



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Implementación de Filtros Recursivos en FPGA

Procesado Digital de la Señal en FPGA

Máster Universitario en Ingeniería de Sistemas Electrónicos

Objetivos

- Diseñar e implementar filtros recursivos aplicando las técnicas adecuadas para cuantificar y segmentar los circuitos

Contenidos

- Filtros IIR. Estructuras
- Cuantificación
- Segmentación en filtros IIR
- Filtro IIR Secuencial
- Ciclos límite

Filtros IIR

$$y(n) = \sum_{k=0}^{M-1} b_k x(n-k) - \sum_{k=1}^{N-1} a_k y(n-k)$$

Ventajas frente a los Filtros FIR

- Mejor respuesta en frecuencia para el mismo orden
- Pueden implementarse filtros muy selectivos con ordenes pequeños

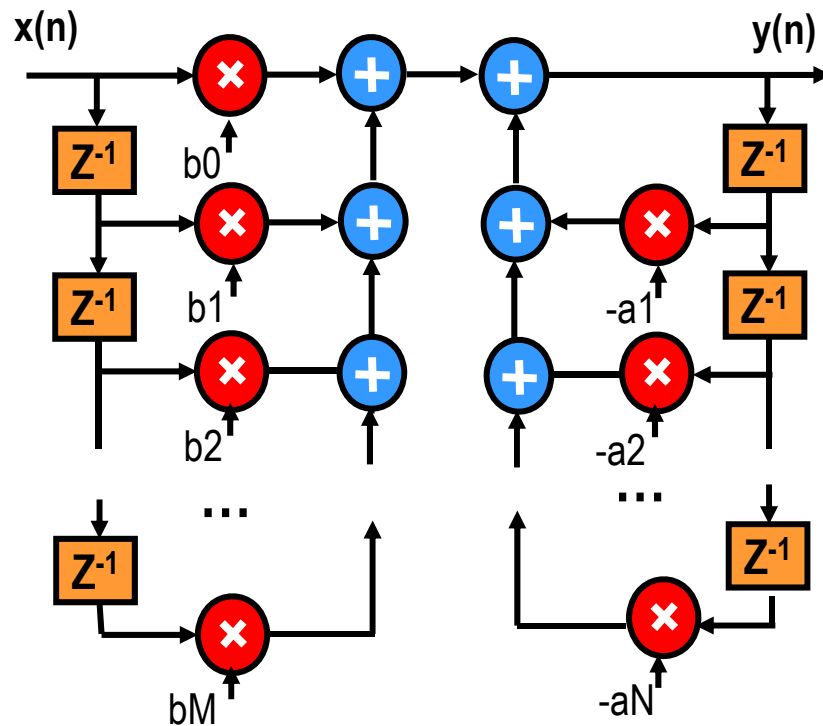
$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} \cdots + b_M z^{-M}}{1 + a_1 z^{-1} \cdots + a_N z^{-N}}$$

Ceros
Polos

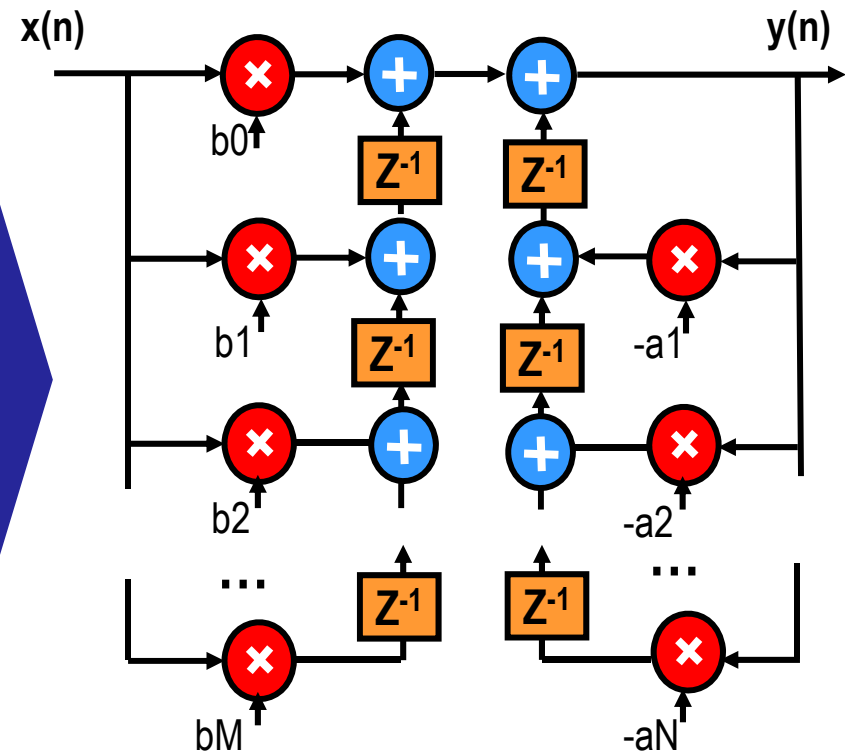
Estructuras IIR

$$y(n) = \sum_{k=0}^M b_k x(n-k) - \sum_{k=1}^N a_k y(n-k)$$

Directa I



Transpuesta

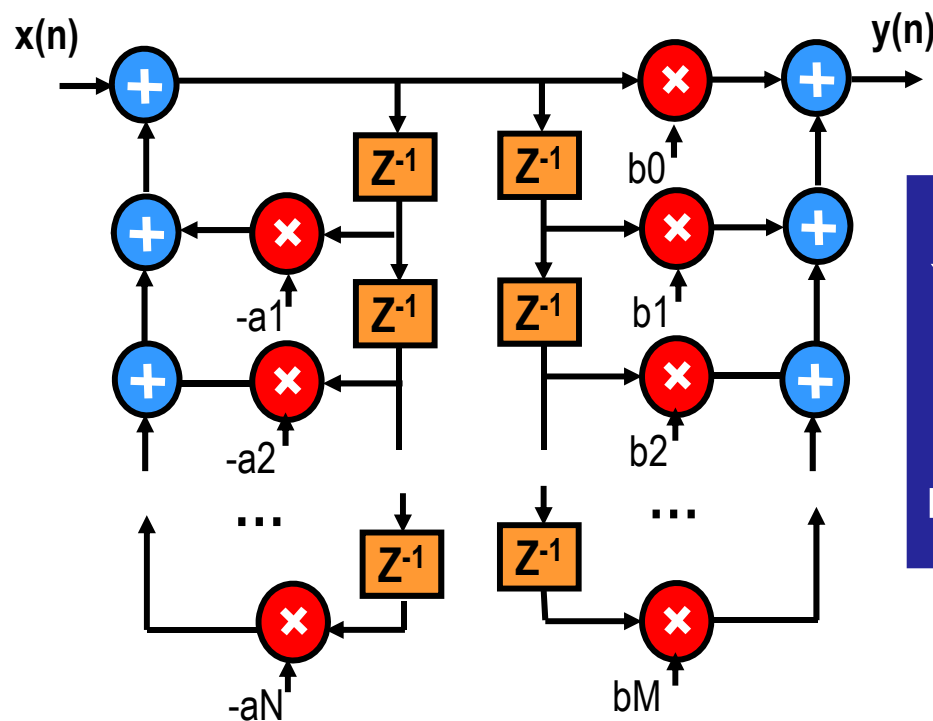


Feedforward + Feedback

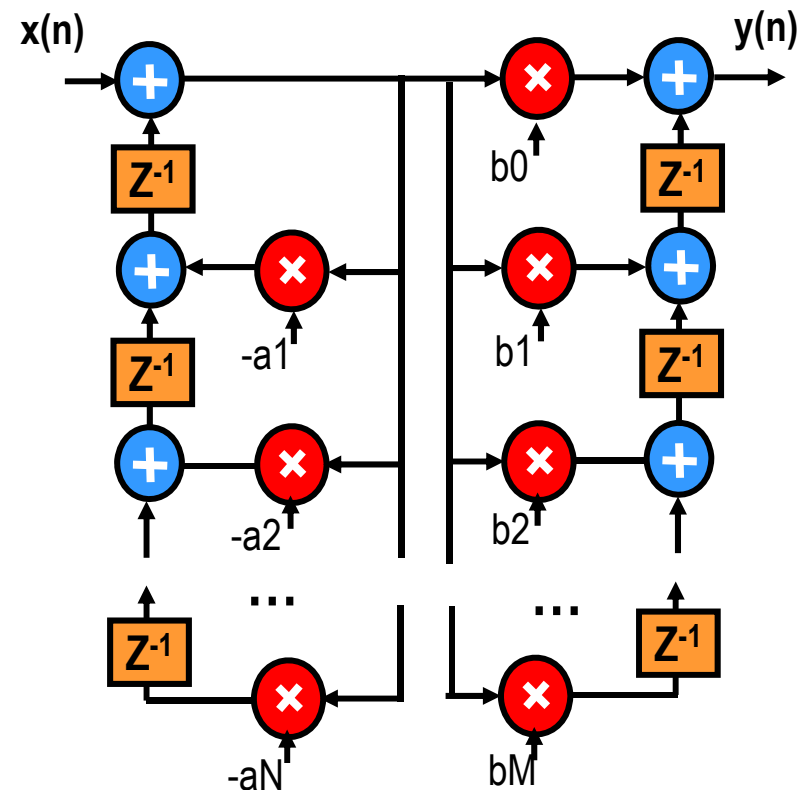
Estructuras IIR

$$y(n) = \sum_{k=0}^M b_k x(n-k) - \sum_{k=1}^N a_k y(n-k)$$

Directa II



Transpuesta

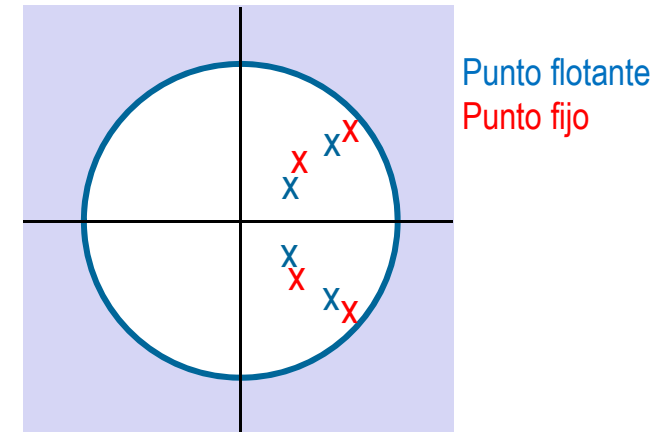


Feedback + Feedforward

Estructuras IIR

$$y(n) = \sum_{k=0}^M b_k x(n-k) - \sum_{k=1}^N a_k y(n-k)$$

- La cuantificación de los coeficientes modifica la posición de los polos/ceros:
 - Cambia la función de transferencia (respuesta en frecuencia)
 - Puede llegar a ser inestable
- Estas estructuras no se utilizan tal cual por ser muy sensibles a la cuantificación de los coeficientes:
 - Todos los polos (ceros) dependen unos de otros
 - Aumenta el error con el orden del filtro



SOLUCIÓN: Descomponer en filtros de menor orden (2^o) dispuestos en cascada:

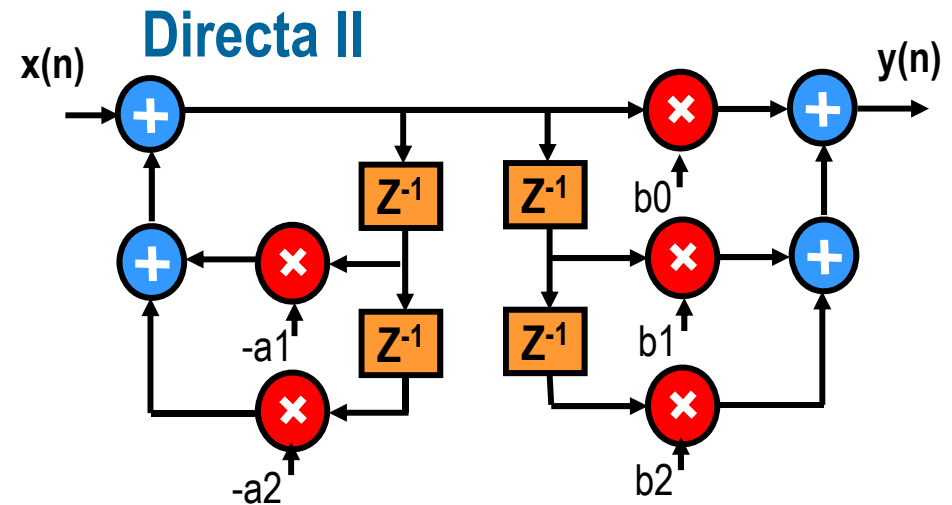
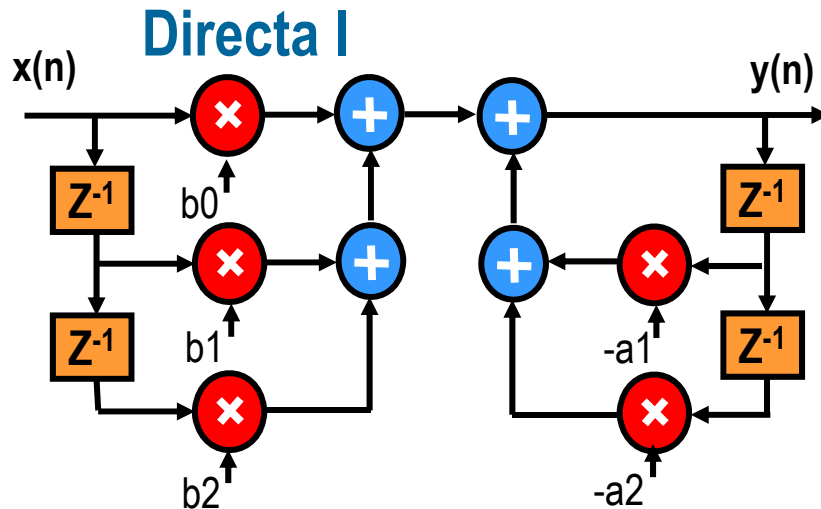
- Los polos/ceros de cada subfiltro son independientes
- Mejora la sensibilidad a la cuantificación de los coeficientes

$$H(z) = \prod_{i=1}^L H_{\text{SOSi}}(z)$$

Estructuras IIR

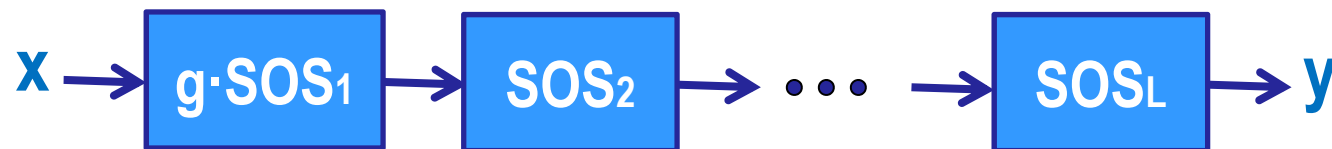
Celda Biquad IIR o SOS (Second Order Section)

$$H_{\text{SOS}_i}(z) = \left(\frac{b_{i0} + b_{i1} \cdot z^{-1} + b_{i2} \cdot z^{-2}}{1 + a_{i1} \cdot z^{-1} + a_{i2} \cdot z^{-2}} \right)$$



Descomposición en Biquad en Cascada

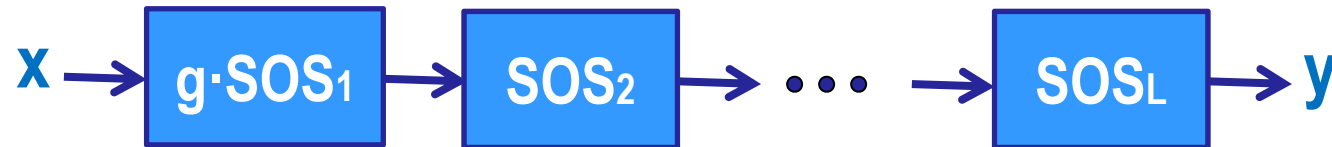
$$H(z) = g \cdot \prod_{i=1}^L H_{\text{SOS}_i}(z)$$



Estructuras IIR

Descomposición en Biquad en Cascada

$$H(z) = g \cdot \prod_{i=1}^L H_{\text{SOS}_i}(z)$$



Descomposición en SOS:

- Polos y ceros son independientes
- Estructura con coeficientes menos sensible a la cuantificación

Estructuras IIR

Ejemplo: Descomposición en Biquad en Cascada con matlab

```
% Filtro IIR Butterworth  
N=10; Fp=0.3;  
[B,A]=butter(N,Fp);  
% Celdas 2° orden en cascada  
[sos, gain]=tf2sos(B,A,'up','inf')
```

$$H(z) = g \cdot \prod_{i=1}^L H_{\text{SOSi}}(z)$$

$$H_{\text{SOSi}}(z) = \left(\frac{b_{i0} + b_{i1} \cdot z^{-1} + b_{i2} \cdot z^{-2}}{1 + a_{i1} \cdot z^{-1} + a_{i2} \cdot z^{-2}} \right)$$

gain = 0.4583

	bi0	bi1	bi2	1	ai1	ai2	i
sos =	0.1197	0.2397	0.1197	1.0000	-0.6534	0.1117	1
	0.1256	0.2620	0.1368	1.0000	-0.6831	0.1622	2
	0.1484	0.3013	0.1533	1.0000	-0.7478	0.2722	3
	0.1596	0.3137	0.1546	1.0000	-0.8598	0.4628	4
	0.3039	0.5823	0.2792	1.0000	-1.0435	0.7753	5

Estructuras IIR

Ejemplo: Diseño utilizando fdatool (filterDesigner)

The screenshot displays the MATLAB Filter Designer (fdatool) window. The interface includes a menu bar (File, Edit, Analysis, Targets, View, Window, Help), a toolbar, and several panels. On the left, a vertical toolbar contains icons for different filter types: Lowpass, Highpass, Bandpass, Bandstop, Differentiator, IIR, and FIR. The main area is divided into several sections:

- Current Filter Information:** Shows the filter's structure and properties. A blue arrow points to this section with the label "Estructura y secciones". The information displayed is:
 - Structure: Direct-Form II, Second-Order Sections
 - Order: 10
 - Sections: 5
 - Stable: Yes
 - Source: Designed
- Magnitude Response (dB):** A plot showing the filter's magnitude response. The x-axis is "Normalized Frequency ($\times \pi$ rad/sample)" ranging from 0 to 1.0. The y-axis is "Magnitude (dB)" ranging from 0 to -250. A blue curve labeled "Lowpass Butterworth" shows the response, which is flat at 0 dB until approximately 0.3, then rolls off. A blue arrow points to this plot with the label "Cuantificar".
- Response Type:** A section with radio buttons for Lowpass, Highpass, Bandpass, Bandstop, and Differentiator. The "Lowpass" button is selected.
- Filter Order:** A section with radio buttons for "Specify order" and "Minimum order". The "Specify order" button is selected, and the order is set to 10.
- Frequency Specifications:** A section with input fields for "Units" (set to "Normalized (0 to 1)"), "Fs" (set to 48000), and "wc" (set to 0.3).
- Magnitude Specifications:** A section with a text box stating "The attenuation at cutoff frequencies is fixed at 3 dB (half the passband power)".
- Design Method:** A section with radio buttons for "IIR" and "FIR". The "IIR" button is selected, and the "Butterworth" method is chosen.
- Options:** A section with a text box stating "There are no optional parameters for this design method."
- Design Filter:** A button at the bottom right of the main area.

At the bottom of the window, the status bar shows "Ready".

Additional annotations on the left side of the image include:

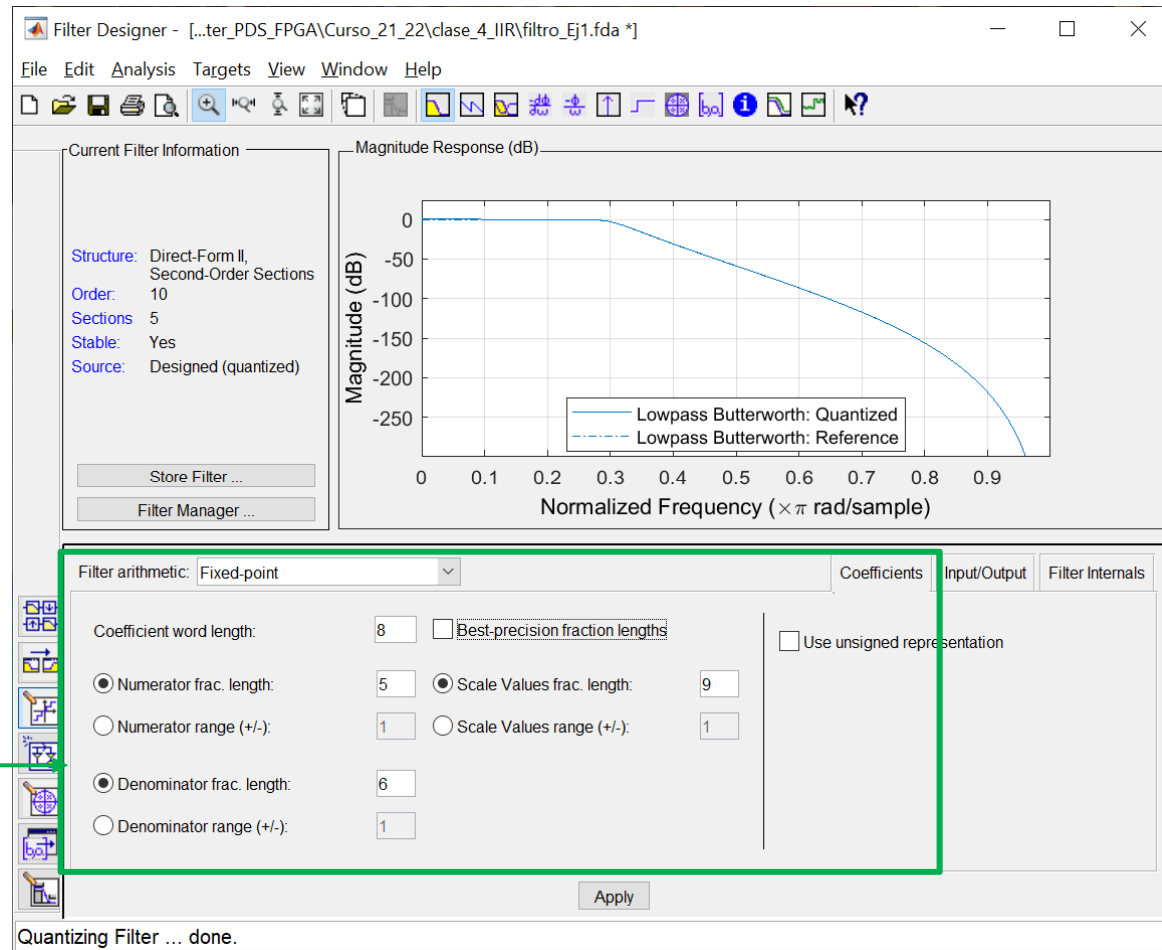
- "Crear modelo Simulink" with an arrow pointing to the Simulink icon in the left toolbar.
- "Diagrama de polos y ceros" with an arrow pointing to the pole/zero plot icon in the left toolbar.

Estructuras IIR

Ejemplo: Diseño utilizando fdatool (filterDesigner)

Permite variar la cuantificación de los coeficientes a la vez que observamos como varía la respuesta del filtro

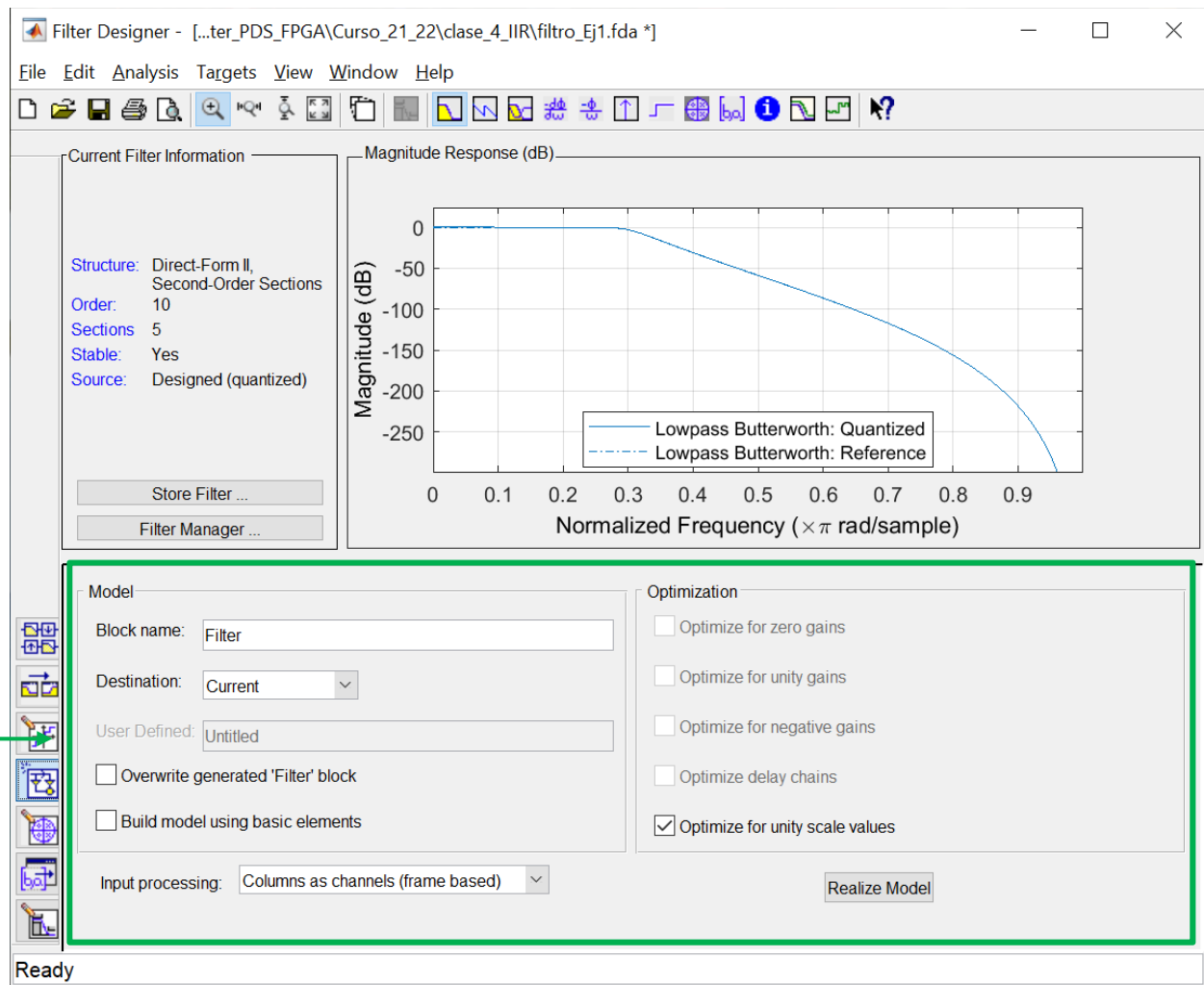
Estudio de la cuantificación de los coeficientes



Estructuras IIR

Ejemplo: Diseño utilizando fdatool (filterDesigner)

Permite crear un modelo simulink

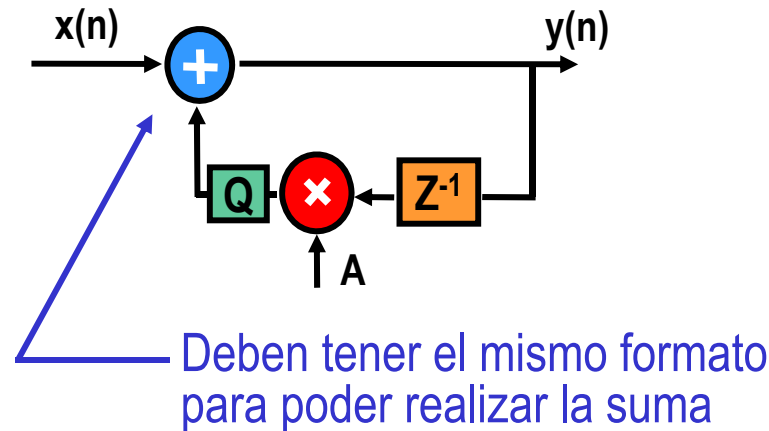


Filtros IIR: Cuantificación

Ejemplo: Integrador

$$y(n) = x(n) + Ay(n-1)$$

$$\begin{aligned} X &= X_e 2^{-x} \text{ (} N_x \text{ bits)} \\ Y &= Y_e 2^{-x} \text{ (} N_x \text{ bits)} \\ A &= A_e 2^{-a} \text{ (} N_a \text{ bits)} \\ P &= Y \cdot A \text{ (} N_x + N_a \text{ bits)} \end{aligned}$$



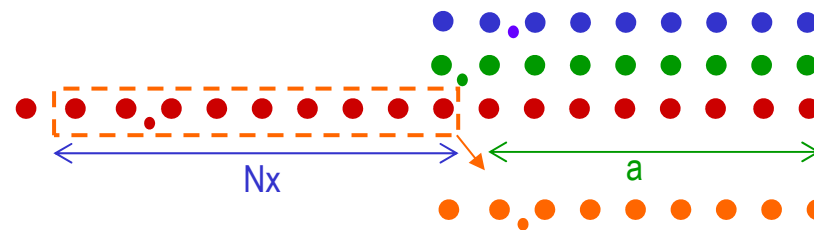
Ejemplo:

$$X \Rightarrow [9, 7]$$

$$A \Rightarrow [9, 8]$$

$$P = Y \cdot A = (Y_e \cdot A_e) 2^{-(x+a)} \Rightarrow [18, 15]$$

$$P_Q = Q(P \cdot 2^a) = P_e 2^{-x} \Rightarrow [9, 7]$$

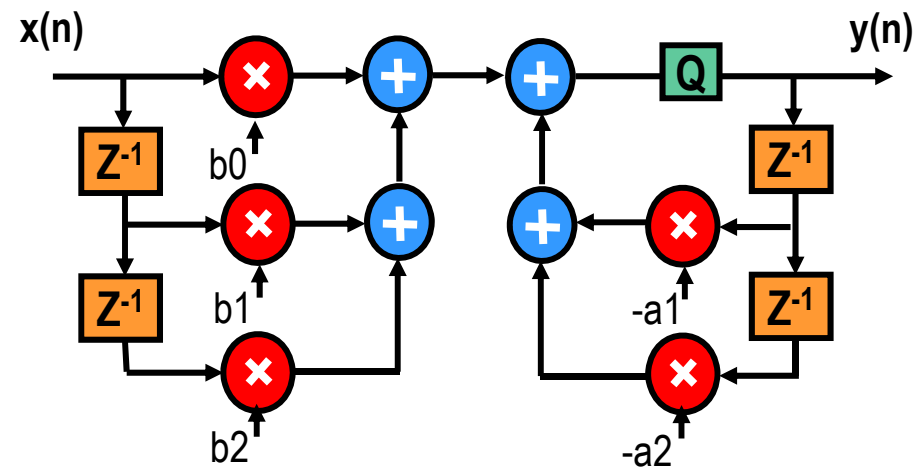


$$Y_e(n) = X_e(n) + P_{e_Q}(n-1)$$

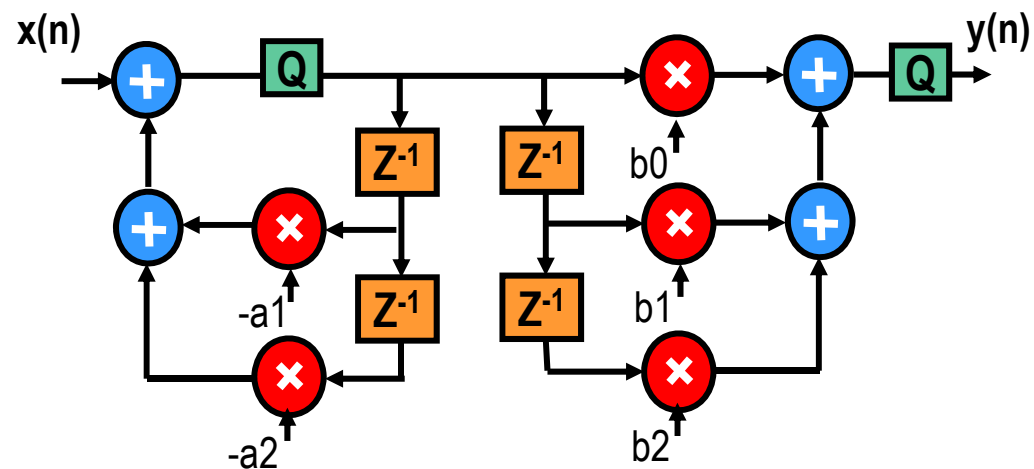
Filtros IIR: Cuantificación

Celda Biquad IIR: puntos de cuantificación

Directa I



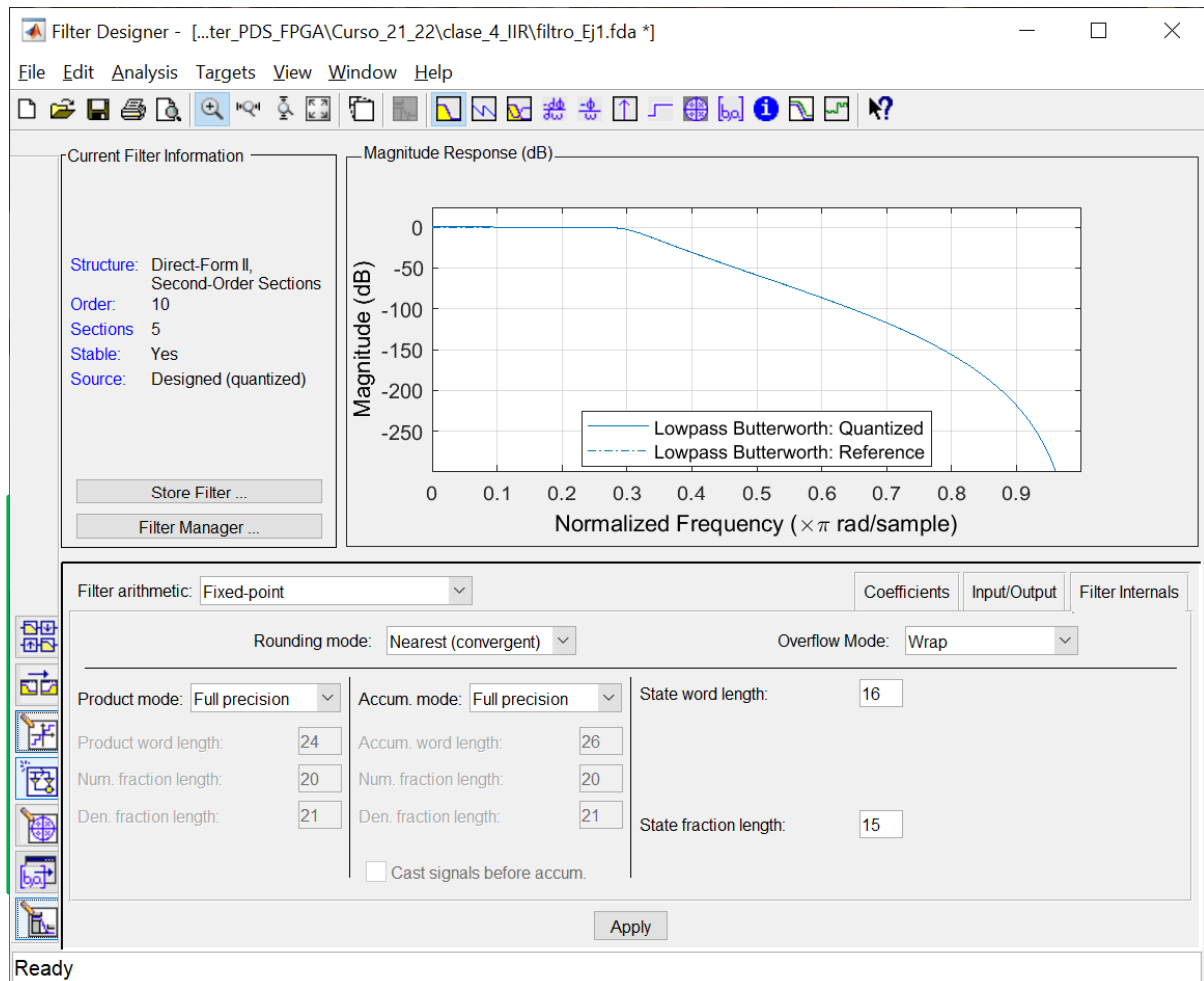
Directa II



Filtros IIR: Cuantificación

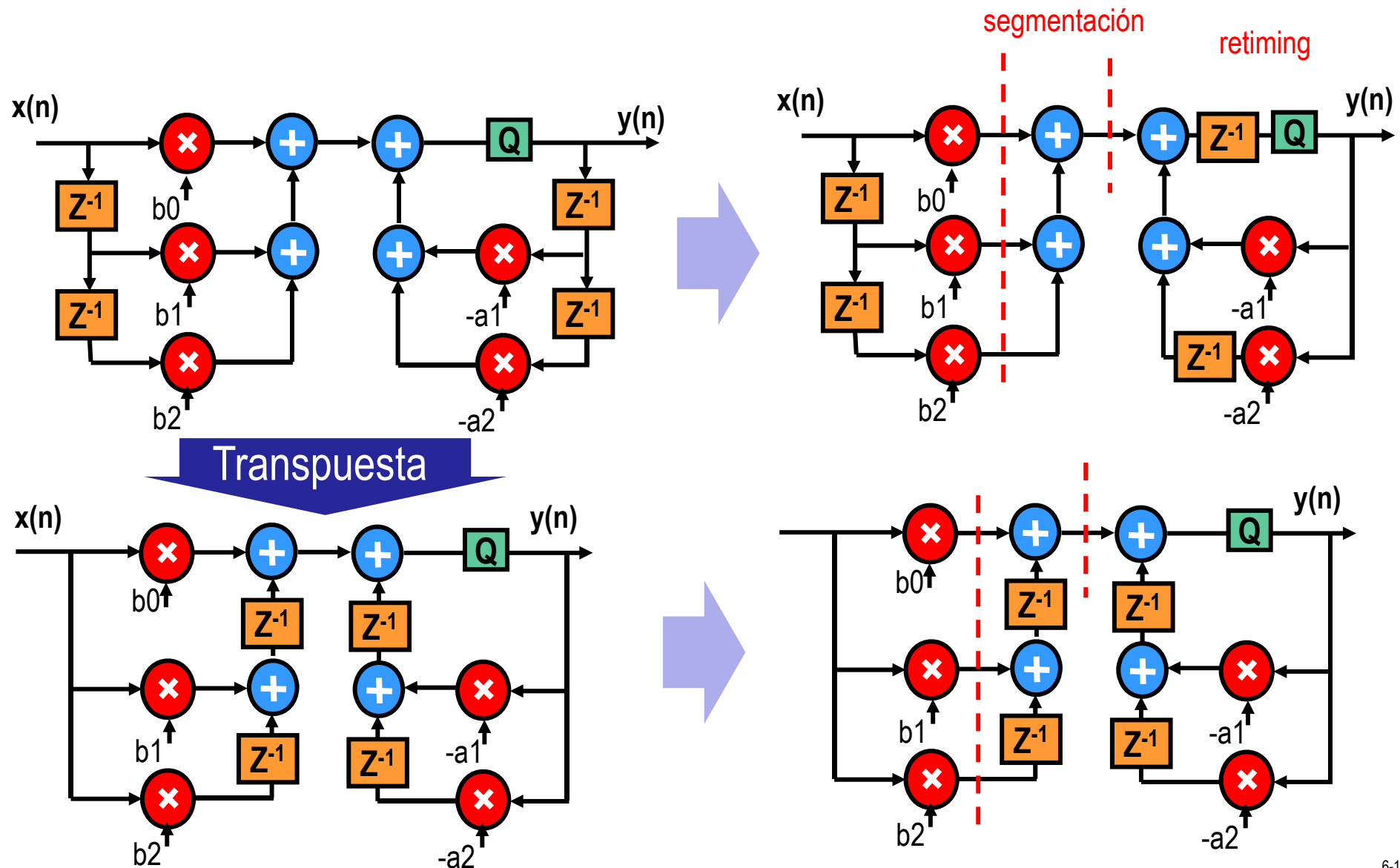
Cuantificación interna con FDATool

Permite revisar la cuantificación interna



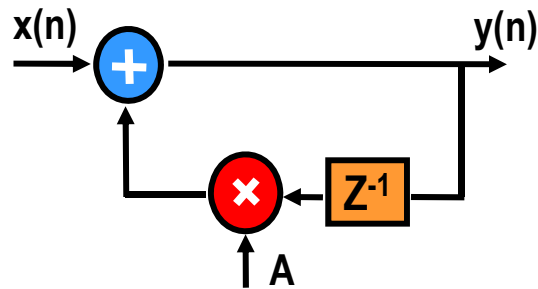
Filtros IIR: Segmentación

Implementación celda Biquad IIR con directa I

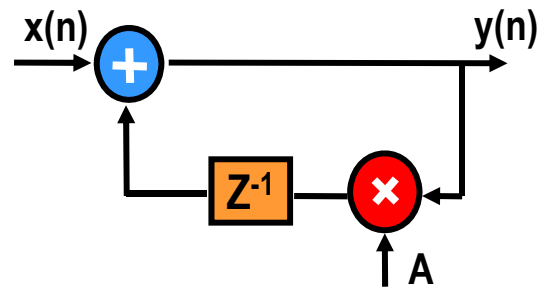


Segmentación de Filtros IIR

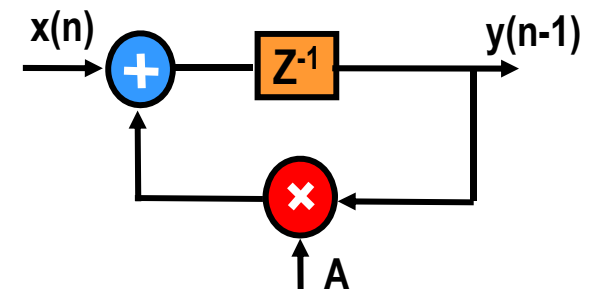
Retiming no mejora el camino crítico



$$T_c = t_{\text{mult}} + t_{\text{add}}$$



$$T_c = t_{\text{mult}} + t_{\text{add}}$$



$$T_c = t_{\text{mult}} + t_{\text{add}}$$

- Look-ahead Pipeline
- Pipeline-Interleaving

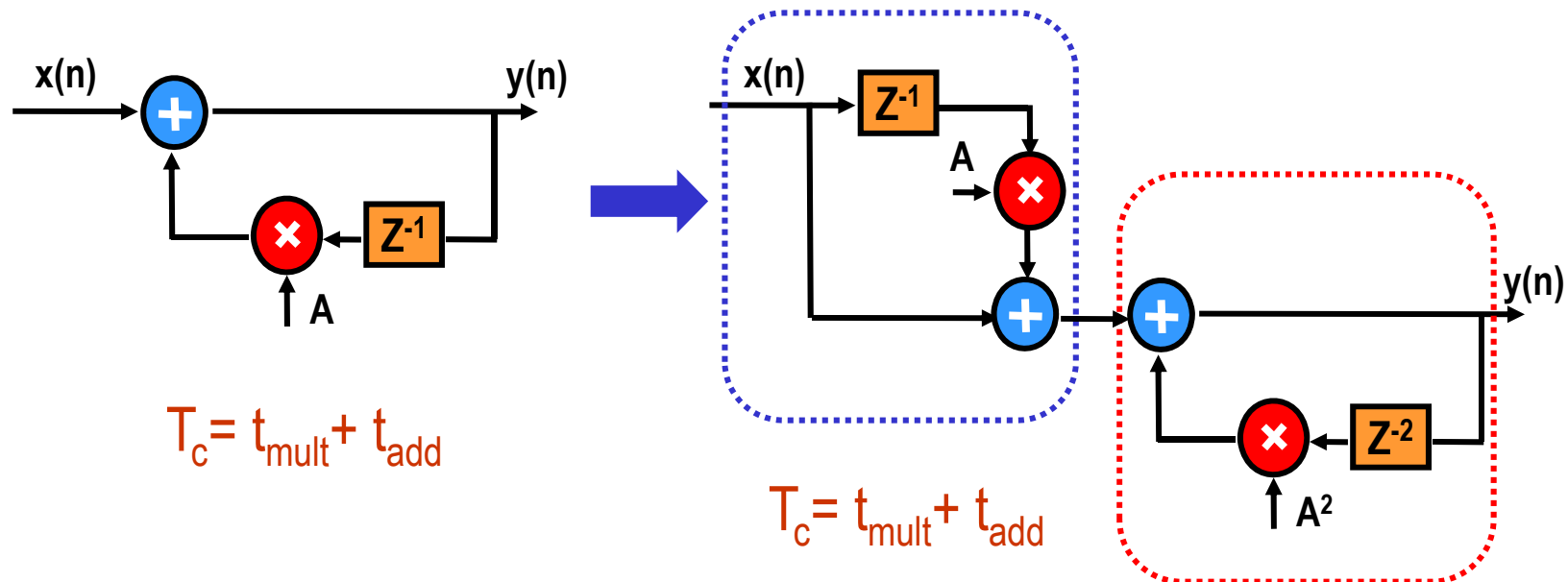
Look-ahead Pipeline

Ejemplo: Integrador

$$y(n) = x(n) + Ay(n-1)$$

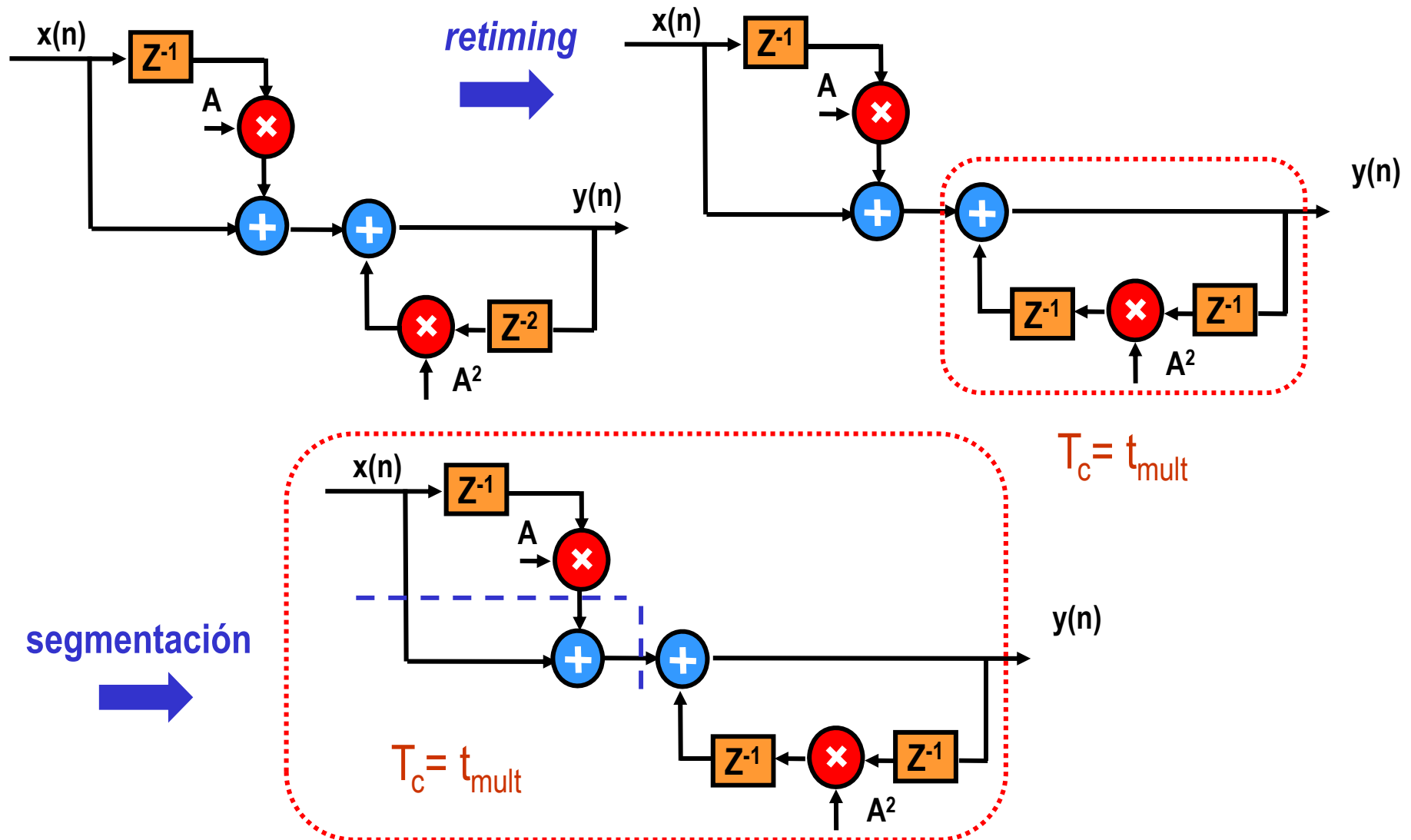
$$y(n+1) = x(n+1) + Ay(n) = x(n+1) + Ax(n) + A^2y(n-1)$$

$$y(n) = \underbrace{x(n) + Ax(n-1)}_{\text{Filtro FIR}} + \underbrace{A^2y(n-2)}_{\text{Recursividad de orden 2}}$$



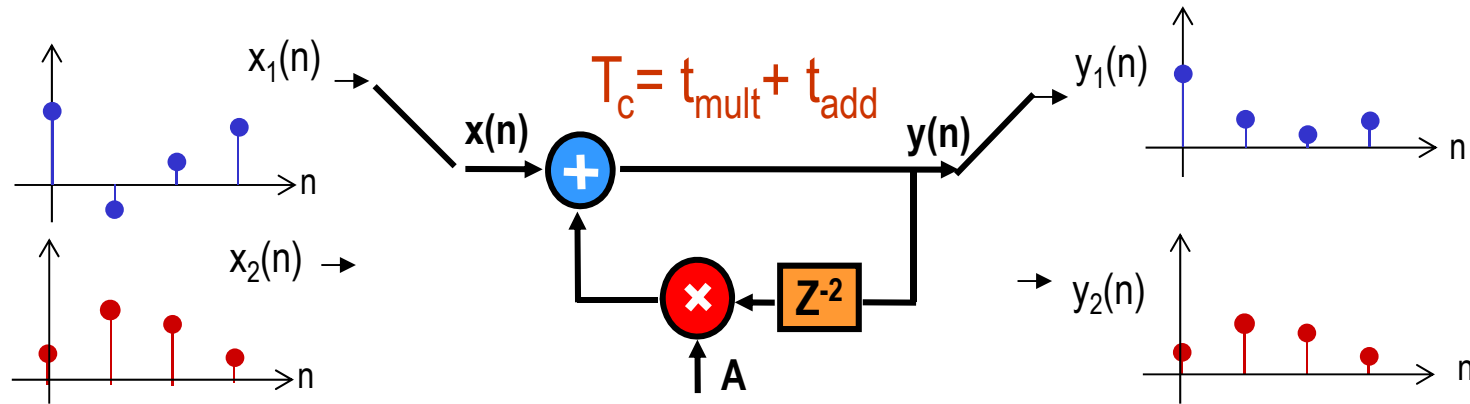
Look-ahead Pipeline

Ejemplo: Integrador

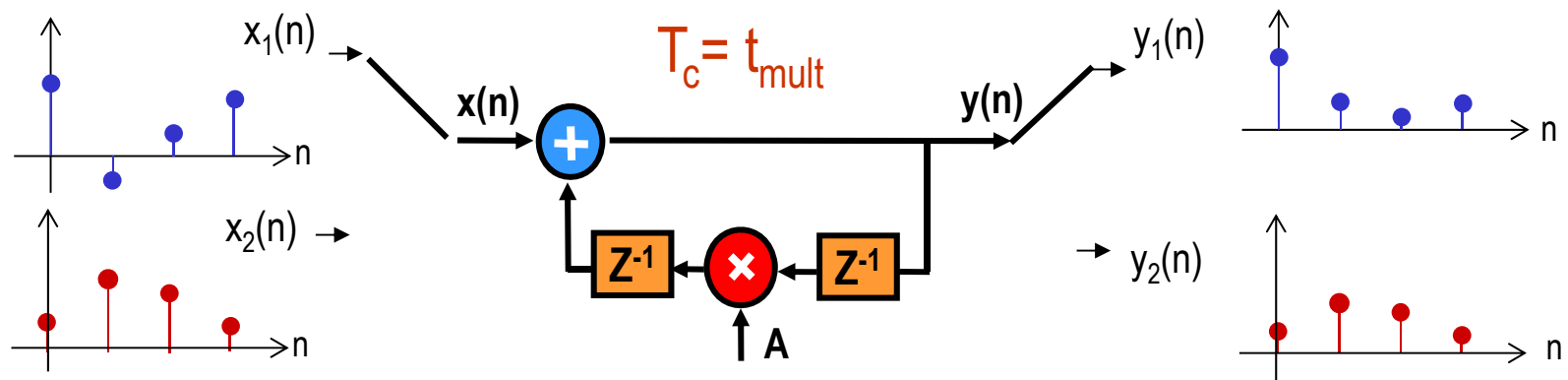


Interleaving+ retiming

- Se utiliza para procesar M señales con el mismo HW
- Se reemplaza cada retardo por M retardos



- Aplicando *retiming* aumenta la f_{max} y posibilita el procesamiento de tramas con mayor f_s

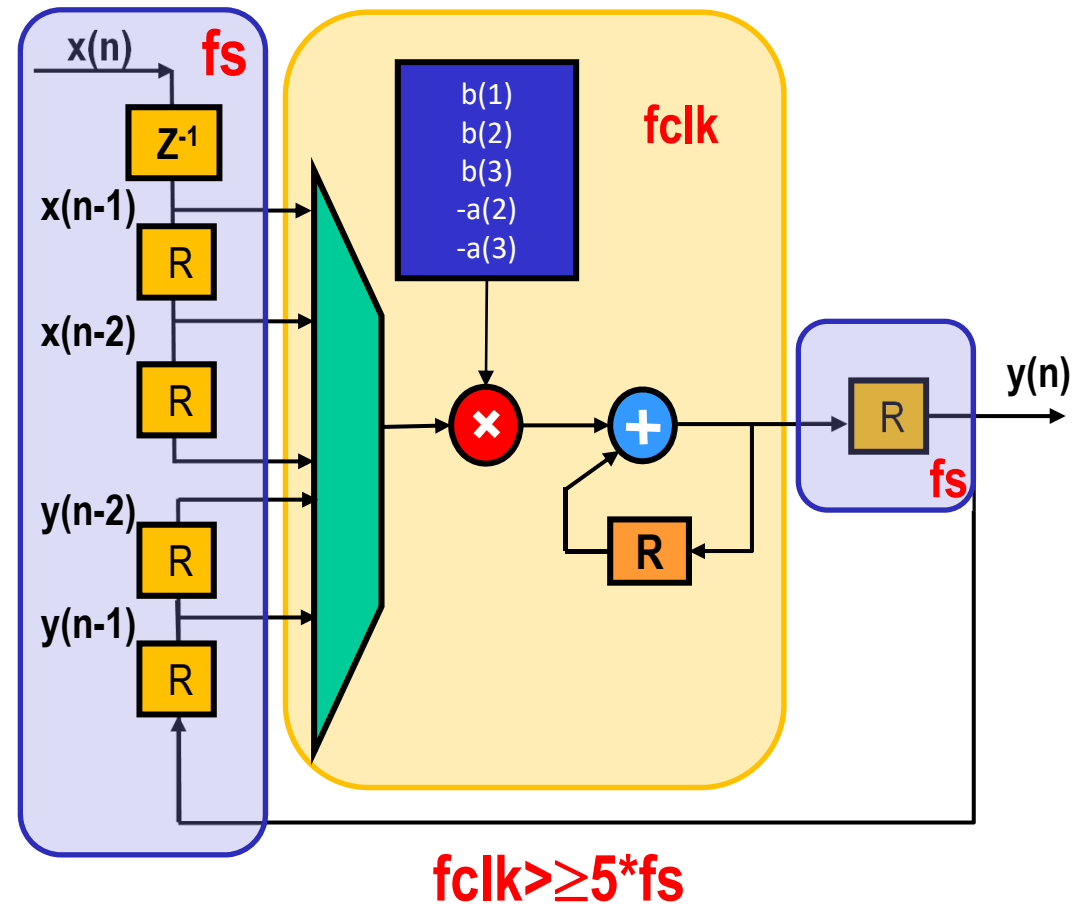


Filtros IIR: Arquitectura Secuencial

```
% Script de Matlab

acc(i)=0;
for j=1:3
    acc(i)=acc(i)+x(i-j+1)*b(j);
end
for j=1:2
    acc(i)=acc(i)-y(i-j)*a(j+1);
end

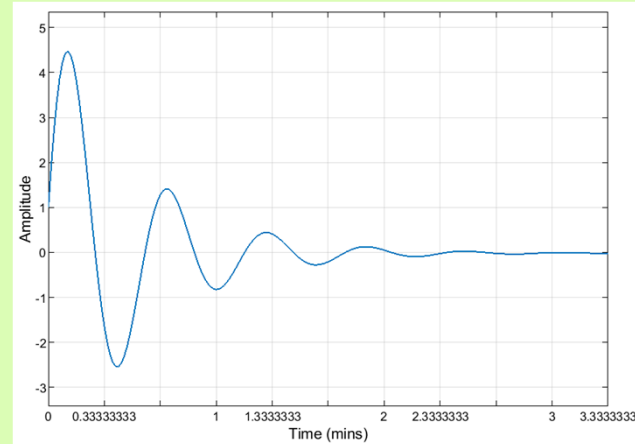
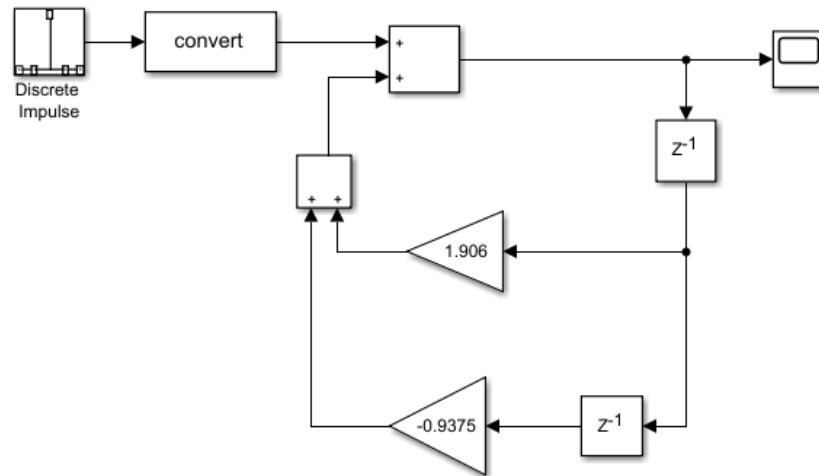
y(i)=acc(i);
```



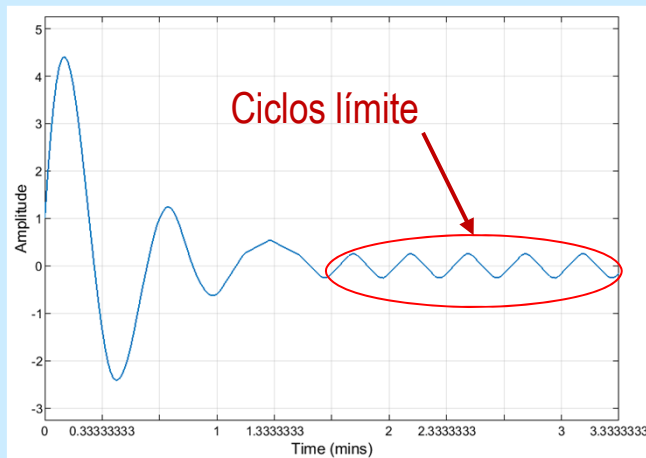
Ciclos Límite

- Oscilaciones que aparecen en los filtros IIR cuando la entrada se anula durante varias muestras
- Debidas al redondeo o truncado en las multiplicaciones o al overflow en las sumas
- Pueden eliminarse utilizando longitudes de palabra más largas
- Se ha estudiado el comportamiento de estas oscilaciones en filtros de 1^{er} y 2^o orden [Oppenheim]
- En estructuras de 2^o orden pueden eliminarse utilizando algunos esquemas de cuantificación complejos

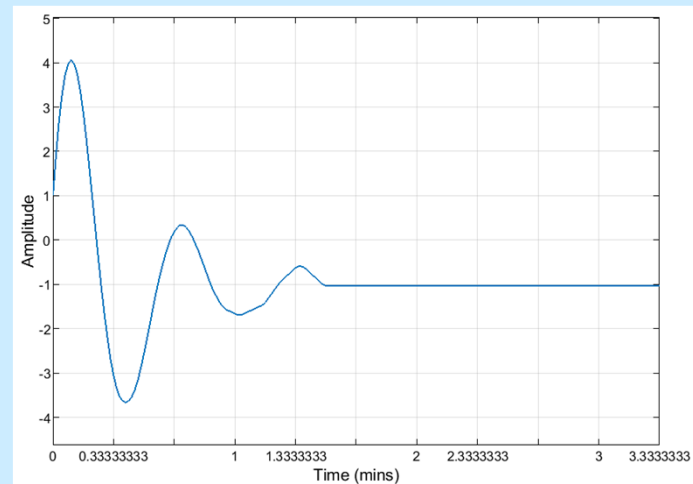
Ciclos Límite: Ejemplo



Cuantificación sfix(16,11), sumadores con saturación, multiplicadores con redondeo



Cuantificación sfix(10,5), sumadores con saturación, multiplicadores con redondeo

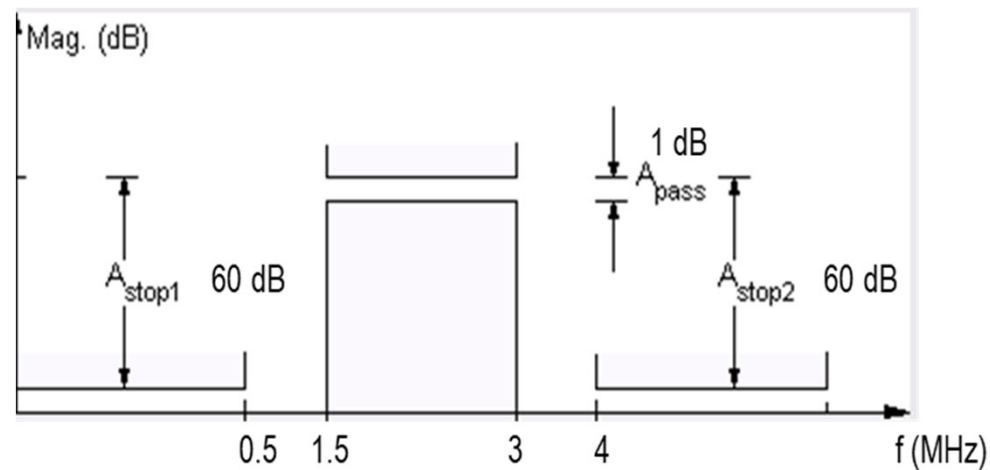


Cuantificación sfix(10,5), sumadores con overflow, multiplicadores con redondeo .

Ejercicio:

Se desea implementar un filtro **paso banda IIR elíptico de mínimo orden** que cumpla las especificaciones dadas por la siguiente plantilla, para una frecuencia de muestreo de 10 MHz, utilizando celdas de segundo orden (SOS). Se desea una relación señal a ruido mayor de 60 dB.

El formato de los datos de entrada es `sfix(16,15)`.



Paso 1: Diseñar con el FDATool y elegir la cuantificación de los coeficientes

Paso 2: Implementar en Simulink el modelo ideal y el modelo con secciones de segundo orden

Paso 3: Elegir la cuantificación de la ruta de datos y cuantificar los operadores intermedios

Paso 4: Simular y comprobar SNR

Bibliografía

Libros de procesamiento de señales

- A.V. Oppenheim, R.W. Schaffer, Discrete-Time Signal Processing

Libros de procesamiento digital

- U. Meyer-Baese, *Digital signal processing with Field Programmable Gate Arrays*, Springer, 2001 (Capítulo 4)

Help de Matlab