ECE 4270
Fundamentals of Digital Signal Processing

Lecture 15:

Digital Filter Structures & Coefficient Quantization

School of Electrical and Computer Engineering
Georgia Institute of Technology
Summer, 2004

Filter Structures

- To implement rational systems
- To achieve desired result when numerical precision is limited
  - Internal data representation with finite-length word
- To achieve maximum efficiency when computational resources are limited
  - Number of delay elements
  - Size of (internal) data buffer
  - Number of numerical operations (multiply-add)

Digital Filters

- General N^{th}-order difference equation:

$$y[n] = \sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k]$$

- System function:

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}} = A \frac{\prod_{k=1}^{M} (1 - c_k z^{-1})}{\prod_{k=1}^{N} (1 - d_k z^{-1})}$$

- There is a direct correspondence between the difference equation and the system function when the numerator and denominator are written as polynomials in z^{-1}.

Direct Form I Implementation

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}} = \underbrace{\left(\frac{1}{1 - \sum_{k=1}^{N} a_k z^{-k}} \right)}_{\text{poles}} \underbrace{\sum_{k=0}^{M} b_k z^{-k}}_{\text{zeros}}$$

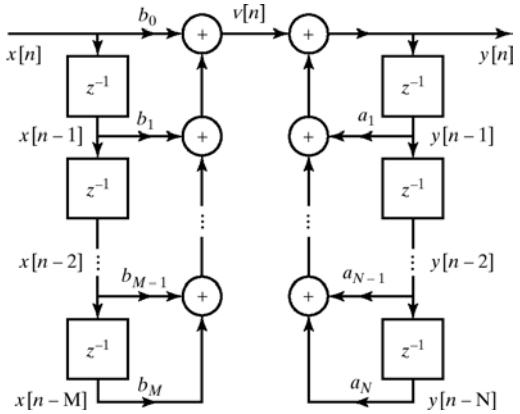
$$Y(z) = H(z)X(z)$$

$$v[n] = \sum_{k=0}^{M} b_k x[n-k] \quad \text{(zeros)}$$

$$y[n] = \sum_{k=1}^{N} a_k y[n-k] + v[n] \quad \text{(poles)}$$

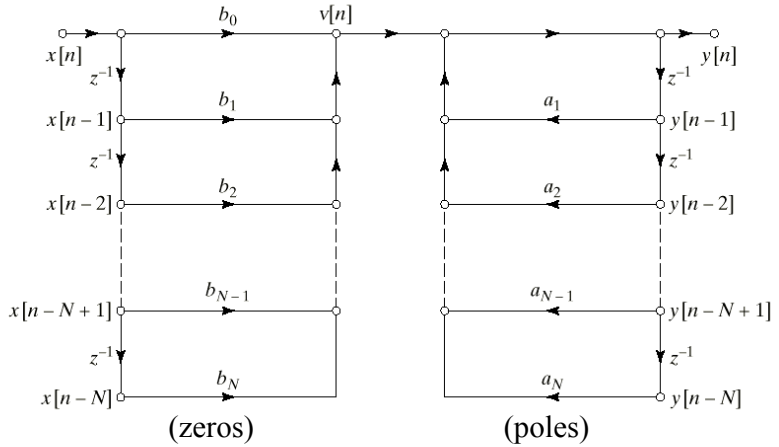
Block Diagram of IIR Direct Form I

$$v[n] = \sum_{k=1}^{M} b_k x[n-k] \qquad y[n] = \sum_{k=1}^{N} a_k y[n-k] + v[n]$$



Signal Flow Graph IIR Direct Form I

$$v[n] = \sum_{k=1}^{M} b_k x[n-k] \qquad y[n] = \sum_{k=1}^{N} a_k y[n-k] + v[n]$$



Direct Form II Implementation

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}} = \underbrace{\sum_{k=0}^{M} b_k z^{-k}}_{\text{zeros}} \underbrace{\left(\frac{1}{1 - \sum_{k=1}^{N} a_k z^{-k}} \right)}_{\text{poles}}$$

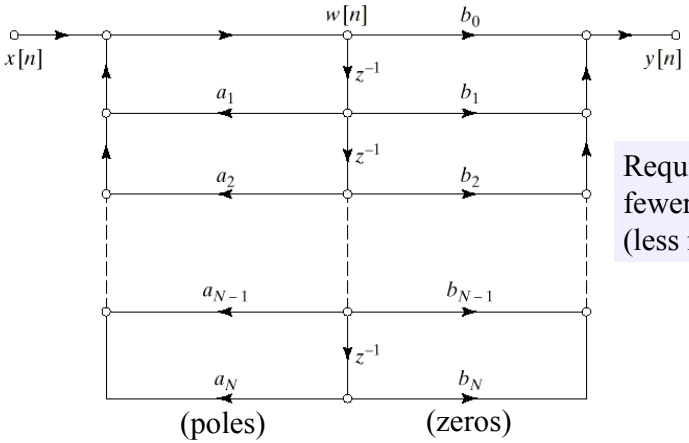
$$Y(z) = H(z)X(z)$$

$$w[n] = \sum_{k=1}^{N} a_k w[n-k] + x[n] \quad \text{(poles)}$$

$$y[n] = \sum_{k=0}^{M} b_k w[n-k] \quad \text{(zeros)}$$

IIR Direct Form II

$$w[n] = \sum_{k=1}^{N} a_k w[n-k] + x[n] \qquad y[n] = \sum_{k=1}^{M} b_k w[n-k]$$



Requires fewer delays (less memory)

Cascade Form

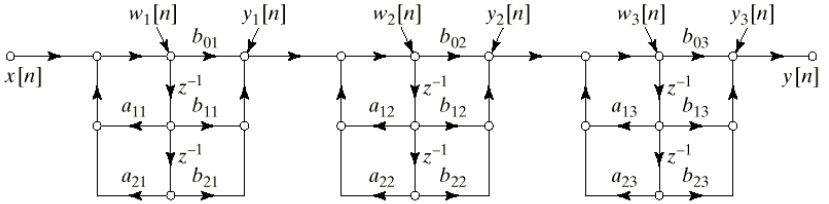
- Express $H(z)$ in terms of poles and zeros

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}} = A \frac{\prod_{k=1}^{M} (1 - c_k z^{-1})}{\prod_{k=1}^{N} (1 - d_k z^{-1})}$$

- Group poles and zeros as second-order factors

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

IIR Cascade Form



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

$$y_0[n] = x[n],$$

$$w_k[n] = a_{1k} w_k[n-1] + a_{2k} w_k[n-2] + y_{k-1}[n], \quad k = 1, 2, \ldots, N_s,$$

$$y_k[n] = b_{0k} w_k[n] + b_{1k} w_k[n-1] + b_{2k} w_k[n-2], \quad k = 1, 2, \ldots, N_s,$$

$$y[n] = y_{N_s}[n].$$

Parallel Form

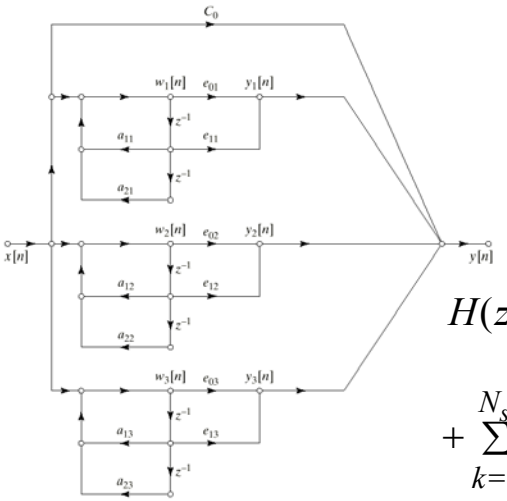
- Make a partial fraction expansion of H(z)

$$H(z) = \sum_{k=0}^{N_p} C_k z^{-k} + \sum_{k=1}^{N_1} \frac{B_k}{1 - c_k z^{-1}} + \sum_{k=1}^{N_2} \left(\frac{A_k}{1 - d_k z^{-1}} + \frac{A_k^*}{1 - d_k^* z^{-1}} \right)$$

- Group terms in second order factors

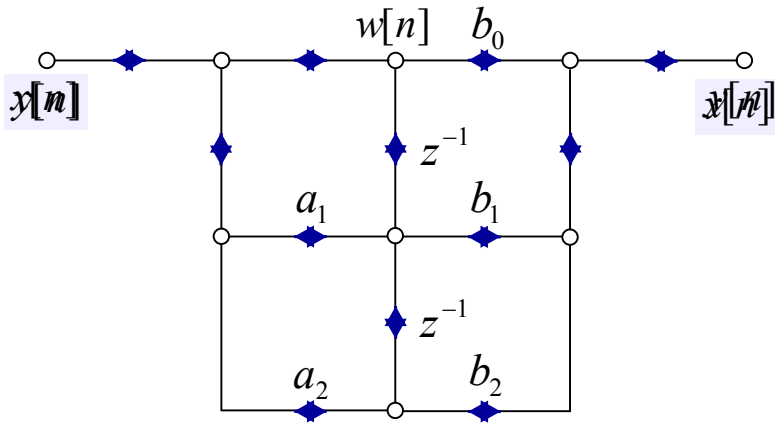
$$H(z) = \sum_{k=0}^{N_p} C_k z^{-k} + \sum_{k=1}^{N_s} \frac{e_{0k} + e_{1k} z^{-1}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}}$$

IIR Parallel Form

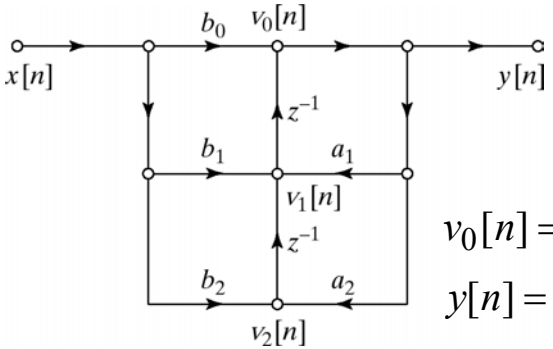


$$H(z) = \sum_{k=0}^{N_p} C_k z^{-k} + \sum_{k=1}^{N_s} \frac{e_{0k} + e_{1k} z^{-1}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}}$$

Transposed Form



Transposed Form Difference Equations



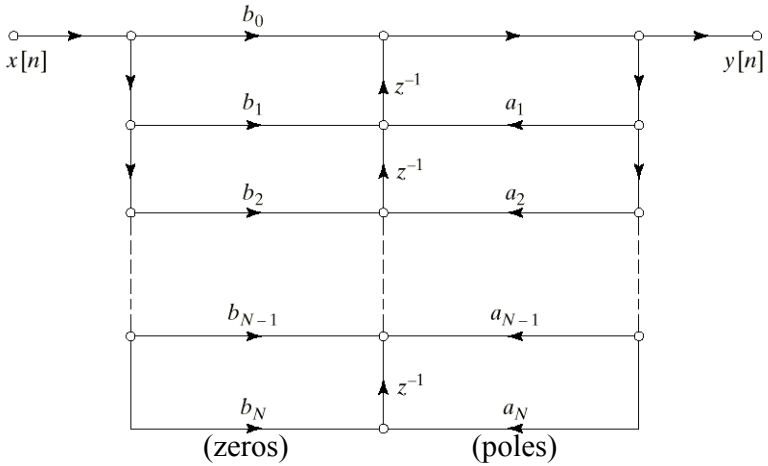
$$v_0[n] = b_0 x[n] + v_1[n-1]$$

$$y[n] = v_0[n]$$

$$v_1[n] = b_1 x[n] + v_2[n-1] + a_1 v_0[n]$$

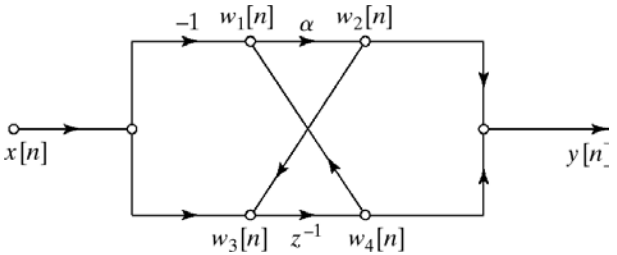
$$v_2[n] = b_2 x[n] + a_2 v_0[n]$$

Transposed IIR



Used in MATLAB's `filter( )` function

Finding a System Function



$$w_1[n] = w_4[n] - x[n] \Leftrightarrow W_1(z) = W_4(z) - X(z)$$

$$w_2[n] = \alpha w_1[n] \Leftrightarrow W_2(z) = \alpha W_1(z)$$

$$w_4[n] = w_3[n-1] \Leftrightarrow W_4(z) = z^{-1} W_3(z)$$

$$w_3[n] = w_2[n] + x[n] \Leftrightarrow W_3(z) = W_2(z) + X(z)$$

$$y[n] = w_2[n] + w_4[n] \Leftrightarrow Y(z) = W_2(z) + W_4(z)$$

Finding a System Function

$$W_1(z) = W_4(z) - X(z)$$

$$W_2(z) = \alpha W_1(z) = \alpha W_4(z) - \alpha X(z)$$

$$W_3(z) = W_2(z) + X(z)$$

$$W_4(z) = z^{-1} W_3(z) = z^{-1} W_2(z) + z^{-1} X(z)$$

$$W_4(z) = \alpha z^{-1} W_4(z) - \alpha z^{-1} X(z) + z^{-1} X(z)$$

$$W_4(z) = \frac{(1-\alpha) z^{-1}}{1 - \alpha z^{-1}} X(z)$$

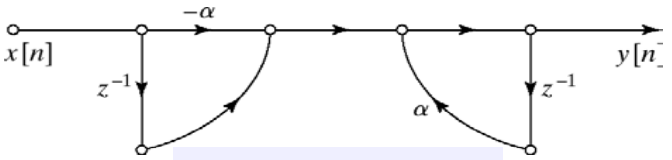
Solve for the System Function

$$W_2(z) = \frac{\alpha(z^{-1} - 1)}{1 - \alpha z^{-1}} X(z) \qquad W_4(z) = \frac{z^{-1}(1-\alpha)}{1 - \alpha z^{-1}} X(z)$$

$$Y(z) = W_2(z) + W_4(z) = \left(\frac{z^{-1} - \alpha}{1 - \alpha z^{-1}} \right) X(z)$$

$$\Rightarrow H(z) = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}$$

Allpass system



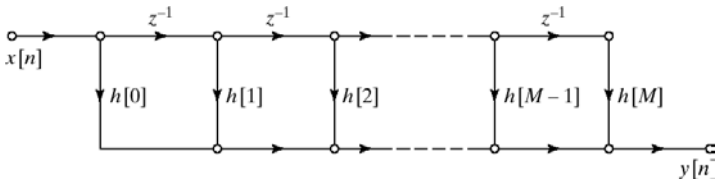
Equivalent Direct Form I

FIR Systems

- FIR system functions have only zeros

$$H(z) = \sum_{k=0}^{M} b_k z^{-k} = \sum_{k=0}^{M} h[k] z^{-k}$$

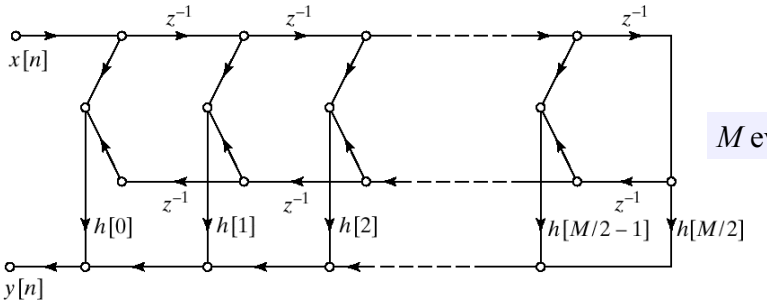
- In this case, direct form I is just convolution.



$$y[n] = (((h[0]x[n] + h[1]x[n-1]) + h[2]x[n-2]) + \ldots)$$

Multiply-accumulate (MAC)

FIR Linear Phase Systems



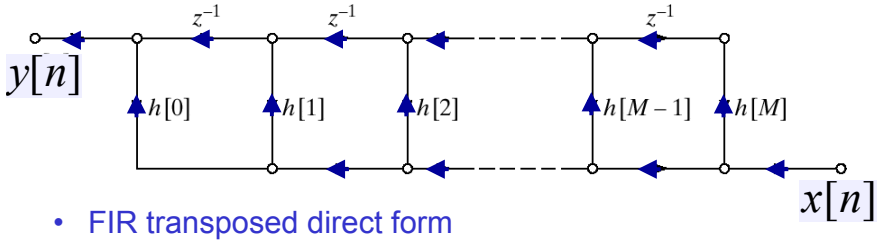
M even

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

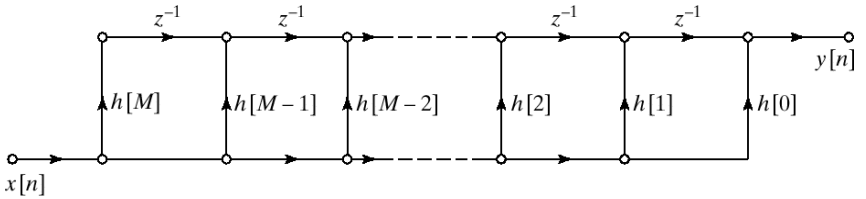
$$y[n] = h[0](x[n] + x[M]) + h[1](x[n-1] + x[M-1]) + \ldots$$

FIR Transposed Direct Form

- FIR direct form



- FIR transposed direct form

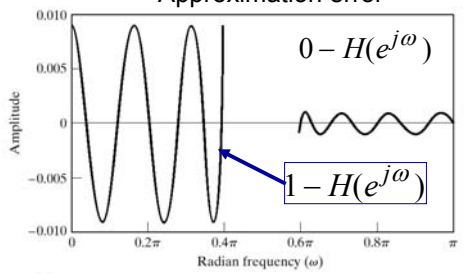


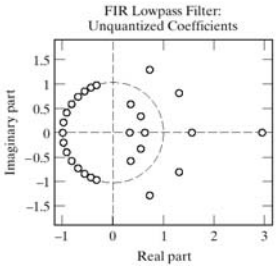
Fixed-Point Implementation Issues

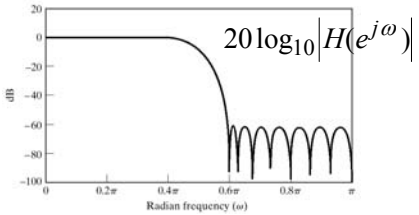
- We need to represent coefficient and signal values by integers in a fixed range.
- Quantization errors in coefficients imply shifts of poles and zeros (even instability).
- For a given word-length, the quantization error is fixed in size. Therefore, signal values should be maintained as large as possible to maximize SNR.
- If signal values get too large, additions can overflow (or clip), thereby creating large errors.
- Thus, fixed-point implementations require careful attention to scaling the signal values.

Unquantized FIR Filter Response

Approximation error

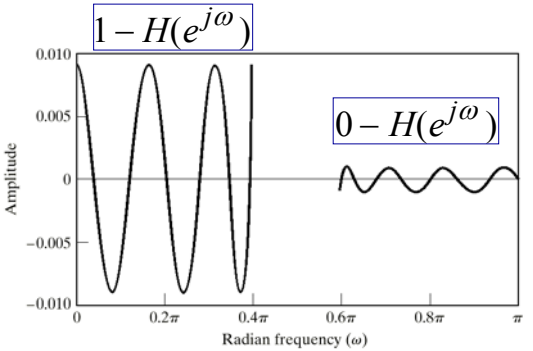


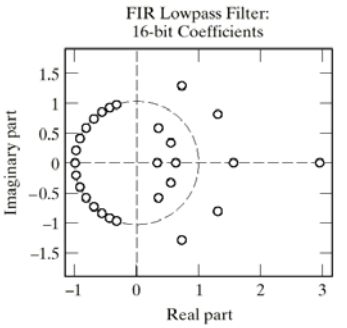




Note that zeros are in conjugate reciprocal groups (1,2, or 4). This is a basic property of FIR linear phase filters.

16-Bit Quantization of FIR Filter



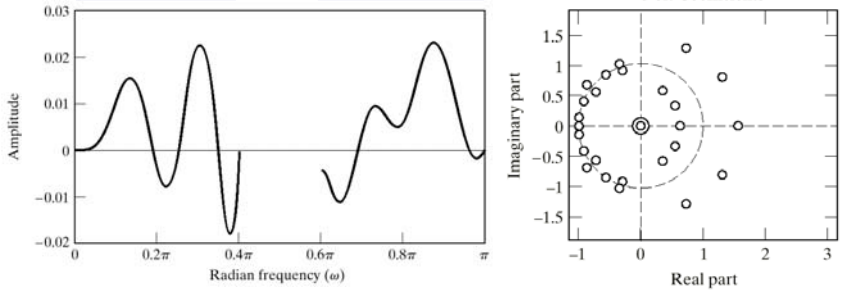


We preserve the linear phase property since

$$\hat{h}[M-n] = \pm\hat{h}[n]$$

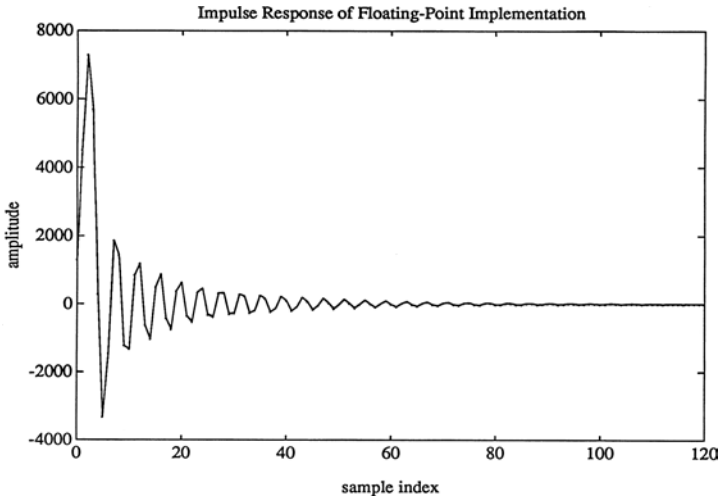
8-Bit Quantization of FIR Filter

$1 - H(e^{j\omega})$ $0 - H(e^{j\omega})$

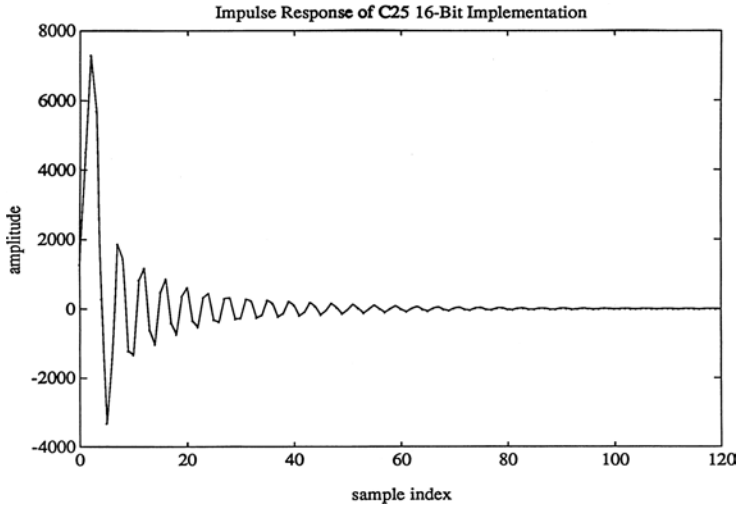


Zeros stay in reciprocal quads, but shift significantly due to the quantization.

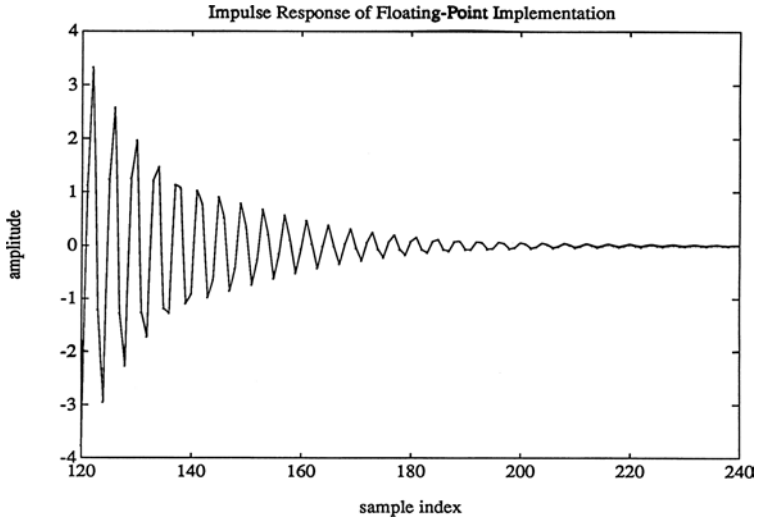
Unquantized IIR Implementation - I



Quantized IIR Implementation - I



Unquantized IIR Implementation - II



Quantized Implementation - II