Introducción al filtrado digital y Filtros FIR

Dr. Ing. Hernán Garrido

Control y sistemas
Universidad Nacional de Cuyo, Facultad de Ingeniería

carloshernangarrido@gmail.com

Mayo de 2023

FACULTAD
DE INGENIERÍA

UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO

Contenidos

Definición de filtro

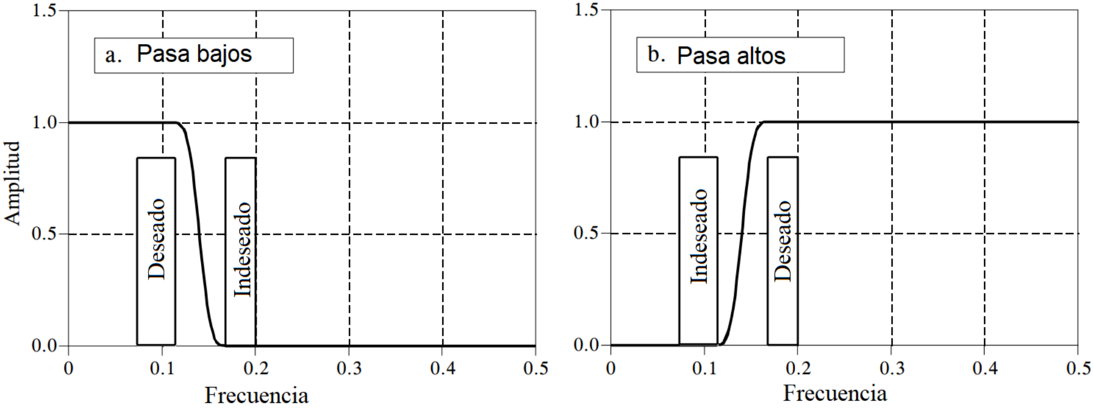
Definición a los efectos de este curso

Filtro: Sistema con una señal de entrada y una señal de salida cuyo propósito es:

- Separar señales que han sido combinadas, o
  - Eliminar ruido de línea
  - Eliminar componente de continua
  - Filtro anti-aliasing en ADC
- Restaurar señales que han sido distorsionada.
  - Suavizar
  - Filtro de reconstrucción en DAC

Clasificaciones

- Según la banda pasante
  - Pasa bajos
  - Pasa altos
  - Pasa banda
  - Suprime banda



Clasificaciones

- Modelo matemático
  1. Lineales: Veremos algunos en este curso.

$$\sum_{k=0}^{N} a_k \frac{d^k y(t)}{dt^k} = \sum_{k=0}^{M} b_k \frac{d^k x(t)}{dt^k} \qquad H(s) = \frac{\sum_{k=0}^{M} b_k s^k}{\sum_{k=0}^{N} a_k s^k}$$

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k] \qquad H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}}$$

  2. No lineales: Filtro de mediana, filtro de Kalman extendido, etc.

$$\dot{z}(t) = f(z(t), x(t), t), y(t) = g(z(t), x(t), t)$$

- Implementación
  1. Analógicos: Las señales son magnitudes físicas reales.
     FRF: $H(j\Omega)$, FT: $H(s)$. ¡Cuidado con la nomenclatura en DSP!
  2. Digitales: Las señales están discretizadas en abscisa (muestreo) y en ordenada (cuantización).
     FRF: $H(e^{j\omega})$, FT: $H(z)$.

Clasificaciones (cont.)

- Causalidad
  1. Causal = no anticipativo: $h[k] = 0 \ \forall k < 0$ [1]
  2. Anticausales = anticipativo
- Modo de operación
  1. On-line: Aplicaciones en tiempo real, como los sistemas de control automático.
  2. Off-line: Procesos por lotes, como el análisis de datos.
- Finitud de la respuesta impulsiva y ecuación en diferencias

FIR	$\exists K : h[k] = 0 \ \forall k > K$	convolución
IIR	$\nexists K : h[k] = 0 \ \forall k > K$	recursión

- Especificaciones
  1. En el dominio del tiempo
  2. En el dominio de la frecuencia

[1] Respuestas impulsivas y FTs: $H(z) = z\{h[k]\}$

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

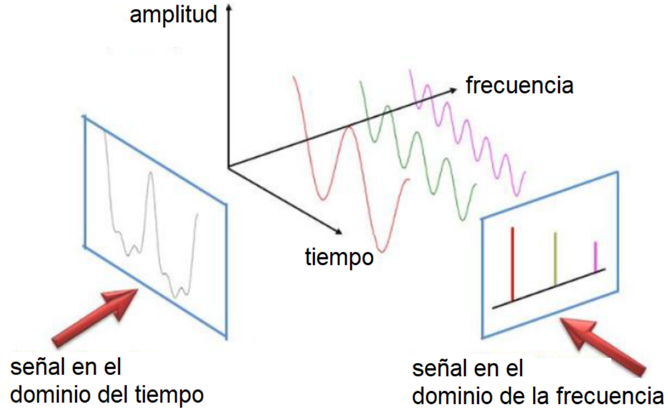


Figura: ¿Cómo está codificada la información en la señal a filtrar?

Filtrado en el dominio del tiempo y en el dominio de la frecuencia (cont.)

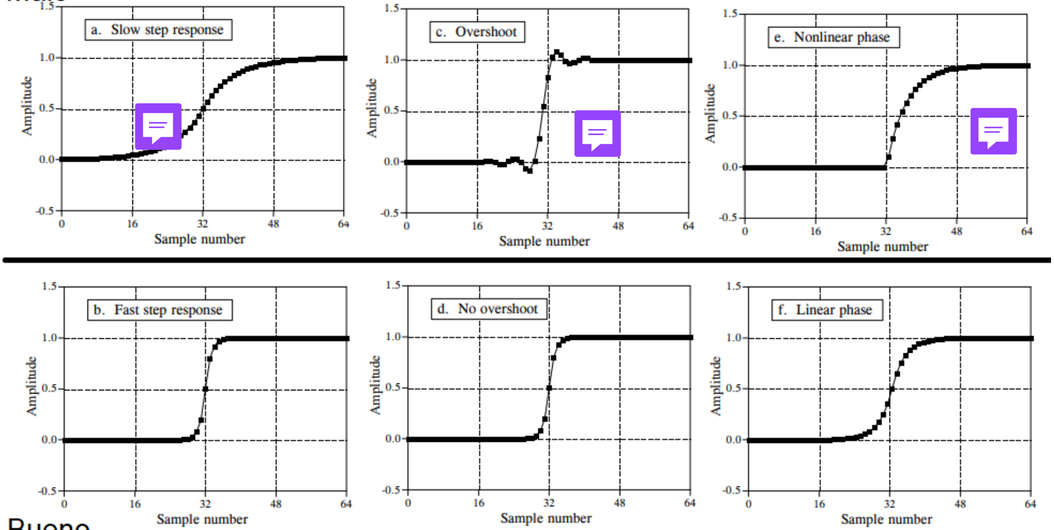
- En el tiempo:
  ✓ Diseñamos un filtro para corregir una distorsión
  ✓ Lo especificamos en el dominio del tiempo:
    - tiempo de subida
    - sobre-pico
    - retardo
- En la frecuencia:
  ✓ Diseñamos un filtro para separar contenidos de frecuencias
  ✓ Lo especificamos en el dominio de la frecuencia:
    - frecuencia de corte
    - ondulación en la banda pasante
    - ancho de la banda de transición
    - ondulación en la banda suprimida
- En ambos: ¡Desafío ingenieril!

Clasificación de filtros digitales

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

Comportamientos en el dominio del tiempo

Malo



Bueno

Filtro de media móvil - moving average filter

Definición:

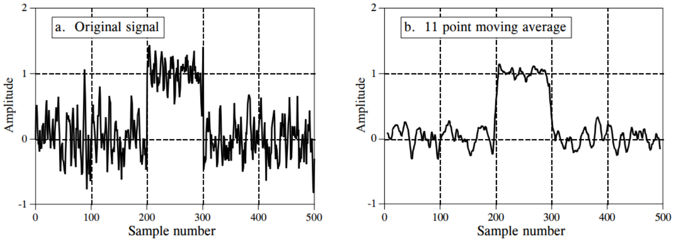
$$h[n] = \frac{1}{M} \sum_{k=0}^{M-1} \delta[n-k]$$

$$h[n] = \begin{cases} \frac{1}{M} & \text{si } 0 \leq n < M \\ 0 & \text{en otro caso} \end{cases}$$

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

Longitud del kernel $= M \in \mathbb{N}$

Filtro de media móvil en el dominio del tiempo



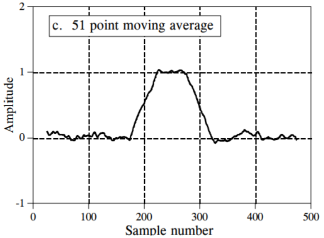
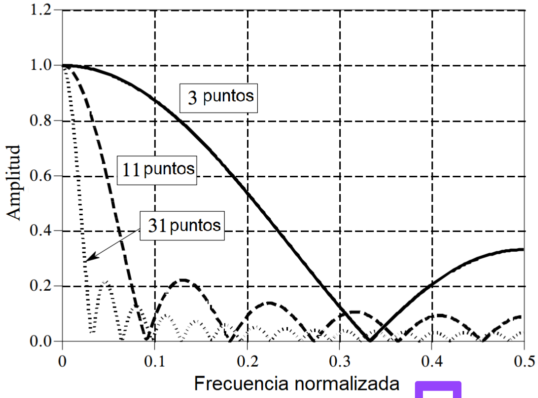


FIGURE 15-1
Example of a moving average filter. In (a), a rectangular pulse is buried in random noise. In (b) and (c), this signal is filtered with 11 and 51 point moving average filters, respectively. As the number of points in the filter increases, the noise becomes lower; however, the edges becoming less sharp. The moving average filter is the *optimal* solution for this problem, providing the lowest noise possible for a given edge sharpness.

Simple, pero óptimo: La respuesta más rápida para una reducción de ruido blanco dada.

Filtro de media móvil: en el dominio de la frecuencia

$$|H(F)| = \left|\frac{\sin(M\pi F)}{M\sin(\pi F)}\right|, \quad |H(e^{j\omega})| = \left|\frac{\sin(M\omega/2)}{M\sin(\omega/2)}\right| \text{ (seno cardinal periódico)}$$



Frecuencia de tiempo discreto: $\omega \in [0, pi]$ (rad)
Frecuencia de tiempo continuo: $\Omega \in [0, \Omega_s/2]$ (rad/s)
Frecuencias normalizadas: $F = f/f_s \in [0, 1/2]$, $F_{ns} = 2f/f_s \in [0, 1]$

Filtro de media móvil en el dominio de la frecuencia

Frecuencia de corte de -3 dB (ventana = medio ciclo)

$$\sqrt{2}/2 = \left| \frac{\sin(\pi F_{co} M)}{M \sin(\pi F_{co})} \right|$$

No tiene solución analítica, pero se puede aproximar como:

$$\omega_{co} \approx \pi / M$$

$$F_{co} \approx \frac{\omega_{co}}{2\pi} = \frac{1}{2M}$$

Primer cero en la FRF (ventana = un ciclo)

$$F_{1er0} = \frac{1}{M}$$

$$f_{1er0} = \frac{f_s}{M}$$

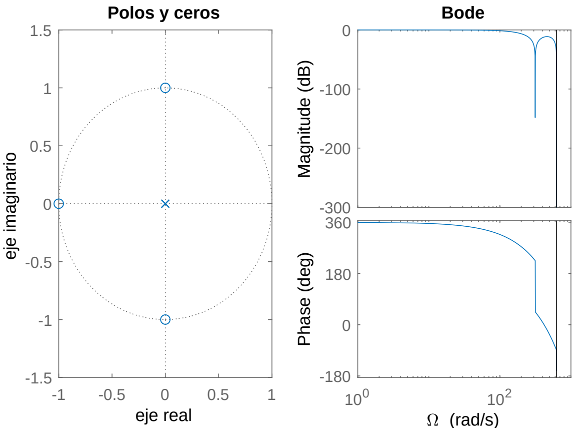
Filtro de media móvil: Ejemplo en Matlab

Filtro anti-aliasing digital en una conversión AD con oversampling:

- Señal de interés: tonos de 1 y 2 Hz
- Contaminantes: ruido de línea de 50 Hz
- Frecuencia de muestreo del ADC: 200 Hz
- Decimación: 200 Hz $\rightarrow$ 30 Hz
- Nueva frecuencia de Nyquist-Shannon: 15 Hz $<$ 50 Hz

Filtro de media móvil: Ejemplo en Matlab (una solución posible)

$$H(z) = 0.25 + 0.25z^{-1} + 0.25z^{-2} + 0.25z^{-3} = 0.25\frac{z^3 + z^2 + z + 1}{z^3}$$



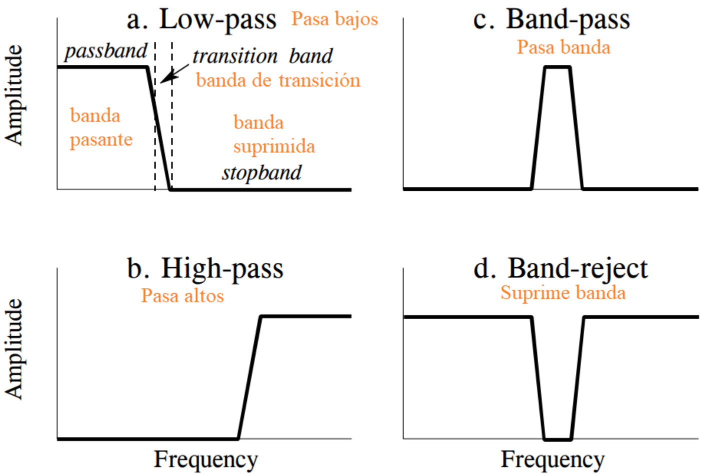
Referencias

- Steven W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing. Chapters 14, 15, 16, 19. California Technical Publishing. 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.

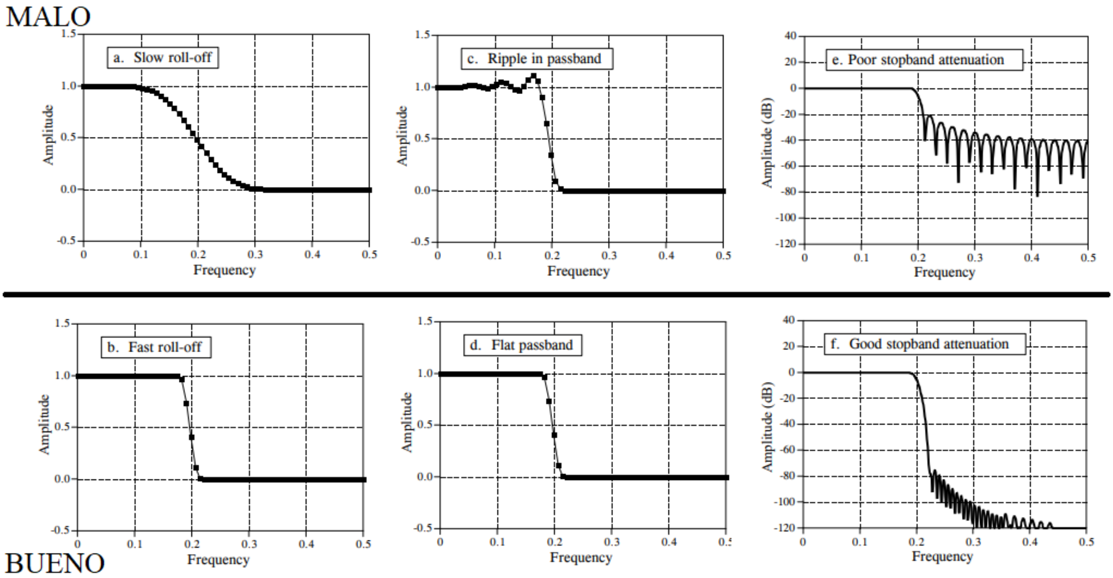
Clasificación de filtros digitales

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

Clasificación de filtros especificados en el dominio de la frecuencia según su banda pasante



Comportamientos en el dominio de la frecuencia



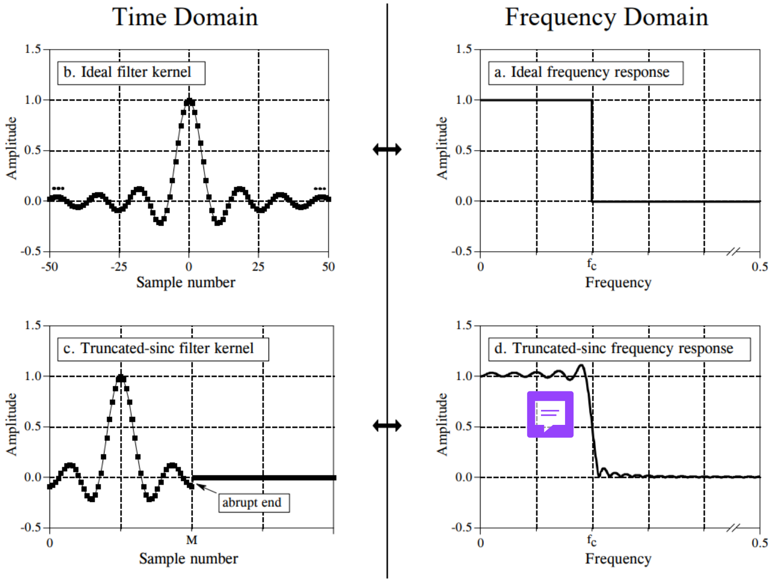
Filtro Ideal y Función sinc

- Tomando la Transformada Inversa de Fourier de una FRF ideal (rectángulo), se obtiene el siguiente kernel (respuesta impulsiva):

$$h[n] = \frac{\sin(2\pi F_c n)}{\pi n}$$

- $y[n] = x[n] * h[n]$ resulta un filtro pasa bajos perfecto. *Cuando la limosna es grande ...*

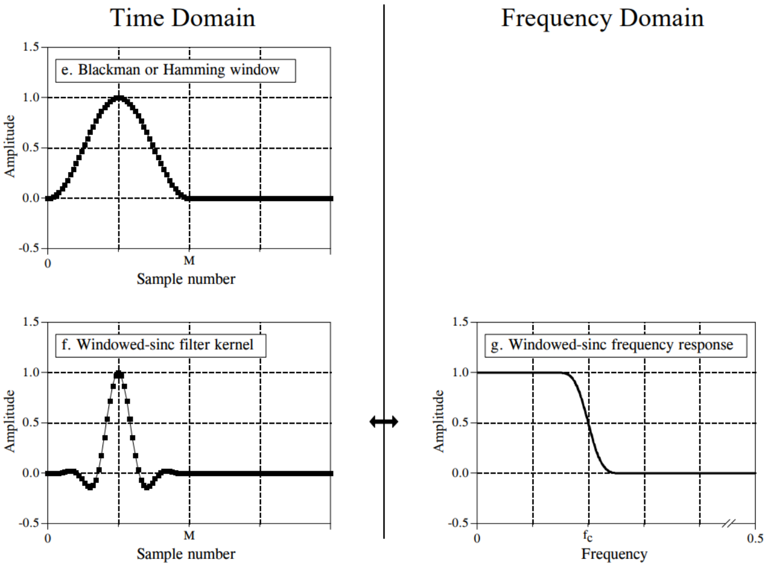
Filtro Ideal y Función sinc



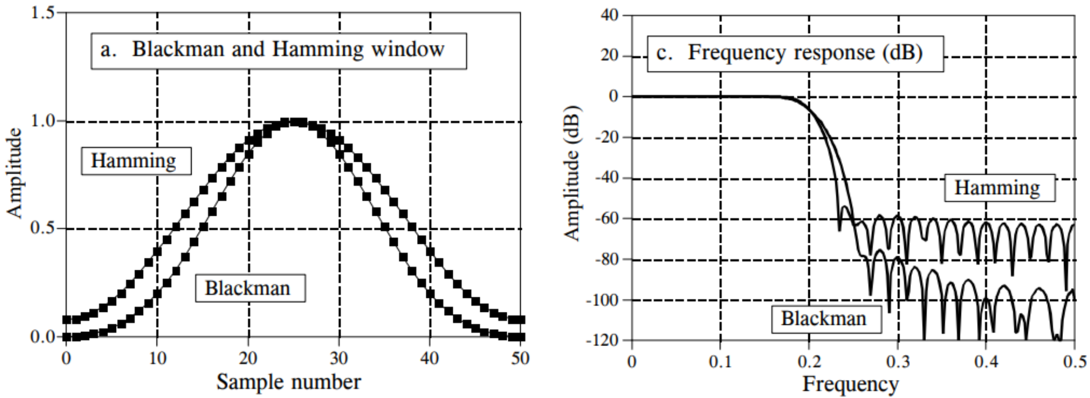
Filtro Ideal y Función sinc truncada

- Problema 1: la función sinc continúa tanto hacia $-\infty$ como hacia ∞ sin llegar a cero.
  - Sistema anticausal
  - Memoria ROM infinita para almacenar el kernel
  - Memoria RAM infinita para almacenar los resultados parciales
  - Tiempo computacional infinito para completar una convolución
- ✓Solución: desplazar el sinc hacia la derecha y truncarlo simétricamente respecto a su máximo global.
- Problema 2: gran ondulación en la banda pasante.
- ✓Solución: suavizar los cortes abruptos del sinc truncado mediante una función ventana

Filtro seno cardinal con ventana



Elección de la ventana



- Hamming: produce un decaimiento aproximadamente un 20 % más rápido que la ventana de Blackman.
- Blackman: tiene mejor atenuación de la banda de suprimida (Blackman: 0.02 %, Hamming: 0.2 %) y menor ondulación en la banda pasante (Blackman: 0.02 %, Hamming: 0.2 %).

Filtro seno cardinal con ventana: Ejemplo 2 en Matlab con filterDesigner

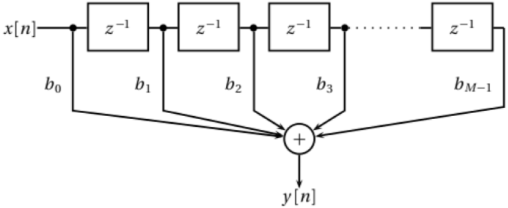
Filtro anti-aliasing digital en una conversión AD con oversampling:

- Señal de interés: tonos de 1 y 2 Hz
- Contaminantes: ruido de línea de 50 Hz
- Frecuencia de muestreo del ADC: 200 Hz
- Decimación: 200 Hz $\rightarrow$ 30 Hz
- Nueva frecuencia de Nyquist-Shannon: 15 Hz $<$ 50 Hz

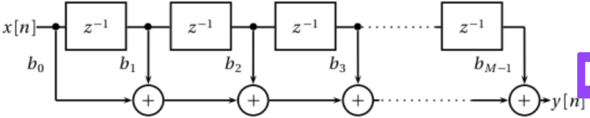
Estructuras típicas para implementar filtros FIR

$$H(z) = b_0 + b_1 z^{-1} + \ldots + b_{M-1} z^{-(M-1)}$$

Implementación directa



Implementación transversal



Implementación de filtros FIR en lenguaje C

Receta de cocina

1. Diseñamos el filtro con Matlab/filterDesigner
2. Exportamos el kernel con Targets/Generate C header
3. Incluimos el header (kernel.h) en una plantilla *provista* para filtro online u offline, con punto fijo o punto flotante, escrita en c (filter.c).

Referencias

- Steven W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing. Chapters 14, 15, 16, 19. California Technical Publishing. 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.