Diseño de Filtros Digitales
(Parte 1)

- Terminología y Clasificación
- Filtros IIR
  - Diseño Filtros Analógicos (Butterworth, Chebyshev I y Chebyshev II).
  - Métodos de Transformación del plano *s* al plano *z*.
  - Diseño de Filtros IIR con MATLAB.

Terminología y Clasificación

- El término *filtro digital* lo entenderemos como cualquier procesamiento realizado en una señal de entrada digital.
- Un filtro digital es la implementación en hardware o software de una ecuación diferencia.
- Ventajas de los filtros digitales
  - Alta inmunidad al ruido
  - Alta precisión (limitada por los errores de redondeo en la aritmética empleada
  - Fácil modificación de las características del filtro
  - Muy bajo coste (y bajando)
- Por estas razones, los filtros digitales están reemplazando rápidamente a los filtros analógicos.

Terminología y Clasificación

- Clasificación de los Filtros Digitales
  - Filtros FIR (Finite Impulse Response)
    - Un filtro FIR de orden M se describe por la siguiente ecuación diferencia $y[n] = B_0 x[n] + B_1 x[n-1] + \cdots + B_M x[n-M]$, lo que da lugar a la función de transferencia $H(z) = B_0 + B_1 z^{-1} + \cdots + B_M z^{-M}$.
    - La secuencia $\{B_K\}$ son los coeficientes del filtro.
    - No hay recursión, es decir, la salida depende sólo de la entrada y no de valores pasados de la salida.
    - La respuesta es por tanto una suma ponderada de valores pasados y prresentes de la entrada. De ahí que se denomine Media en Movimiento (Moving Average)
    - La función de Transferencia tiene un denominador constante y sólo tiene ceros.
    - La respuesta es de duración finita ya que si la entrada se mantiene en cero durante M periodos consecutivos, la salida será también cero.

Terminología y Clasificación

- Filtros IIR (Infinite Impulse Response)

  Veremos dos variaciones de este tipo de filtros: AR y ARMA

  Filtros AR (Autoregresivo)

  - La ecuación diferencia que describe un filtro AR es $y[n]+A_1 y[n-1]+A_2 y[n-2]+\cdots+A_N y[n-N]=x[n]$, lo que da lugar a una función de transferencia

$$H(z)=\frac{1}{1+A_1 z^{-1}+A_2 z^{-2}+\cdots+A_N z^{-N}}$$

  - La función de transferencia contiene solo polos.
  - El filtro es recursivo ya que la salida depende no solo de la entrada actual sino además de valores pasados de la salida (Filtros con realimentación).
  - El término autoregresivo tiene un sentido estadístico en que la salida *y[n]* tiene una regresión hacia sus valores pasados.
  - La respuesta al impulso es normalmente de duración infinita, de ahí su nombre.

Terminología y Clasificación

Filtros ARMA (Autoregresivo y Media en Movimiento)

- Es el filtro más general y es una combinación de los filtros MA y AR descritos anteriormente. La ecuación diferencia que descibe un filtro ARMA de orden N es

$$y[n]+A_1 y[n-1]+A_2 y[n-2]+\cdots+A_N y[n-N]$$
$$=B_0 x[n]+B_1 x[n-1]+\cdots+B_M x[n-M]$$

Y la función de transferencia

$$H(z)=\frac{B_0+B_1 z^{-1}+\cdots+B_M z^{-M}}{1+A_1 z^{-1}+\cdots+A_N z^{-N}}$$

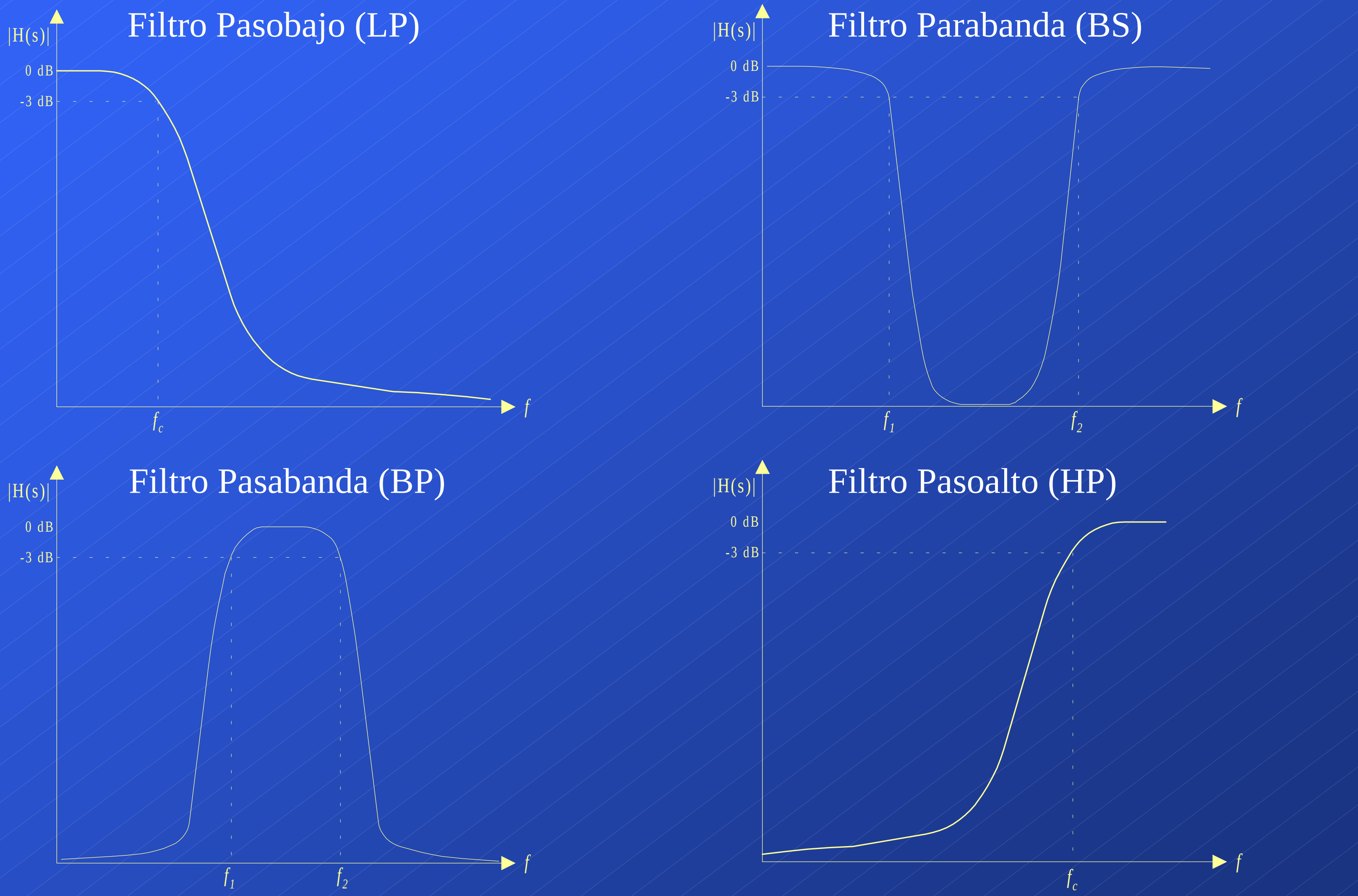
- Un filtro de este tipo se denota por ARMA(N,M), es decir es Autoregresivo de orden N y Media en Movimiento de orden M.
- Su respuesta a impulso es también de duración infinita y por tanto es un filtro del tipo IIR.

Terminología y Clasificación

Clasificación de los Filtros Digitales

Ecuación Diferencia	***Tipo de Filtro***
$y[n] = \sum_m B_m x[n-m]$	*FIR(Finite Impulse Response), No Recursivo, Moving Average (MA) orden M, Todo ceros*
$\sum_k A_k y[n-k] = x[n]$	*IIR (Infinite Impulse Response), Recursivo, Autoregresivo(AR) orden N, Todo polos*
$\sum_k A_k y[n-k] = \sum_m B_m x[n-m]$	*IIR Recursivo, ARMA(N,M), Polos y Ceros*

Terminología y Clasificación



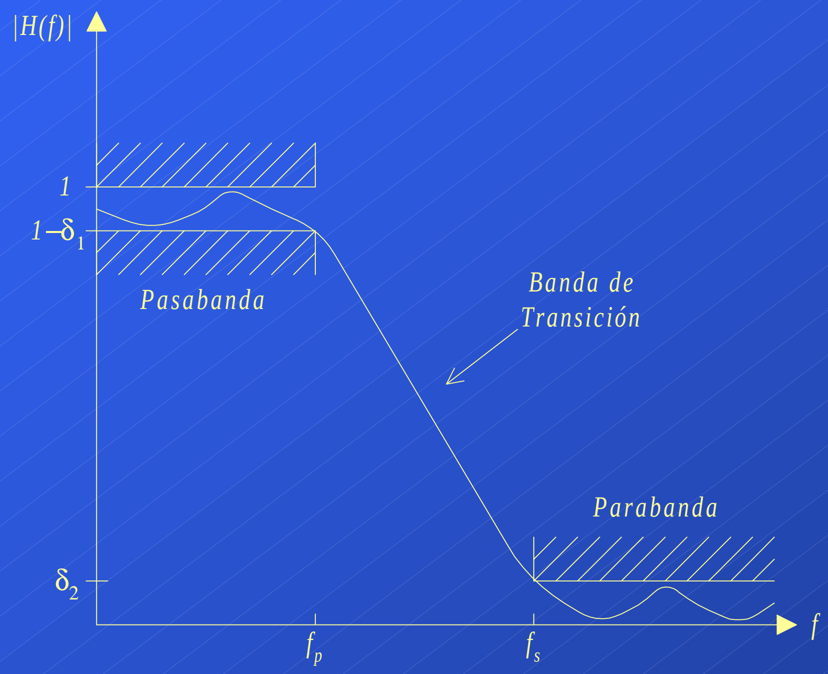
Proceso de Diseño de Filtros Digitales

- El proceso de diseño de un filtro digital requiere tres pasos
  - Establecer las especificaciones del filtro para unas determinadas prestaciones. Estas especificaciones son las mismas que las requeridas por un filtro analógico : frecuencias de parabanda y pasabanda, atenuaciones, ganancia dc, etc.
  - Determinar la función de transferencia que cumpla las especificaciones.
  - Realizar la función de transferencia en hardware o software.
- ¿IIR o FIR?
  - Los filtros IIR producen en general distorsión de fase, es decir la fase no es lineal con la frecuencia.
  - Los filtros FIR son de fase lineal.
  - El orden de un filtro IIR es mucho menor que el de un filtro FIR para una misma aplicación.
  - Los filtros FIR son siempre estables.

Filtros IIR-Método I

- Técnicas de diseño de filtros IIR
  - Mediante métodos de diseño analógico, seguido de una transformación del plano *s* al plano *z* (Método I).
  - Diseñar un prototipo de filtro pasobajo digital y hacer las oportunas transformaciones (Método II)
- Método I
  - Discutiremos este método para el diseño de filtros pasobajo. Empezaremos discutiendo el proceso de diseño de filtros analógicos para luego transformarlo al dominio discreto.
  - El diseño analógico se realiza a partir de unas especificaciones como las dadas en la figura.
    - δ_1 es el rizado de pasabanda.
    - δ_2 es el rizado de parabanda
    - f_p es la frecuencia límite de pasabanda.
    - f_s es la frecuencia límite de parabanda.

Filtros IIR-Método I



- Partimos de un prototipo de filtro pasobajo normalizado en el que usamos una frecuencia ν normalizada. Para otro tipo de filtro se requerirá la consiguiente transformación de frecuencia. Para ese filtro pasobajo normalizado la función de Transferencia es

$$|H(\nu)|^2 = \frac{1}{1 + L_n^2(\nu)}$$

donde $L_n(\nu)$ es un polinomio de grado n.

Filtros IIR-Método I-Aprox. Butterworth

- El objetivo del diseño de un filtro es encontrar $L_n(\nu)$ que mejor cumple las especificaciones. Para ello se utilizan algunas aproximaciones (Butterworth, Chebyshev, etc).
- Etapas del diseño
  - ① Normalizar la frecuencia de acuerdo a las especificaciones.
  - ② Determinar el orden del prototipo de filtro pasobajo.
  - ③ Determinar la función de Transferencia normalizada.
  - ④ Desnormalizar a través de las transformaciónes de frecuencia en ③ y ①.
- Aproximación de Butterworth
  - Consiste en hacer $L_n(\nu)=\varepsilon\nu^n$. Esta aproximación es tal que

$$L_n(0)=0, L'_n(0)=0, \cdots, L_n^{n-1}(0)=0$$

  - Por tanto,

$$|H(\nu)|^2=\frac{1}{1+\varepsilon^2\nu^{2n}}$$

  - ① Normalizamos las frecuencias por la frecuencia límite de pasabanda f_p, de forma que $\nu_p=1$ y $\nu_s=f_s/f_p$.

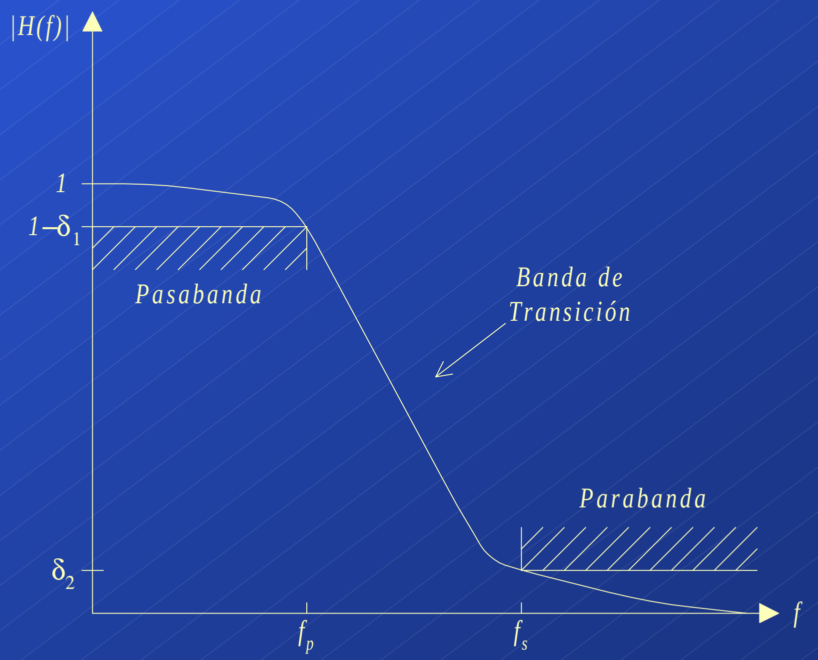
Filtros IIR-Método I-Aprox. Butterworth

② A partir de δ_1 (en $v=1$) y δ_2 (en $v=v_s$), podemos calcular los valores de ε y n.

$$\delta_1 = 10\log\left|H(v)\right|^2_{v=1} = 10\log\frac{1}{1+\varepsilon^2} = -10\log\left(1+\varepsilon^2\right) \Rightarrow \boxed{\varepsilon^2 = 10^{-0.1\delta_1} - 1}$$

$$\delta_2 = 10\log\left|H(v)\right|^2_{v=v_s} = 10\log\frac{1}{1+\varepsilon^2 v_s^{2n}} = -10\log\left(1+\varepsilon^2 v_s^{2n}\right)$$

$$\boxed{n = \frac{\log\left[\dfrac{10^{-0.1\delta_2}-1}{10^{-0.1\delta_1}-1}\right]^{\frac{1}{2}}}{\log v_s}}$$



Filtros IIR-Método I-Aprox. Butterworth

- Ejemplo: Diseñar un filtro de Butterworth de pasobajo con las siguientes especificaciones: una atenuación de pasabanda que sea menor que *1dB* a *f=0.6366 Hz*, una atenuación de parabanda que sea mayor que *20dB* para *f=1.2732 Hz*.
  - Frecuencia normalizada $\nu=f/f_p=f/0.6366$.
  - En $\nu=1$ $\quad -1dB = 10\log\dfrac{1}{1+\varepsilon^2} \Rightarrow \varepsilon^2 = 10^{0.1}-1 = 1.2589-1 = 0.2589$
  - En $\nu=1.2732/0.6366=2$

$$-20dB = 10\log\frac{1}{1+0.2589\cdot 2^{2n}} \Rightarrow n = \frac{\log\left[\dfrac{10^2-1}{0.2589}\right]^{\frac{1}{2}}}{\log 2} = 4.289$$

Tomamos como orden del filtro el entero más cercano por arriba *n=5*.

② Filtro de Butterworth Normalizado : se normaliza con respecto a la frecuencia ν_3 ($|H(\nu_3)|^2=$ ½).

$$\left|H_N(\nu)\right|^2 = \left|H(\nu/\nu_3)\right|^2 = \frac{1}{1+(\nu/\nu_3)^{2n}} = \frac{1}{1+\nu_N^{2n}}$$

Filtros IIR-Método I-Aprox. Butterworth

- Función de Transferencia Normalizada. Se trata de determinar $H_N(s)$ a partir de $|H_N(\nu)|^2$.

$$|H_N(\nu_N)|^2 = H(\nu_N)\cdot H^*(\nu_N) = H(\nu_N)\cdot H(-\nu_N) \Rightarrow H(s)\cdot H(-s)\big|_{s=j\nu_N} = H(\nu_N)\cdot H(-\nu_N) = |H_N(\nu_N)|^2$$

$$H(s)H(-s) = |H_N(\nu_N)|^2_{\nu_N^2=-s^2} = \frac{1}{1+\left(-s^2\right)^n}$$

  - Reemplazamos ν_N^2 por $-s^2$.
  - Los polos los calculamos de

$$1+\left(-s^2\right)^n = 0 \Rightarrow \left(-s^2\right)^n = -1 \Rightarrow (-1)^n p_k^{2n} = -1$$

$$\exp(-j\pi n)p_k^{2n} = \exp[j(2k-1)\pi] \quad k = 1,2,\cdots,2n$$

$$p_k = \exp\left[j\left(\theta_k + \tfrac{\pi}{2}\right)\right] \quad \text{donde} \quad \theta_k = \frac{(2k-1)\pi}{2n} \quad k = 1,2,\cdots,2N$$

$$p_k = \sigma_k + j\omega_k = \cos\left(\theta_k + \tfrac{\pi}{2}\right) + j\sin\left(\theta_k + \tfrac{\pi}{2}\right) = -\sin(\theta_k) + j\cos(\theta_k) \Rightarrow$$

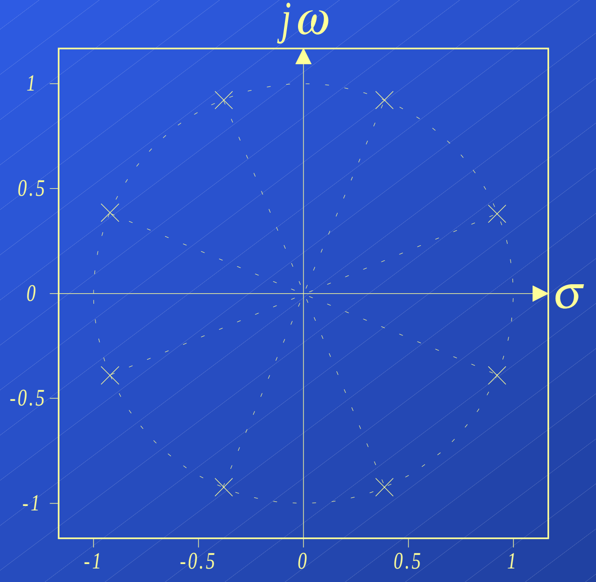
$$\sigma_k = -\sin(\theta_k) \qquad \omega_k = \cos(\theta_k)$$

$$|p_k| = \left(\sigma_k^2 + \omega_k^2\right)^{\frac{1}{2}} = 1$$

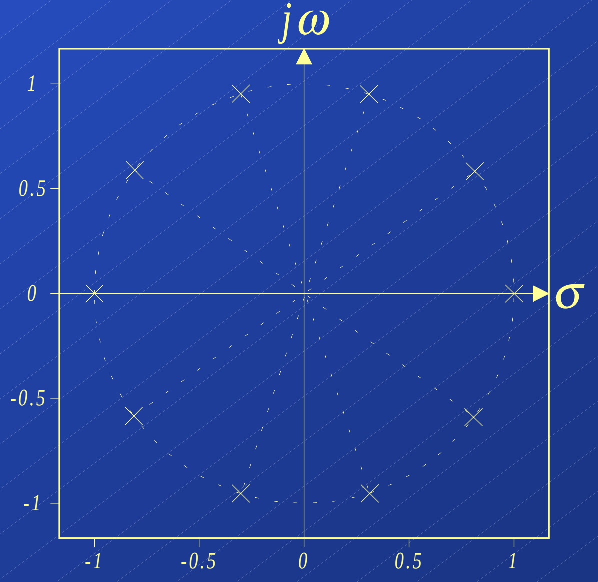
Filtros IIR-Método I-Aprox. Butterworth

- Estos resultados muestran que:
  - Los polos normalizados están sobre un círculo de radio 1 en el plano *s*.
  - Los polos están equiespaciados π/n radianes con $\theta_k=(2k-1)\pi/2n$, donde θ_k se mide con respecto al eje positivo del eje $j\omega$.
  - Los polos nunca estarán sobre el eje $j\omega$ ($2k-1$ nunca puede ser par).
  - Si n es impar, siempre hay un par de polos reales en $s=\pm 1$.

Polos de Butterworth : n=4



Polos de Butterworth : n=5



Filtros IIR-Método I-Aprox. Butterworth

- De los *2n* polos de que consta *H(s)H(-s)* sólo estamos interesados en los polos de la parte izquierda del plano *s*, que son los que dan estabilidad al filtro. De las figuras se puede observar que cada polo tiene su conjugado, excepto en el caso de que *n* sea impar donde tenemos un polo adicional en *s=-1*. Por tanto

$$\begin{aligned}(s+p_k)(s+p_k^*) &= [s-\sin(\theta_k)+j\cos(\theta_k)][s-\sin(\theta_k)-j\cos(\theta_k)] \\ &= s^2-2\sin(\theta_k)s+1\end{aligned}$$

$$H_N(s)=\frac{1}{Q_N(s)},\quad Q_N(s)=\begin{cases}(s+1)\prod_{k=1}^{(n-1)/2}\left[s^2+s(2\sin\theta_k)+1\right] & n\ impar\\ \prod_{k=1}^{n/2}\left[s^2+s(2\sin\theta_k)+1\right] & n\ par\end{cases}$$

- El polinomio $Q_N(s)$ viene en tablas para cada valor de *n*, por lo que solo tenemos que determinar el valor de *n*, ir a las tablas para obtener $H_N(s)$ y desnormalizar para determinar la *H(s)* de nuestro filtro.

Filtros IIR-Método I-Aprox. Butterworth

④ Desnormalización : Si $|H(\nu)|^2=|H_N(\nu/\nu_3)|^2$, entonces $H(s)=H_N(s/\nu_3)$. Si desnormalizamos $H(s)$ a $H_A(s)=H(s/\omega_p)$, $H_A(s)$ cumple las especificaciones dadas. Estos es equivalente a desnormalizar directamente $H_N(s)$ a $H_A(s)=H_N(s/\omega_p\nu_3)$.

- Ejemplo: Calcular la función de Transferencia del filtro del ejemplo anterior.
  - En el ejemplo teníamos $n=5$, $f_p=0.6366$, $f_s=1.2732$ y $\varepsilon^2=0.2589$.
  - Calculamos ν_3. $\left|H(\nu_3)\right|^2 = \dfrac{1}{1+0.2589\nu_3^{10}} = \dfrac{1}{2} \Rightarrow \nu_3 = 1.145$
  - De las tablas se obtiene

$$H_N(s) = \frac{1}{1+3.236s+5.236s^2+5.236s^3+3.236s^4+s^5}$$

  - $H_A(s)=H_N(s/\omega_p\nu_3)= H_N(s/4.578)$

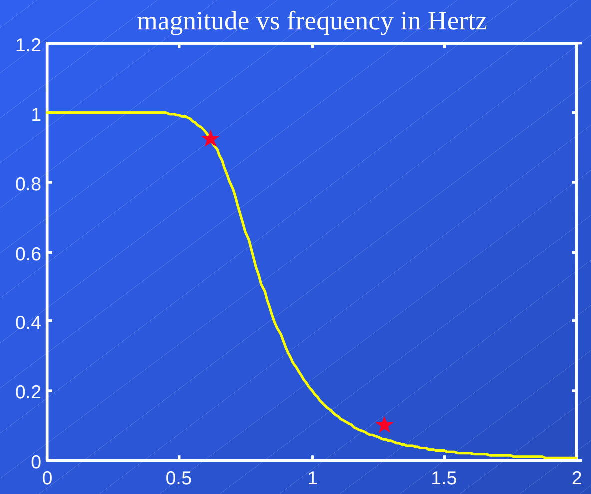
$$H_A(s) = H_N(s/4.578) =$$

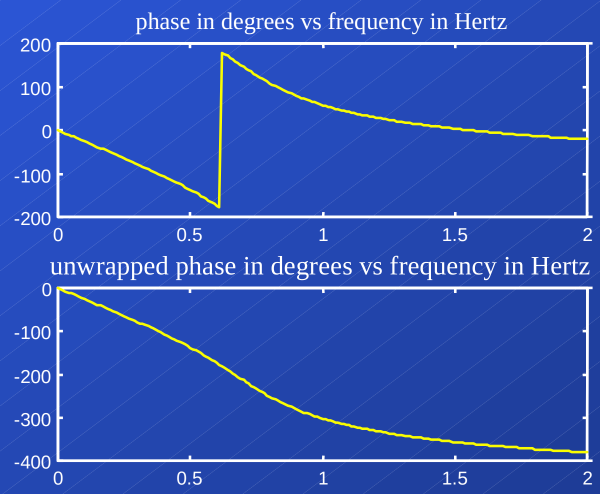
$$= \frac{1}{1+3.236(s/4.578)+5.236(s/4.578)^2+5.236(s/4.578)^3+3.236(s/4.578)^4+(s/4.578)^5}$$

$$= \frac{1}{1+0.7068s+0.2498s^2+0.0546s^3+0.0074s^4+4.973\cdot 10^{-4}s^5}$$

Filtros IIR-Método I-Aprox. Butterworth

$$H_A(s)=\frac{2010.84}{2010.84+1421.26s+502.31s^2+109.79s^3+14.88s^4+s^5}$$





- Filtro Pasoalto de Butterworth
  - Se hace la transformación $\nu \rightarrow 1/\nu$, lo que da lugar a $|H_{HP}(\nu)|^2=|H_{LP}(1/\nu)|^2$. También equivale a hacer $1-|H_{LP}(\nu)|^2$

$$1-\left|H_{LP}(\nu)\right|^2=1-\frac{1}{1+\nu^{2n}}=\frac{\nu^{2n}}{1+\nu^{2n}}=\frac{1}{1+(1/\nu)^{2n}}=\left|H_{LP}(1/\nu)\right|^2$$

Filtros IIR-Método I-Aprox. Chebyshev I

- Aproximación de Chebyshev I $|H(\nu)|^2 = \dfrac{1}{1+L_n^2(\nu)} = \dfrac{1}{1+\varepsilon^2 T_n^2(\nu)}$

donde $T_n(\nu)$ es el polinomio de Chebyshev de orden n.

Polinomios de Chebyshev $T_n(x)=cosh(ncosh^{-1}x)$	
Orden n	**$T_n(x)$**
0	1
1	x
2	$2x^2-1$
3	$4x^3-3x$
4	$8x^4-8x^2+1$
5	$16x^5-20x^3+5x$
6	$32x^6-48x^4+18x^2-1$
7	$64x^7-112x^5+56x^3-7x$
8	$128x^8-256x^6+160x^4-32x^2+1$

Filtros IIR-Método I-Aprox. Chebyshev I

- Proceso de diseño de los filtros de Chebyshev I: Partiremos del filtro prototipo de pasobajo mediante la apropiada transformación de frecuencia. Tendremos las especificaciones para el filtro en forma de las frecuencias límites de pasabanda y parabanda así como sus respectivas atenuaciones.
- En los filtros de Chebyshev I se especifica también para un determinado rizado en la banda pasabanda (Figura).
- Con esas especificaciones calculamos n y ε.
- Determinamos la Función de Transferencia Normalizada en v_3.
- Desnormalizamos y obtenemos $H_A(s)$.
- Ejemplo: Diseñar el filtro de Chebyshev para las siguientes especificaciones : $A_p \leq 1dB$ para $\omega \leq 4$, $A_s \geq 20dB$ para $\omega \geq 8$.
  - Normalizamos en ω_p , $v_p=1$, $v_s=\omega_s/\omega_p=2$.

Filtros IIR-Método I-Aprox. Chebyshev I

$$-A_p = 10\log\frac{1}{1+\varepsilon^2} \Rightarrow \varepsilon^2 = 10^{0.1A_p} - 1 = 0.2589$$

$$-A_s = 10\log\frac{1}{1+\varepsilon^2\cosh^2\left(n\cosh^{-1}\nu_s\right)} = 10\log\frac{1}{1+\varepsilon^2\cosh^2\left(n\cosh^{-1}\nu_s\right)}$$

$$n = \frac{\cosh^{-1}\left[\left(10^{0.1A_s}-1\right)/\varepsilon^2\right]^{\frac{1}{2}}}{\cosh^{-1}\nu_s} = \frac{3.6657}{1.3169} = 2.7834 \rightarrow n = 3$$

$$\frac{1}{2} = \frac{1}{1+\varepsilon^2\cosh^2 n\cosh^{-1}\nu_3} \Rightarrow \nu_3 = \cosh\left[\tfrac{1}{n}\cosh^{-1}\left(\tfrac{1}{\varepsilon}\right)\right] = 1.0949$$

- La magnitud del rizado en la pasabanda es $1-\dfrac{1}{\sqrt{1+\varepsilon^2}} = 0.1087$
- La normalización de la función de transferencia se realiza en $\nu_p=1$.
- Procedemos de la misma forma que en el filtro de Butterworth

$$H_N(s)H_N(-s) = \left|H(\nu)\right|^2_{\nu=\frac{s}{j}} = \frac{1}{1+\varepsilon^2 T_n^2(s/j)}$$

Filtros IIR-Método I-Aprox. Chebyshev I

Los polos de $H_N(s)$ *se obtienen de*

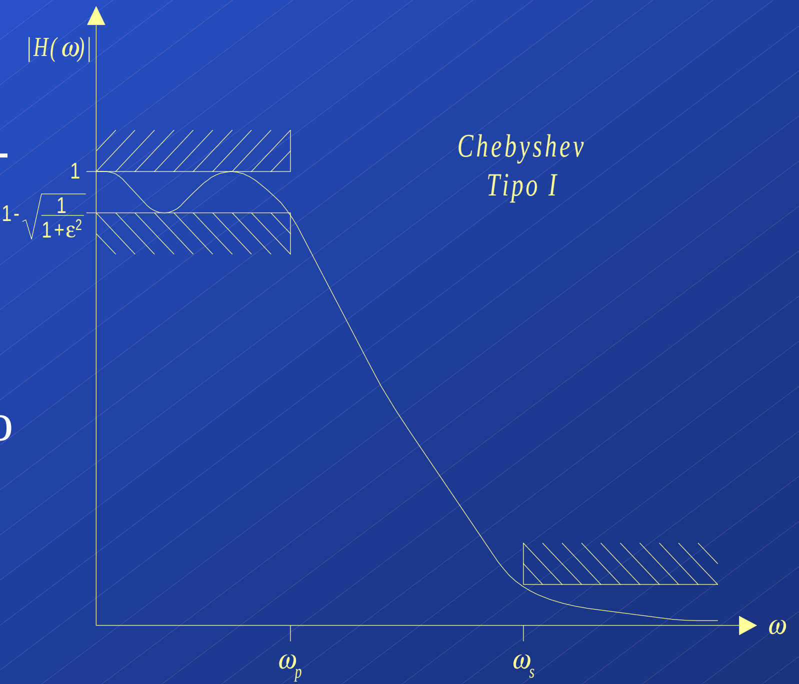
$$1+\varepsilon^2\cos^2\left[n\cos^{-1}(s/j)\right]=0 \Rightarrow \cos\left[n\cos^{-1}(s/j)\right]=\pm j/\varepsilon$$

Llamamos $z=\theta+j\alpha=\cos^{-1}(s/j)$

$$\cos(nz)=\cos(n\theta+jn\alpha)=\cos(n\theta)\cosh(n\alpha)-j\sin(n\theta)\sinh(n\alpha)=\pm j/\varepsilon$$

$$\cos(n\theta)\cosh(n\alpha)=0 \qquad (1)$$

$$\sin(n\theta)\sinh(n\alpha)=\pm 1/\varepsilon \qquad (2)$$



Ya que *cosh(nα)≥1* para todo *nα*, la ecuación (1) nos dice que

$$\cos(n\theta)=0 \Rightarrow \theta_k=\frac{(2k-1)\pi}{2n} \qquad k=1,2,\cdots,2n$$

Para estos valores de θ_k, *sin(nθ)=1*, por lo que la ecuación (2) queda

$$\sinh(n\alpha)=1/\varepsilon \Rightarrow \alpha=\tfrac{1}{n}\sinh^{-1}\left(\tfrac{1}{\varepsilon}\right)$$

Filtros IIR-Método I-Aprox. Chebyshev I

$$s/j = \cos z$$

$$s = j\cos z = j\cos(\theta_k + j\alpha) = \sin(\theta_k)\sinh(\alpha) + j\cos(\theta_k)\cosh(\alpha)$$

Los polos estarán a la izquierda del plano s para garantizar la estabilidad del filtro

$$p_k = -\sin(\theta_k)\sinh(\alpha) + j\cos(\theta_k)\cosh(\alpha) \qquad k = 1,2,\cdots,n$$

En el ejemplo anterior teniamos $\varepsilon^2=0.2589$, $n=3$

$$\alpha = \tfrac{1}{n}\sinh^{-1}\left(\tfrac{1}{\varepsilon}\right) = \tfrac{1}{3}\sinh^{-1}\left(\tfrac{1}{\sqrt{0.2589}}\right) = 0.476$$

$$\theta_k = \tfrac{(2k-1)\pi}{2n} = \tfrac{(2k-1)\pi}{6} \qquad k = 1,2,3$$

$$s_1 = -\sinh(0.476)\sin\left(\tfrac{\pi}{6}\right) + j\cosh(0.476)\cos\left(\tfrac{\pi}{6}\right) = -0.2471 + j0.966$$

$$s_2 = -\sinh(0.476)\sin\left(\tfrac{\pi}{2}\right) + j\cosh(0.476)\cos\left(\tfrac{\pi}{2}\right) = -0.4942$$

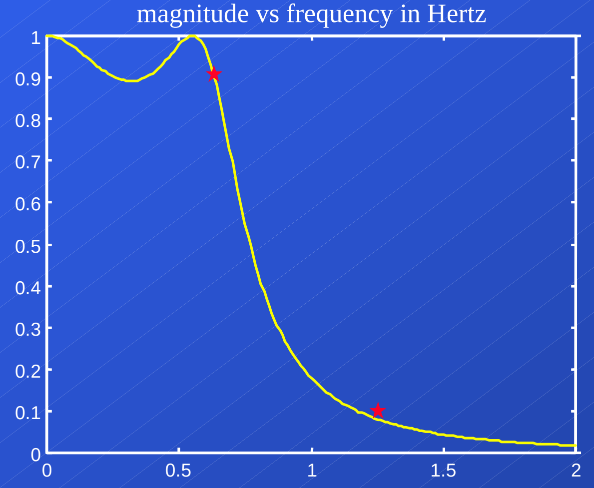
$$s_3 = -\sinh(0.476)\sin\left(\tfrac{5\pi}{6}\right) + j\cosh(0.476)\cos\left(\tfrac{5\pi}{6}\right) = -0.2471 - j0.966$$

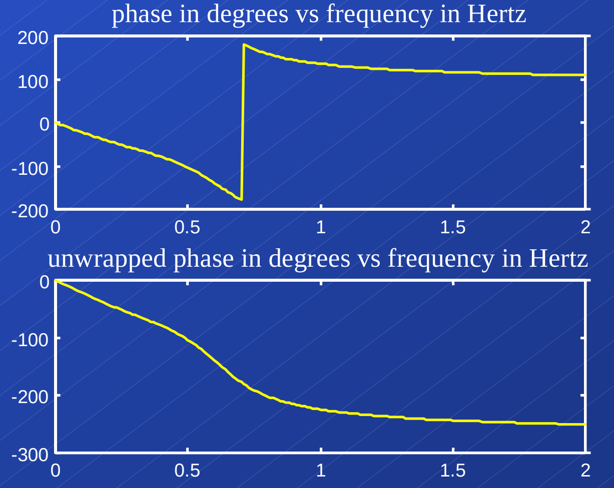
$$Q_N(s) = (s - s_1)(s - s_2)(s - s_3) = (s + 0.4942)\left(s^2 + 0.4942s + 0.9492\right)$$

Filtros IIR-Método I-Aprox. Chebyshev I

$$H_N(s) = \frac{Q_N(0)}{(s+0.4942)\left(s^2+0.4942s+0.9492\right)} = \frac{0.4913}{s^3+0.9883s^2+1.2384s+0.4913}$$

$$H_A(s) = H_N(s/4) = \frac{31.4436}{31.4436+19.8145s+3.9534s^2+s^3}$$





Filtros IIR-Método I-Aprox. Chebyshev I

- Ejemplo: Diseñar un filtro pasabanda de Chebyshev con las siguientes características:
  - Bordes de las bandas : $[\omega_1\ \omega_2\ \omega_3\ \omega_4]=[0.89\ 1.019\ 2.221\ 6.155]$
  - Máxima atenuación en la pasabanda $A_p=2dB$.
  - Mínima atenuación en la parabanda $A_s=20dB$.

  $\omega_x=(\omega_2\omega_3)^{1/2}=(1.019\ 2.221)^{1/2}=1.5045$

  $\omega_4=\omega_x^2/\omega_1=2.54.\ B_w=\omega_3-\omega_2=1.202$

  $\boldsymbol{\omega_{Ap}=1,\ \omega_{As}=(\omega_4-\omega_1)/B_w=1.3727}$

  Con estas especificaciones para el prototipo de filtro pasobajo calculamos ε y n. $\varepsilon^2=10^{0.1A_p}-1=10^{0.1\cdot 2}-1=0.5849$

$$n=\frac{\cosh^{-1}\left[\frac{10^{0.1A_s}-1}{\varepsilon^2}\right]^{\frac{1}{2}}}{\cosh^{-1}\left(\frac{\omega_s}{\omega_p}\right)}=\frac{\cosh^{-1}\left[\frac{99}{0.5849}\right]^{\frac{1}{2}}}{\cosh^{-1}\left(\frac{1.3754}{1}\right)}=3.871\rightarrow n=4$$

Filtros IIR-Método I-Aprox. Chebyshev I

De aquí obtenemos los polos del filtro aplicando las fórmulas de los polos vistas anteriormente.

$$\alpha = \tfrac{1}{n}\sinh^{-1}\left(\tfrac{1}{\varepsilon}\right) = \tfrac{1}{4}\sinh^{-1}\left(\tfrac{1}{\sqrt{0.5849}}\right) = 0.2707$$

$$\theta_k = \tfrac{(2k-1)\pi}{2n} = \tfrac{(2k-1)\pi}{8} \qquad k = 1,2,3,4$$

$$s_1 = -\sinh(0.2707)\sin\left(\tfrac{\pi}{8}\right) + j\cosh(0.2707)\cos\left(\tfrac{\pi}{8}\right) = -0.1049 + j0.958$$

$$s_2 = -\sinh(0.2707)\sin\left(\tfrac{3\pi}{8}\right) + j\cosh(0.2707)\cos\left(\tfrac{3\pi}{8}\right) = -0.2532 + j0.3968$$

$$s_3 = -\sinh(0.2707)\sin\left(\tfrac{5\pi}{8}\right) + j\cosh(0.2707)\cos\left(\tfrac{5\pi}{8}\right) = -0.2532 - j0.3968$$

$$s_4 = -\sinh(0.2707)\sin\left(\tfrac{7\pi}{8}\right) + j\cosh(0.2707)\cos\left(\tfrac{7\pi}{8}\right) = -0.1049 - j0.958$$

$$Q_N(s) = (s-s_1)(s-s_2)(s-s_3)(s-s_4) = \left(s^2 + 0.2098s + 0.9287\right)\left(s^2 + 0.5064s + 0.2216\right)$$

$$= s^4 + 0.7162s^3 + 1.2565s^2 + 0.5167s + 0.2058$$

Como n es par

$$K = \frac{Q_N(0)}{\sqrt{1+\varepsilon^2}} = 0.1634 \rightarrow H_N(s) = \frac{0.1634}{s^4 + 0.7162s^3 + 1.2565s^2 + 0.5167s + 0.2058}$$

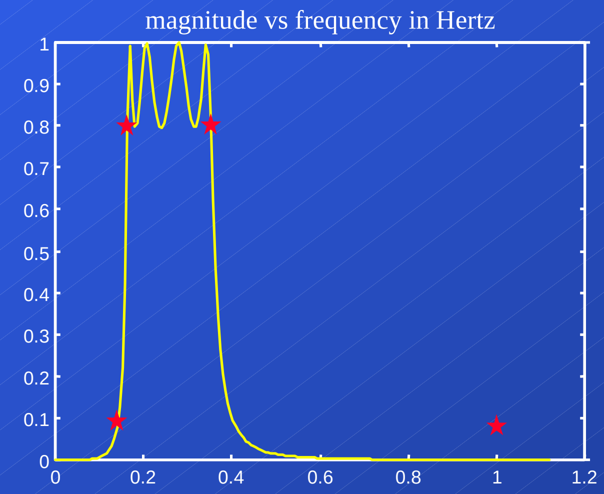
Filtros IIR-Método I-Aprox. Chebyshev I

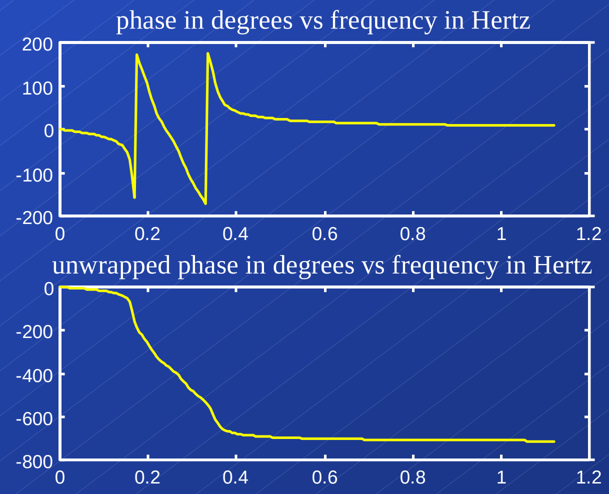
Ahora hay que hacer la transformación para el filtro pasabanda

$$s \rightarrow \frac{s^2 + \omega_x^2}{sB_w} = \frac{s^2 + 2.2635}{1.202s}$$

$$H_A(s) = \frac{0.1148}{\left[\frac{s^2 + 2.2635}{1.202s}\right]^4 + 0.7162\left[\frac{s^2 + 2.2635}{1.202s}\right]^3 + 1.2565\left[\frac{s^2 + 2.2635}{1.202s}\right]^2 + 0.5167\left[\frac{s^2 + 2.2635}{1.202s}\right] + 0.2058} =$$

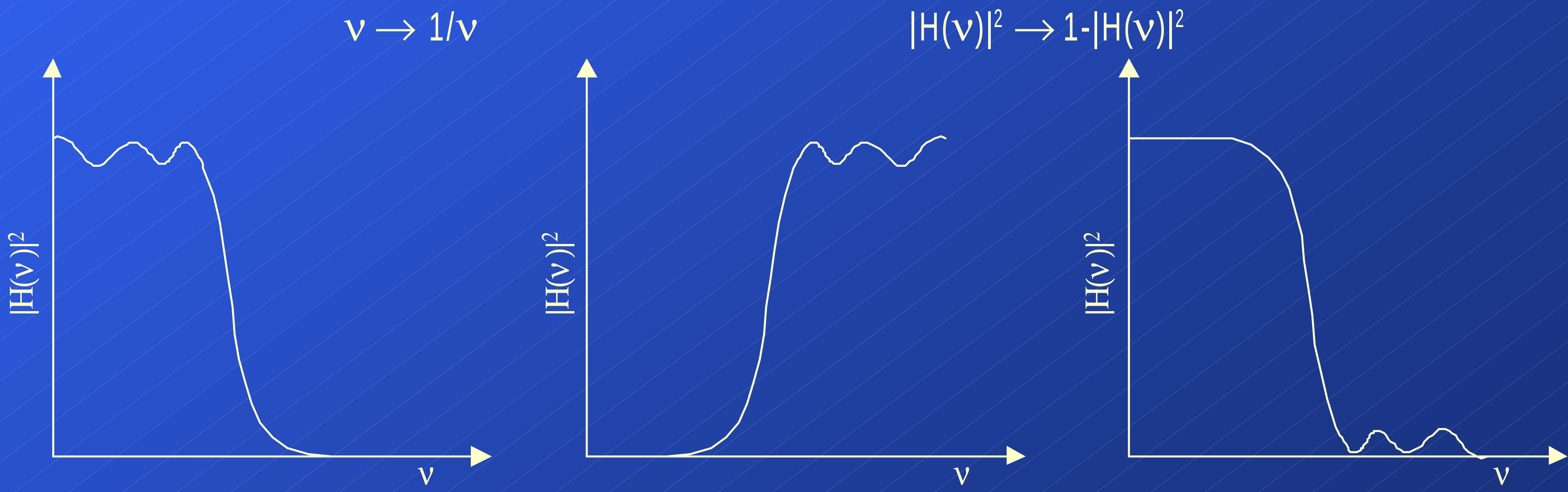
$$= \frac{0.34s^4}{s^8 + 0.8604s^7 + 10.87s^6 + 6.75s^5 + 39.39s^4 + 15.27s^3 + 55.69s^2 + 9.99s + 26.25}$$





Filtros IIR-Método I-Aprox. Chebyshev II

- Aproximación de Chebyshev II
  - Se desea disponer de un filtro con una fuerte transición como es el caso de filtro de Chebyshev tipo I, pero que a la vez tenga una respuesta lo más plana posible en la pasabanda. Esto se logra transfiriendo el rizado del pasabanda en el filtro de Chebyshev I al parabanda y viceversa. De esta forma, mediante una transformación de frecuencia, logramos un filtro con unas características inversas a las del filtro Chebyshev I.
  - Hacemos la Transformación $\nu \rightarrow 1/\nu$, partiendo del filtro de Chebyshev II.



Filtros IIR-Método I-Aprox. Chebyshev II

- Chebyshev I $\rightarrow |H(\nu)|^2 = \dfrac{1}{1+\varepsilon^2 T_n^2(\nu)}$ Mediante la Transformación ν→*1/ν* obtenemos

$$|H(1/\nu)|^2 = \frac{1}{1+\varepsilon^2 T_n^2(1/\nu)}$$

$$|H(\nu)|^2 = 1-|H(1/\nu)|^2 = \frac{\varepsilon^2 T_n^2(1/\nu)}{1+\varepsilon^2 T_n^2(1/\nu)} = \frac{1}{1+\left[1/\varepsilon^2 T_n^2(1/\nu)\right]} = \frac{1}{1+L_n^2(\nu)}$$

- La función $L_n^2(\nu)$ es ahora una función racional y no polinomial.

$$L_2^2(\nu) = \frac{1}{\varepsilon^2 T_2^2(1/\nu)} = \frac{1}{\varepsilon^2\left[2\left(\frac{1}{\nu}\right)^2-1\right]^2} = \frac{1}{\varepsilon^2\left[\frac{2-\nu^2}{\nu^2}\right]^2} = \frac{1}{\varepsilon^2\,\frac{4-4\nu^2+\nu^4}{\nu^4}} = \frac{\nu^4}{\varepsilon^2\left(4-4\nu^2+\nu^4\right)}$$

Filtros IIR-Método I-Aprox. Chebyshev II

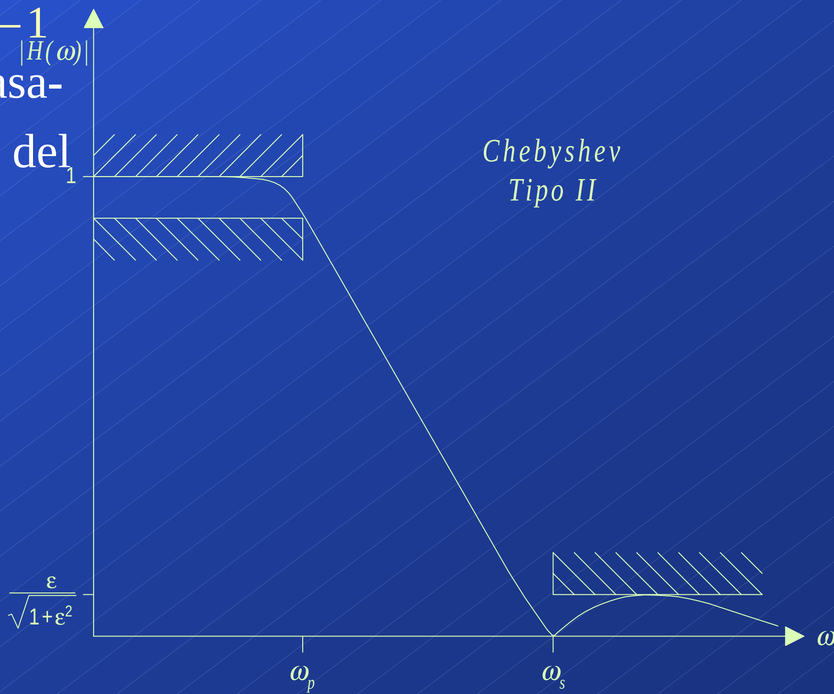
- Proceso de diseño para los filtros Chebyshev II
  - Partimos de las especificaciones que aplicaremos de acuerdo al prototipo de filtro pasobajo.
  - Normalización de frecuencias a la frecuencia de <u>parabanda</u> $\nu_s=1$, $\nu_p=\omega_p/\omega_s$. La atenuación en parabanda A_s la utilizamos para calcular ε^2.

$$A_s = 10\log\left(1+1/\varepsilon^2\right) \Rightarrow \varepsilon^2 = \frac{1}{10^{0.1A_s}-1}$$

  - Empleamos la atenuación en la pasa-banda A_p, para calcular el orden n del filtro.

$$A_p = 10\log\left[1+\frac{1}{\varepsilon^2\cosh^2\left(n\cosh^{-1}\left(\frac{1}{\nu_p}\right)\right)}\right]$$

$$n = \frac{\cosh^{-1}\left[\frac{1}{\left(10^{0.1A_p}-1\right)\varepsilon^2}\right]^{\frac{1}{2}}}{\cosh^{-1}\left(\omega_s/\omega_p\right)}$$



Filtros IIR-Método I-Aprox. Chebyshev II

- El valor de ε^2 está relacionado con el rizado en la parabanda. En esa zona ($\nu_s=1$), el valor $|H(\nu)|^2$ es $\left|H(1)\right|^2 = \dfrac{1}{1+1/\varepsilon^2} = \dfrac{\varepsilon^2}{1+\varepsilon^2}$
- Al contrario que en el filtro Chebyshev I, $|H(0)|^2$ es siempre igual a *1*.
- Función de Transferencia normalizada $H_N(s)$:

$$H(s)H(-s) = \left|H(\nu)\right|^2_{\nu=s/j} = \left.\frac{\varepsilon^2 T_n^2(1/\nu)}{1+\varepsilon^2 T_n^2(1/\nu)}\right|_{\nu=s/j}$$

Esta vez no solo tendremos polos en $H_N(s)$ sino también ceros.

- Los polos se calculan de forma parecida al filtro Chebyshev I,

$$\alpha = \tfrac{1}{n}\sinh^{-1}\left(\tfrac{1}{\varepsilon}\right)$$

$$\cos(n\theta) = 0 \Rightarrow \theta_k = \frac{(2k-1)\pi}{2n} \qquad k = 1,2,\cdots,2n$$

$$1/p_k = -\sin(\theta_k)\sinh(\alpha) + j\cos(\theta_k)\cosh(\alpha) \qquad k = 1,2,\cdots,n$$

Filtros IIR-Método I-Aprox. Chebyshev II

- Los ceros se calculan a partir de los ceros del polinomio de Chebyshev $T_n(1/\nu)=\cos[n\cos^{-1}(1/\nu)]$. Haciendo el cambio de variable $1/\nu=\cos\theta$.

$$T_n(\theta)=\cos[n\theta]=0 \Rightarrow \theta_k=\frac{(2k-1)\pi}{2n} \qquad k=1,2,\cdots,n$$

Los ceros estarán por tanto en $1/\nu_k=\cos\theta_k \Rightarrow \nu_k=\sec\theta_k$

Sustituyendo ν_k por s/j, obtengo los ceros de la función de Transferencia.

$z_k=j\sec\theta_k \quad k=1,2,\cdots,n$. Estos ceros tienen simetría conjugada, por lo que $z_k=\pm j\sec\theta_k=\pm j\omega_k \quad k=1,2,\cdots\mathrm{int}\left(\tfrac{1}{2}n\right)$

- La forma final de la función de Transferencia Normalizada $H_N(s)$ es

$$H_N(s)=K\frac{P_N(s)}{Q_N(s)}=K\frac{\left(s^2+\omega_1^2\right)\left(s^2+\omega_2^2\right)\cdots\left(s^2+\omega_m^2\right)}{(s-p_1)(s-p_2)\cdots(s-p_n)} \qquad m=\mathrm{int}\left(\tfrac{1}{2}n\right)$$

$$K=\frac{P_N(0)}{Q_N(0)}$$

Filtros IIR-Método I-Aprox. Chebyshev II

- Ejemplo: El mismo filtro pasabaja realizado anteriormente $\omega_p=4$, $A_p=1dB$, $\omega_s=8$, $A_p=20dB$.
  Normalizamos en $\omega_s=8$ de forma que $\nu_s=1$, $\nu_p=0.5$.

$$\varepsilon^2 = \frac{1}{10^{0.1As}-1} = 0.0101 \qquad n = \frac{\cosh^{-1}\left[\frac{1}{\left(10^{0.1Ap}-1\right)\varepsilon^2}\right]^{\frac{1}{2}}}{\cosh^{-1}\left(\omega_s/\omega_p\right)} = 2.78 \to n = 3$$

$$\alpha = \left(\tfrac{1}{n}\right)\sinh^{-1}\left(\tfrac{1}{\varepsilon}\right) = 0.9977$$

$$1/s_1 = -\sinh(0.9977)\sin\left(\tfrac{\pi}{6}\right) + j\cosh(0.9977)\cos\left(\tfrac{\pi}{6}\right) = -0.5859 + j1.3341$$

$$1/s_2 = -\sinh(0.9977)\sin\left(\tfrac{\pi}{2}\right) + j\cosh(0.9977)\cos\left(\tfrac{\pi}{2}\right) = -1.1717$$

$$1/s_3 = -\sinh(0.9977)\sin\left(\tfrac{5\pi}{6}\right) + j\cosh(0.9977)\cos\left(\tfrac{5\pi}{6}\right) = -0.5859 - j1.3341$$

$$\left.\begin{array}{l} s_1 = -0.276 - j0.6284 \\ s_2 = -0.8535 \\ s_3 = -0.276 + j0.6284 \end{array}\right\} Q_N(s) = (s+0.8535)\left(s^2+0.552s+0.4711\right)$$

Filtros IIR-Método I-Aprox. Chebyshev II

$$z_k = \pm j \sec\theta_k \qquad k = 1$$

$$z_1 = \pm j \sec\left(\tfrac{\pi}{6}\right) = \pm j1.1547$$

$$P_N(s) = s^2 + 1.3333 \qquad\qquad K = \frac{Q_N(0)}{P_N(0)} = \frac{0.8535 \cdot 0.4711}{1.3333} = 0.3016$$

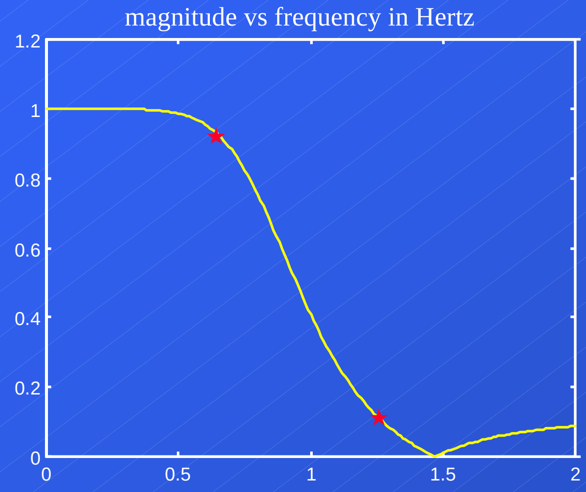
$$H_N(s) = 0.3016 \frac{\left(s^2 + 1.3333\right)}{(s + 0.8535)\left(s^2 + 0.552s + 0.4711\right)}$$

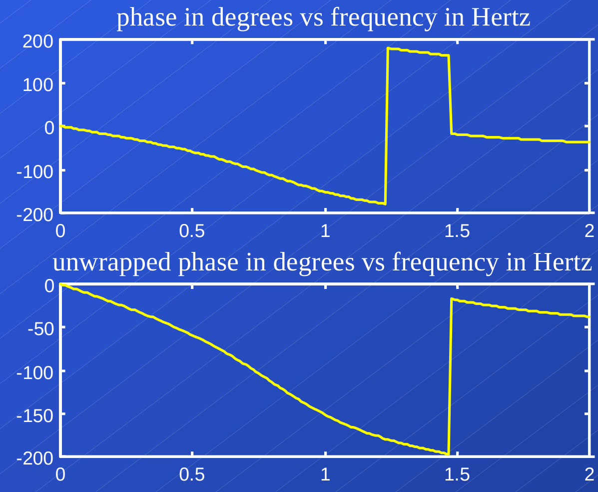
Para desnormalizar esta función y obtener $H_A(s)$ hacemos la transformación $s \rightarrow s/\omega_s = s/8$.

$$H_A(s) = 2.4128 \frac{\left(s^2 + 85.3312\right)}{(s + 6.828)\left(s^2 + 4.416s + 30.1504\right)}$$

$$= \frac{\left(2.412s^2 + 205.867\right)}{\left(s^3 + 11.244s^2 + 60.3s + 205.867\right)}$$

Filtros IIR-Método I-Aprox. Chebyshev II





- Ejemplo: Haremos el ejemplo anterior del pasabanda con un filtro de Chevishev II. $[\omega_1\ \omega_2\ \omega_3\ \omega_4]=[0.89\ 1.019\ 2.221\ 6.155]$, $A_p=2dB$, $A_s=20dB$. Con estos datos y a partir de las tablas de transformación de parabanda a pasobajo :
  - $\omega_x=(\omega_2\omega_3)^{1/2}= (1.019\cdot 2.221)^{1/2}=1.5045$
  - $\omega_4=\omega_x^2/\omega_1=2.54$. $B_w=\omega_3-\omega_2=1.202$
  - $\omega_{Ap}=1$, $\omega_{As}=(\omega_4-\omega_1)/B_w=1.3754$

  Como normalizamos en la frecuencia de parabanda : $\nu_s=1$, $\nu_p=\omega_{Ap}/\omega_{As}=0.727$

Filtros IIR-Método I-Aprox. Chebyshev II

Con estos datos calculamos ε^2 y n :

$$\varepsilon^2 = \frac{1}{10^{0.1As}-1} = 0.0101 \qquad n = \frac{\cosh^{-1}\left[\dfrac{1}{\left(10^{0.1Ap}-1\right)\varepsilon^2}\right]^{\frac{1}{2}}}{\cosh^{-1}\left(\omega_s/\omega_p\right)} = 3.87 \rightarrow n = 4$$

Y a partir de aquí el resto

$$\alpha = \left(\tfrac{1}{n}\right)\sinh^{-1}\left(\tfrac{1}{\varepsilon}\right) = 0.7483$$

$$1/s_1 = -\sinh(0.7483)\sin\left(\tfrac{\pi}{8}\right) + j\cosh(0.7483)\cos\left(\tfrac{\pi}{8}\right) = -0.3138 + j1.1948$$

$$1/s_2 = -\sinh(0.7483)\sin\left(\tfrac{3\pi}{8}\right) + j\cosh(0.7483)\cos\left(\tfrac{3\pi}{8}\right) = -0.7577 + j0.4949$$

$$1/s_3 = -\sinh(0.7483)\sin\left(\tfrac{5\pi}{8}\right) + j\cosh(0.7483)\cos\left(\tfrac{5\pi}{8}\right) = -0.7577 - j0.4949$$

$$1/s_4 = -\sinh(0.7483)\sin\left(\tfrac{7\pi}{8}\right) + j\cosh(0.7483)\cos\left(\tfrac{7\pi}{8}\right) = -0.3138 - j1.1948$$

$$\left.\begin{aligned} s_1 &= -0.2056 - j0.7829 \\ s_2 &= -0.9251 - j0.6042 \\ s_3 &= -0.9251 + j0.6042 \\ s_4 &= -0.2056 + j0.7829 \end{aligned}\right\} Q_N(s) = \left(s^2 + 0.4112s + 0.6552\right)\left(s^2 + 1.8502s + 1.2209\right)$$

Filtros IIR-Método I-Aprox. Chebyshev II

$$z_k = \pm j \sec\theta_k \qquad k = 1,2$$

$$\left.\begin{aligned} z_1 &= \pm j\sec\left(\tfrac{\pi}{8}\right) = \pm j1.0824 \\ z_2 &= \pm j\sec\left(\tfrac{3\pi}{8}\right) = \pm j2.6131 \end{aligned}\right\} P_N(s) = (s^2 + 1.1716)(s^2 + 6.8284)$$

$$K = \frac{Q_N(0)}{P_N(0)} = 0.1 \Rightarrow$$

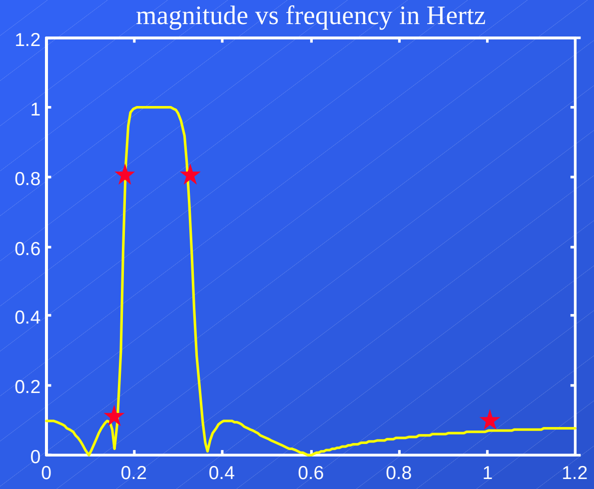
$$H_N(s) = 0.1\frac{(s^2 + 1.1716)(s^2 + 6.8284)}{\left(s^2 + 0.4112s + 0.6552\right)\left(s^2 + 1.8502s + 1.2209\right)}$$

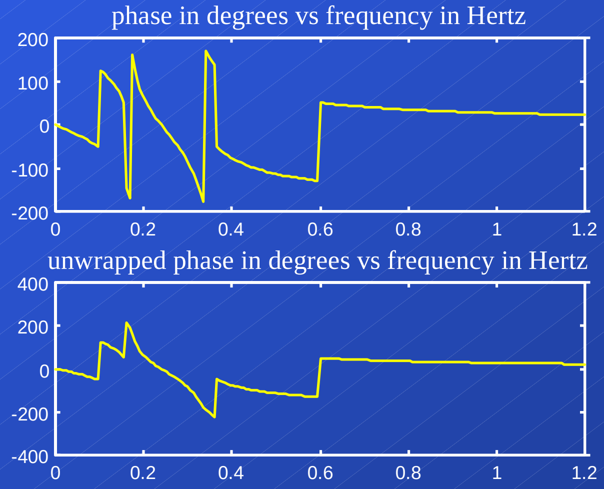
$$= 0.1\frac{s^4 + 8s^2 + 8}{s^4 + 2.2614s^3 + 2.6369s^2 + 1.7143s + 0.8}$$

Haciendo la transformación a pasabanda $\quad s \rightarrow \dfrac{s^2 - \omega_x^2}{sB_w} = \dfrac{s^2 - 2.1635}{1.202s}$

$$H_A(s) = \frac{0.1s^8 + 2.0613s^6 + 9.9766s^4 + 10.5609s^2 + 2.625}{s^8 + 2.718s^7 + 12.864s^6 + 21.435s^5 + 49.658s^4 + 48.519s^3 + 65.908s^2 + 31.524s + 26.250}$$

Filtros IIR-Método I-Aprox. Chebyshev II





Filtros IIR-Método I

- Una vez estudiadas las técnicas de diseño de filtros analógicos retomamos el objetivo inicial de este tema, que era transformar las funciones de Transferencia de los filtros del plano *s* al plano *z*. En esto consistía el Método I de diseño de filtros digitales.
- Hay varios métodos para transformar una función de *s* en otra función de *z*. Estamos interesados en transformaciones que hagan que la función en *z* sea también racional. Esto hace que las transformaciones que vamos a ver son sólo aproximaciones.
- Una transformación $s \rightarrow z$ debe cumplir dos condiciones fundamentales,
  - Estabilidad: la mitad izquierda del plano *s* debe transformarse dentro del círculo unidad en el plano *z*.
  - A cada frecuencia analógica dentro del intervalo $(-\infty, \infty)$ le debe corresponder una única frecuencia digital en el intervalo $(-f_s/2, f_s/2)$. Esto evita el problema del “aliasing”.

Filtros IIR-Método I

- Los métodos que vamos a ver son :
  - Igualar las respuestas temporales como un impuso, un escalón, una rampa, etc. (*Transformación Invariante a la Respuesta*).
  - Igualar términos en una *H(s)* factorizada (*Transformada Z Pareada*).
  - Conversión de ecuaciones diferenciales a ecuaciones diferencia utilizando operadores diferencia.
  - Integración numérica de ecuaciones diferenciales usando algoritmos de integración.
  - Aproximaciones racionales a $z \rightarrow exp(st_s)$ o $s \rightarrow (1/t_s)ln(z)$.

Filtros IIR-Método I-TIR

- Transformación Invariante a la Respuesta (TIR)
  - Se elige una entrada $x(t)$ (impulso, escalón o rampa).
  - Determinar la respuesta $y(t)$ como la $L^{-1}\{H(s)X(s)\}$.
  - Muestrear $y(t)$ a intervalos t_s y obtener $y[n]$ y su Transformada Z $Y(z)$.
  - Muestrear $x(t)$ para obtener $x[n]$ y $X(z)$.
  - Evaluar $H(z)$ como $Y(z)/X(z)$.

  Este tipo de transformación está limitada por la frecuencia de muestreo f_s que restringe su aplicación a aquellos sistemas cuya respuesta está limitada por $\pm\frac{1}{2}f_s$ por problemas de aliasing ya que la función de transferencia transformada es una función periódica de periodo $1/t_s$. Esto hace que este tipo de transformación es más apropiada para filtros pasobajos de Butterworth y Chebyshev I que para filtros de Chebyshev II.

  Una caso particular de este tipo es la *Transformación Invariante a Impulso*. Supongamos que tenemos $H(s)$ en forma de fracciones parciales. Para cada término podemos determinar su equivalente en z a través de su respuesta a impulso. Los términos y sus equivalencias en z están tabulados en la tabla adjunta.

Filtros IIR-Método I-TIR

Transformaciones Invariantes al Impulso		
Término	***H(s)***	***H(z)*** *a=exp(-pt$_s$)*
Unico	$\frac{A}{(s+p)}$	$\frac{Az}{(z-a)}$
Complejo Conjugado	$\frac{A\exp(j\Omega)}{(s+p+jq)}+\frac{A\exp(-j\Omega)}{(s+p-jq)}$	$\frac{2z^2A\cos(\Omega)-2Aaz\cos(\Omega+qt_s)}{z^2-2az\cos(qt_s)+a^2}$
Repetido	$\frac{A}{(s+p)^M}$	$\frac{A}{(M-1)!}t_s^{(M-1)}\left(-z\frac{d}{dz}\left(-z\frac{d}{dz}\cdots-z\frac{d}{dz}\left[\frac{z}{z-a}\right]\right)\right)$
Repetido	$\frac{A}{(s+p)^2}$	$At_s^2\frac{az}{(z-a)^2}$
Repetido	$\frac{A}{(s+p)^3}$	$\frac{1}{2}At_s^3\frac{za(z+a)}{(z-a)^3}$
Modificado	$\frac{A}{(s+p)}$	$\frac{Az}{(z-a)}-\frac{1}{2}A=\frac{1}{2}\frac{z+a}{z-a}$
Modificado	$\frac{A\exp(j\Omega)}{(s+p+jq)}+\frac{A\exp(-j\Omega)}{(s+p-jq)}$	$\frac{2z^2A\cos(\Omega)-2Aaz\cos(\Omega+qt_s)}{z^2-2az\cos(qt_s)+a^2}-A\cos(\Omega)$

Filtros IIR-Método I-TIR

Ejemplo: Determinar *H(z)* a partir de *H(s)* utilizando la Transformación Invariante al Impulso.

$$H(s)=\frac{4}{(s+1)\left(s^2+4s+5\right)}=\frac{2}{(s+1)}+\frac{-1-j}{(s+2+j)}+\frac{1+j}{(s+2-j)}$$

$$=\frac{2}{(s+1)}+\frac{\sqrt{2}\exp\left(-j\frac{3\pi}{4}\right)}{(s+2+j)}+\frac{\sqrt{2}\exp\left(j\frac{3\pi}{4}\right)}{(s+2-j)}$$

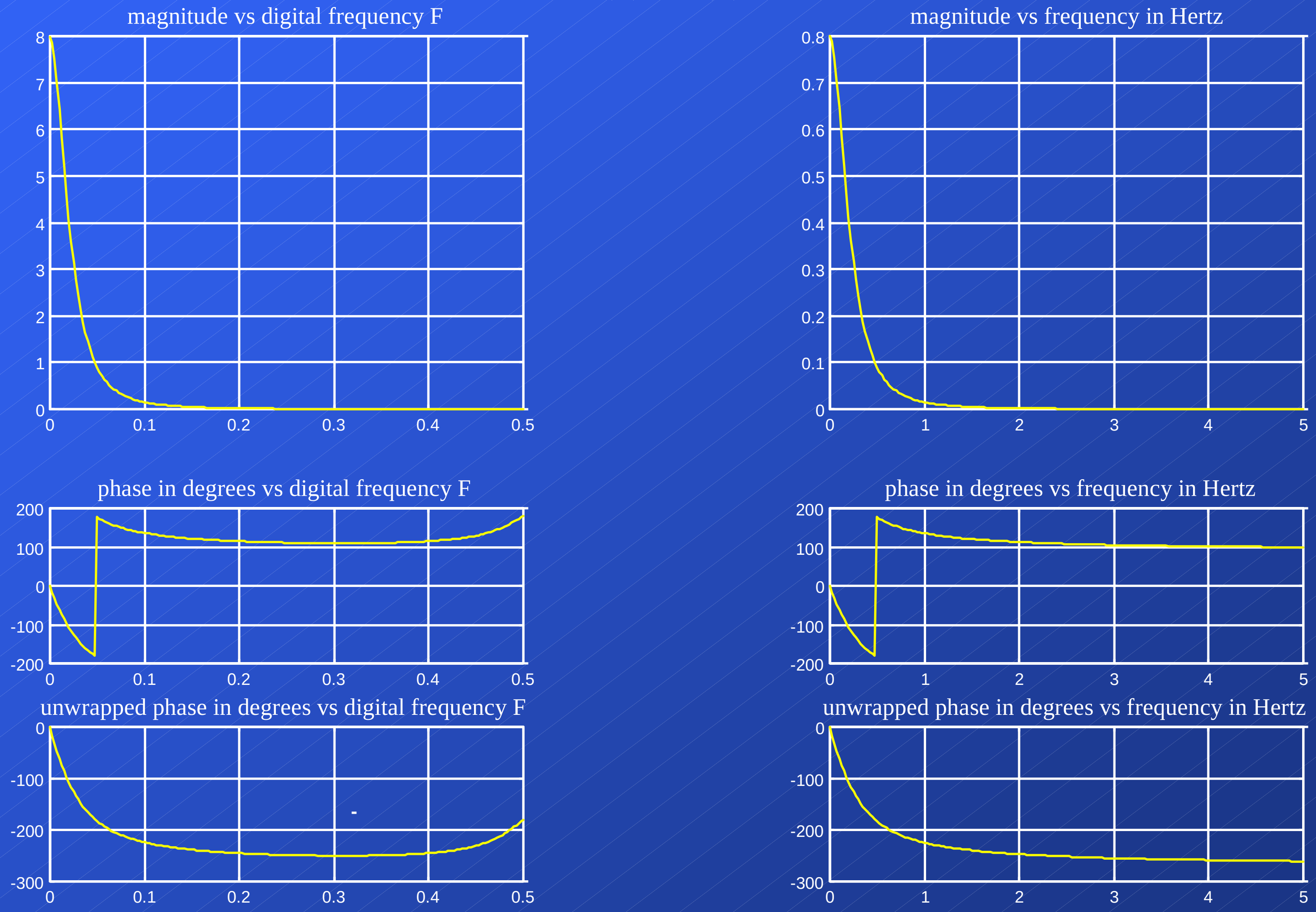
$$H(z)=\frac{2z}{\left(z-e^{-t_s}\right)}+\frac{2z^2\sqrt{2}\cos\left(\frac{3\pi}{4}\right)-2ze^{-2t_s}\sqrt{2}\cos\left(t_s-\frac{3\pi}{4}\right)}{z^2-2e^{-2t_s}z\cos\left(t_s\right)+e^{-4t_s}}$$

$$Si\ t_s=0.1\Rightarrow H(z)=\frac{2z}{z-0.9048}+\frac{-2z^2+1.4658z}{z^2-1.629z+0.67}=\frac{0.0174z^2+0.0137z}{(z-0.9048)\left(z^2-1.629z+0.67\right)}$$

$$=\frac{0.0174z^2+0.0137z}{z^3-2.5338z^2+2.1439z-0.6062}$$

Filtros IIR-Método I-TIR

H(z)-H(s)



Filtros IIR-Método I-TZP

- Transformada Z Pareada (TZR)
  - Surge a partir de la Transformación Invariante al Impulso de *1/(s+a)*.

$$\frac{1}{s+a} \to \frac{z}{z-\exp(-at_s)} \Rightarrow (s+a) \to \frac{z-\exp(-at_s)}{z}$$

  - Esta transformación usa esta forma para sustituir cada término del numerador y del denominador en una *H(s)* factorizada y generar el consiguiente *H(z)* :

$$H(s) = K_0 \frac{\prod_{i=1}^{M}(s-z_i)}{\prod_{k=1}^{N}(s-p_k)} \to H(z) = K \cdot z^{(N-M)} \frac{\prod_{i=1}^{M}\left[z-\exp(z_i t_s)\right]}{\prod_{k=1}^{N}\left[z-\exp(p_k t_s)\right]}$$

  - La constante *K* se escoge para que las ganancias sean iguales a una frecuencia determinada. En el caso de haber raíces complejas, podemos hacer la siguiente transformación :

$$(s+p-jq)(s+p+jq) \to \frac{\left[z^2 - 2ze^{-pt_s}\cos(qt_s) + e^{-2pt_s}\right]}{z^2}$$

Filtros IIR-Método I-TZP

- Modificaciones al TZR: Hay que transformar también los ceros de *H(s)* en $s=\infty$, que equivale a $z=-1$. La razón de esta elección es que $f=\infty$ en el dominio analógico equivale a $f=½fs$ en el dominio discreto o $z=-1$. Haremos dos tipos de modificaciones:
  1. Reemplazar todos los ceros en $z=0$ por $z=-1$ (o $z^{(N-M)}$ por $(z+1)^{(N-M)}$).
  2. Reemplazar todos los ceros excepto uno en $z=0$ por $z=-1$.

Ejemplo: Utilizar el método TZR para transformar *H(s)* en *H(z)*.

$$H(s)=\frac{4}{(s+1)\left(s^2+4s+5\right)}\to H(z)=Kz^3\frac{1}{\left(z-e^{-t_s}\right)\left(z^2-2ze^{-2t_s}\cos\left(t_s\right)+e^{-4t_s}\right)}$$

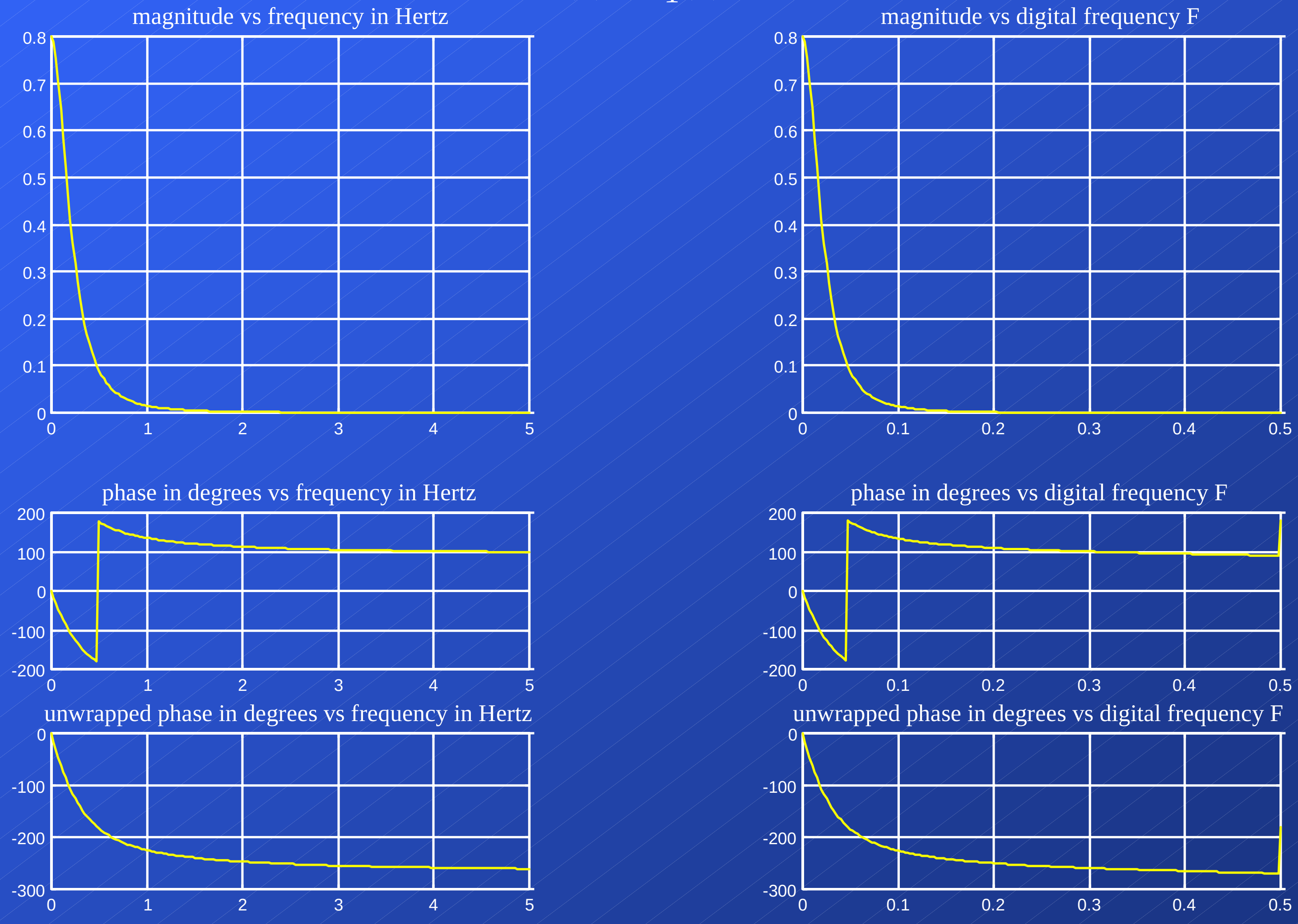
$$Modificaciones \quad ts=0.1s$$

$$H_1(z)=\frac{K(z+1)^3}{(z-0.9048)\left(z^2-1.6293z+0.67\right)}=\frac{390\cdot10^{-6}(z+1)^3}{z^3-2.5341z^2+2.1442z-0.6062}$$

$$H_2(z)=\frac{Kz(z+1)^2}{(z-0.9048)\left(z^2-1.6293z+0.67\right)}=\frac{780\cdot10^{-6}z(z+1)^2}{z^3-2.5341z^2+2.1442z-0.6062}$$

Filtros IIR-Método I-TZP

$H(s)$-$H_1(z)$



Filtros IIR-Método I-TAD

- Transformación a través de algoritmos diferencia (TAD)
  - Se trata de convertir una operación de derivada en su correspondiente ecuación diferencia.
  - Existen varias formas de calcular la derivada a partir de datos discretos (Tabla).

Algoritmos para la Diferenciación Discreta		
Diferencia	***Algoritmo***	***Transformación***
Hacia Atrás	$y[n] = \dfrac{x[n] - x[n-1]}{t_s}$	$s \to \dfrac{z-1}{zt_s}$
Central	$y[n] = \dfrac{x[n+1] - x[n-1]}{2t_s}$	$s \to \dfrac{z^2-1}{2zt_s}$
Hacia Adelante	$y[n] = \dfrac{x[n+1] - x[n]}{t_s}$	$s \to \dfrac{z-1}{t_s}$

  - El algoritmo que mejor se comporta en cuanto a la estabilidad es el algoritmo hacia atrás. Por tanto, esta transformación consistirá en sustituir el valor de *s* en *H(s)* por $(z-1)/zt_s$ siendo el resultado *H(z)*.

Filtros IIR-Método I-TAD

- Veamos lo que ocurre en esta transformación en cuanto a la estabilidad.

$$s \rightarrow \frac{1}{t_s}\frac{z-1}{z}$$

$$z = \frac{1}{1-s\cdot t_s} = \frac{1}{2} + \frac{1}{2}\frac{(1+s\cdot t_s)}{(1-s\cdot t_s)}$$

Sólo estamos interesados en lo que ocurre en el régimen estacionario por lo que sustituiremos *s* por *jω.*

$$z = \frac{1}{2} - \frac{1}{2}\frac{(1+j\omega\cdot t_s)}{(1-j\omega\cdot t_s)} \Rightarrow \left|z-\frac{1}{2}\right| = \frac{1}{2}$$

Si $\quad z = u + jv$

$$\left(u-\frac{1}{2}\right)^2 + v^2 = \frac{1}{4}$$

Esta ecuación representa una circunferencia de radio *1/2* centrada en *(1/2,0)*

Esto significa que el plano izquierdo del plano s se transforma en un círculo de radio 1/2 que está dentro del circulo unidad. Ya que los polos de un filtro pasobajo se encuentran aprox. en esa zona, esta transformación es más adecuada para esos filtros.

Filtros IIR-Método I-TAI

- <u>Transformación a través de algoritmos de integración numérica (TAI)</u>
  - Aquí tenemos bastantes más algoritmos para hacer una integración numérica. La Tabla muestra algunos algoritmos típicos.

Algortimos de Integración numérica *Entre paréntesis se indica el orden del algoritmo*	
Algoritmo	***Fórmula para y[n]***
Rectangular(1)	$y[n] = y[n-1] + x[n]t_s$
Trapezoidal(1)	$y[n] = y[n-1] + \frac{\{x[n] - x[n-1]\}t_s}{2}$
Adams(2)	$y[n] = y[n-1] + \frac{\{5x[n] + 8x[n-1] - x[n-2]\}t_s}{12}$
Adams(3)	$y[n] = y[n-1] + \frac{\{9x[n] + 19x[n-1] - 5x[n-2] + x[n-3]\}t_s}{24}$
Simpsom(2)	$y[n] = y[n-2] + \frac{\{x[n] + 4x[n-1] + x[n-2]\}t_s}{3}$
Tick(2)	$y[n] = y[n-2] + \{0.3584x[n] + 1.2832x[n-1] + 0.3584x[n-2]\}t_s$

Filtros IIR-Método I-TAI

- Las transformaciones para cada algoritmo son

Transformación $s \to z$	
Algoritmo	***Transformación***
Rectangular (1)	$s \to \frac{1}{t_s}\frac{z-1}{z}$
Trapezoidal (1)	$s \to \frac{2}{t_s}\frac{z-1}{z+1}$
Adams (2)	$s \to \frac{12}{t_s}\frac{z^2-z}{5z^2+8z-1}$
Adams (3)	$s \to \frac{24}{t_s}\frac{z^3-z^2}{9z^3+19z^2-5z+1}$
Simpson (2)	$s \to \frac{3}{t_s}\frac{z^2-1}{z^2+4z+1}$
Tick (2)	$s \to \frac{1}{t_s}\frac{z^2-1}{0.3584z^2+1.2832z+0.3584}$

Filtros IIR-Método I-TBL

- Excepto los algoritmos de Simpson y Tick, el resto producen *H(z)* estables a partir de un *H(s)* estable para cualquier valor de t_s.

Transformación Bilineal

- Es la Transformación dada por el algoritmo trapezoidal

$$z \to \frac{2+st_s}{2-st_s} \Rightarrow s \to \frac{2}{t_s}\frac{z-1}{z+1} \qquad (1)$$

- Tomamos $t_s=2$, para simplificar las expresiones. Sustituimos *s* por $j\omega$ en la expresión anterior : $z=\frac{1+j\omega}{1-j\omega}$

  Esta expresión nos dice que un punto del eje imaginario de *s*, se transforma en un punto en el círculo unidad en el plano *z*.

  Generalizando, para $s=\sigma+j\omega$, $z=\frac{1+(\sigma+j\omega)}{1-(\sigma+j\omega)}=\frac{(1+\sigma)+j\omega}{(1-\sigma)-j\omega} \Rightarrow |z|^2=\frac{(1+\sigma)^2+\omega^2}{(1-\sigma)^2+\omega^2}$

  lo que implica que un punto de la mitad del plano *s* donde $\sigma < 0$, se transforma en un punto dentro del círculo unidad en el plano *z*. Por lo tanto, un diseño estable en *s*, será también estable en *z* bajo la transformación bilineal. Además, no hay aliasing, ya que a cada frecuencia analógica le corresponde una única frecuencia digital.

Filtros IIR-Método I-TBL

- Nos interesa la relación entre los puntos del eje imaginario de *s* con los punto del círculo unidad en el plano *z*, de la ecuación (1) :

$$j\omega_A = \frac{2}{t_s}\frac{e^{j\omega_D t_s}-1}{e^{j\omega_D t_s}+1} = \frac{2}{t_s}\frac{e^{j\omega_D t_s}-1}{e^{j\omega_D t_s}+1} = j\frac{2}{t_s}\tan\left(\frac{\omega_D t_s}{2}\right)$$

$$\omega_A = \frac{2}{t_s}\tan\left(\frac{\omega_D t_s}{2}\right) \Rightarrow \omega_D = \frac{2}{t_s}\tan^{-1}\left(\frac{\omega_A t_s}{2}\right)$$

- Estas expresiones nos indican que existe una distorsión en las frecuencias cuando se hace la Transformación Bilineal. El rango entero de frecuencias analógicas se mapea con frecuencias digitales entre $-f_s/2$ y $f_s/2$. Hay una compresión de frecuencias o una distorsión frecuencial (la relación es no-lineal). Para compensarlo, lo que se hace es predistorsionar (prewarping) las especificaciones del filtro analógico original mediante la ecuación (Ω es la frecuencia digital, $2\pi f/f_s$):

$$\omega_A = \frac{2}{t_s}\tan\left(\frac{\omega_D t_s}{2}\right) = \frac{2}{t_s}\tan\left(\frac{\omega_D}{2 f_s}\right) = \frac{2}{t_s}\tan\left(\frac{\Omega}{2}\right)$$

Filtros IIR-Método I-TBL

- Ejemplo 1: Dado el filtro $H(s)=3/(s^2+3s+3)$, diseñar un filtro digital cuya magnitud a $f_0=3KHz$ sea igual a la magnitud de $H(s)$ en $\omega_A=4rad/s$, siendo la frecuencia de muestreo $f_s=12\ KHz$.
  - $\omega_X = 2\ f_s \tan(\omega_D / 2f_s) = 24000 \tan(2\pi 3/24) = 24000\ rad/s$
  - Escalamos $H(s)$ con $\omega_X \rightarrow H_X(s)=H(s\omega_A/\omega_X)=H(s/6000)$
  - $H(z)=H_X(s)|_{s=24000\cdot(z-1)/(z+1)}= H(s/6000)|_{s=24000\cdot(z-1)/(z+1)} = 3\cdot(z+1)^2/(31z^2-26z+7)$
- Ejemplo 2 : El filtro $H(s)=(s^2+1)/(s^2+4s+1)$ tiene una frecuencia de corte de $1rad/s$. Diseñar un filtro digital cuya respuesta a $f_D=60Hz$ sea igual a la de $H(s)$ en $\omega_A=1\ rad/s$, siendo $f_s=240Hz$.
  - $\omega_X = 2\ f_s \tan(\omega_D / 2f_s) = 480 \tan(2\pi 60/480) = 480\ rad/s$
  - $H_X(s)=H(s\omega_A/\omega_X)=H(s/480)$
  - $H(z)=H_X(s)|_{s=480\cdot(z-1)/(z+1)}= (z^2+1)/(3z^2-1)$

Filtros IIR-Método I-TBL

- En los ejemplos anteriores se observa que las operaciones a realizar son transparentes al valor tomado para f_s, de forma que podríamos haber utilizado la siguiente ecuación para predistorsionar :

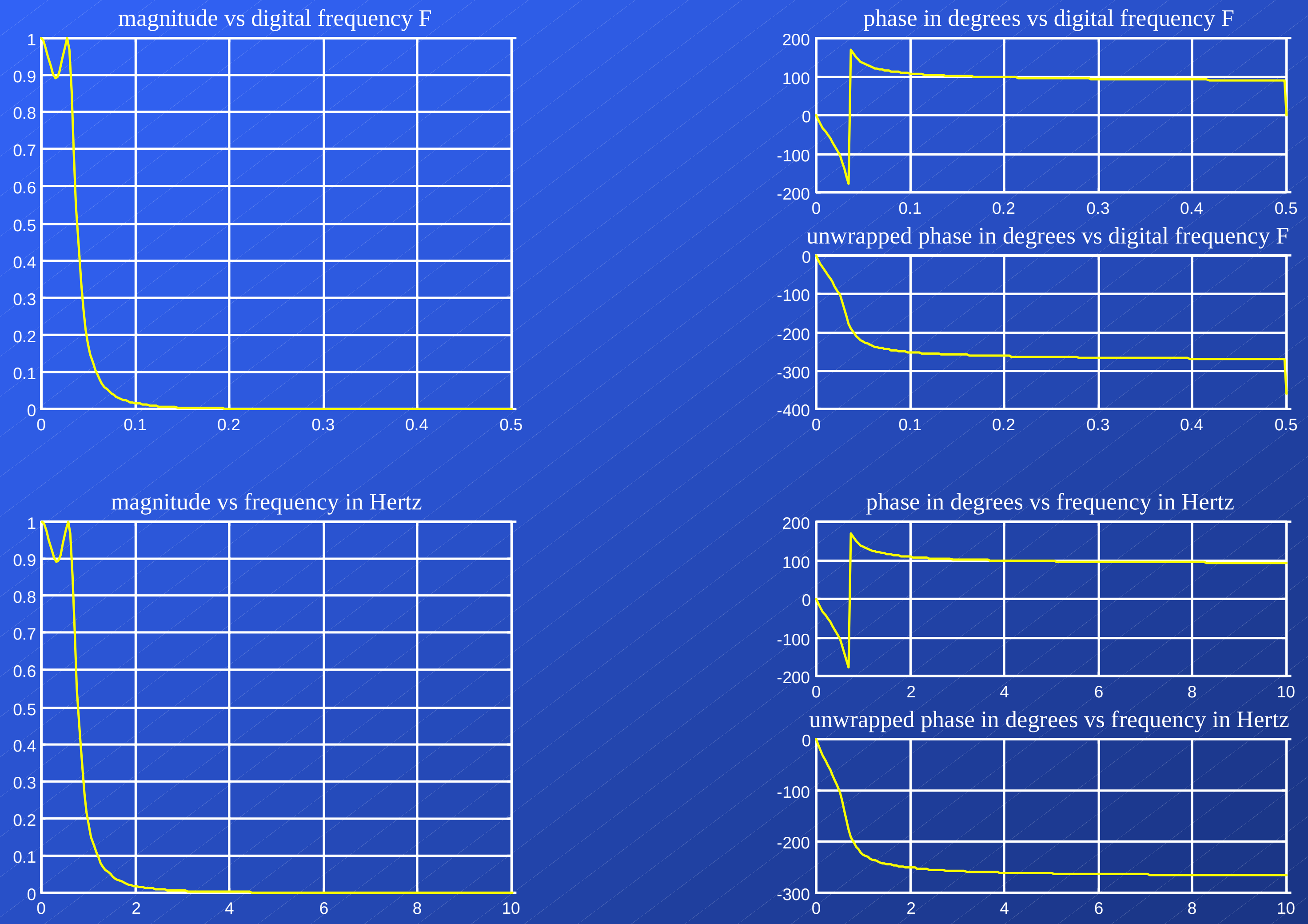
$$\omega_X = 2\tan(\Omega/2)$$

Filtros IIR-Método I-TBL

- Ejemplo 3 : El filtro de pasobajo de Chebyshev I obtenido en anteriores ejemplos era del tipo $H(s)=31.4436/(s^3+3.9534s^2+19.8145s+31.4436)$. Utilizando la transformación bilineal diseñar un filtro digital cuya respuesta a $f_D=0.7Hz$ sea igual a la respuesta de $H(s)$ en $f=0.7Hz$. La frecuencia de muestreo es $f_s=20Hz$.
  - $\omega_X = 2\tan(\Omega/2) = 2\tan(2\pi\, 0.7/(2\, 20)) = 0.2208$
  - $H_X(s)=H(s\omega_A/\omega_X)=H(19.919s)$
  - $H(z)=H_X(s)|_{s=2\cdot(z-1)/(z+1)} = H(19.919s)|_{s=2\cdot(z-1)/(z+1)}$

$$H(z)=\frac{z^3+3z^2+3z+1}{2236.73z^3-6204.64z^2+5811.52z-1835.6}$$

Filtros IIR-Método I-TBL



Filtros IIR-Método I-TBL

- Ejemplo : Diseñar un filtro digital de Chebyshev de Pasabanda con las siguientes especificaciones : Frecuencias de pasabanda *1.8KHz* y *3.2 KHz*. Frecuencias de Parabanda *1.6KHz* y *4.8KHz*. Atenuaciones $A_p=2dB$ y $A_s=20dB$. La frecuencia de muestreo es $f_s=12KHz$.
  - Predistorsionamos las especificaciones:

$$\omega_1 = 2\tan\left(2\pi\, 1.6/24\right) = 0.89$$

$$\omega_2 = 2\tan\left(2\pi\, 1.8/24\right) = 1.019$$

$$\omega_3 = 2\tan\left(2\pi\, 3.2/24\right) = 2.221$$

$$\omega_4 = 2\tan\left(2\pi\, 4.8/24\right) = 6.155$$

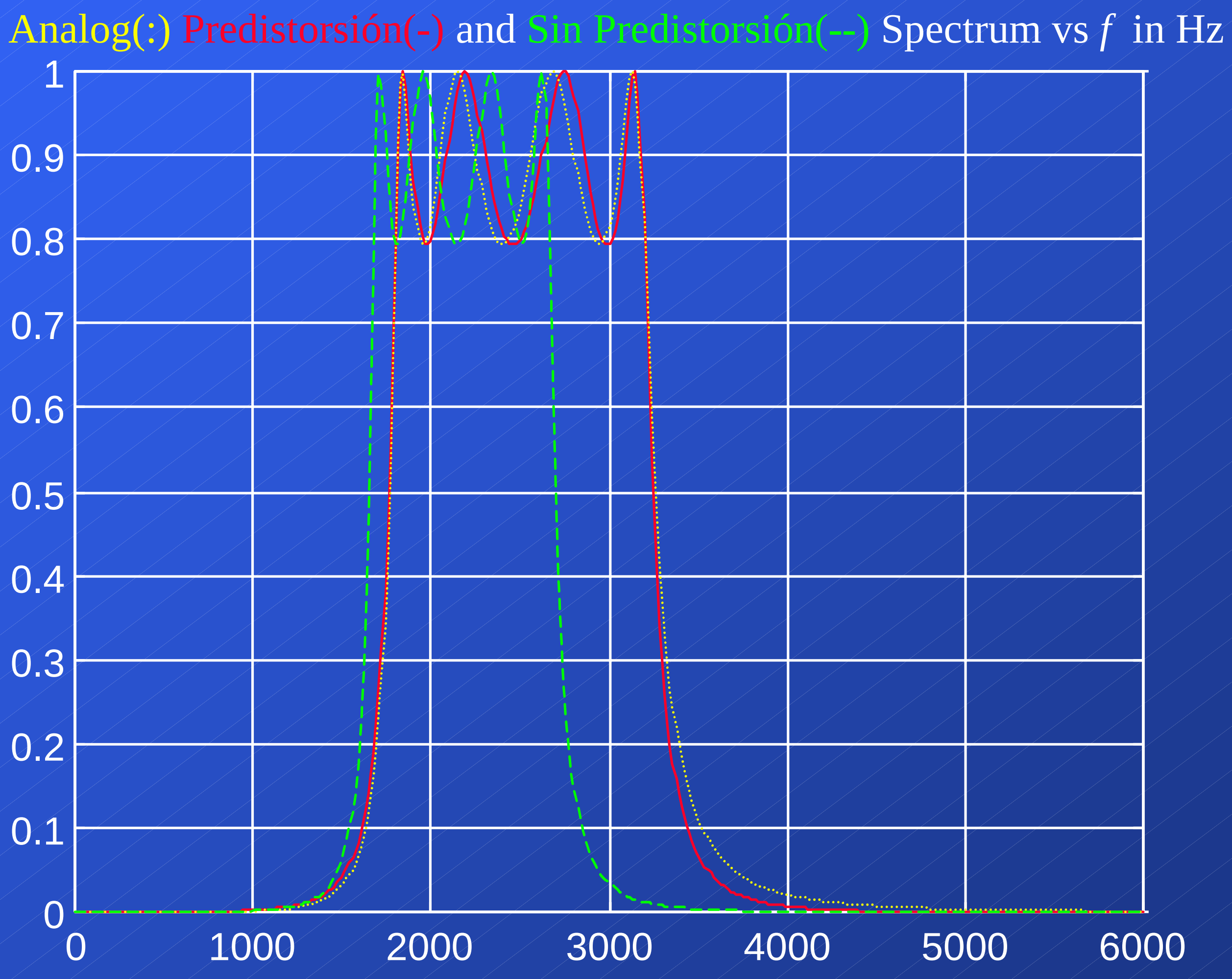
  - Diseñamos el filtro,

$$H(s) = \frac{0.3411 s^4}{s^8 + 0.8608 s^7 + 10.89 s^6 + 6.74 s^5 + 39.38 s^4 + 15.26 s^3 + 55.67 s^2 + 9.98 s + 26.24}$$

  - Se hace la transformación $s \rightarrow 2(z-1)/(z+1)$

$$H(z) = \frac{0.0024z^8 - 0.0096z^6 + 0.0144z^4 - 0.0096z^2 + 0.0024}{z^8 - 1.938z^7 + 4.435z^6 - 5.073z^5 + 6.239z^4 - 4.464z^3 + 3.438z^2 - 1.304z + 0.5928}$$

Filtros IIR-Método I



Diseño de Filtros IIR con MATLAB

- MATLAB dispone de funciones que facilitan el diseño de filtros, tanto analógicos como digitales.
- Funciones para determinar el orden necesario para implementar un determinado filtro :

```
>> [N, Wn] = buttord(Wp, Ws, Rp, Rs)
```

Calcula el orden de un filtro pasobajo digital de Butterworth, con *Ws* la frecuencia de pasabanda, *Wp* la parabanda, y *Rp* y *Rs* las atenuaciones respectivas de pasabanda y parabanda en decibelios. *Wp* y *Ws* deben ser entre (0,1), siendo 1 la frecuencia de Nyquist ($f_s/2$).

N es el orden del filtro y *Wn* la frecuencia de 3db.

```
>> [N, Wn] = buttord(Wp, Ws, Rp, Rs,'s')
```

Lo mismo que antes, pero para un filtro pasobajo analógico. Aquí los valores de *Wp* y *Ws* pueden tomar cualquier valor en radianes.

Para calcular el orden de otros tipos de filtro (pasoalto, parabanda o pasabanda) deberemos aplicar primero las transformaciones al prototipo de filtro pasobajo (Tabla 2).

Diseño de Filtros IIR con MATLAB

```
>> [N, Wn] = cheb1ord(Wp, Ws, Rp, Rs)
```

Cálculo del orden necesario para un filtro digital pasobajo de Chebyshev I, con las especificaciones dadas. Las mismas consideraciones que en el caso del filtro de Butterworth.

```
>> [N, Wn] = cheb1ord(Wp, Ws, Rp, Rs, 's')
```

Lo mismo pero para el filtro analógico

```
>> [N, Wn] = cheb2ord(Wp, Ws, Rp, Rs)
```

Filtro digital de Chebyshev II

```
>> [N, Wn] = cheb2ord(Wp, Ws, Rp, Rs, 's')
```

Filtro analógico de Chebyshev II

```
>> [N, Wn] = ellipord(Wp, Ws, Rp, Rs)
```

Filtro digital elíptico

```
>> [N, Wn] = ellipord(Wp, Ws, Rp, Rs, 's')
```

Filtro analógico elíptico

Diseño de Filtros IIR con MATLAB

- Ejemplo 1: Determinar el orden necesario para un filtro analógico pasabanda con las siguientes especificaciones y para todos los tipos de filtros estudiados :

  Pasabanda 30-50 Hz , Parabanda < 5Hz y >200 Hz, A_p<2dB, A_s>40dB

  - Hacemos primero la transformación de Pasabanda a Pasobajo, de acuerdo con la Tabla 2: Recalculo f_1, $f_1 = f_2 f_3 / f_4 = 7.5$ Hz. La frecuencia de pasobajo es $f_3 - f_2 = 20$ Hz, y la frecuencia de pasoalto es $f_4 - f_1 = 192.5$ Hz.

```
>> [N,Wn]=buttord(2*pi*20,2*pi*192.5,2,40,'s')
   N = 3      Wn = 2*pi*41.4736
>> [N,Wn]=cheb1ord(2*pi*20, 192.5,2,40,'s')
   N = 2      Wn = 2*pi*20
>> [N,Wn]=cheb2ord(2*pi*20, 2*pi*192.5,2,40,'s')
   N = 2      Wn = 2*pi*162.3265
>> [N,Wn]=elliord(2*pi*20, 2*pi*192.5,2,40,'s')
   N = 2      Wn = 2*pi*20
```

Diseño de Filtros IIR con MATLAB

- Ejemplo 2: Lo mismo de antes, pero con filtros digitales. Elegir la frecuencia de muestreo.
  - La frecuencia de muestreo deberá ser > *2 $f_{máx}$ = 400 Hz.* Tomo *f_s = 1000 Hz.* Normalizando las frecuencias con respecto a la frecuencia de Nyquist ($f_s/2$) :

    f_1 = 0.01, f_2 = 0.06, f_3 = 0.1, f_4 = 0.4.

    Recalculamos f_1 de acuerdo con Tabla 2, f_1 = 0.015 y hacemos la Transformación a pasobajo,

```
>> [N,Wn]=buttord(0.04,0.385,2,40)
    N = 3          Wn = 0.0941
>> [N,Wn]=cheb1ord(0.04,0.385,2,40)
    N = 2          Wn = 0.04
>> [N,Wn]=cheb2ord(0.04,0.385,2,40)
    N = 2          Wn = 0.3006
>> [N,Wn]=elliord(0.04,0.385,2,40)
    N = 2          Wn = 0.04
```

  - Se puede obtener un filtro de menor orden si utilizamos una frecuencia de muestreo menor (por ejemplo, 500). El aliasing no será tan significativo ya que el la respuesta de filtro a esas frecuancias es muy pequeña (> 40 dB).

Diseño de Filtros IIR con MATLAB

- Funciones para determinar los coeficientes del filtro:

```
>> [B,A] = butter(N,Wn)
```

B y *A* son los coeficientes del numerador y del denominador respectivamente, en orden decreciente de un filtro de Butterworth digital. *N* es el orden del filtro (calculado previamente) y *Wn* es la frecuencia de corte. El valor de *Wn* debe estar normalizado con la frecuencia de Nyquist. Para diseñar un filtro pasobajo *Wn* es un escalar entre (0,1). La pasabanda es (0,*Wn*) y la parabanda es (*Wn*,1). Para diseñar un filtro de pasoalto, el comando a escribir es:

```
>> [B,A] = butter(N,Wn,'high')
```

donde *Wn* es un escalar.

Un filtro Parabanda se determina de la siguiente forma:

```
>> [B,A] = butter(N,[W1 W2])
```

Es decir, *Wn* es en este caso un vector que especifica las frecuencias de pasabanda.

Finalmente, para un filtro Parabanda:

```
>> [B,A] = butter(N,[W1 W2],'stop')
```

[W1 W2] son las frecuencias de Parabanda.

Diseño de Filtros IIR con MATLAB

```
>> [B,A] = cheby1(N,R,Wn)
```

Diseño de filtros digitales de Chebyshev I. Se deben especificar el orden del filtro *N*, el rizado de pasabanda permitido *R* y la frecuencia de corte normalizada con respecto a la frecuencia de Nyquist. Para diseñar filtros de pasoalto, pasabanda y parabanda se siguen las mismas reglas que en el diseño de filtros de Butterworth.

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

Lo mismo que antes, pero *R* es el rizado de parabanda.

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

Rp y *Rs* son los rizados de pasabanda y parabanda.

Añadiendo a los comandos anterior la optión ‘s’, los vectores *B* y *A* son los coeficientes del filtro analógico correspondiente. Sigue siendo válido lo que se mencionó anteriormente acerca del diseño de filtros pasoalto, pasabanda y parabanda, pero Wn puede tomar cualquier valor en radianes (no está limitado entre (0,1)):

```
>> [B,A] = butter(N,Wn,'s')
>> [B,A] = cheby1(N,R,Wn,'s')
>> [B,A] = cheby2(N,R,Wn,'s')
>> [B,A] = ellip(N,Rp,Rs,Wn,'s')
```

Diseño de Filtros IIR con MATLAB

- Ejemplo 1: Diseñar el filtro analógico con las especificaciones anteriores para todos los tipos de filtros estudiados (Butterworth, Chebyshev I, Chevyshev II y Elíptico).
  - Utilizaremos los resultados del anterior ejercicio.

```
>> [B,A]=butter(3,2*pi*[30 50],'s')
   B = 1.0e+006 * [0 0.0000 0.0000 1.9844 0.0000 0.0000 0.0000]
   A = 1.0e+014 * [0.0000 0.0000 0.0000 0.0000 0.0001 0.0088 2.0766]
>> [B,A]=cheby1(2,2,2*pi*[30 50],'s')
   B = 1.0e+004 * [0 0.0000 1.0324 0.0000 0.0000]
   A = 1.0e+009 * [0.0000 0.0000 0.0001 0.0060 3.5067]
>> [B,A]=cheby2(2,40,2*pi*[30 50],'s')
   B = 1.0e+007 * [0.0000 0.0000 0.0002 0.0000 3.5067]
   A = 1.0e+009 * [0.0000 0.0000 0.0001 0.0015 3.5067]
>> [B,A]=ellip(2,2,40,2*pi*[30 50],'s')
   B = 1.0e+007 * [0.0000 0.0000 0.0012 0.0000 3.5120]
   A = 1.0e+009 * [0.0000 0.0000 0.0001 0.0059 3.5067]
```

Diseño de Filtros IIR con MATLAB

- Ejemplo 2: Diseñar el filtro digital con las especificaciones anteriores para todos los tipos de filtros estudiados (Butterworth, Chebyshev I, Chevyshev II y Elíptico).
  - La frecuencia de muestreo elegida es f_s=*1000 Hz*.

```
>> [B,A]=butter(3,[0.06 0.1])
   B = 1.0e-003 * [0.2196 0.0000 -0.6588 0.0000 0.6588 0.0000 -0.2196]
   A = [1.0000 -5.5792 13.1338 -16.6913 12.0771 -4.7177 0.7776]
>> [B,A]=cheby1(2,2,[0.06 0.1])
   B = [0.0025 0 -0.0049 0.0000 0.0025]
   A = [1.0000 -3.7768 5.4667 -3.5906 0.9040]
>> [B,A]=cheby2(2,40,[0.06 0.1])
   B = [0.0100 -0.0383 0.0568 -0.0383 0.0100]
   A = [1.0000 -3.8577 5.6956 -3.8097 0.9753]
>> [B,A]=ellip(2,2,40,[0.06 0.1])
   B = [0.0120 -0.0369 0.0499 -0.0369 0.0120]
   A = [1.0000 -3.7773 5.4682 -3.5922 0.9046]
```

Diseño de Filtros IIR con MATLAB

- Funciones para hacer transformaciones de *s* a *z* :

```
>> [NUMd,DENd] = bilinear(NUM,DEN,Fs)
```

Hace la Transformación bilineal entre la función de Transferencia en s dada por los coeficientes *NUM* y *DEN*, a la función de Transferencia en z dada por *NUMd* y *DENd*. Se debe especificar también la frecuencia de muestreo *Fs* en Hz.

```
>> [Bz,Az] = impinvar(B,A,Fs)
```

Hace la transformación invariante al impulso de la función de Transferencia en *s* definida por los vectores de coeficientes *B* y *A*, a una frecuencia de muestreo de *Fs* Hz.

- Ejemplo: Transformar el filtro analógico de Butterworth diseñado anteriormente en un filtro digital (*Fs=1000 KHz*).

Predistorsionamos las frecuencias que nos piden

[5 30 50 200]Hz → [0.03149 0.18904 0.31676 1.45308] rad

Diseñamos el filtro analógico para estas frecuencias.

```
>> [N,Wn] = buttord(0.12772,1.41187,2,40,'s')
    N = 3     Wn =  0.3042
>> [B,A] = butter(N,[0.18904 0.31676],'s')
    B = [0 0.0000 0.0000 0.0021 0.0000 0.0000 0.0000]
    A = [1.0000 0.2554 0.2123 0.0327 0.0127 0.0009 0.0002]
```

Diseño de Filtros IIR con MATLAB

```
>> [Bd,Ad] = bilinear(B,A,1)
    Bd = 1e-3 *[0.2196 0.0000 -0.6589 0.0000 0.6589 0.0000 -0.2196]
    Ad = [1.0000 -5.5792 13.1338 -16.6913 12.0770 -4.7177 0.7776]
```

Por la respuesta invariante al impulso, diseño el filtro analógico (no hay predistorsión). Los coeficientes de este filtro fueron calculados anteriormente y eran,

```
>> B = 1.0e+6 *[0 0.0000 0.0000 1.9844 0.0000 0.0000 0.0000];
>> A = 1.0e+14 *[0.0000 0.0000 0.0000 0.0000 0.0001 0.0088 2.0766];
>> [Bdi,Adi] = impinvar(B,A,1000);
    Bdi = [-2.5839 2.8862 0.0333 -0.0160]
    Adi = [1.0000 -1.8959 0.9158]
```

Diseño de Filtros IIR con MATLAB

- Como obtener la respuesta frecuencial del filtro diseñado
  - Para los filtros analógicos

```
>> H = freqs(B,A,W)
```

Devuelve el vector *H* de números complejos, que es la respuesta frecuencial al filtro cuya función de transferencia en s viene dada por *B* y *A*. La respuesta frecuencial se evalúa en los puntos especificados por el vector *W* en radianes. Más opciones en el Help de MATLAB.

```
>> plot(W,abs(H))
```

Dibuja la magnitud de la respuesta frecuencial del filtro.

```
>> plot(W,unwrap(angle(H)))
```

Dibuja la fase de la respuesta frecuencial del filtro. La función *unwrap* hace que no haya discontinuidad en la fase por el paso de $+\pi$ a $-\pi$.

Diseño de Filtros IIR con MATLAB

- Para los filtros digitales

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

Devuelve el vector *H* de números complejos, que es la respuesta frecuencial al filtro cuya función de transferencia en z viene dada por *B* y *A*. La respuesta frecuencial se evalúa en los puntos especificados por el vector *F* en Hz, siendo la frecuencia de mustreo *Fs* Hz. Más opciones en el Help de MATLAB.

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

Calcula retraso de grupo ($-d\Phi/dt$) de la función de Transferencia formada por los polinomios *B* y *A*. Se evalúa en los puntos especificados por *W* en radianes. Para más opciones de esta función ver el Help de MATLAB.

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

Dibuja la magnitud de la respuesta frecuencial del filtro.

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

Dibuja la fase de la respuesta frecuencial del filtro.

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

Dibuja el retraso de grupo de la función de Transferencia Discreta.

Diseño de Filtros IIR con MATLAB

- También podemos diseñar filtros digitales mediante métodos recursivos. Uno de estos métodos es el de Yule-Walker, que calcula los coeficientes del filtro de orden N utilizando mínimos cuadrados. Para ello debemos especificar la respuesta deseada para cada frecuencia. En MATLAB la función se denonima “yulewalk”:

```
>> [B,A]=yulewalk(N,F,M)
```

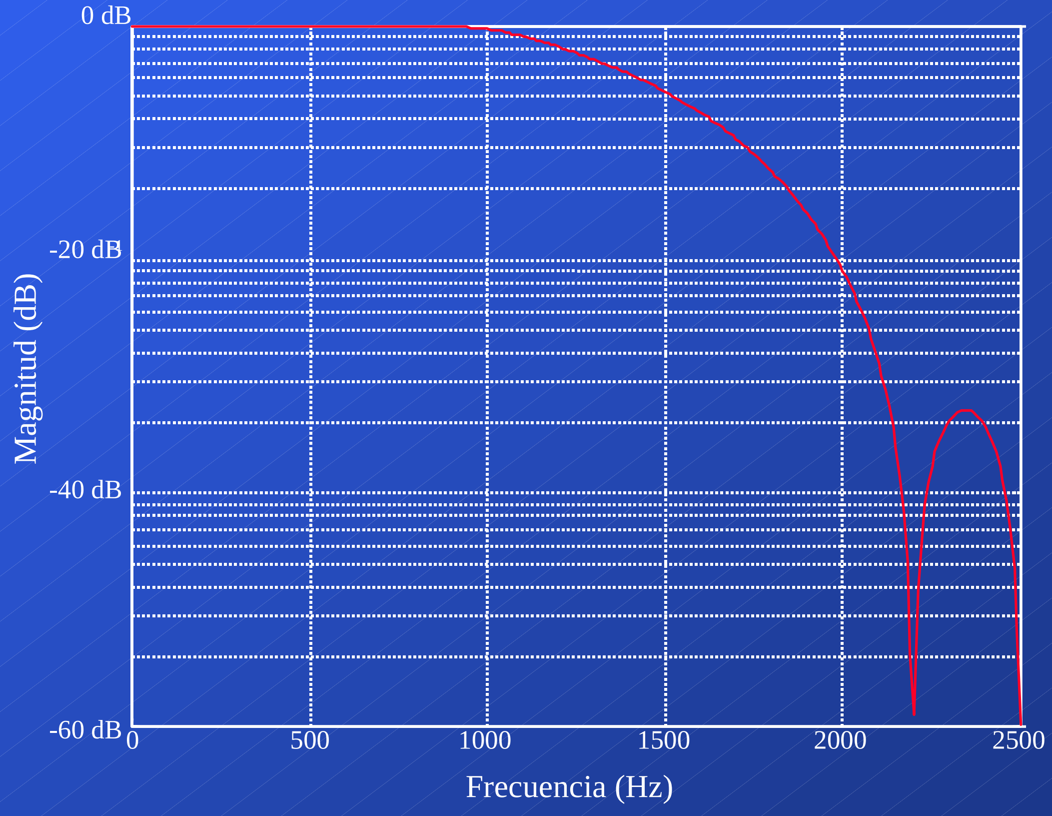
  N es el orden del filtro y *F* y *M* son dos vectores de igual longitud. *F* es la frecuencia normalizada con respecto a la frecuencia de Nyquist (0-1). Debe estar en orden creciente y el primer y último elemento del vector deben ser 0 y 1 respectivamente. *M* es el vector que especifica la maginitud de la respuesta para cada elemento de *F*.

- Ejemplo: Diseñar un filtro digital pasobajo de orden 5 con frecuencia de corte f_c*=1.3 KHz*, por el método de Yule-Walker. Calcular las atenuaciones a *1 KHz* y *2 KHz*.
  - Primero hay que elegir una frecuencia de muestreo. Tomamos f_s*=5 KHz*.

```
>> [B,A]=yulewalk(5,[0 1 1.3 2 2.5]/2.5,[1 1 0.708 0 0])
    B = [0.3155 0.8087 0.7811 0.4675 0.3284 0.1492]
    A = [1.0000 0.6490 0.5046 0.5031 0.1355 0.0577]
>> H = 20*log10(abs(freqz(B,A,[1000 2000],5000)))
    ans = -0.2422    -20.9538    % En decibelios
```

Diseño de Filtros IIR con MATLAB

```
>> F=0:10:2500;Fs=5000;
>> Hz = freqz(B,A,F,Fs);
>> semilogy(F,abs(Hz));
```



Diseño de Filtros IIR con MATLAB

- Finalmente, cuando diseñemos un filtro digital nos interesará poder aplicar ese filtro a una señal temporal. Eso se consigue con la función de MATLAB “*filter*”.

```
>> y = filter(B,A,x)
```

El vector *x* es la entrada y el vector *y* es la salida filtrada. *B* y *A* son los coeficientes del filtro digital.

- Existe otra función llamada *“filtfilt”*, que funciona de la misma manera que *“filter”*, pero hace dos filtrados. Primero filtra el vector *x*, y su respuesta la rota y le vuelve a aplicar el mismo filtro. La respuesta final evita la distorsión de fase propia de los filtros IIR. Más detalles en el Help de MATLAB.

```
>> y = filtfilt(B,A,x)
```

Apéndice

Tablas de Transformación desde un Prototipo de Filtro Pasobajo

Transformaciones desde un Prototipo Pasobajo

Tabla 1

Transformación	***Regla***	***Comentarios***
LP2LP	$s \rightarrow s/\omega_x$	$\omega_x = \omega_{new} / \omega_{old}$
LP2HP	$s \rightarrow \omega_x/s$	$\omega_x = \omega_{new} / \omega_{old}$
LP2BP	$s \rightarrow \dfrac{s^2 + \omega_x^2}{sB_w}$	ω_x=*frecuencia central* $B_w = \omega_2 - \omega_1, \quad \omega 1 \omega 2 = \omega_x^2$
LP2SP	$s \rightarrow \dfrac{sB_w}{s^2 + \omega_x^2}$	ω_x=*frecuencia central* $B_w = \omega_2 - \omega_1, \quad \omega 1 \omega 2 = \omega_x^2$

Transformaciones a un Prototipo Pasobajo

Tabla 2

Transformación de un Prototipo de Filtro Pasobajo a Filtros Pasabanda y Parabanda:
$[\omega_1\ \omega_2\ \omega_3\ \omega_4\ ]$=frecuencias en los bordes de las bandas. En los filtros Pasabanda, los bordes pasabanda son ω_2 *y* ω_3 . En los filtros Parabanda, los bordes pasabanda son ω_1 *y* ω_4.

Requerimientos	***Frecuencia Central***	***Elección de frecuenaias en los bordes de la banda***
Fijadas ω_2,ω_3	$\omega_x^2= \omega_2\omega_3$	*Si* $\omega_1\omega_4< \omega_x^2$, $\omega_1= \omega_x^2/\omega_4$ *Si* $\omega_1\omega_4>\omega_x^2$, $\omega_4= \omega_x^2/\omega_1$
Fijadas ω_1,ω_4	$\omega_x^2= \omega_1\omega_4$	*Si* $\omega_2\omega_3<\omega_x^2$, $\omega_3= \omega_x^2/\omega_2$ *Si* $\omega_2\omega_3>\omega_x^2$, $\omega_2= \omega_x^2/\omega_3$
Fijada ω_x	ω_x	*Si* $\omega_1\omega_4< \omega_x^2$, $\omega_1= \omega_x^2/\omega_4$ *Si* $\omega_1\omega_4>\omega_x^2$, $\omega_4= \omega_x^2/\omega_1$
Compromiso	$\omega_x^2= (\omega_1\omega_2\omega_3\omega_4)^{1/2}$	*Si* $\omega_1\omega_4>\omega_2\omega_3$, $\omega_3=\omega_x^2/\omega_2$ $\omega_4=\omega_x^2/\omega_1$ *Si* $\omega_1\omega_4<\omega_2\omega_3$, $\omega_2=\omega_x^2/\omega_3$ $\omega_1=\omega_x^2/\omega_4$

Utilización de la función para calcular el orden de los filtros IIR analógicos (buttord, cheb1ord y cheb2ord)

Parámetros de entrada Wp,Ws,Ap,As
Especificaciones: wp,ws (filtros PasoBajo y PasoAlto),
[wp1 wp2], [ws1 ws2] (filtros PasaBanda y Parabanda), ap y as
Salidas : [N Wn]

PasoBajo	Wp=wp, Ws=ws, Ap=ap, As=as $\rightarrow$ [B1,A1]=lp2lp(B,A,Wn)
PasoAlto	Wp=wp, Ws=ws, Ap=ap, As=as $\rightarrow$ [B1,A1]=lp2hp(B,A,Wn)
PasaBanda	Wp=[wp1 wp2], Ws=[ws1 ws2], Ap=ap, As=as $\rightarrow$ [B1,A1]=lp2bp(B,A,wx,Bw), $wx^2=wp1\cdot wp2$, Bw=wp2-wp1
ParaBanda	Wp=[wp1 wp2], Ws=[ws1 ws2], Ap=ap, As=as $\rightarrow$ [B1,A1]=lp2bs(B,A,wx,Bw), $wx^2=wp1\cdot wp2$, Bw=wp2-wp1