Digital Filtering

Prof Gelman

Introduction

- **Digital filtering is an important function that can be implemented in the DSP unit**
- **We use the term *filter* to describe a linear system used to perform frequency-selective filtering**
- **The mathematical foundation of filtering is the convolution**

Analogue and Digital Filters

- **Digital filters are now extremely cheap and their use is usually preferred over analogue filters for the following main reasons:**

- **Digital filters are programmable**

- **They do not suffer from the aging distortion**

- **It is extremely difficult to realize RLC analogue filters with low passband edge frequencies and good stopband attenuation; however, it is possible to realize digital filters with very low passband edge frequencies which provide good stopband attenuation**

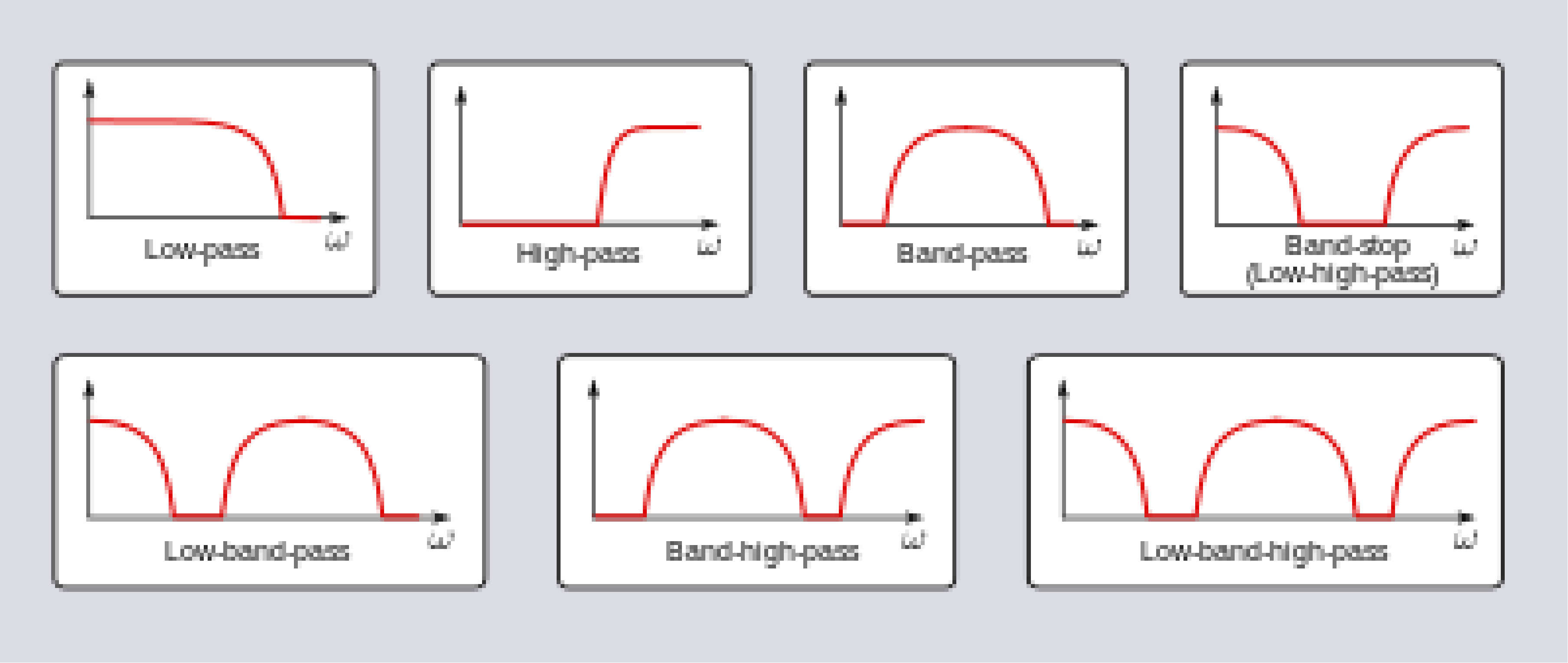
Filter Classification

Filters are usually classified according to their frequency-domain characteristics as

- **lowpass, passes only low frequencies**
- **highpass, passes only high frequencies**
- **bandpass, passes only frequencies within the selected band**
- **bandstop (or band-elimination filters), eliminates frequencies within the selected band**

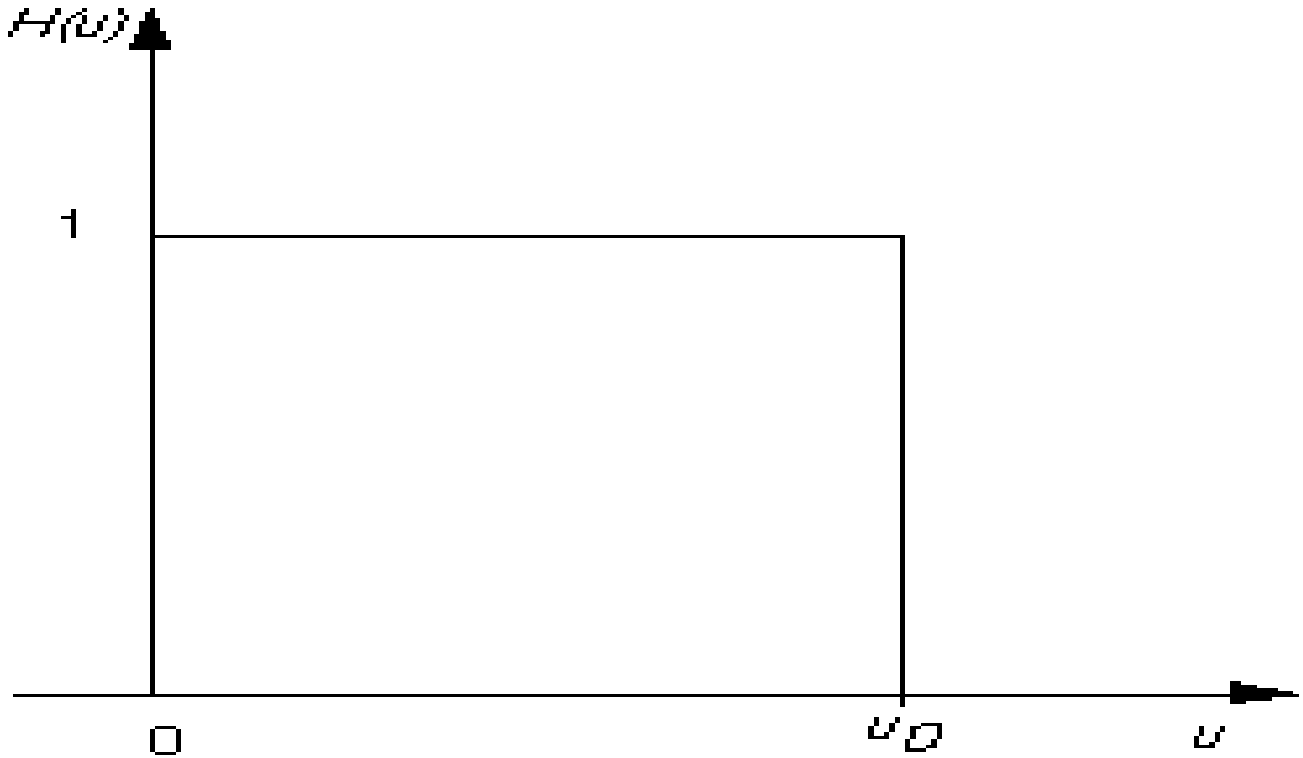
- **The ideal filters have a constant-gain (usually taken as unity-gain) in their passband and zero gain in their stopband**

Filter Classification



The Ideal Filter

The ideal lowpass filter is shown below



The Ideal Filter

Ideal filters are physically unrealizable because

- **the frequency response of the filter *cannot be zero* except at a finite set of points in the frequency range**

- **frequency response cannot have an infinitely sharp cutoff from passband to stopband**

- **In addition, the frequency response characteristics possessed by ideal filters are not absolutely necessary in most practical applications**

The Ideal and Real Filters

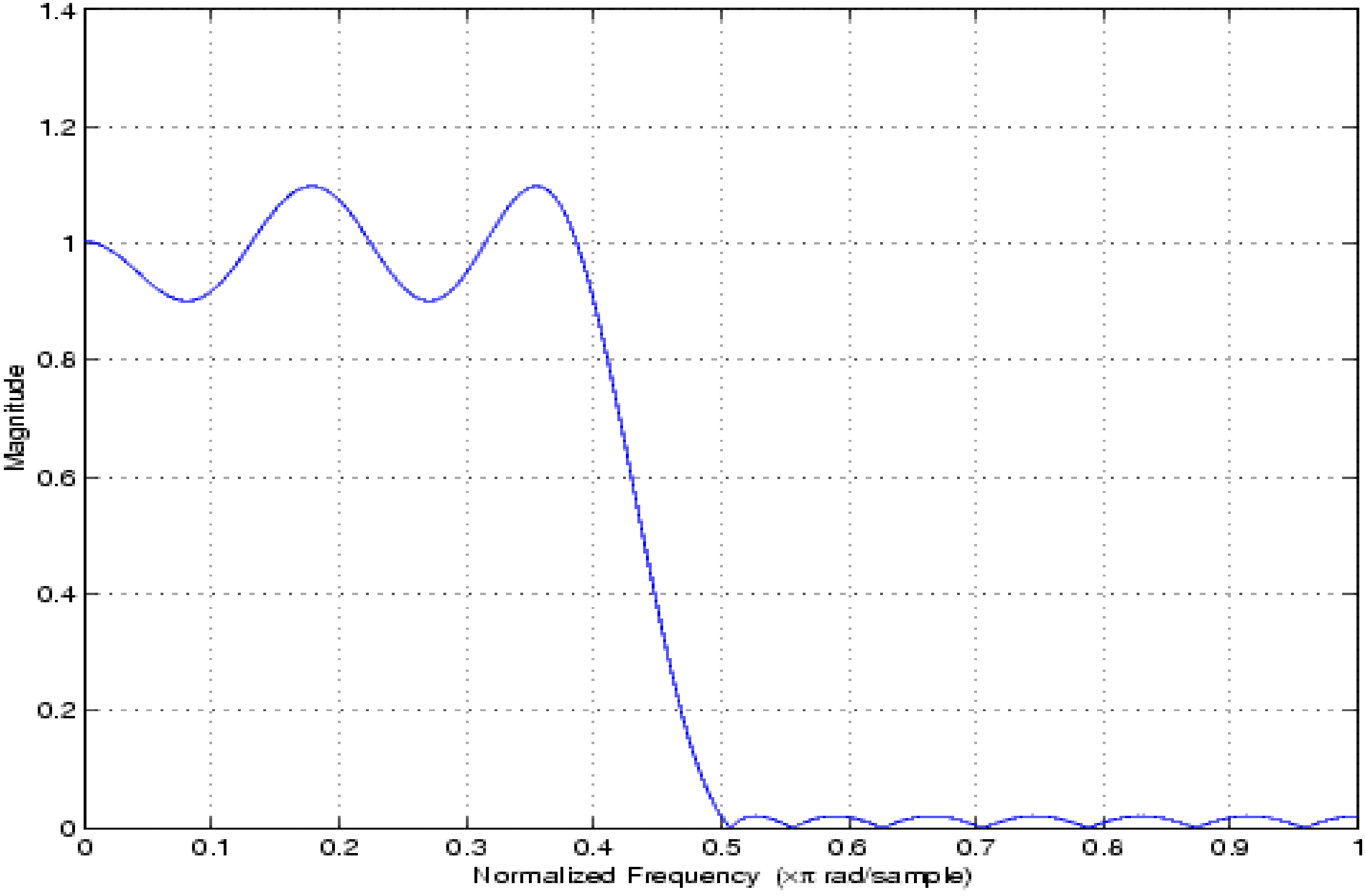
- **In particularly, it is not necessary that the magnitude of the frequency response to be constant in the entire passband**

- **Similarly, it is not necessary for the magnitude to be zero in the entire stopband**

- **A small amount of ripples is usually tolerable**

Basic Lowpass Filter

The normalized magnitude frequency response of the basic non-ideal lowpass filter is as follows



Lowpass Filter: the Passband Frequency

- **The passband edge frequency $\overline{\omega}_p$ defines the edge of the passband**

- **If there is ripple in the passband, its value is denoted as δ_1**

- **Thus, we require that in the passband the magnitude approximates unity with an error of $\pm\delta_1$, e.g.**

$$1-\delta_1 \le \left|H(\omega)\right| \le 1+\delta_1 \quad for \quad \left|\omega\right| \le \overline{\omega}_p$$

Lowpass Filter: the Stopband Frequency

- **The stopband edge frequency $\overline{\omega}_s$ denotes the beginning of the stopband**
- **The ripple in the stopband is denoted as δ_2**
- **Thus, we require that in the stopband the magnitude approximates zero with an error of $\pm\delta_2$, e.g.**

$$\left|H(\omega)\right| \le \delta_2 \;\; for \; \left|\omega\right| \ge \overline{\omega}_s$$

Lowpass Filter: the Transition Band

- **The transition of the frequency response from passband to stopband defines the *transition band* of the filter**

- **Thus, the width of the transition band is** $\overline{\omega}_s - \overline{\omega}_p$

- **Since all filter design techniques are developed in terms of *normalized frequencies*, the specified frequencies need to be normalized by sampling frequency** f_s

$$\omega_p = \frac{\overline{\omega}_p}{f_s} \qquad \omega_s = \frac{\overline{\omega}_s}{f_s}$$

Lowpass Filter: a Specification

- ***In any filter design, we need to specify*** $\delta_1, \delta_2, \omega_p$ **and** ω_s

- **Based on this specification, we can design a filter**

Lowpass Filter

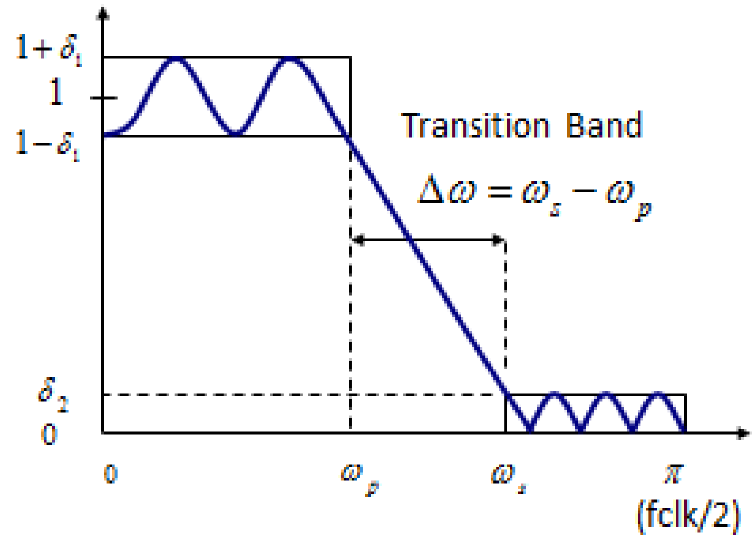
Filter Specification

δ_1 Peak Passband Ripple: $-20\log_{10}(1-\delta_1)$

δ_2 Peak Stopband Ripple $-20\log_{10}(\delta_2)$

ω_p Passband edge frequency

ω_s Stopband edge frequency



The Transfer Function

- **The *transfer function* of a discrete-time filter is defined as**

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

where $Y(z)$ **denotes the *z*-transform of the filter output signal , and** $X(z)$ **denotes the *z*- transform of the filter input signal**

- **The transfer function is the z- transform of the filter impulse response function** $h(n)$

- **Filter impulse response function is a filter response to the unit impulse excitation**

The Frequency Response Function

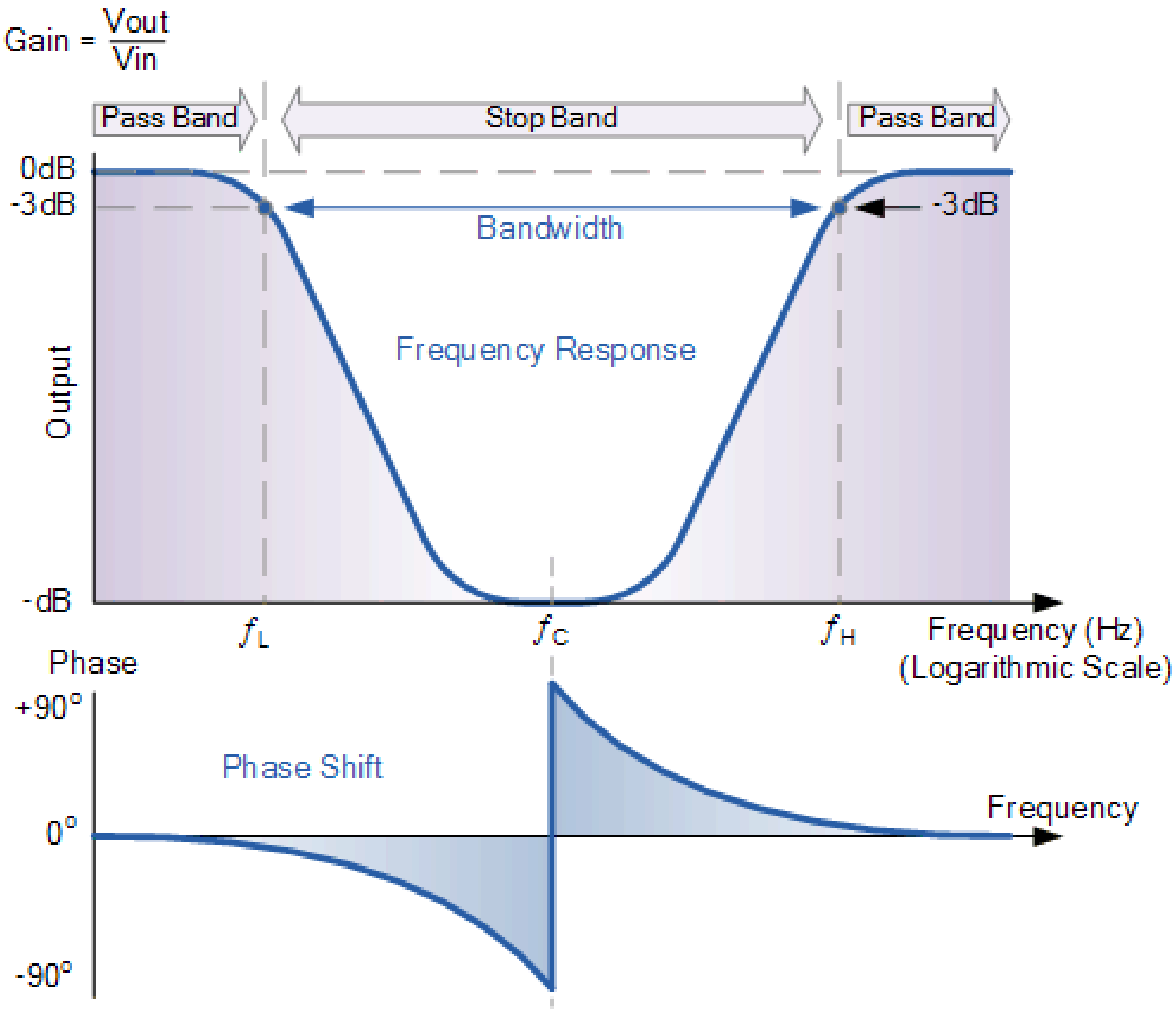
- **We can obtain the frequency response function (generally, we shorten this to the *frequency response*) of the filter by evaluating the transfer function on the unit circle**

- **Thus**

$$H(\omega) = H(z)\Big|_{z=e^{i\omega}}$$

- **Frequency response is the ratio of the complex Fourier transform of filter output to the Fourier transform of filter input**

Lowpass Filter



Linear Phase Filters

- **Let's consider a filter with a linear phase in the frequency response:**

$$H(\omega) = \begin{cases} Ae^{-i\omega\alpha} & \omega_1 \le \omega \le \omega_2 \\ 0, & otherwise \end{cases}$$

where A **and** α **are constants**

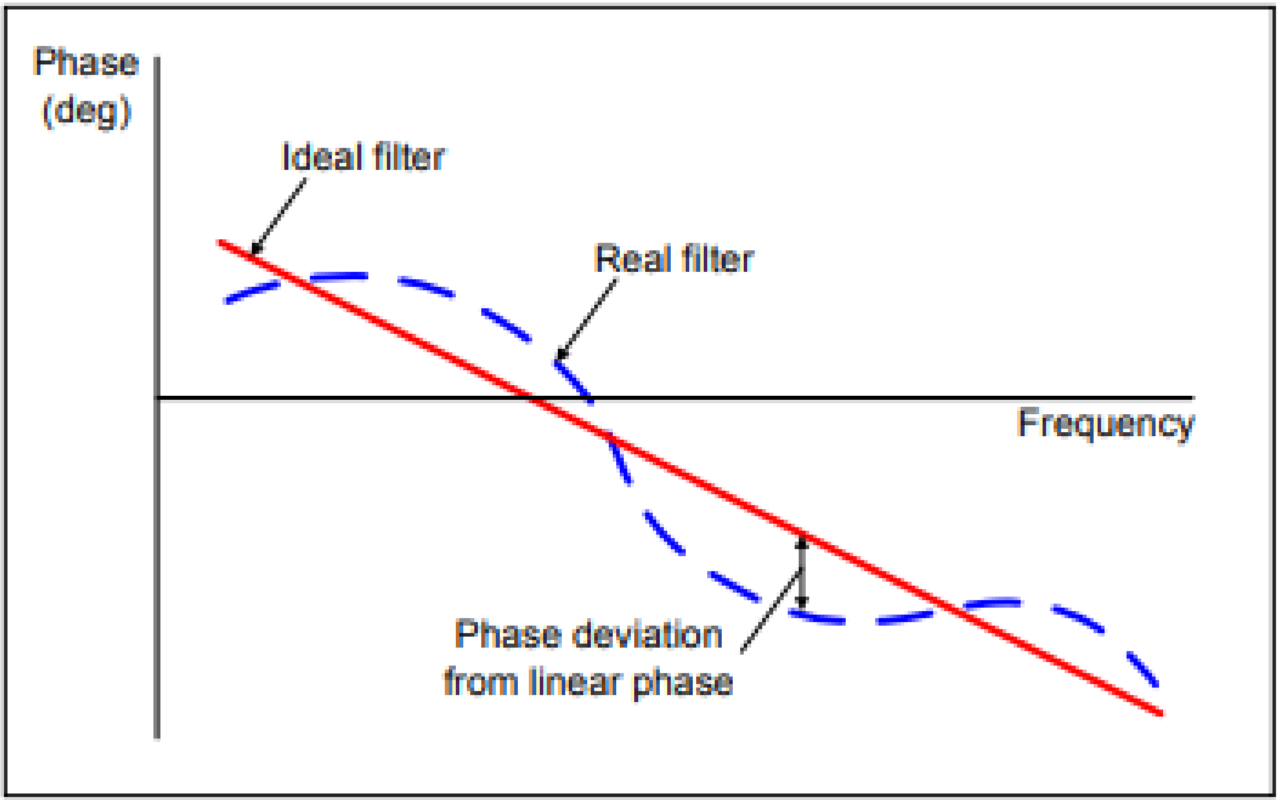
- **The Fourier transform of the output of the filter is**

$$Y(\omega) = X(\omega)H(\omega) = AX(\omega)e^{-i\omega\alpha}$$

- **Applying the scaling and time-shifting properties of the Fourier transform, we obtain the time-domain output**

$$y(n) = Ax(n-\alpha)$$

Linear Phase Filter



Linear Phase Filters

- **The filter output is simply a delayed and amplitude-scaled version of the input signal**

- **A pure delay is usually tolerable and is not considered a distortion of the signal. Neither is amplitude scaling**

- **Therefore, ideal filters have a constant magnitude characteristic and a linear phase characteristic within their passband**

- **In all cases, such filters are not physically realizable, but serve as a mathematical idealisation of practical filters**

Filter Representation

- **A digital filter can be described in two main ways**
- **First, a digital filter can be described as a mathematical algorithm or constant-coefficient difference equation, e. g.**

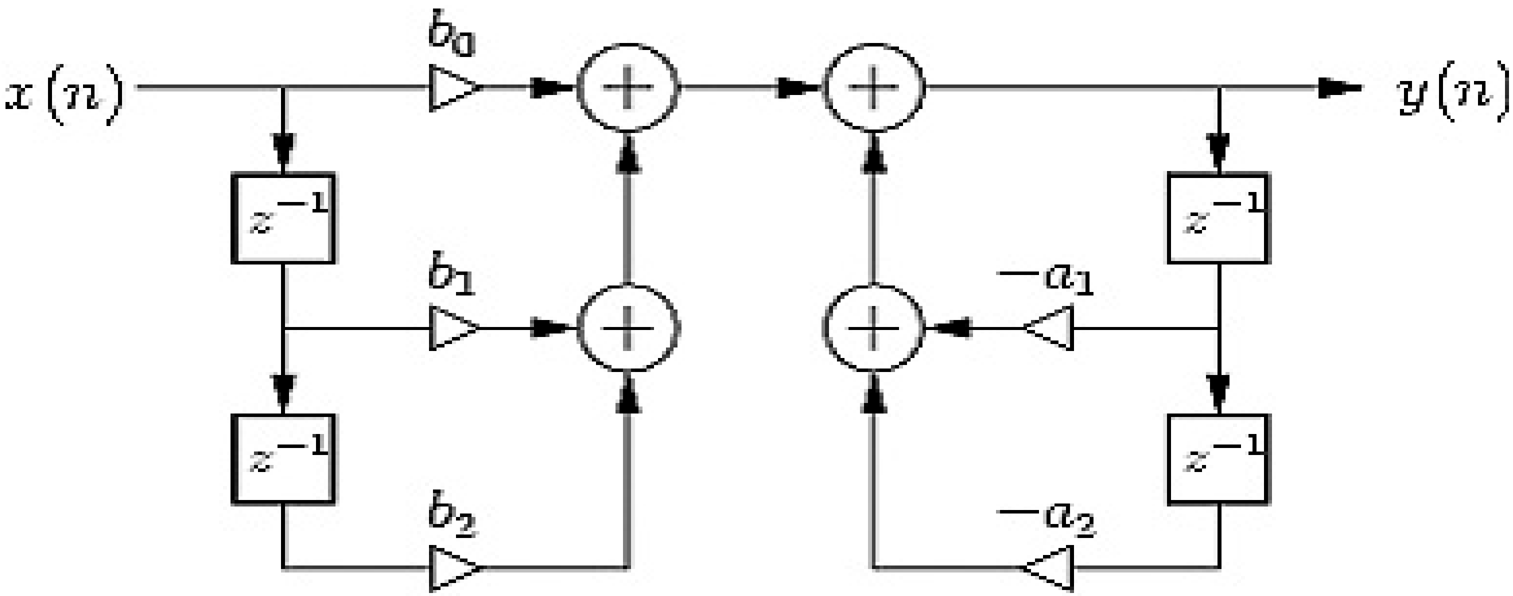
$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) - a_1 y(n-1) - a_2 y(n-2)$$

- **The properties of this particular filter will be entirely described by the constants** a_1, a_2, b_0, b_1, b_2

Filter Representation

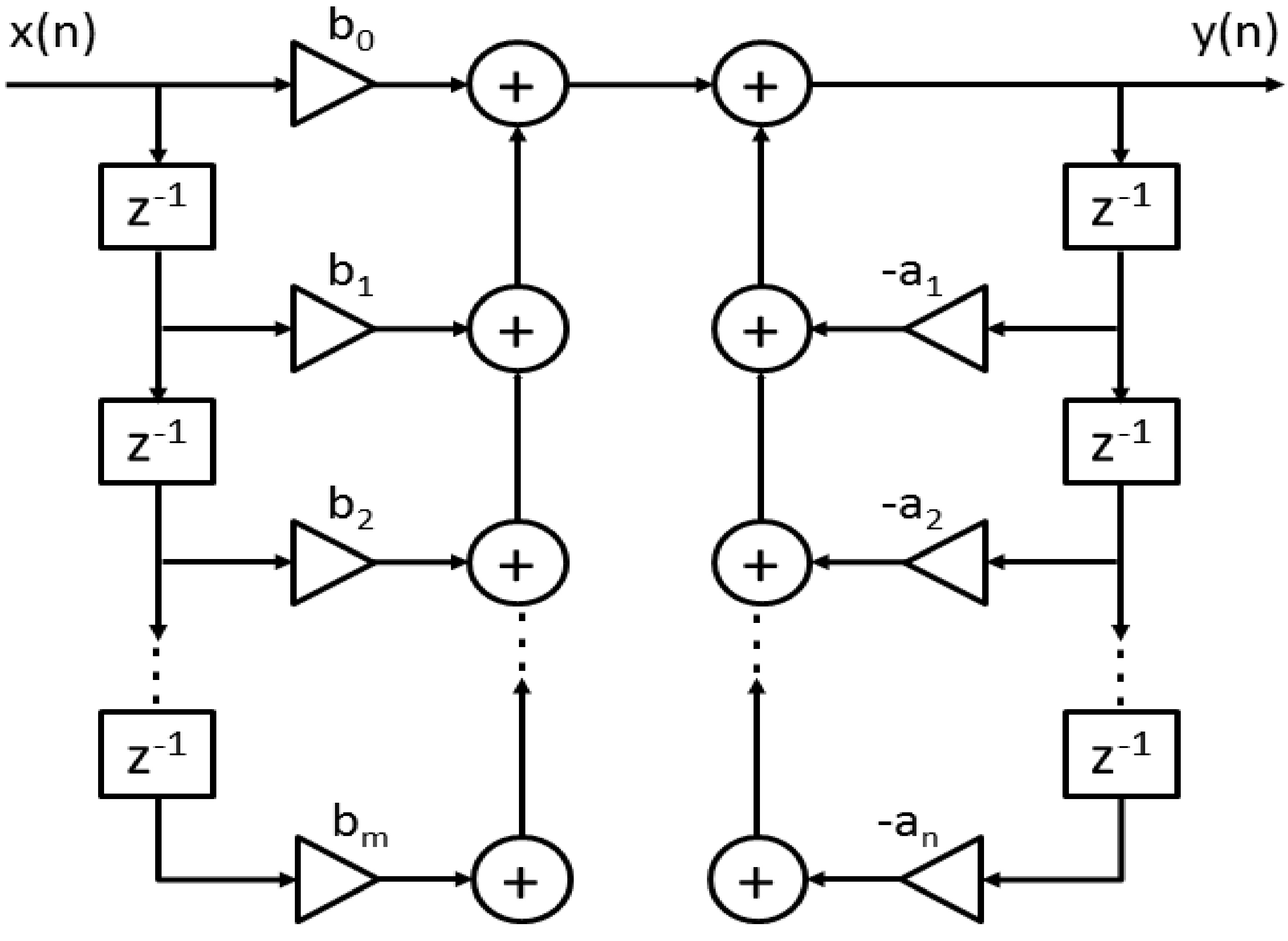
- The filter can be described graphically as a block diagram

(a)



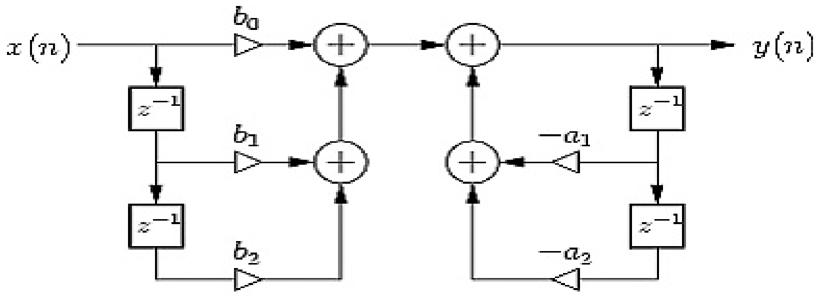
- This diagram represents the above difference equation (a box labeled z^{-1} denotes a one-sample delay in time)

Filter Representation: Direct Structure



Filter Representation

(a)



- **This structure is known as a *direct structure* of a digital filter and it has two main sections: *feedforward and feedback***
- **The input signal is applied directly to the feedforward section**
- **The overall output is formed by adding together the outputs from feedforward and feedback sections**
- **"Older" outputs are held in a shift register of the feedback section**

The Transfer Function

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) - a_1 y(n-1) - a_2 y(n-2)$$

- **Taking the z-transform of both sides of difference equation and using the delay theorem, it is resulted in:**

$$Y(z) = Z[y(n)] = b_0 X(z) + b_1 z^{-1} X(z) + b_2 z^{-2} X(z) - a_1 z^{-1} Y(z) - a_2 z^{-2} Y(z)$$

- **Collecting terms, it is resulted in:**

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Recursive and Non-Recursive Realizations

- **The general form of the filter difference equation is given by**

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

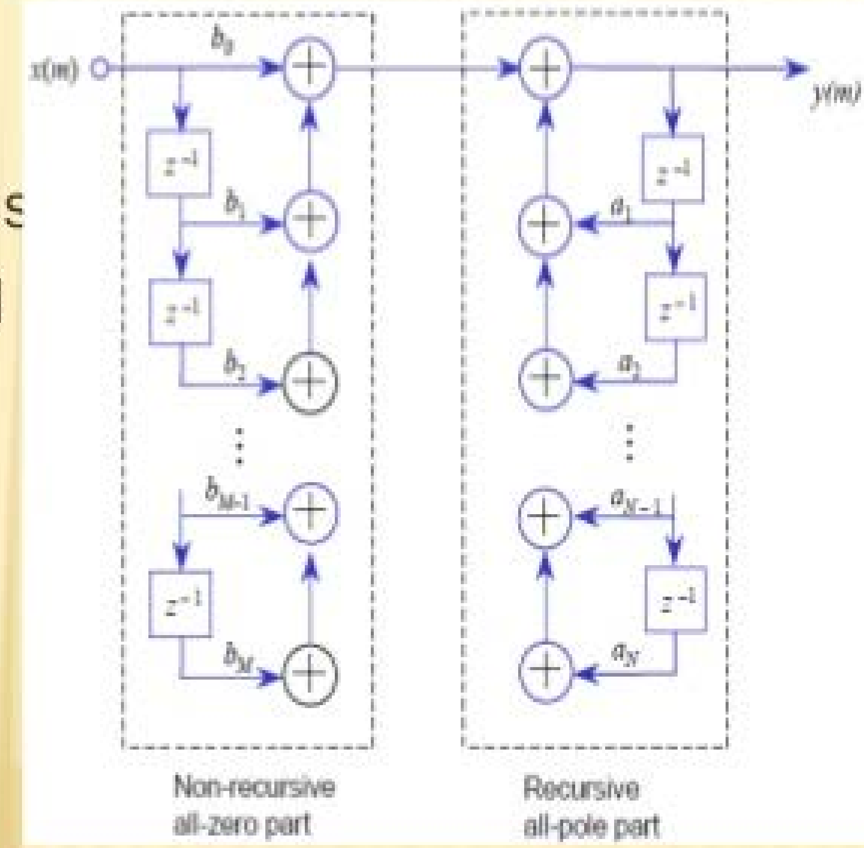
- **This is a *recursive* realization, e.g. the current output is a function of past and present inputs and outputs**

- **For a *non-recursive* realization, the current output is a function only of past and present *inputs*:**

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

RECURSIVE AND NON RECURSIVE FILTERS

- A recursive filter has feedback from output to input, and in general its output is a function of the previous output samples and the present and past input samples



The Transfer Functions for Recursive and Non-Recursive Realizations

- **Taking z-transform of both sides of recursive and non-recursive equations leads to the general forms of the transfer function for recursive and non-recursive realization respectively**

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

$$H(z) = \sum_{k=0}^{M-1} b_k z^{-k}$$

Poles and Zeros of the Transfer Function

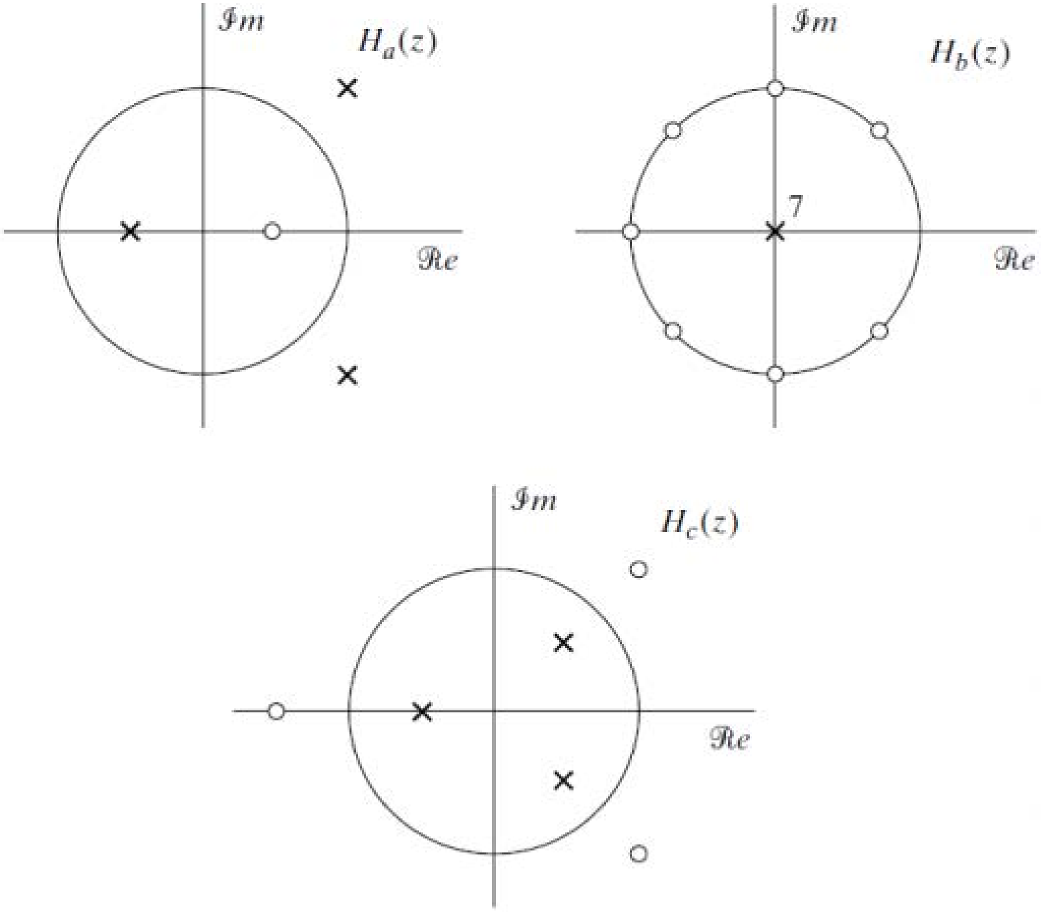
- **The location of poles and zeros affects the frequency response characteristics of filters**

- **The basic principle underlying the pole-zero placement is to locate poles near points of the unit circle corresponding to frequencies to be emphasized, and to place zeros near the frequencies to be deemphasized**

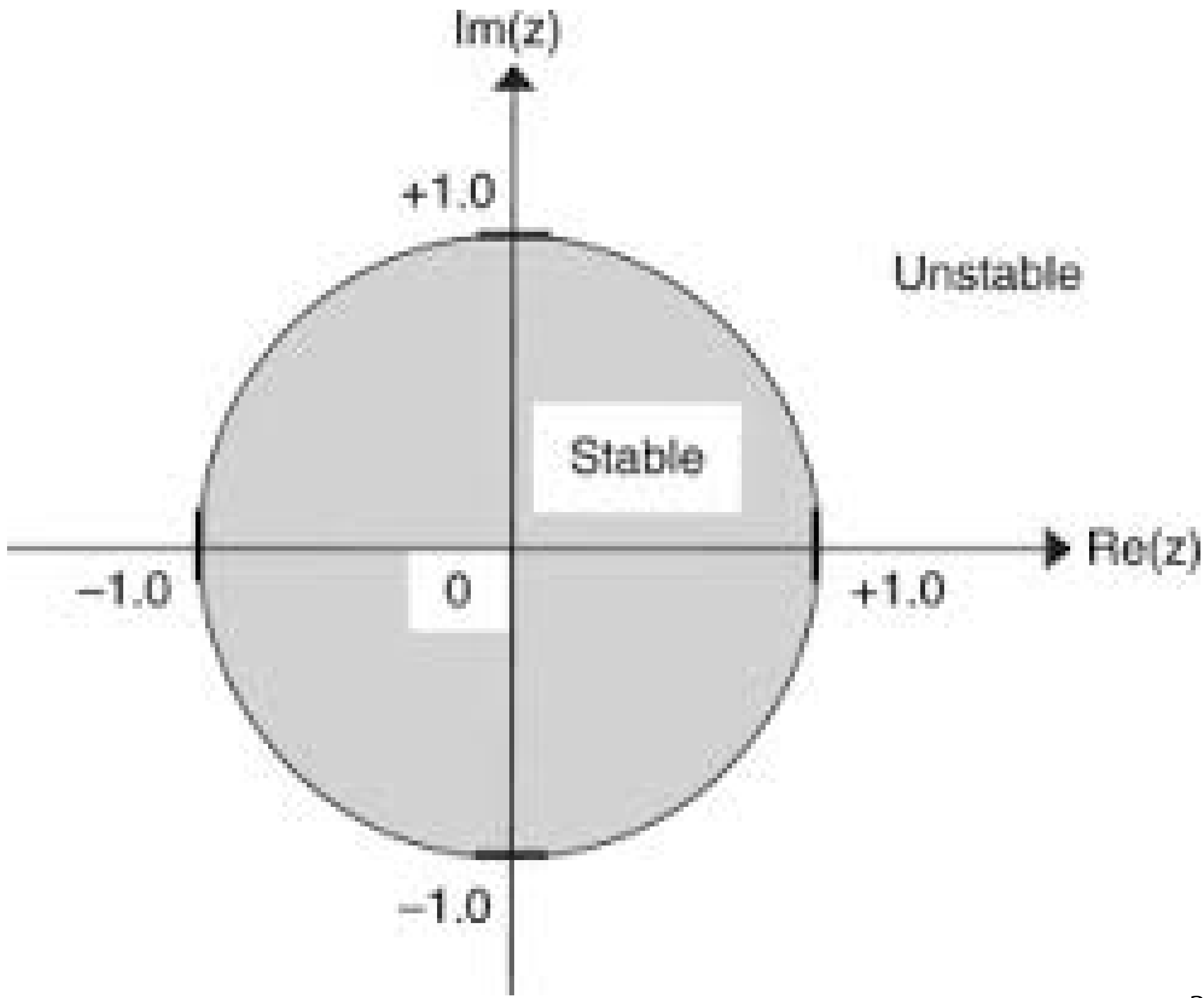
- **The position of the poles is an indicator of filter stability**

- **A filter will be stable if all its poles lie inside the *unit circle* in the z-plane; pole modulus should be less than unity**

Poles of Stable Filter

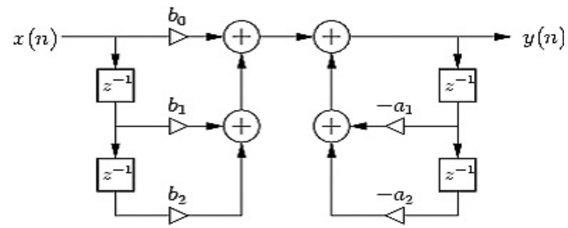


Stable and Unstable Regions in Z-Plane



Finite Impulse Response (FIR) Filter

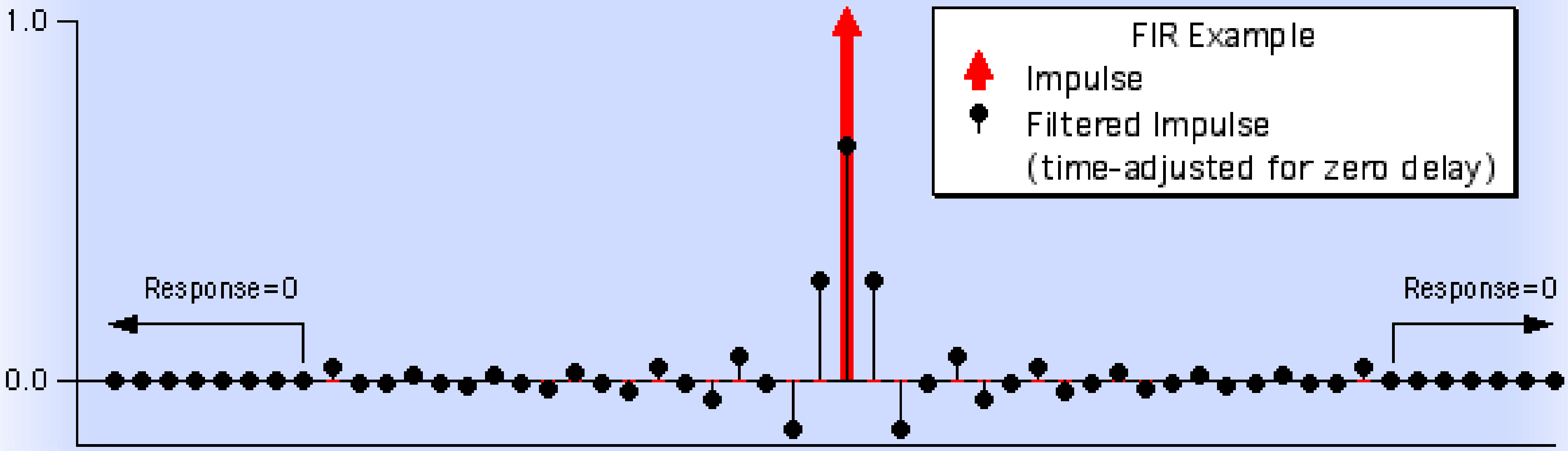
(a)



- **There is no feedback section, all feedback coefficients are zero**
- **Apply a unit impulse: all samples after it are zero.**
- **At time zero the output will be b_0 . We then clock the filter which causes the single sample to shift one memory unit down. The output of the filter is now b_1 . We keep clocking the filter until the single impulse reaches the end of the feedforward section, in which case the output will be b_2 . If we clock the filter once more, the output will be zero.**

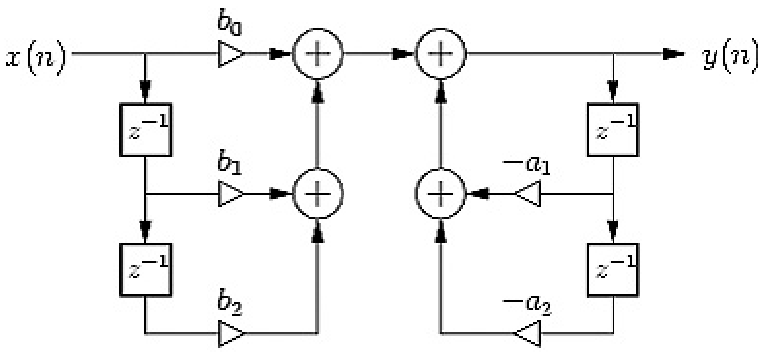
- **So, the output is the impulse response** (b_0, b_1, b_2)
- **The impulse response is *finite* and hence the filter is the *finite impulse response* (FIR)**

Finite Impulse Response



Infinite Impulse Response (IIR) Filter

(a)



- **For simplicity, lets consider that there is a *feedback* section and *no feedforward section*, i.e. all the coefficients b_i are zero except b_0 which is 1.**
- **Apply a unit impulse: the first output at $n=1$ will be 1**

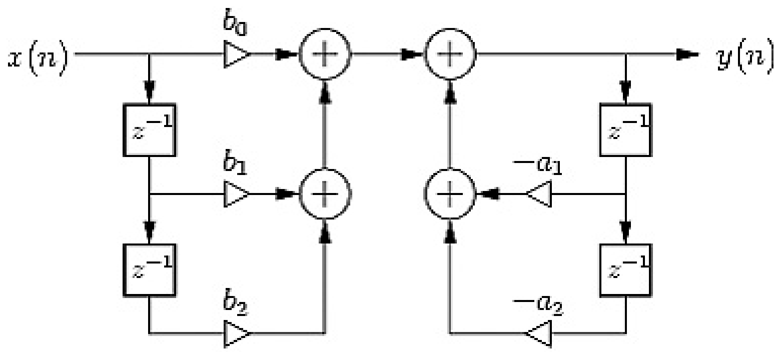
We then clock the filter. i. e. $n=2$

- **Although the new input is zero, the previous output value of 1 is sitting at the output of the first delay in the feedback section**

Thus, the second output will be $-a_1$

Infinite Impulse Response (IIR) Filter

(a)



- **We then clock the filter once more,** $n = 3$

  The previous outputs circulate round the feedback section, i.e.

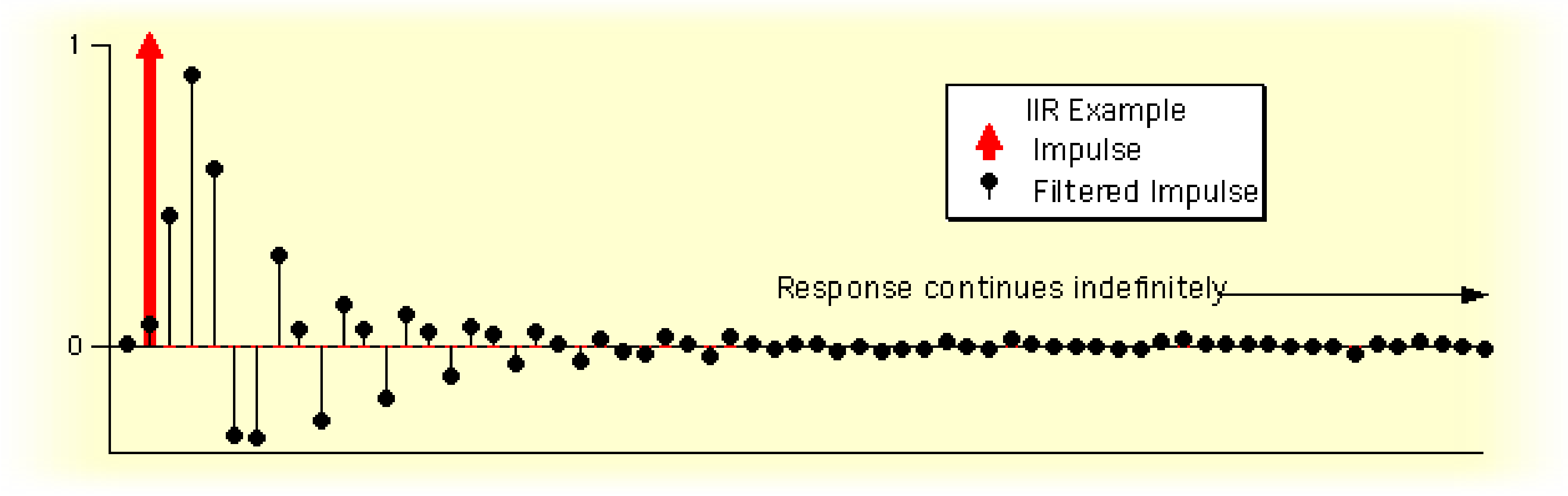
$$y(3) = -a_1 y(2) - a_2 y(1) = a_1^{\ 2} - a_2$$

- **The next output will be**

$$y(4) = 2a_1 a_2 - a_1^{\ 3}$$

- **If we continue to clock the filter, we will continue to get an output as the data circulate round and round the feedback section. If the filter is stable, the output will decay towards zero but never quite**

Infinite Impulse Response



Infinite Impulse Response (IIR) Filter

The impulse response is *infinite* and the filter is the *infinite impulse response* (IIR)

FIR Filters

FIR Filters: Advantages

Among the main *advantages* of FIR filters are:

- **FIR filters with exactly *linear phase* can be easily designed**

  Almost all MATLAB design functions for FIR filters design *linear phase filters only*

  Therefore, FIR linear phase filters are important for speech processing and data transmission

- **FIR filters are always stable**

- **The filter start up transients have a low duration.**

FIR Filters: Equations and the Convolution

- **FIR filter is described by the following difference equation:**

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

- **or equivalently by the convolution**

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

- **This structure requires $M-1$ memory locations for storing the previous inputs, and has a complexity of M multiplications and $M-1$ additions per output point**

- **FIR filter defines a weighted running average of M samples**

FIR Filters: the Convolution

The parameter M **is** ***order*** **of the FIR filter**

- **The process of computing the convolution involves the following four steps:**

- **folding**
- **shifting**
- **multiplication**
- **summation**

The Transfer Function and Stability

- **The transfer function of FIR filter is**

$$H(z)=\sum_{k=0}^{M-1} b_k z^{-k}$$

- **Now the poles and zeros of this polynomial are identified by expressing the above equation in powers of which are all greater than or equal to zero:**

$$H(z)=\frac{b_0 z^{M-1}+b_1 z^{M-2}+\ldots+b_{M-1} z^{0}}{z^{M-1}}$$

- **All the denominator poles are at the origin in the z-plane and hence FIR filters are *unconditionally stable* as the poles cannot be placed outside the unit circle**
- **Thus, the frequency response is controlled by the positions of the *numerator zeros***

The Frequency Response

- **The frequency response of the FIR filter is obtained directly by replacing z with exp $(i\omega)$:**

$$H(\omega) = \sum_{k=0}^{M-1} b_k \exp(-i\omega k) = \sum_{k=0}^{M-1} h(k) \exp(-i\omega k)$$

- **Thus, given a set of filter weights, we can evaluate the complex frequency response directly**
- **The above equation is a Fourier series (i.e. DFT) of the frequency response, which is periodic function of with period 2π**

- **Hence, the filter coefficients may be calculated by integrating in the frequency domain over the period of the frequency response (IDFT)**

$$b_k(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(\omega) \exp(i\omega n) d\omega$$

Computer-Aided Design of Linear-Phase FIR Filters

- **Optimized design of linear–phase filters can be achieved by iterative application of DFT/IDFT processing**

- **At the design stage, one has a desired ideal frequency response: i.e. without ripples in passband and stopband and maximum filter order**
- **An optimum design criterion is used in the sense that the weighted approximation error between the *desired* frequency response and the *actual* frequency response is spread evenly across the passband and evenly across the stopband of the filter**

- **In this method, one starts with a set of filter coefficients and perform a DFT to get the actual frequency response** $H(\omega)$

Computer-Aided Design of Linear-Phase FIR Filters

- **The filter performance limits are then imposed *on parts of the actual response* $H(\omega)$ which deviate from desired frequency response and then the *updated* actual response $H(\omega)$ is given for IDFT to yield an *updated* set of filter coefficients**

- **This process will result in an updated filter coefficients normally having component number beyond the maximum filter order permitted for this design**

- **These additional values are simply truncated and then the iterative process is repeated to get a new actual frequency response $H(\omega)$, which is again compared with the desired frequency response**

- **This optimization technique is guaranteed to converge**

Computer-Aided Design of Linear-Phase FIR Filters

- **The objective of this technique is to determine *iteratively* the filter coefficients so that the difference (i.e. weighted error function) between the *desired* frequency response, and *actual* frequency response for all frequencies is minimized**

- ***The Parks-McClellan algorithm* which reduces significantly the filter complexity compared with window design, uses a minimum weighted Chebyshev error to approximate the desired frequency response $H_d(\omega)$ by iteration:**

$$\min\left\{\max\left|L(\omega)\left[H_d(\omega)-H_a(\omega)\right]\right|\right\}$$

Computer-Aided Design of Linear-Phase FIR Filters

- **The set of filter coefficients,** b_k **, which *minimizes* the *maximum* error between the desired frequency response, and actual frequency response provides directly the optimal filter design**

- **The positive weighting function** $L(\omega)$ **, allows the designer to *emphasize* some areas of the frequency response more than others**

- **The minimization is performed *iteratively* using a computer program**

- **The resulting filter have ripples in both the passband and the stopband**

IIR Filters

Infinite Impulse Response (IIR) Filters

- ***FIR filters is not the most general class of filters***

- **The most general class that can be implemented with a finite amount of computation is obtained when the output is formed *not only from the input*, but also from previously computed outputs**

- ***The primary advantage of IIR filters over FIR filters is that they typically meet a given set of specifications with a much lower filter order than a corresponding FIR filter***

IIR Filters: Equation

- **An IIR filter is described by the difference equation:**

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

- **The coefficients a_k are called feedback coefficients, and the coefficients b_k are called the feedforward coefficients.**

- **The number N of feedback terms is *the order of an IIR filter***

- ***In most cases, especially digital filters derived from analog filters, $M \le N$ in order to prevent infinite gain at infinite frequency***

IIR Filters: the Transfer Function

The general form of the transfer function of IIR filters is

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

Linearity and Time Invariance

Although the feedback terms make the proof of linearity and time invariance more complicated than the FIR case, we can easily show that:

- **the principle of superposition (e.g. *filter linearity*) will hold because the difference equation involves only linear combinations of the input and output samples**

- ***time invariance* of the IIR filters also holds**

IIR Filter Design: Analogue Prototyping

- **The *most popular* technique for designing IIR filters is based on converting an analogue filter into digital filter**

- **An analogue filter can be described by its transfer function (with Laplace transform):**

$$H_a(s) = \frac{\sum_{k=0}^{M} \beta_k s^k}{\sum_{k=0}^{N} \alpha_k s^k}$$

where α_k **and** β_k **are the filter coefficients**

- ***No single transform exists to perfectly map*** $H_a(s)$ ***to*** $H(z)$

IIR Filter Design: Analogue Prototyping

- **In the design of IIR filters by analogue prototyping, we shall specify the desired filter characteristics for the *magnitude frequency response only* and *accept the phase response* that is obtained from the design methodology**

- **The classic IIR filter design by analogue prototyping can be realized by the following two approaches**

IIR Filter Design: Analogue Prototyping

Approach 1

- **Find an basic analogue lowpass filter and transform this "prototype" filter to the desired frequency band configuration in the *analogue domain***
- **Transform the filter to the digital domain**

Approach 2

- **Find an basic analogue lowpass filter and translform this "prototype" filter to the *digital domain***
- **Transform digital lowpass filter to the desired frequency band configuration in the digital domain**

IIR Filter Design: Analogue Prototyping

- **Thus, approach 1 is to perform the frequency transformation in the analogue domain and then to convert the analogue filter into a corresponding digital filter**

- **The approach 2 is first to convert the analogue lowpass filter into a lowpass digital filter and then to transform the lowpass digital filter into the desired digital filter by a digital frequency transformation**

- ***In general, these two approaches yield different results***

The Butterworth Analogue Filters

- **The Butterworth filters are characterized by the property that the magnitude response is *maximally flat in the passband***

- **Another property is that the approximation is *monotonic* in the passband and the stopband**

- **The squared magnitude transfer function is**

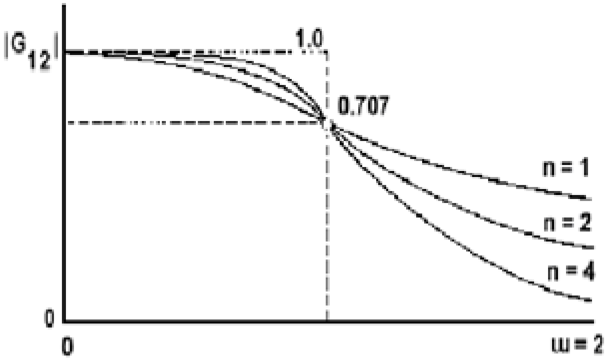
$$|H(\Omega)|^2 = \frac{1}{1 + \left(\dfrac{\Omega}{\Omega_p}\right)^{2N}}$$

where N is the order of the filter, Ω_p is the passband edge frequency

The Butterworth Analogue Filters

- As the order increases, the magnitude become sharper

- They remain closer to unity over more of passband and become close to zero rapidly on the stopband, although the magnitude at the passband edge frequency is always -3dB



The Bessel Analogue Filters

- **An important characteristic of Bessel analogue filters is the *linear-phase* response over the passband of the filter; thus, filtered signals maintain their wave shapes in the passband frequency range**

- **However, we should emphasize that the linear-phase characteristics of the analogue filter are destroyed in the process of converting the filter into the digital domain**

- **Therefore, digital Bessel filters *do not have* this property**

The Bessel Analogue Filters

- **Difficulty with Bessel filters is that unlike Butterworth filters the passband edge frequency varies with the filter order**

- **Bessel filters generally require a higher filter order than other filters for satisfactory stopband attenuation**

The Chebyshev Analogue Filters

- **Chebyshev filters are characterized by the property that over a *prescribed frequency band* the approximation error is minimized**

- **The magnitude error is, in fact, equiripple over the frequency band**

- **Depending on *whether the band of frequencies* over which the error is minimized is the passband or the stopband, the filter designs are called Type I or Type II**

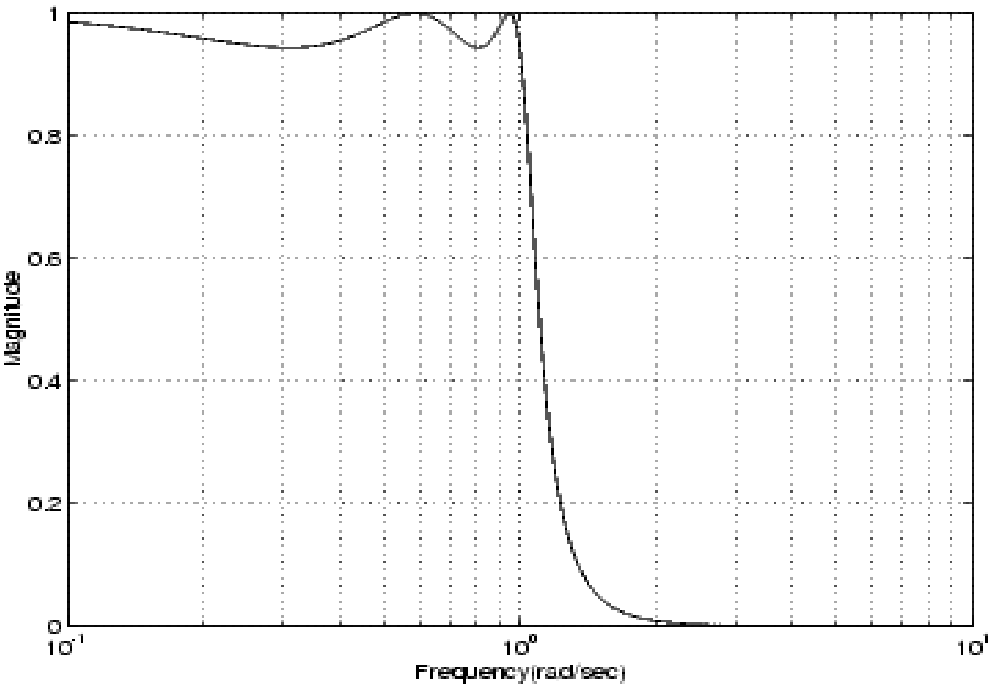
The Chebyshev Analogue Filter Type I

- **Chebyshev Type I filters minimize the absolute difference between the ideal and actual frequency response over the entire *passband* by incorporating an equal ripple in the *passband***

- **Thus, these filters exhibit *equiripple passband behaviour* and monotonic stopband response**

The Chebyshev Analogue Filter Type I

- The typical magnitude response for Chebyshev Type I filter



- The optimality property that Type I Chebyshev filters satisfy is there is no better filter with equal or better performance in both the passband and stopband

The Chebyshev Analogue Filter Type II

- **Type II filters minimize the absolute difference between the ideal and actual frequency response over the entire *stopband* by incorporating an equal ripple in the stopband**

- **These filters exhibit monotonic behaviour in the passband and *equiripple behaviour in the stopband***

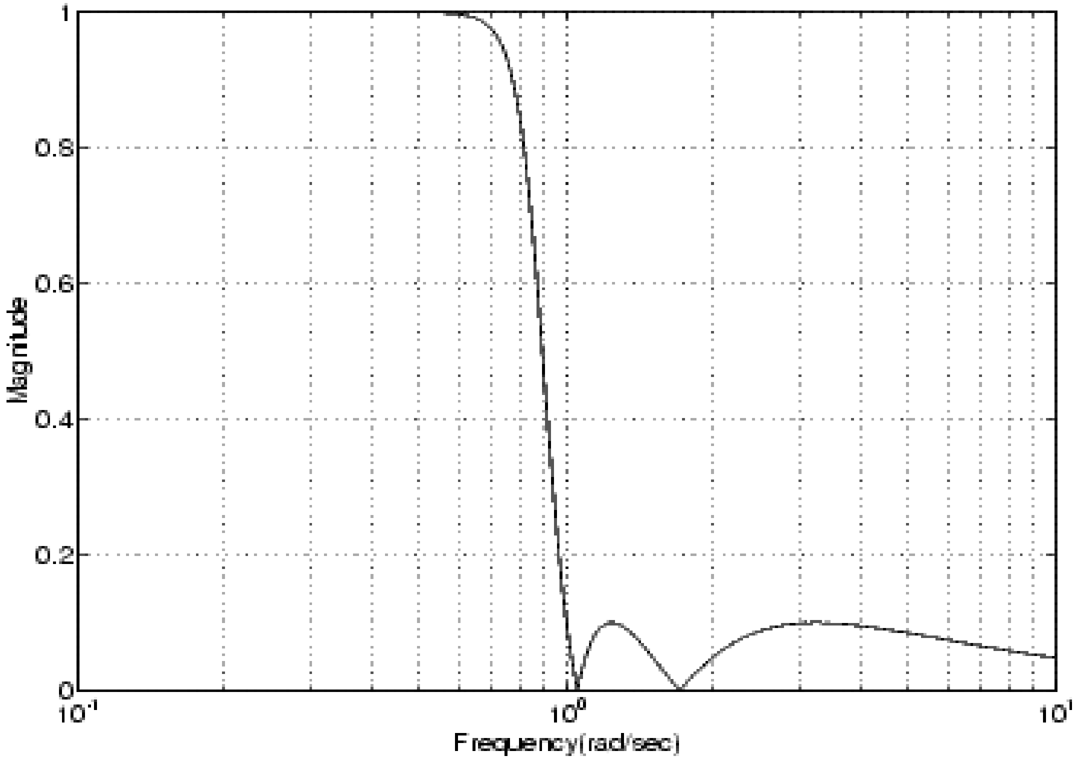
- **The squared-magnitude response of a filter can be expressed as**

$$|H(\Omega)|^2 = \frac{1}{1+\delta_1^{\,2}[T_N\,(\Omega_r/\Omega_p)/T_N(\Omega_r/\Omega)]^2}$$

where Ω_r is the lowest frequency at which the stopband loss attains a prescribed value

The Chebyshev Analogue Filter Type II

- **Typical magnitude response for Chebyshev Type II filter**



Elliptic Filters

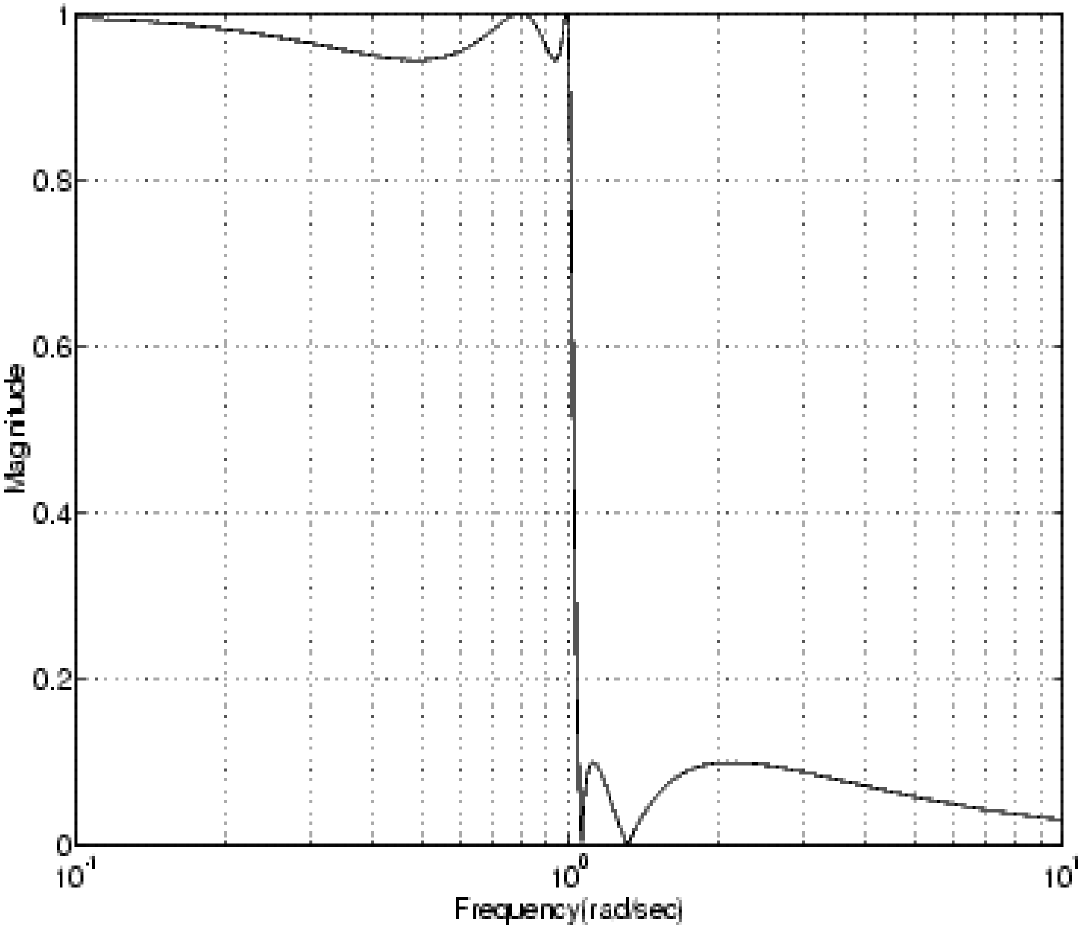
- **Elliptic filters are characterized by a magnitude response that *is equiripple in both the passband and the stopband***

- **It can be shown that elliptic filters are optimum in the sense that for a given order and for given ripple specifications no other filter achieves a faster transition between the passband and stopband, e.g. they have *the smallest transition band***

- **They generally meet filter requirements with the *lowest order* of any supported filter type**

Elliptic Filters

The typical magnitude response for elliptic filters



Elliptic Filters: the Phase Response

- **The phase response of elliptic filters is more nonlinear in the passband than a comparable Butterworth or a Chebyshev filter, especially near the band edge**

- **In view of the optimality of elliptic filters, one important reason that other types of filters might be preferable in some applications is that they possess better phase response characteristics**

Comparison of IIR and FIR Filters

- **The choice between a FIR filter and an IIR filter depends upon the relative weight that one attaches to the advantages and disadvantages of each type of filter**

- **If we put aside phase consideration (IIR filters have a nonlinear phase) it is generally true that a given magnitude frequency response specification will be met more efficiently with an IIR filter**

- **It has been shown that for most practical filter specifications, the ratio N_{FIR}/N_{IIR} is typically of the order of *tens*, where N_{FIR} and N_{IIR} are the numbers of multiplications per output sample for an FIR and IIR filters respectively**

Comparison of IIR and FIR Filters

- **However, IIR design disregards the *phase response* of the filter**
- **In contrast, FIR filters can have precisely linear phase**
- **In addition, *they always stable***
- **In many applications, the linearity of phase response is not an issue**
- **However, in many cases, the linear phase available with an FIR filter may be well worth the extra cost**

Comparison of IIR and FIR Filters

- **IIR filters have the advantage that they can be designed using close-form design expressions**

- **Most of the other FIR design methods are iterative procedures, requiring rather powerful computational facilities for their implementation**

Comparison of IIR and FIR Filters

- **It has been found that the most favorable conditions for an FIR design are large values of passband ripple, small values of stopband ripple and large transition bands**

- **In contrast, it is often possible to design frequency selective IIR filters using simple calculations and tables of analogue filter design parameters**

- **If we consider specific filters, *elliptic IIR filters* are generally more efficient in achieving given specification on the *magnitude response* than optimum FIR filters**

- **However, its phase response will be very nonlinear (especially at the band edge)**