1}{T_{sim}}, 512)$

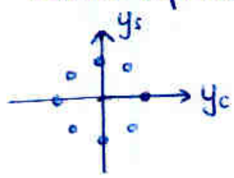
- Modulación para ver constelación fase-cuadratura

Arriba, el modulador hacia señal paso banda real

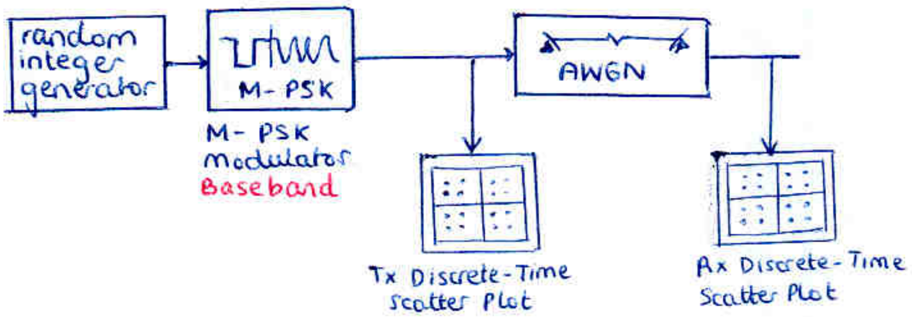
$$y(t) = y_c(t)\cdot\cos(\omega_c t) - y_s(t)\cdot \text{sen}(\omega_c t)$$

Ahora, éste modulador banda base hace equivalente banda base (compleja)

$$y_{bb}(t) = y_c(t) + j\cdot y_s(t)$$ Permite representar la constelación




NOTA: la FSK no se representa adecuadamente en el plano $y_c - y_s$, ya que va cambiando




Nota: representan $y_{bb}(t)$ en determinados instantes

- Parámetros

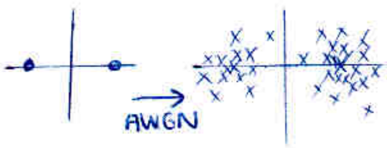
$D(s)$: periodo simbolo $= 1/R_s$    EbNo: $E_b/N_0$ (dB)

$f_c(Hz)$: frec. portadora    Tsim

- PSK

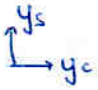
Tiempo: $T_{sim} = D/150 \rightarrow$ dibuja (xmodulada, Tsim)

Frecuencia: $T_{sim} = D/40 \rightarrow$ psd (xmodulada, 1024, $\frac{1}{T_{sim}}$, 512)




$\frac{E_b}{N_0}\downarrow \rightarrow$ mas dispersos

## FSK

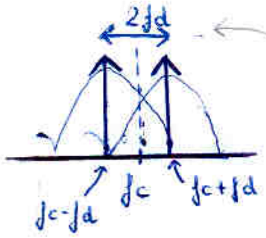
constelación 


$$\text{"1"} \quad \cos(2\pi(f_c + f_d)\cdot t) = \overbrace{\cos(2\pi f_d t)}^{y_c(t)}\cdot\cos(2\pi f_c t) - \overbrace{\text{sen}(2\pi f_d t)}^{y_s(t)}\cdot\text{sen}(2\pi f_c t)$$

$$\text{"0"} \quad \cos(2\pi(f_c - f_d)\cdot t) = \cos(-2\pi f_d t)\cdot\cos(2\pi f_c t) - \text{sen}(-2\pi f_d t)\cdot\text{sen}(2\pi f_c t)$$

$y_c(t)$ e $y_s(t)$ varían con el tiempo. No tiene sentido representarlos como una constelación.

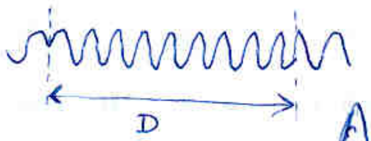
Frecuencia



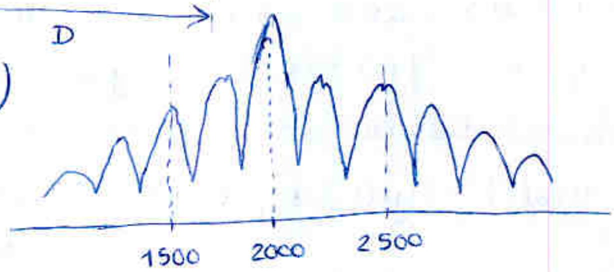

Nota: En simulink, hay un parametro llamado Frequency Separation (Hz) que corresponde a $2f_d$

## QAM

dibuja (senyal, Tsim)




psd (senyal, 1024, $\frac{1}{Tsim}$, 512)



# Laboratorio de Señales y Sistemas (5º cuatrimestre)

## EXAMEN FINAL ENERO 2005

Profesores: José Javier López, Narcís Cardona, Gema Piñero, Miguel Ángel Rodríguez

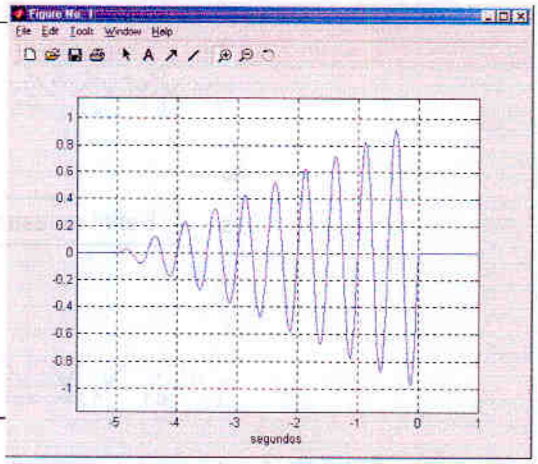
| C1 | C2 | C3 | C4 | C5 |
|---|---|---|---|---|
| | | | | |

- Responda en el espacio habilitado para cada cuestión. Todas ellas tienen igual puntuación.
- **Tiempo máximo: 1 hora**

## Cuestión 1

Escriba las instrucciones necesarias para calcular la respuesta al escalón de un sistema lineal e invariante continuo cuya respuesta impulsional h(t) es la de la figura.

```
» entorno1
>> h= sin(2*pi*2*t).*rampa(t+5,5);
>> u = escalon(t);
>> y = convol(h,u); ← convol, no conv
>> dibut(y);
```




## Cuestión 2

Se quiere representar en una gráfica la señal y(n) entre n=0 y n=20. Dicha señal y(n) es la salida del sistema lineal e invariante $h(n) = \left(\frac{1}{4}\right)^n u(n)$ cuando la entrada es $x(n)=\cos(8\pi n/9)$.

Para ello se dispone de un programa MATLAB, del que se han borrado partes de algunas instrucciones. Se pide completar las partes borradas de las instrucciones.

**Nota:** en cada línea no puede haber más de una instrucción, no se pueden añadir nuevas líneas y la parte que se mantiene de cada instrucción no se puede modificar.

```
> n= 0:1:20 ..........................;
> x= cos(8*pi*n/9..................); % genera x para la siguiente instrucción
> a= coef_DSF(x);
> b= pract_H(4,21......................);
> c= a*b .................................. ; % genera c para la siguiente instrucción
> y=x_DSF(c);
> stem( n, abs(y)..............................); representa y(n) entre n=0 y n=20
```

elimina valores complejos producidos por errores de redondeo que si aparecen estropean la gráfica.

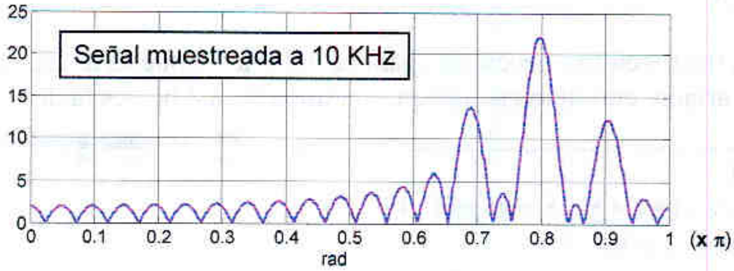
Apellidos y Nombre: ____________________

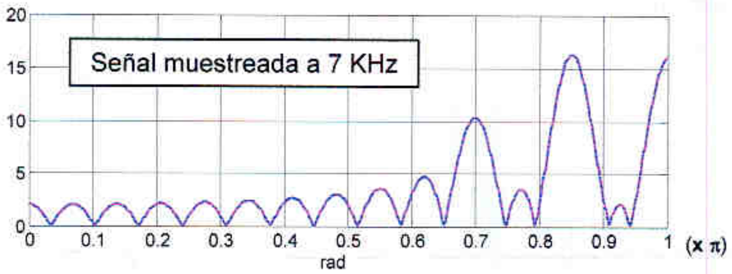
## Cuestión 3

Sea una señal continua $x(t) = (1+\cos(2\pi f_0 t))\cos(2\pi f_c t)$, donde $f_0 = 500\ Hz$ y $f_c = 4\ KHz$.
Dicha señal *x(t)* se muestrea con dos frecuencias de muestreo distintas:

- $f_{s1}$ = 10 KHz.
- $f_{s2}$ = 7 KHz.

De las señales muestreadas resultantes se calculan sus espectros correspondientes y se representan mediante la función *dibuja_tf_senal_muestreada* obteniendo las siguientes gráficas:




a) Calcule el valor exacto en radianes de la posición de los picos significativos:

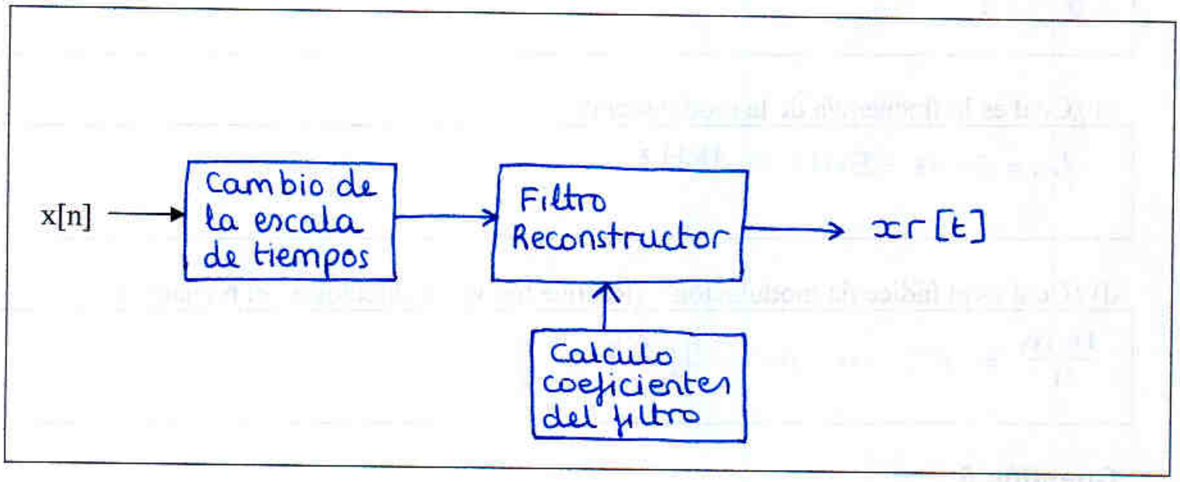
| Señal muestreada a 10 KHz | Señal muestreada a 7 KHz |
|---|---|
| 4kHz → pico central $\Omega = 2\pi\cdot\left(\frac{f_c}{f_{s1}}\right) = 2\pi\cdot\left(\frac{4k}{10k}\right) = \pi\cdot 0'8 = 0'8\pi$ rad<br>pico izquierdo $\Omega = 2\pi\cdot\left(\frac{3'5}{10}\right) = 0'7\pi$ rad<br>pico derecho $\Omega = 2\pi\cdot\left(\frac{4'5}{10}\right) = 0'9\pi$ rad | pico 4kHz – $\Omega = 2\pi\left(\frac{4k}{7k}\right) = \left(1+\frac{1}{7}\right)\pi$ que por ser periódica se convierte (aliasing) en $\left(1-\frac{1}{7}\right)\pi = \frac{6}{7}\pi$ rad<br>4'5kHz – $\Omega = 2\pi\left(\frac{4'5}{7}\right) = \left(1+\frac{2}{7}\right)\pi$ pasa a ser $\frac{5}{7}\pi$ rad<br>3'5kHz $\Omega = 2\pi\left(\frac{3'5}{7}\right) = \pi$ rad |

b) Explique si ambas señales pueden ser recuperadas originalmente:

| Señal muestreada a 10 KHz | Señal muestreada a 7 KHz |
|---|---|
| Si que se puede ya que se cumple el criterio de Nyquist $f_s > 2\cdot f_{max}$ | No se puede, no se cumple el criterio de Nyquist y por tanto |

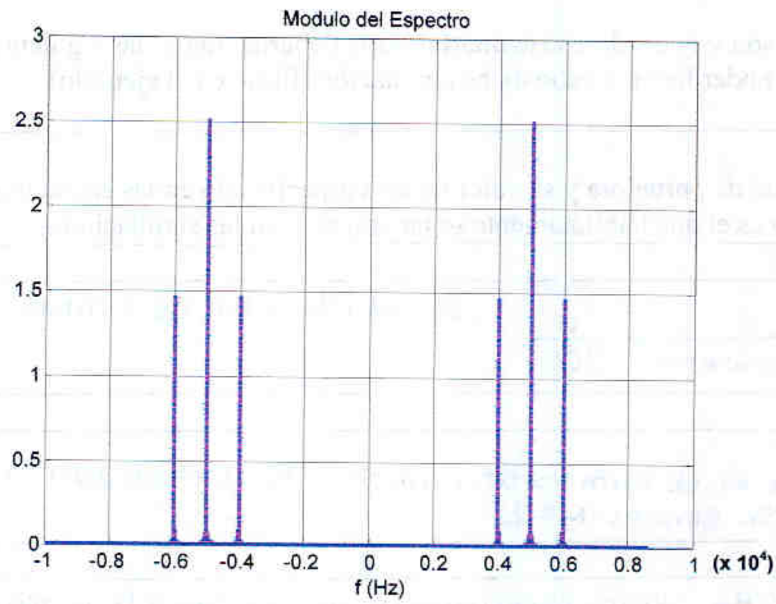
Apellidos y Nombre: ______________________

c) Indique el diagrama de bloques del sistema recuperador para una frecuencia de muestreo genérica $f_s$:




## Cuestión 4

A la salida de un modulador AM se realiza un análisis espectral de la señal con la función `dibuja_tf_bi`, obteniendo el resultado que muestra la gráfica:




Conteste las siguientes preguntas **razonando adecuadamente** la respuesta (es decir, indicando como obtiene el resultado):

a) ¿Cuál es la frecuencia de portadora?

conociendo el espectro de la modulación AM sé que $f_c$ es la frecuencia a la que aparece la delta espectral central
$f_c = 5 kHz$

Apellidos y Nombre:________________________________

b) ¿Cuál es la amplitud de la portadora? (Ac)

$\frac{Ac}{2} = 2'5 \rightarrow Ac = 5\ V$

c) ¿Cuál es la frecuencia de la moduladora?

$f_m = 6kHz - 5kHz = 1kHz$

d) ¿Cuál es el índice de modulación? (Realice todos los cálculos en el recuadro)

$\frac{Ac\,m}{4} = 1'5 \rightarrow m = \frac{4 \times 1'5}{Ac} = \frac{6}{5}$

## Cuestión 5

Sea una simulación PSK binaria definida mediante las siguientes características:

- Velocidad de transmisión: 300 bps = 300 simb/s
- Señal Portadora sinusoidal a 1500 Hz
- Relación señal-a-ruido: 15 dB

Indique qué son y qué valores **razonadamente** deberían tomar las siguientes variables de Matlab para poder llevar a cabo dicha simulación (fíjese en el ejemplo):

>> Ac=1;
Es la amplitud de portadora y su valor no está especificado en las características anteriores. Sin embargo es el que habitualmente se ha utilizado en las simulaciones.

>> D=
$D = \frac{1}{V_{modulacion}} = \frac{1}{300}$ D es el intervalo de símbolo

>> M= 2
ya que es el número de símbolos de la modulación y por tanto en PSK Binaria M=2

>> fc= 1500Hz Viene especificado en las características anteriores

>> EbNodB=

$EbNodB = \left(\frac{S}{W}\right)_{dB} - \underbrace{10\log(2\log_2 M)}_{3\ (\text{si } M=2)} = 12\ dB$

También $\frac{S}{N} = \frac{\frac{Ac^2}{2}}{2\cdot\frac{No}{2}\cdot W} = \frac{\frac{Ac^2}{2}}{2\cdot\frac{No}{2}\cdot\frac{1}{D}} = \frac{\frac{Ac^2}{2}\cdot D}{2\cdot\frac{No}{2}} = \frac{Eb}{No}$

NOTA: $\left(\frac{S}{N}\right)$ en lineal $= \frac{Eb\,[J/bit]}{No\,[W/Hz]} \cdot \frac{R_I\,[bit/s]}{W\,[Hz]}$ $R_I = \log_2 M \cdot R_m = \frac{\log_2 M}{D}$ $W = \frac{1+\alpha}{D} = \frac{1}{D}$

Apellidos y Nombre:__________

$\Rightarrow \frac{Eb}{No} = \frac{(S/N)}{D\frac{\log_2 M}{D}} = \frac{(S/N)}{\log_2 M}$

# Laboratorio de Señales y Sistemas (5º cuatrimestre)

## EXAMEN EXTRAORDINARIO JUNIO 2005

Profesores: José Javier López, Narcís Cardona, Gema Piñero, Miguel Ángel Rodríguez

| C1 | C2 | C3 | C4 | C5 |
|---|---|---|---|---|
| | | | | |

- Responda en el espacio habilitado para cada cuestión. Todas ellas tienen igual puntuación.
- **Tiempo máximo: 1 hora**

### Cuestión 1

Se ha calculado el desarrollo en serie de Fourier de una señal con periodo T=8, resultando el siguiente vector de coeficientes:

$$a_k=[0 , 0 , 0.5 , 0 , 0 , 0 , 0.5 , 0]$$

Escriba las instrucciones necesarias en Matlab para recuperar la señal original:

```
» n = 0:1:7;
>> x = x_DSF(a);
>> stem(n, ~~abs~~(x));
```

mejor real

### Cuestión 2

Se quiere obtener el resultado de la siguiente operación que involucra señales discretas $k=\sum_{n=-\infty}^{\infty}[z(n)\cdot y(n)]$, donde $z(n) = 2\delta(n+3)-\delta(n+2)+3\delta(n+1)+5\delta(n-1)-2\delta(n-3)$ e $y(n)$ es la convolución $y(n)= x(n)*x(n)$, siendo $x(n) = \delta(n+2)-\delta(n+1)-2\delta(n) +3\delta(n-2)$. Para ello se dispone de un programa MATLAB, del que se han borrado partes de algunas instrucciones. Se pide completar las partes borradas de las instrucciones.

**Nota:** en cada línea no puede haber más de una instrucción, no se pueden añadir nuevas líneas, la parte que se mantiene de cada instrucción no se puede modificar.

```
> x= [1,-1,-2,0,3]........................;
> nx = -2:1:2 ..........................;
> z= [2,-1,3,0,5,0,-2]........; % primer y último valor de z son distintos de cero
> [y, ny] = pract_conv(x,nx,x,nx); % primer y último valor de y son distintos de cero
> yseleccion=y(2:end-1.............); ya que al multiplicar por z(n) se irán todos los que no sean [-3, 3]
> zseleccion= ...z.......................;
> pp= zselección.........*..........yseleccion; % producto de z(n) por y(n)
                                   % primer y último valor de pp son distintos de cero
> stem(-3:3,..pp....);% representa la señal pp con sus valores correctos de tiempo
> k= sum(pp).......................................; % obtiene el valor final pedido
```

| nz | -3 | -2 | -1 | 0 | 1 | 2 | 3 |
|---|---|---|---|---|---|---|---|
| z | 2 | -1 | 3 | 0 | 5 | 0 | -2 |

| nx | -2 | -1 | 0 | 1 | 2 |
|---|---|---|---|---|---|
| x | 1 | -1 | -2 | 0 | 3 |

x*x comienza y acaban [-4, 4]

Apellidos y Nombre: ____________________

## Cuestión 3

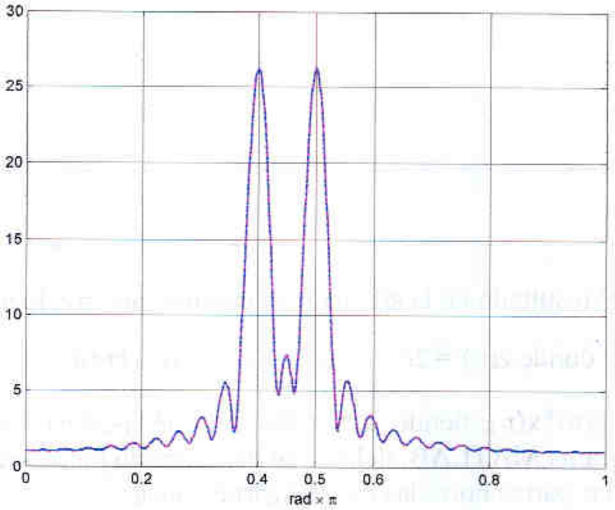
Sea una señal continua $x(t)=\cos(2\pi f_1 t)+\cos(2\pi f_2 t)$, donde solo se conoce el valor de $f_2 = 1.5\ KHz$ y que $f_2 > f_1$. Dicha señal *x(t)* se muestrea con una frecuencia de muestreo $f_s$ desconocida, de la que únicamente se conoce que es mayor que la frecuencia de Nyquist, obteniéndose *x[n]*.

no hay antialiasing

a) Indique los 3 efectos que diferenciarán el espectro de la señal continua original *x(t)* del espectro de la señal discreta muestreada *x[n]*.

- El eje de frecuencias está normalizado a la frecuencia de muestreo ($\Omega = \frac{\omega}{f_s}$) o tambien ($\Omega = \pi \cdot \frac{f}{f_s/2}$)
- El espectro de $x(n)$ se repite $2\pi$-periodicamente.
- El espectro está multiplicado (amplitud) por la frecuencia de muestreo

En la figura se representa el espectro de *x[n]* obtenido mediante la función *dibuja_tf_senal_muestreada*:




A la vista de ella, responda justificadamente:

b) ¿Cuál es la frecuencia de muestreo, $f_s$, utilizada?

La amplitud de las deltas en $x(t)$ será 0'5
La amplitud de las deltas en $x(n)$ será $0'5 * f_s \simeq 25 \rightarrow f_s = 50$ ?? no tiene sentido

si $\Omega = 2\pi \frac{f}{f_s}$ $\quad \pi\, 0'5 = 2\pi \frac{1'5\,kHz}{f_s} \rightarrow f_s = 2 * \frac{1'5\,kHz}{0'5} = 6\,kHz$

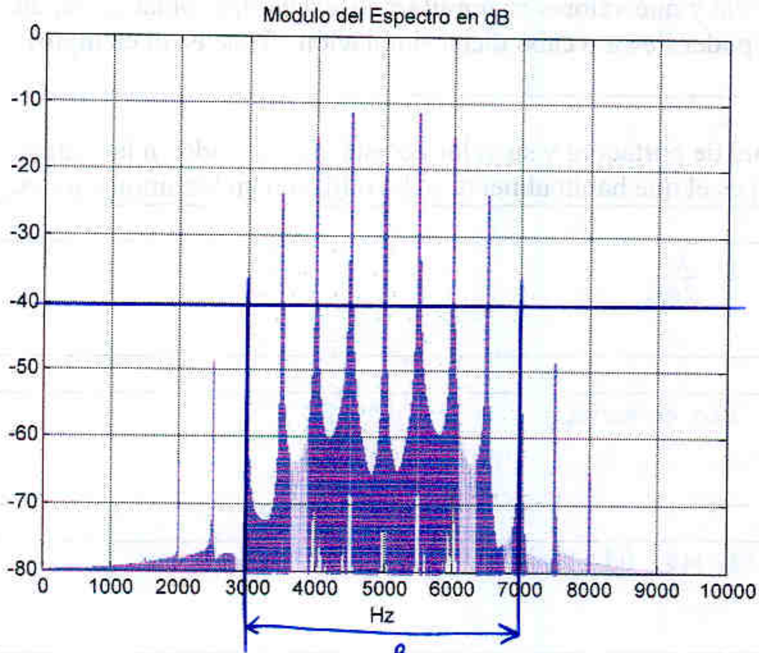
c) ¿Cuál es el valor de la frecuencia $f_1$ en Hertzios?

$\Omega = 2\pi \cdot \frac{f}{f_s} \rightarrow f = \frac{f_s \cdot \Omega}{2\pi} = \frac{6\,kHz \cdot 0'4\pi}{2\pi} = 1'2\,kHz$

Apellidos y Nombre: ____________________

## Cuestión 4

Simulamos con MATLAB y Simulink un modulador de FM. A la salida del modulador hacemos un análisis espectral de la señal con la función `dibuja_tf_bi_dB`, obteniendo el resultado que muestra la gráfica. Conteste las siguientes preguntas razonando adecuadamente la respuesta (es decir, indicando cómo obtiene el resultado).




a) ¿Cuál es la frecuencia de portadora?

El espectro FM es simétrico respecto a $f_c$, por tanto $f_c = 5\text{kHz}$

b) ¿Cuál es la frecuencia de la moduladora? llamar a las deltas: rayas espectrales

Las distintas 'deltas' aparecen en múltiplos de $f_m$ alrededor de $f_c$ (es decir $f_c \pm i f_m$) por tanto $f_m = 500\text{ Hz}$

c) Considerando que el ancho de banda efectivo es el ocupado por las rayas espectrales no menores que 40 dB por debajo de la raya más alta, calcule dicho ancho de banda, indicándolo también en la gráfica.

$B = 7000 - 3000 = 4000\text{ Hz}$

d) A partir de la medida anterior, calcule el índice de modulación FM.

Indice de modulación $\beta = \frac{f_d \cdot A_m}{f_m} =$ según Carson: $B = 2(\beta+1)\cdot f_m$ o para una señal genérica $2(D+1)W$

$f_d$: desviación de frecuencia
$f_m$: frec moduladora
$A_m$: amplitud moduladora

$\hookrightarrow \beta = \frac{B}{2 f_m} - 1 = \frac{4k}{2\cdot 500} - 1 = 3$

Apellidos y Nombre: ______________________

## Cuestión 5

Se desea realizar una simulación PSK binaria en Simulink definida mediante las siguientes características:

- Velocidad de transmisión: 200 bps
- Señal Portadora sinusoidal a 1500 Hz
- Relación señal-a-ruido: 0 dB

Indique qué son y qué valores **razonadamente** deberían tomar las siguientes variables de Matlab para poder llevar a cabo dicha simulación (fíjese en el ejemplo):

>> Ac=1;
Es la amplitud de portadora y su valor no está especificado en las características anteriores. Sin embargo es el que habitualmente se ha utilizado en las simulaciones.

>> D= $\frac{1}{V_{tx}} = \frac{1}{200}$

>> M= 2 (en binaria)

>> fc= 1500 Hz (dato)

>> EbNodB=

Apellidos y Nombre: ______________________________

# Laboratorio de Señales y Sistemas (5º cuatrimestre)

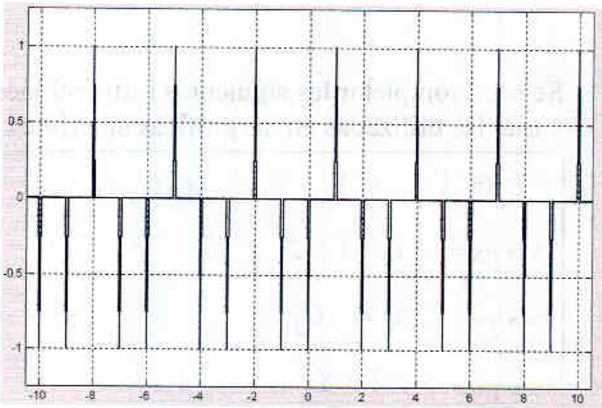
## EXAMEN FINAL ENERO 2004

Profesores: Miguel Ángel Rodríguez, Narcís Cardona, Gema Piñero

- Responda en el espacio habilitado para cada cuestión. Todas ellas tienen igual puntuación.
- **Tiempo máximo: 1 hora**

### Cuestión 1

Escriba las instrucciones necesarias para generar y dibujar en Matlab el tren de impulsos de la figura:



```
>> entorno1
>> x = tren(t-1,3)
       - tren(t,3)
       - tren(t+1,3)
>> dibut(x)
```

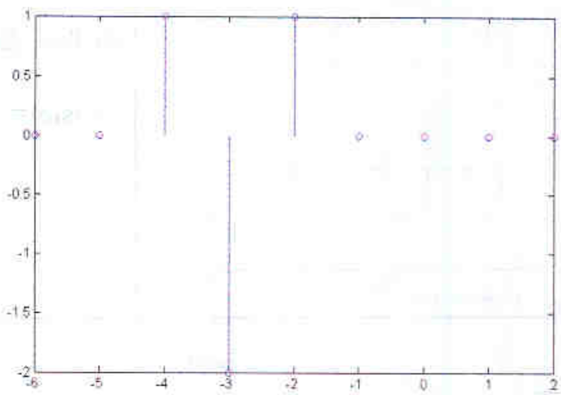
### Cuestión 2

En esta cuestión se va a utilizar la función realizada en la práctica 2:

**function [z, n]= pract_conv(x, nx, y, ny)**

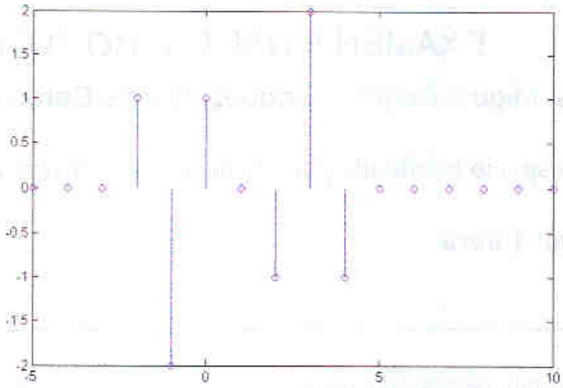
Los datos conocidos son:

- El resultado de ejecutar **stem(nx,x)** es:



Apellidos y Nombre:__________________________________

- El resultado de ejecutar **[z, n]= pract_conv(x, nx, y, ny); stem(n,z)** es:



Se pide completar las siguientes instrucciones MATLAB necesarias para generar las variables utilizadas en las gráficas anteriores:

>> x= [0, 0, 1, -2, 1, 0, 0, 0, 0]

>> nx= -6 : 1 : 2 ;

>> y= [0, 1, 0, 0, 0, -1, 0, 0]      funciona usar la intuición

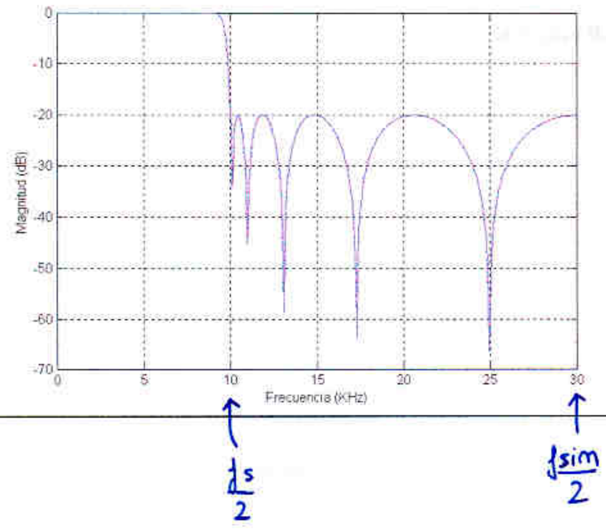
>> ny= 1 : 1 : 8
(-6+1 = 5) (8+2=10)

## Cuestión 3

Se ha ejecutado la función generada en la práctica 3:

**resp_freq_filtro(orden,atenuacion, fs, fsim)**

obteniéndose la respuesta en frecuencia del filtro de reconstrucción. Indique los valores de **atenuacion**, **fs** y **fsim** utilizados para que el resultado sea la siguiente gráfica:




>> atenuacion= 20 (dB)

>> fs= 20 000 (Hz)

>> fsim= 60 000 (Hz)

Apellidos y Nombre:____________________________

Para calcular los coeficientes del filtro anterior se ha utilizado la función

**[B,A]=cheby2(N,R,Wn)**

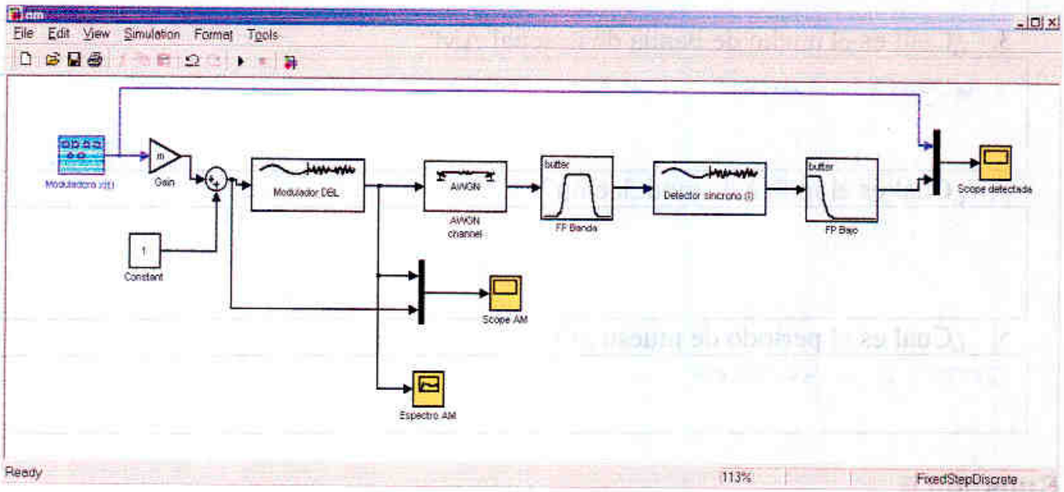
Indique el valor de **Wn** seleccionado para generar la gráfica anterior.

| >> Wn= |
|---|

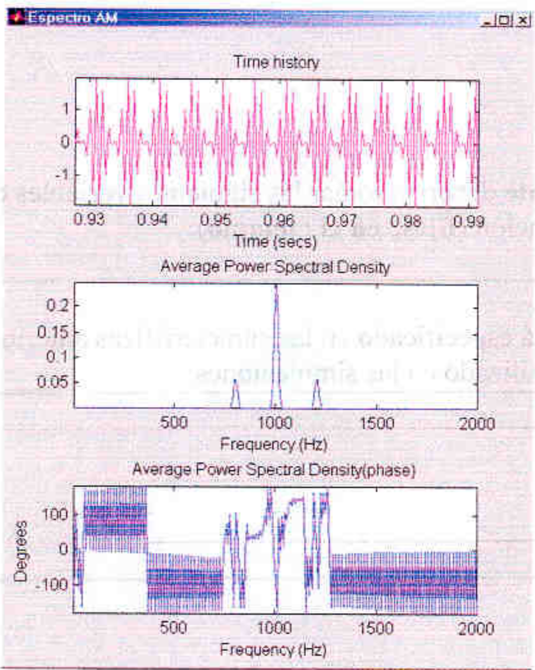
Inicio de la banda atenuada en digital $W_n = \frac{f_s/2}{f_{sim}/2} = \frac{f_s}{f_{sim}} = \frac{20\,000}{60\,000} = \frac{1}{3}$

## Cuestión 4

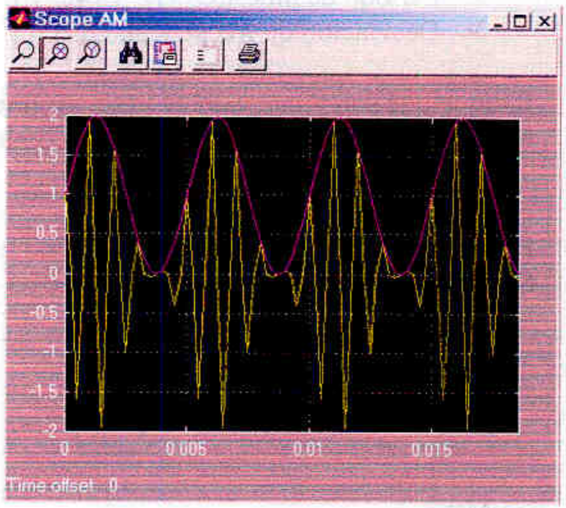
La figura representa el diagrama de bloques de una modulación AM:




Las gráficas del analizador de espectros denominado 'Espectro AM' y del osciloscopio denominado 'Scope AM' vienen dadas por las figuras (a) y (b) respectivamente:




(a)




(b)

Apellidos y Nombre: ___________________________________________

Responda a las siguientes cuestiones **indicando de cuál de las 2 gráficas deduce el resultado y cómo lo deduce**:

1. ¿Cuál es la frecuencia de portadora?

1000 Hz

2. ¿Cuál es la frecuencia de la moduladora?

$\frac{1}{0'005} = 200$ Hz

3. ¿Cuál es el ancho de banda de la señal AM?

400 Hz

4. ¿Cuál es el índice de modulación?

1

5. ¿Cuál es el periodo de muestreo?

2000 · 2 = 4000 Hz

## Cuestión 5

Sea una simulación FSK binaria con las siguientes características:

- Velocidad de transmisión: 200 bps
- Señal Portadora: 1800 Hz
- Relación $E_b/N_0$: 10 dB
- Periodo de muestreo: $T_s=1/(40\cdot v_t)$
- Desviación de frecuencia: 1/(2D)

Indique qué son y qué valores **razonadamente** deberían tomar las siguientes variables de Matlab para poder llevar a cabo dicha simulación (**fíjese en el ejemplo**):

>> Ac=1;
Es la amplitud de portadora y su valor no está especificado en las características anteriores. Sin embargo es el que habitualmente se ha utilizado en las simulaciones.

>> D= $\frac{1}{200}$

>> M= 2

Apellidos y Nombre:

>> fc= 1800 Hz

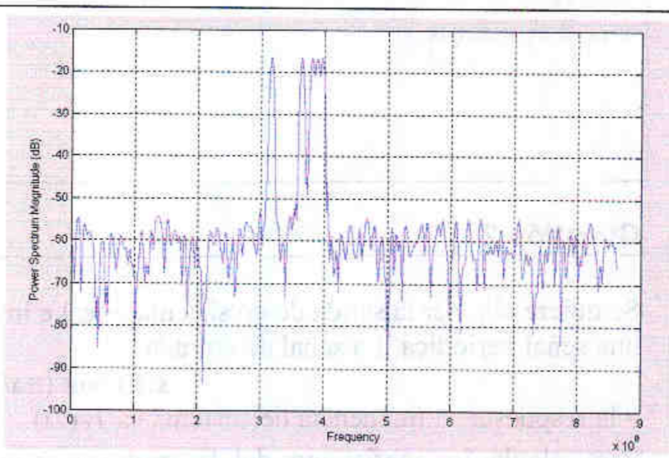
>> fd= desviacion de frecuencia = $\frac{1}{2D}$ (del enunciado)

>> Ts= $\frac{1}{40 \cdot 200}$

>> sigma2=

## Cuestión 6

A la salida de un modelo que simula un conversor optoelectrónico de CATV se ha medido la densidad espectral de potencia de la figura.




A la vista de ese resultado, complete las siguientes afirmaciones:

a. El valor de CSO es mayor que ______ dB

b. La relación señal a ruido es del orden de 40 dB

c. No podemos calcular el nivel de CTB porque ______

Apellidos y Nombre: ______

# Laboratorio de Señales y Sistemas (5º cuatrimestre)

## EXAMEN FINAL JUNIO 2004

Profesores: Miguel Ángel Rodríguez, Narcís Cardona, Gema Piñero

| C1 | C2 | C3 | C4 | C5 | C6 |
|---|---|---|---|---|---|
| | | | | | |

- Responda en el espacio habilitado para cada cuestión. Todas ellas tienen igual puntuación.
- **Tiempo máximo: 1 hora**

## Cuestión 1

Considere un sistema cuya respuesta impulsional se describe como

**sin(pi*t).*(rampa(-t+6,6))**

Escriba las instrucciones necesarias para calcular y dibujar la respuesta al escalón de dicho sistema.

```
>> entorno1
>> h = sin(pi*t).*(rampa(-t+6,6));
>> x = escalon(t);
>> y = convol(x,h);
>> dibut(y)
```

## Cuestión 2

Se quiere obtener la salida de un sistema lineal e invariante cuando a la entrada se presenta una señal periódica. La señal de entrada es:

$$x(n)=\sin(\pi n/2) = \sin\left(2\pi \cdot \frac{1}{4} n\right)$$

→ periodo 4

y la respuesta en frecuencia del sistema es $H(\Omega)$.

Para calcular los coeficientes del desarrollo en serie de Fourier se utiliza la función realizada en la practica 2:

**function a=coef_DSF(x)**

$n=0$ –
$x(0)=\sin(0)=0$
$x(1)=\sin(\pi/2)=1$
$x(2)=\sin(\pi)=0$
$x(3)=\sin(3\pi/2)=-1$

a) Complete la instrucción MATLAB que asigna directamente los valores del vector **x** para este caso:

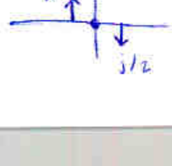
```
>>x=[0, 1, 0, -1]
```

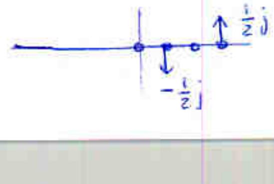
b) Escriba el resultado que aparecerá en pantalla al teclear **coef_DSF(x)**

```
a= [0,-0'5i, 0, 0'5i]
```

$$\sin\left(\frac{\pi n}{2}\right) = \frac{e^{j\frac{\pi n}{2}} - e^{-j\frac{\pi n}{2}}}{2j} = -j/2\, e^{j\frac{\pi n}{2}} + j/2 \cdot e^{-j\frac{\pi n}{2}}$$

Apellidos y Nombre:________________



c) Escriba la expresión teórica de la salida y(n) en función de los coeficientes del DSF de x(n) y de $H(\Omega)$:

$$y(n) = \sum_{k=0}^{N-1} H\left(\frac{2\pi}{N}k\right) a_k \, e^{j\frac{2\pi}{N}kn}$$

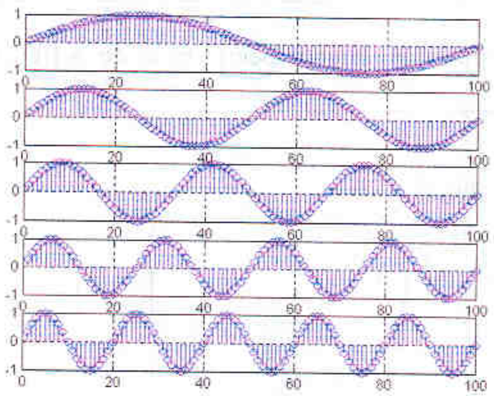
## Cuestión 3

Se han ejecutado las siguientes instrucciones

```
>> n=1:100;
>> subplot(511);stem(sin(2*f1*pi*n/10000));grid
>> subplot(512);stem(sin(2*f2*pi*n/10000));grid
>> subplot(513);stem(sin(2*f3*pi*n/10000));grid
>> subplot(514);stem(sin(2*f4*pi*n/10000));grid
>> subplot(515);stem(sin(2*f5*pi*n/10000));grid
```

resultando la siguiente gráfica. Se pide calcular **f1**, **f2**, **f3** **f4** y **f5**, sabiendo que no se ha producido solapamiento:

sabiendo $\sin\left(2\pi \cdot \frac{m}{N} n\right)$ periodo



f1= 100 Hz   N=100

f2= 200 Hz

f3= 300 Hz

f4= 400 Hz

f5= 500 Hz

Se ejecutan a continuación las siguientes instrucciones

```
>> n=1:100;
>> subplot(511);stem(sin(9500*pi*n/fs));grid
>> subplot(512);stem(sin(9600*pi*n/fs));grid
>> subplot(513);stem(sin(9700*pi*n/fs));grid
>> subplot(514);stem(sin(9600*pi*n/fs));grid
>> subplot(515);stem(sin(9900*pi*n/fs));grid
```

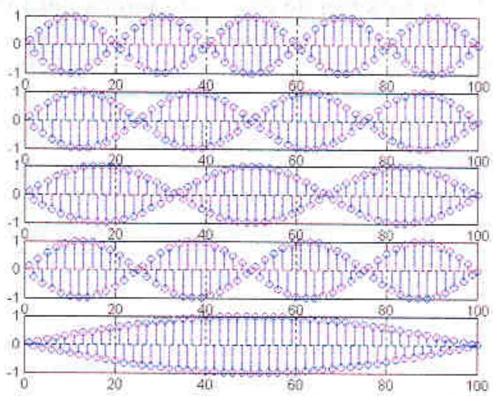
$\sin(2\pi \cdot 4750 \cdot n/f_s)$

$\frac{4750}{f_s} = \frac{m_1}{40}$

$\frac{4800}{f_s} = \frac{m_2}{26}$

resultando la siguiente gráfica. Se pide calcular fs, sabiendo que no se ha producido solapamiento:

Apellidos y Nombre:______________________________

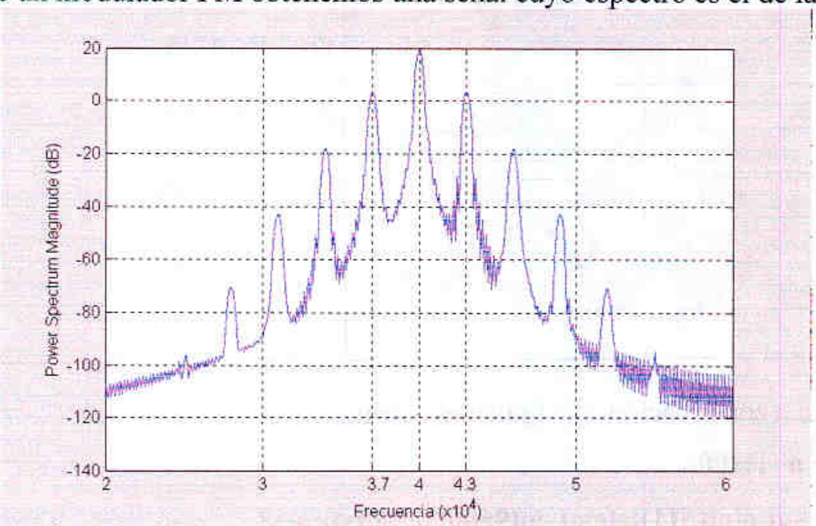
| | |
|---|---|
|  | fs= |

Indique la frecuencia de un seno que al ser muestreado con una frecuencia de muestreo de 2KHz produzca una secuencia no periódica. Nota: la frecuencia seleccionada debe ser menor que 1KHz:

frecuencia= $\frac{1}{2\pi}$ kHz

## Cuestión 4

A la salida de un modulador FM obtenemos una señal cuyo espectro es el de la figura:




Conteste **razonadamente** a las siguientes preguntas:

a) ¿Cuál es la frecuencia de portadora?

40 kHz

b) ¿Cuál es la frecuencia de la señal moduladora?

3 kHz

Apellidos y Nombre: ____________________

c) Sabiendo que la desviación de frecuencia del modulador es $f_d = 1KHz$, ¿Cuál es el ancho de banda de esta señal según Carson?

~~2(f+fd~~ ~~2(...)fm~~ $B = 2(\beta+1)\cdot f_m$ $\beta = \frac{f_d \cdot A_m}{f_m} = \frac{f_d}{f_m} = \frac{1kHz}{3kHz}$

$= 8000\ Hz$

d) ¿Cómo clasificaría esta señal FM, de banda estrecha o de banda ancha?

De banda estrecha : $\frac{f_c}{B} = 5 >> 1$

## Cuestión 5

Sea una simulación PSK binaria con las siguientes características:

- Velocidad de transmisión: 300 bps
- Señal Portadora: 1500 Hz
- Relación $E_b/N_0$: 15 dB
- Periodo de muestreo: $T_s=1/(30\cdot v_t)$

Indique qué son y qué valores **razonadamente** deberían tomar las siguientes variables de Matlab para poder llevar a cabo dicha simulación (fíjese en el ejemplo):

>> Ac=1;
Es la amplitud de portadora y su valor no está especificado en las características anteriores. Sin embargo es el que habitualmente se ha utilizado en las simulaciones.

>> D= $\frac{1}{300}$ s

>> M= 2

>> fc= 1500 Hz

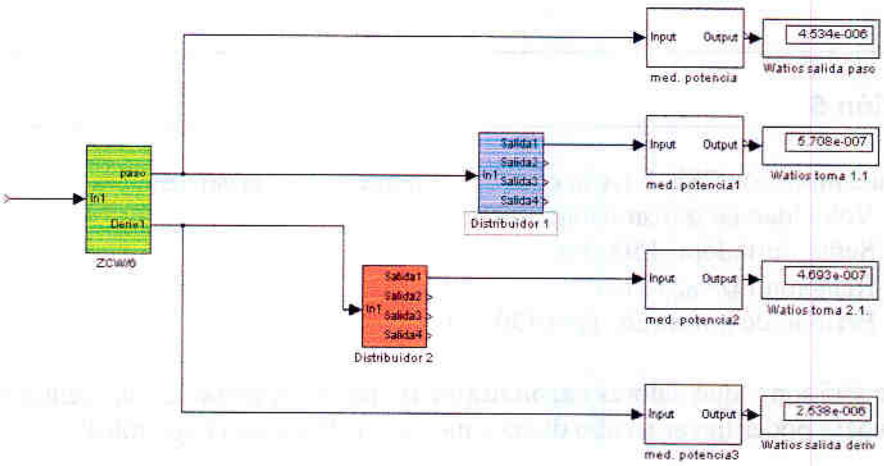
>> Ts= $\frac{1}{30\cdot v_t} = \frac{1}{30\cdot 300} = \frac{1}{9000}$ s

>> sigma2=

Apellidos y Nombre: ____________________

## Cuestión 6

Se está diseñando la parte de distribución en el interior de edificio de una red CATV. Los bloques de la figura corresponden a un derivador y dos distribuidores de distinto modelo que se emplean en el simulador del sistema. Se miden las potencias de señal a la salida de cada uno de dichos elementos como indica la figura, siendo el nivel a la entrada del derivador de 1e-5 Watios.




Tras los distribuidores se conectan los usuarios con tramos de cable y conectores iguales para todos ellos ¿Cuál de las siguientes afirmaciones no es cierta y por qué ?

a. El factor de ruido del distribuidor 1 es mayor que el del distribuidor 2.

b. El valor de CSO en dB medido a la salida del distribuidor 2 es mayor que a la salida del distribuidor 1.

Apellidos y Nombre: ______________________________

050

# LABORATORIO DE SEÑALES Y SISTEMAS

## PRACTICA 1

### CURSO 2005-2006

# PRÁCTICA 1

# SEÑALES Y SISTEMAS CONTINUOS

Las presente practica trata distintos aspectos de las señales y los sistemas en tiempo continuo. Los diferentes apartados estudian las señales continuas, la convolución, la correlación, el desarrollo en serie de Fourier y la transformada de Fourier.

La práctica se desarrollo sobre MATLAB. MATLAB no permite el manejo directo de señales continuas por lo que es necesario la utilización de 2 simuladores para realizar esta práctica. El primer simulador, *entorno1*, se utiliza para estudiar las señales, la convolución y la correlación, mientras que el segundo simulador, *entorno2*, se utiliza para el análisis de Fourier.

Por último indicar que aunque esta práctica se presenta en un nuevo formato en el curso 2003/2004, utiliza el material preparado por Vicenç Almenar Terré y Pablo Bernabeu Soler presentado en la publicación "Prácticas de Sistemas Lineales, Teoría de la Comunicación y Laboratorio de Señales y Sistemas" SPUPV-98.1205.

Miguel Angel Rodríguez Hernández

## 1. SEÑALES CONTINUAS

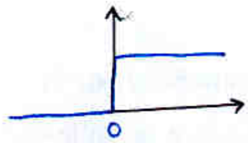
Para realizar los primeros apartados de la práctica se utiliza un simulador de señales y sistemas continuos sobre MATLAB. Para iniciar este simulador es necesario teclear *entorno1*. Teclee *entorno1*.

### 1.1. Señales básicas

En esta sección vamos a ver cómo trabajar con las funciones que permiten generar las señales básicas. También veremos cómo visualizar dichas señales, para realizar esto último utilizaremos la función dibut(). A continuación se presentan y se visualizan las distintas señales básicas:

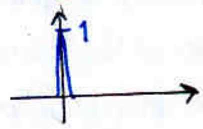
- **Escalón:**

  dibut(escalon(t))



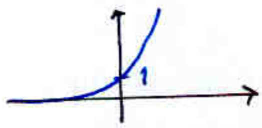

- **Delta:**

  dibut(delta(t))



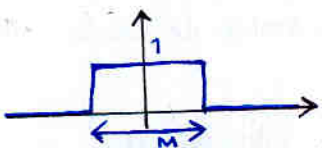

- **Exponencial:**

  dibut(exp(t))



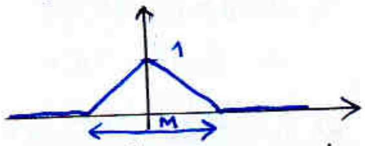

- **Pulso:** define un pulso de amplitud (altura) 1 y anchura $M$ centrado en $t = 0$, es decir, empieza en $-M/2$ y llega hasta $M/2$. En el siguiente ejemplo $M = 4$:

  dibut(pulso(t,4))



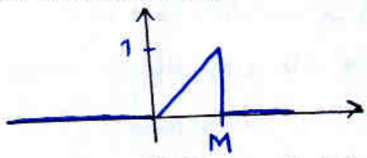

- **Triángulo:** define un triángulo de altura 1 y anchura $M$ centrado en $t = 0$. En el ejemplo $M = 4$:

  dibut(trian(t,4))




- **Rampa:** define una rampa creciente que empieza con amplitud cero en $t = 0$, llega a $t = M$, donde alcanza su amplitud máxima uno. En el ejemplo $M = 4$:

  dibut(rampa(t,4))




Para cambiar la amplitud de una señal basta con multiplicar o dividir dicha función, veamos un par de ejemplos:

dibut(2*pulso(t,4))

dibut(trian(t,4)/2)

Otra manera de trabajar y visualizar señales consiste en almacenarlas previamente en una variable para usar posteriormente dicha variable, en este caso vamos a almacenar la función escalón en la variable u, fíjese que es importante poner el ";" ya que si no lo hace en pantalla aparecerán todos los valores que contiene el vector u:

u = escalon(t);

dibut(u)

Fíjese en que u es una variable, y no una función, por lo que si al utilizarla escribe: dibut(u(t))

### 1.2. Transformaciones de la variable independiente

Vamos a ver cómo se aplican las transformaciones de la variable independiente a las señales generadas con las funciones del apartado anterior.

- **Inversión:** basta con poner un signo menos antes de la t dentro de la función.

  dibut(escalon(-t))

  dibut(exp(-t))

  dibut(rampa(-t,4))

- **Cambio de escala:** aquí se ve la señal original, la comprimida y la expandida.

  dibut(pulso(t,4))

  dibut(pulso(2*t,4))

  dibut(pulso(t/2,4))

En los últimos dos casos observará que el ancho del pulso no es 4, sino que en un caso se trata de un pulso comprimido en el tiempo y vale 2, y en el otro de un pulso expandido y vale 8.

- **Desplazamiento:** retardo y avance:
  - Retardo

    dibut(escalon(t-4))
  - Avance

dibut(escalon(t+4))

En el caso de haber inversión el desplazamiento queda (fíjese en los signos):

- Retardo

  dibut(escalon(-t+4))

- Avance

  dibut(escalon(-t-4))

- **Cambio de escala y desplazamiento:** a la hora de efectuar estas dos operaciones combinadas al utilizar MATLAB basta con escribir directamente la función. Supongamos que deseamos realizar la operación $x(at+b)$, señal que corresponde un desplazamiento de valor $b$, y a un cambio de escala de factor $a$. Por ejemplo:

  dibut(pulso(2*t-2,4))

### 1.3. Simetría par e impar

Genere y almacene en una variable, llámela x, la señal de la Figura 1. Dicha señal está formada por una rampa que va desde $t=-4$ hasta $t=0$, y un pulso que empieza en $t=0$ y llega hasta $t=4$. A continuación, genere y almacene en otra variable, llámela xn, esa misma señal invertida en el tiempo, obviamente no podrá hacer:

xn = x(-t);

pues x no es una función y MATLAB se quejará. Deberá generarla manualmente.

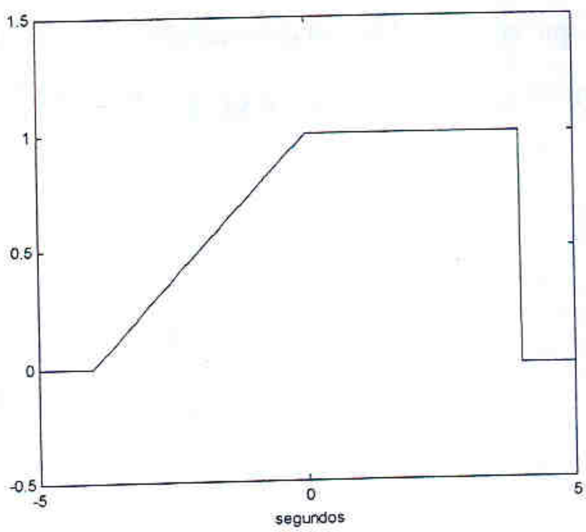



**Figura 1**

Las ecuaciones que permiten calcular la parte par e impar de una señal son:

$$x_p(t) = \frac{1}{2}\left(x(t) + x(-t)\right) \qquad x_i(t) = \frac{1}{2}\left(x(t) - x(-t)\right)$$

utilícelas para calcular la parte par e impar de la señal arriba definida. Observe que dichas ecuaciones están escritas en notación matemática, por lo que al pasarlas a MATLAB deberá "traducirlas" correctamente, por ejemplo la primera podría quedar:

```
xp = (x+xn)/2;
```

Recuerde que para comprobar si se han realizado bien los cálculos la suma de la parte par y de la impar vuelven a dar la señal original.

### 1.4. Limitación de la duración

Para limitar en el tiempo la existencia de una señal se la debe multiplicar por un escalón. Por ejemplo, si queremos limitar una exponencial decreciente para que empiece en cero habría que hacer lo siguiente:

```
dibut(escalon(t).*exp(-t)),grid
```

Observe la instrucción anterior, en ella se ha realizado el producto entre dos señales, para ello hay que utilizar el operador ".*", mientras que cuando hacemos productos por constantes el operador utilizado es "*". Es importante recordar esta diferencia, ya que si no se hace así MATLAB dará error.

La instrucción grid introduce una rejilla que le puede servir de referencia, esto puede hacerlo siempre que lo considere necesario.

Obtenga ahora la misma exponencial recortada pero desplazada a $t = -2$ (en la Figura 2 tiene el resultado, fíjese que en ese instante la exponencial vale 1 en amplitud).

dibut (escalon (t+ 2).* exp (-t - 2))

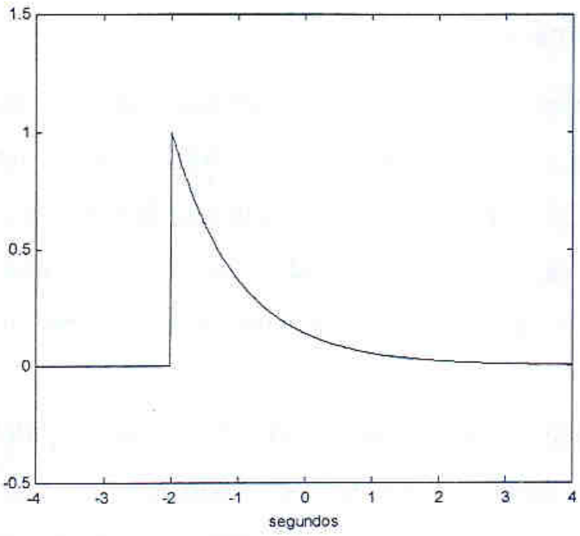



**Figura 2**

## 2. LA CONVOLUCION

En las siguientes líneas veremos algunas características de la convolución de dos señales que pueden sernos útiles para saber los límites temporales en los que encontraremos el resultado de una convolución, para ello usaremos la convolución de dos pulsos.

- **La anchura** de la señal resultante es la suma de la anchura de las dos señales originales, para comprobarlo teclee:

```
pu = pulso(t,2);
dibut(convol(pu,pu));
```

Puede comprobar que el resultado es un triángulo con el doble de anchura que el pulso. En este ejemplo ha utilizado una variable para guardar resultados intermedios, trabajar de esta manera permite reutilizar dichas señales.

Otro caso para comprobar lo anterior es

```
ti = trian(t,4);
dibut(convol(ti,pu))
```

- **Los instantes inicial y final** de la señal resultante se calculan, respectivamente, sumando los instantes inicial y final de las señales originales:

```
pu2 = pulso(t-1,2);
```

dibut(convol(pu2,pu2))

dibut(convol(pu,pu2))

1. Para el primer caso, tenemos la convolución de dos pulsos que empiezan en el instante $t = 0$, y tienen anchura 2. Por lo tanto el instante de inicio será $0 + 0 = 0$, y como la anchura de la convolución es la suma de ambas, la señal resultante tendrá anchura 4. El instante final de la convolución será la suma del instante inicial con el ancho $0 + 4 = 4$, que coincide con la suma de los instantes finales.

2. Para el segundo caso, tenemos la convolución de un pulso que empieza en $t = -1$ con otro que empieza en $t = 0$, en consecuencia la señal convolución se iniciará en el instante $-1 + 0 = -1$, y como la anchura de la convolución es 4, el instante final será $t = 3$.

### 2.1 Convolución con un tren de impulsos.

Sea $h(t)$ una señal triangular de anchura 2 centrada en cero, y $x(t)$ un tren de impulsos:

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

La función tren(t,T) permite generar un tren de impulsos de periodo $T$. Calcule:

$$y(t) = x(t) * h(t)$$

con $T = 4$, $T = 2$, $T = 1.5$, y $T = 1$.

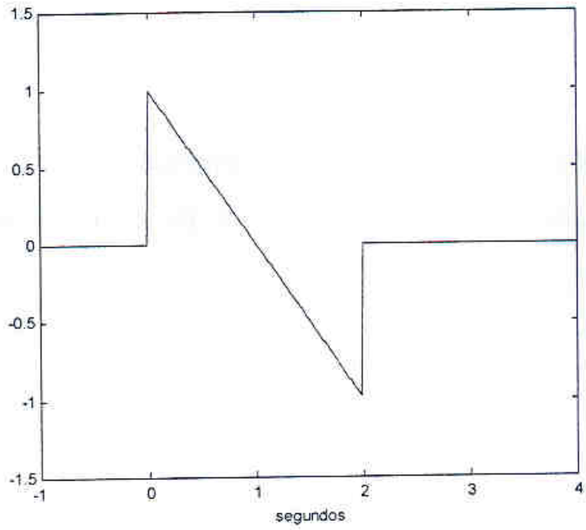
### 2.2 Convolución con un tren de impulsos de signo alterno.

Calcule ahora $y(t)$, sabiendo que la respuesta al impulso $h(t)$ es la que aparece representada en la Figura 3 y $x(t)$ es:

$$x(t) = \sum_{k=0}^{\infty} (-1)^k \delta(t - k)$$

observe que el sumatorio empieza en $k = 0$. Para generar la señal $x(t)$ es aconsejable dibujarla sobre el papel y descomponerla en partes sencillas, haga uso de la función tren anterior.

tren1 = tren(t,1)
tren2 = tren(t-1,2)
x1 = (tren1) - 2(tren2)
x = x1 · encalon(t)




h = rampa (-t+1,1)
− rampa (t-1, 1)

**Figura 3**

El resultado que debe obtener es el siguiente:

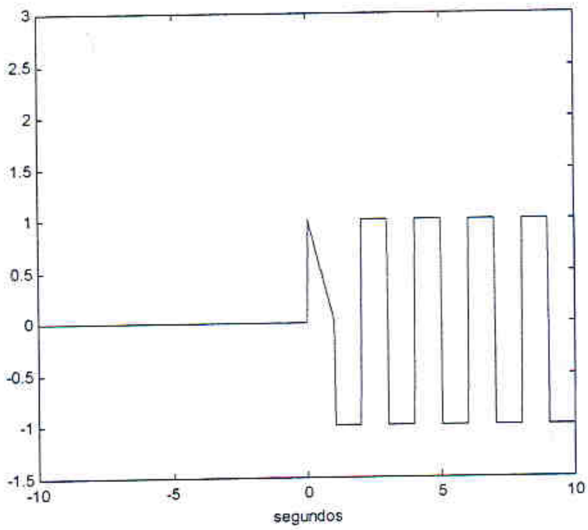



**Figura 4**

## 3. LA CORRELACION

En este apartado vamos a comprobar la utilidad de la correlación en la transmisión digital de señales. Para ello se han definido una serie de funciones que permiten simular la transmisión de señales binarias. Antes de nada ejecute la siguientes instrucciones para ajustar la variable de visualización:

```
clear
entorno1
ejet = axc;
```

En la transmisión de señales binarias se utilizan símbolos más complejos que el simple nivel bajo o alto. En este caso se han elegido los que aparecen en la siguiente figura:

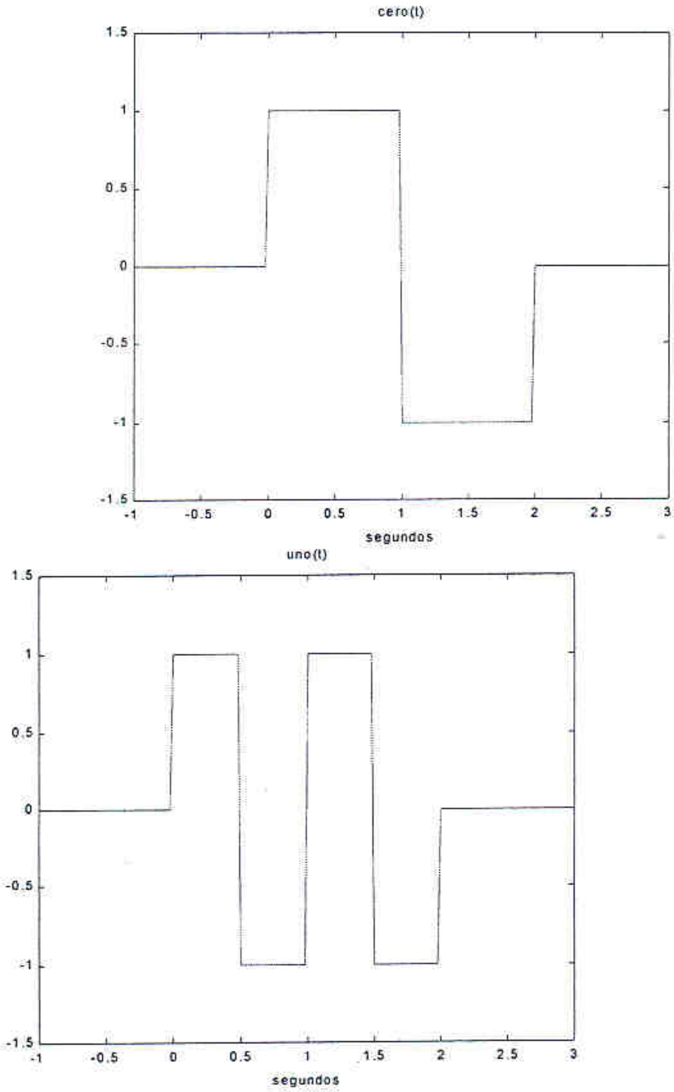



**Figura 5**

Defina dichos símbolos utilizando las funciones: cero(t) y uno(t).

En recepción, al llegar un símbolo, para poder distinguir de cuál de los dos se trata, se pasa la señal recibida por dos filtros, cada uno de ellos "adaptado" a uno de los símbolos. La respuesta impulsional de un filtro adaptado a un símbolo es $h(t) = s^*(-t)$, donde $s(t)$ es la señal que representa al símbolo, como en nuestro caso tratamos con señales reales los filtros adaptados serán:

```
h0 = cero(-t);
dibut(h0)
h1 = uno(-t);
dibut(h1)
```

donde puede observarse que un filtro adaptado ideal es no causal.

La salida de dichos filtros será igual a la correlación cruzada del símbolo recibido con las respuestas impulsionales asociadas a cada posible símbolo recibido. Para ver que el resultado de dichas correlaciones es diferente en función del símbolo de que se trate:

```
x00 = convol(cero(t),h0);
x01 = convol(cero(t),h1);
dibut([x00 x01])
```

La figura amarilla representa la correlación cruzada entre el símbolo cero y su filtro adaptado, resultado que coincide con la autocorrelación del cero; la violeta, en cambio, representa la correlación cruzada entre el uno y el cero. Realice lo mismo para el uno, y de nuevo verá que el filtro adaptado da el máximo cuando a la entrada tiene su símbolo correspondiente.

Ahora vamos a ver cómo se ha simulado un canal de transmisión. Para ello se han desarrollado dos funciones trxs.m y trxn.m, las dos introducen un retardo en la señal transmitida, pero la segunda además le añade ruido. Para comprobar el retardo del canal hacemos lo siguiente:

```
men = cero(t);
rs = trxs(men);
rn = trxn(men);
dibut(men)
dibut(rs)
dibut(rn)
dibut(convol(rs,h0))
dibut(convol(rn,h0))
```

Si lo que queremos generar es un mensaje, transmitirlo, y observar a la salida qué es lo que hemos transmitido:

```
men = mensaje(t,[0 1 1 0]);
```

```
rn = trxn(men);
dibut(men)
dibut(rn)
rn0 = convol(rn,h0);
rn1 = convol(rn,h1);
dibut([rn0 rn1])
```

Viendo la última figura y sabiendo que los picos amarillos son ceros y los violetas son unos se puede distinguir la información transmitida, lo que no era posible observando la señal original.

## 4. DESARROLLO EN SERIE DE FOURIER DE SEÑALES PERIÓDICAS.

Para realizar los siguientes apartados se utiliza un nuevo simulador. Para iniciar este nuevo simulador es necesario teclear *entorno2*. Teclee *entorno2*.

### 4.1 Señales sinusoidales

Genere y visualice un coseno de periodo 2, para ello ejecute las siguientes instrucciones:

```
c1 = cos(2*pi*t/2);
dibut(c1)
```

El espectro de señales periódicas se calcula mediante la función especsf(), ésta devuelve como resultado un vector complejo, para poder visualizar dicha información espectral dispone de las funciones abs() para calcular la magnitud y angle() para la fase. Por ejemplo, para visualizar la magnitud del espectro del coseno anterior:

```
mec1 = abs(especsf(c1));
dibuf(mec1)
```

En la figura puede ver que la magnitud de su espectro consta de dos deltas de altura 1/2 centradas en $f = \pm 0.5$ Hz.

Fíjese que para representar los espectros de las señales utilizaremos la función dibuf(), mientras que para visualizar las señales en el tiempo usaremos dibut(). La diferencia entre ambas es que usan diferentes escalados en la visualización, y etiquetan en sus respectivas unidades el eje horizontal.

### 4.2. Suma de sinusoides

Genere dos señales coseno c2 y c3 de periodos $T_2 = 1$ y $T_3 = 0{,}5$, y amplitudes 1,5 y 2 respectivamente. Visualícelas para comprobar que tienen el periodo y la amplitud deseados. Obtenga la señal suma de los tres cosenos c1, c2 y c3 generados hasta este momento.

Calcule la magnitud del espectro de la señal suma y visualícela. Deben aparecer deltas en 0,5 Hz, 1 Hz y 2 Hz con alturas distintas en función de la amplitud de los cosenos.

Para comprobar la propiedad de linealidad en la transformada de Fourier, calcule los espectros de cada uno de los cosenos por separado, súmelos y comprobará al representarlos que obtiene el mismo resultado que en la imagen anterior.

### 4.3. El fenómeno de Gibbs

Recuerde que el fenómeno de Gibbs aparece cuando se trabaja con el desarrollo en series de Fourier de señales con discontinuidades y se toma un número finito de coeficientes. Para comprobarlo vamos a trabajar con las series de Fourier de una señal cuadrada y de una señal triangular y veremos qué sucede cuando tomamos sólo los primeros armónicos. Sea la señal cuadrada s1 que tiene el siguiente desarrollo en serie de Fourier:

$$x(t) = \frac{1}{2} + \frac{2}{\pi}\cos(2\pi t/2) - \frac{2}{3\pi}\cos(2\pi 3t/2) + \frac{2}{5\pi}\cos(2\pi 5t/2)\cdots$$

En MATLAB podemos representar la señal s1, los tres primeros armónicos y la componente continua como:

```
s1=square(2*pi*t/2+pi/2)/2 + 0.5;
x0 = .5;
x1 = 2*cos(2*pi*t/2)/pi;
x3 = -2*cos(2*pi*t*3/2)/3/pi;
dibut([s1 x0+x1+x3])
```

En la pantalla se ven superpuestas la señal cuadrada y la aproximación que utiliza los tres primeros armónicos (recuerde que los armónicos pares son cero) y la continua. Podemos ver que en las discontinuidades aparece el rizado conocido como fenómeno de Gibbs. Si queremos visualizar el error cometido:

```
dibut(s1-(x0+x1+x3))
```

Para calcular el error en un periodo debemos realizar las siguientes operaciones:

```
err = (s1-(x0+x1+x3)) .* pulso(t,2);
dibut(err)
```

Ahora vamos a comprobar que el rizado sigue apareciendo al aumentar el número de armónicos, aunque el fenómeno de Gibbs no desaparece el error va disminuyendo y se va concentrando en las discontinuidades. Para ello, defina el armónico cinco y compare de nuevo la señal original y la suma de armónicos, calcule también el nuevo error.

A continuación veremos que cuando la señal no tiene discontinuidades y se trunca su DSF, no aparece el rizado característico del fenómeno de Gibbs. Trabajaremos para ello con una señal triangular, para generarla usamos la función de MATLAB sawtooth(). Dicha función genera un diente de sierra entre ±1:

```
dibut(sawtooth(2*pi*t/2))
```

para poder generar una señal triangular hay que añadir un parámetro:

```
dibut(sawtooth(2*pi*t/2,.5))
```

Si queremos una señal triángular similar a la señal cuadrada haremos lo siguiente:

```
s2 = sawtooth(2*pi*t/2+pi,.5)/2+.5;
dibut(s2)
```

Su desarrollo en series de Fourier es:

$$x(t) = \frac{1}{2} + \frac{4}{\pi^2}\cos(2\pi t/2) + \frac{4}{9\pi^2}\cos(2\pi 3t/2) + \frac{4}{25\pi^2}\cos(2\pi 5t/2)\cdots$$

Calcule los armónicos y compárelos con s2, comprobará que no hay rizado.

## 5. TRANSFORMADA DE FOURIER DE SEÑALES APERIODICAS

### 5.1. Dualidad tiempo-frecuencia

En este apartado vamos a comprobar la dualidad entre el dominio espectral y el temporal. Para ello primeramente definimos una señal *sinc* en el tiempo y calculamos su espectro, que para el caso de señales aperiódicas es la función espectf() (fíjese en que la terminación tf hace referencia a la transformada de Fourier, mientras que en las señales periódicas era sf haciendo referencia a las series de Fourier):

```
x = 2*sinc(2*t);
```

```
ex = espectf(x);
```

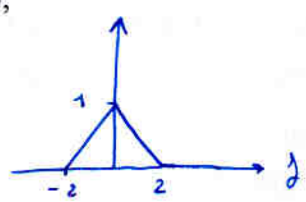
Viendo cómo hemos definido el *sinc* sabemos que en la frecuencia le corresponde un pulso de anchura 2 y altura 1. Veamos dicha señal en ambos dominios:

```
dibut(x)
dibuf(real(ex))
```

en el espectro observamos que no hemos obtenido exactamente un pulso pues ha aparecido un rizado cerca de las discontinuidades. Esto es debido al fenómeno de Gibbs que aparece por la manera en que se está trabajando: recuerde que MATLAB debe utilizar vectores de longitud finita, por lo que el vector que contiene el *sinc* sólo contiene una parte del mismo, es decir, se ha truncado las colas. En realidad cuando usamos la variable tiempo t para realizar los cálculos sólo trabajamos entre -20 y 20 segundos, mientras que la visualización la estamos haciendo entre -4 y 4.

A continuación calcularemos el cuadrado de un *sinc* y observaremos su espectro, ya sabemos que deberíamos obtener un triángulo en la frecuencia. En este ejemplo la anchura del triángulo será 4 y su altura 1 (fíjese en los parámetros del *sinc*):

```
x2 = 2*(sinc(2*t).^2);
ex2 = espectf(x2);
dibut(x2)
dibuf(real(ex2))
```




Aquí no aparece el fenómeno de Gibbs puesto que no hay discontinuidades. En ambos casos trabajamos con la parte real porque la señal en el tiempo es real y par.

Para comprobar la dualidad defina un pulso de anchura 2 y compruebe que tiene como par transformado un *sinc* de altura 2 y cuyo primer cero corta el eje $x$ en $f = 1/2$. Haga lo mismo para un triángulo de anchura 4 y observe si la altura y el primer cero del espectro son los esperados. Recuerde que puede utilizar la función grid para ver mejor los puntos de corte.

### 5.2. Diferenciación e integración.

En teoría se vio que la diferenciación realzaba las frecuencias altas y la integración las bajas. Vamos a comprobar esta propiedad utilizando una función coseno alzado, la cual se describe matemáticamente como:

$$x(t) = A\left(1 + \cos\left(\frac{\pi t}{\tau}\right)\right) rect\left(\frac{t}{2\tau}\right)$$

Defina un vector en MATLAB que contenga un coseno alzado con A=0.5, y $\tau = 2$. A continuación calcule y dibuje su derivada y su integral utilizando las funciones deriva() e integra().

Ahora ya puede comparar las tres señales en el tiempo, verá que la derivada es la que tiene más variaciones (señal de frecuencia más alta), mientras que la integral es la que varía más lentamente (señal de frecuencia más baja).

Compare esto mismo en la frecuencia: calcule la magnitud de los espectros de las tres señales anteriores. Al dibujarlas verá que las tres tienen espectros de distintos tamaños, para que le sea más sencillo realizar la comparación entre las tres debería visualizar la magnitud de cada espectro dividido por su máximo (la función de MATLAB max() le permite calcula el máximo de un vector), de esta manera, las tres tendrán como máximo la unidad. Una vez hecho esto podrá comprobar que el espectro de la derivada es el que tiene amplitudes mayores a frecuencias altas, mientras que el de la integral tiene toda la energía concentrada en las frecuencias bajas.

### 5.3. Modulación.

Defina dos señales coseno, una de frecuencia 2Hz y otra de periodo 5s. Visualice ambas señales y compruebe que tienen el periodo deseado. Calcule el producto de ambas señales y visualícelo también.

Para comprobar el efecto de la modulación en el dominio de la frecuencia calcule la transformada de Fourier de las tres señales utilizando la función especsf() que calcula el espectro de una señal periódica. Utilizando la magnitud compruebe que la señal modulada está compuesta de cuatro deltas de altura 1/4 tal y como era de esperar.

A continuación defina un pulso de anchura 4 y un coseno de frecuencia 2Hz, calcule el producto de ambas señales y visualícelo. Calcule su transformada de Fourier usando la función espectf() (se trata de una señal aperiódica debido al pulso), dibuje la magnitud de este resultado, debería haber obtenido dos sinc (en valor absoluto) de altura 2 y anchura 1/2 centrados en ±2Hz.

### 5.4. Retardo e inversión

Defina un pulso de anchura 3 y utilice la función espectf() para calcular su transformada de Fourier. A continuación haga lo mismo para un pulso desplazado a $t_0 = 1$. El primer pulso es una señal par y real, por lo que su espectro será par y real, es decir, tendrá parte imaginaria cero. Para comprobarlo haga lo siguiente:

```
ejef=axf2;
dibuf(real(ep))
dibuf(imag(ep))
```

siendo ep el espectro del primer pulso. Compruebe ahora cómo es la magnitud y la fase:

```
dibuf(abs(ep))
dibuf(angle(ep))
dibuf([angle(ep) real(ep)])
```

verá que la magnitud es el valor absoluto de la parte real y es par, y que la fase es impar y vale 0 para los valores positivos de la amplitud y $\pm\pi$ para los negativos.

# LABORATORIO DE SEÑALES Y SISTEMAS

## PRACTICAS 2 Y 3

## CURSO 2005-2006

**Miguel Angel Rodríguez Hernández**

# PRACTICA 2

## SEÑALES Y SISTEMAS DISCRETOS

En esta práctica se realiza un estudio de las señales y de los sistemas en tiempo discreto aprovechando las facilidades que ofrece MATLAB.

La práctica consta de 3 partes claramente diferenciadas. En la primera parte el alumno aprenderá a generar y representar correctamente señales discretas utilizando como herramienta MATLAB. Para ello el alumno deberá implementar una serie de pequeñas funciones que permitan generar y representar algunas señales básicas como son el impulso unidad, las sinusoides, el escalón unidad, las exponenciales reales y las exponenciales complejas. La segunda parte trata de familiarizar al alumno con la operación de convolución. Para ello el alumno deberá implementar un pequeño programa que realice la convolución de dos señales, para a continuación comprobar las propiedades de la convolución mediante la utilización del programa previamente diseñado. La tercera parte trata el desarrollo en serie de Fourier. En esta parte el alumno deberá obtener los coeficientes del desarrollo en serie de Fourier de diversas señales periódicas, así como las señales correspondientes a partir de coeficientes de desarrollos en serie de Fourier.

El apartado 0 de la práctica contiene un cuestionario de aspectos teóricos relacionados con el contenido de la práctica que el alumno deberá entregar al profesor antes del inicio de la práctica.

Como nota final de esta introducción, se recomienda a los alumnos guardar las funciones que se irán realizando a lo largo de la práctica.

## 1. SEÑALES BASICAS

En este apartado el alumno realizará una serie de funciones que implementen algunas de las señales discretas más utilizadas. Para ello se utilizarán las facilidades de MATLAB para generar señales discretas definiéndolas como vectores.

MATLAB presenta un pequeño inconveniente a la hora de manejar las señales discretas por medio de vectores, que es la imposibilidad de manejar índices negativos o cero para designar las posiciones del vector. Por ello para obtener toda la información de la señal generada es necesario facilitar junto con los valores de la señal el vector de índices de las posiciones de la señal.

La representación de señales discretas en MATLAB se realiza mediante la función *stem*. La función *stem*(Y) dibuja la secuencia de datos Y mediante líneas terminadas en círculos. La función *stem*(X,Y) dibuja la secuencia Y en los valores especificados por el vector X.

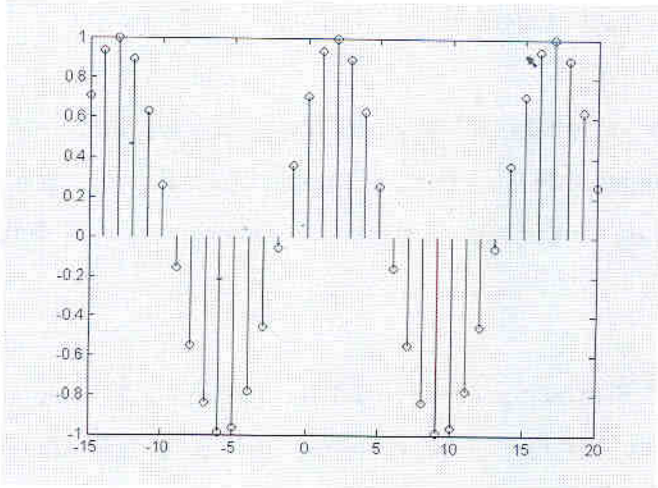
La anterior explicación se entenderá mejor con un ejemplo. Supongamos que se quiere dibujar la señal

$\mathrm{sen}(2\pi n/15 + \pi/4)$ entre n =-15 y n=20.

La secuencia de instrucciones a teclear será:

```
>> n=-15:20;
>> x=sin(2*pi*n/15+pi/4);
>> stem(n,x)
```

y el resultado gráfico obtenido deberá ser:




### 1.1. Señal impulso unidad.

Cualquier señal discreta se puede expresar en función de la señal impulso unidad mediante la expresión:

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

En este apartado se van a realizar los pasos necesarios para visualizar esta expresión.

a) Genere y represente la función impulso unidad, δ(n), en el intervalo n=-8 a n=12.

Nota: Para generar un vector en MATLAB se puede utilizar la instrucción *zeros*(1, N). La función *zeros*(M, N) genera una matriz de dimensión MxN cuyos valores son todos cero. Para situar el valor de la delta en el lugar correcto puede usar al función *find*(d).

Así el código que resuelve este apartado es:

```
delta=zeros(1,21);
n=-8:12;
i=find(n==0);
delta(i)=1;
plot(n,delta);
stem(n,delta)
```

Un código alternativo sin usar la función find sería:

```
delta=zeros[1,21];
n=-8:12;
delta(9)=1;
plot(n,delta);
```

b) Genere y represente las siguiente señales:

$x_1(n) = 0.5\delta(n)$ $\quad 0 \le n \le 10$

$x_2(n) = 0.75\delta(n-3)$ $\quad -10 \le n \le 10$

$x_3(n) = 1.25\delta(n+30)$ $\quad -40 \le n \le 40$

$x_4(n) = 1.5\delta(n-7) + 1.5\delta(n+2)$ $\quad -10 \le n \le 10$

cuidado ! 0'75.* delta(n, $\underbrace{+3}_{n_0}$)

Se tiene la señal x(n) definida por

$$x(n) = \begin{cases} -2 & n = -2 \\ 1 & n = -1 \\ 2 & n = 0 \\ -1 & n = 1 \\ 0 & \text{resto de valores} \end{cases}$$

c) Utilizando la ecuación $x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k)$ exprese matemáticamente la señal $x(n)$ como suma de deltas.

d) Represente en el intervalo $-10 \le n \le 10$ cada uno de los sumandos de la expresión que ha obtenido junto con $x(n)$ y compruebe visualmente que la suma de todos los sumandos corresponde a $x(n)$.

Nota utilice para ver los resultados la función *subplot*(51p) de MATLAB, siendo p el número de gráfica. La función *subplot*(mnp) divide la figura en de MATLAB en varias representaciones siendo m el número de divisiones verticales.

### 1.2. Sinusoides.

Otras señales básicas muy utilizadas son las la ondas sinusoidales. Estas señales tienen las expresiones

$$x(n) = A\cos(\omega n + \theta)$$

$$x(n) = A\mathrm{sen}(\omega n + \theta)$$

siendo $A$ la amplitud, $\omega$ la pulsación y $\theta$ la fase.

Para generar estas señales MATLAB proporciona las funciones *cos*(x) y *sin*(x).

a) Genere y represente las siguientes señales:

$x_1(n) = 2\cos(\frac{\pi}{17}n)$ $\quad 0 \le n \le 50$

$x_2(n) = 0.5\sin(\frac{\pi}{17}n)$ $\quad -20 \le n \le 30$ $\quad = 0'5 \sin\left(\frac{2\pi}{35}n\right)$

$x_3(n) = \sin(3\pi n + \frac{\pi}{2})$ $\quad -10 \le n \le 10$ $\quad = \sin\left(\frac{2\pi}{2/3} + \frac{\pi}{2}\right)$

$= \sin\left(2\pi \cdot \frac{3}{2} + \frac{\pi}{2}\right)$

$x_4(n) = \cos(\frac{\pi}{\sqrt{23}}n)$ $\quad -10 \le n \le 50$

Diga que señales de las anteriores son periódicas y cuales no. Explique el motivo.

b) Escriba una función MATLAB que genere y dibuje un seno de longitud finita. Para guardar esta función deberá crear un fichero MATLAB. Para ello primero deberá pinchar con el ratón en "File", después en "New" y finalmente en "M-file". Guarde el fichero que acaba de abrir en su directorio de trabajo con el nombre *pract_seno*.

La primera línea de la función será:

function [x, n] = pract_seno(A, pulsacion, fase, inicio, fin),

siendo

x la señal seno resultante tras aplicar la función,

n el vector de índices que acompaña al seno,

A la amplitud del seno,

pulsacion la pulsación del seno,

fase la fase inicial del seno,

inicio el primer valor de n a calcular y representar,

fin el último valor de n a calcular y representar

c) Compruebe la correcta implementación de la función *pract_seno* con las señales $x_2(n)$ y $x_3(n)$ del apartado a).

### 1.3. El escalón unidad.

La función escalón se define como

$$u(n) = \begin{cases} 1 & \text{n} \geq 0 \\ 0 & \text{resto de valores} \end{cases}$$

a) Escriba la función *pract_escalon* que genere y dibuje la función escalón. La primera línea de la función será:

function [x, n]=pract_escalon(inicio, fin),

siendo

x la señal escalón resultante tras aplicar la función,

n el vector de índices que acompaña al escalón,

inicio el primer valor de n a calcular y representar,

fin el último valor de n a calcular y representar

b) Compruebe el correcto funcionamiento de la señal que acaba de escribir tomando como valores de inicio y fin –10 y 20.

### 1.4. Exponenciales reales.

La exponencial es una señal básica que tiene la siguiente expresión

$x(n) = a^n$, siendo $a$ una constante.

Es de destacar que el valor de $a$ debe de cumplir algunas restricciones cuando $n$ tiende a infinito (positivo o negativo). Así por ejemplo para señales causales, $a$ debe ser menor que 1 para que la secuencia no tienda a infinito.

a) Escriba la función *pract_exp* que genere y dibuje la función exponencial. La primera línea de la función será:

function [x, n] = pract_exp(a, inicio, fin),

siendo

x la señal exponencial resultante tras aplicar la función,

n el vector de índices que acompaña a la señal,

a el valor al que se le aplica el exponente,

inicio el primer valor de n a calcular y representar,

fin el último valor de n a calcular y representar

Nota: La instrucción MATLAB *y=a.^n* genera como resultado el valor de *a* elevado a *n*.

b) Compruebe el correcto funcionamiento de la señal que acaba de escribir generando una exponencial con a = 0.5, inicio = -2 y fin = 10.

En numerosas ocasiones es necesario realizar una suma de la secuencia exponencial causal $x(n) = a^n u(n)$, para ello se utiliza la expresión

$$\sum_{n=0}^{L-1} a^n = \frac{1-a^L}{1-a}$$

c) Escriba la función *pract_suma_exp* que implemente la ecuación anterior. La primera línea de la función será:

function x =pract_suma_exp(a, L),

siendo

x el resultado de la suma,

a el valor al que se la aplica el exponente,

L el último valor de n a calcular

d) Suponga que tiene una secuencia exponencial causal con a = 0.7 y L = 33. Compruebe el funcionamiento de la función *pract_suma_exp* comparando el resultado con el obtenido al generar una exponencial con la función *pract_exp* y aplicar la función *sum* de MATLAB. La función *sum*(X) genera como resultado la suma de los valores del vector X.

### 1.5. Exponenciales complejas.

Las exponenciales complejas tienen una gran importancia en el tratamiento de señales debido a su utilización para representar señales por el desarrollo en serie y la transformada de Fourier. Su expresión es

$$x(n) = Ae^{j(\omega n+\theta)}$$

a) Escriba la función *pract_exp_compl* que genere y dibuje la función exponencial compleja. En este caso para representar la función será necesario utilizar 2 gráficas una para la parte real y otra para la parte imaginaria. Por ello se utilizarán las funciones de MATLAB *subplot*, *real* e *imag*. Para generar la exponencial será necesario utilizar la función *exp*. Las funciones *real*, *imag* y *exp* dan como resultado la parte real, la parte imaginaria y el resultado de e elevado a un valor (real o complejo)

La primera línea de la función será:

function [x, n] = pract_exp_compl(a, pulsación, fase, inicio, fin)

siendo

x la señal resultante tras aplicar la función,

n el vector de índices que acompaña a la señal,

A la amplitud,

pulsacion la pulsación,

fase la fase inicial,

inicio el primer valor de n a calcular y representar,

fin el último valor de n a calcular y representar

b) Genere algunas señales para comprobar el correcto funcionamiento de la función *pract_exp_compl*.

## 2.- LA CONVOLUCION

La convolución es una operación que permite obtener la señal a la salida de un sistema lineal e invariante conociendo únicamente la respuesta al impulso del sistema y la señal de entrada.

La expresión matemática de la convolución es:

$$z(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

Para implementar la convolución MATLAB facilita la función *conv*. La función Z = *conv*(X, H) convoluciona los vectores X y H. La longitud del vector resultante Z es la suma de la longitud del vector X más la longitud del vector H menos 1.

El problema que presenta la utilización de la función *conv* es que al igual que ocurría con la representación de señales, MATLAB no permite manejar vectores con índices negativos.

a) Utilizando la función *conv* de MATLAB Escriba la función *pract_conv* que genere y dibuje correctamente la convolución de dos señales. La primera línea de la función será:

function [z, n] = pract_conv(x, nx, y, ny ),

siendo

z la convolución resultante tras aplicar la función,

n el vector de índices que acompaña a la convolución,

x la primera señal a convolucionar,

nx el vector de índices de x,

y la segunda señal a convolucionar,

ny el vector de índices de y

Para comprobar la función que acaba de realizar, va a estudiar a continuación las propiedades de la convolución. Si alguna de las siguientes propiedades no se verifica experimentalmente probablemente será debido a una mala implementación de la función *pract_conv*.

### 2.1. Propiedad de identidad.

Esta propiedad dice que el operador identidad de la convolución es la función δ(n). Se expresa matemáticamente por

$$x(n) * \delta(n) = x(n)$$

a) Genere 2 funciones δ(n) y compruebe que el resultado de convolución de 2 deltas es otra función delta.

Nota: para el estudio de esta propiedad y de las posteriores, la elección del intervalo del vector de tiempos n debe realizarlo el alumno.

b) Genere la señal x(n) =δ(n+2)-δ(n)-3δ(n-2)+δ(n-3), convoluciónela con una de las deltas anteriores y verifique que se cumple la propiedad.

### 2.2. Propiedad conmutativa.

Esta propiedad dice que el orden de los operandos que intervienen en la convolución no influye en el resultado final. Matemáticamente se expresa por:

$$x_1(n) * x_2(n) = x_2(n) * x_1(n)$$

a) Genere las señales

$x_1(n)$ = δ(n+1)-δ(n)+2δ(n-2)+δ(n-3),

$x_2(n)$ = -δ(n)+2δ(n-2)-δ(n-3)-δ(n-5),

y realice la comprobación de la propiedad anterior.

### 2.3. Propiedad asociativa.

Matemáticamente se expresa por

$$x_1(n)*x_2(n)*x_3(n)=(x_1(n)*x_2(n))*x_3(n)=x_1(n)*(x_2(n)*x_3(n))$$

a) Utilizando las señales generadas en el apartado anterior y la señal

$x_3(n) = \delta(n)+\delta(n-2)$

compruebe dicha propiedad.

### 2.4. Propiedad distributiva.

Matemáticamente se expresa por

$$(x_1(n)*x_3(n))+(x_2(n)*x_3(n))=(x_1(n)+x_2(n))*x_3(n)$$

a) Utilizando las señales generadas en el apartado anterior compruebe esta propiedad.

## 3. ANALISIS DE FOURIER

El análisis de Fourier es una potente herramienta para el estudio de señales. Se divide a su vez en dos herramientas el "Desarrollo en Serie de Fourier" para señales periódicas y la "Transformada de Fourier" para señales no periódicas. Existen técnicas específicas para implementar la transformada de Fourier, pero en este apartado el alumno se va a limitar a estudiar el desarrollo en serie de Fourier.

El desarrollo en serie de Fourier de una señal periódica $x(n)$ se define por:

$$x(n) = \sum_{k=0}^{N-1} a_k e^{j\frac{2\pi}{N}kn}$$

siendo $N$ el periodo de la señal y

$$a_k = \frac{1}{N}\sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn}$$ los coeficientes del desarrollo en serie de Fourier.

a) Escriba la función *coef_DSF* que calcule los coeficientes del desarrollo en serie de Fourier de una señal periódica. La primera línea de la función será

function a = coef_DSF(x)

siendo

x el vector que contiene los valores de señal del intervalo 0 – (N-1), que corresponden a un periodo de la señal.

a el vector que contiene los coeficientes 0 – (N-1) del desarrollo en serie.

b) Genere la señal x(n) = cos(5πn/7) entre n=-30 y n=30. Represéntela y obtenga su periodo. Calcule de manera teórica el valor del los coeficientes de su desarrollo en serie de Fourier. Aplique la función que acaba de desarrollar y compruebe que los resultados teóricos y prácticos coinciden.

cuidado: pasarle sólo un periodo de x

c) Escriba la función *x_DSF* que calcule una señal periódica a partir de sus coeficientes del desarrollo en serie de Fourier. La primera línea de la función será

function x= x_DSF(a)

cuidado al representar valores imaginarios con stem, hace una cosa rara.
usas stem(real(a))
stem(real(x))



Para verla periodicamente puedes hacer stem([real(a) real(a) real(a)]) siendo

a el vector que contiene los coeficientes 0 – (N-1) del desarrollo en serie.

x el vector que contiene los valores de señal del intervalo 0 – (N-1).

Nota: los valores de x obtenidos a partir de *x_DSF* serán complejos, pero si las señales utilizadas son reales la parte imaginaria de dichos valores deberá ser cero o muy cercana a cero.

d) Con los coeficientes calculados en el apartado b) y utilizando la función *x_DSF* represente la señal $x(n) = \cos(5\pi n/7)$ entre n=-30 y n=30 y compruebe el resultado.

e) Utilice consecutivamente las funciones *coef_DSF* y *x_DSF* sobre los siguientes vectores que representan el intervalo 0 – (N-1) de dos señales periódicas:

c1) x1=[0 0 0 1 1 1 1 1 0 0 0];

c2) x2=[0 1 2 -3 4 -5 6 3 1 ];

Compruebe que las señales resultantes tras obtener los coeficientes de los desarrollos en serie y reconstruir las señales a partir de dichos coeficientes coinciden con las señales iniciales.

### 3.1. Cálculo de la señal a la salida de un sistema utilizando el análisis de Fourier.

La salida de un sistema lineal e invariante cuando la entrada es una señal periódica, se puede calcular en función del desarrollo en serie de Fourier de la señal de entrada y de la respuesta en frecuencia de sistema. Si se tiene una señal cuyo desarrollo en serie de Fourier es

$$x(n) = \sum_{k=0}^{N-1} a_k e^{j\frac{2\pi}{N}kn}$$

y esta señal es la entrada de un sistema lineal en invariante con respuesta al impulso $h(n)$ y respuesta en frecuencia H(Ω), la señal a la salida del sistema seguirá la expresión

$$y(n) = \sum_{k=0}^{N-1} H(\frac{2\pi}{N}k) a_k e^{j\frac{2\pi}{N}kn}$$

La expresión anterior implica que los coeficientes del desarrollo en serie de Fourier de la secuencia periódica y(n) serán:

$$b_k = H(\frac{2\pi}{N}k) a_k$$

Sea el sistema $h(n) = \left(\frac{1}{a}\right)^n u(n)$ con $a$>1 cuya respuesta en frecuencia es

$$H(\Omega) = \frac{1}{1 - \frac{1}{a} e^{-j\Omega}}$$

a) Escriba la función *pract_H* que calcule $H(\frac{2\pi}{N}k)$ para el sistema $H(\Omega) = \dfrac{1}{1 - \frac{1}{a}e^{-j\Omega}}$. La primera línea de la función será

function H= pract_H(a, N)

siendo

a el valor de la constante *a*.

N el periodo de la señal x(n) que entra al sistema

H el vector que contiene los valores de $H(\frac{2\pi}{N}k)$ desde k= 0 a k= (N-1).

Sea el sistema $h1(n) = \left(\frac{1}{3}\right)^n u(n)$ cuya respuesta en frecuencia es

$$H1(\Omega) = \frac{1}{1 - \frac{1}{3}e^{-j\Omega}}$$

b) Utilizando las funciones *coef_DSF, x_DSF* y *pract_H* calculadas previamente, calcule la salida del sistema h1(n) cuando la entrada es x(n) = cos(5πn/7). Represente la señal de entrada y la señal de salida entre n= 0 y n=13. Compárelas.

```
n=0:1:13;
x=cos(5*pi*n/7);
a=coef_DSF(x);
H=pract_H(3,14);
b=a.*H;
y=x_DSF(b);
stem(real(x));
stem(real(y));
```

periodo de señal

c) Calcule la salida del sistema h1(n) cuando la señal de entrada es la repetición periódica del siguiente periodo

$$x(n) = \begin{cases} 2 & n = 0 \\ -1 & n = 1 \\ 1 & n = 2 \\ 2 & n = 3 \end{cases}$$

Represente la señal de entrada y la señal de salida entre n= -10 y n=10. Compárelas.

# PRACTICA 3

## MUESTREO

En esta práctica se va a estudiar el muestreo uniforme de señales continuas.

La práctica se divide en dos partes: el estudio de los efectos del muestreo en el dominio temporal y el estudio de los efectos del muestreo en el dominio frecuencial. El estudio del los efectos en el dominio temporal puede resultar engañoso cuando las frecuencias de muestro, aun cumpliendo el teorema de Nyquist, son bajas. Por ello aunque la práctica comienza estudiando los efectos del muestreo en el dominio temporal, es mucho más clarificador el estudio de los efectos en el dominio frecuencial que se realiza en la segunda parte de la práctica.

Esta práctica mezcla señales continuas (antes de muestrear) y señales discretas (después de muestrear). Sin embargo la herramienta utilizada, MATLAB, únicamente permite el uso de señales discretas, por lo que va a ser necesario simular las señales continuas. Esta simulación de señales continuas es en si mismo otro aspecto interesante de esta práctica.

Como resultado final de la práctica, el alumno obtendrá un simulador de un sistema de muestreo/recuperación de señales. Para implementar este simulador se hará uso de las funciones y procedimientos desarrollados a lo largo de la práctica por lo que se recomienda guardar dichas funciones y procedimientos según se vayan generando.

El apartado 0 de la práctica contiene un cuestionario de aspectos teóricos relacionados con el contenido de la práctica que el alumno deberá entregar al profesor antes del inicio de la práctica.

# 1. MUESTREO EN EL DOMINIO TEMPORAL.

El muestreo uniforme de señales continuas consiste en tomar valores de forma equiespaciada en el tiempo de una señal continua, y formar con dichos valores una nueva señal discreta relacionada con la continua. Matemáticamente se puede expresar de la siguiente manera:

$$x(n) = x_a(t)\,|_{t=nT}$$

siendo

$x_a(t)$ la señal continua a muestrear

$x(n)$ la señal discreta resultante

$T$ el periodo de muestreo

La inversa del periodo de muestreo $T$ se denomina frecuencia de muestreo $f_s = \frac{1}{T}$.

Para que la señal muestreada se pueda recuperar sin perdida de información es necesario que no se produzca solapamiento y para ello debe cumplirse el teorema de Nyquist:

$$f_s \geq 2B$$

donde $B$ es el ancho de banda (frecuencia más alta en nuestro caso) de la señal a muestrear.

En los siguientes puntos de este apartado se van a estudiar los efectos del muestreo y solapamiento en el dominio temporal sobre una señal sinusoidal y sobre una señal chirp.

### 1.1. Muestreo de una sinusoide.

Considere la señal sinusoidal expresada en tiempo continuo por la ecuación:

Tsim = 0'0001
t = 0 : Tsim : 0'01;
f0= 200
xa = sin(2 * pi * f0 * t)

$$x_a(t) = sen(2\pi f_0 t)$$

Se puede obtener una señal en tiempo discreto a partir de la señal anterior utilizando el muestreo uniforme resultando

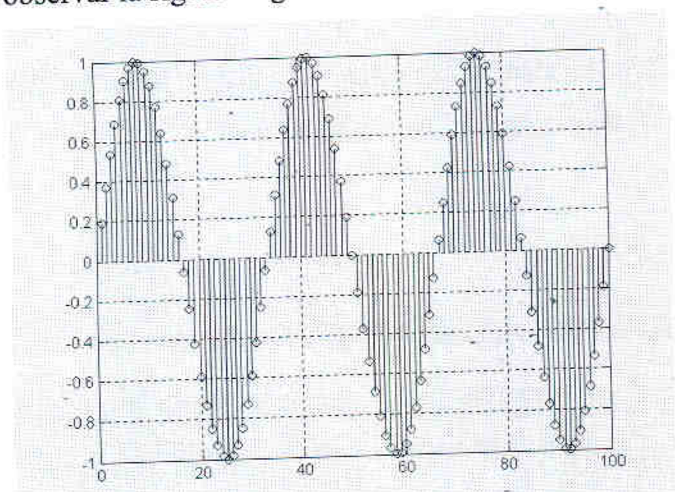
fs = 10 000    n= 0 : fs * t(end)
Ts = 1 / fs
x = sin (2 * pi * f0 * n * Ts)

$$x(n) = x_a(t)\,|_{t=nT} = sen(2\pi f_0 nT)$$

x = muestreo_sinusoide (f0, fs, T)
frec de la sinusoide
frec de muestreo
tiempo [0,T] segundos y T=0'01 = 10ms

En este apartado se van a ver los efectos del muestreo sobre diversas sinusoides. Para ello se pide realizar los epígrafes a), b), c) y d) utilizando una frecuencia de muestreo, $f_s$ , de 10 KHz.

a) Se pide representar el resultado de muestrear un seno de frecuencia 300 Hz, en el intervalo de tiempo continuo t comprendido entre 0 y 10 ms. Para la representación deberá utilizar la función *stem* de MATLAB y como resultado se deberá observar la siguiente gráfica.




b) Realice ahora una serie de representaciones con senos de frecuencias que vayan desde 100 hasta 500 Hz, con una diferencia entre ellos de 100Hz. Observe simultáneamente mediante la función *subplot* las 5 sinusoides. Deberá observar como va creciendo la frecuencia de los senos.

for i = 1 : 1 : 5
f0 = 100 * i ;
subplot ( 5, 1, i )
stem (muestreo-sinusoide (f0, 10 000, 0'01))

c) Realice ahora una serie de representaciones con senos de frecuencias que vayan desde 9500 Hz hasta 9900 Hz, con una diferencia entre ellos de 100Hz. Observe simultáneamente mediante la función *subplot* las 5 sinusoides. En este caso, ¿continua creciendo la frecuencia aparente de los senos representados?. Explique a que es debido este efecto.

d) Realice ahora una serie de representaciones con senos de frecuencias que vayan desde 4750 Hz hasta 4950 Hz, con una diferencia entre ellos de 50Hz. Observe simultáneamente mediante la función *subplot* las 5 sinusoides y conteste a las siguientes preguntas. ¿Se continua observando la representación de un seno muestreado en las gráficas?. ¿Podría recuperar alguno de los senos en tiempo continuo a partir de estos senos muestreados?, en caso afirmativo indique cuales. ¿Son periódicas las señales representadas?. Explicar a que se debe este efecto del cambio aparente de forma de onda y de periodo. Nota: realizar la representación tanto con *stem* como con *plot*.

Todos los que cumplan el criterio de Nyquist se pueden recuperar

e) Calcule la frecuencia de un seno que genere al ser muestreado con una $f_s$ de 10 KHz una secuencia no periódica. Compruébelo.

### 1.2. Muestreo de una señal chirp.

La señal chirp se define como una señal senoidal cuya frecuencia varia linealmente con el tiempo. Esta variación de la frecuencia con el tiempo implica, a diferencia de la función seno utilizada anteriormente, que su representación espectral contiene más de una frecuencia y que además dichas frecuencias dependen de los valores concretos de tiempo en los que se represente la señal chirp.

La definición matemática de la señal chirp es:

$$c(t) = \cos(\pi \mu t^2 + 2\pi f_0 t)$$

siendo $f_0$ la frecuencia inicial de la chirp y $\mu$ el coeficiente de variación lineal de la frecuencia.

Su frecuencia instantánea se puede obtener haciendo la derivada de la fase con respecto al tiempo, resultando:

$$f_i(t) = \mu t + f_0$$

En la expresión anterior se puede observar la variación lineal de la frecuencia instantánea $f_i(t)$ con el tiempo.

a) Con parámetros de la señal chirp $f_0$=0, $\mu$=600KHz/s y con un intervalo de duración de la señal chirp de 0 ms a 10 ms, se pide calcular teóricamente el intervalo de frecuencias recorrido por la señal chirp en dicho intervalo.

en $t=0$ → $f = f_0 = 0$
en $t = 10ms$ → $f = \mu \cdot t = 600k \cdot 0'01$

b) Si se muestrea la señal chirp con una frecuencia de muestreo de 10 KHz, calcule teóricamente a partir de que muestra se producirá solapamiento. Nota: el solapamiento se producirá cuando la frecuencia sea mayor que 5 KHz.

c) Tomando como frecuencia de muestreo 10 KHz, represente las muestras de la chirp en tiempo discreto. Observe el resultado utilizando la función *stem* y la función *plot* de MATLAB. ¿Se produce solapamiento?.

al muestrear $f_s = 10kHz$

$$c(n) = \cos\left(\pi \cdot \mu \cdot \left(\frac{n}{f_s}\right)^2 + 2\pi \cdot f_0 \cdot \left(\frac{n}{f_s}\right)\right)$$

$n = 0 : 1 : 10ms \cdot f_s$

$$f_i(n) = \mu \cdot \left(\frac{n}{f_s}\right)^2$$

a partir de que muestra hay solapamiento?

$$f_i(n) = \mu \cdot \left(\frac{n}{f_s}\right) = 5000 \Rightarrow n = \frac{5000}{\mu} \cdot f_s = \frac{5000}{600 \cdot 10^3} \cdot 10 \cdot 10^3 = 83'\overline{3}$$

## 2. MUESTREO EN EL DOMINIO DE LA FRECUENCIA

En este apartado se va a realizar la simulación del muestreo de una señal analógica y de sus efectos en el dominio de la frecuencia.

Cuando se muestrea una señal en tiempo continuo, el espectro de la señal discreta resultante está relacionado con el espectro de la señal continua inicial pero ha experimentado 3 transformaciones. La primera transformación es un cambio en el eje de frecuencias consistente en un paso de frecuencias de señales continuas a frecuencias de señales discretas conjuntamente con un escalado del eje de frecuencias. El resultado de este escalado produce una correspondencia del intervalo de frecuencias de señales continuas 0-$f_s/2$ Hz con el intervalo de frecuencias de señales discretas 0-$\pi$. La segunda transformación que produce el muestreo es que el espectro de la señal muestreada está formado por la repetición periódica del espectro de la señal continua, esta repetición periódica puede producir solapamiento si el periodo de muestreo no cumple el teorema de Nyquist. El tercer efecto que produce el muestreo es la multiplicación de la máxima amplitud de la señal discreta por un factor igual a $1/T$, siendo $T$ el periodo de muestreo.

En este apartado se van a utilizar tanto señales continuas como señales discretas. Sin embargo MATLAB no permite trabajar con señales continuas, por lo que será necesario simular estás señales continuas. Para simular estas señales continuas se va a utilizar una frecuencia de muestreo muy superior a la frecuencia de Nyquist, dicha frecuencia se va denominar frecuencia de simulación, $f_{sim}$, para no confundirla con las otras frecuencias de muestreo utilizadas en la práctica. De esta forma lo que realmente se va a hacer en este apartado es muestrar una señal discreta, sin embargo mediante esta técnica podemos simular los efectos que ocurren en el muestreo de una señal continua real.

La parte final de este apartado es la realización de un simulador completo del proceso muestreo/reconstrucción por lo que se recomienda ir guardando los procedimientos y funciones que se vayan realizando.

Para el resto de la práctica apartados y salvo que se indique expresamente lo contrario la frecuencia de simulación $f_{sim}$ estará fijada a 100 KHz.

$f_{sim} = 100 \cdot 10^3$ Hz

$t$ = linspace(0, 10ms, 10ms * $f_{sim}$)

ó $t = 0 : T_{sim} : t_{final}$

**2.1. Generación y representación de la señal analógica.**

a) Genere una señal analógica senoidal, $x(t) = sen(2\pi f\, t)$, con frecuencia 6 KHz, en el intervalo comprendido entre 0 y 10 ms.

Note que lo que realmente está generando es una señal muestreada con $f_{sim}$ que simula la señal analógica.

b) Usando la función *plot* genere la función *dibuja* que permita representar gráficamente la señal anterior con sus verdaderas magnitudes continuas. La función *dibuja* siempre comenzará la representación de la señal en 0 segundos. La primera línea de la función será

function dibuja(x, T),

siendo

x la señal a representar. (Note que aunque se este simulando una señal continua, lo que en realidad se tiene es una señal muestreada con frecuencia $f_{sim}$).

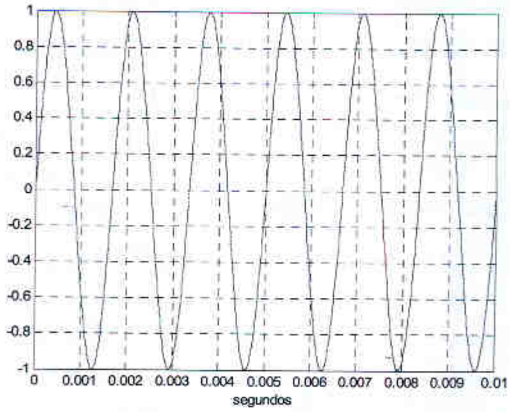
T el periodo de muestreo. (de simulación)

Nota: Si se utiliza como prueba de la función *dibuja* un seno con frecuencia 600 Hz y el intervalo de representación está entre 0 y 10 ms el resultado de la función anterior debería ser:

$\frac{f_{sim}}{f_s} = n \quad f_s = \frac{f_{sim}}{n}$



al muestrear el eje de frec se normaliza respecto a $f_s$




c) Usando la función *dibuja_tf*(x,T) proporcionada en la práctica, represente el módulo del espectro de la señal continua simulada en el apartado a) y verifique que se trata de un seno de 6 KHz.

$2\pi \frac{f_0}{f_s} = 2\pi f_0 T_s$

### 2.2. Conversión A/D.

El conversor A/D (analógico/digital) toma muestras de la señal continua cada cierto intervalo $T_s$ (lo que implica una frecuencia de muestreo $f_s = 1/T_s$). La conversión A/D se va a simular tomando un subconjunto de las muestras de la señal generada en el apartado anterior. En realidad lo que se está haciendo es muestrear una señal que ya está muestreada. Para evitar problemas en la simulación se deben tomar valores de $T_s$ que sean múltiplos enteros del valor de $T_{sim}=1/f_{sim}$ utilizado en la generación de la señal "analógica". $T_s = n T_{sim}$

a) Represente la señal discreta resultante al muestrear el seno de 6 KHz del apartado 2.1.a) con una frecuencia de muestreo de 25 KHz. La generación de esta señal se puede realizar como en el apartado 2.1.a) , pero también de una forma más cercana a la realidad muestreando la señal generada en el apartado 2.1.a). Haga el muestreo por los dos procedimientos y compruebe que el resultado es el mismo.

para de x(n) generada

b) Utilizando la función de MATLAB *freqz*(x) que genera muestras del espectro de la señal x entre 0 y $\pi$, y las facilidades de la función plot; genere la función *dibuja_tf_senal_muestreada*(x) que calcule y represente el módulo del espectro de la señal muestreada x entre 0 y π. Compruebe la función con la señal generada en el apartado a)

Nota: La instrucción X=freqz(x); genera la transformada de Fourier de la señal discreta x entre 0 y π muestreada con 512 puntos.

c) Muestree el seno de 6 KHz del apartado 2.1.a) con las siguientes frecuencias de muestreo, $f_s$, de

c1) 15 KHz

c2) 13 KHz

c3) 11 Khz

c4) 10 Khz

Represente mediante la función *dibuja_tf_senal_muestreada* las señales que acaba de obtener. ¿Qué efecto observa a medida que disminuye la frecuencia de muestreo?

### 2.3. Diseño del filtro reconstructor.

En la conversión discreto/continuo una parte muy importante es el filtro reconstructor. En este apartado se va a abordar el diseño de dicho filtro reconstructor.

El filtro reconstructor de un sistema real de reconstrucción debe ser continuo, sin embargo al igual que ocurre con la señales, MATLAB no permite trabajar con filtros en tiempo continuo. Por este motivo el filtro reconstructor a diseñar será un filtro digital que simule el verdadero filtro reconstructor analógico. Para ello se va a diseñar un filtro de Chebyshev digital mediante la función de MATLAB *cheby2*. La función [B,A] = *cheby2*(N,R,Wn) diseña un filtro paso bajo digital de orden N con ~~rizado~~ atenuacion en la banda atenuada R y frecuencia de inicio de la banda atenuada Wn. La función *cheby2* devuelve los coeficientes B y A de un filtro con longitud de N+1 coeficientes. La frecuencia Wn debe estar ente 0 y 1, donde 1 corresponde con la mitad de la frecuencia de muestreo.

Se desea diseñar un filtro con frecuencia de inicio de la banda atenuada analógica $f_s/2$ (la mitad de la frecuencia de muestreo). Para simular esta frecuencia será necesario seleccionar como frecuencia de la banda atenuada al hacer el diseño del filtro digital equivalente Wn=$2*(f_s/2)/f_{sim}$.

a) Calcule mediante la función *cheby2* los coeficientes del filtro reconstructor de orden 10 y 60 dB de atenuación en la banda atenuada. Utilice una $f_s$ de 25 KHz y una $f_{sim}$ de 100 KHz.

A continuación va a representar la respuesta en frecuencia del filtro reconstructor que acaba de diseñar utilizando la función *freqz* de MATLAB. La función *freqz* genera la respuesta en frecuencia de un filtro digital. [H,W] = *freqz* (B, A, N) genera la respuesta en frecuencia del sistema con una resolución de N puntos, conteniendo el vector H la información compleja de cada frecuencia y el vector W las frecuencias expresadas en radianes/muestra. Si la llamada a la función es de la forma [H,W] = *freqz* (B,A), N toma un valor por defecto de 512 muestras.

Para interpretar la representación que va a realizar, debe tener en cuenta que la frecuencia del filtro digital $\pi$ corresponde con la frecuencia $f_{sim}/2$ del filtro analógico que se está simulando.

b) Se pide generar la función *resp_freq_filtro* que calcula y representa el módulo de la respuesta en frecuencia del filtro analógico simulado. En el eje horizontal deben aparecer las frecuencias en KHz y en el eje vertical la magnitud de la respuesta en frecuencia del filtro expresada en dB. Para la representación utilice las facilidades de la función *plot* de MATLAB. Para calcular los coeficientes del filtro use la función *cheby2*. La primera línea de la función *resp_freq_filtro* será:

function resp_freq_filtro(orden, atenuacion, fs, fsim)

siendo

orden el orden del filtro a calcular y representar,

atenuacion la atenuación del filtro en la banda atenuada expresada en dB

fs la frecuencia de muestreo

fsim la frecuencia de simulación

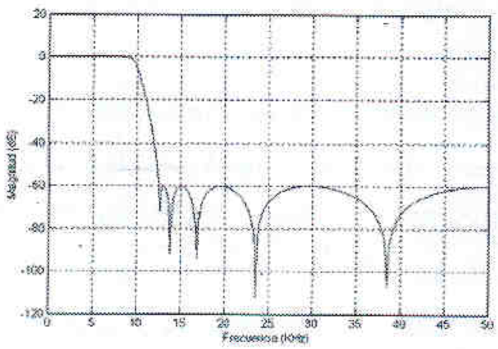
Finalmente, una vez generada la gráfica, y únicamente para mejorar la presentación de dicha gráfica, teclee estas 2 instrucciones y compruebe su efecto:

```
>>xlabel('Frecuencia (KHz)')
>> ylabel('Magnitud (dB)')
```

c) Para comprobar el correcto funcionamiento de la función que acaba de realizar ejecute la siguiente instrucción

```
>>resp_freq_filtro(10, 60, 25000, 100000)
```

El resultado obtenido deberá ser

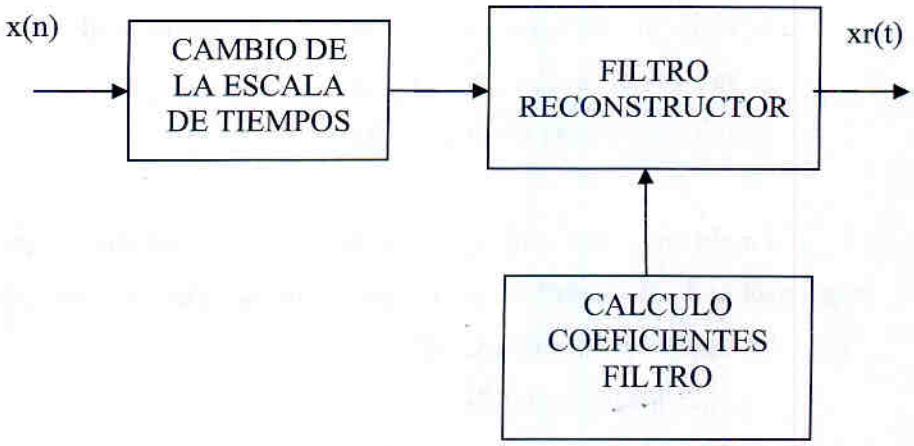



**2.4. Conversión D/A.**

La conversión discreto/continuo (o más en general la digital/analógica, D/A) involucra dos acciones, por una parte hay que realizar un cambio de escala en el eje de tiempos y por otra parte es necesario aplicar un filtro reconstructor. Ambas situaciones se van a simular de manera digital.



Laboratorio de Señales y Sistemas-MUESTREO




El cambio de la escala de tiempos se va a simular mediante la inserción de un determinado número de ceros entre las muestras de la señal muestreada. El número de ceros a insertar depende del cociente $f_{sim}/f_s$ de ahí que un apartado anterior de la práctica se recomendara que la relación $f_{sim}/f_s$ fuera entera.

El segundo paso consiste en la realización del filtrado. Una vez obtenidos los coeficientes del filtro en el apartado anterior, se realizará el filtrado mediante la utilización de la función de MATLAB *filter*. La función Y = *filter*(B,A,X) filtra los valores contenidos en el vector X con el filtro descrito por los coeficientes A y B dando como resultado el vector Y. Esta función es una implementación de la siguiente ecuación en diferencias:

insertar $\left(\frac{f_{sim}}{f_s}-1\right)$ ceros

a(1)*y(n) = b(1)*x(n) + b(2)*x(n-1) + ... + b(nb+1)*x(n-nb)
- a(2)*y(n-1) - ... - a(na+1)*y(n-na)

Utilizando la señal generada en el apartado 2.2.a),

a) Realice el cambio en el eje de tiempos mediante la inserción de ceros entre los valores de la señal muestreada. El numero de ceros a insertar entre cada dos muestras consecutivas es $(f_{sim}/f_s)$-1. Utilizando la función *stem* visualice las 50 primeras muestras de la señal antes y después de insertar los ceros.

b) Realice el filtrado de señal generada en el apartado anterior.

c) Represente la señal recuperada en tiempo continuo mediante la función *dibuja*

d) Represente la magnitud de la transformada de Fourier de la señal recuperada mediante la función *dibuja_TF*

e) Compare la señal recuperada y su transformada de Fourier con la señal original y su transformada de Fourier. Compruebe si son iguales o si existe alguna diferencia. Explique a que son debidas las diferencias y soluciónelas hasta donde sea posible.

**2.5. Simulador del proceso de muestreo.**

A partir de las funciones desarrolladas previamente se pide realizar la función *simula_muestreo* que simule el proceso completo de muestreo/reconstrucción. La primera línea de la función será:

function [xm, xzeros, xrec]=simula_muestreo(x, fsim, fs, N, R),

teniendo como parámetros de entrada:

x la señal que simula la señal analógica,

fsim la frecuencia de simulación a la cual está muestreada x,

fs la frecuencia a la cual se debe muestrear x,

N el orden del filtro de reconstrucción y

R la atenuación mínima de la banda atenuada del filtro.

Las resultados qûe debe producir la función son:

xm la señal muestreada a frecuencia fs,

xzeros la señal xm tras añadirle los ceros y

xrec la señal reconstruida después de todo el proceso.

Además la función deberá generar 3 figuras de MABLAB:

*figure*(1) contendrá la representación en tiempo continuo de la señales x y xrec. Se representarán en dos gráficas distintas mediante *subplot* *figure*(2) contendrá la representación del modulo del espectro de la señales x y xrec. Se representarán en dos gráficas distintas mediante *subplot*

*figure*(3) contendrá la representación de la respuesta en frecuencia del filtro simulado, donde en el eje horizontal deben aparecer las frecuencias en KHz y en el eje vertical la magnitud de la respuesta en frecuencia del filtro expresado en dB.

Compruebe el correcto funcionamiento del simulador mediante la introducción de diferentes casos.

ETSI Telecomunicación, Departamento de Comunicaciones

# PRÁCTICAS 4 Y 5 DEL LABORATORIO DE SEÑALES Y SISTEMAS

## CURSO 2005-06

Gema Piñero Sipán

# ÍNDICE


## INTRODUCCIÓN A SIMULINK ........................................ 5

**1. ¿QUÉ ES SIMULINK?** ........................................ **5**
**2. ACCESO A SIMULINK (MATLAB 6.5)** ........................................ **5**
2.1 MENÚS DE LA VENTANA DE TRABAJO DE SIMULINK ........................................ 6
2.2 LIBRERÍAS DE SIMULINK ........................................ 9
**3. CONSTRUCCIÓN DE UN MODELO SENCILLO** ........................................ **10**
3.1 BLOQUE *SCOPE* ........................................ 13

## ESTUDIO TEÓRICO DE LAS MODULACIONES ANALÓGICAS ........ 17

**1. MODULACIONES ANALÓGICAS** ........................................ **17**
1.1 MODULACIÓN EN AMPLITUD (AM) ........................................ 17
1.2 MODULACIÓN EN DOBLE BANDA LATERAL (DBL) ........................................ 18
1.3 MODULACIÓN EN BANDA LATERAL ÚNICA (BLU) ........................................ 18
1.4 MODULACIÓN EN FRECUENCIA (FM) Y EN FASE (PM) ........................................ 19
1.5 MODULADORES Y DEMODULADORES ........................................ 20
1.6 RELACIÓN SEÑAL-A-RUIDO ........................................ 21

## PRÁCTICA 4: MODELOS SENCILLOS. MODULACIONES ANALÓGICAS ........................................ 23

**1. IMPORTANTE: ANTES DE EMPEZAR** ........................................ **23**
**2. RUIDO FILTRADO** ........................................ **24**
**3. MEZCLADOR** ........................................ **25**
**4. LECTURA Y ESCRITURA DE DATOS** ........................................ **26**
**5. RESOLUCIÓN EN TIEMPO Y EN FRECUENCIA** ........................................ **27**
**6. ESTUDIO PRÁCTICO DE LAS MODULACIONES ANALÓGICAS** ........................................ **28**
6.1 IMPORTANTE: FICHEROS MDL ........................................ 28
6.2 MODULACIÓN EN AMPLITUD CON PORTADORA (AM) ........................................ 29
6.3 MODULACIÓN EN DOBLE BANDA LATERAL (DBL) ........................................ 29
6.4 MODULACIÓN EN FRECUENCIA (FM) ........................................ 30
6.5 EVALUACIÓN DE LA RELACIÓN SEÑAL-A-RUIDO DE PRE Y POST-DETECCIÓN ........... 30

## ESTUDIO TEÓRICO DE LAS MODULACIONES DIGITALES ............. 35

**1. MODULACIONES DIGITALES** ........................................ **35**
1.1 TIPOS DE MODULACIÓN ........................................ 35
1.2 ANÁLISIS ESPECTRAL DE LAS MODULACIONES DIGITALES ........................................ 37
1.3 REPRESENTACIÓN GEOMÉTRICA DE SEÑALES Y RUIDO ........................................ 38
1.4 DETECCIÓN DE SEÑALES EN RUIDO GAUSSIANO ........................................ 39
1.5 PROBABILIDAD DE ERROR ........................................ 40

PRÁCTICA 5: MODULACIONES DIGITALES ........................................ 41

**1. Importante: Antes de Empezar** ........................................ **41**
1.1 Ficheros MDL y MAT ........................................ 41
1.2 Estructura del modelo PSK.MDL ........................................ 41
1.3 Representación del Espectro de las Modulaciones Digitales ........................................ 42
1.4 Cálculo de la Probabilidad de Error ........................................ 45
**2. Estudio de las Modulaciones Digitales.** ........................................ **43**
2.1 Estudio de la Modulación PSK ........................................ 43
2.2 Estudio de la Modulación BFSK ........................................ 43
2.3 Estudio de la Modulación QAM ........................................ 44
**3. Evaluación del error en las Modulaciones Digitales.** ........................................ **44**

## PRÓLOGO

### Introducción a SIMULINK

# Importante

Todos los modelos usados a lo largo de estas prácticas requieren de la versión de SIMULINK implementada sobre **Matlab 6.5**.

Su uso en otras versiones de Matlab puede dar lugar a errores importantes de funcionamiento.

# Introducción a SIMULINK

## 1. ¿Qué es SIMULINK?

SIMULINK® es un paquete informático ejecutable sobre MATLAB® para modelar, simular y analizar sistemas dinámicos. Su entorno de modelado gráfico usa los familiares diagramas de bloques, de forma que los sistemas objeto de las prácticas 4 y 5 del Laboratorio de Señales y Sistemas se pueden implementar fácilmente en SIMULINK. Soporta sistemas lineales y no lineales, modelados en tiempo continuo, muestreados o un híbrido de los dos. Por otro lado, la simulación es interactiva, es decir, se pueden cambiar los parámetros e inmediatamente ver lo que sucede. Las herramientas de análisis incluyen aquellas incorporadas en SIMULINK, más la gran variedad de que dispone MATLAB y sus *toolboxes* de aplicación.

SIMULINK proporciona una interfaz de usuario gráfica (*Graphic User Interface*, GUI) para construir los modelos como diagramas de bloques, utilizando operaciones con el ratón del tipo pulsar y arrastrar. Con esta interfaz se pueden dibujar los modelos como se haría con lápiz y papel (o como lo representan la mayoría de libros de texto de sistemas de comunicaciones). En SIMULINK hay dos tareas bien diferenciadas:

1) Definición del modelo: Para ello, SIMULINK usa un nuevo tipo de ventana Windows® llamada *block diagram*. En ella los modelos son creados y modificados con la ayuda del ratón.
2) Análisis por simulación: después de definir un modelo puede simularlo diseñando los parámetros de control que tiene a su disposición SIMULINK o introduciendo órdenes desde la ventana de órdenes de MATLAB.

## 2. Acceso a SIMULINK (Matlab 6.5)

Para comenzar a trabajar con SIMULINK tan solo debemos pulsar una vez con el botón izquierdo del ratón en el icono correspondiente de la barra de herramientas de MATLAB (ver figura 1).


Figura 1: Ventana de comandos de MATLAB 6.5 e icono de acceso a SIMULINK.

A continuación aparecerá la ventana principal de SIMULINK, llamada *Simulink Library Browser* en donde se encuentran todos los bloques y librerías que se necesitarán para diseñar los modelos (ver figura 2).

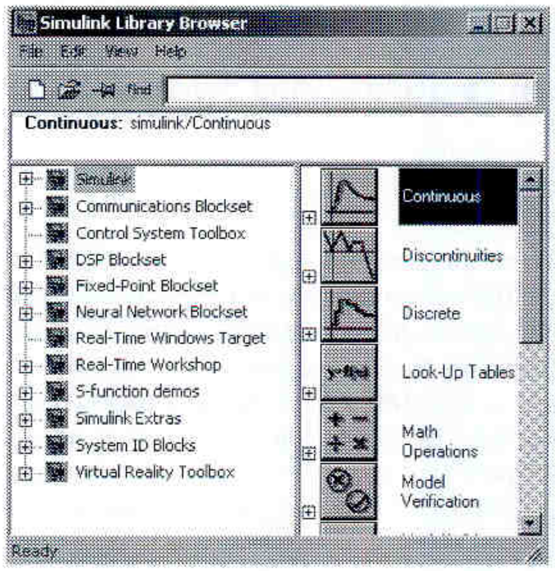



Figura 2: Ventana principal de SIMULINK.

La carpeta **Simulink** contiene los bloques básicos mientras que el resto de carpetas corresponden a ciertas *toolboxes* de MATLAB. Algunas de ellas se describen a continuación:

- **Communications Blockset**: Librería basada en la *Communications Toolbox* de MATLAB que contiene múltiples bloques necesarios para diseñar cualquier sistema complejo de comunicaciones, tanto analógico como digital.
- **DSP Blockset**: Herramientas para el procesamiento de señales, equivalentes a la *(Digital) Signal Processing Toolbox* de MATLAB, pero cuya visualización facilita el diseño de sistemas complejos.
- **Fixed-Point Blockset**: Herramientas para el procesamiento de señales como si se implementaran en un DSP de coma fija. Es muy interesante para analizar los efectos de precisión finita en el diseño de sistemas DSP.
- **Simulink Extras**: Contiene algunos bloques avanzados de visualización, etc.

### 2.1 Menús de la Ventana de Trabajo de SIMULINK

Para crear una nueva ventana de trabajo pinche una vez sobre el icono  de la barra de herramientas. SIMULINK crea una nueva ventana titulada *untitled* con una barra de menús como la de la figura 3.

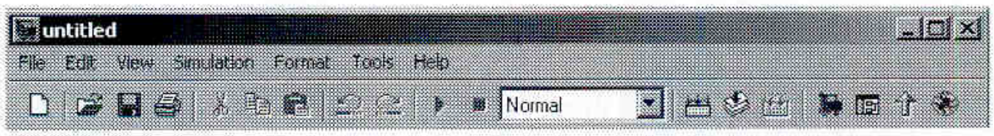



Figura 3. Barra de comandos de la ventana de trabajo de SIMULINK.

A continuación, detallamos los más importantes, para más información consultar la ayuda de SIMULINK (guía de usuario y guía de referencia) que se encuentra en los ficheros *sl_using.pdf* y *slref.pdf* dentro del directorio \\FALLES\APLICS\ASIG\3\Lab.SyS\help_simulink.

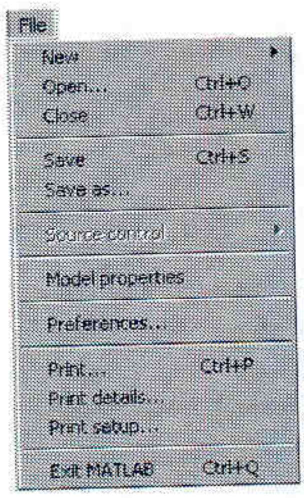



Figura 4. Menú File.

- Menú **File**: Incluye las siguientes opciones (a algunas de ellas se puede acceder como icono en la barra de herramientas de la ventana activa):
  - **New, Open, Close**: Comandos para abrir una nueva ventana de trabajo (*New Model*), un diagrama de bloques ya existente (extensión MDL), o bien, cierra la actual ventana de trabajo.
  - **Save, Save as**: Permite salvar el contenido de la ventana de trabajo con el nombre actual o bien con un nombre distinto.
  - **Exit MATLAB**: Cierra todas las ventanas de MATLAB y SIMULINK.

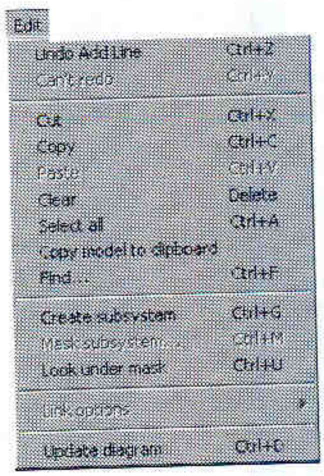



Figura 5. Menú Edit.

- Menú **Edit**: Incluye las siguientes opciones:
  - **Undo, Redo**: Comandos para deshacer y rehacer las acciones oportunas.
  - **Cut, Copy, Paste, Clear**: Corta, copia, pega o borra el bloque o diagrama de bloques seleccionado.
  - **Select All**: Selecciona todos los elementos de la ventana de trabajo.
  - **Copy model to clipboard**: Copia todos los elementos de la ventana de trabajo en el portapapeles de Windows.

- **Create Subsystem, Mask Subsystem**: Activo sólo cuando se selecciona un subsistema de bloques de la ventana activa. Comandos para crear y enmascarar un subsistema, es decir, para agrupar varios elementos en un único bloque de forma que sea más sencillo usarlo e identificarlo.
- **Look Under Mask**: Activo sólo cuando se selecciona un bloque. Abre una nueva ventana donde aparecen los elementos sencillos de los que está formado el bloque complejo. Sirve tanto para máscaras creadas por el usuario como para bloques ya definidos en las librerías de SIMULINK.
- **Link options**: Cuando se copia un bloque a la ventana de trabajo, se crea un enlace (*link*) con el bloque definido en la correspondiente librería. Estos comandos permiten romper dicho enlace o desbloquear una librería activa para poder modificar sus bloques.
- **Update Diagram**: Actualiza las características del modelo después de que se haya realizado algún cambio en los parámetros, bloques, conexiones, etc.

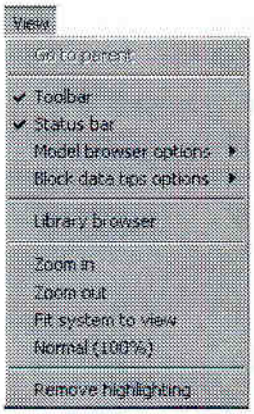



Figura 6. Menú View.

➢ Menú **View**: Incluye las siguientes opciones:

- **Toolbar, Status Bar**: Si se seleccionan (recomendado) se visualizan la barra de herramientas y la barra de estado de la simulación.
- **Library Browser**: Muestra la ventana principal de SIMULINK, la misma que se accede mediante el icono de la barra de herramientas.
- **Zoom In, Zoom Out, Fit Selection to View, Normal (100%)**: Diversas posibilidades para escalar la ventana de trabajo al tamaño deseado.

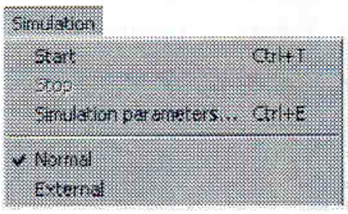



Figura 7. Menú Simulation.

- Menú **Simulation**: Incluye las siguientes opciones:
  - **Start/Stop**: Comienza una simulación, o la detiene si ya estaba iniciada. Se puede acceder a estos mismos comandos mediante los iconos  de la barra de herramientas de la ventana de trabajo.
  - **Simulation parameters**: Abre una nueva ventana de diálogo como la de la figura 8, donde se establecen los parámetros de la simulación. De entre todos los ellos, el más interesante es el tiempo de ejecución de la simulación (*Simulation Time* en la pestaña *Solver*) que se mide en segundos. El resto de parámetros establecen la resolución matemática, el tipo de errores o *warnings* que pueden aparecer, etc.

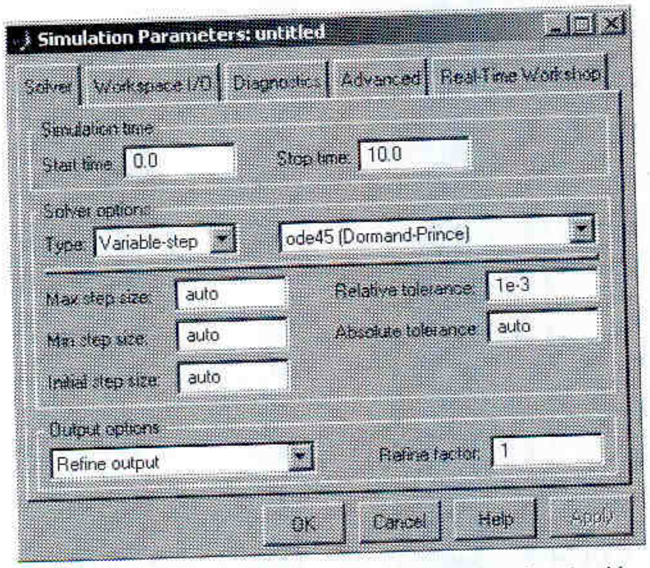



Figura 8. Ventana de parámetros de la simulación.

- Menú **Format**: Este menú permite definir el tipo de letra, mostrar u ocultar los nombres de los bloques, girarlos 90 o 180 grados, cambiar el color del fondo, del bloque, de la pantalla, definir la anchura de las conexiones, etc.
- Menú **Tools**: Accede a herramientas avanzadas de Simulink. No se usará en estas prácticas.

## 2.2 Librerías de SIMULINK

Una vez vistas las opciones más importantes del menú de la ventana activa de SIMULINK, pasemos a describir brevemente las librerías propias de que dispone. Las librerías son principalmente una agrupación coherente de bloques que luego utilizaremos para crear un modelo. Para acceder a ellas, sólo hay que pinchar dos veces con el ratón en la carpeta **Simulink** de la ventana principal (*Simulink Library Browser*) y aparecerán desplegadas las siguientes librerías (ver figura 9)[1]:

- **Continuous**: Contiene bloques que describen funciones lineales (integración, derivación, retardos, etc.)

[1] Se puede encontrar la descripción somera de cada bloque en las páginas 1-2 a 1-18 del documento *sl_using.pdf*.

- **Discrete**: Contiene los bloques con funciones similares al anterior, pero en el dominio discreto (integración, retardos, muestreadores, filtros, etc.)
- **Math**: Contiene bloques que realizan ciertas operaciones matemáticas elementales, como por ejemplo: módulo, multiplicación, suma, producto escalar, operadores lógicos, operaciones con complejos, etc.
- **Signal Routing**: Contiene los bloques que permiten multiplexar y demultiplexar señales.

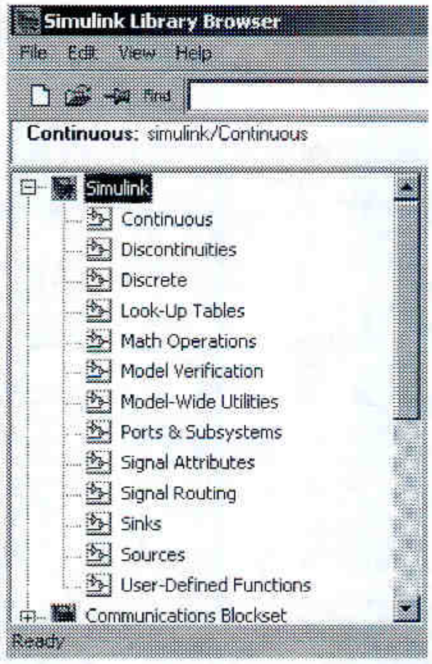



Figura 9. Contenido de la librería SIMULINK.

- **Sinks**: Proporciona sumideros para la salida de los bloques. Dichos sumideros pueden ser únicamente visualizadores como el osciloscopio (*scope*), o bien nos permiten salvar la señal al *workspace* de MATLAB (*To Workspace*) o a un fichero (*To File*).
- **Sources**: Contiene los bloques que generan señales, entre las que cabe destacar la señal escalón, una constante (o componente continua), ruido blanco, variables aleatorias, generador de señales, etc. Asimismo, permite leer datos del *workspace* de MATLAB (*From Workspace*) , o de un fichero (*From File*).

## 3. Construcción de un Modelo Sencillo

Vamos a dar los primeros pasos de construcción de un modelo a fin de empezar a familiarizarnos con ciertos procedimientos, comandos y bloques que nos serán muy útiles en SIMULINK.

Para crear un nuevo modelo pinche sobre el icono  de la ventana principal de SIMULINK. Mueva la ventana lo que necesite a fin de poder visualizar los contenidos de dicha ventana y de la que contiene las librerías (*Simulink Library Browser*) simultáneamente.

Abra la librería **Sources** haciendo doble click sobre ella. Se desplegarán todos los bloques que la forman. Fíjese que al hacer click sobre un bloque determinado, en la parte superior de la ventana principal aparece una breve descripción de dicho bloque, tal y como se muestra en la figura 10.

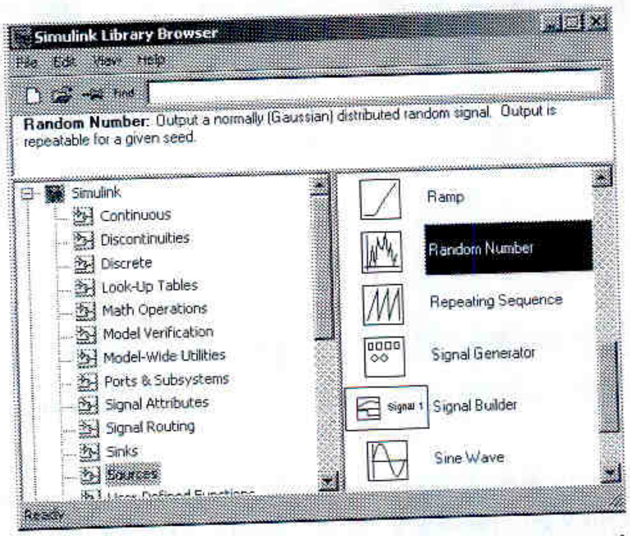



Figura 10. Selección de un bloque (en nuestro caso *Random Number*). En la parte superior aparece una breve descripción del bloque.

Añada bloques a su modelo pinchándolos con el ratón y arrastrándolos hasta la ventana de trabajo. En nuestro caso vamos a seleccionar el generador de funciones (*signal generator*). Colóquelo en la ventana de trabajo y haga doble click sobre él. Aparece entonces una ventana de diálogo que permite especificar los parámetros del bloque, así como ayuda sobre el mismo (ver figura 11). Especifique una sinusoide de 1 voltio de amplitud y 1 Hz de frecuencia.

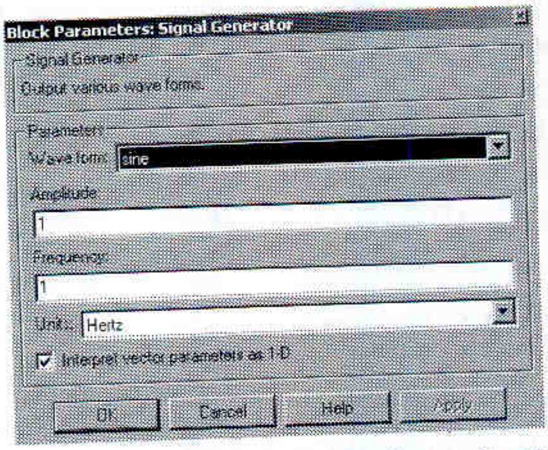



Figura 11. Ventana de diálogo para establecer los parámetros del generador de señales.

A continuación copie el bloque *scope* de la librería **Sinks** a la ventana de trabajo, a la derecha del bloque *signal generator*.

Ahora conecte estos bloques. Para ello, coloque el puntero sobre el puerto de salida del bloque *signal generator* (cuando tiene posibilidad de generar una conexión el puntero cambia la forma de flecha a cruz). Mantenga presionado el botón del ratón, desplace el puntero al puerto de entrada del bloque *scope* y suelte. La línea de conexión es siempre una flecha para mostrar la dirección del flujo de señal. El modelo debería parecerse al de la figura 12.

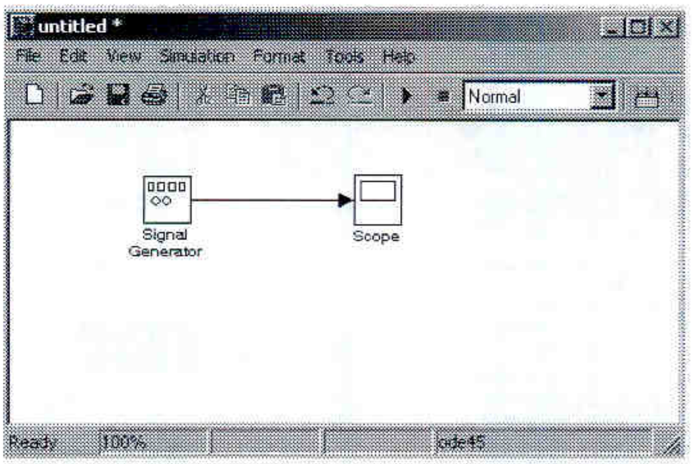



Figura 12. Modelo sencillo compuesto por un generador de señales y un osciloscopio.

Ahora puede comenzar la simulación. Haga doble click sobre el bloque *scope* para visualizar el osciloscopio y en la barra de menús escoja **Simulation → Start**, o bien haga click en el icono  de la barra de herramientas.

Compruebe que la señal observada en el osciloscopio corresponde a la elegida.

En los parámetros de la simulación (**Simulation → Parameters**) se puede ejecutar el modelo como si fuera discreto cambiando ciertos parámetros. Por ejemplo, establezca los siguientes valores en la ventana de parámetros de la simulación:

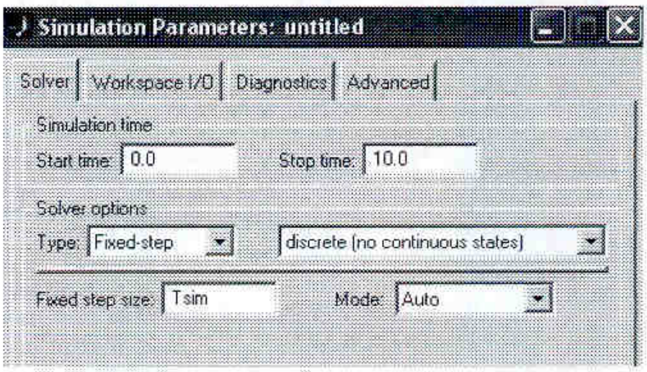



Figura 13. Parámetros de una simulación discreta

Si ahora comienza a ejecutar la simulación aparecerá la pantalla de la herramienta de diagnóstico de SIMULINK (figura 14). Esta herramienta nos indica algún error en el modelo, generalmente se trata de variables no definidas, conexiones erróneas, etc. En este caso, el valor del paso de resolución (*Fixed step size*) que hemos introducido "Tsim" no está previamente definido, por lo que en la ventana de comandos de MATLAB debe escribir:

>> Tsim=0.01;

Si ahora vuelve a ejecutar la simulación no debería tener ningún problema.

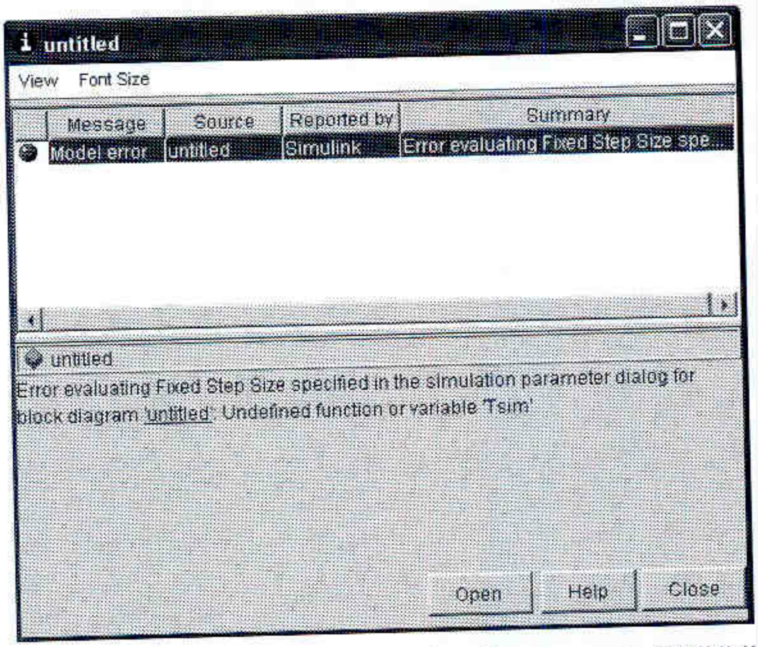



Figura 14. Herramienta de diagnóstico de errores de SIMULINK

### 3.1 Bloque *Scope*

Dado que el osciloscopio va a ser uno de los bloques más utilizados a lo largo de las prácticas siguientes, conviene detallar los comandos de su barra de herramientas, así como sus parámetros de funcionamiento.

-  Realiza un zoom sobre la pantalla (en las dos dimensiones, sólo en el eje X ó sólo en el eje Y, respectivamente).
-  Auto escala los ejes a la curva dibujada.
-  Salva el escalado actual de los ejes.
-  Abre una ventana de diálogo para establecer las propiedades del bloque. La más interesante es *Number of axes*, que permite crear varias gráficas en el mismo osciloscopio.
-  Imprime la ventana gráfica.

El escalado de los ejes también es posible haciendo click sobre ellos con el botón derecho del ratón.

Para cualquier duda que se le presente recuerde que puede consultar al profesor, o bien los ficheros de ayuda en formato PDF.

# PRÁCTICA 4

## Modelos Sencillos. Modulaciones Analógicas

En esta práctica vamos a implementar algunos modelos sencillos que se utilizan en comunicaciones, tanto analógicas como digitales. Se trata de diagramas de bloques elementales que van a permitir familiarizarnos con la configuración de los distintos bloques y con los valores más comunes de los parámetros. Se recomienda al alumno que utilice estos modelos sencillos para explorar los comandos que vienen explicados en el capítulo anterior, a fin de adquirir experiencia sobre el manejo de los iconos de la barra de herramientas, de los comandos del ratón, formas de conectar los bloques, etc.

En esta práctica simularemos también diferentes tipos de modulaciones analógicas, podremos visualizar la señal modulada en el dominio temporal y en el dominio frecuencial, así como variar los diferentes parámetros que definen una determinada modulación. También podremos apreciar la protección frente al ruido que presenta cada una de ellas.

# Importante

Antes de empezar la práctica 4 se pide realizar un **estudio teórico** de los diagramas de bloques y de las prestaciones de las modulaciones lineales y angulares.

Sus resultados teóricos deberán ser verificados en los modelos de SIMULINK. Por tanto, **el alumno deberá enseñar dicho estudio teórico al profesor antes de comenzar la práctica**.

# Estudio Teórico de las Modulaciones Analógicas

## 1. Modulaciones Analógicas

El hecho de modular una señal situada en la banda base del espectro de frecuencias mediante otra señal de mayor frecuencia llamada portadora se debe a razones puramente prácticas. A saber, las ventajas que se obtienen de una modulación son las siguientes:

1. La señal en banda base está muy expuesta a la acción de las interferencias (motores, red eléctrica, etc.)
2. Los elementos radiantes (antenas) para transmitir frecuencias bajas son de dimensiones impracticables.
3. Las características de propagación de las señales en banda base son pésimas, por lo que se precisa una potencia de transmisión muy grande para conseguir una calidad aceptable en el receptor.
4. El espectro se desaprovecharía para frecuencias altas, mientras que se colapsaría a frecuencias bajas. Con las modulaciones se pueden aprovechar todas las bandas del espectro.

En todas las modulaciones consideraremos como señal moduladora $x(t)$, una señal banda base normalizada, $|x(t)|_{max}=1$, de ancho de banda $W$ Hz y potencia $S_x$. La señal portadora será una sinusoide de amplitud $A_c$ voltios y frecuencia $f_c$ Hz, tal que $c(t)=A_c cos(\omega_c t)$.

### 1.1 Modulación en Amplitud (AM)

La expresión de una señal modulada en amplitud viene dada por:

$$y(t) = A_c[1 + mx(t)]\cos(\omega_c t)$$

donde $m$ es el llamado **índice de modulación** y debe cumplir $0<m<1$.

El espectro de la señal AM está centrado en la frecuencia de portadora y su ancho de banda de transmisión es $\boldsymbol{B_T=2W}$ Hz. La expresión exacta de dicho espectro es la siguiente:

$$Y(f) = \frac{A_c}{2}[\delta(f - f_c) + \delta(f + f_c)] + \frac{A_c m}{2}[X(f - f_c) + X(f + f_c)]$$

donde se aprecia un término correspondiente a la portadora (las deltas en $f_c$) y otro correspondiente a la información.

La **potencia transmitida** se puede calcular fácilmente como:

$$P_T = \frac{1}{T}\int_{-T/2}^{T/2} y^2(t)dt = \frac{A_c^2}{2}[1 + m^2 S_x]$$

en donde se vuelve apreciar el término de portadora y el correspondiente a la información.

Teniendo en cuenta el rango limitado del índice de modulación ($0<m<1$) y que la potencia de la señal banda base nunca será mayor que 1 ($S_x<=1$), se puede observar como más de la mitad de la potencia transmitida se gasta en la portadora que no lleva información alguna. Por este motivo, la AM se considera una modulación poco eficiente en potencia.

La principal ventaja de esta modulación es, sin embargo, la posibilidad de utilizar un receptor no coherente implementado mediante un detector de envolvente.

## 1.2 Modulación en Doble Banda Lateral (DBL)

La expresión de una señal modulada en doble banda lateral viene dada por:

$$y(t) = A_c x(t)\cos(\omega_c t)$$

Esta modulación también lleva el nombre de Modulación en Amplitud con Portadora Suprimida, ya que el término de portadora sola ya no aparece. A cambio, la envolvente de esta modulación ya no es proporcional a la señal banda base *x(t)* por lo que el detector utilizado cambiará.

El espectro de la señal DBL también está centrado en la frecuencia de portadora y su ancho de banda de transmisión es también $B_T=2W$ Hz. La expresión exacta de dicho espectro viene dada por:

$$Y(f) = \frac{A_c}{2}\left[X(f-f_c)+X(f+f_c)\right]$$

La **potencia transmitida** es tal que

$$P_T = \frac{1}{T}\int_{-T/2}^{T/2} y^2(t)dt = \frac{A_c^2}{2}S_x$$

Se dice que la potencia transmitida se debe a las dos bandas laterales de *x(t)*. Las bandas laterales son de igual potencia y con simetría hermítica[2] respecto a la frecuencia de portadora, debido a que la señal banda base es real. La banda lateral superior (BLs)es aquella comprendida entre $f_c$ y $f_c+W$, mientras que la banda lateral inferior (BLi) cubre la banda $[f_c-W, f_c]$.

## 1.3 Modulación en Banda Lateral Única (BLU)

La expresión de una señal modulada en banda lateral única viene dada por:

$$y(t) = \frac{A_c}{2}\left[x(t)\cos(\omega_c t) \mp \hat{x}(t)\,\mathrm{sen}(\omega_c t)\right]$$

donde $\hat{x}(t)$ es la transformada de Hilbert de *x(t)* y el signo menos (-) ó más (+) define la modulación en BLU superior o inferior, respectivamente.

El espectro de la señal BLU, como su nombre indica, corresponde al de una sola banda de la señal de información *x(t)*, por lo que el ancho de banda de transmisión será el mínimo posible: $B_T=W$ Hz. La expresión de su espectro es un poco complicada:

$$Y(f) = \frac{A_c}{4}\left[\left(1 \pm \mathrm{sgn}(f-f_c)\right)X(f-f_c) + \left(1 \mp \mathrm{sgn}(f+f_c)\right)X(f+f_c)\right]$$

donde sgn(*f*) es la función signo.

La **potencia transmitida** es tal que

[2] Una función compleja tiene simetría hermítica si su módulo es par y su fase impar respecto al origen de coordenadas.

$$P_T = \frac{1}{T}\int_{-T/2}^{T/2} y^2(t)dt = \frac{A_c^2}{4}S_x$$

## 1.4 Modulación en Frecuencia (FM) y en Fase (PM)

La expresión de una señal modulada en frecuencia o en fase responde a una misma expresión genérica, ya que ambas transportan la información en la denominada *Desviación de Fase Instantánea* $\varphi(t)$:

$$y(t) = A_c \cos(\omega_c t + \varphi(t))$$

La expresión particular para cada modulación se expresa según la siguiente tabla donde se especifican la *desviación de fase instantánea* y la *frecuencia instantánea* para cada una de ellas:

fase instantanea

| | ***Desviación de Fase Instantánea*** $\psi(t) = \omega_c t + \varphi(t)$ | ***Frecuencia Instantánea*** $f(t)$ |
|---|---|---|
| PM | $\omega_c t + \phi_d x(t)$ | $f_c + \frac{1}{2\pi}\phi_d \frac{dx(t)}{dt}$ |
| FM | $2\pi f_c t + 2\pi f_d \int_{-\infty}^{t} x(\lambda)d\lambda$ | $f_c + f_d x(t)$ |

Donde $\phi_d$ (rads/volt) es la ***sensibilidad de fase*** en una modulación PM y $f_d$ (Hz/volt) es la ***sensibilidad de frecuencia*** de una modulación FM.

El espectro de una modulación FM y PM solo es posible analizarlo con señales moduladoras sencillas, como un tono, un tren de pulsos, o una suma de tonos, por ejemplo. Para el caso de modulación con un tono de amplitud $A_m$ y frecuencia $f_m$, la expresión resultante en el tiempo es:

$$y(t) = A_c \sum_{n=-\infty}^{\infty} J_n(\beta)\cos(\omega_c t + n\omega_m t)$$

donde $J_n(\beta)$ es la función de Bessel de primera especie de orden $n$ y argumento $\beta$.

Dicho parámetro $\beta$ se denomina **índice de modulación** y su función consiste principalmente en controlar el ancho de banda de la señal modulada, como veremos a continuación. Su expresión analítica es la siguiente:

- FM: $\beta = \frac{A_m f_d}{f_m} = \frac{\Delta f}{f_m}$, donde $\Delta f$ se denomina **máxima desviación de frecuencia**
- PM: $\beta = A_m \phi_d$

Como se puede apreciar de la expresión de $y(t)$, el espectro de una señal FM o PM modulada por un tono contiene infinitos tonos a frecuencias separadas de la portadora múltiplos de $f_m$ Hz, cada uno de ellos con amplitud distinta dada por el correspondiente valor de la función de Bessel $J_n(\beta)$.

En la práctica, las propiedades de dichas funciones de Bessel hacen que el espectro sea simétrico alrededor de $f_c$ y conste de un número finito de componentes significativas, de forma que el ancho de banda de transmisión se puede calcular mediante la llamada **regla de Carson**: $B_T=2(\beta+1)f_m$ Hz.

Para una señal arbitraria $x(t)$ se utiliza la misma regla, pero sustituyendo el índice de modulación por la llamada **relación de desviación** $D=\Delta f/W$, y la frecuencia del tono por el ancho de banda de $x(t)$, de tal forma que el **ancho de banda de transmisión** se calcula como: $B_T=2(D+1)W$ Hz.

Las expresiones anteriores evidencian que el ancho de banda viene gobernado por el índice de modulación, o la relación de desviación en su caso, por lo que podemos distinguir dos tipos de modulaciones FM:

- FM de banda estrecha: cuando $\beta << 1$ (en la práctica < 0.4)
- FM de banda ancha: cuando $\beta >> 1$ (en la práctica > 3)

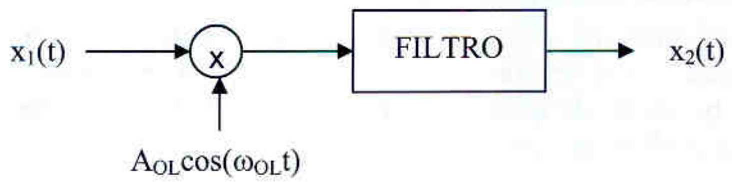
Para ambas modulaciones angulares, FM y PM, la **potencia transmitida** es independiente de la señal banda base y es igual a la de la portadora sola:

$$P_T = \frac{1}{T}\int_{-T/2}^{T/2} y^2(t)dt = \frac{A_c^2}{2}$$

**1.4.a Obtenga el espectro de cada una de las modulaciones anteriormente explicadas si la señal moduladora es un tono de amplitud $A_m$ y frecuencia $f_m$.**

### 1.5 Moduladores y Demoduladores

Uno de los circuitos más utilizados en cualquier modulación es el denominado **mezclador** o **convertidor de frecuencia**. Su diagrama de bloques es el siguiente:




Se puede demostrar que si $x_1(t)$ tiene un espectro situado sobre la frecuencia $f_1$, al multiplicarse por la salida del oscilador local, se genera una nueva señal a frecuencias $f_1+f_{LO}$ y $f_1-f_{LO}$. El filtro dejará pasar únicamente una de las dos componentes por lo que la señal $x_2(t)$ tiene las mismas componentes espectrales que $x_1(t)$ pero en otra banda de frecuencias. De aquí se deduce que la mezcla o conversión en frecuencia es una operación básica para modular y demodular modulaciones lineales.

El mismo diagrama de bloques del mezclador se usa para detectar cualquier modulación lineal sin más que obligar a que la frecuencia del oscilador local sea igual a $f_c$ y el filtro sea paso bajo. Si además aseguramos que la fase de la portadora y la del oscilador local son iguales (están sincronizadas), entonces el convertidor de frecuencia pasa a llamarse **detector síncrono**.

**1.5.a Halle la expresión de la señal de salida de un detector síncrono si la entrada es $y(t)=[K_1+K_2x(t)]\cos(\omega_c t)-K_3\hat{x}(t)\,\mathrm{sen}(\omega_c t)$. Particularice los valores de las constantes $K_1$, $K_2$ y $K_3$ para las modulaciones lineales estudiadas.**

Una modulación AM se suele demodular, sin embargo, con un **detector de envolvente**, que como su nombre indica, obtiene una señal proporcional a la envolvente de la sinusoide que le entra. La ventaja frente al detector síncrono es que no necesita recuperar el sincronismo de la portadora para su buen funcionamiento.

Por último, comentaremos que la modulación FM se puede implementar mediante dos formas, según el **método indirecto** y según el **método directo**. Este último utiliza un dispositivo denominado *Oscilador Controlado por Tensión* (VCO en sus iniciales inglesas), cuya desviación de frecuencia de salida es proporcional a la tensión de entrada. El método indirecto hace uso de un modulador de fase de banda estrecha cuya entrada es la integral de la señal banda base, resultando por tanto una señal FM de banda estrecha a su salida. Mediante mezcladores y multiplicadores de frecuencia, se puede hacer que dicha señal FM de banda estrecha se convierta en una FM de banda ancha con la frecuencia de portadora y máxima desviación de frecuencia deseadas.

### 1.6 Relación señal-a-ruido

Una parte importante del análisis de las modulaciones analógicas es comprobar sus prestaciones frente a una transmisión por un canal ruidoso. Dicho canal simplemente atenúa la señal modulada y le añade un ruido blanco gaussiano de densidad espectral de potencia $N_0/2$. El diagrama de bloques del receptor sería el siguiente:

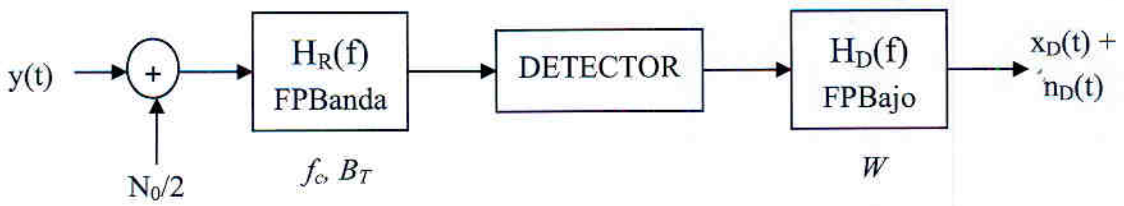



Figura 15. Diagrama de bloques genérico de un receptor analógico.

El detector varía según la modulación, así como el filtro paso banda de predetección $H_R(f)$ que siempre está situado sobre la banda de paso de la señal modulada.

El parámetro de comparación entre modulaciones es la denominada **relación señal-a-ruido de postdetección** $(S/N)_D$ definida como el cociente entre la potencia de la señal detectada $x_D(t)$ y la potencia de ruido de salida $n_D(t)$, expresada en dB.

A continuación, se dan las expresiones de la relación señal-a-ruido de postdetección para las modulaciones estudiadas, pero se recomienda al alumno que repase en el tema correspondiente de la asignatura *Teoría de la Comunicación* cómo se obtienen cada una de ellas:

- **AM:** $\left(S/N\right)_D = \dfrac{m^2 S_x}{1+m^2 S_x}\gamma$
- **DBL:** $\left(S/N\right)_D = \gamma$
- **BLU:** $\left(S/N\right)_D = \gamma$
- **PM:** $\left(S/N\right)_D = \phi_d^2 S_x \gamma$

- **FM:** $\left(S/N\right)_D = 3D^2 S_x \gamma$

donde $\gamma$ es un parámetro de comparación denominado **relación señal-a-ruido en banda base**, cuya expresión viene dada por $\gamma = \dfrac{S_R}{N_0 W}$, donde $S_R$ es la potencia de señal recibida a la entrada del receptor. En nuestro caso, supondremos que la atenuación del canal es nula, por lo que la potencia recibida coincide con la potencia transmitida $P_T$.

| |
|---|
| **1.6.a Calcule la relación señal-a-ruido de postdetección de cada una de las modulaciones anteriores cuando la señal moduladora es un tono de amplitud $A_m$ y frecuencia $f_m$.** |

# Práctica 4: Modelos Sencillos. Modulaciones Analógicas

## 1. Importante: Antes de Empezar

Para esta práctica se proporcionan varios ficheros de extensión **MDL** además de otro fichero de variables llamado **CONFIG4.MAT**. Estos ficheros, junto con los que visualizan el espectro, se deben situar en el directorio de trabajo.

**Antes de empezar a realizar las prácticas deberá cargar el fichero CONFIG4.MAT en el *Workspace* de MATLAB ejecutando en la ventana de comandos**:

```
>> load config4
```

Conviene que identifique en cada simulación qué variables del *workspace* está usando y cuáles son sus valores correspondientes:

- Tsim (*Periodo de muestreo de la simulación*): Interviene en la mayoría de los bloques, especialmente para los filtros. Recuerde que la frecuencia de muestreo de la simulación, fsim=1/Tsim, debe ser al menos el doble de la máxima frecuencia de las señales utilizadas.
- fc (Frecuencia de portadora): Da el valor de la frecuencia de la señal portadora en las modulaciones. Normalmente se utiliza en los bloques moduladores y demoduladores, al igual que las variables Ac y ph.
- Ac (Amplitud de portadora).
- ph (Fase inicial de la portadora).
- fm (Frecuencia de la señal moduladora): Contiene el valor de la frecuencia utilizada por la señal moduladora, en el caso de que sea una señal periódica (sinusoide, de pulsos, etc.).
- BT (Ancho de banda): Es un valor que se suele usar en el diseño de los filtros en FM.
- No (DEP del ruido blanco): Usada en el bloque *AWGN channel.*

El hecho de utilizar variables del *workspace* para definir los parámetros de los bloques de la simulación permite que al cambiar el valor de la variable, automáticamente su valor se actualice en todos los bloques. Por ejemplo, el periodo de muestreo, Tsim, es un parámetro usado en todos los filtros, en el analizador de espectros, etc. Si en la ventana de comandos de MATLAB ejecutamos:

```
>> Tsim=2.5e-4;
```

al correr de nuevo la simulación actual, el valor de Tsim se actualiza automáticamente en todos los bloques que lo usan.

Por último, **es recomendable que siempre que modifique los bloques** de una simulación para realizar lo que le piden en las diferentes prácticas, **salve la simulación actual bien con su mismo nombre o con un nombre nuevo** (la extensión siempre es por defecto MDL).

## 2. Ruido filtrado

Abra el fichero RUIDO.MDL donde se implementa un esquema correspondiente al filtrado de un ruido blanco gaussiano (ver figura 16).

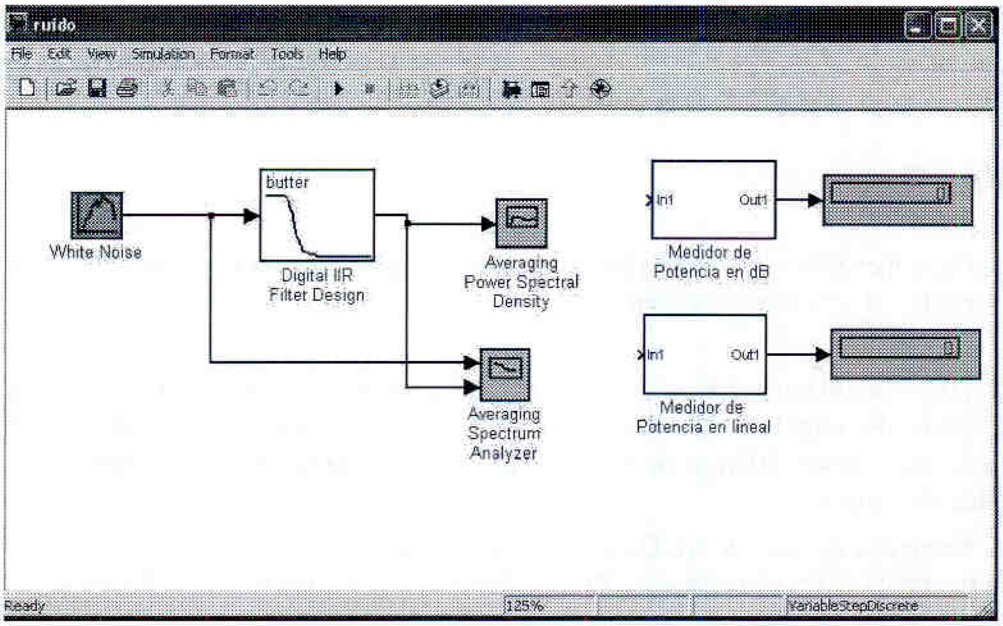



Figura 16. Diagrama de bloques del modelo RUIDO.MDL

El bloque etiquetado como *Digital IIR Filter Design* es un filtro paso-bajo. Podemos construir filtros en SIMULINK mediante la librería *Filter Designs* que se encuentra dentro de la carpeta **Filtering** de la **DSP Blockset**. En nuestras prácticas usaremos generalmente filtros IIR[3] discretos que están implementados en el bloque *Digital IIR Filter Design*. Los parámetros de dicho bloque son:

- *Design Method*: Permite elegir entre los siguientes métodos de diseño: Butterworth, Chebychev tipo I, Chebychev tipo II y elíptico.
- *Filter Type*: Especifica el tipo de filtro: paso bajo, paso banda, paso alto y de banda eliminada.
- *Filter order*: Orden del filtro. Generalmente tomaremos un número entre 4 y 8.
- *Passband edge frequency (0 to 1)*: Especifica la frecuencia de corte para los filtros paso bajo y paso alto. Debe ser un número entre 0 y 1 ya que está normalizado respecto a la mitad de la frecuencia de muestreo. Por tanto, al especificar la frecuencia deberemos poner la expresión: *frecuencia_de_corte**2*Tsim, siendo Tsim el periodo de muestreo de la simulación. En los filtros paso banda y de banda eliminada no aparece este parámetro sino las frecuencias que definen la banda por abajo (*Lower passband edge frequency (0 to 1)*) y por arriba (*Upper passband edge frequency*). $= \frac{f_c}{(f_s/2)}$

[3] IIR son las siglas de *Infinite Impulse Response*.

**Ejemplos**:

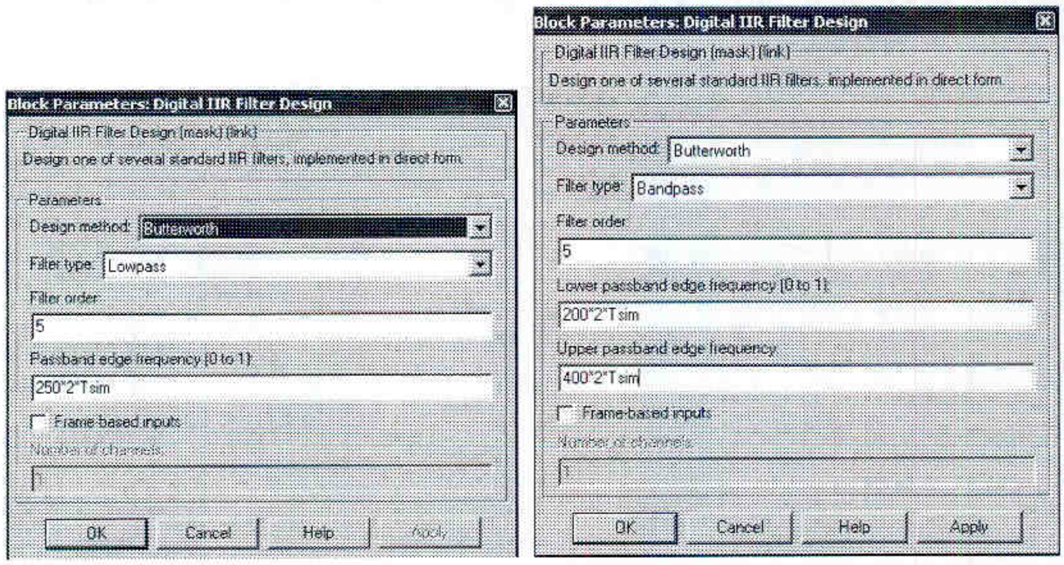



Figura 17. Diseño de un filtro paso bajo de frecuencia de corte 250 Hz y de un filtro paso-banda con banda de paso (200-400) Hz.

**NOTA:** Es importante recordar que el espectro que podemos filtrar debe estar incluido en el rango [0, fsim/2] Hz, siendo fsim=1/Tsim la frecuencia de muestreo de la simulación.


**2.1.a Ejecute la simulación y dibuje aproximadamente la función de transferencia del filtro que aparece en el bloque *Averaging Spectrum Analyzer* ¿corresponde a un filtro paso-bajo? ¿cuál es su frecuencia de corte?** $f_c$ = 500 Hz (500 · 2 · Tsim)

**2.1.b Obtenga la potencia del ruido blanco filtrado en las siguientes bandas: (0-500) Hz, (500-1000) Hz, (1000-1500) Hz. ¿Son valores similares? ¿Porqué?**

0-500: 0'2532
500-1000: 0'2425
1000-1500: 0'2305

es ruido blanco

## 3. Mezclador

Abra el fichero MEZCLADOR.MDL donde se implementa un diagrama de bloques de un convertidor de frecuencia o mezclador (figura 18). Coloque tantos osciloscopios (*scope*) y analizadores de espectro (*Averaging Power Spectral Density*) como sean necesarios para observar las señales a la salida de los distintos bloques.

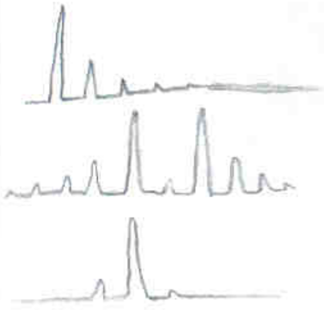
Antes de ejecutar la simulación especifique los valores siguientes en la ventana de comandos de Matlab:

- Frecuencia de portadora (**fc**): 1000 Hz.
- Fase inicial (**ph**): 0 rad.
- Frecuencia de la señal a la salida del Generador de Señales (**fm**): 200 Hz.
- Periodo de muestreo de la simulación (**Tsim**): 1/4000 s.

**3.1.a El bloque llamado *Conversor de Frec* realiza el producto de la señal de entrada por un coseno de frecuencia especificada en la casilla 'Frecuencia de portadora', ¿para qué es necesario entonces el filtro paso banda a la salida?**

para eliminar la delta simétrica que aparece alrededor de $f_c$

**3.1.b Dibuje de forma aproximada el espectro de la señal de salida cuando la entrada es un diente de sierra a la frecuencia $f_m$ en lugar de una sinusoide. ¿Qué frecuencias aparecen?**



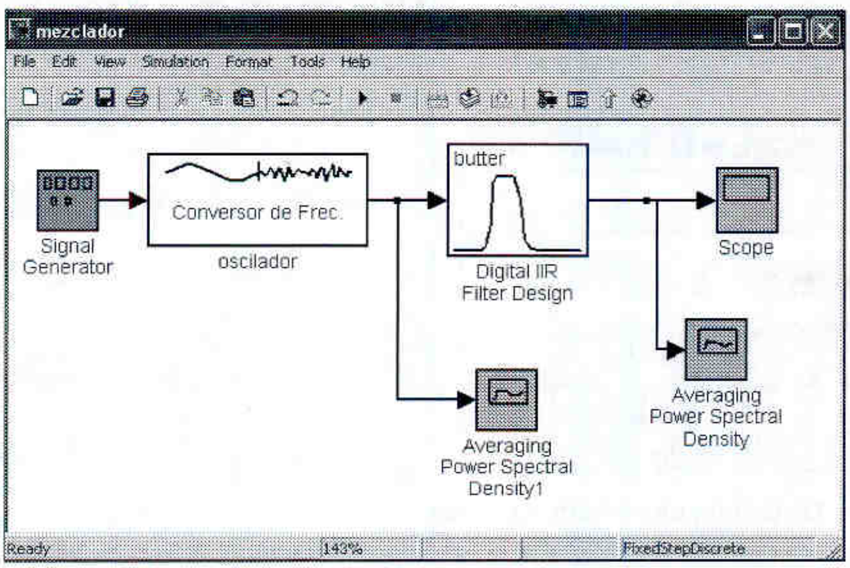



Figura 18. Diagrama de bloques del modelo MEZCLADOR.MDL

## 4. Lectura y Escritura de Datos

Abra el fichero ESCRITURA.MDL. En él se toma como base el diagrama de bloques del ruido blanco filtrado de la primera simulación para salvar en una variable de Matlab los datos correspondientes a un ruido filtrado en la banda de 0 a 500 Hz. Para ello se utiliza el bloque *To Workspace* de la librería **Sinks**. Para conocer las distintas opciones de este bloque consulte la ayuda y fíjese en las distintas estructuras de variable con las que puede operar.

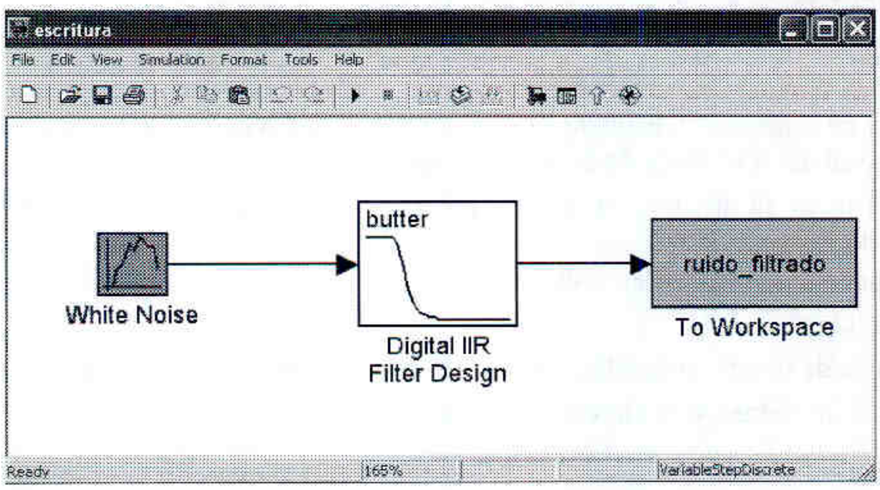



Figura 19. Diagrama de bloques del modelo ESCRITURA.MDL

## 5. Resolución en Tiempo y en Frecuencia

Como se ha podido apreciar en las simulaciones vistas hasta ahora, en simulink conviene especificar el periodo de muestreo Tsim con el que se trabaja. Este valor determina a su vez la resolución en frecuencia y tiempo que se puede obtener, o lo que es lo mismo, el detalle con el que se puede observar la forma de onda temporal y el espectro de las señales que intervienen.

Como regla práctica para la ejecución de **todas las simulaciones**, se recomienda usar estos valores (o de parecida magnitud) según el dominio en el que se quiera observar la señal:

**Visualización de la forma de onda temporal:**

```
>> Tsim=1/(15*fc);
```

La forma de onda se puede observar mediante el osciloscopio (bloque *Scope*), o bien salvando la señal a una variable del *workspace* (bloque *To Workspace*) y ejecutando en la ventana de comandos de Matlab la función dibuja.m:

```
>> dibuja(nombre_de_la_variable,Tsim);
```

**Visualización del espectro:**

```
>> Tsim=1/(4*fc);
```

El espectro se puede observar salvando la señal a una variable del *workspace* (bloque *To Workspace*) y ejecutando en la ventana de comandos de Matlab la función dibuja_tf_bi.m que dibuja el módulo del espectro bidimensional:

```
>> dibuja_tf_bi(nombre_de_la_variable,Tsim);
```

o bien la función dibuja_tf_bi_dB.m que dibuja el módulo del espectro bidimensional en unidades logarítmicas (dB) :

```
>> dibuja_tf_bi_dB(nombre_de_la_variable,Tsim);
```

# 6. Estudio Práctico de las Modulaciones Analógicas

## 6.1 Importante: Ficheros MDL

1. Se proporciona el fichero de SIMULINK llamado **AM.MDL**, donde se simula un sistema completo de comunicaciones con modulación en amplitud AM (en inglés *Double-SideBand Amplitude Modulation*, DSB AM) y canal de ruido blanco aditivo gaussiano (*AWGN channel*). En él se encuentran asimismo 2 osciloscopios (*Scope*) y un bloque *To workspace*.
2. Cuando tenga que estudiar las modulaciones DBL (en inglés *Double-SideBand Suppressed Carrier Amplitude Modulation*, DSBSC AM) y FM, deberá usar el fichero **AM.MDL**, salvarlo con otro nombre, y reemplazar el modulador y el demodulador por los de la respectiva modulación. Dichos bloques los puede encontrar en la *Communications Blockset* tal y como muestra la figura 20.
3. A continuación, deberá establecer los parámetros del modulador, del demodulador y de los filtros para que simule el sistema que se le pide en cada apartado. No olvide actualizar también el valor de Tsim.

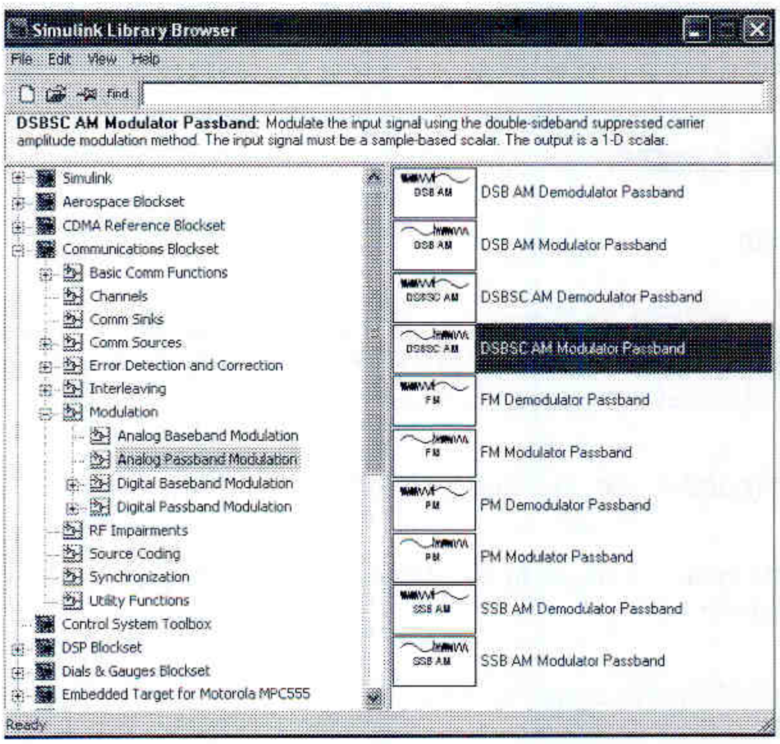


Figura 20. Moduladores y Demoduladores Paso-Banda Analógicos.

## 6.2 Modulación en Amplitud con Portadora (AM)

Especifique los valores de los parámetros necesarios para simular en el modulador (antes del canal) la siguiente modulación AM:

| | |
|---|---|
| **Señal Moduladora:** | $f_m$ = 100 Hz |
| | Forma de onda senoidal. |
| **Señal Portadora:** | $f_c$ = 1000 Hz |
| | $A_c$ = 1 V |
| | m = 0.5 |

Nota: el modulador DSB-AM de Simulink tiene un parámetro llamado **Input signal offset** que NO es el índice de modulación **m**, sino que equivale a **1/m**.

**6.2.a Rellene la siguiente tabla con la amplitud de las deltas del espectro de la señal modulada:**

| Frecuencia (Hz) | 900 | 1000 | 1100 |
|---|---|---|---|
| Amplitud aproximada (V/Hz) | 0'12 | 0'5 | 0'12 |

Rellene el siguiente cuadro con los resultados teóricos que debería obtener y compárelos con la amplitud medida. $A_c/2$ $A_m A_c m/4$ normalmente $A_m=1$

| Frecuencia (Hz) | 900 | 1000 | 1100 |
|---|---|---|---|
| Amplitud teórica (V/Hz) | 0'125 | 0'5 | 0'125 |

**6.2.b Repita la modulación anterior variando el índice de modulación m = 1.**

Rellene la siguiente tabla con la amplitud de las deltas del espectro de la señal modulada y compare los resultados obtenidos con los teóricos.

| Frecuencia (Hz) | 900 | 1000 | 1100 |
|---|---|---|---|
| Amplitud aproximada (V/Hz) | 0'23 | 0'5 | 0'23 |

Observe también la señal en el dominio temporal. ¿En qué se diferencia con la modulación anterior cuyo índice *m* era 0.5? En este, la envolvente llega hasta cero

**6.2.c Repita la modulación anterior variando el índice de modulación m=1.4 (sobremodulación). De nuevo observe la señal en el dominio temporal, y compárela con los resultados obtenidos en las modulaciones anteriores. ¿Porqué esta señal no puede ser recuperada por un detector de envolvente?** el detector de envolvente no puede demodular correctamente si hay sobremodulación.

**6.2.d Observe la señal modulada en el dominio temporal, realizando un zoom en el cruce por cero de la envolvente ¿Existe inversión de fase de la portadora?** Si

## 6.3 Modulación en Doble banda Lateral (DBL)

<u>Cambie ahora el modulador y el demodulador</u> y especifique los valores de los parámetros necesarios para simular la siguiente modulación DBL:

| | |
|---|---|
| **Señal Moduladora:** | $f_m$ = 100 Hz |
| | Forma de onda senoidal. |
| **Señal Portadora:** | $f_c$ = 1000 Hz |
| | $A_c$ = 1 V |

**6.3.a Rellene las siguientes tablas con la amplitud de las deltas del espectro de la señal modulada.**

| | | |
|---|---|---|
| Frecuencia (Hz) | 900 | 1100 |
| Amplitud aproximada (V/Hz) | 0'24 | 0'24 |

Rellene el siguiente cuadro con los resultados teóricos que debería obtener y compárelos con los resultados anteriores.

| | | |
|---|---|---|
| Frecuencia (Hz) | 900 | 1100 |
| Amplitud teórica (V/Hz) | 0'25 | 0'25 |

**6.3.b Observe la señal modulada en el dominio temporal, realizando un zoom en el cruce por cero de la envolvente ¿Existe inversión de fase de la portadora?**

### 6.4 Modulación en Frecuencia (FM)

Cambie ahora el modulador y el demodulador y especifique los valores de los parámetros necesarios para simular la siguiente modulación FM. Para la sensibilidad de frecuencia de momento no especificaremos ningún valor:

fm = 200;
fc = 4000;
Tsim = 1/(4*fc);

| | |
|---|---|
| **Señal Moduladora:** | Forma de onda senoidal. |
| | $f_m$ = 200 Hz |
| **Señal Portadora:** | $f_c$ = 4000 Hz |
| | $A_c$ = 1 V |

**6.4.a Manteniendo constantes los valores de la tabla anterior, realice tres modulaciones FM, cada una para un valor de $f_d$ diferente, y rellene la siguiente tabla. (Recuerde utilizar un periodo de muestreo Tsim=1/(4$f_c$) para poder visualizar todo el rango de frecuencias del espectro FM)**

$BW = 2(D+1)\cdot W = 2\left(\frac{fd \cdot Am}{fm}+1\right)\cdot fm$

NOTA: Para calcular el ancho de banda teórico, emplee la regla de Carson. Para medir el que aparece en la pantalla gráfica utilice la función **dibuja_tf_bi_dB.m** y considere todas las rayas espectrales que sea capaz de apreciar.

fd = 100;
fd = 400;
fd = 1000;

Ancho de banda teórico:

$\beta = \frac{fd \cdot Am}{fm}$ → 1 ~200

| **$f_d$: desviación de frecuencia (*modulation constant*)** | **100 Hz/V** | **400 Hz/V** | **1 KHz/V** |
|---|---|---|---|
| $\beta$: Indice de modulación | 1/2 | 2 | 5 |
| $\mathbf{B_{TT}}$ :Ancho de banda teórico | 600 Hz | 1200 Hz | 2400 Hz |
| $\mathbf{B_{TM}}$ :Ancho de banda medido | 1600 Hz | 2400 Hz | 4000 Hz |
| Relación $\mathbf{B_{TT}/B_{TM}}$ (en %) | 37'5 % | 50 % | 60 % |
| Separación entre rayas espectrales | 200 Hz | 200 Hz | 200 Hz |

**6.4.b Observe la diferencia entre los valores de ancho de banda teórico (según Carson) y medido, ¿a qué es debido? Antes de contestar visualice el espectro en lineal mediante dibuja_tf_bi.m.**

### 6.5 Evaluación de la relación Señal-a-Ruido de pre y post-detección

El bloque *AWGN channel* permite controlar la potencia de ruido que entra en el receptor mediante el parámetro de la varianza, *variance*. Esta varianza coincide con el doble de la DEP del ruido, $N_0$, pero sus unidades no corresponden con Watios/Hz reales, sino con Watios/Hz normalizados respecto de la frecuencia de muestreo de la simulación, fsim.

Por tanto, para calcular la potencia de ruido teórica a la salida del filtro de pre-detección, $P_N$, se debe usar la siguiente fórmula:

$$\boxed{P_N(Watios) = N_0 \cdot B_T \cdot 2T_{sim}}$$

donde $B_T$ es el ancho de banda del filtro en Hz, $T_{sim}$ es el periodo de muestreo de la simulación en segundos y $N_0$ es el valor de la varianza especificada en el bloque *AWGN channel*.

En este apartado evaluaremos las prestaciones en cuanto a relación señal-a-ruido de pre y postdetección de las tres modulaciones analizadas anteriormente (AM, DBL y FM). Para ello estableceremos los siguientes valores de los parámetros comunes a todas ellas:

| | |
|---|---|
| **Señal Moduladora:** | Forma de onda senoidal. |
| | $f_m$ = 200 Hz |
| **Señal Portadora:** | $f_c$ = 2500 Hz |
| | $A_c$ = 1 V |
| **DEP del Ruido:** | $N_0$ = 1 W/Hz normalizados |
| **Tsim:** | $1/(15 \cdot f_c)$ |

→ cuidado, N° hay que hacer 1 · 2 · Tsim

Y los siguientes parámetros específicos de AM y FM (de ésta última modulación haremos 2 pruebas, una con una FM de banda estrecha y otra de banda ancha):

| | |
|---|---|
| **AM:** | m=1 |
| **FM de b. estrecha:** | $f_d$= 70 Hz/V |
| **FM de b. ancha:** | $f_d$= 800 Hz/V |

**6.5.a Observe la señal demodulada a la salida de cada sistema y mida por separado la potencia de señal y la potencia debida al ruido (utilice para ello la función <u>VAR.M</u> en la ventana de comandos de Matlab). ¿Cuál de ellas presenta mejor relación señal-a-ruido? ¿Por qué?**

| | ruido predetección | ruido postdetecc. | señal predetección | señal post detección | |
|---|---|---|---|---|---|
| AM | 0'0415 | | 0'7495 | 0'4972 | |
| DBL | 0'0431 | 0'0428 | 0'2485 | 0'4879 | |
| $FM_{BE}$ | 0'0301 | 0'029 | 0'4910 | 0'0545 | ← cambiando BT |

# PRÁCTICA 5

## Modulaciones Digitales

En esta práctica simularemos diferentes tipos de modulaciones digitales, podremos visualizar la señal modulada en el dominio temporal y en el dominio frecuencial, así como variar los diferentes parámetros que definen una determinada modulación. También se podrán visualizar las diferentes constelaciones y apreciar los efectos del ruido sobre ellas.

Concretamente la práctica 5 estudia las modulaciones binarias en fase (PSK), frecuencia (FSK), y la modulación QAM de 4 niveles. Dicho estudio consiste básicamente en comprobar las características temporales, espectrales y de comportamiento frente al ruido propias de cada modulación.

# Importante

Antes de empezar la práctica 5 se pide realizar un **estudio teórico** de los diagramas de bloques y de las prestaciones de las modulaciones digitales.

Sus resultados teóricos deberán ser verificados en los modelos de SIMULINK. Por tanto, **el alumno deberá enseñar dicho estudio teórico al profesor antes de comenzar la práctica**.

# Estudio Teórico de las Modulaciones Digitales

## 1. Modulaciones Digitales

### 1.1 Tipos de Modulación

- **ASK (Amplitude Shift Keying)**

La modulación ASK es básicamente una modulación de pulsos en amplitud (PAM) trasladada a la frecuencia deseada. Esto es, a cada secuencia de $n$ bits se la asocia una amplitud predefinida de la portadora. Las señales transmitidas tienen la forma:

$$s_i(t) = A_c a_i \cos(\omega_c t + \theta), \quad 0 \le t \le D$$

con $a_i = 0,\ldots,(M\text{-}1)$ y $D$ el periodo de símbolo o periodo de señalización definido como $D$=$1/v_t$, siendo $v_t$ la velocidad de transmisión en baudios. $\theta$ es una fase inicial de la portadora que consideraremos nula a partir de ahora sin pérdida de generalidad.

En esta práctica nos limitaremos al estudio de la ASK binaria ($i$=0,1), más conocida por el nombre de OOK (*On-Off Keying*). Históricamente fue la primera en emplearse y consiste en transmitir portadora cuando se envía un "1" y no transmitir ninguna señal cuando se envía un "0". Por lo tanto, las señales transmitidas son:

$$s_1(t) = A_c \cos(\omega_c t), \quad 0 \le t \le D$$

$$s_0(t) = 0, \quad 0 \le t \le D$$

Con la misma probabilidad de transmitir unos y ceros tenemos que la energía por bit será:

$$E_b = P(H_0)E_0 + P(H_1)E_1 = \frac{A_c^2 D}{4}$$

Una ventaja de la modulación OOK frente a PSK binaria es que puede ser detectada con receptores no coherentes como la AM.

- **PSK (Phase Shift Keying)**

Es una modulación digital en la que a cada secuencia de bits se le asocia una de las fases de la portadora escogida de entre un conjunto finito de valores equiespaciados. Así pues, una señal PSK de $M$ niveles es generada asociando un bloque de $n = \log_2 M$ dígitos binarios a una de las $M$ fases correspondientes, por lo que las señales transmitidas serán de la forma:

$$s_i(t) = A_c \cos(\omega_c t + \phi_i), \quad 0 \le t \le D$$

donde

$$\phi_i = \frac{2\pi}{M}(i-1), \quad i = 1,2,\ldots,M$$

En esta práctica se propone un estudio sobre PSK binaria, por lo que las dos señales transmitidas están desfasadas 180 grados. Dicha variación de fase se corresponde realmente con un cambio de signo en la amplitud, de tal forma que las señales transmitidas son:

$$s_1(t)=A_c\cos(\omega_c t),\quad 0\le t\le D$$

$$s_0(t)=-A_c\cos(\omega_c t),\quad 0\le t\le D$$

Estas señales pueden ser representadas como vectores opuestos y reciben el nombre de señales antipodales. La energía de bit en este caso, suponiendo símbolos equiprobables resulta

$$E_b=E_0=E_1=\frac{A_c^2 D}{2}$$

- **FSK (Frequency Shift Keying)**

Es aquella modulación tal que el conjunto de señales transmitidas se diferencia únicamente en la frecuencia de la portadora, siendo independiente de su amplitud y fases iniciales. A cada símbolo, o agrupación de *n* bits, se le asocia una frecuencia de la sinusoide $f_i$ que puede tomar los valores:

$$f_i=f_c+i\frac{\Delta f}{2}=f_c+if_d\quad,\quad i=\pm1,\pm3,\ldots,\pm(M-1)$$

donde $\Delta f=2f_d$ representa la separación de frecuencias entre dos símbolos consecutivos y a $f_d$ se le denomina desviación de frecuencia.

La expresión analítica de la señal es:

$$s_i(t)=A_c\cos(\omega_i t)\quad,\quad 0\le t\le D$$

donde

$$\omega_i=\omega_c+2\pi if_d\quad,\quad i=\pm1,\pm3,\ldots,\pm(M-1)$$

Por tener todos los símbolos la misma amplitud, la energía de todos ellos es la misma, por lo que la energía media por bit resulta:

$$E_b=E_i=\frac{A_c^2 D}{2}$$

Para reducir la probabilidad de error se impone que las señales sean ortogonales, para lo cual el coeficiente de correlación entre ellas debe ser nulo. Esto se cumple para desviaciones de frecuencia $f_d$ múltiplos de $1/(2D)$.

En esta práctica se propone un estudio de la FSK binaria, BFSK, para valores de la desviación de frecuencia, $f_d$,dados por $1/(2D)$ y $1/D$. La detección podrá ser asimismo coherente y/o no coherente.

- **QAM (Quadrature Amplitude Modulation)**

Debido a que el espectro de frecuencias es un bien escaso, se debe transmitir la máxima información en el mínimo ancho de banda posible. En otras palabras, debemos utilizar modulaciones con alta eficiencia espectral. Una implementación que aumenta la eficiencia de la modulación ASK consiste en transmitir simultáneamente dos portadoras en cuadratura, *cos(*$\omega_c$*t)* y *sen(*$\omega_c$*t)*. De ahí su nombre QAM.

Estas portadoras estarán moduladas en amplitud por dos secuencias independientes de $n$ bits. La expresión analítica de las señales a transmitir es, por tanto:

$$s_i(t) = A_c a_i \cos(\omega_i t) + A_c b_i \,\mathrm{sen}(\omega_i t) \quad , \quad 0 \le t \le D$$

con

$$a_i, b_i \in \left\{\pm 1, \ldots, \pm(\sqrt{M} - 1)\right\}$$

Se observa que se producen variaciones tanto en la amplitud como en la fase de las señales a transmitir, lo que indica que la modulación QAM se puede considerar una combinación de las modulaciones en amplitud y en fase.

En esta práctica se propone un estudio de la modulación QAM de 4 niveles, modulación equivalente a la QPSK, que permite aumentar la capacidad de transmisión respecto a las binarias sin aumentar la ocupación espectral proporcionalmente.

## 1.2 Análisis Espectral de las Modulaciones Digitales

Cualquier señal digital paso-banda (como son las modulaciones digitales) puede expresarse en función de sus componentes en fase, $y_c(t)$, y cuadratura, $y_s(t)$, denominadas componentes paso-bajo o banda-base:

$$y(t) = A_c\left[y_c(t)\cos(\omega_c t) - y_s(t) sen(\omega_c t)\right]$$

El equivalente banda-base de la señal modulada se define como una señal compleja tal que:

$$\boxed{y_{bb}(t) = y_c(t) + j y_s(t)}$$

$j y_s(t)$

$y_c(t)$

para plotear la constelación en Matlab

La densidad espectral de potencia (DEP), $G_y(f)$, de la señal modulada paso-banda se puede expresar mediante la correspondiente DEP de su equivalente banda-base, $G_{bb}(f)$:

$$\boxed{G_y(f) = \frac{A_c^2}{4}\left[G_{bb}(f - f_c) + G_{bb}(f + f_c)\right]}$$

siendo $G_{bb}(f) = G_c(f) + G_s(f)$ el *espectro equivalente paso-bajo*.

Para calcular la expresión anterior es necesario particularizarla para cada modulación. A modo de ejemplo, si la expresión temporal de la componente en fase viene dada por:

$$y_c(t) = \sum_k a_k p(t - kD)$$

su densidad espectral correspondiente es:

$$G_c(f) = \frac{\sigma_a^2}{D}\left|P(f)\right|^2 + \left(\frac{m_a}{D}\right)^2 \sum_n \left|P\left(\frac{n}{D}\right)\right|^2 \delta\left(f - \frac{n}{D}\right)$$

donde $D$ es el periodo de símbolo, $\sigma_a^2$ es la varianza y $m_a$ la media de las amplitudes $\{a_k\}$, y $P(f)$ es la transformada de Fourier del pulso temporal $p(t)$.

**1.2.a Represente de forma aproximada los espectros de las modulaciones digitales descritas en el punto anterior y cuyo espectro equivalente banda-base viene dado por:**

- OOK: $G_{bb}(f)=\frac{D}{4}\sin c^2(fD)+\frac{9}{4}\delta(f)$
- BPSK: $G_{bb}(f)=T\sin c^2(fD)$
- BFSK con $f_d=1/2D$: $G_{bb}(f)=\frac{4D}{\pi^2}\left[\frac{\cos(\pi fD)}{(2fT)^2-1}\right]^2+\frac{1}{4}\left[\delta\left(f-\frac{1}{2D}\right)+\delta\left(f+\frac{1}{2D}\right)\right]$
- QAM: $G_{bb}(f)=\frac{\sigma_a^2}{D}\sin c^2(fD)$

Para cada una de ellas especifique el ancho de banda de transmisión ($B_T$) según el criterio utilizado en *Teoría de la Comunicación*.

## 1.3 Representación Geométrica de Señales y Ruido

- **Representación Geométrica de Señales**

Cualquier forma de onda realizable físicamente y de duración $D$ segundos puede expresarse como una combinación lineal de las $N$ funciones base ortogonales $\phi_j(t)$, $j=1,\ldots,N$, que forman el llamado *espacio de señal*. Su formulación es:

$$s(t)=a_1\phi_1(t)+a_2\phi_2(t)+\cdots+a_N\phi_N=\sum_{j=1}^{N}a_j\phi_j(t)$$

donde

$$a_j=\frac{1}{E_j}\int_0^D s(t)\phi_j(t)dt \quad , \quad j=1,\ldots,N$$

y $E_j$ es la energía de la función base j-ésima $\phi_j(t)$, es decir, $E_j=\int_0^D\phi_j^2(t)dt$. Se dice que la base del espacio de señal es ortonormal cuando $E_j=1,\ \forall j$.

A partir de esta representación de las señales paso-banda podemos establecer dos definiciones muy interesantes que van a ser fundamentales a la hora de detectar las señales:

- Energía de una señal: $E_s=\int_0^D s^2(t)dt=\int_0^D\left(\sum_{j=1}^{N}a_j\phi_j(t)\right)^2dt=\sum_{j=1}^{N}a_j^2E_j$
- Distancia entre señales: $d_{ik}^2=\int_0^D[s_i(t)-s_k(t)]^2dt=\sum_{j=1}^{N}(a_{ij}-a_{kj})^2E_j$

**1.3.a Dadas las siguientes funciones base ortogonales, calcular las componentes $a_j$, la energía de cada señal, $E_j$, y la distancia entre las distintas señales $s_i(t)$ para las modulaciones digitales descritas en el punto 2.1:**

- OOK, BPSK, 4-QAM: $\phi_1(t)=A_c\cos(\omega_c t),\ 0\le t\le D\ ;\phi_2(t)=A_c\,\mathrm{sen}(\omega_c t),\ 0\le t\le D$
- BFSK: $\phi_1(t)=A_c\cos((\omega_c+2\pi f_d)t),\ 0\le t\le D\ ;\phi_2(t)=A_c\cos((\omega_c-2\pi f_d)t),\ 0\le t\le D$

Realice asimismo la representación geométrica de dichas señales en el eje de coordenadas constituido por $\phi_1(t)$ y $\phi_2(t)$.

## 1.4 Detección de Señales en Ruido Gaussiano

### ➢ Detección Coherente: Receptor por Correlación

Bajo la perspectiva geométrica introducida en el punto anterior, el detector óptimo de señales digitales paso-banda será aquél que detecte como señal transmitida, $s_i(t)$, aquélla cuya distancia a la señal recibida, $y(t)$, sea la mínima. Por tanto, el detector deberá calcular la distancia a cada $s_i(t)$, $i$=1,...,$M$ y elegir aquélla que de el mínimo valor. Si desarrollamos la expresión de la distancia obtenemos:

$$d_{ys_i}^2 = \int_0^D [y(t) - s_i(t)]^2 dt = \int_0^D y^2(t)dt - 2\int_0^D y(t)s_i(t)dt + \int_0^D s_i^2(t)dt$$

El primer término es común a todas las señales, y si todas ellas tienen igual energía el tercer término también tendrá el mismo valor para cualquier $i$. Por tanto, el término discriminatorio es el segundo y el detector óptimo es aquél que realiza la siguiente operación:

$$\hat{s}_i(t) = \max_{s_i(t)}\left\{\int_0^D y(t)s_i(t)dt\right\}$$

La implementación del detector óptimo mediante filtros correladores se denomina *receptor por correlación* y consta de los siguientes elementos:

1. Un banco de integradores por producto o *correladores*
2. Un circuito muestreador en cada rama con $t_s = kD$
3. Un circuito de decisión

En el caso en que no todas las energías sean iguales (OOK por ejemplo), habría que restar el valor $E_{si}/2$ en cada rama antes de seleccionar el máximo.

Otra posible implementación del receptor por correlación se obtiene en función de las funciones base $\phi_j(t)$. En efecto, desarrollando la expresión de la integral de correlación en función de $\phi_j(t)$ se obtiene:

$$\hat{s}_i(t) = \max_{s_i(t)}\left\{\int_0^D y(t)\sum_{j=1}^{N} a_{ij}\phi_j(t)dt = \sum_{j=1}^{N} a_{ij}\int_0^D y(t)\phi_j(t)\right\}$$

siendo el criterio de decisión en este caso el de seleccionar aquella $s_i(t)$ cuyos $a_{ij}$ son los más próximos a los medidos en las ramas.

### ➢ Receptor por Correlación para el Caso Binario

En el caso binario el circuito de decisión se implementa de una forma muy sencilla: se resta el valor obtenido en la rama correspondiente a $s_1(t)$ menos el valor de la rama de $s_0(t)$ y el resultado se compara con un umbral nulo, es decir

$$\left[\int_0^D y(t)s_1(t)dt - E_1/2\right] - \left[\int_0^D y(t)s_0(t)dt - E_0/2\right] \underset{H_0}{\overset{H_1}{\gtrless}} 0$$

o, lo que es lo mismo,

$$\int_0^D y(t)[s_1(t) - s_0(t)]dt \underset{H_0}{\overset{H_1}{\gtrless}} \frac{E_1 - E_0}{2}$$

donde el umbral óptimo viene dado por el cociente $\frac{E_1 - E_0}{2}$.

## 1.5 Probabilidad de Error

La probabilidad de error de las modulaciones utilizadas en esta práctica se da a continuación sin demostración previa. Sin embargo, se recomienda repasar la obtención de dichas expresiones dada en clase. Todas ellas suponen símbolos equiprobables y receptor por correlación, excepto en los casos en los que explícitamente se indica un tipo de detección no coherente.

$E_b$ representa la energía media por bit y $N_0$ la densidad espectral en W/Hz del ruido blanco gaussiano a la entrada del receptor.

- **OOK**

$$\boxed{P_{B_{OOK}} = Q\left(\sqrt{\frac{E_b}{N_0}}\right)}$$

- **BPSK**

$$\boxed{P_{B_{BPSK}} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)}$$

- **BFSK**

$$\boxed{P_{B_{BFSK}} = Q\left(\sqrt{\frac{E_b}{N_0}}\right)}$$

- **QAM**

En las modulaciones de más dos niveles la expresión del error se refiere a la probabilidad de error por símbolo, $P_e$. En QAM la probabilidad de error por símbolo se expresa como:

$$\boxed{P_e \approx 2Q\left(\sqrt{\frac{2E_b}{N_0}}\right)}$$

Usando codificación Gray, la probabilidad de error por bit resulta:

$$\boxed{P_{B_{QAM}} = P_e / 2 \approx Q\left(\sqrt{\frac{2E_b}{N_0}}\right)}$$

# Práctica 5: Modulaciones Digitales

## 1. Importante: Antes de Empezar

### 1.1 Ficheros MDL y MAT

- Se proporcionan los ficheros **PSK.MDL, EvalBER_QAM** y **CONFIG5.MAT** que se deben situar en el directorio de trabajo. Las variables usadas por los bloques de las modulaciones digitales son:
  - D (Periodo de símbolo). Es la duración de un símbolo en segundos, y su inverso equivale a la velocidad de transmisión, $v_t$.
  - M (Número de símbolos de la modulación). En las modulaciones binarias vale 2 y en QAM vale 4.
  - fc (Frecuencia de portadora): En las modulaciones digitales es recomendable que el periodo de símbolo (D) coincida con un número entero de periodos de la portadora. Esto se realiza obligando a que fc·D sea un número entero.
  - Tsim (*Periodo de muestreo de la simulación*): En estas simulaciones la relación entre D y Tsim debe ser tal que D/Tsim también sea un número entero.
  - EbNodB (Relación entre la energía por bit y la DEP del ruido blanco, en dB): Usada en el bloque *AWGN channel*.
- Cuando tenga que estudiar las modulaciones FSK y QAM, deberá usar el fichero **PSK.MDL**, salvarlo con otro nombre, y reemplazar los moduladores por los de la respectiva modulación. Dichos bloques los puede encontrar en la *Communications Blockset*.
- A continuación, deberá establecer los parámetros de cada modulador para que simule el sistema que se le pide. No olvide actualizar también el valor de Tsim.

### 1.2 Estructura del modelo PSK.MDL

- El fichero **PSK.MDL** consta de 2 diagramas de bloques:
  - El **superior** (ver figura 21) simula la transmisión paso-banda como tal. Se usará para visualizar las **señales en el tiempo y en la frecuencia**.
  - El **inferior** (ver figura 22) simula la modulación mediante su equivalente banda base. Se usará para visualizar la **constelación de las señales**.

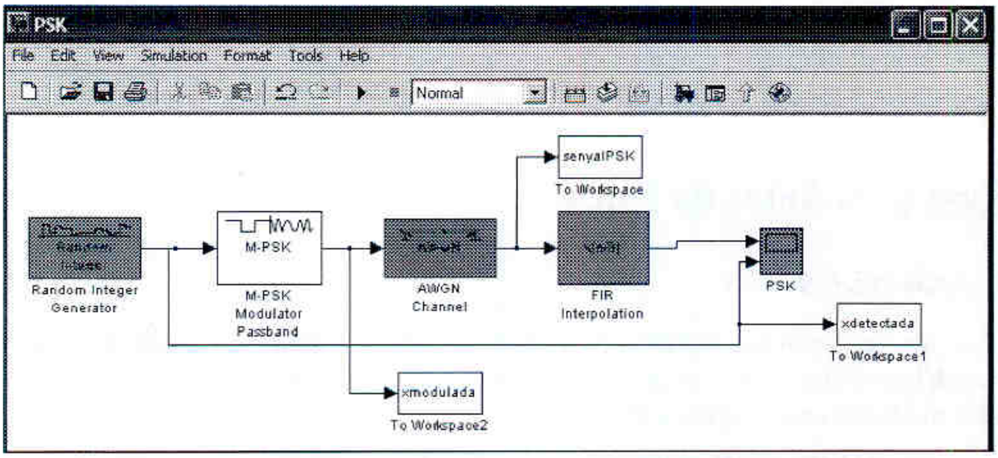



Figura 21. Diagrama de bloques de la modulación simulada como paso-banda.

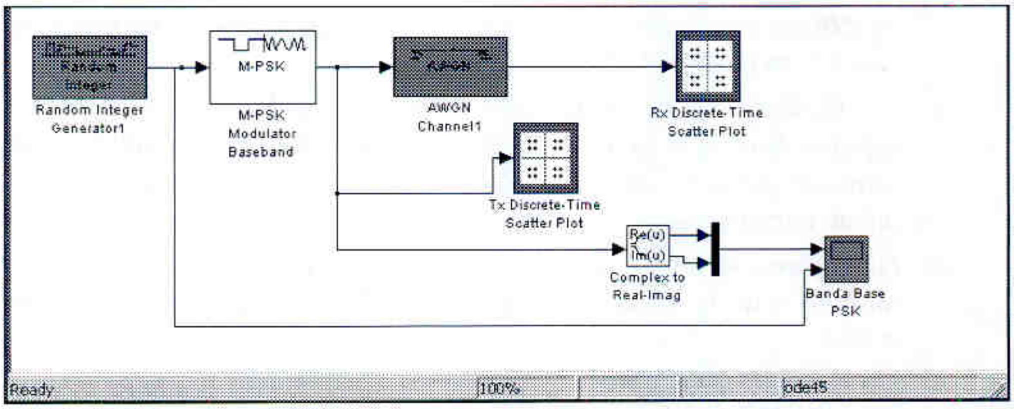



Figura 22. Diagrama de bloques de la modulación simulada como banda-base.

## 1.3 Representación del Espectro de las Modulaciones Digitales

- Debido a que las señales digitales son procesos aleatorios, su espectro no puede visualizarse a través de la transformada de Fourier, sino mediante la Densidad Espectral de Potencia (DEP). Para visualizar dicha DEP, matlab dispone de la instrucción **PSD.M** (iniciales de *Power Spectral Density*, Densidad Espectral de Potencia). Se recomienda usarla con los siguientes parámetros a fin de obtener una buena resolución del espectro:
    - >> psd(nombre_de_la_variable,1024,1/Tsim,512);

## 2. Estudio de las Modulaciones Digitales.

En el modelo propuesto para PSK aparece un bloque nuevo denominado **Discrete-Time Scatter-plot Scope** (ver figura 22) que permite visualizar la constelación de la señal modulada. Antes de empezar a ejecutar la simulación se recomienda consultar la ayuda del bloque para conocer sus parámetros.

### 2.1 Estudio de la Modulación PSK

Genere una modulación binaria PSK con los siguientes parámetros:

- Velocidad de transmisión: 200 bps $D = 1/200$ s
- Frecuencia de portadora: 2000 Hz
- Relación $E_b/N_0$: 30 dB
- Tsim: D/40 para ver el espectro y D/150 para ver la forma de onda temporal.

**2.1.a Observe la <u>señal modulada en el tiempo</u> y compruebe la diferencia de fase existente entre el bit '0' y el bit '1'.** dibuja (xmodulada, Tsim)

**2.1.b Observe la <u>señal modulada en la frecuencia</u> y mida el ancho de banda entre nulos. Compárelo con el teórico, $B_T = v_t$.** psd(xmodulada, 1024, 1/Tsim, 512)

**2.1.c Observe el espectro del código fuente a la salida del primer bloque (*Random Integer Generador*) ¿Cuál es su ancho de banda considerando hasta el primer nulo de la sinc? ¿Qué relación tiene con el espectro de la señal modulada PSK?.**
200 Hz Es la misma pero desplazada

**2.1.d Cambie el parámetro $E_b/N_0$ a discreción y observe como le afecta el ruido a la constelación, a la forma de onda de la señal y al espectro.** A menor $E_b/N_0$ mas dispersos salen los puntos en la constelación

### 2.2 Estudio de la Modulación FSK

<u>Cambie ahora los respectivos moduladores paso-banda y banda-base</u> y especifique los valores de los parámetros necesarios para simular la siguiente modulación binaria FSK:

- Velocidad de transmisión: 200 bps
- Frecuencia de portadora: 2100 Hz
- Relación $E_b/N_0$: 30 dB
- Tsim: D/40 para ver el espectro y D/150 para ver la forma de onda temporal.
- *Frequency separation* (Hz): 1/D. **<u>Nota</u>**: La separación 1/D que utilizan los moduladores FSK de Simulink es el doble de la desviación de frecuencia $f_d$ (ver pág. 36, apartado FSK). Por tanto, la señal y el espectro de esta modulación se corresponde con $f_d$=1/(2D). $D = \frac{1}{2f_d}$

**2.2.a Observe la <u>señal modulada en el tiempo</u> y verifique las diferentes frecuencias para codificar el bit '0' y el bit '1'.**

**2.2.b Observe el espectro de la señal transmitida y compárelo con su expresión teórica, ¿Se aprecian las dos componentes frecuenciales? ¿dónde están situadas?**

**2.2.c Explique porqué en la constelación de esta señal aparecen más puntos de los que a priori debería tener.**

<u>Para el 1</u>:

$$\cos(2\pi \cdot (f_d + f_c) \cdot t) = \overbrace{\cos(2\pi f_d t)}^{y_c(t)} \cdot \cos(2\pi f_c t) - \overbrace{\text{sen}(2\pi f_d t)}^{y_s(t)} \cdot \text{sen}(2\pi f_c t)$$

$$y_c(t)\big|_{t=nD} = \cos(2\pi \cdot f_d \cdot n \cdot D) = \cos(\pi n)$$

$$y_s(t) = \text{sen}(\pi n)$$

} depende de cada cuanto se muestree para mostrar la constelación veremos mas o menos puntos en una circunferencia

Cambie ahora el siguiente parámetro:

- $f_d$=1/D


La separación entre deltas es de 2fd

**2.2.d Observe en el espectro de la señal transmitida 2 sincs alrededor de las componentes frecuenciales.**

**2.2.e Cambie el parámetro $E_b/N_0$ a discreción y observe como le afecta el ruido a la constelación, a la forma de onda de la señal y al espectro.** Los puntos de la constelación van saliendo más dispersos y el ruido en la psd es mayor

### 2.3 Estudio de la Modulación QAM

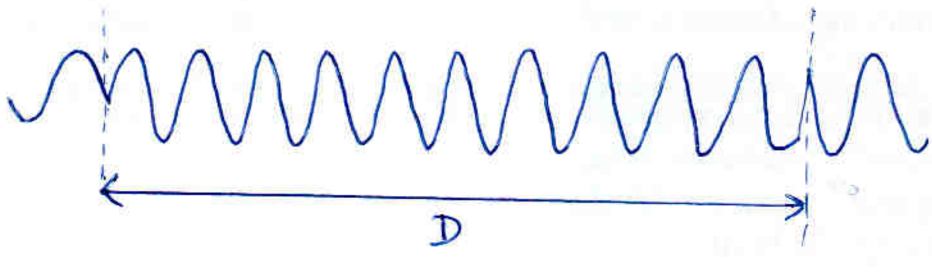
<u>Cambie ahora los respectivos moduladores paso-banda y banda-base</u> y especifique los valores de los parámetros necesarios para simular la siguiente modulación QAM:

- Velocidad de transmisión: 200 baudios
- Frecuencia de portadora: 2000 Hz
- Relación $E_b/N_0$: 30 dB
- Parámetro *M* de la simulación ***M*=4.**
- Tsim: D/40 para ver el espectro y D/150 para ver la forma de onda temporal.

**2.3.a Observe la señal modulada, fíjese cuándo se producen los cambios de fase y cuando la fase se mantiene continua. Visualice asimismo la constelación y compárela con la teórica.**

**2.3.b Observe el espectro de la señal <u>modulada</u> y verifique la mayor eficiencia espectral de QAM respecto a las modulaciones binarias.**

**2.3.c Cambie el parámetro $E_b/N_0$ a discreción y observe como le afecta el ruido a la constelación, a la forma de onda de la señal y al espectro.**



# 3. Evaluación del error en las Modulaciones Digitales.

Para este apartado deberá utilizar el fichero **EvalBER_QAM.MDL**.

## 3.1 Fichero EvalBER_QAM.MDL

En él aparece un nuevo bloque denominado **Bit Error Details**. Este bloque es un Medidor de la Tasa de Error y tiene la apariencia de la figura:

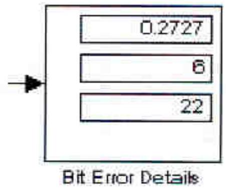



Figura 23. Medidor de la Tasa de Error

donde la cifra superior indica la **Tasa de Error** (equivalente al Bit Error Rate, BER), la cifra intermedia indica el **Número de Bits Erróneos** y la inferior indica el **Número de Bits Recibidos**.

A la hora de comparar la Tasa de Error de una modulación con la probabilidad de error teórica es conveniente dejar correr la simulación hasta que se reciban por lo menos 4000 bits. Para ello deberá cambiar el *Stop time* dentro del menú de parámetros de la simulación (ver figura 24).

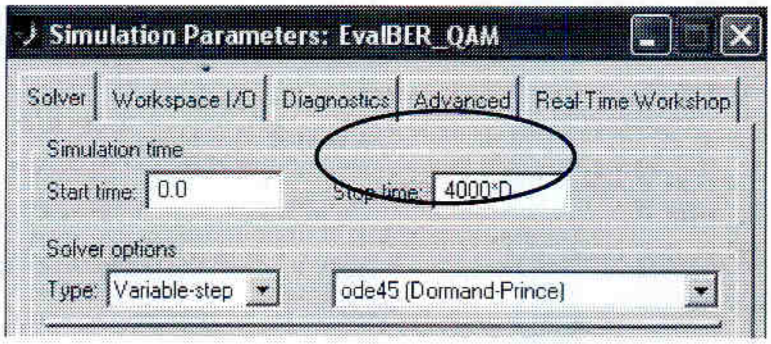



Figura 24. Menú de parámetros de la simulación.

## 3.2 Cálculo de la Probabilidad de Error

- En las modulaciones estudiadas, la probabilidad de error obtenida se calcula mediante la función *Q(x)*, siendo *x* el argumento de dicha función. Para calcular el valor de *Q(x)* se puede utilizar la llamada *función error complementario* que Matlab tiene implementada como **erfc.m** de esta forma:

$$Q(x) = \frac{1}{2}\mathrm{erfc}\left(\frac{x}{\sqrt{2}}\right)$$

### 3.3 Evaluación de la Tasa de Error (BER)

Establezca los siguientes parámetros de la simulación:

- Velocidad de transmisión: 200 baudios
- Tsim: D/10 s.
- Parámetro $M$ de la simulación $M$**=4.**
- Relación $E_b/N_0$: variable entre -6 y 8 en saltos de 2 dB.
- Stop time: 8000*D.

**3.3.a Ejecute la simulación para los distintos valores de $E_b/N_0$ y rellene la siguiente tabla con el resultado de la Tasa de Error por Bit (BER) y por Símbolo (SER):**

| $E_b/N_0$ | -6 dB | -4 dB | -2 dB | 0 dB | 2 dB | 4 dB | 6 dB | 8 dB |
|---|---|---|---|---|---|---|---|---|
| BER | 0'23914 | 0'1871 | 0'13235 | 0'078704 | 0'038519 | 0'013333 | 0'002222 | 0'00012346 |
| SER | 0'41889 | 0'33827 | 0'24765 | 0'15173 | 0'075556 | 0'026543 | 0'004444 | 0'00024691 |

**3.3.b Represente la BER en funcion de $E_b/N_0$ en Matlab, pero en donde el eje de ordenadas esté en escala logarítmica (función semilogy.m). Compárela con la curva teórica de la probabilidad de error por bit ¿Es coherente el resultado práctico con el teórico?.**

**3.3.c** **<u>EJERCICIO OPCIONAL</u>: Rellene una tabla similar para una modulación binaria PSK y compárela con la QAM anterior.**

```
BER_vector = [ ];
SER_vector = [ ];

for EbNodB = -6:2:8
    sim ('Eval Ber_QAM');
    BER_vector(size(BER_vector,1)+1) = BER(1);
    SER_vector(size(SER_vector,1)+1) = SER(1);
```

```
for i= 1 : 1 : 8
    EbNodB= (i-4)*2;
    sim ('EvalBER_QAM');
    BER_vector (i)= BER(1);
    SER_vector (i) = SER(1);
end

semilogy ([-6:2:8], SER_vector)
```