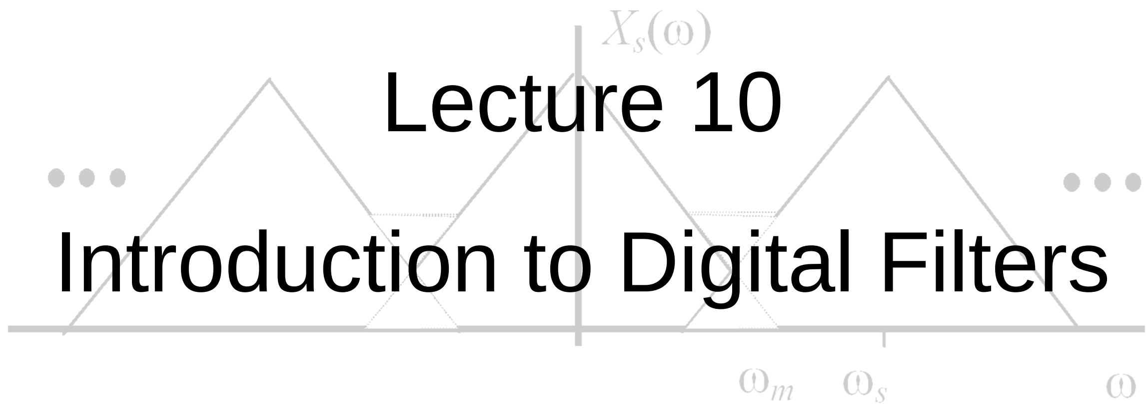


Lecture 10

Introduction to Digital Filters

Review of Basic Concepts and Definitions

Dr. Dimitris Agrafiotis, Dept. Of Electrical & Electronic Engineering

Introduction to Digital Filters

Review of Basic Concepts and Definitions

- Linearity
- Time Invariance
- Linear Time Invariant Systems
- Impulse Response
- Convolution
- Causality
- Linear Constant Coefficient Difference Equations

Time Domain

- Sinusoidal Input Response
- Frequency Response
- Z Transform
- Transfer function
- Stability

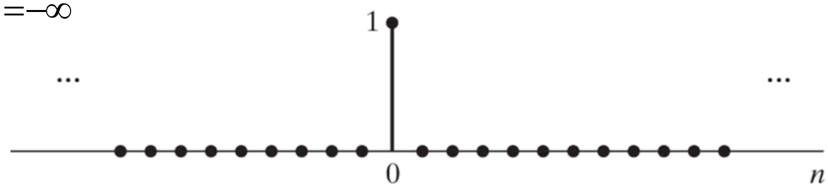
Frequency Domain

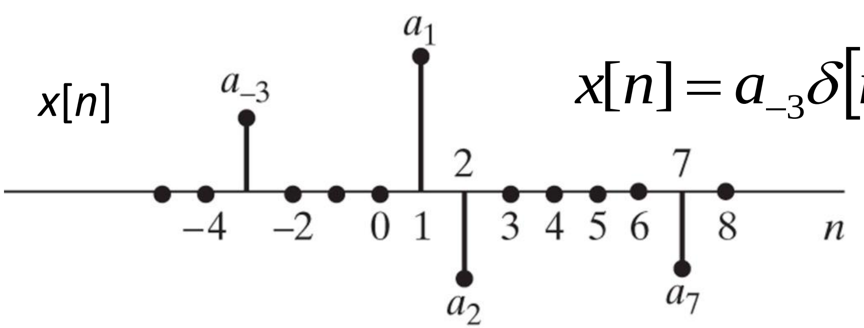
Discrete Time Signals & Systems

Discrete time signals

Sum of scaled and delayed impulses $x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]$ where

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$ is the impulse function

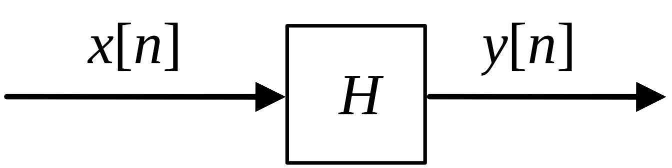




$$x[n] = a_{-3}\delta[n+3] + a_1\delta[n-1] + a_2\delta[n-2] + a_7\delta[n-7]$$

Discrete time systems

A transformation that maps an input signal with values x[n] to an output signal with values y[n]



Linearity

Linear Systems - Weighted sum of inputs gives weighted sum of outputs

A discrete time system $y[n]=h\{x[n]\}$ is linear if it obeys the <u>principle of superposition</u>:

$$h\{a_1x_1[n]+a_2x_2[n]\}=a_1h\{x_1[n]\}+a_2h\{x_2[n]\}=a_1y_1[n]+a_2y_2[n]$$

The principle of superposition consists of the <u>additivity property</u> :

$$h\{x_1[n]+x_2[n]\}=h\{x_1[n]\}+h\{x_2[n]\}=y_1[n]+y_2[n]$$

and the <u>homogeneity (or scaling) property</u> :

$$h\{ax[n]\}=ah\{x[n]\}=ay[n]$$

More general: if the input signal is $x[n]=\sum_k a_k x_k[n]$ then the output of a linear system is $y[n]=\sum_k a_k y_k[n]$ where $y_k[n]$ is the output of system *h* to input $x_k[n]$

Linearity

Example of a Linear System

(general proof required to show that a system is linear)

$$y[n] = \sum_{k=-\infty}^{n} x[k]$$

The Accumulator system: the output at time *n* is the sum of the present and all previous samples

- Define two arbitrary inputs and their associated outputs: $y_1[n] = \sum_{k=-\infty}^{n} x_1[k]$, $y_2[n] = \sum_{k=-\infty}^{n} x_2[k]$
- Check what the output $y_3[n]$ is if the input is : $x_3[n] = ax_1[n] + bx_2[n]$
- The system is linear if the superposition principle holds : $y_3[n] = ay_1[n] + by_2[n]$

$$y_3[n] = \sum_{k=-\infty}^{n} x_3[k] = \sum_{k=-\infty}^{n} (ax_1[k] + bx_2[k]) = a\sum_{k=-\infty}^{n} x_1[k] + b\sum_{k=-\infty}^{n} x_2[k] = ay_1[n] + by_2[n]$$

Example of Nonlinear Systems

(existence of counterexample enough to prove system not linear)

$y[n] = \log_{10}\{x[n]\}$

$$y[1+10] = \log_{10}[11] = 1.041 \neq y[1] + y[10] = \log_{10}[1] + \log_{10}[10] = 1$$

$y[n] = \{x[n]\}^2$

$$y[1+10] = 11^2 = 121 \neq y[1] + y[10] = 1^2 + 10^2 = 101$$

Time Invariance

Time Invariant Systems – Delaying the input causes similar delay to the output

A time shift in the input sequence causes a corresponding shift in the output sequence (also known as shift invariant systems): $h\{x[n-k]\} = y[n-k]$

Example of a Time Invariant System

$$y[n] = \sum_{k=-\infty}^{n} x[k]$$

- Define a delayed input : $x_1[n] = x[n-n_0]$
- Evaluate both $y[n-n_0]$ and $y_1[n]$ and see if they are equal

$$y[n-n_0] = \sum_{k=-\infty}^{n-n_0} x[k]$$

Change of variable $k_1 = k - n_0$

Dummy indices can have any label k, k_1

$$y_1[n] = \sum_{k=-\infty}^{n} x_1[k] = \sum_{k=-\infty}^{n} x[k-n_0] \Rightarrow y_1[n] = \sum_{k_1=-\infty}^{n-n_0} x[k_1] \Rightarrow y_1[n] = \sum_{k=-\infty}^{n-n_0} x[k] = y[n-n_0]$$

Example of a Time Variant System

$$y[n] = x[Mn]$$

Down-sampler: output is every M_{th} input sample

replace n with $n - n_0$

$$y[n-n_0] = x[M(n-n_0)] = x[Mn - Mn_0]$$

delay input by n_0

$$y_1[n] = x_1[Mn] = x[Mn-n_0] \neq y[n-n_0]$$

Linear Time Invariant (LTI) Systems – Why do we care?

What is the output of an LTI system to a discrete time signal ?

- Discrete Time Signal: linear combination of delayed impulses $x[n]=\sum_{k=-\infty}^{\infty} x[k]\delta[n-k]$
- Linear System: if the input signal is $x[n]=\sum_{k} a_k x_k[n]$ then the output is $y[n]=\sum_{k} a_k y_k[n]$
- Time Invariant System: shifting the input causes same shift to output $h\{x[n-k]\}=y[n-k]$

Let $h[n]$ be the response of an LTI system to an impulse $\delta[n]$ (1)

and $h_k[n]$ the response of the system to a delayed impulse $\delta[n-k]$ (2)

The response of the system to discrete time signal *x*[*n*] *is*: $y[n]=h\{x[n]\}=h\left\{\sum_{k=-\infty}^{\infty} x[k]\delta[n-k]\right\}$

- Due to the system being Linear: $y[n]=\sum_{k=-\infty}^{\infty} x[k]h\{\delta[n-k]\} \overset{(2)}{=} \sum_{k=-\infty}^{\infty} x[k]h_k[n]$
- Due to the system being Time Invariant: $h_k[n]=h\{\delta[n-k]\} \overset{(1),(2)}{=} h[n-k]$

- The response of the LTI system to the discrete time signal *x*[*n*] is: $y[n]=\sum_{k=-\infty}^{\infty} x[k]h[n-k]$

Linear Time Invariant (LTI) Systems – Why do we care?

What is the output of an LTI system to a discrete time signal ?

Due to linearity

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] h\{\delta[n-k]\}$$

Due to time invariance

$$h[n-k] = h\{\delta[n-k]\}$$

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] h[n-k] \Leftrightarrow y[n] = x[n] * h[n]$$

The Convolution Sum

where $h[n]$ is the response of the LTI system to impulse $\delta[n]$

Impulse response - System's output given an impulse as input : $h[n] = h\{\delta[n]\}$

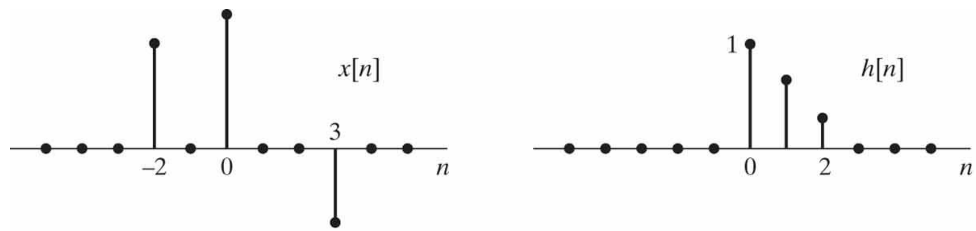
LTI systems are fully characterised by their impulse response in the time domain

The impulse response $h[n]$ of an LTI system can be used to compute the output of the system for any input via the convolution sum and check whether the system is causal and stable

The Convolution Operation

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k]$$

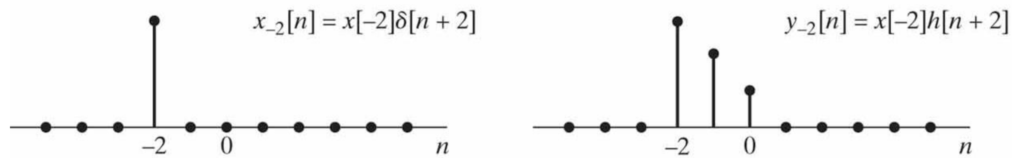
Seen as the response of an LTI system to scaled ($x[k]$) & delayed ($\delta[n-k]$) impulses



$k=-2$, Input : $x[-2]\ \delta[n+2]$

Output : $x[-2]\ h[n+2]$

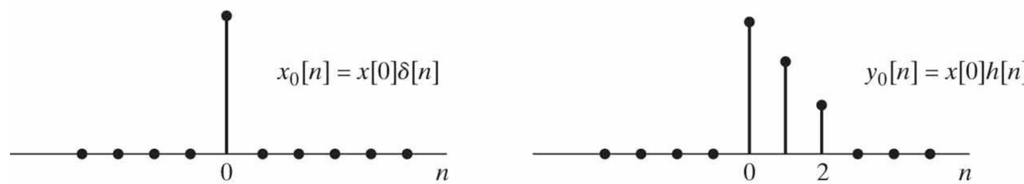
(1)



$k=0$, Input : $x[0]\ \delta[n]$

Output : $x[0]\ h[n]$

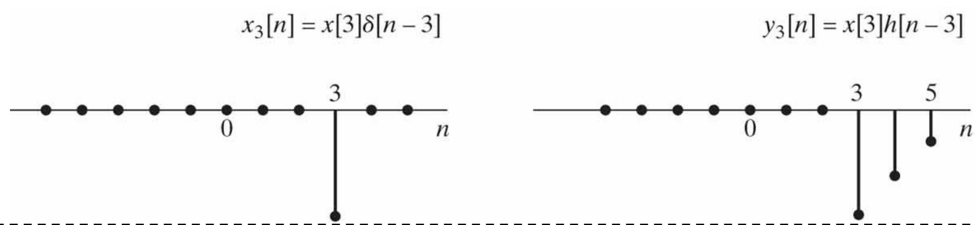
(2)



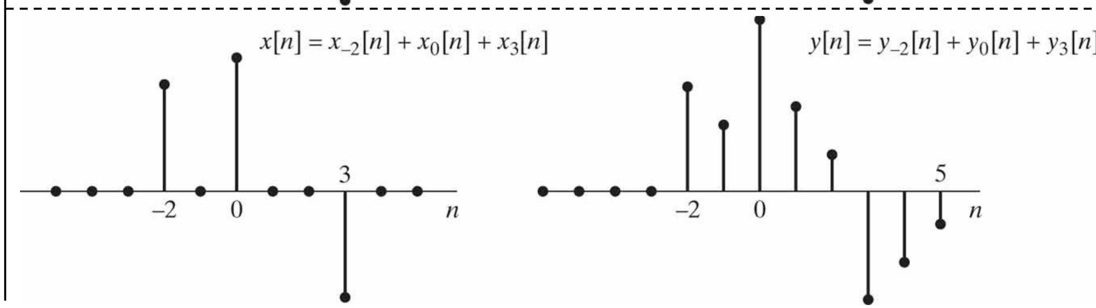
$k=3$, Input : $x[3]\ \delta[n-3]$

Output : $x[3]\ h[n-3]$

(3)



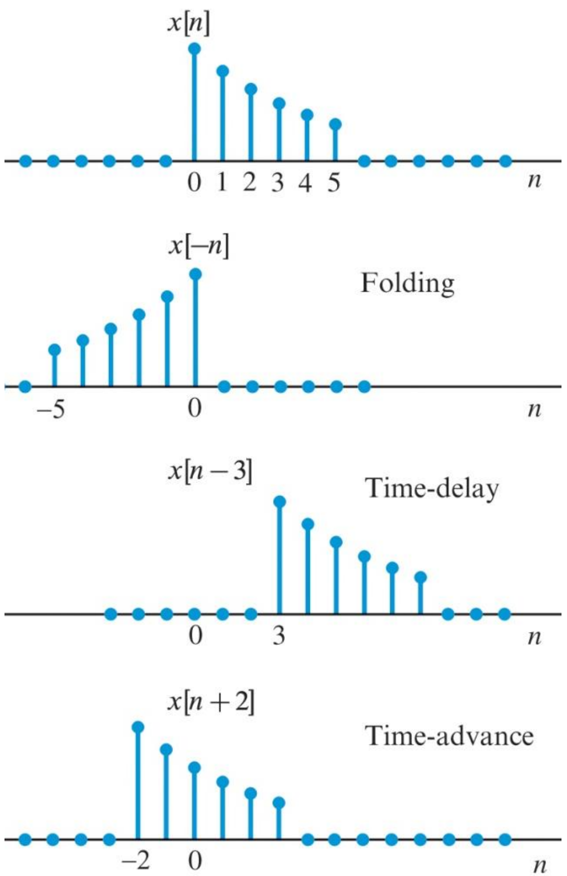
$$y[n] = (1) + (2) + (3)$$



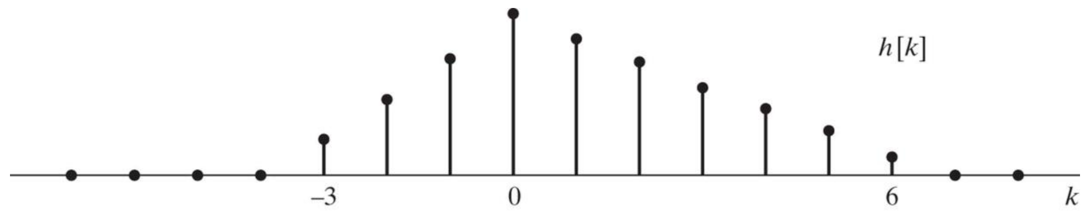
The Convolution Operation

$$y[n]=x[n]*h[n]=\sum_{k=-\infty}^{\infty}x[k]\,h[n-k]$$

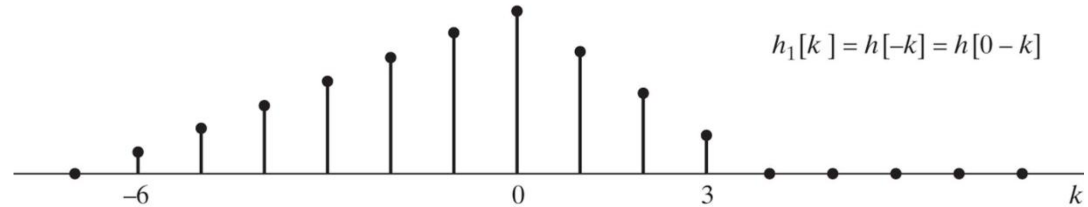
Seen as multiplication of input sequence *x*[k] with the values of sequence *h[n-k].*



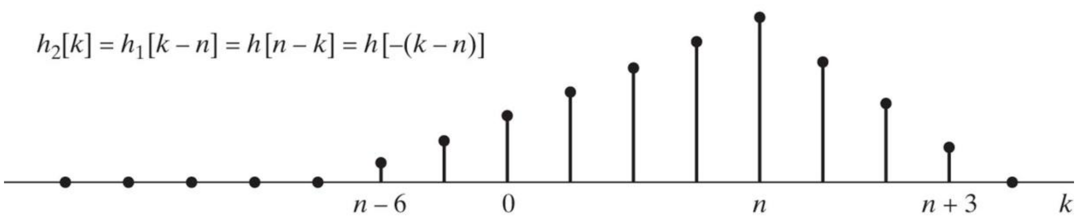
How do we form *h*[n-k] from *h*[k] ?



Fold *h*[k] (Reverse h[k] in time about *k*=0)



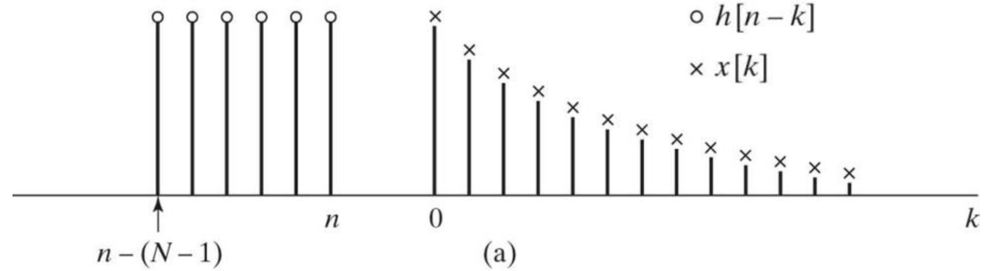
Delay folded h[k] by n samples (n=4 here)



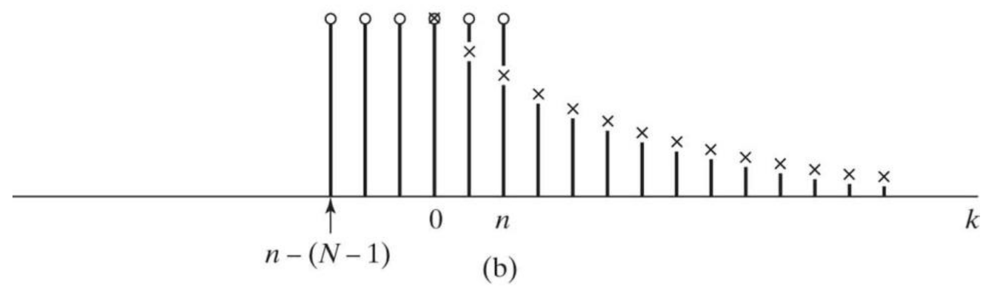
The Convolution Operation

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k]$$

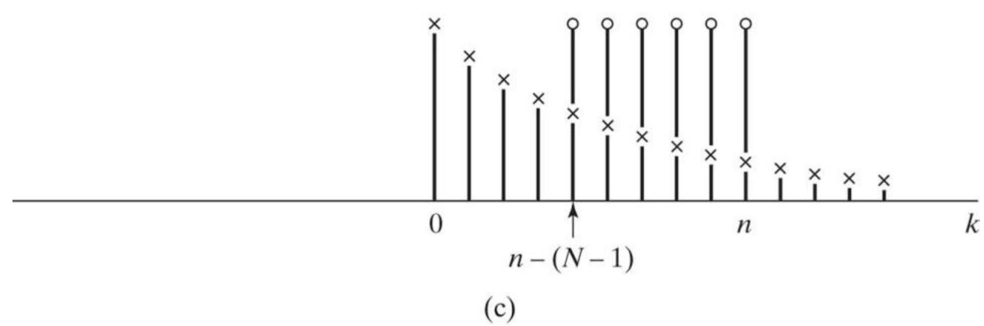
Seen as multiplication of input sequence x[k] with the values of sequence *h[n-k].*



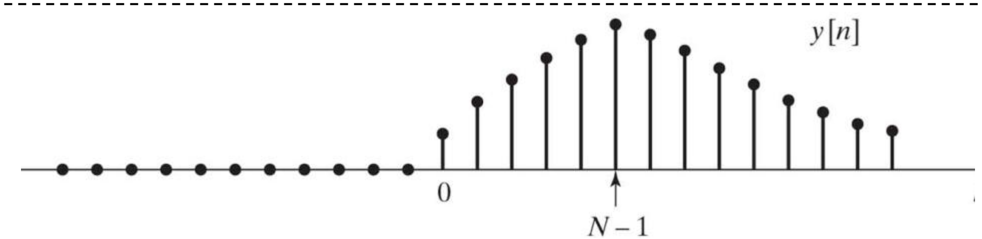
Slide *h*[-k] from left to right (delay *h*[-k] with an increasing *n*)



For each *n*-sample shift,
multiply corresponding samples
and add products



The sum of each product is one sample of the output (the n^{th} sample)



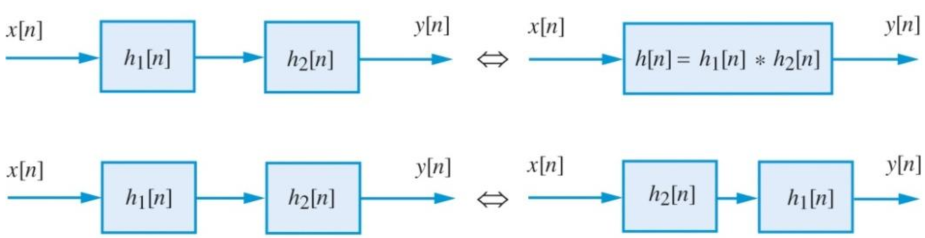
Linear Time Invariant (LTI) Systems – Properties

Convolution Properties

Convolution is Commutative

$$y[n] = x[n] * h[n] = h[n] * x[n]$$

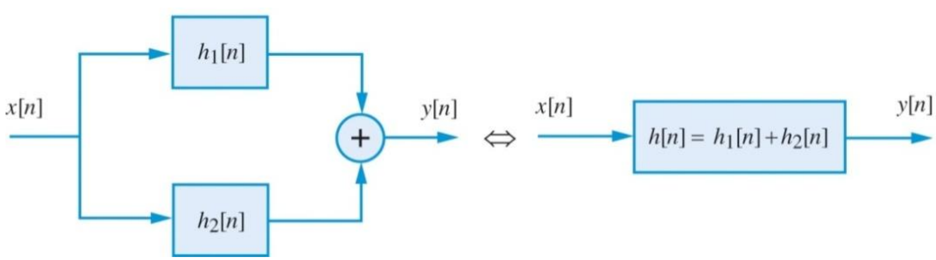
Convolution is Associative



$$(x[n] * h_1[n]) * h_2[n] = x[n] * (h_1[n] * h_2[n]) =$$

$$(x[n] * h_2[n]) * h_1[n]$$

Convolution is Distributive



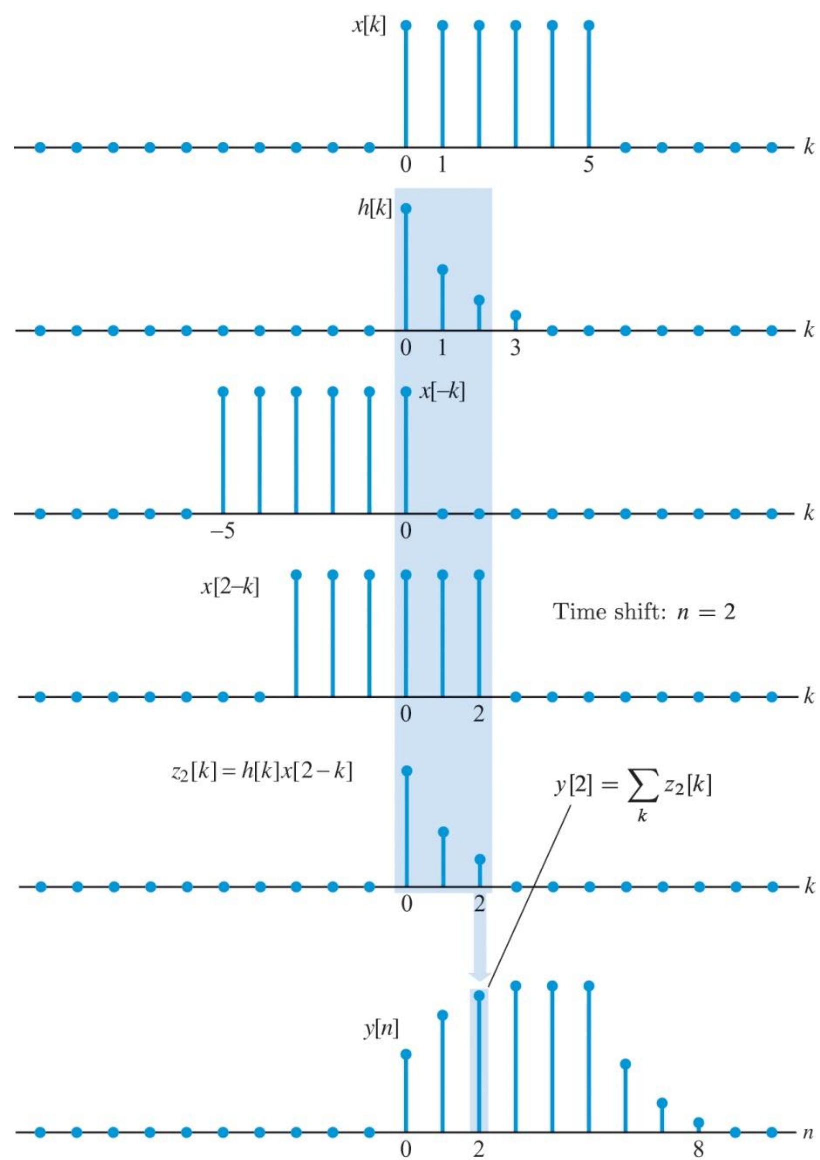
$$x[n] * (h_1[n] + h_2[n]) = x[n] * h_1[n] + x[n] * h_2[n]$$

The Convolution Operation

$$y[n] = x[n] * h[n] = h[n] * x[n] = \sum_{k=-\infty}^{\infty} h[k]\, x[n-k]$$

Convolution is Commutative

- Fold the incoming signal *x*[k] to form *x*[-k]
- Slide the folded incoming signal *x*[-k] from left to right (increase the delay *n* in *x*[n-k])
- Multiply corresponding samples
- Sum resulting products to get nth output



Digital Filters – FIR & IIR

Constant coefficient difference equations

Filter implementation usually realised through such equations

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

N : filter order

N+1 : filter length

Current output

Weighted sum of previous outputs

Weighted sum of the current and previous inputs

IIR – Infinite Impulse Response :

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

$$a_k \neq 0$$

Dependence on previous outputs (feedback) $\Rightarrow$ Unlimited (infinite) response to single impulse

FIR – Finite Impulse Response :

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

$$a_k = 0$$

No dependence on previous outputs $\Rightarrow$ Finite response (num of outputs) to single impulse

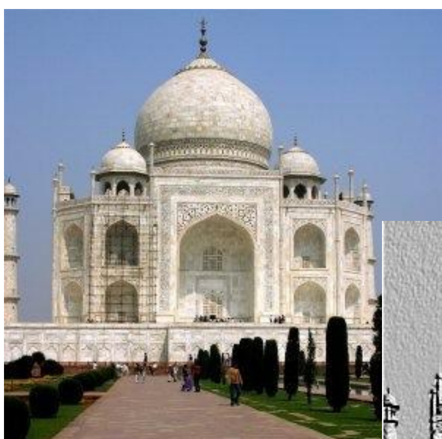
Digital Filters – FIR & IIR

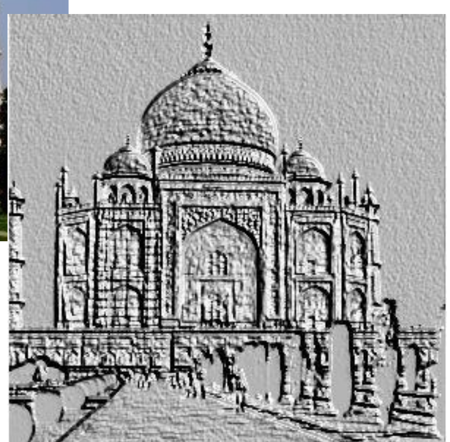
Causality

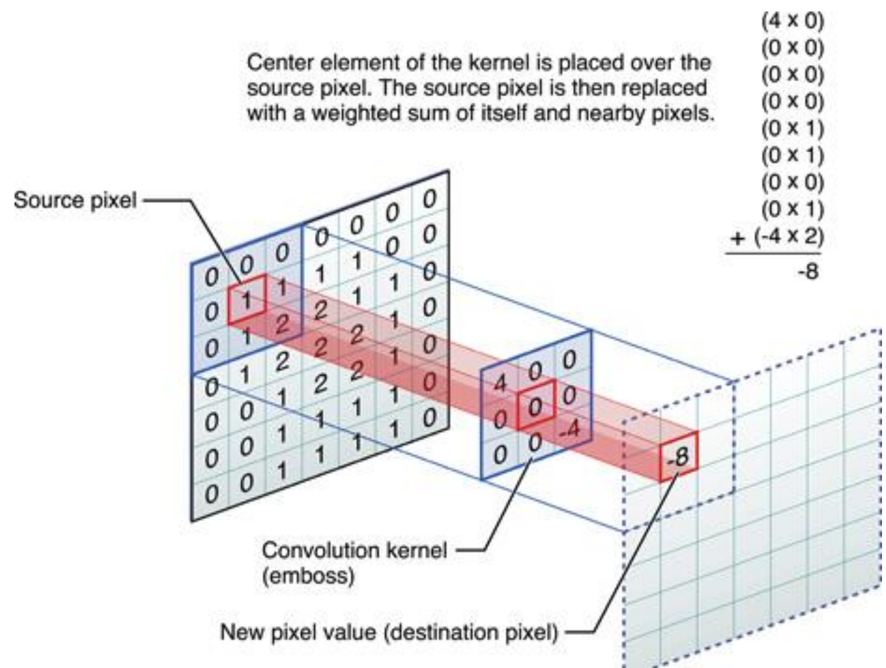
Causal FIR filter - output depends on current and past inputs : $y[n]=\sum_{k=0}^{N} b_k x[n-k]$

Non Causal FIR filter - output depends on current, past & future inputs : $y[n]=\sum_{k=-N}^{N} b_k x[n-k]$
(offline and non-real time signal processing)

Example of **Non Causal** Filters : Image Processing (Spatial Filtering)







LTI Systems –Frequency Domain Representation

Response of LTI systems to sinusoidal input

- LTI system input : complex exponential $x[n]=e^{j\Omega n}\ ,\ -\infty<n<\infty$

- LTI system output given by convolution

$$y[n]=\sum_{k=-\infty}^{\infty}x[k]h[n-k]=\sum_{k=-\infty}^{\infty}h[k]x[n-k]=\sum_{k=-\infty}^{\infty}h[k]e^{j\Omega(n-k)}=\underbrace{\sum_{k=-\infty}^{\infty}h[k]e^{-j\Omega k}}_{H(\Omega)}\,e^{j\Omega n}$$

$$y[n]=\underbrace{H(\Omega)}_{\text{Eigenvalue}}\ \underbrace{e^{j\Omega n}}_{\text{Eigenfunction}}$$

Eigenfunction

Eigenvalue

System output to complex exponential is complex exponential with altered magnitude and phase

- LTI system input : linear combination of complex exponentials $x[n]=\sum_{k}a_k e^{j\Omega_k n}$

- LTI system output due to linearity principle : $y[n]=\sum_{k}a_k\,H(\Omega_k)\,e^{j\Omega_k n}$

LTI Systems –Frequency Domain Representation

Frequency response - Fourier transform of the impulse response

$$h[n] = h\{\delta[n]\} \Rightarrow H(\Omega) = \sum_{n=-\infty}^{\infty} h[n] e^{-j\Omega n}$$

Convolution in time ⇔ Multiplication in frequency

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] h[n-k] \Leftrightarrow Y(\Omega) = H(\Omega) X(\Omega)$$

Z Domain representation – preferred notation for filter design & analysis

$$\left.\begin{aligned} z &= re^{j\Omega} \\ H(z) &= \sum_{n=-\infty}^{\infty} h[n] z^{-n} \end{aligned}\right\} \quad Y(\Omega) = H(\Omega) X(\Omega) \overset{r=1}{\Leftrightarrow} Y(Z) = H(Z) X(Z)$$

Filters : LTI systems that modify certain frequencies relative to others

Digital Filters and Z Transforms

The Delay operator (time shifting property)

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

Bilateral (2 sided) Z transform

Causal systems / signals (x[n]=0 for n<0)

$$X(z) = \sum_{n=0}^{\infty} x[n] z^{-n} \quad or \quad X(z) = \sum_{n=0}^{\infty} x_n z^{-n}$$

One sided Z transform

-(n-k)-k=-n+k-k=-n

$$\boxed{Z\{x[n-k]\}} = \sum_{n=-\infty}^{\infty} x[n-k] z^{-n} = \sum_{n=-\infty}^{\infty} x[n-k] z^{-(n-k)} z^{-k} = z^{-k} \sum_{n'=-\infty}^{\infty} x[n'] z^{-n'} = \boxed{z^{-k} X(z)}$$

n′ = n-k

---

z^{-k} : delay operator that delays a signal by *k* samples

Digital Filters and Z Transforms

General form and transfer function

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

General form of IIR filter in the time domain

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

General form of IIR filter in Z transform domain

$$Y(z)+\sum_{k=1}^{N}a_k z^{-k}Y(z)=\sum_{k=0}^{N}b_k z^{-k}X(z)\overset{\alpha_0=1}{\Rightarrow}\sum_{k=0}^{N}a_k z^{-k}Y(z)=\sum_{k=0}^{N}b_k z^{-k}X(z)\Rightarrow$$

$$Y(z)\sum_{k=0}^{N}a_k z^{-k}=X(z)\sum_{k=0}^{N}b_k z^{-k}\Rightarrow$$

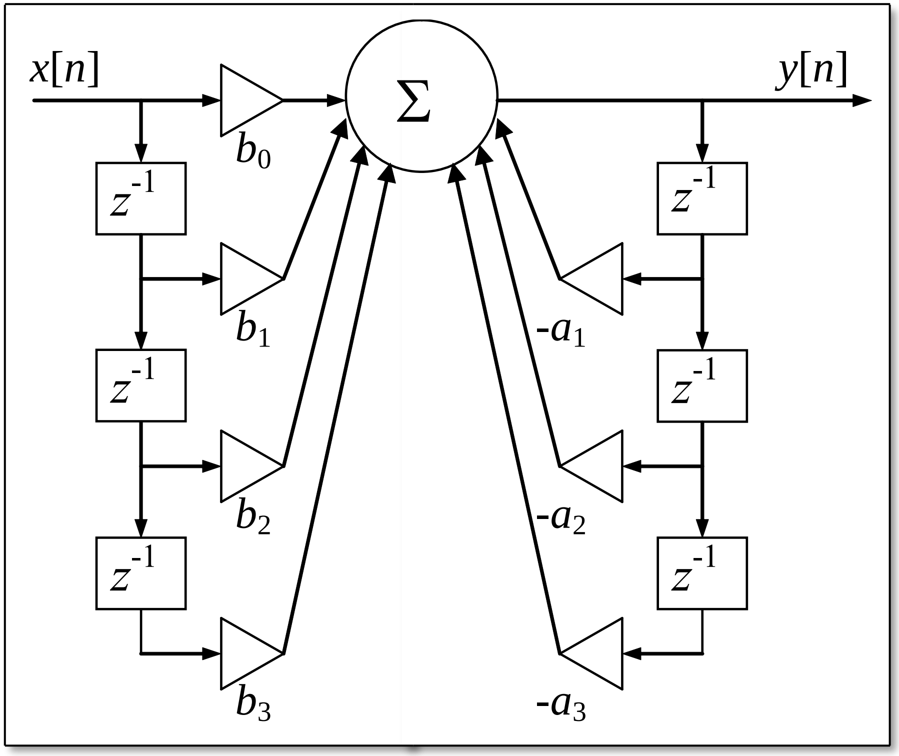
$$H(z)=\frac{Y(z)}{X(z)}=\frac{\sum_{k=0}^{N}b_k z^{-k}}{\sum_{k=0}^{N}a_k z^{-k}}\ with\ a_0=1$$

General form of IIR filter expressed as transfer function

Digital Filters and Z Transforms

General IIR digital filter implementation form (Direct Form I)

FIR Filter



IIR Filter (Feedback)

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} \; with \; a_0 = 1 \qquad \Leftrightarrow \qquad y[n] = \sum_{k=0}^{N} b_k x[n-k] - \sum_{k=1}^{N} a_k y[n-k]$$

Digital Filters and Z Transforms

Factored form of IIR filters

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} \; with\, a_0 = 1 \quad \overset{\text{factorization}}{\Leftrightarrow} \quad H(z) = b_0 \frac{\prod_{k=1}^{N} (1 - z_k z^{-1})}{\prod_{k=1}^{N} (1 - p_k z^{-1})} = b_0 \frac{\prod_{k=1}^{N} (z - z_k)}{\prod_{k=1}^{N} (z - p_k)}$$

- z_k (zeros) - values for which the response = 0
- p_k (poles) - values for which the response = ∞
- Longer filters can be built as a cascade of simpler components

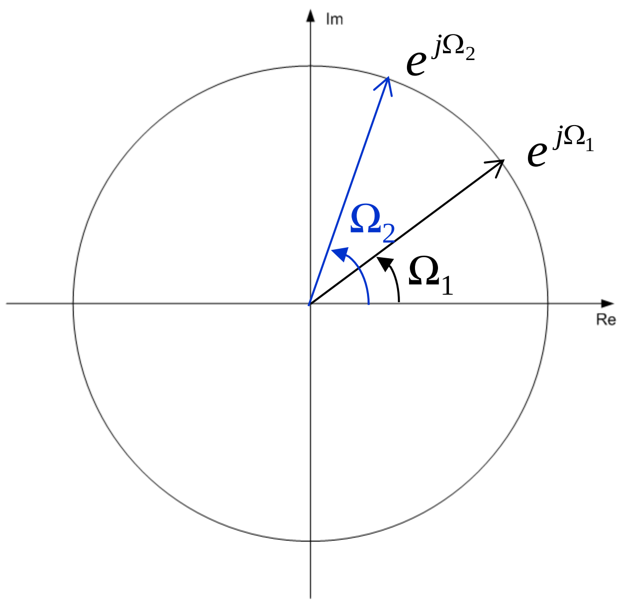
$$H(z) = H_1(z)H_2(z) = H_2(z)H_1(z)$$

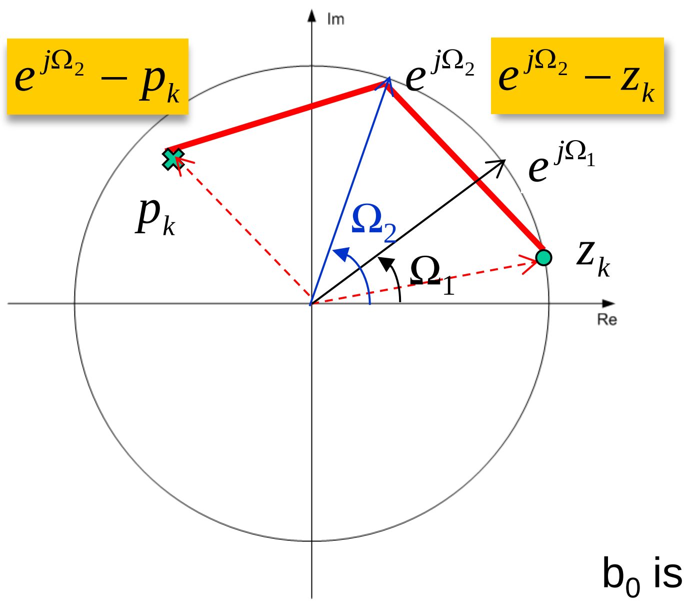
$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} \; with\, a_0 = 1 \quad \Leftrightarrow \quad y[n] = \sum_{k=0}^{N} b_k x[n-k] - \sum_{k=1}^{N} a_k y[n-k]$$

Frequency Response of Digital Filters

DTFT : Z transform evaluated on the unit circle ($z=e^{j\Omega}$)

$$H(e^{j\Omega}) = b_0 \frac{\prod_{k=1}^{N}(1-z_k e^{-j\Omega})}{\prod_{k=1}^{N}(1-p_k e^{-j\Omega})} = b_0 \frac{\prod_{k=1}^{N}(e^{j\Omega}-z_k)}{\prod_{k=1}^{N}(e^{j\Omega}-p_k)} \quad \overset{r=1}{\Longleftrightarrow} \quad H(z) = b_0 \frac{\prod_{k=1}^{N}(1-z_k z^{-1})}{\prod_{k=1}^{N}(1-p_k z^{-1})} = b_0 \frac{\prod_{k=1}^{N}(z-z_k)}{\prod_{k=1}^{N}(z-p_k)}$$





b_0 is gain term

Introduction to Digital Filters

Stability of Digital Filters

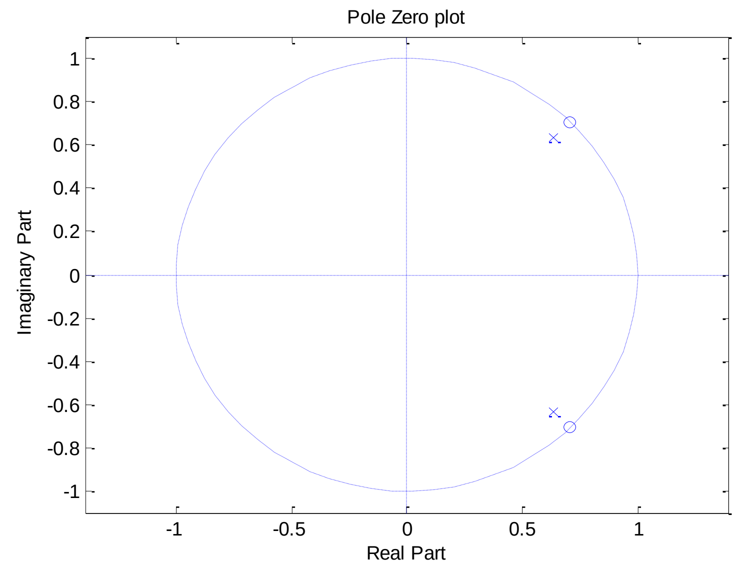
1st Order IIR system	$y[n] = x[n] + p_1 y[n-1],$
Transfer function	$H(z) = \dfrac{1}{1 - p_1 z^{-1}}$
Apply impulse at the input	$x[n] = \delta(n)$
Output	$y[n] = p_1^n \;\; for\; n \geq 0$
$p_1 > 1$	Output grows exponentially
$p_1 \leq 1$	Output decays exponentially

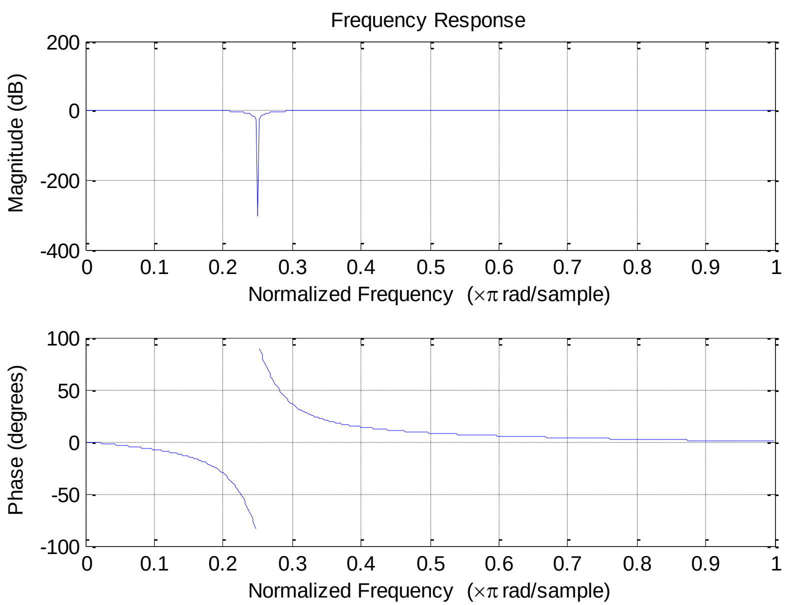
For stability poles must be inside unit circle

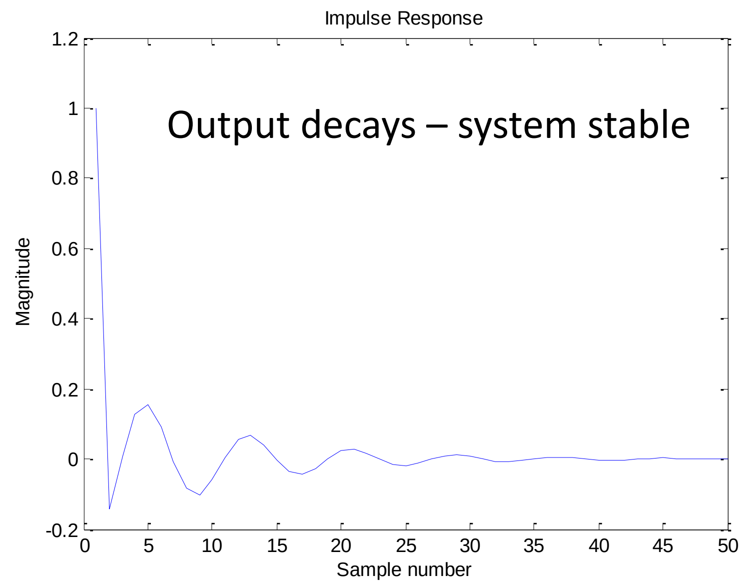
Stability of Digital Filters – Stable Filter Example

$$H(z)=\frac{1-\sqrt{2}z^{-1}+z^{-2}}{1-0.9\sqrt{2}z^{-1}+0.81z^{-2}}$$

$$=\frac{\left(1-e^{\frac{j\pi}{4}}z^{-1}\right)\left(1-e^{\frac{-j\pi}{4}}z^{-1}\right)}{\left(1-0.9e^{\frac{j\pi}{4}}z^{-1}\right)\left(1-0.9e^{\frac{-j\pi}{4}}z^{-1}\right)}$$



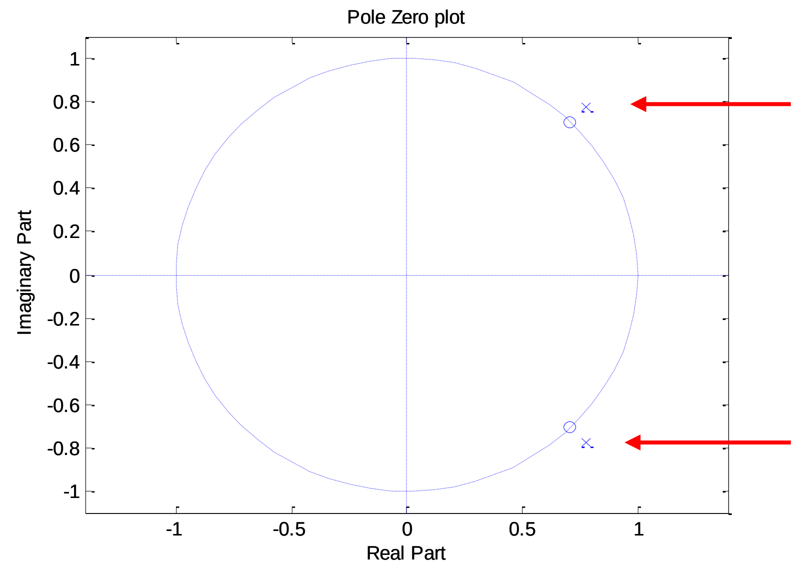


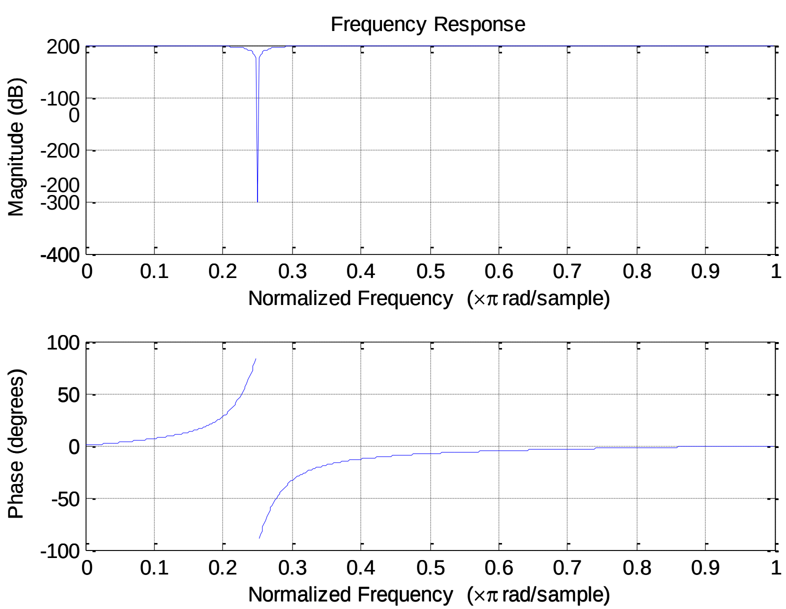


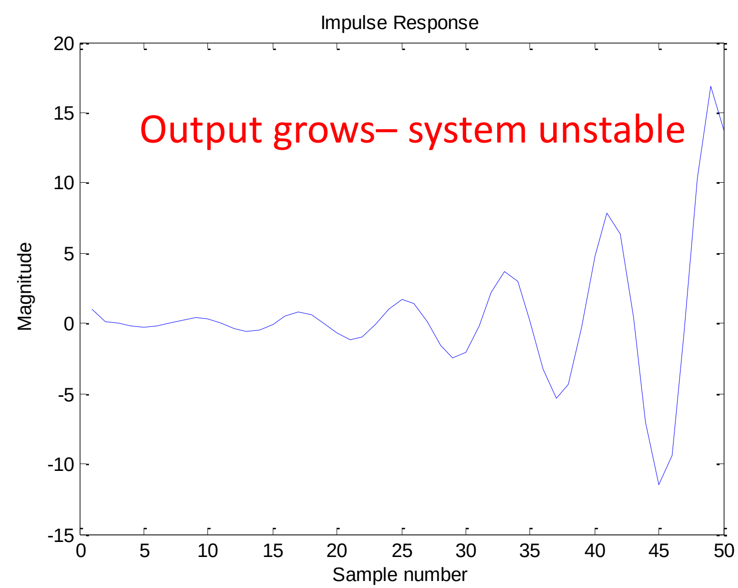
Stability of Digital Filters – Unstable Filter Example

$$H(z)=\frac{1-\sqrt{2}z^{-1}+z^{-2}}{1-\boxed{1.1}\sqrt{2}z^{-1}+\boxed{1.21}z^{-2}}$$

$$=\frac{\left(1-e^{\frac{j\pi}{4}}z^{-1}\right)\left(1-e^{\frac{-j\pi}{4}}z^{-1}\right)}{\left(1-1.1e^{\frac{j\pi}{4}}z^{-1}\right)\left(1-1.1e^{\frac{-j\pi}{4}}z^{-1}\right)}$$







Matlab

```
b1 = [1, -sqrt(2), 1]
a1 = [1, -0.9*sqrt(2), 0.81]
b2 = b1
a2 = [1, -1.1*sqrt(2), 1.21]
plot_filter(b1,a1);
pause
plot_filter(b2,a2);
```

Filter coefficients

$$H(z) = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}}$$

```
function plot_filter(b, a);
figure(1)
zplane(b,a);
title('Pole Zero plot');

figure(2)
freqz(b,a);
title('Frequency Response');

x = zeros(50,1);
x(1) = 1;
y = filter(b,a,x);
figure(3);
plot(y);
ylabel('Magnitude');
xlabel('Sample number');
title('Impulse Response');
```

zplane(b,a); → Pole zero plot

freqz(b,a); → Frequency response (magnitude & phase)

y = filter(b,a,x); → Impulse response (note x is impulse)