EEE5140Z SDR:
Software Defined Radio

Digital Filtering

Dr. Yaseen Zaidi
yaseen.zaidi@ieee.org
July 11, 2018

Outline

- Fundamentals of digital filtering
  - Single dimension digital and discrete time signal processing
  - From differential to discrete equations - z-transform
  - Filter specification: magnitude response in passband and sideband ripples
  - Impulse response, cutoff frequency and order of filter
  - IIR and FIR filters, stability criteria, standard structures of filters

Resources

- Digital Design of Signal Processing Systems, S. Khan, http://www.drshoabkhan.com/Downloads.html
- https://au.mathworks.com/campaigns/offers/download-rtl-sdr-ebook.html

Caveats

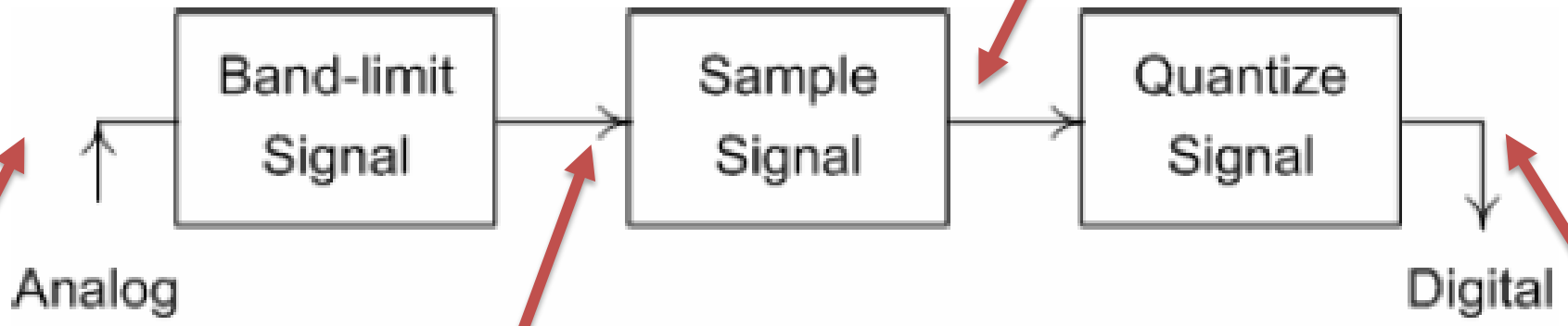
- In communication theory, information theory, estimation and detection theory, coding theory, DSP theory or digital control theory, the term "filtering" refers to any complex signal processing / mathematical operation on the signal that changes the its characteristics and may require lot of computation and algorithmic processing
- E.g., Wiener filter, Kalman filter, Least Square method, filtering in observability, estimatability and controllability
- Our context is classical i.e., "removal / altering"

Filtering Need

Signal Conversion

- The analog signal must be converted to digital equivalent before a digital filter acts on it

ADC noise: SINAD, SFDR, ENOB, THD, etc.



Analog bandpass filtering

RF or HF analog filtering

Digital filtering

Filtering Need

- Standard AWGN and ADC conversion not the only reason
- Each block in the signal path induces undesired variations in the signal content which may be removed before the signal entry into the next block
- Some reasons for in SDR signal chain:
  - Compacting the spectrum by preshaping of pulses (root raised cosine, gaussian)
  - Filtering of out-of-band frequency content / leaked spectrum induced by nonlinear blocks, supressing sidelobe amplitudes / RFI
  - Reducing (decimation) and increasing (interpolation) data rate matching
  - Attenuating distortion
  - Up/down-converting of signal frequency / mixing
  - Optimum detection (matched filter for optimum SNR)

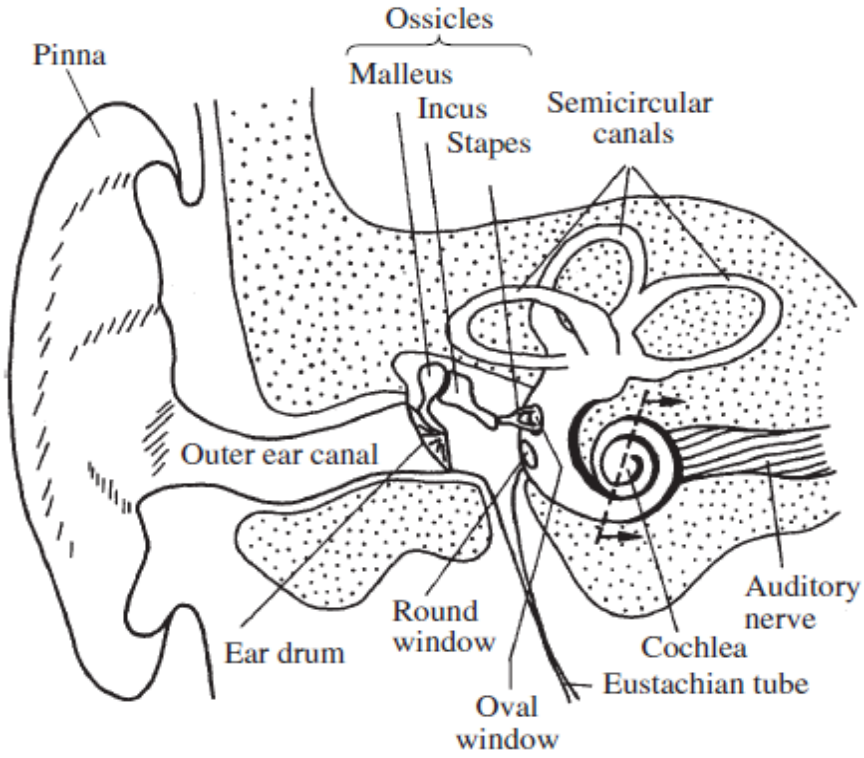
Filtering Need

- Many of these needs may be fulfilled by implementing Finite Impulse Response (FIR) or Infinite Impulse Response (IIR) filters which are two fundamental and atomic schemes in the DSP
- These schemes are generally integrated with other DSP operations to do the full filtering task

Why Filtering Digitally

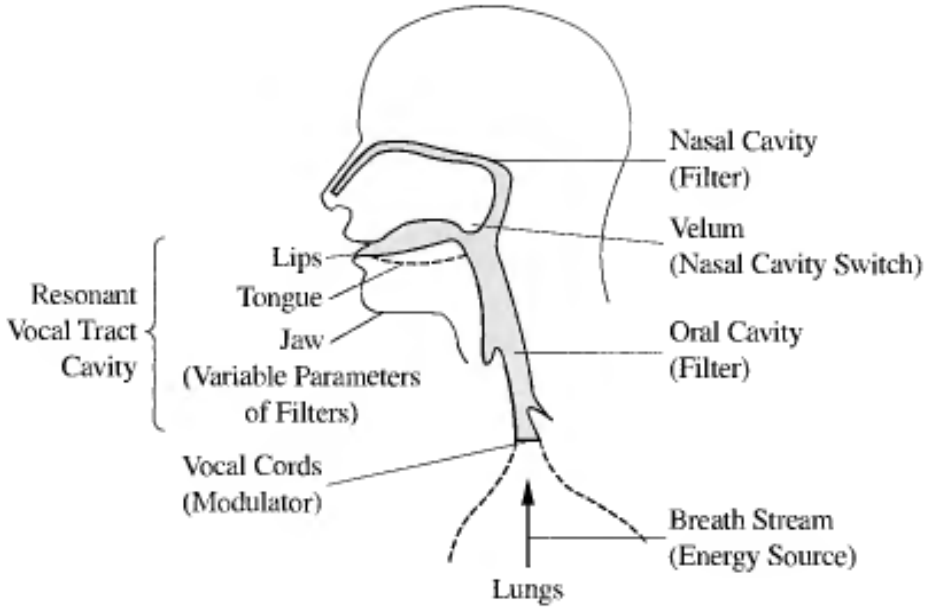
- The benefit of a digital filter is it does not require new discrete components that may change the topology of the filter network when need changes
- A digital filter is programmable e.g., analog subscriber line distortions are compensated with digital filters in Intel Sicofi codec chips
- It is worthwhile to note that after the digital backed/baseband the rest of the filtering in high frequency analog or RF section is implemented with discrete microwave components and considered “black magic”

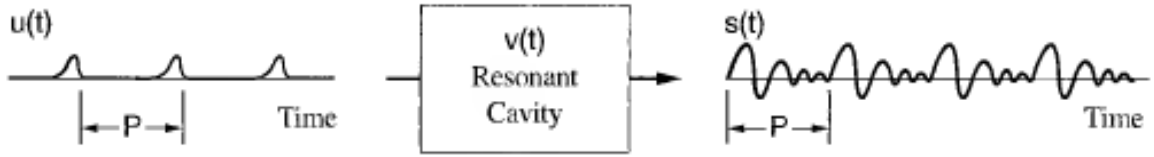
Caveat: Biomechanical Filters and Signal Sources



Credit: Digital Speech Processing, Vary/Martin

Credit: Discrete Time Speech Processing, Quatieri





Caveat

- Digital filter i.e., man made is not the only superior device of filtering
- Some of the greatest and most complex filters fitted in physiological systems are made of biological membranes, cavities and bones, and are mechanical in nature
- E.g., the human ear drum can detect the faintest of frequencies and eye samples wide range of visible band
- Eye performs multidimensional signal processing, time and phase

Digital filtering not always workable

- Digital filters do not power transfer, a basic requirement in RF signal path
- Matching the impedance for maximum power transfer which occurs at when the load and the source impedance are complex conjugates $(a + jb)(a - jb)$ of each other

$$Z_s = Z_L^*$$

- These filters are passive e.g., power matching networks

Digital Filter + and –

- Advantages

- Reproducible response
- Insensitive to temperature
- Programmable
- No parasitic effects

Disadvantages

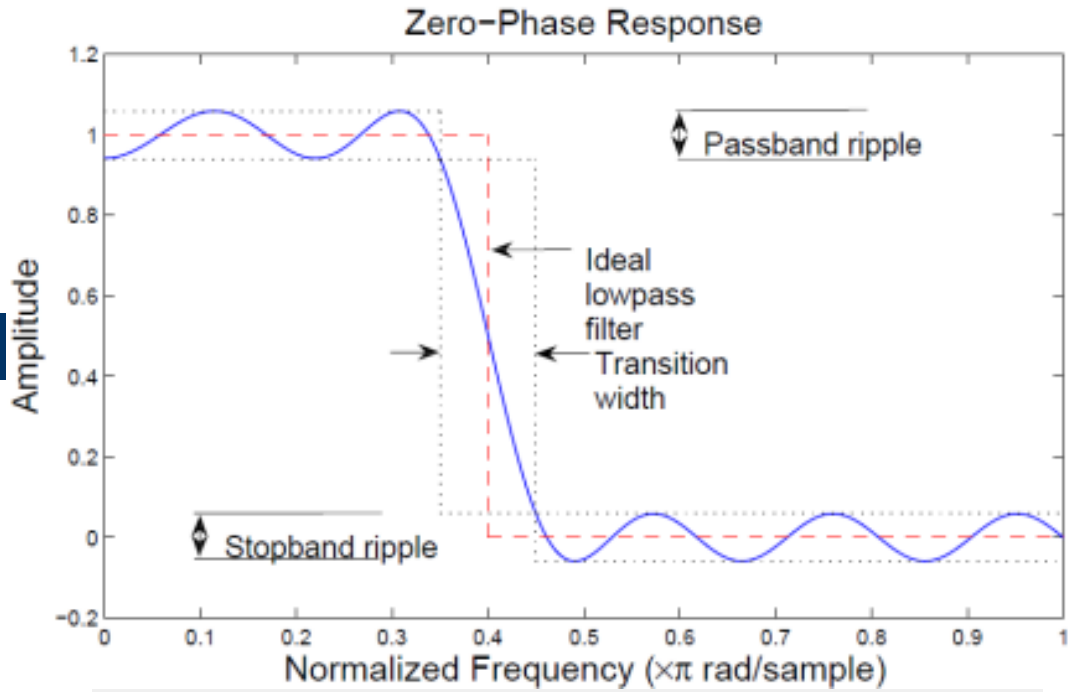
- Can’t work for power transfer / impedance match as no storage elements
- Clock frequency can be limiting
- Large filters require computational power, device area and power
- Need ADC
- Sampling errors of ADC

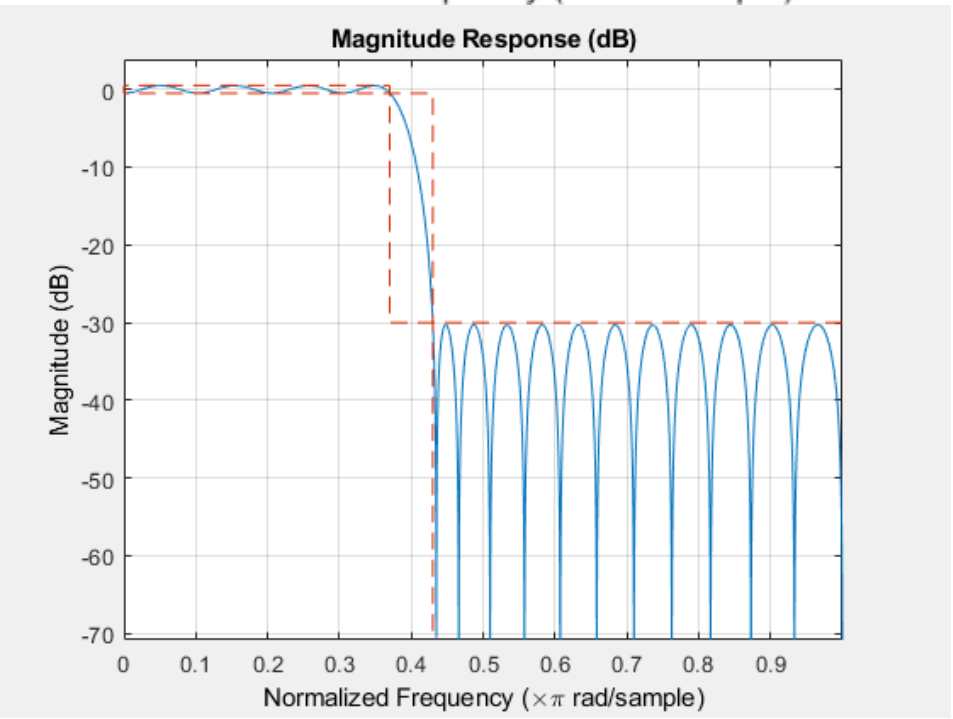
Frequency Selectivity

- A major of any filter is to only process / reject the band of interest which classify them as:
  - Lowpass
  - Highpass
  - Bandpass
  - Bandstop (notch)
- Based on the band requirement, filter specification is set

Filter Specification

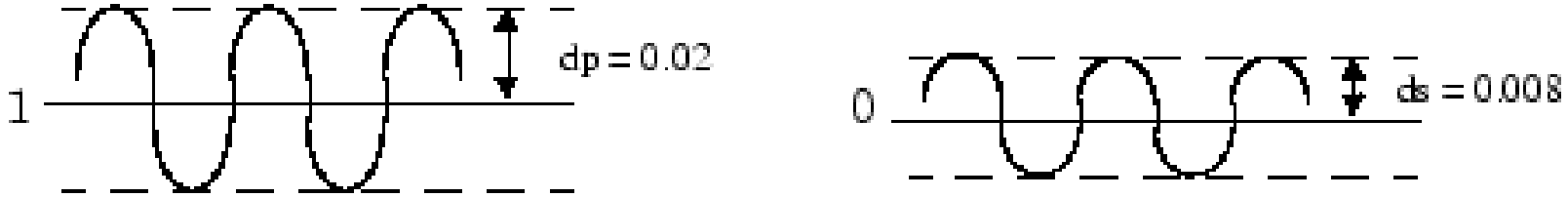
- Passband frequency
- Stopband frequency
- Passband attenuation
- Stopband attenuation
- Cutoff frequency / transition width
- Order
- Magnitude /
- Frequency response
- Phase response





Filter Specification

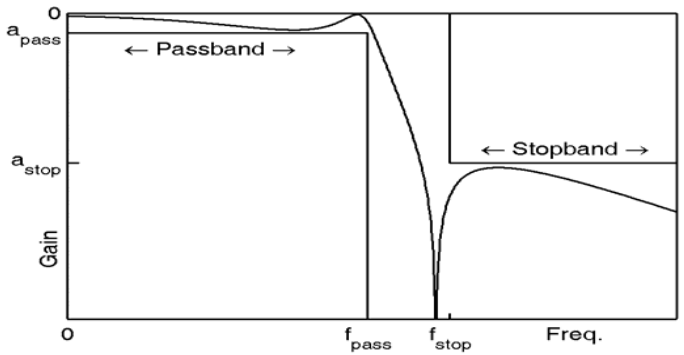
- Passband and stopband ripples



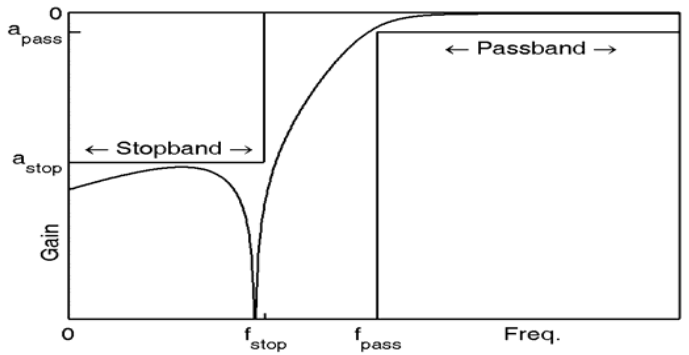
- Order gives the number of coefficients of the filter or taps
- Higher order means better frequency response, that means you have more terms to add to get to an ideal brickwall / rectangular response of a lowpass

Frequency Selectivity

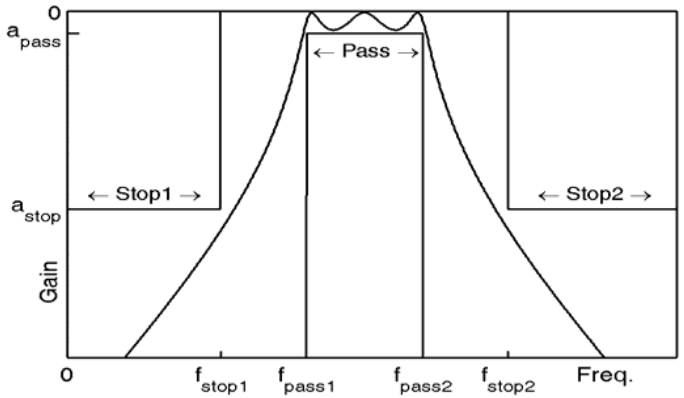
- Lowpass



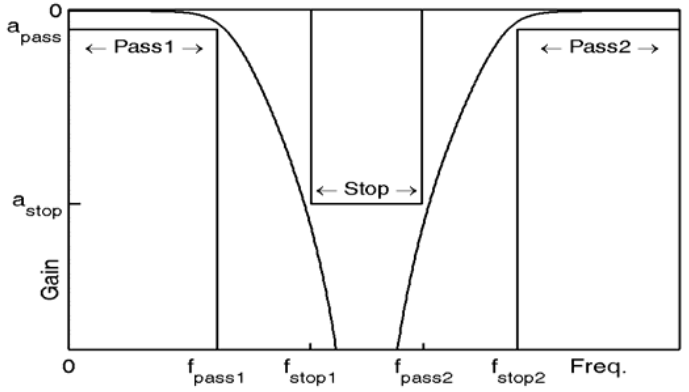
Highpass



- Bandpass

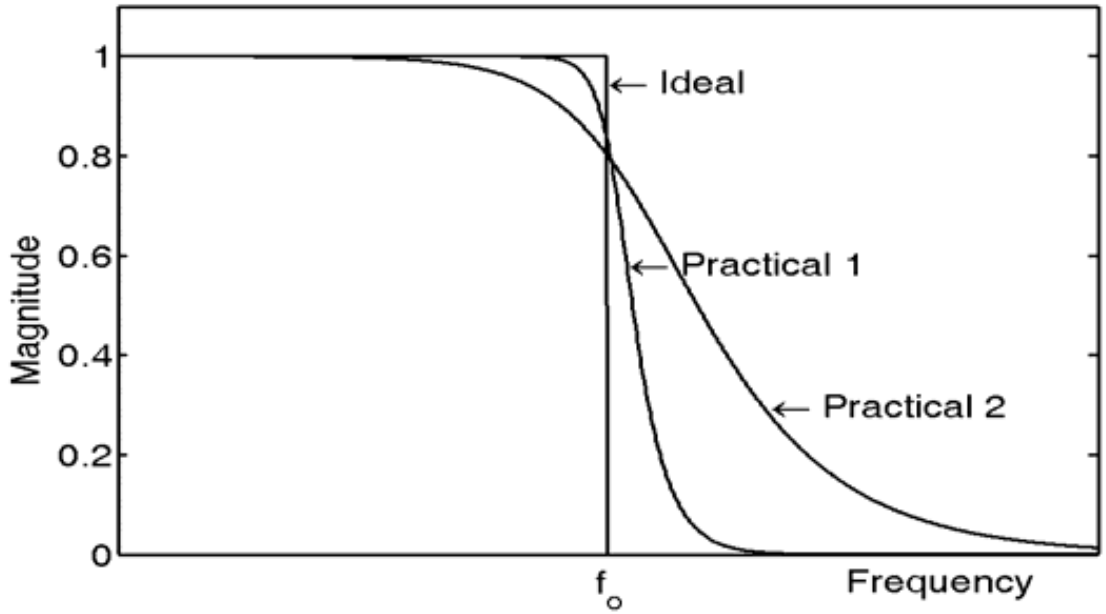


Bandstop



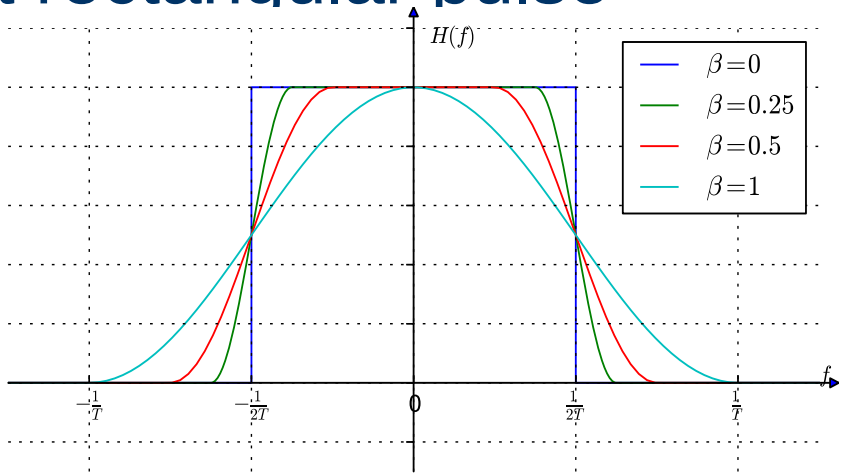
Filter Performance

- Performance is evaluated using impulse input
- The impulse response is an approximation to the smooth, continuous time response as arithmetic and digital representation of filter parameters do not allow perfect response



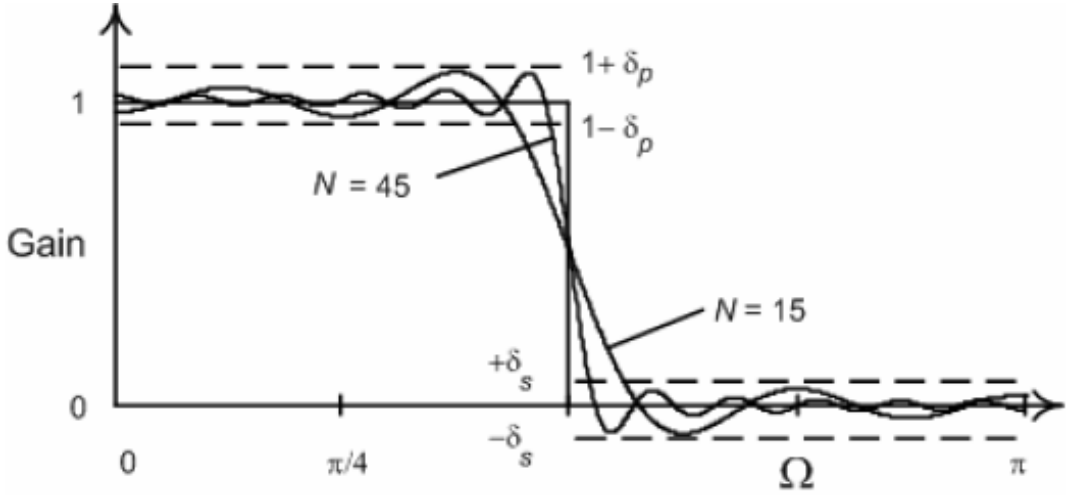
Filter Performance

- Example
- It is not possible to implement the ideal $sinc$ function (brick wall frequency response) required for root raised cosine pulse shaping used in quadrature modulation
- Which is then approximated and the response is never a perfect rectangular pulse



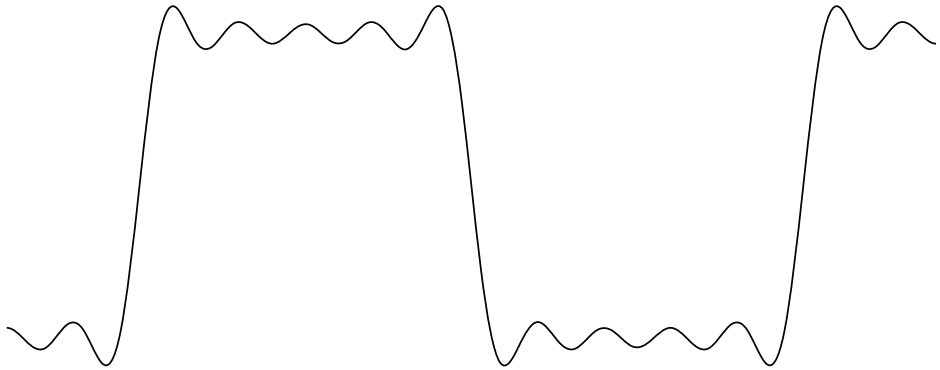
Filter Performance

- We are interested in how accurate the impulse response, the roll off and passband and stopband turns out (in terms of amplitude, sharpness, cutoff and ringing
- Better accuracy incurs complexity in filter design, i.e., in digital logic, hardware area, latency and data representation/size and stability of the filter



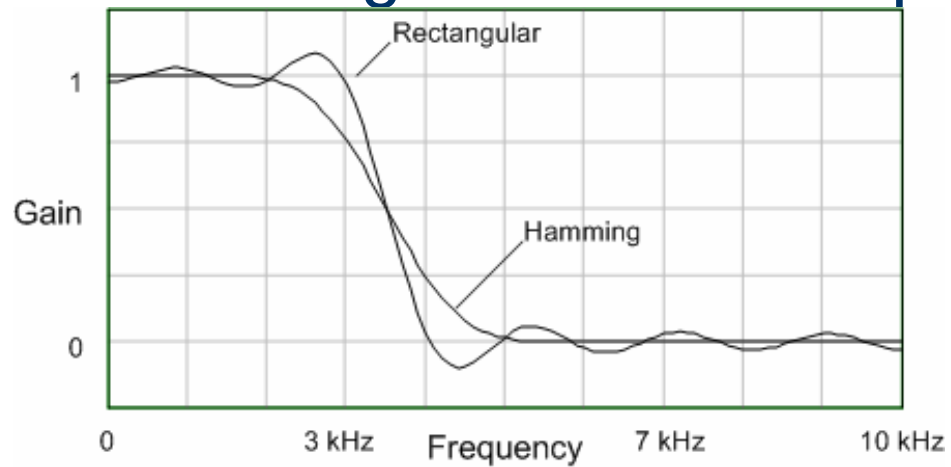
Windowing

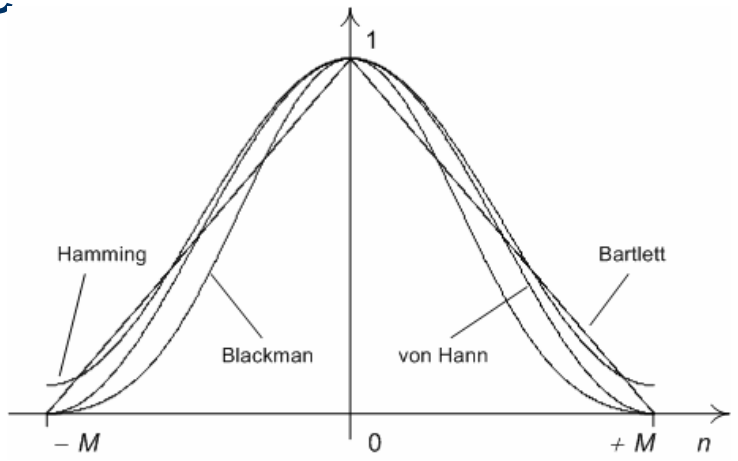
- Purpose of windowing is to smooth or suppress low amplitude harmonic effect called Gibbs phenomena
- The Gibbs phenomenon is an overshoot or undershoot aka ringing of Fourier series occurring at discontinuities
- The overshoot starts at discontinuity (transition) and simply does not die off sooner as more terms are added to the Fourier sum



Windowing – another reason

- DFT computation results in leakage of low amplitude signals in neighbouring bins
- A window function minimizes the leakage of $\frac{\sin x}{x}$
- Windowing is DSP is similar phenomenon of Comm's pulse shaping
- Windowing affects the response





Windowing

- Window controls filter passband and stopband ripples
- Windows usually achieve decay toward ideal rectangle
- Window examples are Rectangular, Hamming, Hann, Tukey, Chebyshev, Kaiser, Blackman, Bartlett etc.
- MATLAB Exercise, type window at MATLAB prompt
- Check frequency response of different windows

Mathematics for Digital Filters

Z-Transform

- Laplace transform is used for continuous time representation of signals domain: e^{-st} (s-plane)
- Fourier transform is use used for frequency domain $j\omega$ representation of signal domain: $e^{-j\omega}$ ($j\omega$ plane)
- z transform is used of discrete time domain: z^{-1}(z plane)
- Each transform exposes new characteristics about signal or makes the realization of signal's transfer function simpler

Analog and Digital Analogy

- Analog system e.g, control, communication, filter etc.
- Continuous in time
- Differential equation
- Smooth
- Solve by Laplace transform

- Digital system e.g, control, communication, filter etc.
- Discrete in time
- Difference equation (algebraic, recursive)
- Successive values of a function of a discrete variable (sampled values of continuous time variable)
- Solve by z-transform

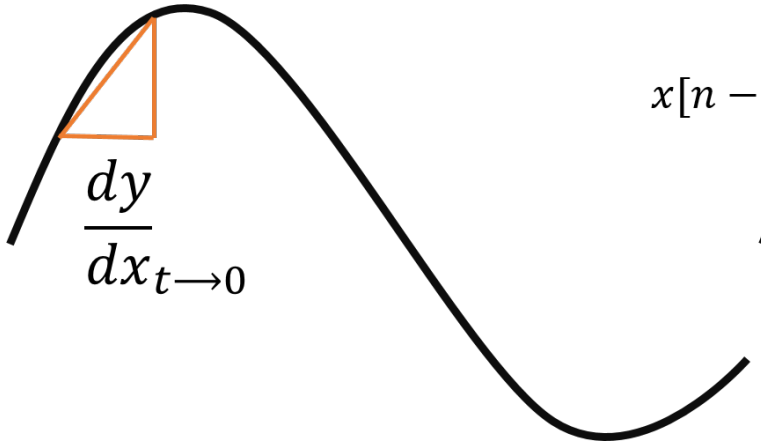
Analog and Digital Analogy

- Analog
- Derivative

$$\frac{d\,x(t)}{dt} \xrightarrow{\mathcal{L}} s\,X(s)$$

- Integration

$$\int_{-\infty}^{t} x(t) \xrightarrow{\mathcal{L}} \frac{1}{s}\,X(s)$$

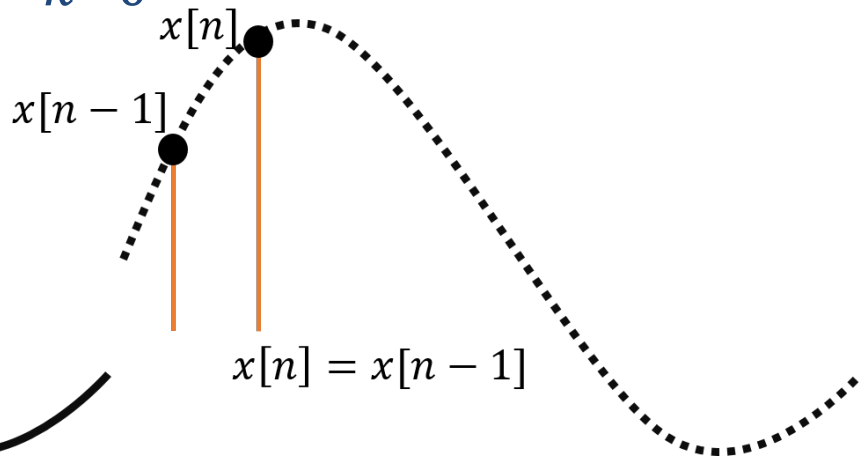


- Digital
- Difference

$$x[n] - x[n-1] \xrightarrow{z} [1 - z^{-1}]\,X(z)$$

- Summation

$$\sum_{k=0}^{n} x[n] \xrightarrow{z} [\frac{1}{1 - z^{-1}}]\,X(z)$$



Z-Transform

- The Fourier transform of a sampled sequence $x[n]$ is defined as

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}$$

Let z be a continuous complex variable such that $z = e^{j\omega}$ then the z-transform of $x[n]$ is defined as

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]\, z^{-n}$$

If $x[n]$ is causal i.e., $x[n] = 0$ for $n < 0$ which most real world signals are, the transform is one-sided or unilateral

Linear Time Invariant

- The theory of digital filter design and analysis applied to linear time invariant systems
- Linear: superposition applies i.e., for inputs $x_1[\mathrm{n}]$ and $x_2[\mathrm{n}]$ the respective outputs are $y_1[\mathrm{n}]$ and $y_2[\mathrm{n}]$ which results in combined response
- Linear: scaling homogeneity applies i.e., $T\left[ax_1[\mathrm{n}]\right] = aT\left[x_1[\mathrm{n}]\right] = \mathrm{a}y_1[\mathrm{n}]$
- So the two properties of linearity mean

$$T\left[ax_1[\mathrm{n}] + bx_2[\mathrm{n}]\right] = aT\left[x_1[n]\right] + bT\left[x_2[n]\right]$$

- We expect resemblance in input and output

Linear Time Invariant

- Time Invariant: a delay or shift in time in input causes a delay or shift in the output

$$x[n] \longrightarrow y[n]$$
$$x[n - n_0] \longrightarrow y[n - n_0]$$

- If input is delayed, output is delayed
- Without these two assumptions, the system transfer is nonlinear
- Nonlinear system theory does not go very far in science
- Mathematics, when exists, too difficult
- Look into Wiener-Hammerstein nonlinear identification

z-Transform

- The z transform defines the input $X(z)$ and output $Y(z)$ relationship of a linear time invariant system transfer function $H(z)$

$$H(z) = \frac{Y(z)}{X(z)}$$

- The $H(z)$ is the z transfer function of a discrete time sequence $h[n]$
- $h[n]$ is impulse response of the system when input $h[n]$ is given an impulse signal i.e., $x[n] = \delta[n]$

nth order Discrete System

- General nth order system:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k \, z^{-k}}{1 - \sum_{k=1}^{N} a_k \, z^{-k}}$$

- General nth order difference equation is

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

which can be rewritten as

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

Digital Filter

$$H(z) = \frac{Y(z)}{X(z)}$$

- The values of z for which $H(z) = 0$ are called zeros
- The values of z for which $H(z) = \infty$ are called poles (roots of the denominator $X(z)$)
- $H(z)$ is the digital filter
- It is complete described by the numerator and denominator polynomials i.e., poles and zeros
- If you know the polynomial coefficients a_k and b_k you know the filter
-

Digital Filter

- Given $y[n] = x[n] - 0.5\,x[n-1] + 0.36\,x[n-2]$
- Recall from signals course the z transform pairs for several types of signals
- We take transform of all the terms

$$Y(z) = X(z) - 0.5\,X(z)\,z^{-1} + 0.36\,X(z)\,z^{-2}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k\, z^{-k}}{1 - \sum_{k=1}^{N} a_k\, z^{-k}} = 1 - 0.5\,z^{-1} + 0.36\,z^{-2}$$

$$b_k = [b_0 \quad b_1 \quad b_2] = [1 \quad -0.5 \quad 0.36]$$

$$a_k = [a_0] = [1]$$

Digital Filter

- Once you know the filter coefficients b_k, a_k you can then filter signal
- Recall Prac 2: y = filter(b,a,x)
- To get frequency response and cut off frequency In MATLAB use [h,w] = freqz(b,a,n)

Discrete System Structures

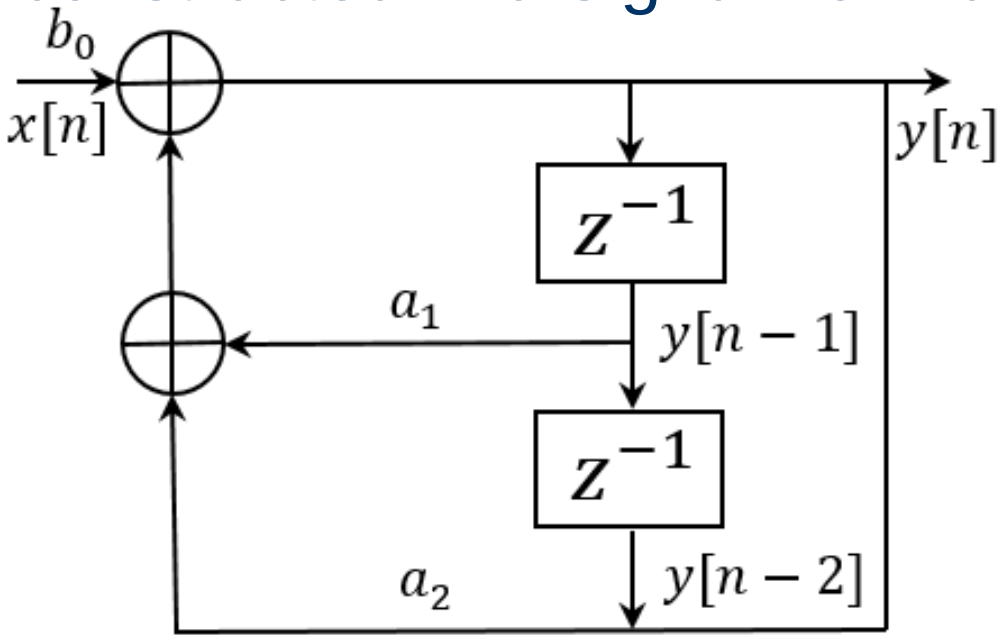
- Or, digital filter structures
- Given any computable transfer function, a number of equivalent difference equations, network topologies and signal flow graphs exist to describe the relation of system inputs to the outputs
- Each structure results in a different but computationally equivalent discrete structure
- You can rearrange the difference equation

Discrete System Structures

- Consider $y[n] = a_1 y[n-1] + a_2\, y[n-2] + b_0 x[n]$

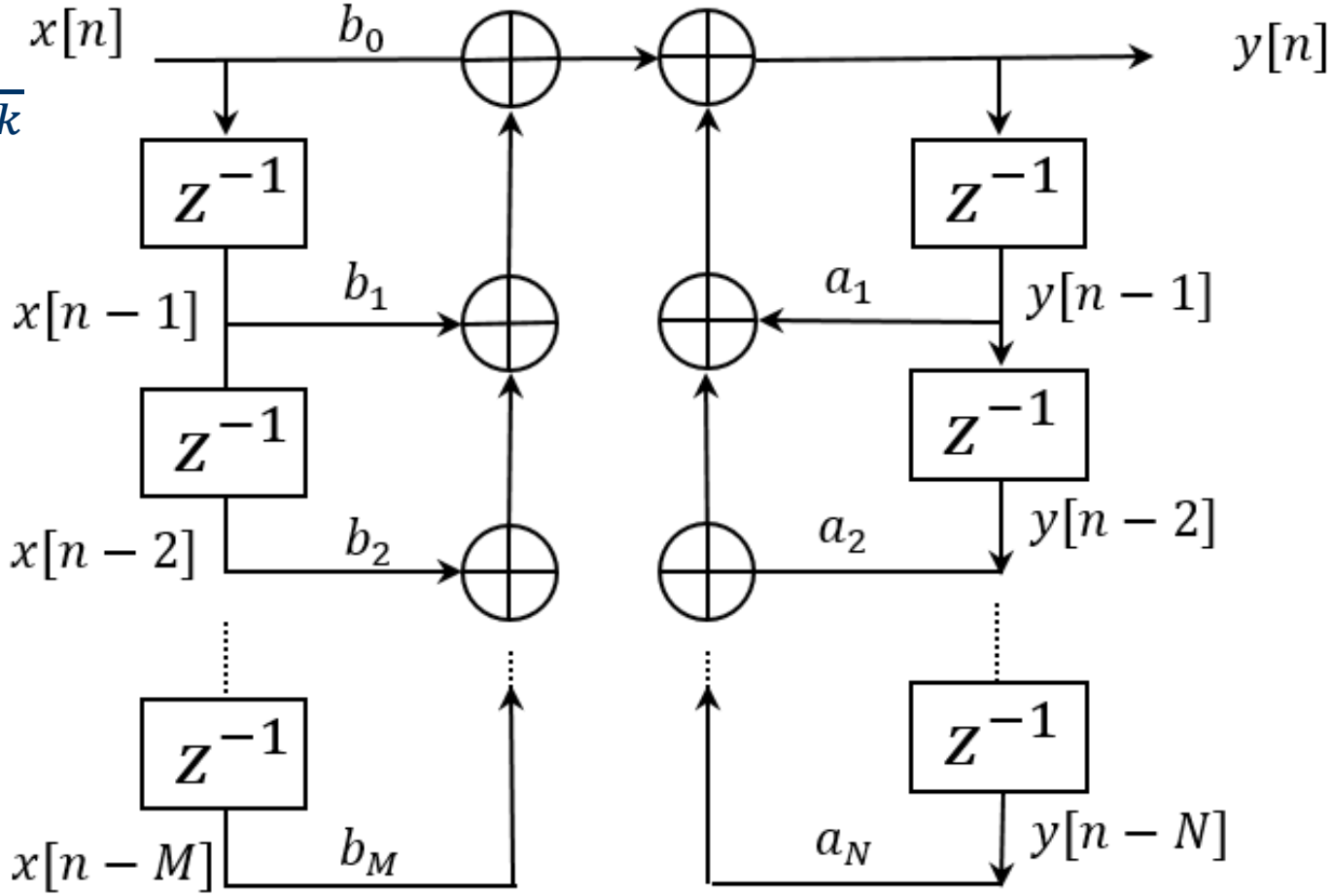
$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0}{1 - a_1\, z^{-1} - a_2\, z^{-2}}$$

which may be constructed in a signal flow fashion as



nth order Digital Filter

- The general nth order system may be constructed as
- $H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1-\sum_{k=1}^{N} a_k z^{-k}}$



Digital Filter Types

- Finite Impulse Response (FIR)
- Infinite Impulse Response (IIR)

Finite Impulse Response Filter

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

- $H(z)$ has no denominator or poles so partial fraction expansion is not possible
- The impulse response is

$$h[n] = \begin{cases} b_n & b = 0,1..M \\ 0 & else \end{cases}$$

which is bounded (finite) and 0 outside the interval hence the name FIR

- Settling time of the response is finite and it dies off soon

Finite Impulse Response Filter

- FIR difference equation is identical to convolution

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

which is discrete convolution with impulse response

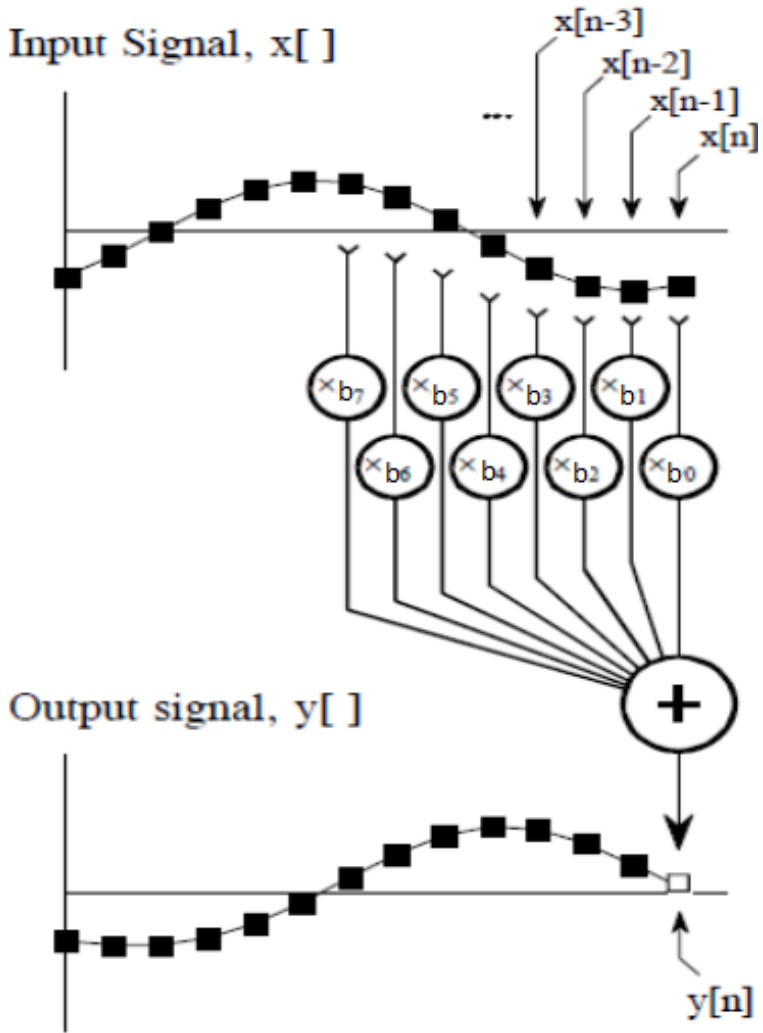
$$h[n] = \begin{cases} b_n & b = 0,1..M \\ 0 & else \end{cases}$$

- FIR filters are computationally convolutional filters
- An FIR filter performs this calculation by multiplying appropriate samples from the input signal by a group of coefficients, denoted by: b_0, b_1, b_2, .. and then adding the products (element by element multiplication)

$$y[n] = b_0 x[n] + b_1 \, x[n-1] + b_2 x[n-2] + ..$$

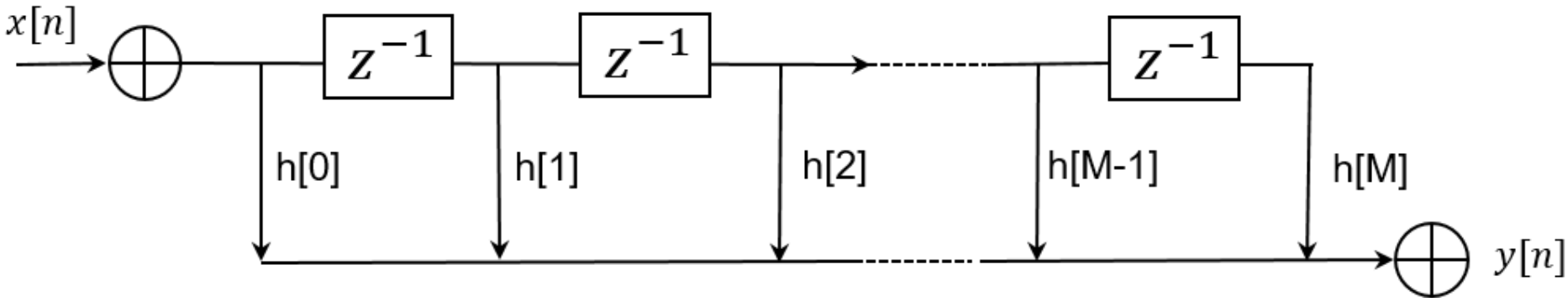
Finite Impulse Response Filter

- This means that the input signal $x[n]$ has been convolved with an impulse response of the filter consisting of b_0, b_1, b_2 …
- Because multiplication in one domain is convolution in other domain



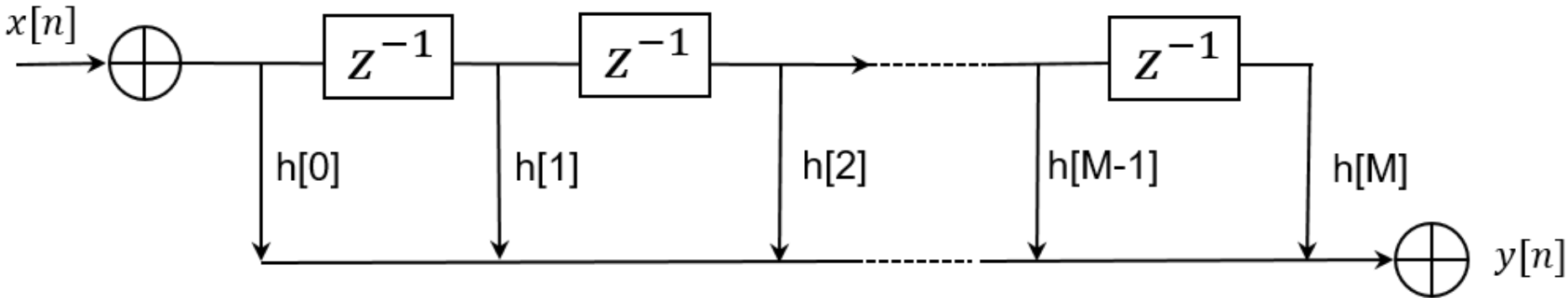
Direct Form FIR Filter

- FIR has the structure



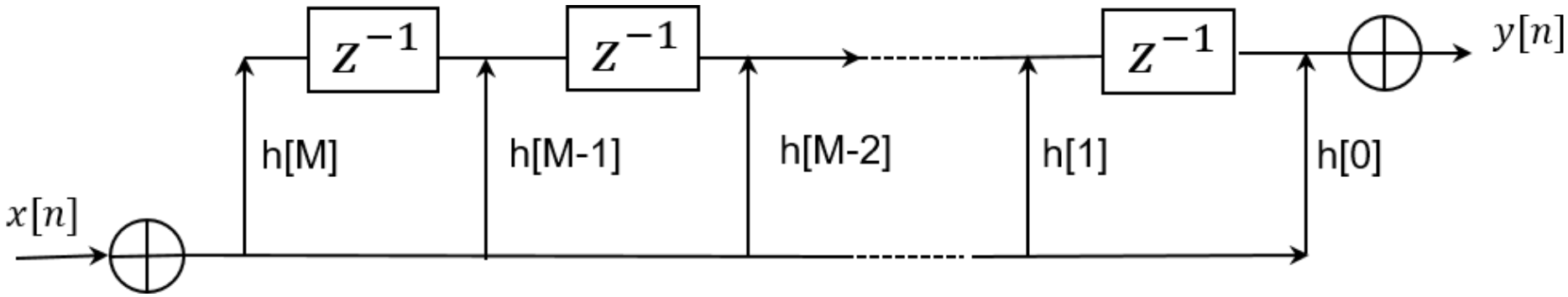
Direct Form FIR Filter

- FIR has the structure

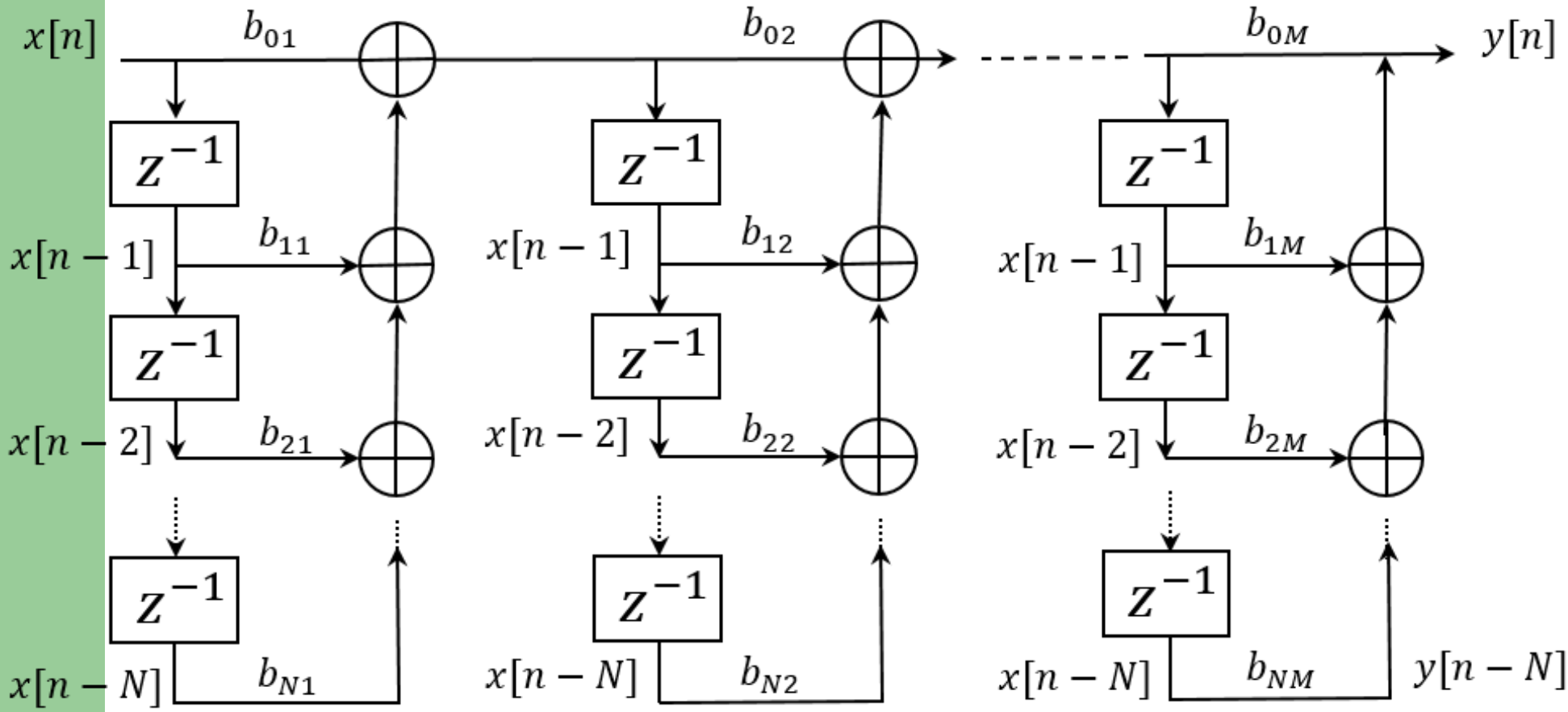


Transposed FIR Filter

- FIR direct structure may be transposed



Cascaded FIR Filter



Design a 48th-order FIR bandpass filter with passband $0.35\pi \leq \omega \leq 0.65\pi$

FIRs in MATLAB

- To design a 48th-order FIR bandpass filter with passband $0.35\pi < \omega < 0.65\pi$ rad/sample, run FIR_BPF.m
- To design a 34th-order FIR highpass filter to attenuate the components of the signal below $Fs/4$. Use a cutoff frequency of 0.48 and a Chebyshev window with $30\ dB$ of ripple. Run FIR_HPF.m on chirp
- Run FIR_HPF_and_LPF.m

Moving Average Filter

- Output is produced by averaging inputs i.e.,

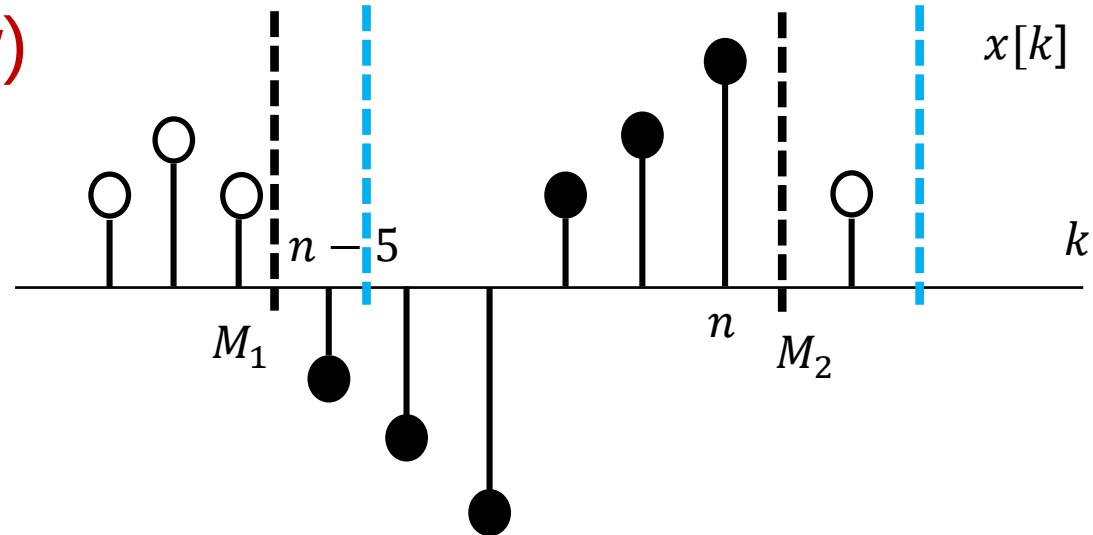
$$y[n] = \frac{1}{1 + M_1 + M_2} \sum_{k=-M_1}^{M_2} x[n-k] =$$

$$\frac{1}{1 + M_1 + M_2} [x[n + M_1] + [x[n + M_1 - 1] + .. + x[n]$$
$$+ x[n-1] + .. + x[n - M_2]$$

- nth sample of the output sequence is computed as the average of $1 + M_1 + M_2$ samples occurring around the nth sample, i.e., $1 + M_1 + M_2$ is the number of points in the average

Moving Average Filter

- For $n = 7, M_1 = 0, M_2 = 5$
- Output is averaged between the black dotted window specified by M_1 and M_2 i.e., $y[7] = \frac{1}{6}[x[2] + x[3] + x[4] + x[5] + x[6] + x[7]]$
- For $y[8]$ the window slides one sample forward (cyan dotted window)



Moving Average Filter

- The moving average filter is a convolution of the input signal with a rectangular pulse having an area of one
- The frequency response smooths the input data
- Moving average filter is a special case of FIR as the window includes finite numbers of samples
- Run MA_1D.m

Difference in Moving Average and FIR Filter

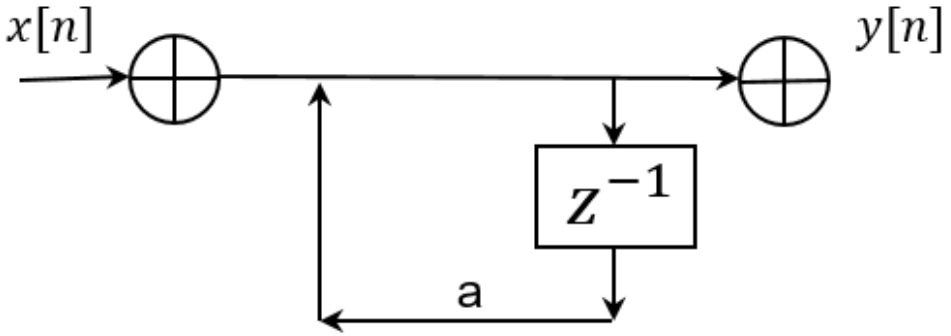
- The smoothing action of the moving average filter attenuates noise but reduces the sharpness of roll off
- The problem is this: the moving average filter uses a sequence of scaled 1's as coefficients
- For example, a 5 point filter has the filter kernel $\ldots 0, 0, 1/5, 1/5, 1/5, 1/5, 1/5, 0, 0 \ldots$
- Note in the difference equation, there is no multiplier
- While the FIR filter $[n] = b_0 x[n] + b_1\, x[n-1] + b_2 x[n-2] + \ldots$ coefficients are multiplied to input samples
- FIR coefficients are designed based on the filter specifications (roll off, pass ripple etc.)

Difference in Moving Average and FIR Filter

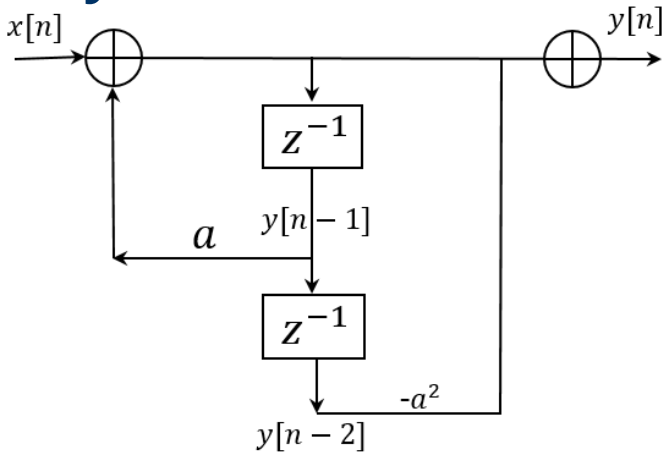
- To see the difference, see response of FIR and MA filters having same coefficients. Run FIR_vs_MA.m
- MA filter main lobe in the passband is not flat and the ripples in the stopband are not constrained, the frequency response does not match the frequency response of the ideal filter
- FIR response in the passband is almost flat similar to the ideal response and the stopband has constrained equiripples
- Check implementation cost: Analysis → Filter Information, Analysis → Overlay Analysis

Frequency Sampling FIR Filter

- Consider FIR $y[n] = ay[n-1] + x[n] \Rightarrow Y(z) = \frac{1}{1-az^{-1}}$



- But if the pole of the system cancels a zero, the impulse response is finite



Frequency Sampling FIR Filter

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 - a^2\, z^{-2}}{1 - a\, z^{-1}} = \frac{(1 - a\, z^{-1})(1 + a\, z^{-1})}{1 - a\, z^{-1}}$$

$$H(z) = 1 + a\, z^{-1} \Rightarrow h[n] = \delta[n] + a\, \delta[n-1]$$

- This class of FIR is called Frequency Sampling filter having a general form

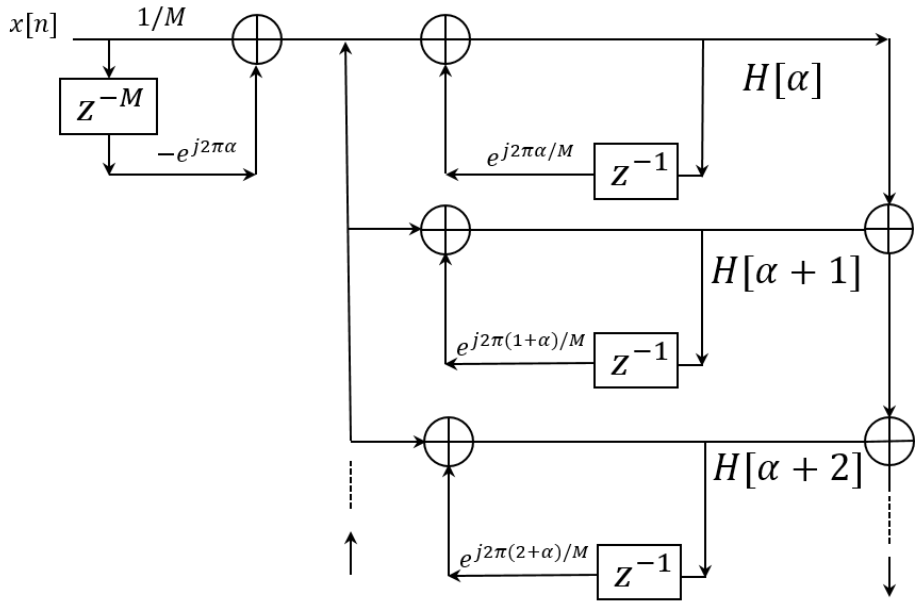
$$H(z) = \left[\frac{1}{M}\left(1 - z^{-M}\, e^{j2\pi\alpha}\right)\right]\left[\sum_{k=0}^{M-1} \left[\frac{H[k+\alpha]}{1 - e^{j2\pi\{k+\alpha)/M}\, z^{-1}}\right]\right]$$

$$H(z) = H_1(z) H_2(z)$$

- The desired frequency response is sampled at regular frequency intervals

Frequency Sampling FIR Filter

- The system function is characterized by frequency samples $H[k+\alpha]$ e.g., $\omega_k = 0, \pm\frac{2\pi}{9}, \pm\frac{4\pi}{9}$,.. $H(\omega_k) = 1,1,0..$
- The realization is two filters $H_1(z)$ all zero filter and $H_2(z)$ is a parallel bank filter with resonant frequencies
- Frequency response is sampled where desired



Frequency Sampling FIR Filter

- Frequencies are equally spaced
- Run FIR_Freq_Sampling.m on chirp
- Chirp is a signal whose frequency increases and decrease, arises in radar and sonar applications
- Frequency sampling filters have linear phase response, confirm in above MATLAB code

Digital Down Conversion (DDC)

- To translate a RF modulated signal to an Intermediate Frequency (IF)
- E.g,, converting a $64kHz$ to $8kHz$ by simply discarding every seven out of eight samples
- DDC requires decimation which is removing samples
- Run decimation.m, sine100hzsamp.m
- Run decimation_chebyshev.m
- Chebyshev filters have steeper roll off than Butterworth filters (which have flatter response but less roll off), but Chebyshevs have ripples in passband unlike Butterworth
- Run decimation_FIR.m

Aliasing Due to DDC

- Aliasing is the distortion that occurs when overlapping copies of the signal's spectrum are added together
- The more the signal's baseband spectral support exceeds $2\pi/M$ radians, the more severe the aliasing
- Run DDC_aliasing.m

Root Raised Cosine Filter

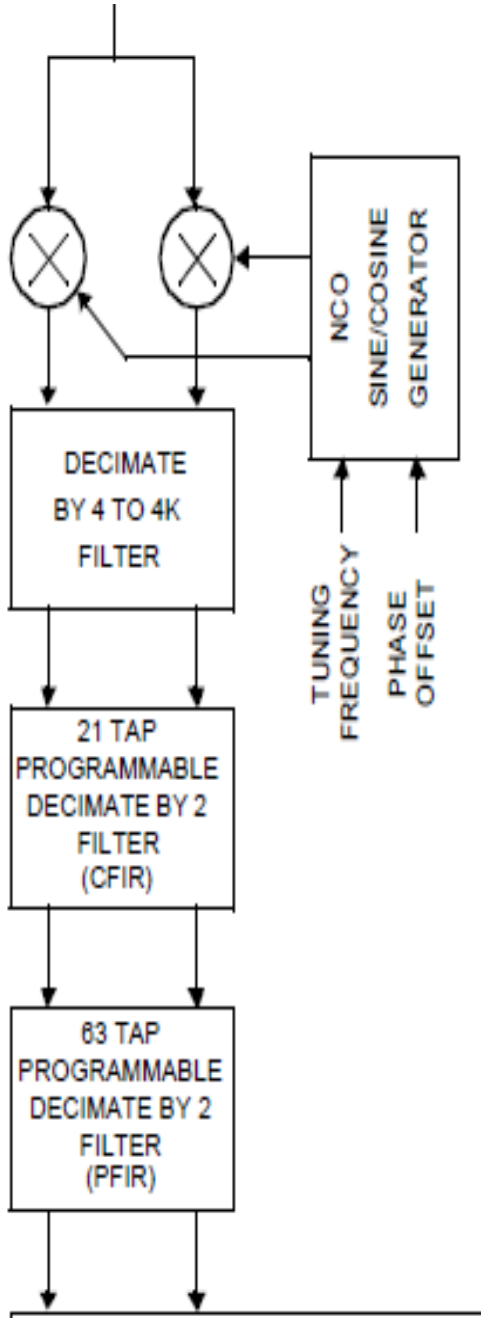
- RRC needed in many comms applications such as using QPSK modulation
- Need zero crossing to keep ISI zero
- RRC is an anti-ISI filter modeled with $sinc()$
- Run rrc.m
- Run rrc2.m
- Run rrc_data_passing.m
- Run rrc_data_passing.m

Comb FIR Filter CIC

- Cascaded Integrator Comb (CIC) filter is an FIR used in Digital Down Converter need when higher sampling rates are changed to lower sampling rates i.e., high frequency signal is converted to low frequency
- Comb can be implemented as a simple moving average filter
- Practical DDC applications require an M stage integrator integrating all sampling frequencies, then a decimator of L and a cascaded M stage comb filter
- The filter has a response resembles a toothcomb

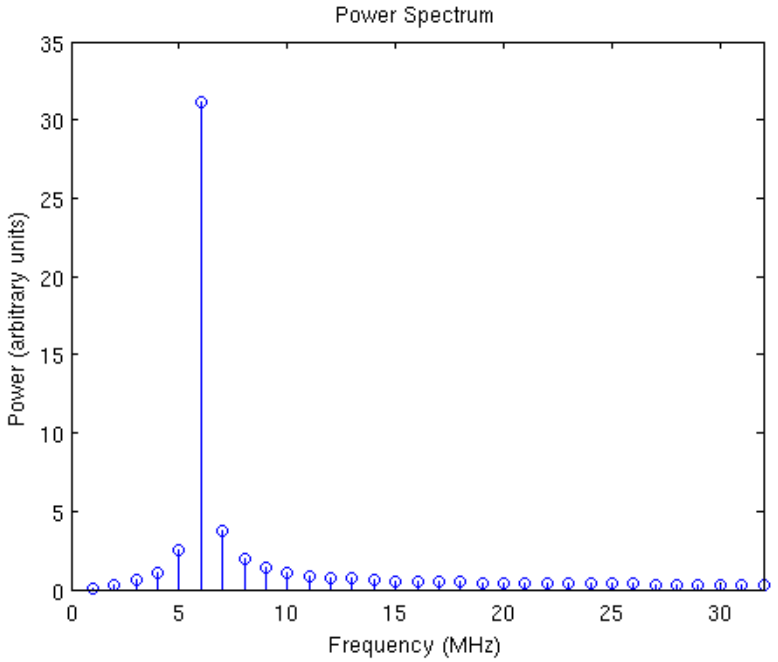
Comb FIR Filter CIC

- $H(z) = \sum_{k=0}^{M-1} z^{-k} \frac{1-z^{-M}}{1-z^{-1}} =$
- $= \sum_{k=0}^{M-1} z^{-k} \left[1 - z^{-M}\right]\left[\frac{1}{1-z^{-1}}\right]$
- (integrator) (comb)
- Run DDC.m
- Run DUP.m



Polyphase Filters

- DFT computation results in spectral leakage
- Ordinarily, this leakage could be tolerable in regular communication applications
- But in RFI, radio astronomy, and spectrometery, the spectral leakage could be devastating
- This can be resolved with a polyphase filter bank and window



https://casper.berkeley.edu/wiki/The_Polyphase_Filter_Bank_Technique

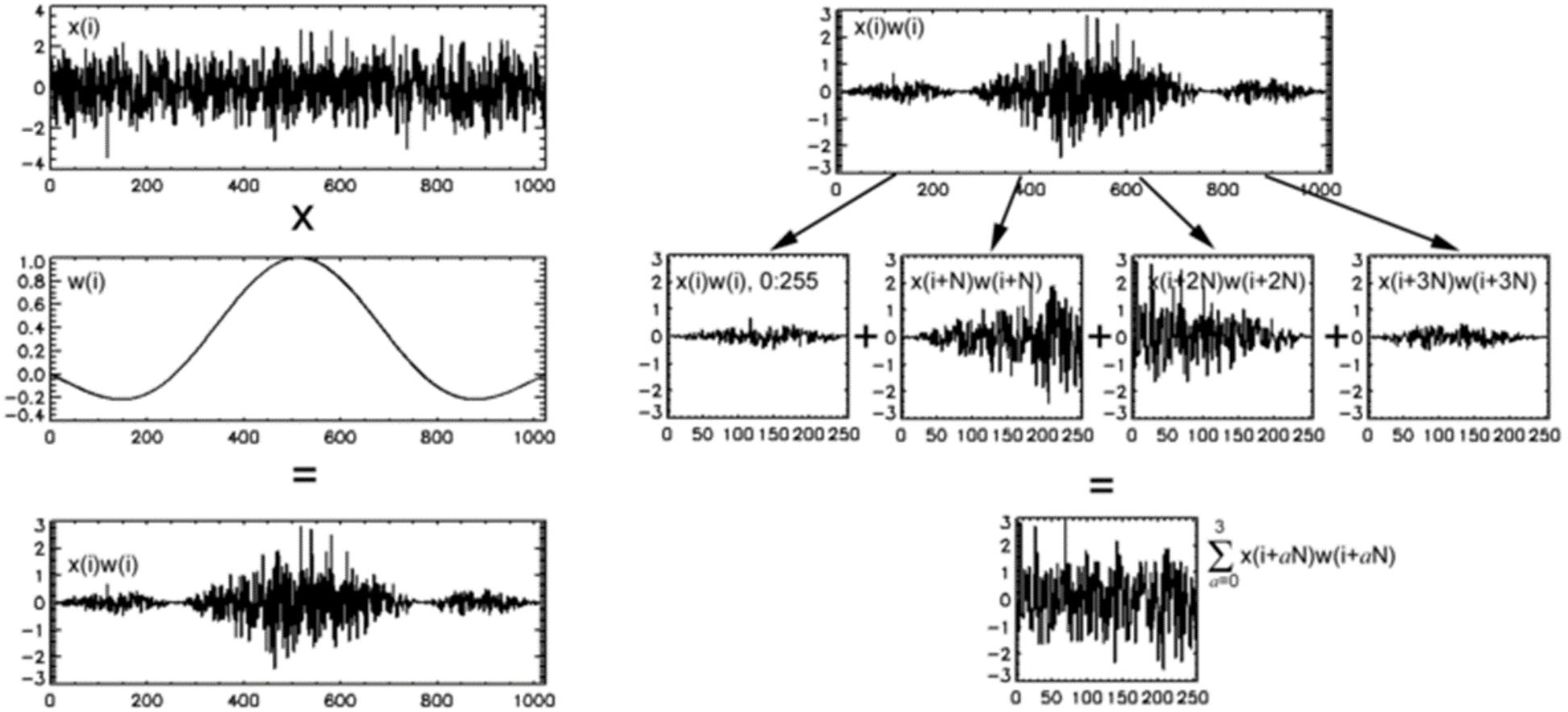
Polyphase Filters

- Data is windowed i.e., weighted multiplication
- After multiplication, the data is chunked (split) into P sets, each set of length N e.g., 1024 split in 4 chunks of length 256
- All chunks are added element wise
- Once added, they are passed thru DFT

Polyphase Filters

- Data is windowed i.e., weighted multiplication
- After multiplication, the data is chunked (split) into P sets, each set of length N e.g., 1024 split in 4 chunks of length 256
- All chunks are added element wise
- Once added, they are passed thru DFT
- Polyphase filtering is computationally expensive than FFT/DFT but response is flatter and leakage is less
- May be useful for parallel / HCA platforms where dataset is large!

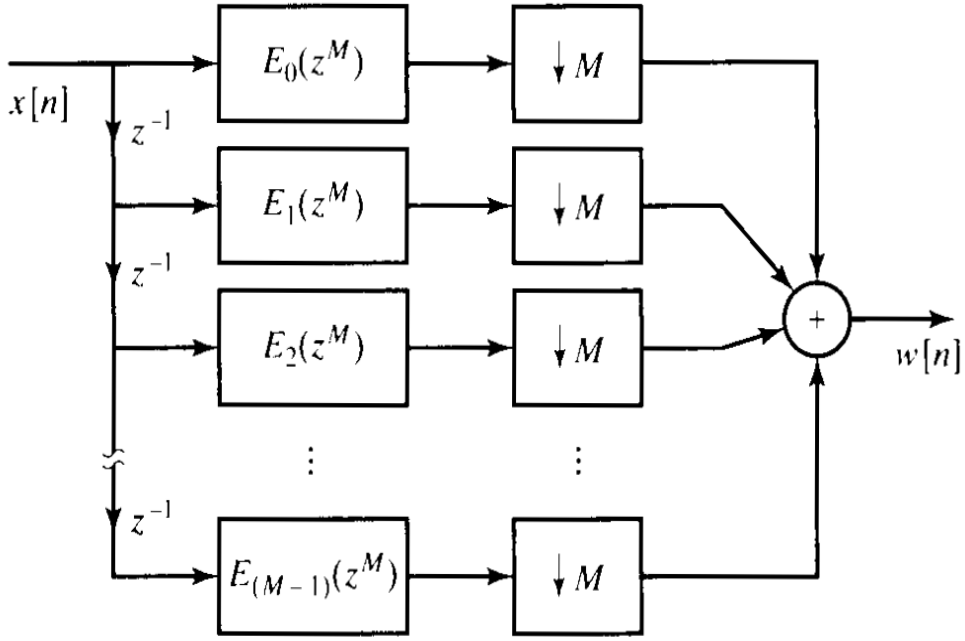
Polyphase Filters



https://casper.berkeley.edu/wiki/The_Polyphase_Filter_Bank_Technique

Polyphase Filters - Channelization

- Input broadband signal, which the channelizer splits into multiple narrow bands
- The channelizer is a polyphase filter bank of FIR implementations for rate conversion whether upsampling / interpolation or downsampling
- Polyphase is a multirate DSP channelization algorithm
- Multirate DSP applications are common to radar and radio astronomy where data is acquired from multiple sources



Polyphase Filters - Channelization

- The input sequence is considered a superposition of M sequences
- This decomposition is applied to impulse response of M parallel FIR filters
- Each filter receives a delayed version of the input sequence i.e., for the Mth filter the delay is z^{M-1}
- By successively delaying the sequence, original impulse response can be reconstructed
- Run HighResolutionFilterBankSpectralEstimation.slx

Polyphase Filters

- Run HighResolutionFilterBankSpectralEstimation.slx
- In this example, the full band estimator requires a 512-phase polyphase FIR filter and a 512-point FFT in order to compute the spectral estimate
- The sinusoid frequencies in each sub-band are spaced further apart as the frequency increases
- The idea is to setup a case in which higher frequency resolution is required at the low frequency band and lower resolution is required at higher frequency bands

Polyphase Filters

- The sub-band approach is more efficient
- It uses an 8-phase polyphase FIR filter (512/64) and an 8-point FFT to divide the broadband signal into 8 sub-bands
- Subsequently, a 64 band filter bank estimator (itself containing a 64-phase polyhase FIR filter and a 64-point FFT) is used with the low frequency sub-band in order to compute the spectral estimate with the same resolution as the full band estimator
- The same implementation is used for the mid-low frequency band

Linear Phase Property of FIRs

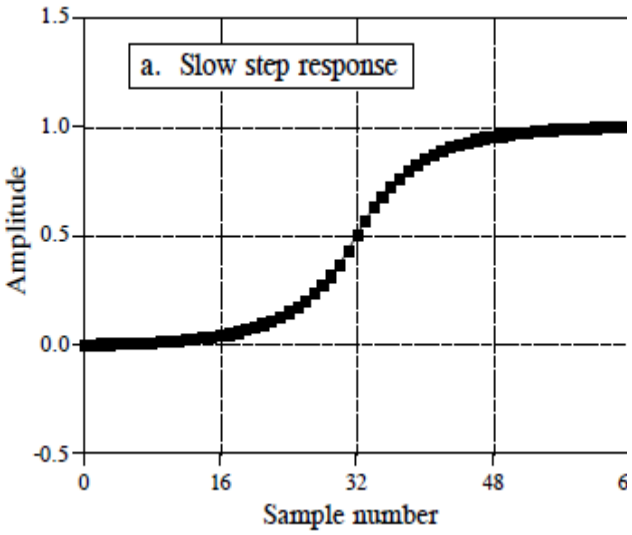
- During transmission amplitude of the sinusoids incur time delays according to the their frequencies, this is called group delay, recall Prac 2
- Similarly, the phase goes thru delays, Both effects change the shape of the bandlimited signal to be no more as designed by pulse shaping
- Therefore, it is generally required to control the group delay to be constant and phase to be linear
- Phase distortion is a terrible effect for phase based modulations and linear modulations (IQ modulations,QPSK)
- FIR filters can preserve pulse shape and phase linearity

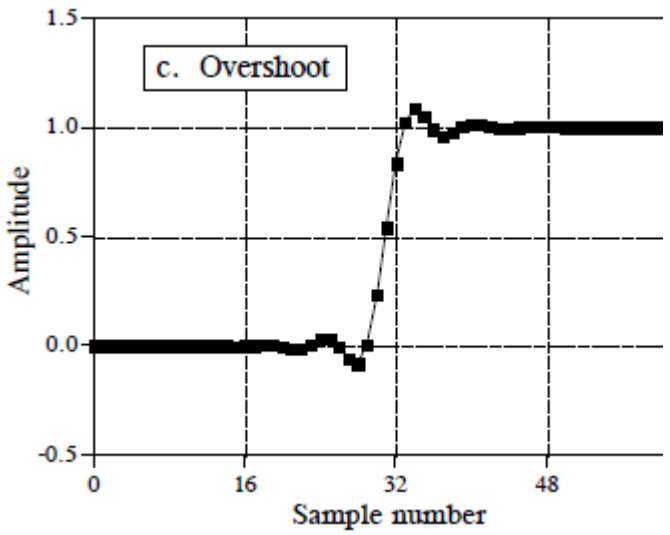
Group Delay

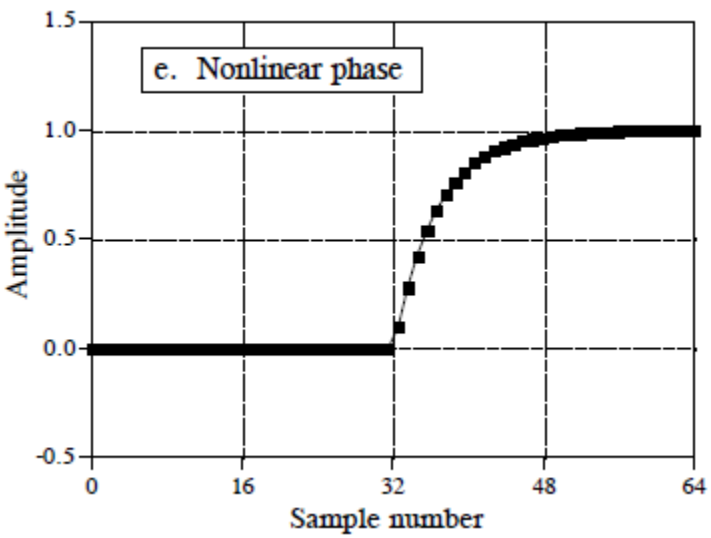
- Signal contents may be delayed at a constant rate or may be dependent on frequencies
- FIR and IIR filters also introduce delays
- The delay must be determined and compensated to bring the waveform forward
- Run FIR_delay_compensation.m
- Run IIR_delay_compensation.m

Linear Phase Property of FIRs

- Phase discontinuities may occur at certain frequencies else where it will be linear
- If the magnitude response has ringing, check the phase response, more than likely it will be nonlinear
- A requirement of all linear phase modulations e.g., QPSK







Other FIR Filters

- Parks-McClellan FIR
- Remez (obsolete in MATLAB)

Infinite Impulse Response Filter

- IIR systems have at least one non-zero pole (which is not cancelled by a zero in the numerator)
- There is at least one term of the form

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k \, z^{-k}}{1 - \sum_{k=1}^{N} a_k \, z^{-k}}$$

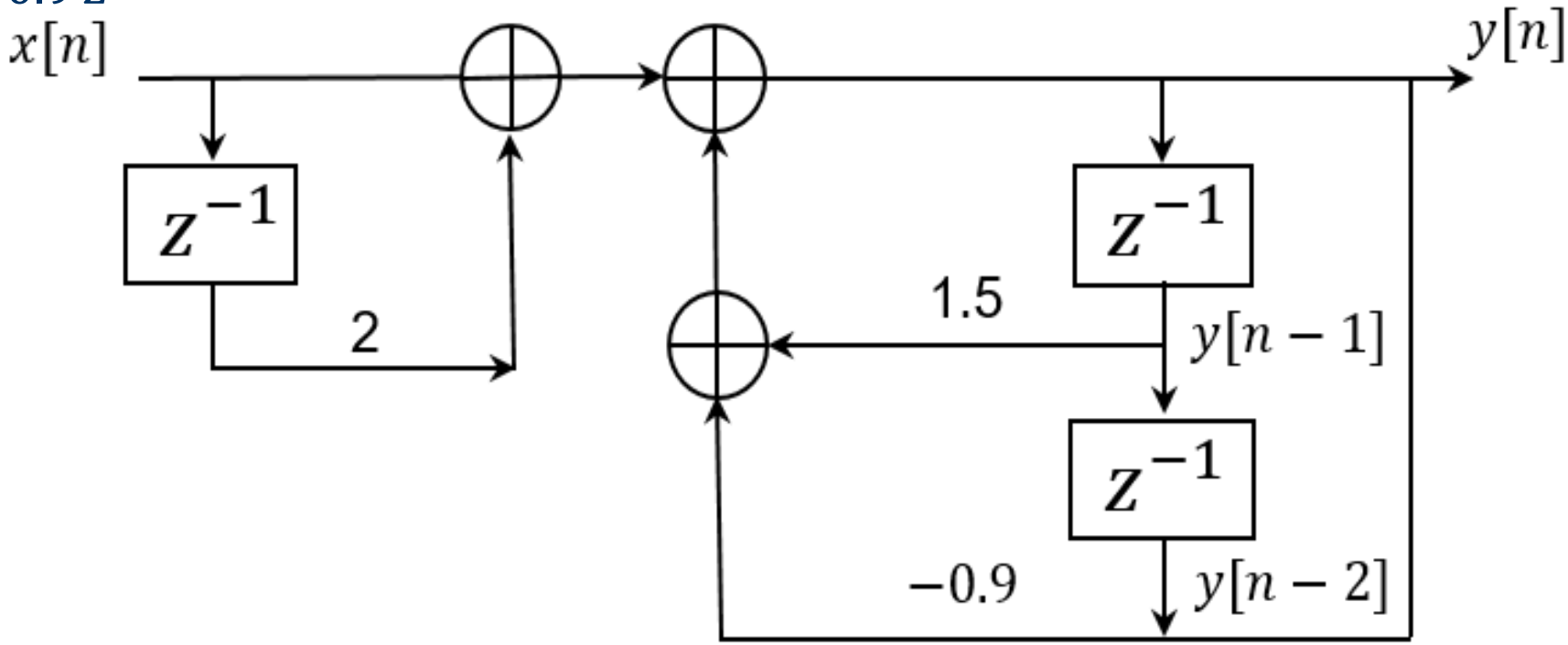
- Therefore $h[n]$ cannot be of the finite length
- I.e., in FIR $H(z)$ tends to 0 for higher k because $\sum_{k=0}^{M} b_k \, z^{-k} \longrightarrow 0$ and converges to 0 (finite non-zero samples of the impulse response)

Infinite Impulse Response Filter

- In IIR $H(z)$ tends to ∞ for higher k because $\sum_{k=1}^{N} a_k \; z^{-k} \longrightarrow \; \infty$ and never converges (impulse response duration is ∞)
- Settling time is large (infinite)

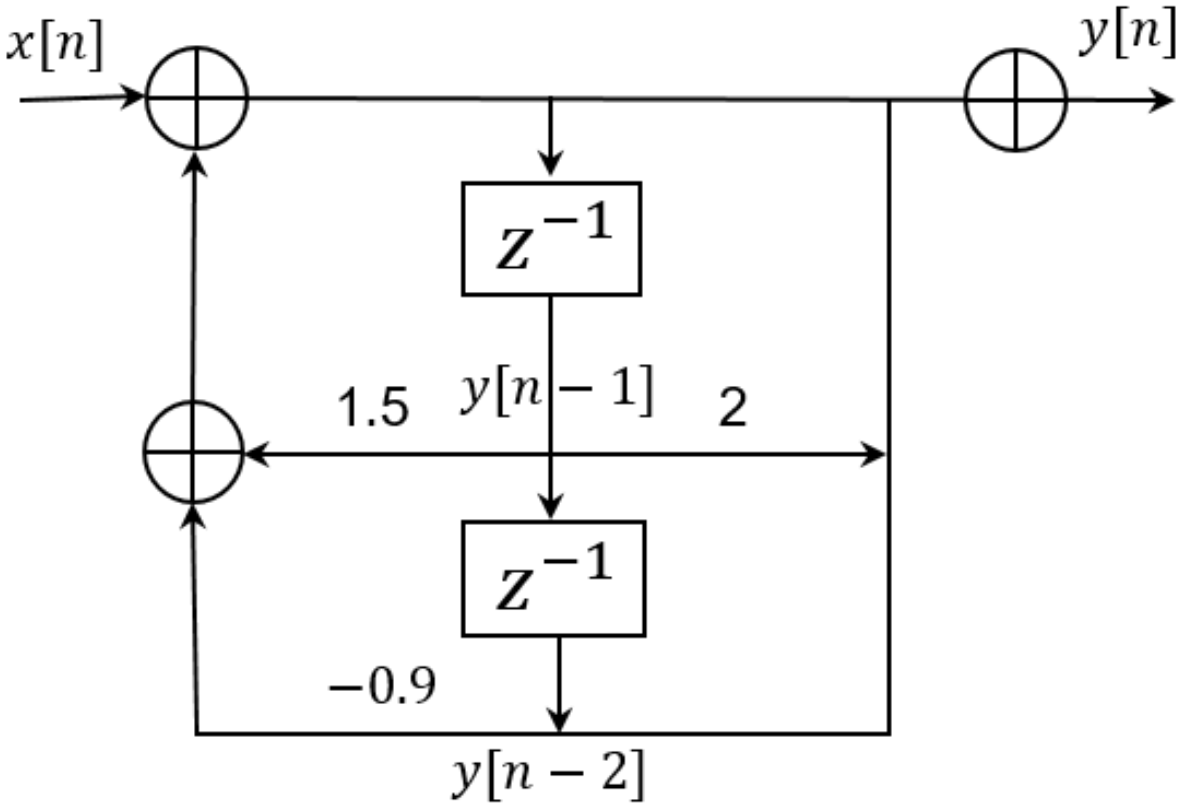
Direct Form I

- Consider $H(z) = \frac{Y(z)}{X(z)} = \frac{1+2\,z^{-1}}{1-1.5\,z^{-1}+0.9\,z^{-2}} = \frac{1}{1-1.5\,z^{-1}+0.9\,z^{-2}} + \frac{2\,z^{-1}}{1-1.5\,z^{-1}+0.9\,z^{-2}}$ which may be constructed as



Direct Form II aka Canonic Direct Form

- Alternatively, another equivalent structure is Direct Form II
- Direct Form II requires less delays and is more efficient on hardware resources



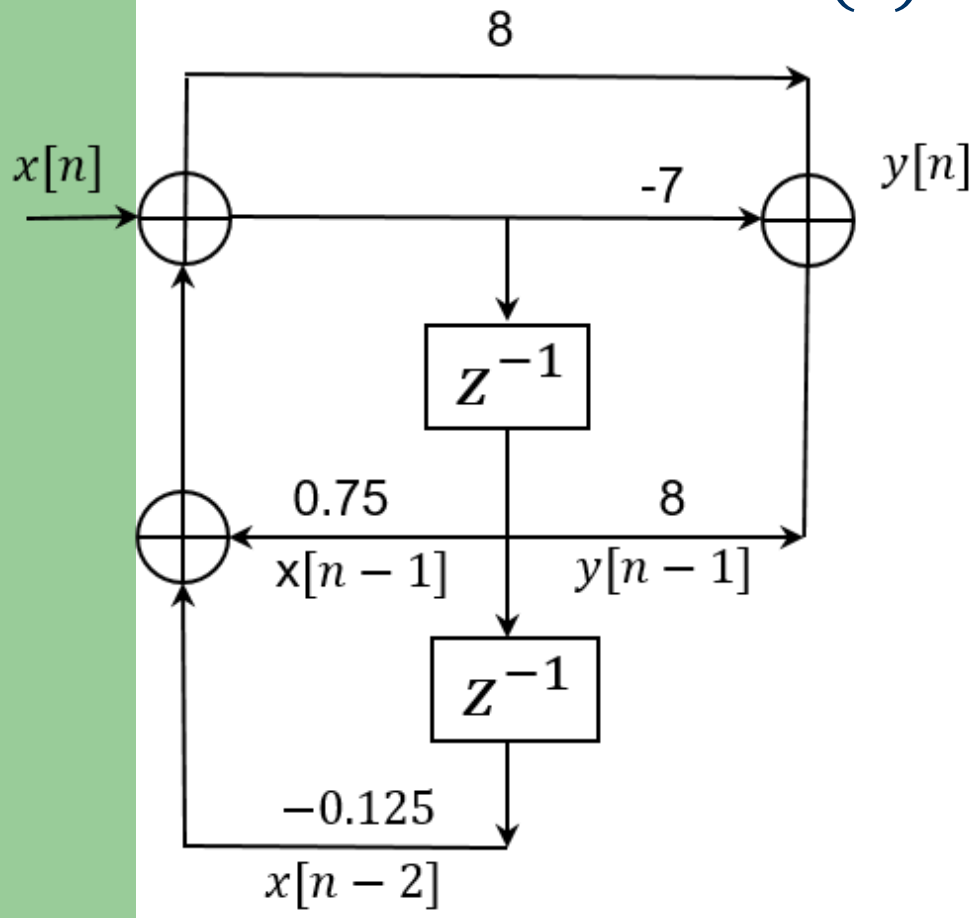
Other Structures

- The design can be broken in different partial fractions of the $H(z)$ and more simpler, reduced order and regular structures may be obtained
- Similarly there are many other useful structures such as Cascaded form, Parallel form and Transpose form -- which can ease recursive computing by repeating a particular mathematical operation (useful for VHDL based design)
- The feedback coefficients always have the opposite signs in the difference equations from their sign in the transfer function

Parallel Form II (Second Order)

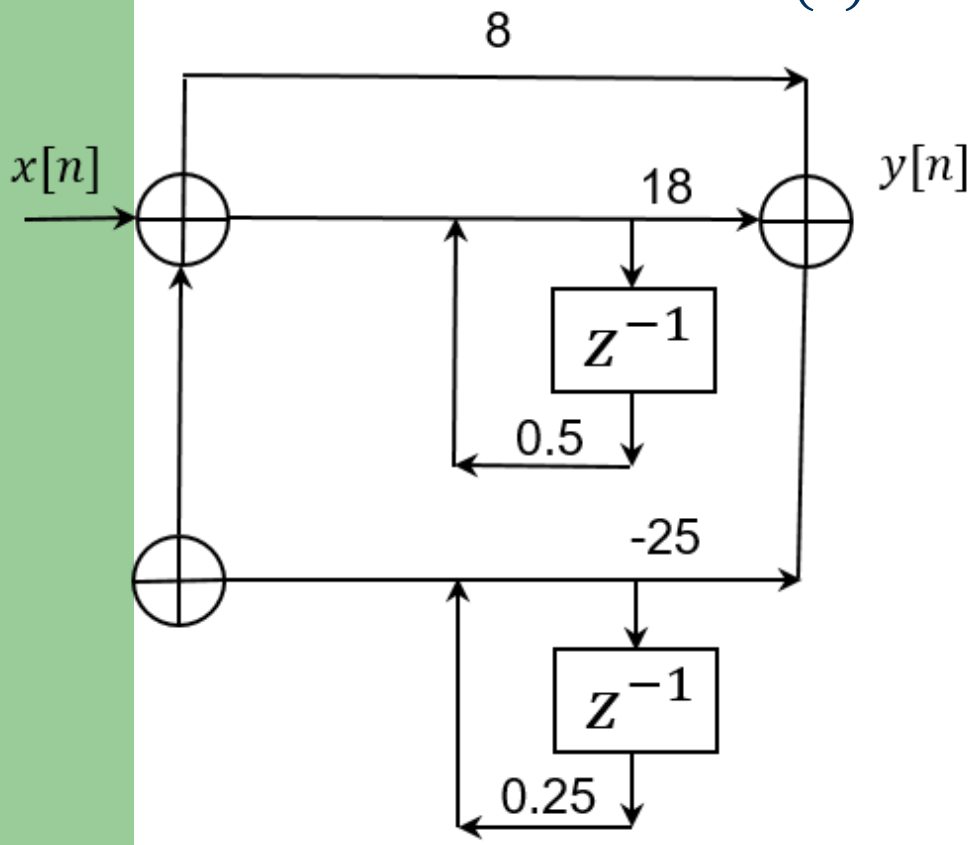
- The transfer function $H(z) = \frac{Y(z)}{X(z)} = \frac{1+2\,z^{-1}+z^{-2}}{1-0.75\,z^{-1}+0.125\,z^{-2}} =$

$$8 + \frac{-7 + 8\,z^{-1}}{1 - 0.75\,z^{-1} + 0.125\,z^{-2}}$$



Parallel Form II (First Order)

- Equivalently, $H(z) = \frac{Y(z)}{X(z)} = \frac{1+2\,z^{-1}+z^{-2}}{1-0.75\,z^{-1}+0.125\,z^{-2}} =$

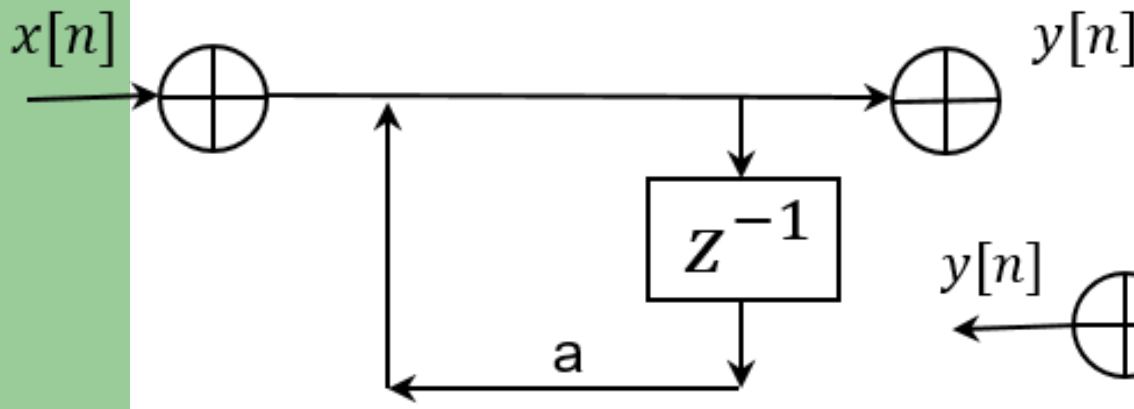


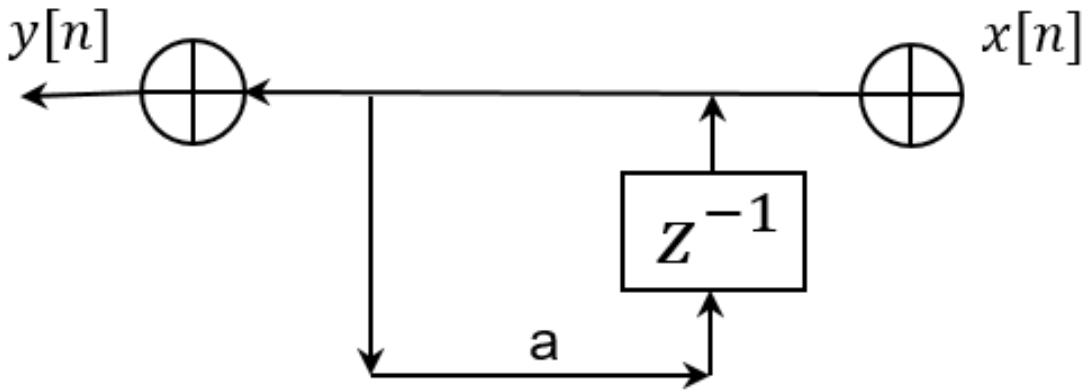
$$8 + \frac{18}{1-0.5\,z^{-1}} - \frac{25}{1-0.25\,z^{-1}}$$

Transposed Form

- Signal flow graphs can be transposed by reversing the branch arrows and swapping outputs with the inputs

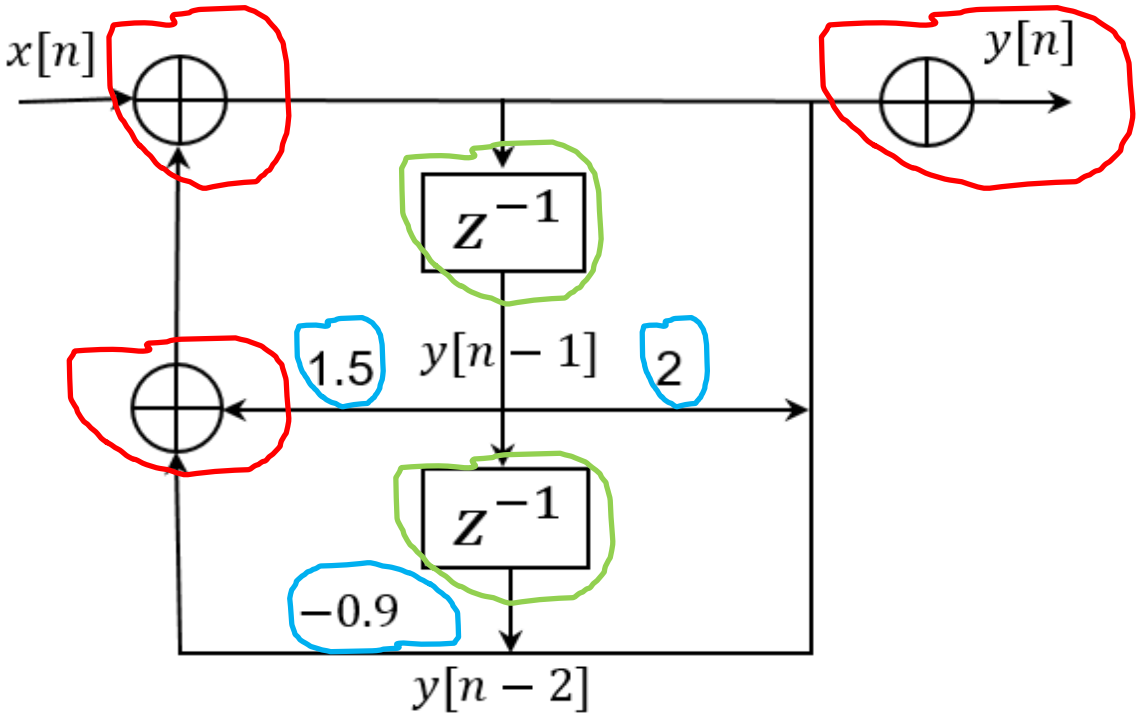
$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - a\,z^{-1}}$$





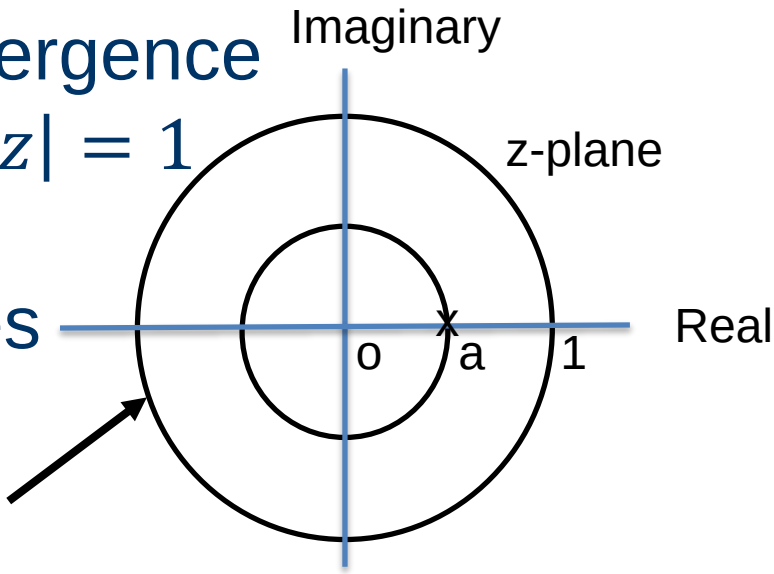
The Benefit of Restructuring

- Designer must weigh pros and cons of each structure for faster algorithm and minimizing structure elements such as number of multiplications, additions and delays (memory)



Stability and Region of Convergence

- A discrete linear time invariant system or a filter is stable if and only for every input there is an output
- This means that the impulse response is summable i.e., $\sum_{k=-\infty}^{\infty}|h[k]| < \infty$
- The all values for z of a sequence for which the z transform converges is called the region of convergence
- If ROC includes the unit circle then $|z| = 1$
- Then system is stable
- There are inner and outer boundaries
- If pole $a > 1$, system is unstable



Digital Up Conversion (DUP)

- To translate a baseband modulated signal to an Intermediate Frequency (IF)
- Aka interpolation e.g., converting a $8kHz$ to $64kHz$ by inserting samples using polynomial based interpolated values
- Run interpolation.m
- Run interpolation_FIR.m
- Upsampling is a similar concept but instead of interpolation, zeros are inserted

Versus Issue

- IIR
- Nonlinear phase causes group / phase delay (linearization techniques difficult)
- Unstable
- Low order to meet specifications
- Low delay due to low order

- FIR
- Linear phase
- Stable
- Linear design and efficient hardware realization
- Higher order needed to meet specifications
- High delay due to higher order

Scaling Hardware Resources and Complexity

Coefficients and Arithmetic Ops

- If integer, how many bits? Dynamic range
- How negative values handled with 2's complement? Overflows? Signed/Unsigned arithmetic
- If fixed point, how it is represented in hardware?
- If floating point, how does the point move, to what accuracy? IEEE 754 standard
- 32-bit single precision or 64-bit double precision
- The fixed-point implementations are widely used for signal processing systems whereas floating points are mainly used in feedback control systems where precision is of paramount importance

Floating Point Arithmetic

- The floating point format stores a number in terms of mantissa and exponent
- Hardware that supports the floating point format, after executing each computation, automatically scales the mantissa and updates the exponent to make the result fit in the required number of bits in a defined way
- All these operations make floating point HW more expensive in terms of area and power than fixed point HW (may be in computation time too)

Fixed Point Arithmetic

- A fixed point hardware, after executing a computation, does not track the position of the decimal point and leaves this responsibility to the developer
- The decimal point is fixed for each variable and is
- predefined
- By fixing the point a variable can take only a fixed range of values
- If the result of a calculation falls outside of this range the data is lost or corrupted. This is known as overflow

Fixed Point Arithmetic

- Rounding in fixed point
- The large gate densities of current generation FPGAs and ASICs allow designers to map floating point algorithms in HW if required
- This is especially true for more complex signal processing applications where keeping the numerical accuracy intact is considered critical
- However, fabric numeric resources multipliers/adders are limited
- MATLAB uses *fi()* to convert to fixed point

Qn.m format Coefficients

- Qn.m mean n bits to the left for mantissa and m bits for the exponent
- Adding an 8 bit Q7.1 format number in an 8 bit Q1.7 format number will yield a 14 bit Q7.7 format number
- Run coefficients_and_stability.m

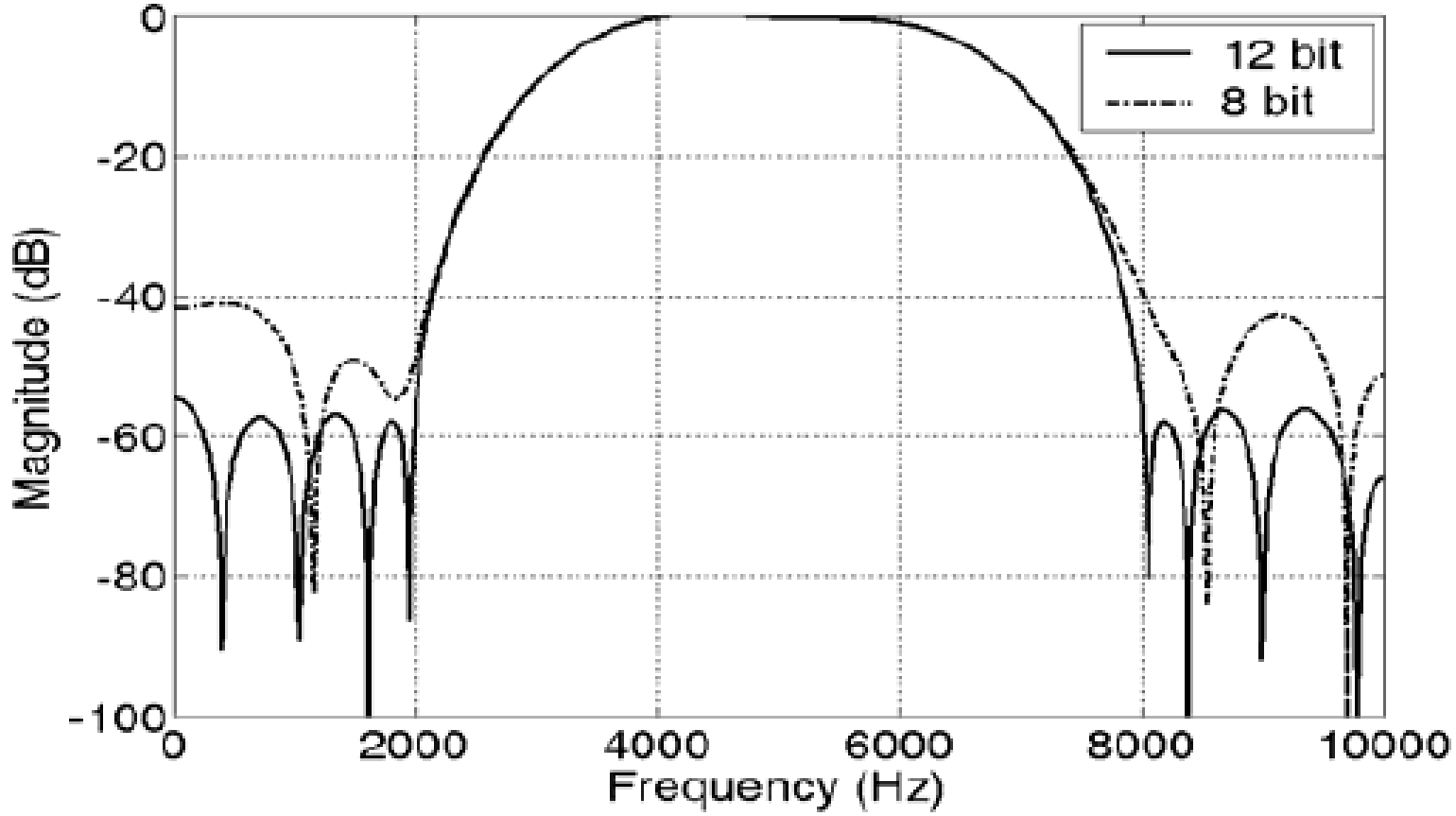
Coefficients

- Integer

Coefs	*Original*	*16-bit*	*12-bit*	*8-bit*
h(10)	0.348822	0.348822	0.348822	0.348822
h(9), h(11)	0.004010	0.004013	0.004090	0.002747
h(8), h(12)	-0.268081	-0.268076	-0.268049	-0.269170
h(7), h(13)	-0.008972	-0.008974	-0.009032	-0.008240
h(6), h(14)	0.102739	0.102740	0.102755	0.101625
h(5), h(15)	0.008322	0.008325	0.008350	0.008240
h(4), h(16)	0.012282	0.012285	0.012269	0.010987
h(3), h(17)	-0.004333	-0.004333	-0.004260	-0.005493
h(2), h(18)	-0.033368	-0.033363	-0.033400	-0.032960
h(1), h(19)	0.001195	0.001192	0.001193	0.000000
h(0), h(20)	0.012527	0.012530	0.012610	0.013733

Data Representation

- Accuracy

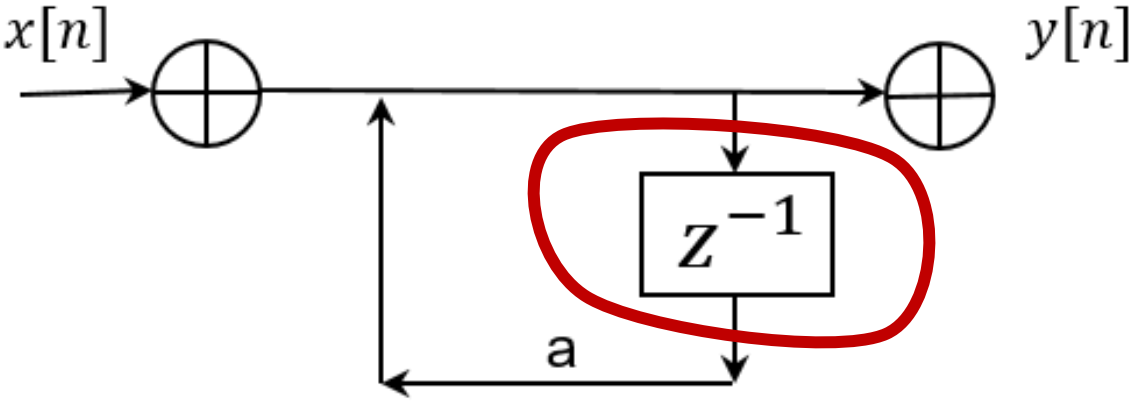


Sizing

- Minimizing order and length
- Minimum-Phase Lowpass FIR Filter
  - Run FIR_Min_Phase_LPF.m
  - Check implementation cost: Analysis → Filter Information
  - How do number of coefficients reduce?
  - How does passband attenuates?
- Minimum-Order Lowpass FIR Filter (aka Interpolation filter)
  - Run FIR_Min_Order_LPF.m

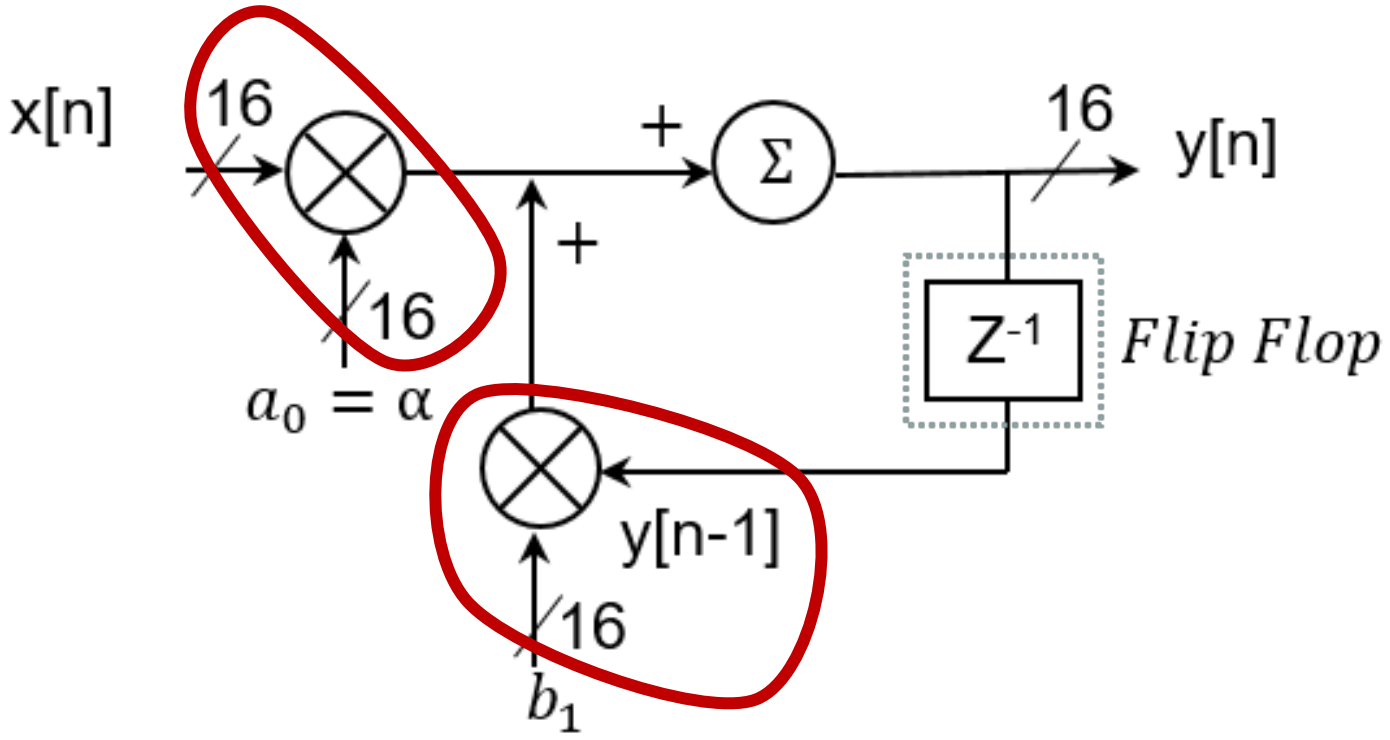
Hardware Issues

- Datapath design
- Retiming and piplining if filters are higher order
- Implementation in DSP vs. FPGA, depends on fabric resources in math ops
- No of multipliers in Altera and Xilinx fabrics, floating point unit, 32-bit or 64-bit data size
- Memory size



Hardware Issues

- Consider IIR $y[n] = b_1 y[n-1] + a_0 x[n]$
- How to handle recursive multiplications that grow with word size?



IIR with Integer Coefficients

- https://au.mathworks.com/help/dsp/ug/create-an-fir-filter-using-integer-coefficients.html