Nota

Estas notas teóricas han sido concebidas para facilitar al alumno la labor de búsqueda del material teórico sobre el que se sustenta la práctica. Están especialmente dirigidas a los alumnos del grupo 31 y no constituyen, en ningún caso, un documento oficial de la asignatura de Tratamiento Digital de la Señal.

Madrid, noviembre de 2020

Jesús Gustavo Cuevas del Río

Anexo teórico a la Práctica 3

1. Resumen sobre clasificación de sistemas.

- *Un sistema es **real** si la respuesta a cualquier entrada real es real. Como consecuencia:*
    - *Los coeficientes de H(z) son reales y sus polos y ceros son simétricos respecto al eje real del plano-Z*
    - *H es simétrica conjugada $H(e^{j\omega})=H^*(e^{-j\omega})$*
- *Un sistema es **causal** si la respuesta al impulso h[n] es nula para n < 0.*
- *Un sistema es **estable** si su respuesta al impulso es absolutamente sumable.*

$$\sum_{n=-\infty}^{\infty} |h[n]| < \infty$$

Como consecuencia:

- *H(z) tiene todos los polos p_k estrictamente dentro (fuera) del círculo unidad si el sistema es además causal (anticausal). La estabilidad se puede determinar, por tanto, calculando las raíces del polinomio denominador en la expresión de la función del sistema.*

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}}$$

- *Un sistema es **FIR** (respuesta finita al impulso) si su respuesta h[n] es de longitud finita. Si el sistema es, además, causal resulta que: $a_k = 0$, $k > 0$ (e. d. $N = 0$ y $b_n = h[n]$, $n=0,\cdots,M$). Obviamente todos los polos son cero. Los M+1 valores de h[n] describen el sistema completa e inmediatamente.*
- *Un sistema **causal** y **estable** se llama de **fase mínima** si su fase es la menor posible de todos los sistemas con la misma magnitud de la respuesta en frecuencia. Hay dos tipos de sistemas de fase mínima:*
    - *Los que son invertibles, es decir, los sistemas cuyas funciones de transferencia H(z) son tales que 1/H(z) es estable y también de mínima fase. En este caso todos los ceros de H(z) residen en el interior del círculo unidad.*
    - *Los que no son invertibles. En este caso parte de los ceros de H(z) están en el exterior del círculo o sobre la circunferencia unidad.*

Un sistema de fase mínima se distingue porque su retardo de grupo, τ_g satisface la condición:

$$\boxed{\int_0^{\pi} \tau_g(\omega)d\omega = 0}$$

(la anterior condición implica que debe haber uno o mas intervalos en donde el retardo de grupo sea negativo)

- *Un sistema **causal** y **estable** se llama de **fase máxima** si su fase es la mayor posible de todos los sistemas con el mismo grado y la misma magnitud de la respuesta en frecuencia. En este caso todos los ceros de H(z) están fuera del círculo unidad.*
    - *Un sistema causal con respuesta al impulso $\{h(n)\}$ tiene asociada una **energía parcial** **E(N)** dada por $E(N)=\sum_{k=0}^{N}|h(k)|^2$. Una vez sumados todos los términos (N $\rightarrow \propto$), todos los sistemas con igual magnitud de la función de transferencia tendrán la misma energía total. Si no se suman todos los términos, se puede demostrar que el sistema de fase mínima es el que tiene mayor energía parcial, y el sistema de fase máxima el que tiene menor energía parcial, para un N dado, es decir, $E_{\max}(N)\leq E(N)\leq E_{\min}(N)$.*
- *Un sistema se llama **paso-todo** si la magnitud de su respuesta en frecuencia cumple: $|H_{PT}(\omega)|=cte$. Los ceros de la función de transferencia de un sistema paso-todo son imágenes especulares, respecto de la circunferencia unidad, de los polos.*
    - *Un sistema paso-todo estable es un sistema de fase máxima.*
    - *Cualquier sistema de fase **no mínima** (e. d., que posee algunos ceros fuera de la circunferencia unidad), caracterizado por $H(z)$, se puede descomponer como producto de un sistema de fase mínima y otro paso-todo:*

$$H(z)=H_{min}(z)H_{PT}(z).$$

    - *Un sistema de fase mínima tiene un inverso de fase mínima también. Si un sistema de fase mínima se conecta en cascada con su inverso (también de fase mínima) se obtiene la señal de entrada. En tal caso se habla de que el sistema original puede ser **totalmente compensado**.*
    - *Un sistema de fase no mínima sólo puede ser **parcialmente compensado** en el sentido siguiente:*
        - *Un sistema $H(z)$ de fase no mínima se puede factorizar como $H(z)=H_{min}(z)H_{PT}(z)$. El subsistema de fase mínima tiene un inverso $H_{\min}^{-1}(z)$, el cual, se puede asociar en cascada con $H(z)$ para producir: $H(z)H_{\min}^{-1}(z)=H_{\min}(z)H_{\min}^{-1}(z)H_{PT}(z)=H_{PT}(z)$ de forma que el sistema residual es un sistema paso todo el cual tiene magnitud de la respuesta en frecuencia constante mientras que la fase es la del sistema paso-todo. Dicho de otra forma, el sistema $H(z)$ es compensado en amplitud pero no puede ser compensado en fase.*
    - *Un sistema **causal** y **estable** tiene **fase lineal** si es FIR y los ceros de su función de transferencia están sobre la circunferencia unidad o dispuestos en parejas reflejadas respecto a la circunferencia unidad. Esto significa que un cero en z_l tiene como imagen*

especular un cero en $1/z^*_l$. Los sistemas de fase lineal presentan una respuesta en frecuencia de la forma

$$H_{fl}(\omega) = A(\omega)e^{-j(\omega\alpha+\beta)}; \qquad A(\omega) = \pm A(\omega)$$

Un sistema de fase lineal generalizada, como cualquier otro sistema, admite una factorización del tipo $H(z) = H_{min}(z)H_{PT}(z)$, pero también admite otra factorización del tipo

$$H_{fl}(z) = H_{\min}(z)H_{\max}(z)H_{cu}(z); \quad H_{\max}(z) = z^{-M}H_{\min}(z^{-1})$$

en donde M es el número de ceros dentro, H_{min}, H_{max}, y H_{cu} son sistemas con todos los ceros dentro, fuera y justo en la circunferencia unida, respectivamente.

- *Hay sistemas que admiten ocasionalmente otras factorizaciones, como por ejemplo*

$$H(z) = H_{fl}(z)H_{PT}(z)$$

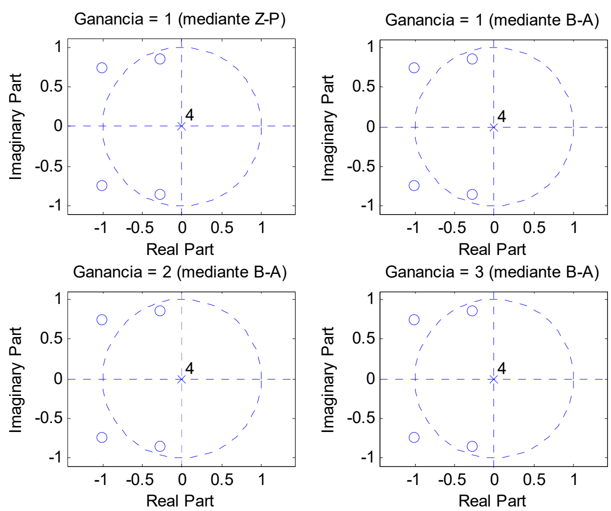
Pero son casos particulares no garantizados para otros sistemas.

2. Representación de sistemas en MATLAB.

Un sistema cuya función del sistema $H(z) = B(z)/A(z)$ es racional, está completamente caracterizado por los coeficientes de los polinomios del numerador y del denominador. En formato MATLAB los coeficientes son sendos **vectores fila:** `B = [b0, b1, …, bM]` y `A = [a0, a1, …, bN]` de longitudes M+1 y N+1, respectivamente, y reales si el sistema es real. Alternativamente podemos conocer los ceros del polinomio $B(z)$ y los ceros del polinomio $A(z)$, es decir, los ceros y los polos del sistema $H(z)$, que en MATLAB se tratan almacenan como **vectores columna**, en general de números complejos: `Z = [c1; … ;cM]` y `P = [p1; … ;pN]` , de orden M *y* N, respectivamente. Observe que el separador es ahora un "`;`", lo que indica que son vectores columna. El problema con los ceros y los polos es que por sí mismos no definien unívocamente el sistema ya que hay infinitos sistemas que se corresponden con el mismo sistema de ceros y polos y que sólo se diferencian en la ganancia. En algunos comandos de MATLAB,(como en `zp2tf`), además de los propios vectores Z y P es necesario especificar una ganancia G. Compárese la utilización de ceros polos y coeficientes polinómicos en el siguiente ejemplo.

```
%Datos
z1=0.9*exp(1i*0.6*pi); z2=1.25*exp(1i*0.8*pi); %números complejos
% Ceros de un FIR de orden M=4
Z=[z1; z1'; z2; z2'];
% Polos de un FIR de orden M=4
P=[0;0;0;0];  %vector de polo 4-druple en el origen
% Construcción de los coeficientes B y A mediante "poly"
[B,A]=zp2tf(Z,P,1);  %alternativamente: B = poly(Z); A=poly(P);
figure(1)
subplot(2,2,1),zplane(Z,P) %diagrama cero-polo mediante Z Y P
title ('Ganancia = 1 (mediante Z-P)')
subplot(2,2,2),zplane(B,A) %diagrama cero-polo mediante B Y A
title ('Ganancia = 1 (mediante B-A)')
subplot(2,2,3),zplane(2*B,2*A) %diagrama cero-polo mediante B Y A
title ('Ganancia = 2 (mediante B-A)')
subplot(2,2,4),zplane(3*B,3*A) %diagrama cero-polo mediante B Y A
title ('Ganancia = 3 (mediante B-A)')
```

que produce la siguiente salida:



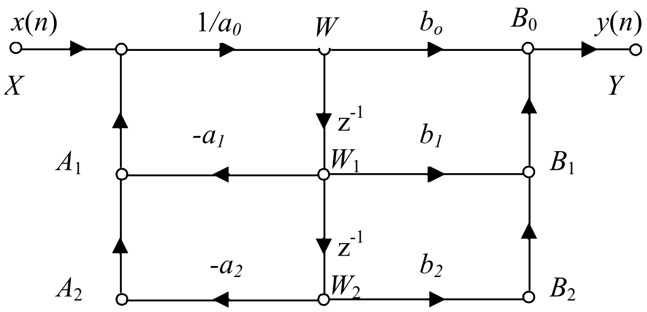
3. Representación de sistemas mediante programas: sección de 2º orden.

Un programa representa a un sistema en el sentido de que, conocido el contenido, puede también deducirse de éste la función de transferencia y/o ecuación en diferencias por simple inspección de sus líneas de código. Por otro lado, la simulación es inmediata mediante la ejecución del programa. Veamos un par de ejemplos.

Ejemplo: Escribir un programa de ordenador que represente una sección de 2º orden en forma canónica. El sistema responde a la siguiente ecuación en diferencias:

$$a_0 y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) - a_1 y(n-1) - a_2 y(n-2)$$

Solución: Asigno un registro a cada nodo, tal como muestra la figura



Las ecuaciones que se obtienen del flujograma son las siguientes.

$$\left.\begin{aligned} W &= \left(x(n) - a_1 W_1 - a_2 W_2\right)/a_0 \\ y(n) &= b_0 W + b_1 W_1 + b_2 W_2 \end{aligned}\right\}$$

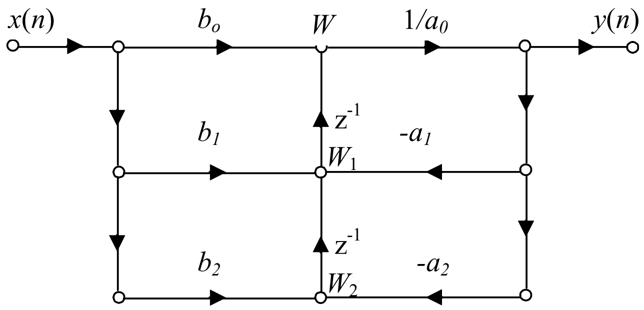
El programa en pseudocódigo sería:

```
% Programa que realiza una sección de 2° orden en forma canónica.
% Llevar valores iniciales, ci a los registros de los retardadores,
W1 ← c1
W2 ← c2
% Filtrado:
for n=0:Ntrama-1  % (bucle do, do forever, while, ····, según el lenguaje)
    % Sumas de productos
    W ← (x(n) - a1*w1- a2*w2)/a0
    y(n) ← b0*W + b1*W1+ b2*W2
    % Actualización ordenada de memoria interna
    W2 ← W1
    W1 ← W
end for
```

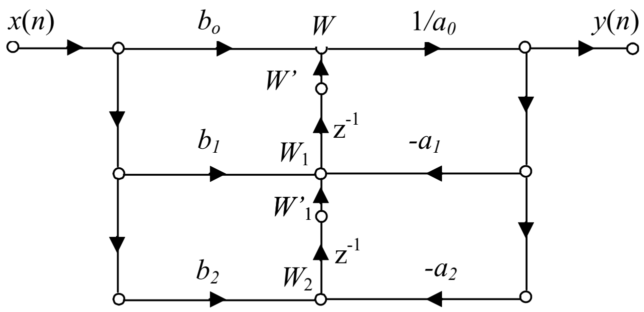
Nótese, después de cada ciclo, la actualización ordenada que se hace de la memoria.

Ejemplo: Escribir un programa de ordenador que represente una sección de 2º orden en forma canónica traspuesta.

Solución: El flujograma de la forma traspuesta es el que se representa en la figura



Sin embargo, y a diferencia del anterior ejemplo, en los nodos W y W_1 confluyen dos o más ramas y las actualizaciones de las memorias, de no hacerse cuidadosamente, podrían dar lugar a errores. Para facilitar la programación conviene introducir dos nuevos nodos, W' y W_1', con el fin de separar los retardadores de los nodos W y W_1.

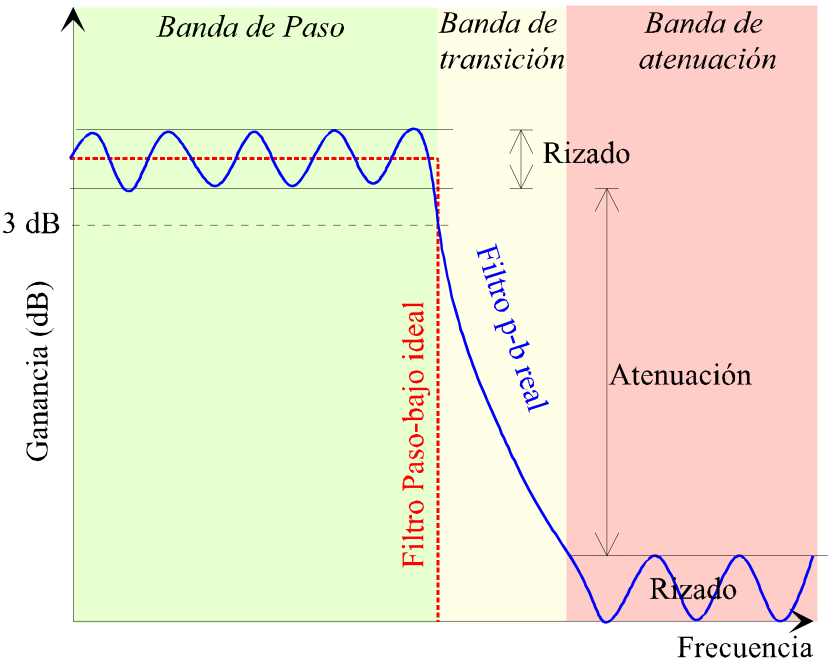


De acuerdo con el anterior flujograma, la programación en pseudocódigo sería la siguiente:

```
%Programa que realiza una sección de 2° orden en forma canónica traspuesta.
% Llevar valores iniciales, ci a los registros de los retardadores,
W1' ← c1
W' ← c2
% Filtrado:
for n=0:Ntrama-1   % (bucle do, do forever, while, ····, según el lenguaje)
   % Entrada de señal
   W    ← b0*x(n) + W'
   y(n) ← W/a0
   W1   ← W1' + b1*x(n)- a1*y(n)
   W2   ← b2*x(n)-a2*y(n)
   % Actualización ordenada de memoria interna
   W' ← W1
   W1'← W2
end for
%Sacar valores de los registros para evetual continuación de filtrado.
cf1 ← W1'
cf2 ← W'
```

4. Diseño de filtros.

Un filtro digital es un sistema que discrimina señales con diferentes contenidos en frecuencia. Las características de un filtro digital se visualizan generalmente mediante representación de su respuesta en frecuencia, tanto en magnitud como en fase. La figura representa la magnitud de la respuesta en frecuencia de un filtro digital paso-bajo (en azul). En línea discontinua roja se representa el comportamiento de un filtro ideal paso-bajo.



Como muestra la figura, y a diferencia de los filtros ideales, un filtro real ha de tener una(s) banda(s) de transición entre la banda(s) de paso y la banda(s) de atenuación. Esta banda puede ser tan estrecha como se quiera –a costa de incrementar el número de coeficientes del filtro- pero no puede ser nunca de anchura nula. Por otro lado, como muestra la figura, algunos filtros reales no puede mantener la magnitud de la respuesta en frecuencia plana en la banda de paso, en la de atenuación o en ambas, sino que han de permitírseles un cierto rizado que puede ser tan pequeño como se quiera, a costa, por supuesto, de aumentar el número de coeficientes del filtro. Finalmente vemos que otra diferencia entre filtros ideales y reales consiste en que estos últimos no atenúan completamente los componentes frecuenciales en la banda de atenuación sino que los atenúan una cierta cantidad (medida en decibelios en la gráfica) respecto de la banda de paso. Esta atenuación puede especificarse tan grande como se quiera, sin llegar a ser infinita, y a costa, nuevamente, de incrementar el número de coeficientes del filtro. La anchura de la banda de paso viene determinada generalmente por la caída a 3 dB de la magnitud de la respuesta en frecuencia del filtro real.

En MATLAB hay varias funciones para diseñar filtros y hasta una interfaz gráfica que puede ser de ayuda para los primeros pasos del diseño. Dadas unas especificaciones[1]: rizado en banda de paso, R_p, rizado en banda atenuada, R_s, anchura de la banda de paso, W_p, anchura de la banda de atenuación, W_s, número de coeficientes del filtro, N, se puede elegir un diseño basado en un filtro IIR o FIR.

- Filtros IIR.

[1] Según el tipo de filtro, no siempre son necesarias todas las especificaciones.

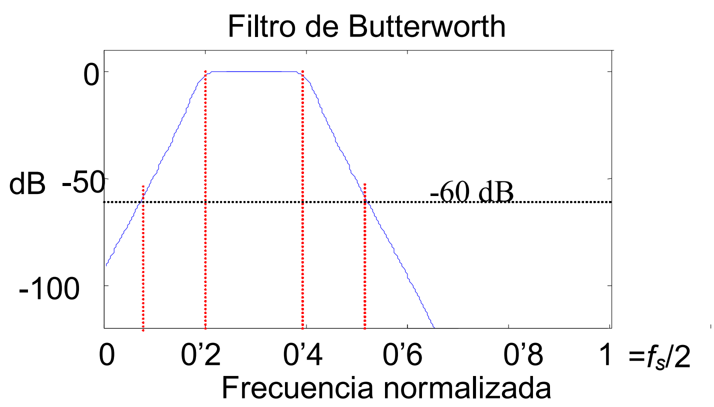
Si se desea un filtro FIR, MATLAB ofrece varias posibilidades: un filtro de Butterworth, un filtro de Tchebytcheff tipo I, un filtro de Tchebytcheff tipo II, o un filtro elíptico. Los comandos respectivos son:

```
[B,A] = butter(N,Wp,'ftype')

[A,A] = cheby1(N,Rp,Wn,'ftype')

[B,A] = cheby2(N,Rs,Wn,'ftype')

[B,A] = ellip(N,Rp,Rs,Wn,'ftype')
```

en donde *B* y *A* son los coeficientes del filtro (numerador y denominador, respectivamente) y *'ftype'* es una cadena de caracteres que denota el tipo del filtro (paso-bajo, paso-alto, etc.).

A veces puede ocurrir que las especificaciones sobre anchos de banda, rizado, atenuación, etc. sean tan restrictivas que no se puedan cumplir con el número de coeficientes indicado, *N*. Para saber el mínimo número de coeficientes que se necesitan para un determinado conjunto de especificaciones están los comandos `buttord`, `cheb1ord`, `elliporð`, etc.

Ejemplo: Diseño de un filtro paso-banda que responda a las especificaciones de la figura.



Como no conocemos el órden mínimo que ha de tener el filtro, utilizamos primero el comando `buttord`:

```
[n, Wn]=buttord([1000 2000]/5000, [500 2500]/5000, 1, 60)
```

lo que nos proporciona la siguiente salida (número mínimo de coeficientes y banda normalizada):

```
n=12,    Wn=[0.1951 0.4080]
```

Si ahora utilizamos esos datos como entrada del commando butter, obtendríamos los coeficientes *B* y *A* que definen el filtro que andábamos buscando.

```
[B, A]=butter(12, [0.1951 0.4080])
```

Nótese que en las especificaciones de banda de frecuencias en los comandos de diseño de filtros digitales de MATLAB, la frecuencia está normalizada a $F_s/2$, con F_s la frecuencia de muestreo. Así, hemos especificado, por ejemplo, una banda de paso de [1000 2000] Hz a la frecuencia de muestreo de 10 KHz como:

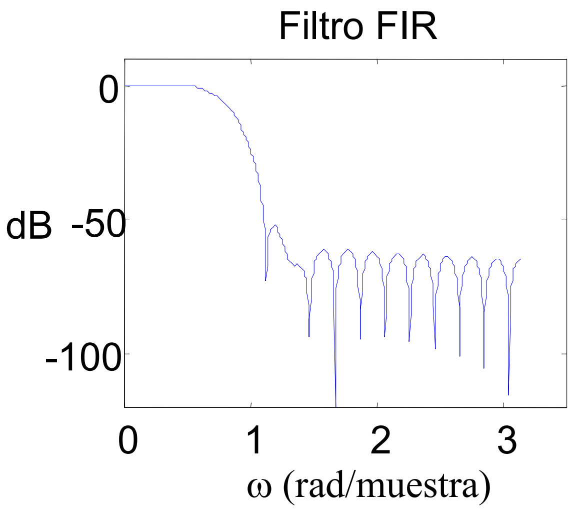
[1000 2000]/5000

- Filtros FIR.

Alternativamente se puede realizar el diseño mediante un filtro FIR. La ventaja de los filtros FIR es que proporcionan fase lineal en la banda de paso – lo que significa un retardo constante o bien una no distorsión de la señal de entrada en dicha banda- a cambio de un mayor número de coeficientes que si el diseño fuese hecho mediante un filtro IIR. MATLAB proporciona varios comandos para el diseño de filtros FIR. El más utilizado es:

```
B = fir1(N,Wn,'ftype')
```

Nótese que en este caso el filtro está totalmente especificado por los coeficientes B. Como siempre, dadas unas especificaciones habrá un número mínimo de coeficientes que las satisfarán; dicho número viene proporcionado por el comando `kaiserord`.



5. Comandos de MATLAB para obtener la respuesta en frecuencia del filtro.

```
H = freqz(B,A,ω)
```

Devuelve el vector H de números complejos, que es la respuesta frecuencial al filtro cuya función de transferencia en s viene dada por los coeficientes del polinomio del numerador almacenados en el vector fila B y los del polinomio del denominador, almacenados en el vector fila A.. La respuesta frecuencial se evalúa en los puntos de la circunferencia especificados por el vector ω en radianes/muestra. Más opciones en el Help de MATLAB.

```
H = freqz(B,A,F,Fs)
```

Devuelve el vector H de números complejos, que es la respuesta frecuencial al filtro cuya función de transferencia $H(z)$ viene dada por los coeficientes del polinomio del numerador almacenados en el vector fila B y los del polinomio del denominador, almacenados en el vector fila A. La respuesta en frecuencia frecuencial se evalúa en los puntos especificados por el vector de frecuencias F (en Hz), siendo la frecuencia de muestreo Fs Hz. Más opciones en el Help de MATLAB

```
plot(ω,abs(H))
```

```
plot(F,abs(H))
```

Dibuja la magnitud de la respuesta en frecuencia del filtro en función de la frecuencia digital o analógica.

```
plot(ω,unwrap(angle(H)))
```

```
plot(F,unwrap(angle(H)))
```

Dibuja la fase de la respuesta en frecuencia del filtro. La función `unwrap` hace que no haya discontinuidades de 2π debido a la obtención del valor principal de la fase con `angle`.

```
gd = grpdelay(B,A,ω)
```

```
gd = grpdelay(B,A,F,Fs)
```

Calcula retraso de grupo (–d{angle(H(ω))}/dω) de la función de Transferencia $H(\omega)$formada por los polinomios B y A. Se evalúa en los puntos especificados por ω en radianes/muestra

```
plot(ω,gd)
```

```
plot(F,gd)
```

Dibuja el retardo de grupo de la función de Transferencia Discreta en función de la frecuencia.

6. Bibliografía.

A. V. Oppenhein, R. W. Schafer: "Discrete-time signal processing", Capítulo 5, pp 240-311, Ed. Prentice-Hall, 2nd Edition.