

# Filtros IIR

Dr. Ing. Hernán Garrido

Control y sistemas  
Universidad Nacional de Cuyo, Facultad de Ingeniería

*carloshernangarrido@gmail.com*

Mayo de 2023



- 1 Repaso de introducción al filtrado digital
- 2 Filtros tipo IIR especificados en el dominio del tiempo
- 3 Filtros tipo IIR especificados en el dominio de la frecuencia
- 4 Implementación de filtros IIR

# Filtrado en el dominio del tiempo y en el dominio de la frecuencia

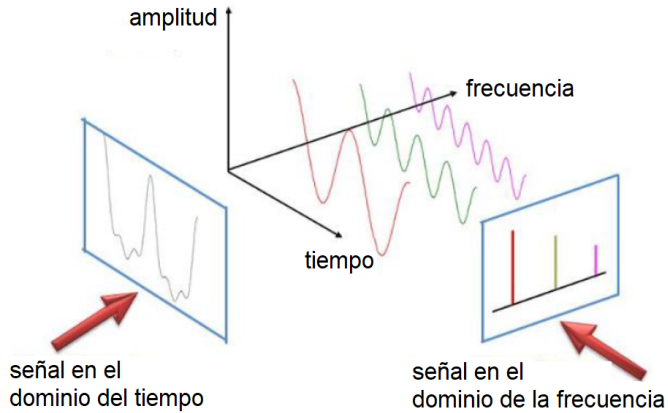


Figura: ¿Cómo está codificada la información en la señal a filtrar?

# Clasificación de filtros digitales

<b>Tipo de especificaciones</b>	<b>Respuesta impulsiva finita - Finite Impulse Response (FIR)</b>	<b>Respuesta impulsiva infinita - Infinite Impulse Response (IIR)</b>
<b>Dominio del tiempo</b>	media móvil - moving average	<b>integrador con pérdida - leaky integrator</b>
<b>Dominio de la frecuencia</b>	seno cardinal con ventana - windowed sinc	transformada bilineal - método de Tustin

# Filtro integrador con pérdida - leaky integrator filter

## Primera intuición

Abordaje: Aproximar el filtro de media móvil con uno que use recursión en lugar de convolución.  
Resultado: Un filtro de primer orden (análogo a una red RC, pero discreto).

Recordemos el filtro de media móvil:

$$y[n] = x[n] * h[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[n-k] = \frac{1}{M} \left( \sum_{k=1}^{M-1} x[n-k] + x[n] \right)$$



Armemos otro para  $y[n-1]$ , pero con ventana de longitud  $M-1$ :

$$y[n-1] = \frac{1}{M-1} \sum_{k=1}^{M-1} x[n-k]$$

# Filtro integrador con pérdida: ecuación en diferencias

Sustituyendo la segunda ecuación en la primera, obtenemos:

$$y[n] = \frac{M-1}{M}y[n-1] + \frac{1}{M}x[n] \quad (5)$$

Definiendo el parámetro  $\lambda = \frac{M-1}{M}$ :

$$y[n] = \lambda y[n-1] + (1-\lambda)x[n] \quad (6)$$



- Ecuación en diferencias recursiva
- Promedio ponderado de la entrada actual y la salida anterior
- Un solo parámetro

# Filtro integrador con pérdida: función de transferencia, función de respuesta en frecuencia y respuesta impulsiva

$$H(z) = \frac{1 - \lambda}{1 - \lambda z^{-1}} = \frac{b_0}{a_0 + a_1 z^{-1}}$$

$$H(e^{j\omega}) = \frac{1 - \lambda}{1 - \lambda e^{-j\omega}}$$

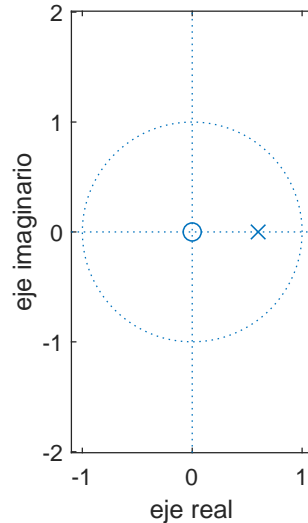
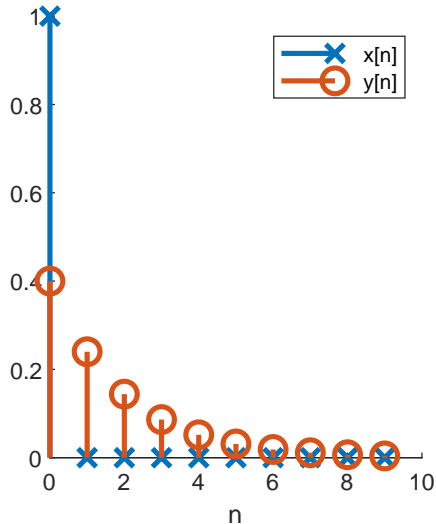
$$h[n] = (1 - \lambda)\lambda^n u[n]$$



- Causal
- Respuesta impulsiva infinita
- El único polo es real y está en  $\lambda = \frac{M-1}{M}$ 
  - Si  $|\lambda| < 1 \implies$  Estable
- Frecuencia de corte de -3dB:  $\omega_{co} \approx \frac{\pi}{M} = \pi(1 - \lambda)$



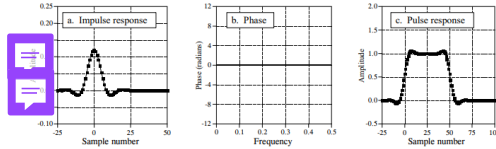
# Filtro integrador con pérdida: respuesta impulsiva y diagrama de polos



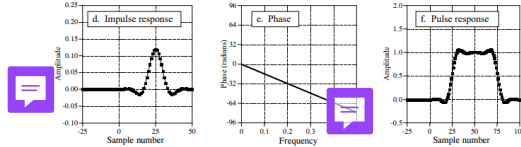


# Respuesta en fase: Repaso

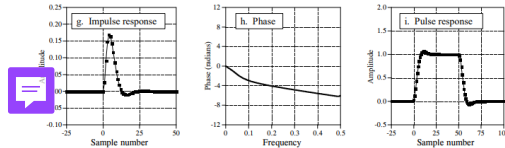
## Filtro de fase cero



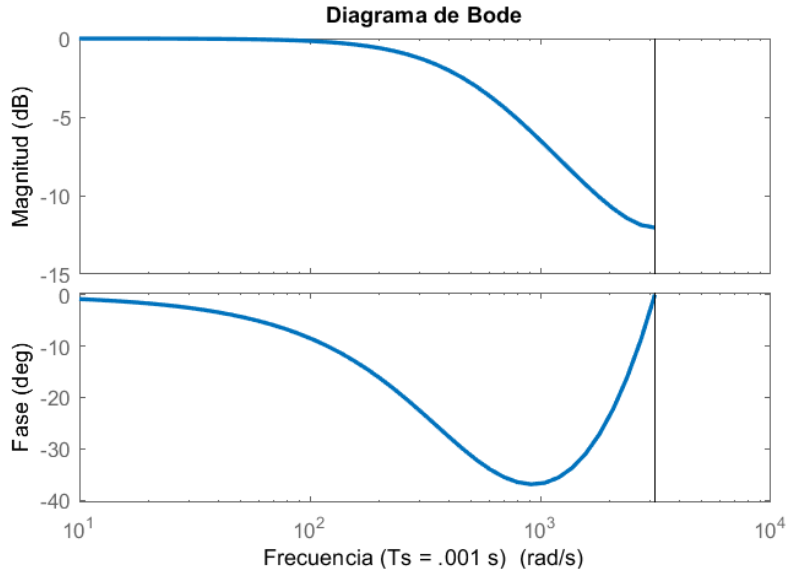
## Filtro de fase lineal



## Filtro de fase no lineal



# Filtro integrador con pérdida: función de respuesta en frecuencia



- Steven W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing. Chapters 14, 15, 16, 19. California Technical Publishing. [www.dspguide.com](http://www.dspguide.com).
- Oppenheim, Schafer (2010). Discrete-Time Signal Processing. Prentice Hall
- Oppenheim, A. V., Willsky, A. S., and Young, I. T. (1983). Signals and Systems. Prentice-Hall, Inc.
- Material de cátedra generado por el Dr. Rodrigo Gonzalez hasta 2022.

# Clasificación de filtros digitales

<b>Tipo de especificaciones</b>	<b>Respuesta impulsiva finita - Finite Impulse Response (FIR)</b>	<b>Respuesta impulsiva infinita - Infinite Impulse Response (IIR)</b>
<b>Dominio del tiempo</b>	media móvil - moving average	integrador con pérdida - leaky integrator
<b>Dominio de la frecuencia</b>	seno cardinal con ventana - windowed sinc	<b>transformada bilineal - método de Tustin</b>

# Especificaciones en el dominio de la frecuencia

Filter Specifications

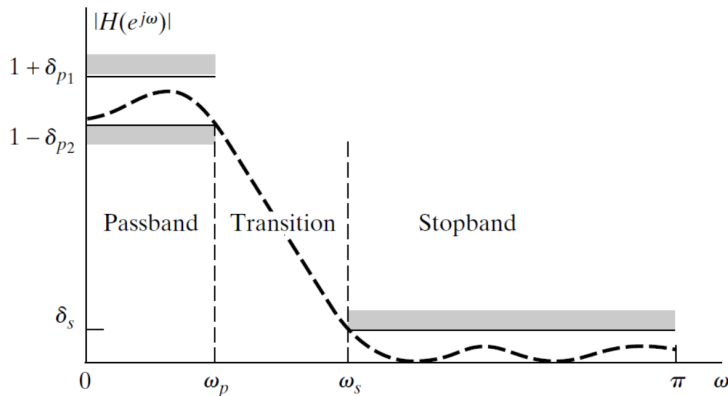


Figura: Tomado de Oppenheim, Schafer (2010). Discrete-Time Signal Processing. Prentice Hall

# Método de Tustin: Transformada bilineal

- Transformación bilineal: transformación algebraica entre las variables  $s$  y  $z$  que mapea todo el eje imaginario  $j$  en el plano  $s$ , a una revolución completa del círculo unitario en el plano  $z$ .
- 

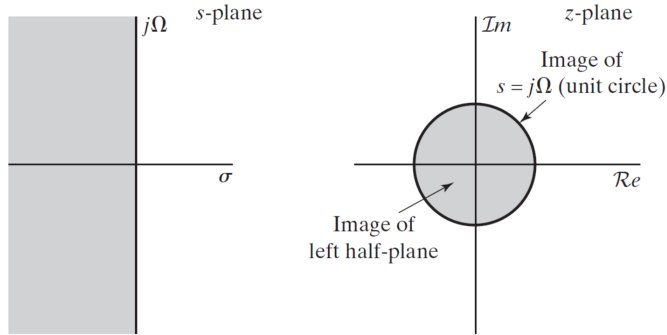


Figura: Tomado de Oppenheim, Schafer (2010). Discrete-Time Signal Processing. Prentice Hall

## Ventaja de este mapeo o sustitución

Un filtro analógico  $H_c(s)$  diseñado para cumplir con ciertas especificaciones en el dominio de la frecuencia se puede transformar en un filtro digital  $H(z)$ . ¡Podemos usar toda la teoría clásica de filtros analógicos!

$$s = \frac{2}{T_d} \left( \frac{1 - z^{-1}}{1 + z^{-1}} \right) \longleftrightarrow z = \frac{1 + \frac{T_d}{2}s}{1 - \frac{T_d}{2}s}$$

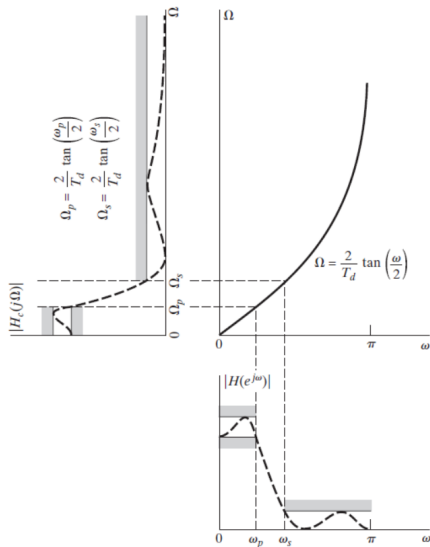
$$H(z) = H_c(s) \Big|_{s = \frac{2}{T_d} \left( \frac{1 - z^{-1}}{1 + z^{-1}} \right)}$$

$$\Omega = \frac{2}{T_d} \tan(\omega/2) \longleftrightarrow \omega = 2 \arctan(\Omega T_d/2)$$

Como  $-\infty \leq \Omega \leq \infty$  se mapea en  $-\pi \leq \omega \leq \pi$ , la transformación entre las frecuencias es necesariamente no lineal.

<sup>1</sup>El valor del parámetro  $T_d$  no tiene consecuencias en el diseño.

# Método de Tustin: Diseño por pre-combado de frecuencias



- 1. Elija el filtro analógico que cumpla con el rendimiento deseado. Por ejemplo, un filtro Butterworth de segundo orden paso bajo.

$$H_c(s) = \frac{\Omega_c^2}{s^2 + s\sqrt{2}\Omega_c + \Omega_c^2}$$

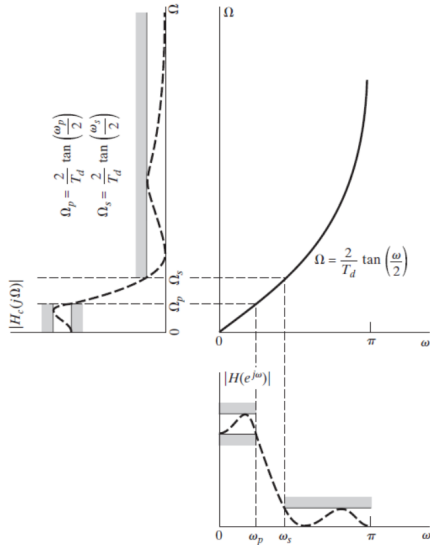
- 2. Encuentre la frecuencia de corte digital:

$$f_{co} = 100 \text{ Hz}, f_s = 1000 \text{ Hz}$$

$$\omega_c = f_{co} \frac{\pi}{f_s/2} = 0.628 \text{ rad}$$



# Método de Tustin: Diseño por pre-combado de frecuencias (cont.)



- 3. Pre-combe las frecuencias analógicas:

$$\Omega_c = \frac{2}{T_d} \tan\left(\frac{\omega_c}{2}\right) = 649.8 \text{ rad/s}$$

$$f_c = 103.42 \text{ Hz}$$

- 4. Construya el filtro analógico (de diseño)

$$H_c(s) = \frac{649.8^2}{s^2 + 649.8\sqrt{2}s + 649.8^2}$$

## Método de Tustin: Diseño por pre-combado de frecuencias (cont.)

- 5. Aplique la transformación bilineal  $s = \frac{2}{T_d} \left( \frac{1-z^{-1}}{1+z^{-1}} \right)$  a  $H_c(s)$  para obtener el filtro digital  $H(z)$ :

$$H(z) = \frac{649.8^2}{\left( \frac{2}{T_d} \left( \frac{1-z^{-1}}{1+z^{-1}} \right) \right)^2 + 649.8\sqrt{2} \left( \frac{2}{T_d} \left( \frac{1-z^{-1}}{1+z^{-1}} \right) \right) + 649.8^2}$$

- 6. Reordene los términos de numerador y denominador:

$$H(z) = \frac{0.067 + 0.135z^{-1} + 0.067z^{-2}}{1 - 1.143z^{-1} + 0.413z^{-2}}$$

- 7. Mediante la transformada z inversa, encuentre la ecuación en diferencias:

$$y[n] = 0.067x[n] + 0.135x[n-1] + 0.067x[n-2] + 1.143y[n-1] - 0.413y[n-2]$$

# Método de Tustin: Diseño por pre-combado de frecuencias (cont.)

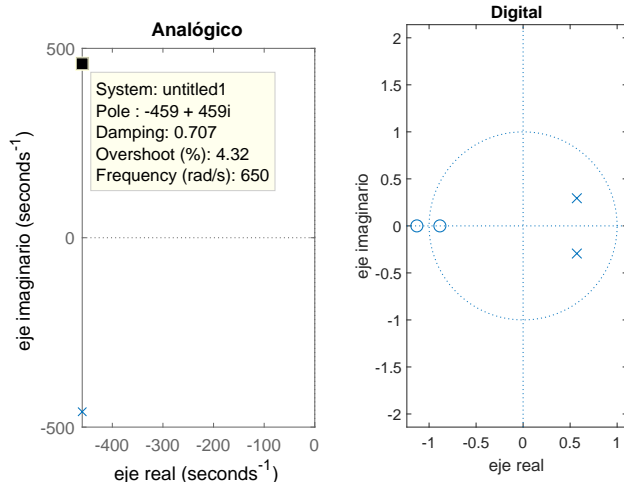
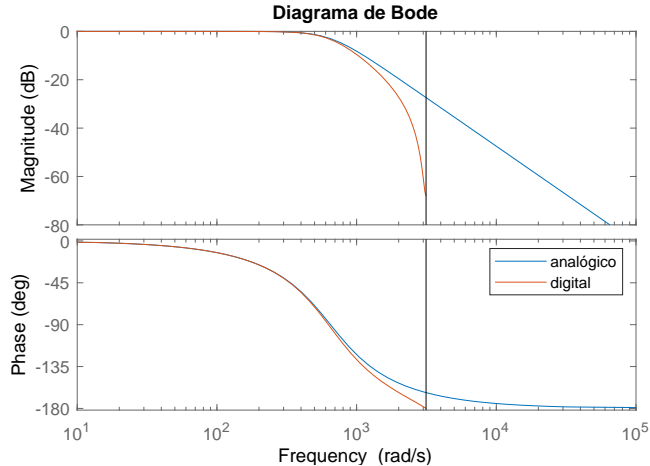


Figura: Mapeo s-z

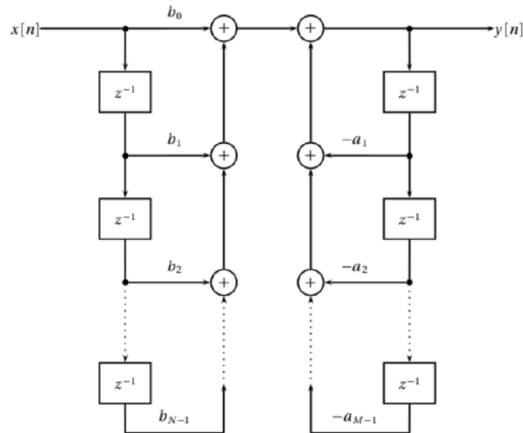
# Método de Tustin: Diseño por pre-combado de frecuencias (cont.)



**Figura:** Función de respuesta en frecuencia

# Estructuras típicas para implementar filtros IIR: Forma Directa I

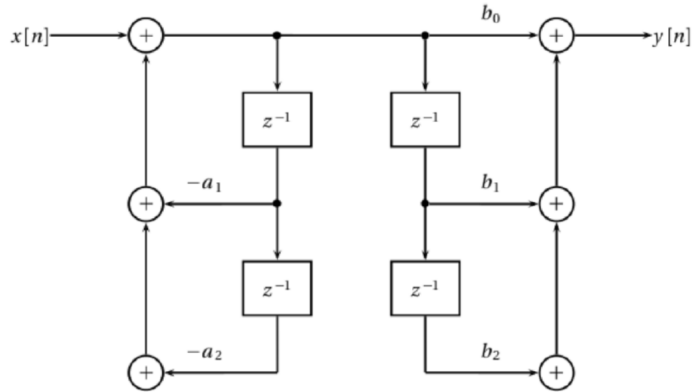
$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_{N-1} z^{N-1}}{1 + a_1 z^{-1} + \dots + a_{M-1} z^{M-1}}$$



# Estructuras típicas para implementar filtros IIR: Forma Directa I invertida

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

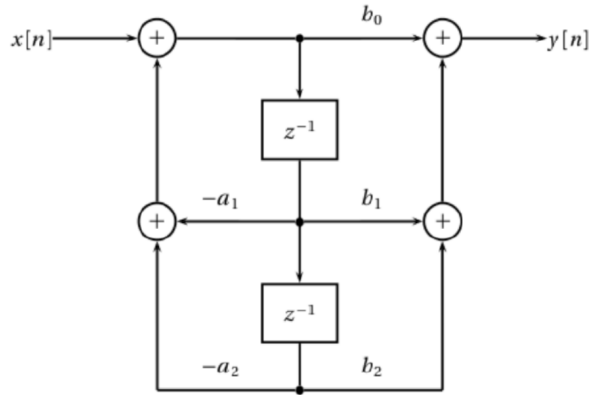
By the commutative properties of the z-transform, we can invert the order of computation to turn the Direct Form I structure into a new structure.



# Estructuras típicas para implementar filtros IIR: Forma Directa II

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

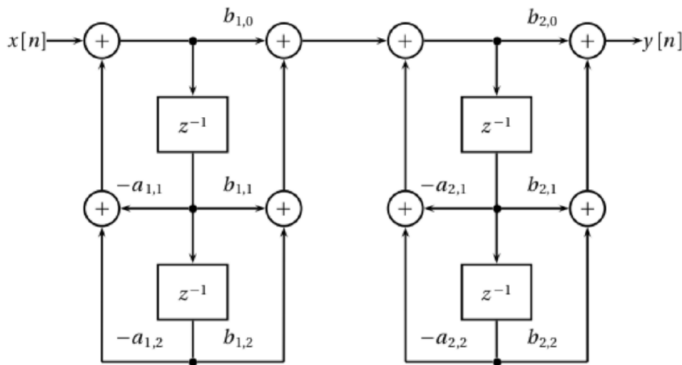
We can then combine the parallel delays together. This implementation is called Direct Form II; its obvious advantage is the reduced number of the required delay elements (hence of memory storage).



# Estructuras típicas para implementar filtros IIR: Cascada de secciones de segundo orden

The cascade structure of  $N$  second-order sections is much less sensitive to quantization errors than the previous Direct form II of order  $2 \cdot N$ .

$$H(z) = \prod_{k=1}^N G_k \frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}}$$





## Receta de cocina

- 1 Diseñamos el filtro usando el método de Tustin o con Matlab/filterDesigner
- 2 Generamos un sistema en cascada de secciones de segundo orden y generamos un archivo matrix.h o exportamos la matriz de parámetros correspondiente con Targets/Generate C header
- 3 Incluimos el header (matrix.h) en una plantilla *provista* para filtro online u offline, con punto fijo o punto flotante, escrita en c (filter.c).

- Steven W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing. Chapters 14, 15, 16, 19. California Technical Publishing. [www.dspguide.com](http://www.dspguide.com).
- Oppenheim, Schafer (2010). Discrete-Time Signal Processing. Prentice Hall
- Oppenheim, A. V., Willsky, A. S., and Young, I. T. (1983). Signals and Systems. Prentice-Hall, Inc.
- Material de cátedra generado por el Dr. Rodrigo Gonzalez hasta 2022.