GANDHI INSTITUTE OF SCIENCE AND TEHNOLOGY,RAYAGADA

DIGITAL SIGNAL PROCESSING (REE4G001,RCS4G002)

BRANCH –CSE / EE / EEE

PREPARED BY PROF. SHRINIBAS PATTNAIK

 4th Semester

REE4D002	**Signal and Systems**	**L-T-P 3-0-0**	**3 CREDITS**

MODULE – I (7 Hours)

Discrete-Time Signals and Systems: Discrete-Time Signals: Some Elementary Discrete-Time signals, Classification of Discrete-Time Signals, Simple Manipulation, Discrete-Time Systems : Input-Output Description, Block Diagram Representation, Classification, Interconnection.

MODULE – II (8 Hours)

Analysis of Discrete-Time LTI Systems: Techniques, Response of LTI Systems, Properties of Convolution, Causal LTI Systems, Stability of LTI Systems; Discrete-Time Systems Described by Difference Equations; Implementation of Discrete-Time Systems. Correlation of Discrete-Time Signals: Cross correlation and Autocorrelation Sequences, Properties.

MODULE – III (10 Hours)

The Continuous-Time Fourier Series: Basic Concepts and Development of the Fourier series; Calculation of the Fourier Series, Properties of the Fourier Series. **The Continuous-Time Fourier Transform:** Basic Concepts and Development of the Fourier Transform; Properties of the Continuous-Time Fourier Transform.

MODULE- IV (10 Hours)

The Z-Transform and Its Application to the Analysis of LTI Systems: The Z-Transform: The Direct Z-Transform, The Inverse Z-Transform; Properties of the Z-Transform; Rational Z-Transforms: Poles and Zeros, Pole Location and Time-Domain Behavior for Causal Signals, The System Function of a Linear Time-Invariant System; Inversion of the Z-Transforms: The Inversion of the Z-Transform by Power Series Expansion, The Inversion of the Z-Transform by Partial-Fraction Expansion; The One-sided Z-Transform: Definition and Properties, Solution of Difference Equations.

MODULE-V (10 Hours)

The Discrete Fourier Transform: Its Properties and Applications: Frequency Domain Sampling: The Discrete Fourier Transform; Properties of the DFT: Periodicity, Linearity, and Symmetry Properties, Multiplication of Two DFTs and Circular Convolution, Additional DFT Properties.

Books:

1. Digital Signal Processing – Principles, Algorithms and Applications, John. G. Proakis and Dimitris. G. Manolakis, 4th Edition, Pearson.

2. Fundamentals of Signals and Systems - M. J. Roberts, TMH

3. Signal & Systems by Tarun Kumar Rawat, Oxford University Press.

4. Signals and Systems – A NagoorKani, TMH

5. Signals and Systems, Chi-Tsong Chen, Oxford

6. Principles of Signal Processing and Linear Systems, B.P. Lathi, Oxford.

7. Principles of Linear Systems and Signals, B.P Lathi, Oxfor

(DIGITAL SIGNAL PROCESSING)

OBJECTIVES:

- To understand the basic concepts and techniques for processing signals and digital signal processing fundamentals.
- To Understand the processes of analog-to-digital and digital-to-analog conversion and relation between continuous-time and discrete time signals and systems.
- To Master the representation of discrete-time signals in the frequency domain, using z-transform, discrete Fourier transforms (DFT).
- To Understand the implementation of the DFT in terms of the FFT, as well as some of its applications (computation of convolution sums, spectral analysis).
- To learn the basic design and structure of FIR and IIR filters with desired frequency responses and design digital filters.
- The impetus is to introduce a few real-world signal processing applications.
- To acquaint in FFT algorithms, Multi-rate signal processing techniques and finite word length effects.

1. Digital Signal Processing, Principles, Algorithms, and Applications: John G. Proakis, Dimitris G. Manolakis, Pearson Education / PHI, 2007.
2. Discrete Time Signal Processing – A. V. Oppenheim and R.W. Schaffer, PHI, 2009.
3. Fundamentals of Digital Signal Processing – Loney Ludeman, John Wiley, 2009

REFERENCE BOOKS:

1. Digital Signal Processing – Fundamentals and Applications – Li Tan, Elsevier, 2008.
2. Fundamentals of Digital Signal Processing using MATLAB – Robert J. Schilling, Sandra L. Harris, b Thomson, 2007.
3. Digital Signal Processing – S.Salivahanan, A.Vallavaraj and C.Gnanapriya, TMH, 2009.
4. Discrete Systems and Digital Signal Processing with MATLAB – Taan S. ElAli, CRC press, 2009.
5. Digital Signal Processing - A Practical approach, Emmanuel C. Ifeachor and Barrie W. Jervis, 2nd Edition, Pearson Education, 2009.
6. Digital Signal Processing - Nagoor Khani, TMG, 2012.

OUTCOMES

On completion of the subject the student must be able to:

- Perform time, frequency and z-transform analysis on signals and systems
- Understand the inter relationship between DFT and various transforms
- Understand the significance of various filter structures and effects of rounding errors
- Design a digital filter for a given specification
- Understand the fast computation of DFT and Appreciate the FFT processing

Understand the trade-off between normal and multi rate DSP techniques and finite length word effects

1.1 Basic Concepts of Signal Processing

Figure 1.1.1 describes the concept of analog signal processing. An analog signal (transducer signal plus noise) produced by a transducer (sensor) is captured for the real-world application. For example, a temperature sensor produces a small voltage (10 mV per ^0C) based on the temperature of environment; a microphone generates a voltage range from approximately 50 mV to 100 mV according to loudness of voice. To be able to use the acquired analog signal, two steps usually are involved. First, the small scale analog signal will be signal conditioned or amplified, we refer this as the time domain processing after which the amplified signal range fits for applications, for example, the amplified temperature signal is feasible to drive the analog device or can be used for the *analog to digital conversion (ADC)* channel for further processing application. Similarly, the amplified microphone signal could drive the loudspeaker, or pass to the ADC channel for digital recording. However, during the noisy sensor environment and amplifying process, signal noise is also added to the desired signal such as signal fluctuation in the temperature signal, or hissing sound in the recorded voice. The noise could be fully or partially removed by using an *analog filter* as shown in Figure 1.1.1. We refer this process as the frequency domain processing.

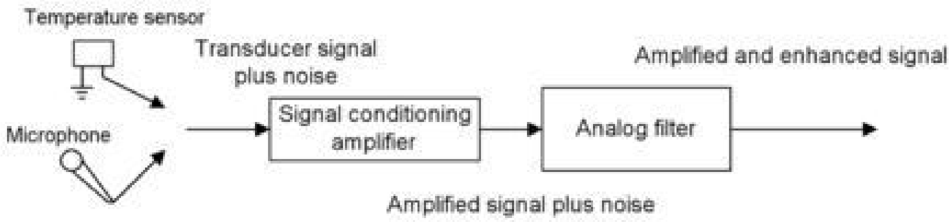


Figure 1.1.1 Analog signal processing scheme.

A major objective of analog signal processing is to design a suitable analog filter, which could be constructed using the electronic devices based on the characteristics of the desired signal and noise. Via analog signal processing, the enhanced signal is produced.

The concept of *digital signal processing (DSP)* is better illustrated by a typical simplified block diagram in Figure 1.1.2, which consists of several blocks such as the analog filter, ADC, *digital signal processor*, *digital to analog converter (DAC)*, and *reconstruction filter (anti-image filter)*.

Figure 1.1.2 Digital signal processing scheme.

As shown in the block diagram, the analog signal, which is continuous both in time and amplitude, is generally encountered in our real life. Examples of such analog signals include current, voltage, temperature, pressure, and light intensity. Usually a transducer (sensor) with an amplifier is used to convert a non-electrical signal to an analog electrical signal (voltage). This analog signal is then fed to an analog filter, where the analog filter performs filtering to limit the frequency range of the analog signal prior to the sampling process. The purpose of filtering is to significantly attenuate the aliasing distortion, which will be explained in Chapter 7. The band-limited signal at the output of the analog filter is then sampled and converted via an ADC into the *digital signal*, which is discrete both in time and amplitude.

The digital signal processor then accepts the digital signal and processes the digital data according to the digital signal processing rules such as *lowpass*, *highpass*, *bandpass* digital filters, or other algorithms for different applications. Notice that the digital signal processor is a special type of digital computer, which could be a general-purpose digital computer, a microprocessor, or an advanced micro-controller; furthermore, digital signal processing rules could be implemented using software in general. With the digital signal processor and corresponding software, a processed digital output signal is generated. This signal behaves in a manner according to the specific algorithm used. The next block in Figure 1.1.2 is the DAC, which converts the processed digital signal to an output signal. As shown in Figure 1.1.2, the signal is continuous in time but discrete in amplitude (usually sample and hold signal). The final block is designated as a function to smooth the obtained output signal back to the analog signal via a reconstruction (anti-image) filter for real-world applications.

As we can see, the analog signal processing does not require software, algorithms, ADC, and DAC. The processing fully relies on the electrical and electronic devices such as resistors, capacitors, transistors, operational amplifiers, and integrated circuits (IC). Digital signal processing requires analog signal processing before the ADC and after the ADC. Since the digital signal processor uses software, digital processing, and algorithms, it has a great deal of flexibility, less noise interference, and no signal distortion in various applications. As shown in Figure 1.1.2, the analog signal processing cannot be avoided and is a must for converting real-world information to a digital form and the digital form back to real world. In next section, we will focus on reviewing some typical applications of digital signal processing.

1.2 List of Signal Processing Application Examples

Applications of DSP are increasing in many areas where analog electronics are replaced by the digital signal processors while new applications are depending on the digital signal processors. With the decreasing cost of the digital signal processors and the increase in its performance, DSP is likely continue to impact engineering design in our modern daily life. Typical examples using the DSP are listed below:

Digital audio and speech:

Digital audio coding such as CD players and MP3 players
Digital crossovers and digital audio equalizers
Digital stereo and surround sound
Nose reduction system
Speech coding
Data compression and encryption

Digital telephone:

Speech recognition
High-speed modems

Echo cancellation
Speech synthesizers
TDMF generation and detection
Answering machines

Automobile industry:
Active noise control system
Active suspension system
Digital audio and radio
Digital control

Electronic communications:
Cellular phones
Digital telecommunications
Wireless LAN
Satellite communications

Medical image equipment:
ECG analyzers
Cardiac monitoring
Medical image and image recognition
Digital X-rays and image processing

Multimedia:
Internet phones, audio, and video
Hard disk drive electronics
Digital pictures
Digital cameras
Text-to-voice and voice-to-text technologies

However, the list of applications above is not meant to cover all signal processing applications. DSP areas are increasing and being explored by engineers and scientists. More and more DSP techniques are impacting and will continue to improve our life.

Classification of discrete-time signals (Along lines similar to continuous-time signals)

Discrete-time Energy and Power signals The *energy E* of a discrete-time signal $x(n)$ is given by

$$E = \lim_{N\to\infty} \sum_{n=-N}^{N} x(n)x^*(n)$$

where x^* is the complex conjugate of x. If $x(n)$ is a real sequence then $x(n)\, x^*(n) = x^2(n)$. The above definition can also be written as

$$E = \lim_{N\to\infty} \sum_{n=-N}^{N} |x(n)|^2 \quad \text{(there are } 2N+1 \text{ terms here)}$$

The *average power P* of the signal is

$$P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x(n)|^2$$

If E is finite but non zero (i.e., $0 < E < \infty$) the signal is an *energy signal*. It is a *power signal* if E is infinite but P is finite and nonzero (i.e., $0 < P < \infty$). Clearly, when E is finite, $P = 0$. If E is infinite P may or may not be finite.

If neither E nor P is finite, then the signal is neither an energy nor a power signal.

PROBLEM: For the signal $x(n) = 1$ for all n,

$$E = \lim_{N\to\infty} \sum_{n=-N}^{N} |x(n)|^2 = \sum_{n=-\infty}^{\infty} 1^2 = \infty \text{ which is infinite energy}$$

$$P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x(n)|^2 = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} 1^2 = \lim_{N\to\infty} \frac{2N+1}{2N+1} = 1 \text{ which is finite}$$

Thus $x(n)$ is a power signal.

Periodic signal The discrete-time signal $x(n)$ is periodic if, for some integer $N > 0$

$$x(n+N) = x(n) \text{ for all } n$$

The smallest value of N that satisfies this relation is the **(fundamental) period** of the signal. If there is no such integer N, then $x(n)$ is an aperiodic signal.

Given that the continuous-time signal $x_a(t)$ is periodic, that is, $x_a(t) = x_a(t+T_0)$ for all t, and that $x(n)$ is obtained by sampling $x_a(t)$ at T second intervals, $x(n)$ will be periodic if T_0/T is a rational number but not otherwise. If $T_0/T = N/L$ for integers $N \geq 1$ and $L \geq 1$ then $x(n)$ has exactly N samples in L periods of $x_a(t)$ and $x(n)$ is periodic with period N.

Periodicity of sinusoidal sequences The sinusoidal sequence $\sin(2\pi f_0 n)$ has several major differences from the continuous-time sinusoid as follows:

a) The sinusoid $x(n) = \sin(2\pi f_0 n)$ or $\sin(\omega_0 n)$ is periodic if f_0, that is, $\omega_0/2\pi$, is rational. If f_0 is not rational the sequence is not periodic. Replacing n with $(n+N)$ we get

$$x(n+N) = \sin(2\pi f_0 (n+N)) = \sin 2\pi f_0 n.\ \cos 2\pi f_0 N + \cos 2\pi f_0 n.\ \sin 2\pi f_0 N$$

Clearly $x(n+N)$ will be equal to $x(n)$ if $f_0N = m$, an integer or $f_0 = m/N$. The fundamental period is obtained by choosing m as the smallest integer that yields an integer value for N. For example, if $f_0 = 15/25$, which in reduced fraction form is 3/5, then we can choose m = 3 and get $N = 5$ as the period. If f_0 is rational then $f_0 = p/q$ where p and q are integers. If p/q is in reduced fraction form then q is the period as in the above example.

On the other hand if f_0 is irrational, say $f_0 = \sqrt{2}$, then N will not be an integer, and thus $x(n)$ is aperiodic.

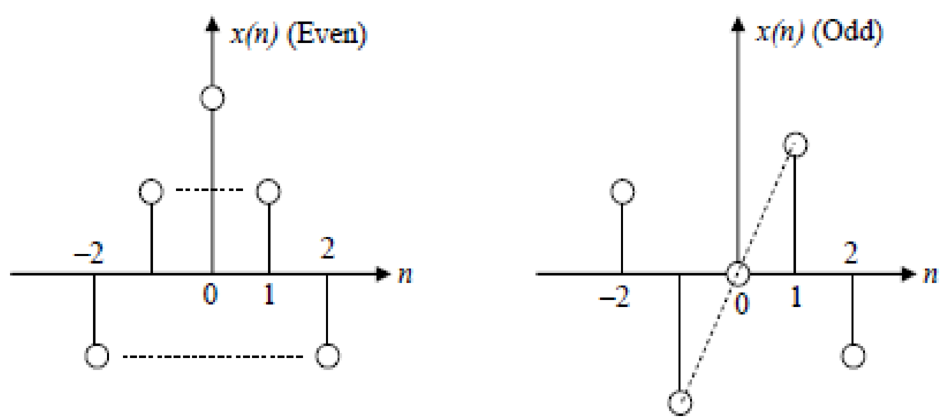
The sum of two discrete-time periodic sequences is also periodic. Let $x(n)$ be the sum of two periodic sequences, $x_1(n)$ and $x_2(n)$, with periods N_1 and N_2 respectively. Let p and q be two integers such that

$$pN_1 = qN_2 = N \qquad (p \text{ and } q \text{ can always be found})$$

Then $x(n)$ is periodic with period N since, for all n,

$$\begin{aligned} x(n+N) &= x_1(n+N) + x_2(n+N) \\ &= x_1(n+pN_1) + x_2(n+qN_2) \\ &= x_1(n) + x_2(n) \\ &= x(n) \text{ for all } n \end{aligned}$$

Odd and even sequences The signal $x(n)$ is an *even* sequence if $x(n) = x(-n)$ for all n, and is an *odd* sequence if $x(n) = -x(-n)$ for all n.



The *even* part of $x(n)$ is determined as $x_e(n) = \dfrac{x(n)+x(-n)}{2}$ and the *odd* part of $x(n)$ is given by $x_o(n) = \dfrac{x(n)-x(-n)}{2}$. The signal $x(n)$ then is given by $x(n) = x_e(n)+x_o(n)$.

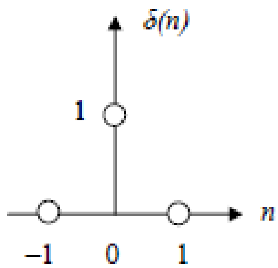
ELEMENTARY DISCRETE TIME SIGNALS:

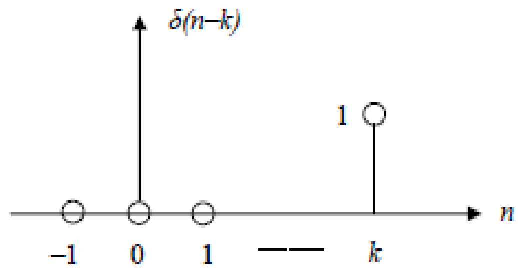
1) The unit sample sequence (discrete-time impulse, aka Kronecker delta)

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$

Whereas $\delta(n)$ is somewhat similar to the continuous-time impulse function $\delta(t)$ – the Dirac delta – we note that the magnitude of the discrete impulse is finite. Thus there are no analytical difficulties in defining $\delta(n)$. It is convenient to interpret the delta function as follows:

$$\delta(\text{argument}) = \begin{cases} 1 \text{ when argument} = 0 \\ 0 \text{ when argument} \neq 0 \end{cases}$$

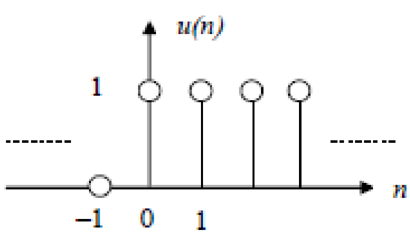


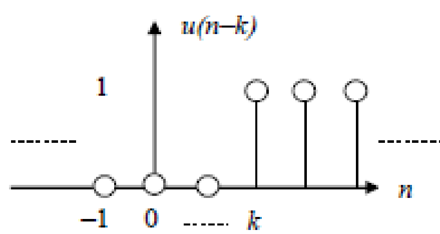


2) The unit step sequence

$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

$$u(\text{argument}) = \begin{cases} 1, \text{ if argument} \geq 0 \\ 0, \text{ if argument} < 0 \end{cases}$$



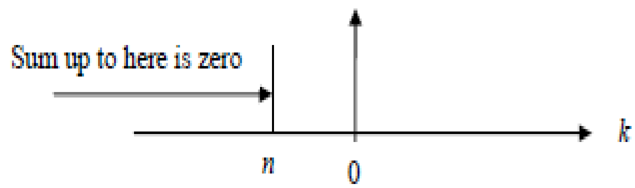


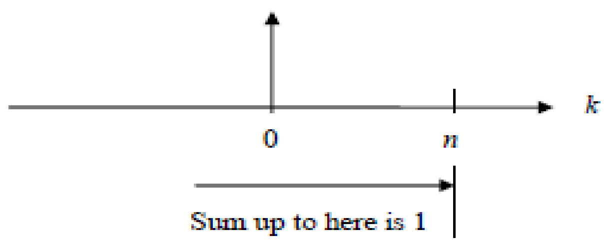
a) The discrete delta function can be expressed as the first difference of the unit step function:

$$\delta(n) = u(n) - u(n-1)$$

b) The sum from $-\infty$ to n of the δ function gives the unit-step:

$$\sum_{k=-\infty}^{n} \delta(k) = \begin{cases} 0 & \text{if } n < 0 \\ 1 & \text{if } n \geq 0 \end{cases} = u(n)$$

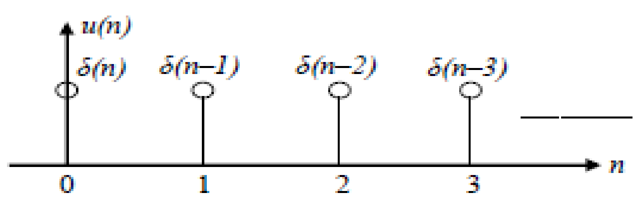




Results (a) and (b) are like the continuous-time derivative and integral respectively.

c) By inspection of the graph of $u(n)$, shown below, we can write:

$$u(n) = \delta(n) + \delta(n-1) + \delta(n-2) + \ldots = \sum_{\lambda=0}^{\infty} \delta(n-\lambda)$$



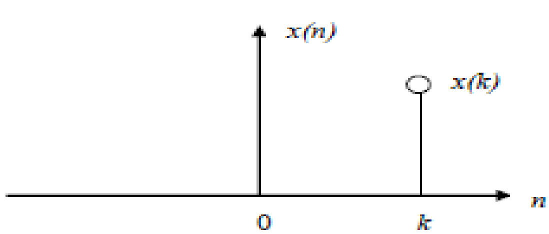
d) For any arbitrary sequence $x(n)$, we have

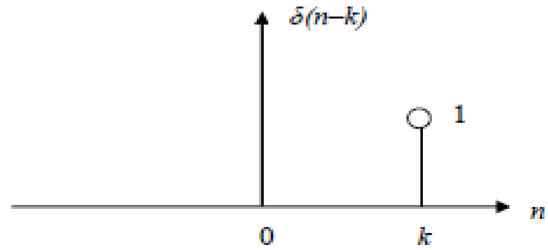
$$x(n)\,\delta(n-k) = x(k)\,\delta(n-k)$$

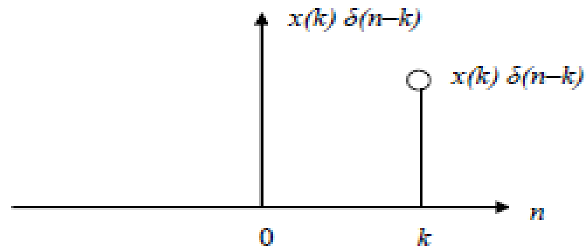
that is, the multiplication will pick out just the one value $x(k)$.

If we find the infinite sum of the above we get the sifting property:

$$\sum_{n=-\infty}^{\infty} x(n)\,\delta(n-k) = x(k)$$







e) We can write $x(n)$ as follows:

$$x(n) = \ldots + x(-1)\,\delta(n+1) + x(0)\,\delta(n) + x(1)\,\delta(n-1) + x(2)\,\delta(n-2) + \ldots$$

This can be verified to be true for all n by setting in turn

$$\ldots, n = -2, n = -1, n = 0, n = 1, n = 2, \text{etc.} \ldots$$

The above can be written compactly as

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\,\delta(n-k)$$

This is a weighted-sum of delayed unit sample functions.

3) The real exponential sequence Consider the familiar continuous time signal

$$x(t) = e^{-\alpha t} = e^{-t/\tau}, \qquad t \geq 0$$

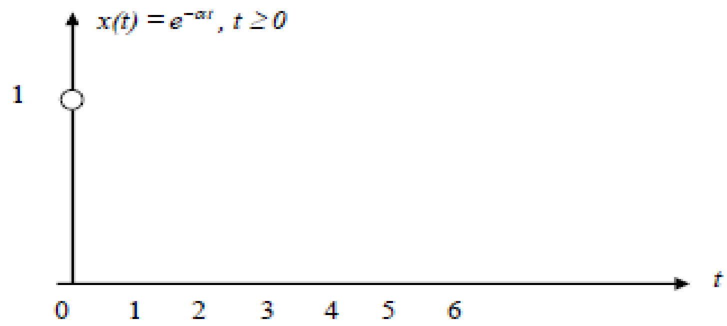
The sampled version is given by setting $t = nT$

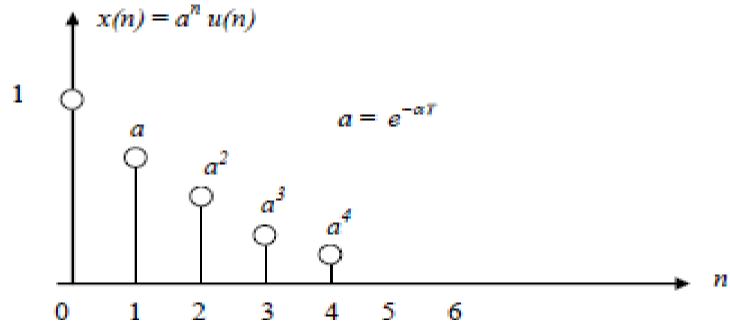
$$x(nT) = e^{-\alpha nT} = \left(e^{-\alpha T}\right)^n, \qquad nT \geq 0$$

Dropping the T from $x(nT)$ and setting $e^{-\alpha T} = a$ we can write

$$x(n) = a^n, \qquad n \geq 0$$

The sequence can also be defined for both positive and negative n, by simply writing $x(n) = a^n$ for all n.





4) The sinusoidal sequence Consider the continuous-time sinusoid $x(t)$

$$x(t) = A \sin 2\pi F_0 t = A \sin \Omega_0 t$$

F_0 and Ω_0 are the analog frequency in Hertz (or cycles per second) and radians per second, respectively. The sampled version is given by

$$x(nT) = A \sin 2\pi F_0 nT = A \sin \Omega_0 nT$$

We may drop the T from $x(nT)$ and write

$$x(n) = A \sin 2\pi F_0 nT = A \sin \Omega_0 nT, \text{ for all } n$$

We may write $\Omega_0 T = \omega_0$ which is the digital frequency in radians (per sample), so that

$$x(n) = A \sin \omega_0 n = A \sin 2\pi f_0 n, \text{ for all } n$$

Setting $\omega_0 = 2\pi f_0$ gives $f_0 = \omega_0/2\pi$ which is the digital frequency in cycles per sample. In the analog domain the horizontal axis is calibrated in seconds; "second" is one unit of the independent variable, so Ω_0 and F_0 are in "per second". In the digital domain the horizontal axis is calibrated in samples; "sample" is one unit of the independent variable, so ω_0 and f_0 are in "per sample".

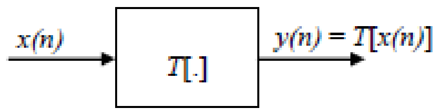
Discrete-time systems

Definition A *discrete-time system* is a mapping from the set of acceptable discrete-time signals, called the input set, to a set of discrete-time signals called the output set.

Definition A discrete-time system is *deterministic* if its output to a given input does not depend upon some random phenomenon. If it does, the system is called a *random (stochastic) system*.

Definition A *digital system* is a mapping which assigns a digital output signal to every acceptable digital input signal.

A discrete-time system can be thought of as a transformation or operator, T, that maps an input sequence $x(n)$ to an output sequence $y(n)$ shown thus:



In what follows we focus on the presence or absence of the following properties in discrete-time systems: linearity, shift invariance, causality and stability.

Filter Some refer to a linear time-invariant (LTI) system simply as a **filter**, that is, a filter is a system T with a single input and a single output signal that is both linear and time-invariant.

Linearity

Definition A discrete-time system $T[.]$ is linear if the response to a weighted sum of inputs $x_1(n)$ and $x_2(n)$ is a weighted sum (with the same weights) of the responses of the inputs separately for all weights and all acceptable inputs. Thus the system $y(n) = T[x(n)]$ is linear if for all a_1, a_2, $x_1(n)$ and $x_2(n)$ we have

$$T[a_1x_1(n)+a_2x_2(n)] = a_1T[x_1(n)] + a_2T[x_2(n)]$$

Another way of saying this is that if the inputs $x_1(n)$ and $x_2(n)$ produce the outputs $y_1(n)$ and $y_2(n)$, respectively, then the input $a_1x_1(n) + a_2x_2(n)$ produces the output $a_1 y_1(n) + a_2y_2(n)$. This is called the **superposition principle**. The a_1, a_2, $x_1(n)$ and $x_2(n)$ may be complex-valued. The above definition combines two properties, viz.,

1. **Additivity**, that is, $T[x_1(n)+x_2(n)] = T[x_1(n)] + T[x_2(n)]$, and
2. **Scaling** (or **homogeneity**), that is, $T[c\,x(n)] = c\,T[x(n)]$

The procedure of checking for linearity is:

1. Find outputs $y_1(n)$ and $y_2(n)$ corresponding to inputs $x_1(n)$ and $x_2(n)$
2. Form the sum $a_1 y_1(n) + a_2y_2(n)$
3. Find output $y_3(n)$ corresponding to input $a_1x_1(n) + a_2x_2(n)$
4. Compare the results of steps 2 and 3

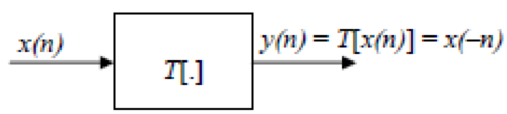
Examples of linear systems:

1. $y(n) = x(n) + x(n-1) + x(n-2)$
2. $y(n) = y(n-1) + x(n)$
3. $y(n) = 0$
4. $y(n) = n\,x(n)$ (But time-varying)

Examples of nonlinear systems:

1. $y(n) = x^2(n)$
2. $y(n) = 2\,x(n)+3$. This is a ***linear equation*** though! This system is made up of a linear part, 2 $x(n)$, and a zero-input response, 3. This is called an ***incrementally linear system***, for it responds linearly to *changes* in the input.

Example Determine if the system $y(n) = T[x(n)] = x(-n)$ is linear or nonlinear.



Answer Determine the outputs $y_1(.)$ and $y_2(.)$ corresponding to the two input sequences $x_1(n)$ and $x_2(n)$ and form the weighted sum of outputs:

$$y_1(n) = T[x_1(n)] = x_1(-n)$$
$$y_2(n) = T[x_2(n)] = x_2(-n)$$

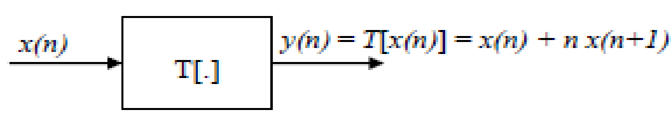
The weighted sum of outputs = $a_1\,x_1(-n) + a_2\,x_2(-n) \rightarrow$ (A).

Next determine the output y_3 due to a weighted sum of inputs:

$$y_3(n) = T[a_1\,x_1(n) + a_2\,x_2(n)] = a_1\,x_1(-n) + a_2\,x_2(-n) \rightarrow \text{(B)}$$

Check if (A) and (B) are equal. In this case (A) and (B) are equal; hence the system is linear.

Example Examine $y(n) = T[x(n)] = x(n) + n\,x(n+1)$ for linearity.



Answer The outputs due to $x_1(n)$ and $x_2(n)$ are:

$$y_1(n) = T[x_1(n)] = x_1(n) + n\,x_1(n+1)$$
$$y_2(n) = T[x_2(n)] = x_2(n) + n\,x_2(n+1)$$

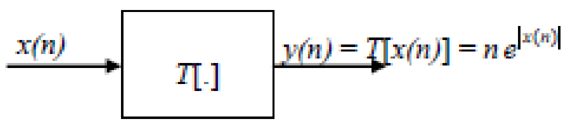
The weighted sum of outputs = $a_1\,x_1(n) + a_1\,n\,x_1(n+1) + a_2\,x_2(n) + a_2\,n\,x_2(n+1) \rightarrow$ (A)

The output due to a weighted sum of inputs is

$$\begin{aligned} y_3(n) &= T[a_1\,x_1(n) + a_2\,x_2(n)] \\ &= a_1\,x_1(n) + a_2\,x_2(n) + n\,(a_1\,x_1(n+1) + a_2\,x_2(n+1)) \\ &= a_1\,x_1(n) + a_2\,x_2(n) + n\,a_1\,x_1(n+1) + n\,a_2\,x_2(n+1) \rightarrow \text{(B)} \end{aligned}$$

Since (A) and (B) are equal the system is linear.

Example Check the system $y(n) = T[x(n)] = n\, e^{|x(n)|}$ for linearity.



Answer The outputs due $x_1(n)$ and $x_2(n)$ are:

$$y_1(n) = T[x_1(n)] = n\, e^{|x_1(n)|}$$

$$y_2(n) = T[x_2(n)] = n\, e^{|x_2(n)|}$$

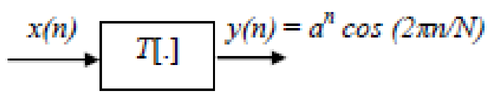
The weighted sum of the outputs $= a_1 n\, e^{|x_1(n)|} + a_2 n\, e^{|x_2(n)|} \rightarrow$ (A)

The output due to a weighted sum of inputs is

$$y_3(n) = T[a_1 x_1(n) + a_2 x_2(n)] = n\, e^{|a_1 x_1(n) + a_2 x_2(n)|} \rightarrow \text{(B)}$$

We can specify a_1, a_2, $x_1(n)$, $x_2(n)$ such that (A) and (B) are not equal. Hence nonlinear.

Example Check the system $y(n) = T[x(n)] = a^n \cos(2\pi n/N)$ for linearity.



Answer Note that the input is $x(n)$. Clearly $y(n)$ is independent of $x(n)$. The outputs due to $x_1(n)$ and $x_2(n)$ are:

$$y_1(n) = T[x_1(n)] = a^n \cos(2\pi n/N)$$

$$y_2(n) = T[x_2(n)] = a^n \cos(2\pi n/N)$$

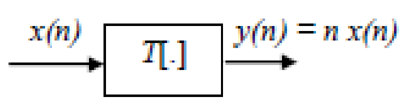
The weighted sum of the outputs $= b_1 a^n \cos(2\pi n/N) + b_2 a^n \cos(2\pi n/N) \rightarrow$ (A)

The output due to a weighted sum of inputs is

$$y_3(n) = T[b_1 x_1(n) + b_2 x_2(n)] = a^n \cos(2\pi n/N) \rightarrow \text{(B)}$$

(A) and (B) are not equal, so the system is not linear. (But (A) $= (b_1+b_2)\, a^n \cos(2\pi n/N)$ and this is equal to (B) within a constant scaling factor.)

Example Check the system $y(n) = T[x(n)] = n\, x(n)$ for linearity.



Answer For the two arbitrary inputs $x_1(n)$ and $x_2(n)$ the outputs are

$$y_1(n) = T[x_1(n)] = n\, x_1(n)$$

$$y_2(n) = T[x_2(n)] = n\, x_2(n)$$

For the weighted sum of inputs $a_1 x_1(n) + a_2 x_2(n)$ the output is

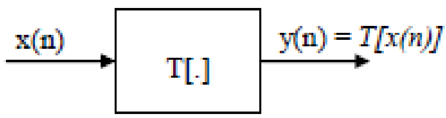
$$\begin{aligned} y_3(n) &= T[a_1 x_1(n) + a_2 x_2(n)] = n\,(a_1 x_1(n) + a_2 x_2(n)) \\ &= a_1 n\, x_1(n) + a_2 n\, x_2(n) \\ &= a_1 y_1(n) + a_2 y_2(n). \end{aligned}$$

Hence the system is linear.

Shift-invariance (time-invariance)

Definition A discrete time system $y(n) = T[x(n)]$ is shift-invariant if, for all $x(n)$ and all n_0, we have: $T[x(n-n_0)] = y(n-n_0)$.

This means that applying a time delay (or advance) to the input of a system is equivalent to applying it to the output.

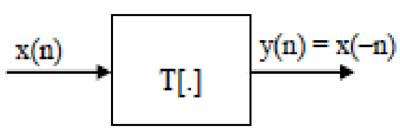


The procedure for determining shift-invariance is:

Step 1. Determine output $y(n)$ corresponding to input $x(n)$.
Step 2. Delay the output $y(n)$ by n_0 units, resulting in $y(n-n_0)$.
Step 3. Determine output $y(n, n_0)$ corresponding to input $x(n-n_0)$.
Step 4. Determine if $y(n, n_0) = y(n-n_0)$. If equal, then the system is shift-invariant; otherwise it is time-varying.

When we suspect that the system is time-varying a very useful alternative approach is to find a counter-example to disprove time-invariance, i.e., use intuition to find an input signal for which the condition of shift-invariance is violated and that suffices to show that a system is not shift-invariant.

Example Test if $y(n) = T[x(n)] = x(-n)$ is shift-invariant.



Answer Find output for x(n), delay it by n_0, and compare with the output for $x(n-n_0)$. The output for x(n) is

$$y(n) = T[x(n)] = x(-n)$$

Delaying y(n) by n_0 gives

$$y(n-n_0) = x(-(n-n_0)) = x(-n+n_0) \rightarrow (A)$$

As an aside this amounts to reflecting first and then shifting.

The output for $x(n-n_0)$ is denoted $y(n, n_0)$ and is given by

$$y(n, n_0) = T[x(n-n_0)] = x(-n-n_0) \rightarrow (B)$$

As an aside this amounts to shifting first and then reflecting.

(A) and (B) are not equal. That is, $y(n, n_0) \neq y(n-n_0)$, so the system is time-varying.

Example Examine $y(n) = T[x(n)] = x(n) + n\, x(n+1)$ for time invariance.

Answer Notice that the difference equation has a **time-varying coefficient**, n. The output y(n) corresponding to x(n) is already given above. Delaying y(n) by n_0 gives

$$y(n-n_0) = x(n-n_0) + (n-n_0)\, x(n-n_0+1) \rightarrow (A)$$

Compare with $y(n, n_0) = T[x(n-n_0)] = x(n-n_0) + n\, x(n-n_0+1) \rightarrow (B)$

(A) $\neq$ (B), so the system is time varying.

Example Check for time invariance of the system $y(n) = T[x(n)] = n\, x(n)$.

Answer We shall do this by counterexample(s) as well as by the formal procedure. The formal procedure is:

$$y(n) = T[x(n)] = n\, x(n)$$

Delay this by n_0 to get $y(n-n_0) = (n-n_0)\, x(n-n_0) \rightarrow (A)$

Compare with $y(n, n_0) = T[x(n-n_0)] = n\, x(n-n_0) \rightarrow (B)$

Since (A) $\neq$ (B), the system is time-varying.

Convolution

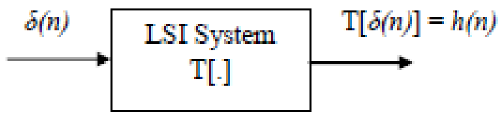
An arbitrary sequence, *x(n)*, can be written as the weighted sum of delayed unit sample functions:

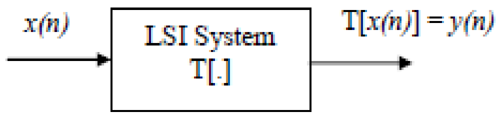
$$x(n) = \ldots + x(-2)\,\delta(n+2) + x(-1)\,\delta(n+1) + x(0)\,\delta(n) + x(1)\,\delta(n-1) + \ldots$$
$$= \sum_{k=-\infty}^{\infty} x(k)\,\delta(n-k)$$

So the response of a linear system to input *x(n)* can be written down using the linearity principle, i.e., linear superposition. For a linear shift-invariant system whose impulse response is $T[\delta(n)] = h(n)$ the reasoning goes like this

- For an input $\delta(n)$ the output is $h(n)$. For an input $x(0)\,\delta(n)$ the output is $x(0)\,h(n)$ by virtue of scaling.
- For an input $\delta(n-1)$ the output is $h(n-1)$ by virtue of shift-invariance. For an input $x(1)\,\delta(n-1)$ the output is $x(1)\,h(n-1)$ by virtue of scaling.
- Therefore for an input of $x(0)\,\delta(n) + x(1)\,\delta(n-1)$ the output is $x(0)\,h(n) + x(1)\,h(n-1)$ by virtue of additivity.

This reasoning can be extended to cover all the terms that make up $x(n)$. In general the response to $x(k)\,\delta(n-k)$ is given by $x(k)\,h(n-k)$.





Given that

$$h(n) = T[\delta(n)], \qquad \text{and} \qquad x(n) = \sum_{k=-\infty}^{\infty} x(k)\,\delta(n-k)$$

we have

$$y(n) = T[x(n)] = T\left[\sum_{k=-\infty}^{\infty} x(k)\,\delta(n-k)\right]$$

Since $T[.]$ is linear we can apply linearity a countable infinite number of times to write

$$y(n) = \sum_{k=-\infty}^{\infty} T\big[x(k)\,\delta(n-k)\big] = \sum_{k=-\infty}^{\infty} x(k)\,T[\delta(n-k)]$$

In above equation since the system is shift-invariant we write $T[\delta(n-k)] = h(n-k)$. Else write $h_k(n)$ or $h(n, k)$ in place of $h(n-k)$. Thus for a linear shift-invariant system

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

Note that if the system is not specified to be shift-invariant we would leave the above result in the form

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n,k) \qquad \text{or} \qquad y(n) = \sum_{k=-\infty}^{\infty} x(k)h_k(n)$$

Then if shift-invariance is invoked we replace $h(n, k)$ with $h(n-k)$.

As in the case of continuous-time systems, the impulse response, $h(n)$, is determined assuming that the system has no initial energy; otherwise the linearity property does not hold, so that $y(n)$, as determined using the above equation, corresponds to only the forced response of the system.

The sum $\sum_{k=-\infty}^{\infty} x(k)h(n,k)$ is called the **convolution sum**, and is denoted $x(n) * h(n)$.

A *discrete-time linear shift-invariant system is completely characterized* by its unit sample response $h(n)$.

Theorem If a discrete-time system linear shift-invariant, T[.], has the unit sample response $T[\delta(n)] = h(n)$ then the output $y(n)$ corresponding to any input $x(n)$ is given by

$$y(n) = \underbrace{\sum_{k=-\infty}^{\infty} x(k)\,h(n-k)}_{= x(n) * h(n)} = \underbrace{\sum_{k=-\infty}^{\infty} x(n-k)\,h(k)}_{= h(n) * x(n)}$$

The second summation is obtained by setting $m = n-k$; then for $k = -\infty$ we have $m = +\infty$, and for $k = \infty$ we have $m = -\infty$. Thus

$$\sum_{k=-\infty}^{\infty} x(k)\,h(n-k) = \underbrace{\sum_{m=\infty}^{-\infty} x(n-m)\,h(m)} = \sum_{k=-\infty}^{\infty} x(n-k)\,h(k)$$

m is a dummy variable. The order of summation (forward or backward) makes no difference. Hence change *m* to *k* and switch limits

Example **[Linear Convolution]** Given the input $\{x(n)\} = \{1, 2, 3, 1\}$ and the unit sample response $\{h(n)\} = \{4, 3, 2, 1\}$ find the response $y(n) = x(n) * h(n)$.

Answer Since $x(k) = 0$ for $k < 0$ and $h(n-k) = 0$ for $k > n$, the convolution sum becomes

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

Now $y(n)$ can be evaluated for various values of n; for example, setting $n = 0$ gives $y(0)$. See table below. The product terms shown in ***bold italics*** need not be calculated; they are zero because the signal values involved are zero.

Linear Convolution of {x(n)} ={1, 2, 3, 1} and {h(n)} = {4, 3, 2, 1}

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

n = 0	$y(0) = \sum_{k=0}^{0} x(k)\,h(0-k)$	= x(0) h(0) = 1 . 4 = 4
n = 1	$y(1) = \sum_{k=0}^{1} x(k)\,h(1-k)$	= x(0) h(1) + x(1) h(0) = 1 . 3 + 2 . 4 = 11
n = 2	$y(2) = \sum_{k=0}^{2} x(k)\,h(2-k)$	= x(0) h(2) + x(1) h(1) + x(2) h(0) = 1 . 2 + 2 . 3 + 3 . 4 = 20
n = 3	$y(3) = \sum_{k=0}^{3} x(k)\,h(3-k)$	= x(0) h(3) + x(1) h(2) + x(2) h(1) + x(3) h(0) = 1 . 1 + 2 . 2 + 3 . 3 + 1 . 4 = 18
n = 4	$y(4) = \sum_{k=0}^{4} x(k)\,h(4-k)$	= x(0) ***h(4)*** + x(1) h(3) + x(2) h(2) + x(3) h(1) + ***x(4)*** h(0) = 1 . 0 + 2 . 1 + 3 . 2 + 1 . 3 + 0 . 4 = 11
n = 5	$y(5) = \sum_{k=0}^{5} x(k)\,h(5-k)$	= ***x(0) h(5)*** + ***x(1) h(4)*** + x(2) h(3) + x(3) h(2) + ***x(4) h(1)*** + ***x(5) h(0)*** = 3 . 1 + 1 . 2 = 5
n = 6	$y(6) = \sum_{k=0}^{6} x(k)\,h(6-k)$	= ***x(0) h(6)*** + ***x(1) h(5)*** + ***x(2) h(4)*** + x(3) h(3) + ***x(4) h(2)*** + ***x(5) h(1)*** + ***x(6) h(0)*** = 1 . 1 = 1
n = 7	$y(7) = \sum_{k=0}^{7} x(k)\,h(7-k)$	= ***x(0) h(7)*** + ***x(1) h(6)*** + ***x(2) h(5)*** + ***x(3) h(4)*** + ***x(4) h(3)*** + ***x(5) h(2)*** + ***x(6) h(1)*** + ***x(7) h(0)*** = 0
y(n) = 0 for n < 0 and n > 6		

Causality

The constraints of linearity and time-invariance define a class of systems that is represented by the convolution sum. The additional constraints of stability and causality define a more restricted class of linear time-invariant systems of practical importance.

Definition A discrete-time system is **causal** if the output at $n = n_0$ depends only on the input for $n \leq n_0$.

The word "causal" has to do with cause and effect; in other words, for the system to act up there must be an actual cause. A causal system does not anticipate future values of the input but only responds to actual, present, input. As a result, if two inputs to a causal system are identical up to some point in time n_0 the corresponding outputs must also be equal up to this same time. The synonyms of "causal" are "(physically) realizable" and "non-anticipatory".

We digress below to introduce memory-less versus dynamic systems and then resume with causality.

Systems with and without memory A system is said to be **memory-less** or **static** if its output for each value of n is dependent only on the input at that same time but not on *past* or *future* inputs.

Examples of static systems

1. $y(n) = x(n)$ $\rightarrow$ the identity system
2. $y(n) = a\,x(n) - x^2(n)$
3. A resistor R: $y(t) = R\,x(t)$ ($y(t)$ is voltage and $x(t)$ is current)

In many physical systems, memory is directly associated with storage of energy. A resistor has no storage of energy. But a circuit with capacitors and/or inductors has storage of energy and is a **dynamic** system, i.e., has **memory**. However, while storage of energy has to do with past inputs only, a static system is independent not only of *past* but also of *future* inputs.

Examples of systems with memory, i.e., dynamic systems:

1. $y(n) = \sum_{k=-\infty}^{n} x(k)$. This is an accumulator or summer. The output $y(n)$ depends on values of $x(.)$ prior to n such as $x(n-1)$ etc.
2. $y(n) = x(n-1)$. This is a delay element.
3. A capacitor C: $y(t) = \dfrac{1}{C}\displaystyle\int_{-\infty}^{t} x(\tau)\,d\tau$, ($y(.)$ is voltage and $x(.)$ is current).

Getting back to causality, *all memory-less systems are causal* since the output responds only to the current value of the input. In addition, some dynamic systems (such as the three listed above) are also causal.

An example of a noncausal system is $y(n) = x(n) + x(n+1)$ since the output depends on a future value, $x(n+1)$.

Although causal systems are of great importance, they are not the only systems that are of practical importance. For example, causality is not often an essential constraint in applications in which the *independent variable is not time*, such as in image processing. Moreover, in processing data that have been *recorded previously* (non real-time), as often happens with speech, geophysical, or meteorological signals, to name a few, we are by no means constrained to causal processing. As another example, in many applications, including *historical* stock market analysis and demographic studies, we may be interested in determining a slowly varying trend in data that also contain higher frequency fluctuations about that trend. In this case, a commonly used approach is to average data over an interval in order to smooth out the fluctuations and keep only the trend. An example of such a noncausal averaging system is

$$y(n) = \frac{1}{2M+1} \sum_{k=-M}^{M} x(k)$$

Definition A discrete-time sequence $x(n)$ is called causal if it has zero values for $n < 0$, i.e., $x(n) = 0$ for $n < 0$.

Theorem A *linear shift-invariant system* with impulse response $h(n)$ is causal if and only if $h(n)$ is zero for $n < 0$.

Proof By convolution the output $y(n)$ is given by

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)\, h(n-k)$$

If $h(n) = 0$ for $n < 0$, then $h(n-k) = 0$ for $n-k < 0$ or $k > n$. So

$$y(n) = \sum_{k=-\infty}^{n} x(k)\, h(n-k) + \underbrace{\sum_{k=n+1}^{\infty} x(k) h(n-k)}_{=0}$$

$$= \sum_{k=-\infty}^{n} x(k)\, h(n-k)$$

Thus $y(n)$ at any time n is a weighted sum of the values of the input $x(k)$ for $k \le n$, that is, only the present and past inputs. Therefore, the system is causal.

Bounded input bounded output stability

Definition A sequence $x(n)$ is bounded if there exists a finite M such that $|x(n)| < M$ for all n. (Note that, as expressed here, M is a bound for negative values of $x(.)$ as well. Another way of writing this is $-M < x(n) < M$.)

As an example, the sequence $x(n) = [1+\cos 5\pi n]\, u(n)$ is bounded with $|x(n)| \le 2$. The sequence $x(n) = \left[\frac{(1+n)\sin 10n}{1+(0.8)^n}\right] u(n)$ is unbounded.

Definition A discrete-time system is bounded input-bounded output (BIBO) stable if every bounded input sequence *x(n)* produces a bounded output sequence. That is, if $|x(n)| \leq M < \infty$, then $|y(n)| \leq L < \infty$.

BIBO stability theorem A *linear shift invariant system* with impulse response *h(n)* is bounded input-bounded output stable if and only if *S*, defined below, is finite.

$$S = \sum_{k=-\infty}^{\infty} |h(k)| < \infty$$

i.e., the unit sample response is *absolutely summable*.

Proof Given a system with impulse response *h(n)*, let *x(n)* be such that $|x(n)| \leq M$. Then the output *y(n)* is given by the convolution sum:

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k)$$

so that

$$|y(n)| = \left| \sum_{k=-\infty}^{\infty} h(k)\,x(n-k) \right|$$

Using the triangular inequality that the sum of the magnitudes $\geq$ the magnitude of the sum, we get

$$|y(n)| \leq \sum_{k=-\infty}^{\infty} |h(k)x(n-k)|$$

Using the fact that the magnitude of a product is the product of the magnitudes,

$$|y(n)| \leq \sum_{k=-\infty}^{\infty} |h(k)||x(n-k)|$$

$$\leq M \sum_{k=-\infty}^{\infty} |h(k)|$$

Thus, a sufficient condition for the system to be stable is that the unit sample response must be absolutely summable; that is,

$$\sum_{k=-\infty}^{\infty} |h(k)| < \infty$$

Example Evaluate the stability of the linear shift-invariant system with the unit sample response $h(n) = a^n u(n)$.

Answer Evaluate

$$S = \sum_{k=-\infty}^{\infty} |h(k)| = \sum_{k=-\infty}^{\infty} |a^k u(k)| = \sum_{k=0}^{\infty} |a^k| = \sum_{k=0}^{\infty} |a|^k$$

Here we have used the fact that the magnitude of a product ($|a^k|$) is the product of the magnitudes ($|a|^k$). The summation on the right converges if $|a| < 1$ so that *S* is finite,

$$S = \frac{1}{1-|a|}$$

and the system is BIBO stable.

Fourier analysis of discrete-time signals and systems

Note For the discrete-time Fourier transform some authors (Oppenheim & Schafer, for instance) use the symbol $X(e^{j\omega})$ while others (Proakis, for instance) use the symbol $X(\omega)$. The symbol ω is used for digital frequency (radians per sample or just radians) and the symbol Ω for the analog frequency (radians/sec). Some authors, on the other hand, use just the opposite of our convention, that is, ω for the analog frequency (radians/sec) and Ω for the digital frequency (radians).

Discrete-time Fourier transform (DTFT) For the continuous-time signal $x(t)$, the Fourier transform is

$$\mathcal{F}\{x(t)\} = X(\Omega) = \int_{-\infty}^{\infty} x(t)\, e^{-j\Omega t}\, dt$$

The *impulse-train* sampled version, $x_s(t)$, is given by

$$x_s(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT)$$

So the Fourier transform of $x_s(t)$ is given by

$$X_s(\Omega) = \int_{-\infty}^{\infty} x_s(t)\, e^{-j\Omega t}\, dt = \int_{-\infty}^{\infty} \left(x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT) \right) e^{-j\Omega t}\, dt$$

$$= \sum_{n=-\infty}^{\infty} x(nT)\, e^{-j\Omega nT}$$

where the last step follows from the sifting property of the δ function. Replace ΩT by ω the discrete-time frequency variable, that is, the **digital frequency**. Note that Ω has units of radians/second, and ω has units of radians (/sample). This change of notation gives the **discrete-time Fourier transform**, $X(\omega)$, of the discrete-time signal $x(n)$, obtained by sampling $x(t)$, as

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

Note that this defines the discrete-time Fourier transforms of *any* discrete-time signal $x(n)$. The transform exists if $x(n)$ satisfies a relation of the type

$$\sum_{n=-\infty}^{\infty} |x(n)| < \infty \qquad \text{or} \qquad \sum_{n=-\infty}^{\infty} |x(n)|^2 < \infty$$

These conditions are sufficient to guarantee that the sequence has a discrete-time Fourier transform. As in the case of continuous-time signals there are signals that neither are absolutely summable nor have finite energy, but still have a discrete-time Fourier transform.

Discrete-time Fourier transform of (non-periodic) sequences The Fourier transform of a general discrete-time sequence tells us what the frequency content of that signal is.

Definition The Fourier transform $X(e^{j\omega})$ of the sequence $x(n)$ is given by

$$\mathcal{F}\{x(n)\} = X(\omega) = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j\omega n} \qquad \rightarrow \text{(A)}$$

The inverse Fourier transform is given by

$$\mathcal{F}^{-1}\{X(\omega)\} = x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}d\omega \qquad \rightarrow \text{(B)}$$

Equations (A) and (B) are called the Fourier transform pair for a sequence *x(n)* with *X(ω)* thought of as the *frequency content* of the sequence *x(n)*. Equation (A) is the analysis equation and equation (B) is the synthesis equation. Since *X(ω)* is a periodic function of *ω*, we can think of *x(n)* as the Fourier coefficients in the Fourier series representation of *X(ω)*. That is, equation (A), in fact, expresses *X(ω)* in the form of a Fourier series.

Example For the exponential sequence $x(n) = a^n u(n)$, $|a| < 1$, the DTFT is

$$X(e^{j\omega}) = \sum_{n=0}^{\infty} a^n e^{-j\omega n} = \sum_{n=0}^{\infty}\left(ae^{-j\omega}\right)^n = \frac{1}{1-ae^{-j\omega}} = \frac{1}{1-a(\cos\omega - j\sin\omega)}$$

We shall put this in the form $X(\omega)=Magnitude\ \{X\}\ e^{j\,Phase\{X\}} = |X(\omega)|\ e^{j\angle X(\omega)}$ from which the magnitude and phase will be extracted. The denominator (Dr.) is

$$\text{Dr.} = 1 - a\cos\omega + ja\sin\omega = \sqrt{(1-a\cos\omega)^2 + a^2\sin^2\omega}\ \ e^{j\tan^{-1}\left(\frac{a\sin\omega}{1-a\cos\omega}\right)}$$

Thus

$$X(\omega) = |X(\omega)|\ e^{j\angle X(\omega)} = \frac{1}{\sqrt{(1+a^2-2a\cos\omega)}}\ e^{-j\tan^{-1}\left(\frac{a\sin\omega}{1-a\cos\omega}\right)}$$

The magnitude and phase are:

$$|X(\omega)| = \frac{1}{\sqrt{(1+a^2-2a\cos\omega)}} \qquad \text{and} \qquad \angle X(\omega) = -\tan^{-1}\left(\frac{a\sin\omega}{1-a\cos\omega}\right)$$

Plots of $|X|$ and $\angle X$ are shown. Note that *X(ω)* is periodic and that the magnitude is an even function of *ω* and the phase is an odd function. (See below on the notation $|X|$ and $\angle X$).

The value of $X(e^{j\omega})$ at $\omega = 0$ is

$$|X(\omega)|_{\omega=0} = \frac{1}{\sqrt{(1+a^2-2a\cos 0)}} = \frac{1}{1-a}$$

$$\angle X(\omega)|_{\omega=0} = -\tan^{-1}\left(\frac{a\sin 0}{1-a\cos 0}\right) = 0$$

Similarly, at $\omega = \pi$ we have $|X(\omega)|_{\omega=\pi} = 1/(1+a)$ and $\angle X(\omega)|_{\omega=\pi} = 0$.

Frequency response of discrete-time system

For a linear shift-invariant system with impulse response $h(n)$, the Fourier transform $H(\omega)$ gives the *frequency response*. Consider the input sequence $x(n) = e^{j\omega n}$ for $-\infty < n < \infty$, i.e., a complex exponential of radian frequency ω and magnitude 1, applied to a linear shift-invariant system whose unit sample response is $h(n)$. Using convolution we obtain the output $y(n)$ as

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

$$= H(\omega)\, e^{j\omega n}$$

Thus we see that $H(\omega)$ describes the change in complex amplitude of a complex exponential as a function of frequency. The quantity $H(\omega)$ is called the **frequency response** of the system. In general, $H(\omega)$ is complex valued and may be expressed either in the Cartesian form or the polar from as

$$H(\omega) = H_R(\omega) + j\, H_I(\omega) \qquad \text{or} \qquad H(\omega) = \hat{H}(\omega)\, e^{j\angle H(\omega)}$$

where H_R and H_I are the real part and imaginary part respectively. $\hat{H}(\omega)$ is loosely called the magnitude and $\angle H(\omega)$ is loosely called the phase. Strictly speaking, $\hat{H}(\omega)$ is called the **zero-phase frequency response**; note that $\hat{H}(\omega)$ is *real valued* but may be positive or negative. We may use the symbol $|H(\omega)|$ for the *magnitude* which is strictly non-negative. If $\hat{H}(\omega)$ is positive then

$$\text{Magnitude} = |H(\omega)| = \hat{H}(\omega) \qquad \& \qquad \text{Phase} = \angle H(\omega)$$

If $\hat{H}(\omega)$ is negative then

$$\text{Magnitude} = |H(\omega)| = |\hat{H}(\omega)| = -\hat{H}(\omega) \qquad \& \qquad \text{Phase} = \angle H(\omega) \pm \pi$$

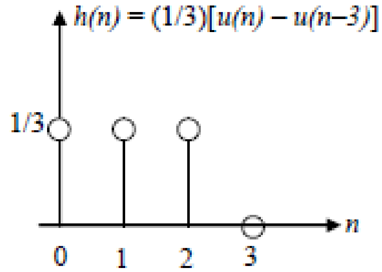
We shall often loosely use the symbol $|H(\omega)|$ to refer to $\hat{H}(\omega)$ as well with the understanding that when the latter is negative we shall take its absolute value (the magnitude) and accordingly adjust $\angle H(\omega)$ by $\pm\pi$.

Example **[Moving average filter]** The impulse response of the LTI system

$$y(n) = \frac{x(n) + x(n-1) + x(n-2)}{3}$$

is

$$h(n) = \begin{cases} 1/3, & n = 0, 1, 2 \\ 0, & \text{otherwise} \end{cases}$$



The frequency response is obtained below.

$$H(\omega) = \sum_{k=-\infty}^{\infty} h(k)e^{-j\omega k} = \sum_{k=0}^{2}(1/3)e^{-j\omega k} = \frac{1}{3}\left(e^{-j\omega 0} + e^{-j\omega 1} + e^{-j\omega 2}\right)$$

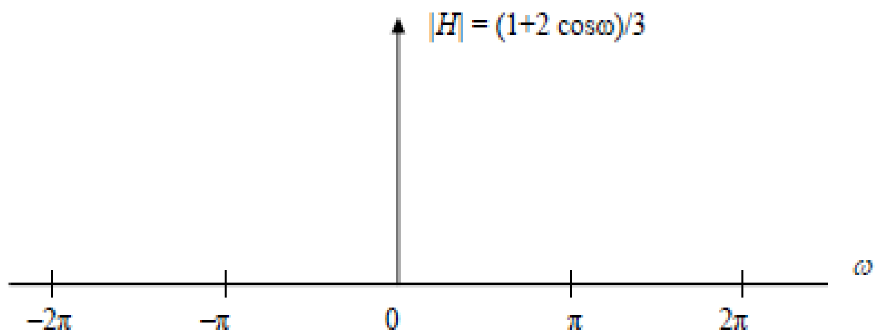
$$= \frac{e^{-j\omega}}{3}\left(e^{j\omega} + 1 + e^{-j\omega}\right) = \frac{e^{-j\omega}}{3}\left(1 + \frac{2\left(e^{j\omega} + e^{-j\omega}\right)}{2}\right) = \frac{(1+2\cos\omega)}{3}\,e^{-j\omega}$$

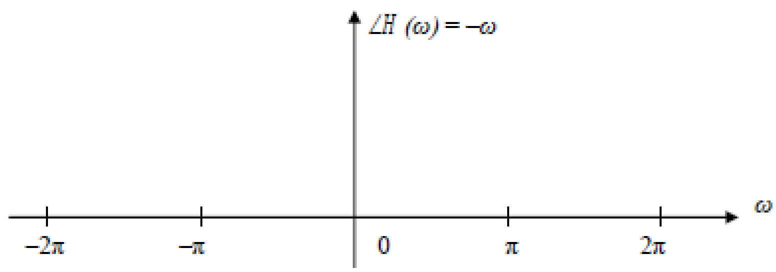
which is already in the polar form $H(\omega) = \pm\left|H(\omega)\right| e^{j\angle H(\omega)}$, so that

$$\left|H(\omega)\right| = (1+2\cos\omega)/3 \qquad \text{and} \qquad \angle H(\omega) = -\omega$$

The zero crossings of the magnitude plot occur where $\left|H(\omega)\right| = (1+2\cos\omega)/3 = 0$, or $\omega = \cos^{-1}(-1/2) = 2\pi/3 = 120^0$. A frequency of $\omega = 2\pi/3$ rad./sample ($f = 1/3$ cycle/sample) is totally stopped (filtered out) by the filter. The corresponding digital signal is $x_5(n) = \cos 2\pi(1/3)n$. The underlying continuous-time signal, $x_5(t)$, depends on the sampling frequency. If, for example, the sampling frequency is 16Hz, then $x_5(t) = \cos 2\pi(16/3)t$, and a frequency of 16/3 Hz will be totally filtered out. If the sampling frequency is 150Hz, then $x_5(t) = \cos 2\pi(150/3)t$, and a frequency of 50 Hz will be eliminated.

In calibrating the horizontal axis in terms of the cyclic frequency, F, we use the relation $\omega = \Omega T = 2\pi FT = 2\pi F/F_s$ from which the point $\omega = 2\pi$ corresponds to $F = F_s$.




∠H (ω) = −ω
ω
−2π
−π
0
π
2π

4.11 Realization of digital filters

Given $H(z)$, the system function, or $h(n)$, the impulse response, the difference equation may be obtained. This difference equation could be implemented by *computer program*, *special purpose digital circuitry*, or *special programmable integrated circuit*. This direct evaluation of the difference equation is not the only possible realization of the digital filter. Alternative realizations of the digital filter are possible by breaking up the direct realization in some form.

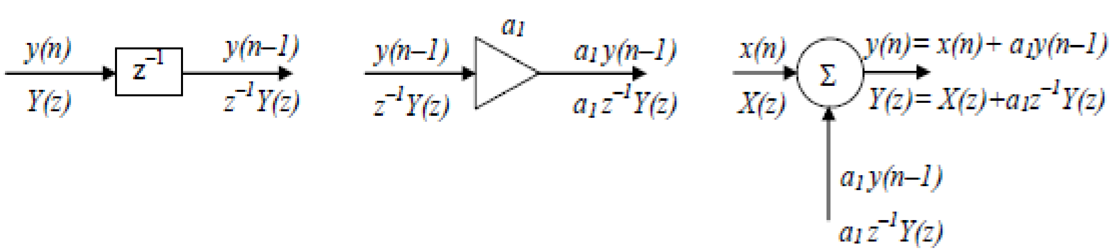
Direct Form realization of IIR filters An important class of linear shift invariant systems can be characterized by the following rational system function (where $X(z)$ is the input, $Y(z)$ the output and we have taken $a_0 = 1$ in comparison with the earlier representation):

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \dots\dots\dots + b_{M-1} z^{-M+1} + b_M z^{-M}}{1 + a_1 z^{-1} + \dots\dots\dots + a_{N-1} z^{-N+1} + a_N z^{-N}} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

By cross multiplying and taking the inverse z-transform we get the difference equation

$$\begin{aligned} y(n) &= -\sum_{k=1}^{N} a_k y(n-k) + \sum_{r=0}^{M} b_r x(n-r) \\ &= -a_1 y(n-1) - a_2 y(n-2) - \dots - a_N y(n-N) \\ &\qquad + b_0 x(n) + b_1 x(n-1) + \dots + b_M x(n-M) \end{aligned}$$

To construct a filter structure we shall need three types of block diagram elements: a delay element, a multiplier and an adder, illustrated below:



We can construct a realization of the filter called the Direct Form I by starting with $y(n)$ and generating all the delayed versions $y(n-1), y(n-2), \dots, y(n-N)$; similarly starting with $x(n)$ and generating all the delayed versions $x(n-1), x(n-2), \dots, x(n-M)$. We then multiply the above terms by the respective coefficients and add them up. This is shown below (next page).

This is an N^{th} order system N being the order of the difference equation. There is no restriction as to whether M should be less than or greater than or equal to N. The total number of delay elements = $(N+M)$. It is not in canonic form because it uses more than the minimum possible number of delay elements. It is called "Direct Form" because the multipliers are the actual filter coefficients $\{a_1, a_2, \dots, a_N, b_0, b_1, b_2, \dots, b_M\}$.

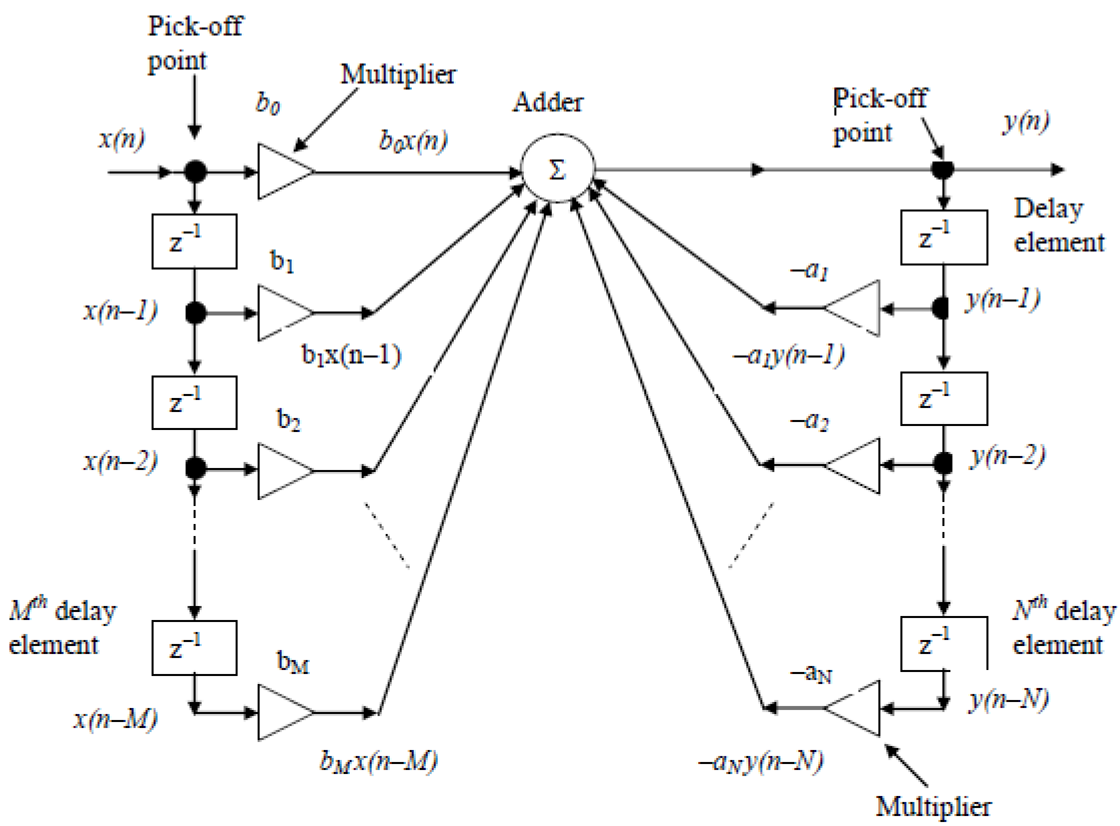
The difference equation of this realization (or structure) continues to be

$$y(n) = \underbrace{-a_1 y(n-1) - a_2 y(n-2) - \ldots - a_N y(n-N)}_{N \text{ multiplications}}$$

$$\underbrace{+ b_0 x(n) + b_1 x(n-1) + \ldots + b_M x(n-M)}_{(M+1) \text{ multiplications}}$$

and will be referred to as the Direct Form I difference equation. The total number of multiplications can be counted and is seen to be $(N+M+1)$. We can also count and see that there are $(N+M)$ additions. Finally, to calculate the value $y(n)$ we need to store N past values of $y(.)$, and M past values of $x(.)$, that is, a total of $(N+M)$ storage locations (storage for the present value of $x(.)$ is not counted).

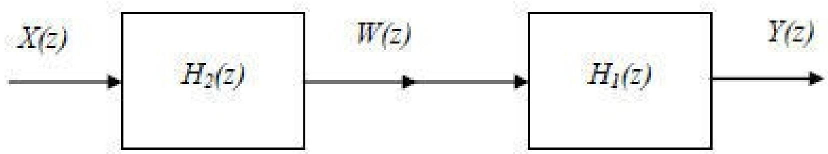
Direct Form I



Rearrangement of Direct Form I The above diagram of Direct Form I, or the corresponding expression for $H(z)$, is sometimes rearranged as below. This shows visually that the transfer function $H(z)$ is arranged as a cascade of an *all-zero system*, $H_2(z)$, followed by an *all-pole system*, $H_1(z)$:

$$H(z) = \frac{Y(z)}{X(z)} = \underbrace{\left(\sum_{k=0}^{M} b_k z^{-k}\right)}_{H_2(z)} \underbrace{\left(\frac{1}{1+\sum_{k=1}^{N} a_k z^{-k}}\right)}_{H_1(z)} = H_2(z)\, H_1(z)$$

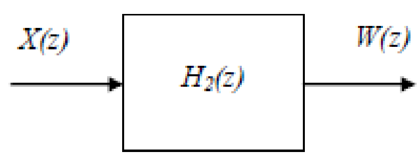
The overall block diagram then is shown thus:



The all-zero system is $H_2(z) = \dfrac{W(z)}{X(z)} = \left(\sum_{k=0}^{M} b_k z^{-k}\right)$ from which, by cross-multiplying and taking the inverse z-transform, we get the difference equation below:

$$W(z) = H_2(z)\, X(z) = X(z)\left(\sum_{k=0}^{M} b_k z^{-k}\right)$$

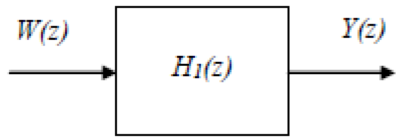
$$w(n) = b_0\, x(n) + b_1\, x(n-1) + \ldots + b_M\, x(n-M)$$



The all-pole system is $H_1(z) = \dfrac{Y(z)}{W(z)} = \left(\dfrac{1}{1+\sum_{k=1}^{N} a_k z^{-k}}\right)$ from which, by cross-multiplying and taking the inverse z-transform, we get the difference equation as below:

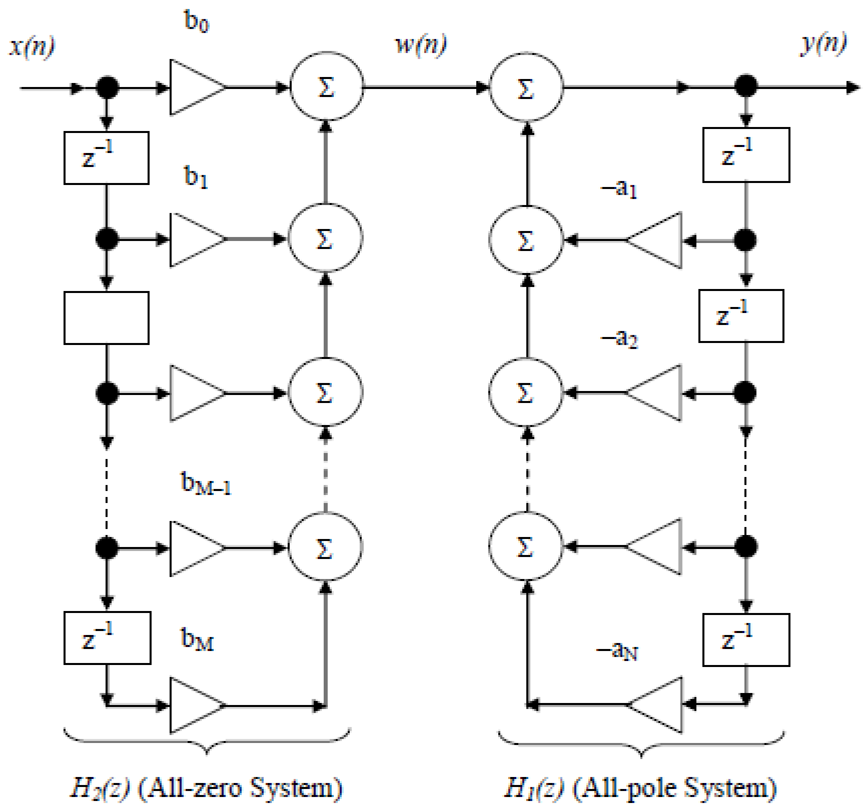
$$Y(z) = H_1(z)\, W(z) = W(z)\left(\frac{1}{1+\sum_{k=1}^{N} a_k z^{-k}}\right)$$

$$y(n) = w(n) - a_1\, y(n-1) - a_2\, y(n-2) - \ldots - a_N\, y(n-N)$$



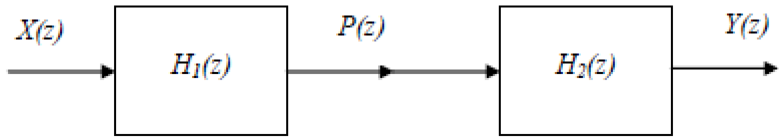
Even though it seems that there are two equations, one for *w(n)* and another for *y(n)*, there is, in effect, only one since *w(n)* in the second equation is simply a short hand notation for the first equation and can be eliminated from the equation for *y(n)*.

Overall, the Direct Form I has the following alternative appearance:



Derivation of Direct Form II The transfer function *H(z)* can be written as the product of the two transfer functions $H_1(z)$ and $H_2(z)$ as follows where we have reversed the sequence of the two blocks:

$$H(z) = \frac{Y(z)}{X(z)} = \underbrace{\left(\frac{1}{1+\sum_{k=1}^{N} a_k z^{-k}}\right)}_{H_1(z)} \underbrace{\left(\sum_{k=0}^{M} b_k z^{-k}\right)}_{H_2(z)} = H_1(z)\, H_2(z)$$

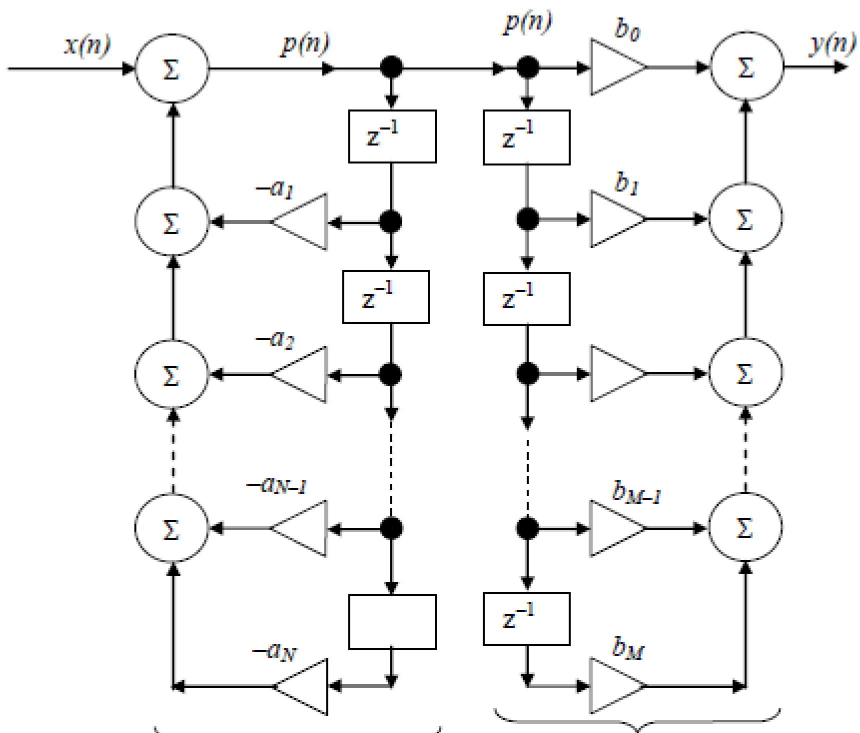


$$P(z) = H_1(z)\,X(z) = \left(\frac{1}{1+\sum_{k=1}^{N} a_k z^{-k}}\right) X(z) \qquad \text{and} \qquad Y(z) = H_2(z)\,P(z) = \left(\sum_{k=0}^{M} b_k z^{-k}\right) P(z)$$

Cross-multiplying, taking the inverse z-transform of the above two and rearranging, we have

$$p(n) = x(n) - \sum_{k=1}^{N} a_k\, p(n-k), \text{ and } \qquad y(n) = \sum_{r=0}^{M} b_r\, p(n-r)$$

The two equations are realized as below:

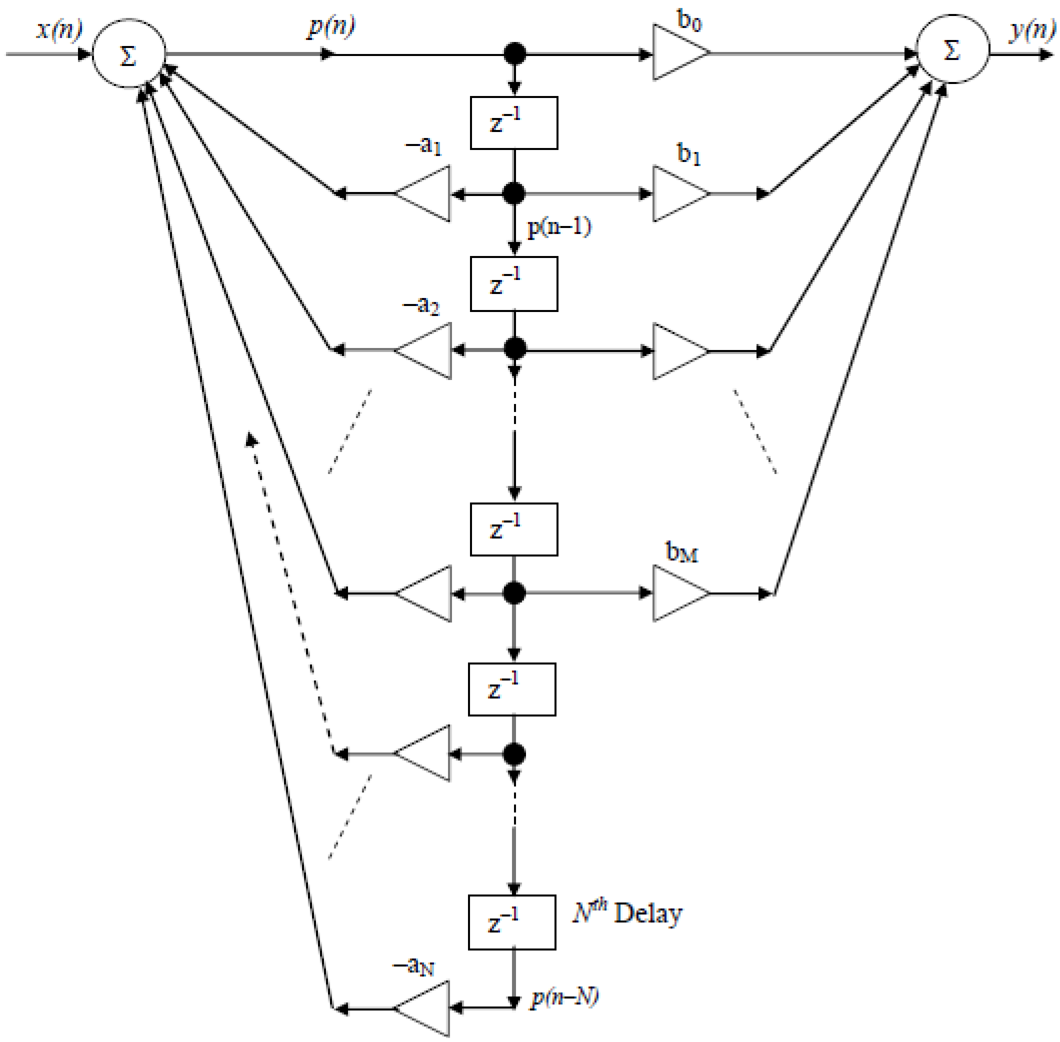


$H_1(z)$ (All-pole System) $H_2(z)$ (All-zero System)

The two branches of delay elements in the middle of the above block diagram can be replaced by just one branch containing either N or M (whichever is larger) delay elements, resulting in the Direct Form II shown below:

In the above diagram each column of adders on each side can be replaced by a single adder resulting in the more familiar form shown below. There are now only two adders.

Direct Form II



The number of delay elements = max {N, M} – this is the minimum possible, hence called *a* **canonic form**. The multipliers are the actual coefficients from the difference equation. Hence this is also a direct form.

The numbers of multipliers and adders are also the minimum possible, but this does not mean that it is the best realization from other considerations like immunity to round off and quantization errors.

The difference equations are:

$p(n) = x(n) - a_1\, p(n-1) - a_2\, p(n-2) - \ldots - a_N\, p(n-N)$, and

$y(n) = b_0\, p(n) + b_1\, p(n-1) + \ldots + b_M\, p(n-M)$

The above equations show that in order to generate $y(n)$ we need the present value of $x(.)$ and N (or M or whichever is larger) past values of $p(.)$. This requires N storage locations not counting the present value of $x(.)$. We also see that the number of multiplications = $N+M+1$, and number of additions = $N+M$.

Comparing the difference equations of Direct Forms I and II:

- To compute $y(n)$ in DF I we need the past N outputs, the present input, and the past M inputs.
- To compute $y(n)$ in DF II we need the N (or M) values of $p(n-k)$ for $k = 1,2, \ldots, N$, and the present input.

This illustrates the concept of the state of a system.

Problem:

Develop a canonic direct form realization of the transfer function

$$H(z) = \frac{6z^5 + 8z^3 - 4}{2z^5 + 6z^4 + 10z^3 + 8z}$$

Solution Write numerator and denominator as polynomials in negative powers of z with the leading term (a_0) in the denominator equal to 1

$$H(z) = \frac{z^5(6+8z^{-2}-4z^{-5})}{z^5(2+6z^{-1}+10z^{-2}+8z^{-4})} = \frac{(6+8z^{-2}-4z^{-5})}{(2+6z^{-1}+10z^{-2}+8z^{-4})}$$

$$= \frac{(6+8z^{-2}-4z^{-5})}{2(1+3z^{-1}+5z^{-2}+4z^{-4})} = \frac{3+4z^{-2}-2z^{-5}}{1+3z^{-1}+5z^{-2}+4z^{-4}}$$

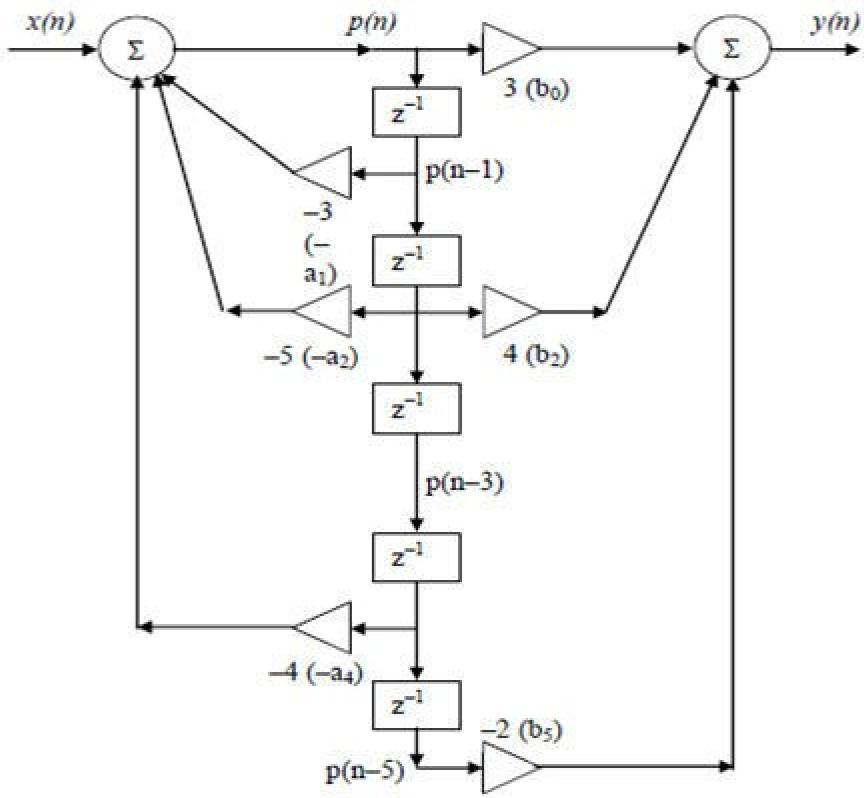
Making the following comparison with the standard notation

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4} + b_5 z^{-5}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}} = \frac{3+4z^{-2}-2z^{-5}}{1+3z^{-1}+5z^{-2}+4z^{-4}}$$

we identify the following parameters:

$b_0 = 3,\ b_1 = 0,\ b_2 = 4,\ b_3 = 0,\ b_4 = 0,\ b_5 = -2$

$a_1 = 3,\ a_2 = 5,\ a_3 = 0,\ a_4 = 4$



Problem:

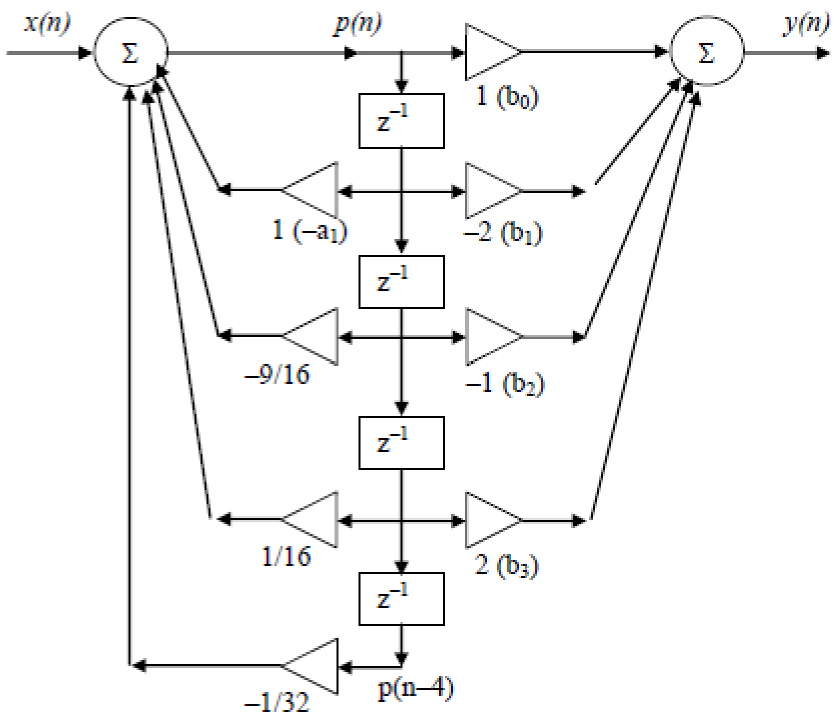
A system is specified by its transfer function as

$$H(z) = \frac{(z-1)(z-2)(z+1)z}{\left[z-\left(\frac{1}{2}+j\frac{1}{2}\right)\right]\left[z-\left(\frac{1}{2}-j\frac{1}{2}\right)\right]\left[z-j\frac{1}{4}\right]\left[z+j\frac{1}{4}\right]}$$

Realize the system in the following forms: (a) Direct Form I, and (b) Direct Form II.

Solution We need to express $H(z)$ as a ratio of polynomials in negative powers of z with the leading term (a_0) in the denominator equal to 1. Multiplying out the factors in the numerator and denominator and rearranging

$$H(z) = \frac{(z^2-1)(z^2-2z)}{\left[z^2-z+\frac{1}{2}\right]\left[z^2+16\right]} = \frac{1-2z^{-1}-z^{-2}+2z^{-3}}{1-z^{-1}+\frac{9}{16}z^{-2}-\frac{1}{16}z^{-3}+\frac{1}{32}z^{-4}}$$



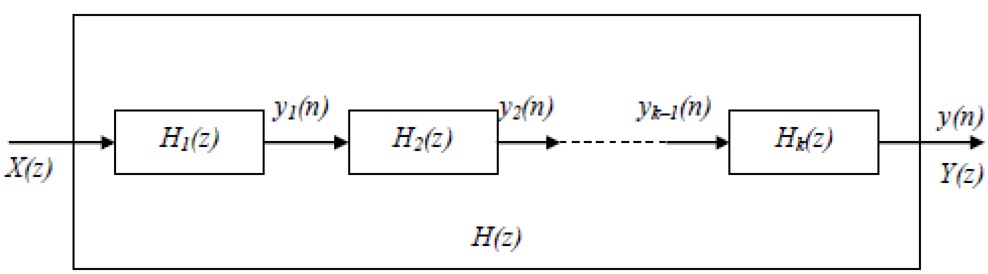
Cascade realization of IIR filters Many different realizations exist depending on how we choose to write and rearrange the given transfer function. Two very important ways of decomposing the transfer function are the *cascade* and *parallel* decompositions.

In the cascade realization $H(z)$ is broken up into a product of transfer functions H_1, H_2, ..., H_k, each a rational expression in z^{-1} as follows:

$$\frac{Y(z)}{X(z)} = H(z) = H_k(z)\, H_{k-1}(z)\, \dots\, H_2(z)\, H_1(z)$$

so that $Y(z)$ can be written as

$$Y(z) = H_k(z)\, H_{k-1}(z)\, \dots\, H_2(z)\, H_1(z)\, X(z)$$



Although $H(z)$ could be broken up in many different ways, the most common cascade realization is to require each of the k product H's to be a *biquadratic section*. In many cases the design procedure yields a product of biquadratic expressions so no further work is necessary to put $H(z)$ in the required form. The product terms $H_i(z)$ could take various forms, depending on the actual problem. Some possible forms are

$$H_i(z) = \frac{b_0}{1+a_1z^{-1}+a_2z^{-2}}, \qquad H_i(z) = \frac{b_0+b_1z^{-1}}{1+a_1z^{-1}+a_2z^{-2}}$$

$$H_i(z) = b_0 + b_1z^{-1} + b_2z^{-2}, \qquad H_i(z) = \frac{b_0+b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}}$$

$$H_i(z) = \frac{b_0+b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}} \qquad \text{(Biquadratic)}$$

$$H_i(z) = \frac{b_0+b_1z^{-1}}{1+a_1z^{-1}} \qquad \text{(Bilinear)}$$

Each of the $H_i(z)$ could then be realized using either the direct form I or II.

Different structures are obtained by changing the ordering of the sections and by changing the pole-zero pairings. In practice due to the finite word-length effects, each such cascade realization behaves differently from the others.

Problem:

Develop two different cascade canonic realizations of the following causal IIR transfer function

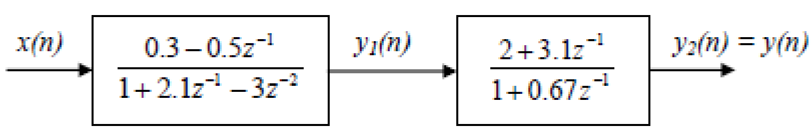
$$H(z) = \frac{z(0.3z-0.5)(2z+3.1)}{(z^2+2.1z-3)(z+0.67)}$$

Solution Write in terms of negative powers of z:

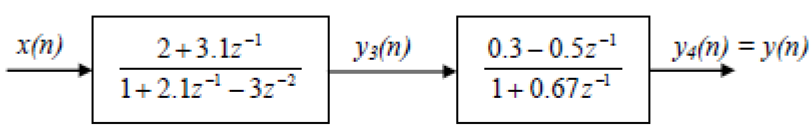
$$H(z) = \frac{z\,z^2(0.3-0.5z^{-1})(2+3.1z^{-1})}{z^3(1+2.1z^{-1}-3z^{-2})(1+0.67z^{-1})} = \frac{(0.3-0.5z^{-1})(2+3.1z^{-1})}{(1+2.1z^{-1}-3z^{-2})(1+0.67z^{-1})}$$

Two (of several) different cascade arrangements, based on how the factors are paired, are shown below in block diagram form. Note that the intermediate signal $y_1(n)$ is different from the intermediate signal $y_3(n)$.

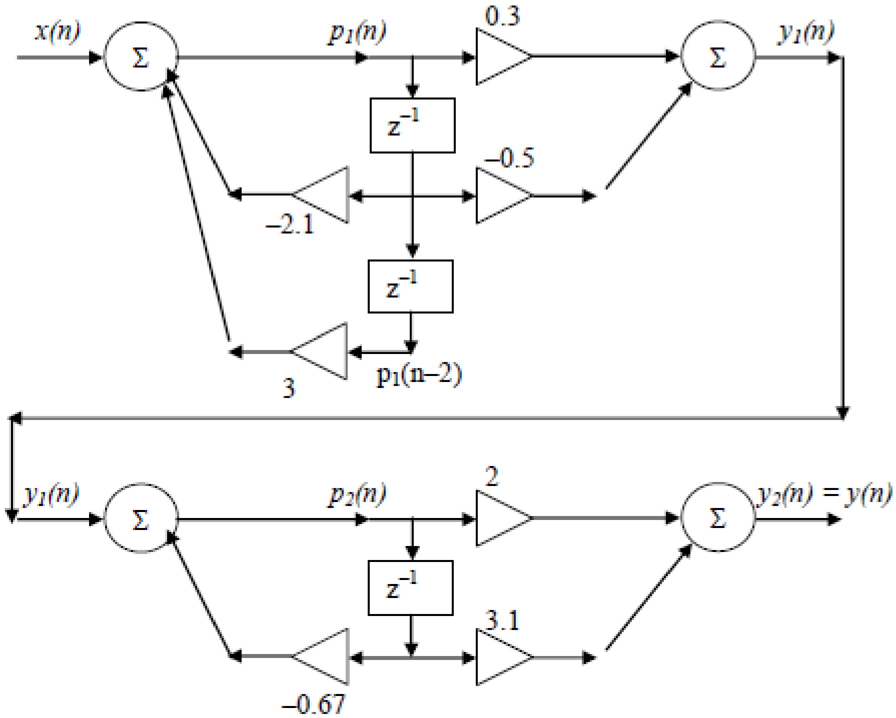
Cascade – *A*



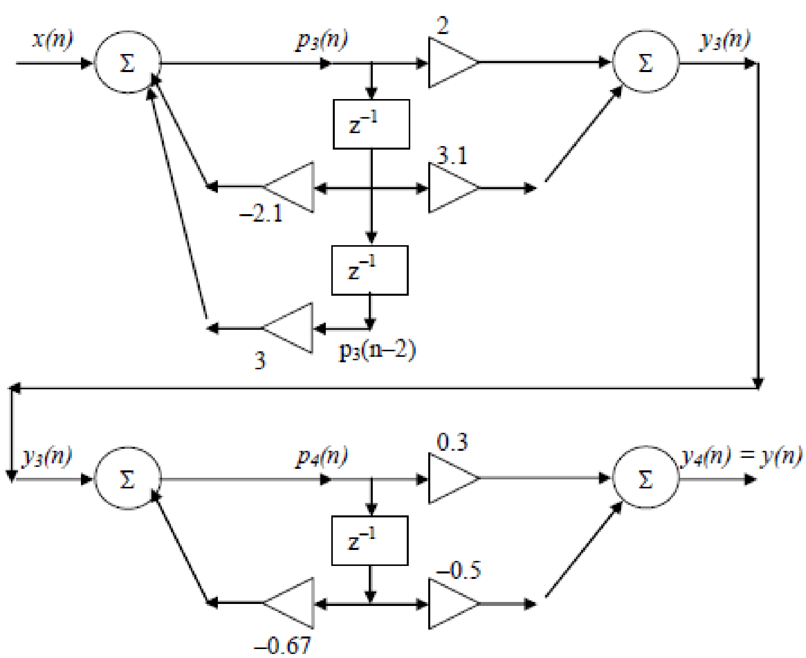
Cascade – *B*



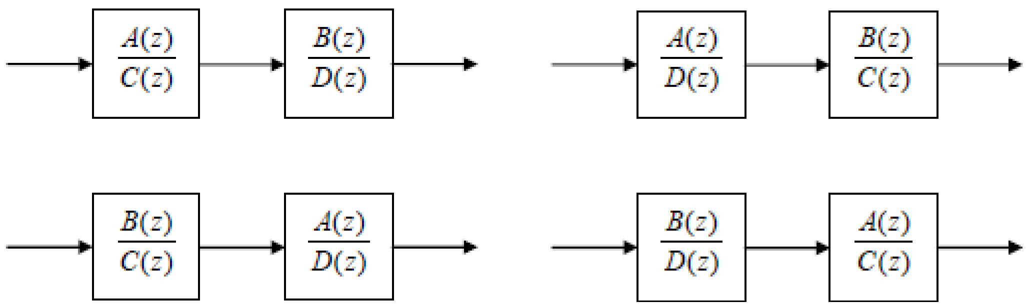
Cascade *A* is shown below using the direct form II for each block separately:



Cascade *B* is shown below using the direct form II for each block separately:



Note in this example that if $H(z) = \dfrac{A(z)\,B(z)}{C(z)\,D(z)}$, then depending on the pole-zero pairings and the sequence order of the blocks in the cascade we can have 4 different implementations (structures). These are equivalent from input to output though not at the intermediate point between the blocks. Moreover the quadratic ($z^2 + 2.1z - 3$) has real roots and so can be split into two factors each of which can be combined with the other factor ($z + 0.67$) in the denominator. This results in more than the 4 structures shown here.



Parallel realization of IIR filters The transfer function H(z) is written as a sum of transfer functions $H_1(z), H_2(z), \ldots, H_k(z)$ obtained by partial fraction expansion:

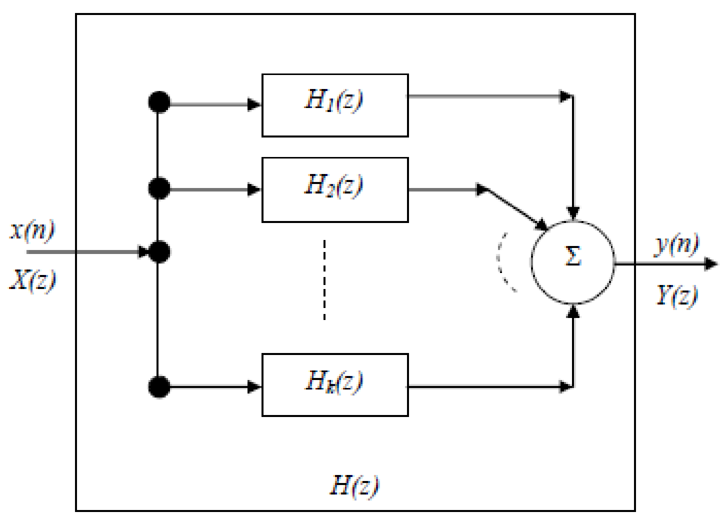
$$\frac{Y(z)}{X(z)} = H(z) = H_1(z) + H_2(z) + \ \ldots + H_{k-1}(z) + H_k(z)$$

Thus

$$\begin{aligned} Y(z) = H(z)\,X(z) &= [H_1(z) + H_2(z) + \ \ldots + H_{k-1}(z) + H_k(z)]\,X(z) \\ &= H_1(z)\,X(z) + H_2(z)\,X(z) + \ \ldots + H_{k-1}(z)\,X(z) + H_k(z)\,X(z) \end{aligned}$$

and is shown in block diagram fashion below. Note that the outputs $y_1(n), y_2(n), \ldots, y_k(n)$ are independent of each other; they are not coupled as in the case of the cascade structure.

Based on whether $H(z)/z$ or $H(z)$ is the starting point for partial fractions we have parallel forms I and II (S. K. Mitra). Both of these methods are illustrated below.



Problem:

Obtain the parallel realization for

$$H(z) = \frac{8z^3 - 4z^2 + 11z - 2}{(z-(1/4))(z^2 - z + (1/2))}$$

Solution For the **Parallel Form I** we expand $H(z)/z$. Note that in the denominator the factor $(z^2 - z + (1/2))$ represents a complex conjugate pair of poles at $((1/2) \pm j(1/2))$.

$$\frac{H(z)}{z} = \frac{8z^3 - 4z^2 + 11z - 2}{z(z-(1/4))(z^2 - z + (1/2))} = \frac{A}{z} + \frac{B}{(z-(1/4))} + \frac{Cz + D}{(z^2 - z + (1/2))}$$

$$A = \left. \frac{8z^3 - 4z^2 + 11z - 2}{(z-(1/4))(z^2 - z + (1/2))} \right|_{z=0} = \frac{-2}{(-1/4)(1/2)} = 16$$

$$B = \left. \frac{8z^3 - 4z^2 + 11z - 2}{z(z^2 - z + (1/2))} \right|_{z=1/4} = \ldots = \frac{\cdots}{\cdots} = 8$$

To determine C and D

$$\frac{8z^3 - 4z^2 + 11z - 2}{z(z-(1/4))(z^2 - z + (1/2))}$$
$$= \frac{A(z-(1/4))(z^2 - z + (1/2)) + Bz(z^2 - z + (1/2)) + (Cz + D)z(z-(1/4))}{z(z-(1/4))(z^2 - z + (1/2))}$$

Putting $A = 16$ and $B = 8$, and equating the numerators on both sides

$$8z^3 - 4z^2 + 11z - 2$$
$$= 16(z-(1/4))(z^2 - z + (1/2)) + 8z(z^2 - z + (1/2)) + (Cz + D)z(z-(1/4))$$

Equating the coefficients of like powers of z on both sides we have

z^3: $\quad 8 = 16 + 8 + C \quad \rightarrow C = -16$

z^0: $\quad -2 = 16\,(-1/4)\,(1/2)$ which is an identity $\quad \rightarrow$ doesn't help

z^1: $\quad 11 = 16\,(1/2) + 16\,(-1/4)\,(-1) + 8\,(1/2) + D\,(-1/4) \quad \rightarrow D = 20$

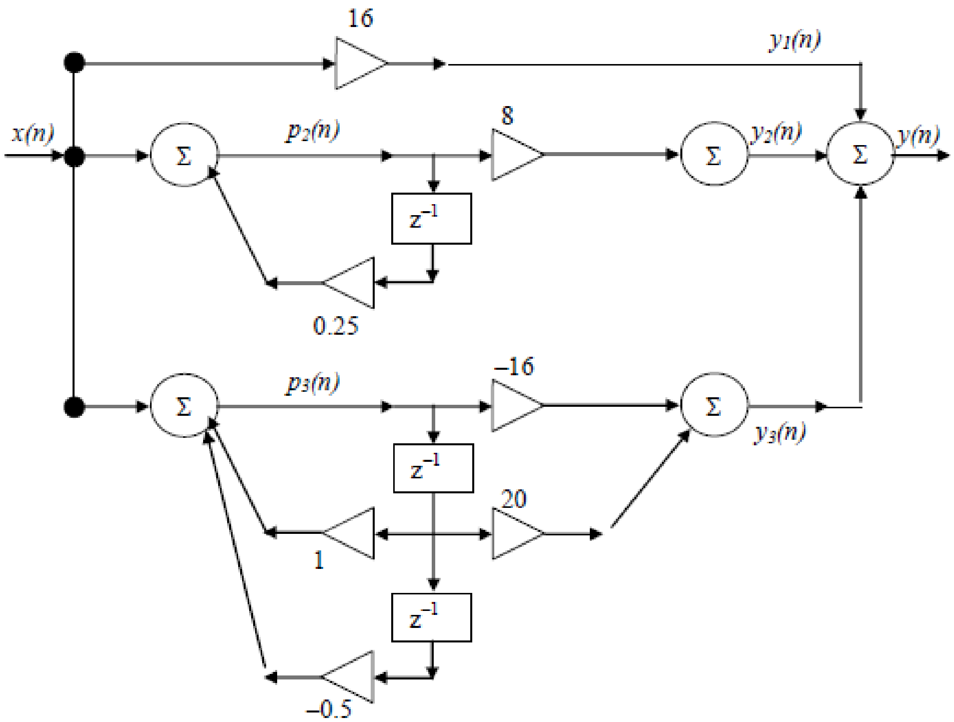
Therefore we have

$$\frac{H(z)}{z} = \frac{16}{z} + \frac{8}{(z-(1/4))} + \frac{(-16)z + 20}{(z^2 - z + (1/2))}$$

$$H(z) = 16 + \frac{8z}{(z-(1/4))} + \frac{z(20 - 16z)}{(z^2 - z + (1/2))}$$

$$H(z) = 16 + \frac{8}{1 - 0.25z^{-1}} + \frac{-16 + 20z^{-1}}{1 - z^{-1} + 0.5z^{-2}} = H_1(z) + H_2(z) + H_3(z)$$

The corresponding parallel form I diagram is shown below.



Realization of FIR filters A causal FIR filter is characterized by its transfer function H(z) given by

$$\frac{Y(z)}{X(z)} = H(z) = \sum_{r=0}^{M} b_r z^{-r} = b_0 + b_1 z^{-1} + \ldots + b_M z^{-M}$$

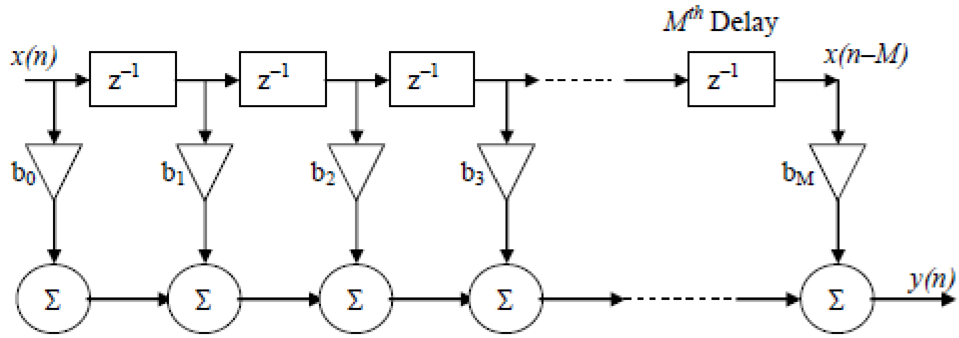
or, by the corresponding difference equation

$$y(n) = \sum_{r=0}^{M} b_r x(n-r) = b_0\, x(n) + b_1\, x(n-1) + b_2\, x(n-2) \ldots + b_M\, x(n-M)$$

Note that some use the notation below with M coefficients instead of $M + 1$

$$y(n) = \sum_{r=0}^{M-1} b_r x(n-r) = b_0\, x(n) + b_1\, x(n-1) + b_2\, x(n-2) \ldots + b_{M-1}\, x(n-M+1)$$

We see that the output $y(n)$ is a weighted sum of the present and past input values; it does not depend on past output values such as $y(n-1)$, etc. The block diagram is shown below. It is also called a **tapped delay line** or a **transversal filter**.



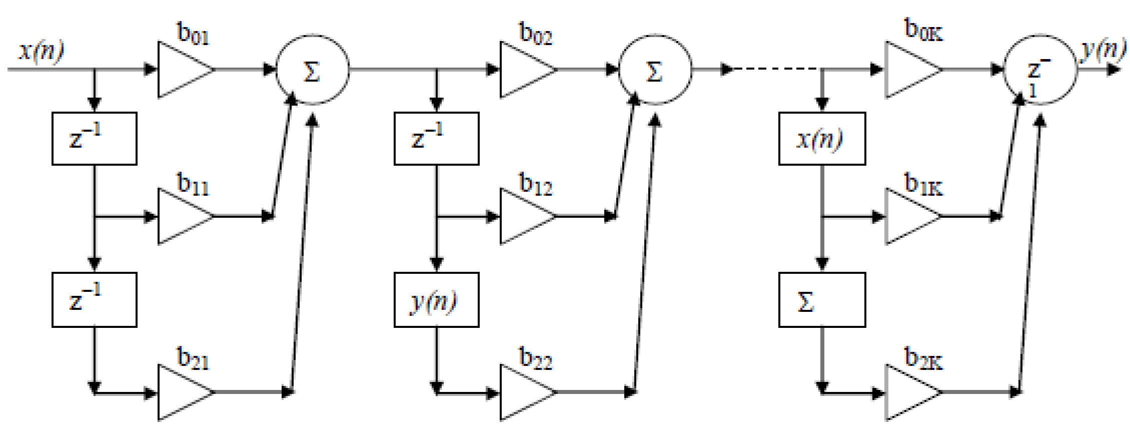
It can be seen that this is the same as the direct form I or II shown earlier for the IIR filter, except that the coefficients a_1 through a_N are zero and $a_0 = 1$; further the delay elements are arranged in a horizontal line. As in earlier diagrammatic manipulation the multipliers can all feed into the rightmost adder and the remaining adders removed.

Other simplifications are possible based on the symmetry of the coefficients $\{b_r\}$, as we shall see in FIR filter design.

Cascade realization of FIR filters The simplest form occurs when the system function is factored in terms of quadratic expressions in z^{-1} as follows:

$$H(z) = \prod_{i=1}^{K} H_i(z) = \prod_{i=1}^{K} \left(b_{0i} + b_{1i}z^{-1} + b_{2i}z^{-2}\right)$$

Selecting the quadratic terms to correspond to the complex conjugate pairs of zeros of $H(z)$ allows a realization in terms of real coefficients b_{0i}, b_{1i} and b_{2i}. Each quadratic could then be realized using the direct form (or alternative structures) as shown below.



INTRODUCTION TO DFT:

Frequency analysis of discrete time signals is usually performed on digital signal processor, which may be general purpose digital computer or specially designed digital hardware. To perform frequency analysis on discrete time signal, we convert the time domain sequence to an equivalent frequency domain representation. We know that such representation is given by The Fourier transform $X(e^{jw})$ of the sequence x(n). However, $X(e^{jw})$ is a continuous function of frequency and therefore, It is not a computationally convenient representation of the sequence.DFT is a powerful computational tool for performing frequency analysis of discrete time signals. The N-point DFT of discrete time sequence x(n) is denoted by X(k) and is defined as

$$DFT[x(n)] = X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}$$ **; k = 0,1,2.........(N-1)**

Where $W_N = e^{-j\left(\frac{2\pi}{N}\right)}$

IDFT of X(k) is given by

$$IDFT[X(k)] = x(n) = \frac{1}{N} \sum_{K=0}^{N-1} X(k) W_N^{-nk}$$ **; n= 0,1,2……..(N-1)**

Where $W_N = e^{-j\left(\frac{2\pi}{N}\right)}$

Example Find the 4-point DFT of the sequence $x(n) = \cos \frac{n\pi}{4}$.

Solution Given $N = 4$,

$$x(n) = [\cos(0), \cos(\pi/4), \cos(\pi/2), \cos(3\pi/4)]$$
$$= \{1, 0.707, 0, -0.707\}$$

The N-point DFT of the sequence $x(n)$ is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi nk/N}, \; k = 0, 1, \ldots N-1.$$

The DFT is

$$X(k) = \sum_{n=0}^{3} x(n)\, e^{-j2\pi nk/4}, \; k = 0, 1, 2, 3$$

$$= \sum_{n=0}^{3} x(n)\, e^{-j\pi nk/2},\ k = 0, 1, 2, 3$$

For $k = 0$

$$X(0) = \sum_{n=0}^{3} x(n) = 1$$

For $k = 1$

$$X(1) = \sum_{n=0}^{3} x(n)\, e^{-j\pi(1)n/2}$$

$$= 1 + 0.707\, e^{-j\pi/2} + 0 + (-0.707) e^{-j3\pi/2}$$

$$= 1 + (0.707)(-j) + 0 - (0.707)(j)$$

$$= 1 - j\,1.414$$

For $k = 2$

$$X(2) = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j\pi(2)n/2} = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j\pi n}$$

$$= 1 + (0.707)\, e^{-j\pi} + 0 + (-0.707)\, e^{-j3\pi}$$

$$= 1 + (0.707)(-1) + 0 + (-0.707)(-1) = 1$$

For $k = 3$

$$X(3) = \sum_{n=0}^{3} x(n)\, e^{-j\pi(3)n/2}$$

$$= 1 + (0.707)\, e^{-j3\pi/2} + 0 + (-0.707)\, e^{-j9\pi/2}$$

$$= 1 + (0.707)(j) + 0 + (-0.707)(-j) = 1 + j\,1.414$$

$$X(k) = \{1, 1 - j\,1.414, 1, 1 + j\,1.414\}$$

Example Find the N-Point DFT for $x(n) = a^n$ for $0 < a < 1$.

Solution The N-point DFT is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi nk/N},\ k = 0, 1, \ldots, N-1.$$

$$= \sum_{n=0}^{N-1} a^n\, e^{-j2\pi nk/N}$$

$$= \sum_{n=0}^{N-1} \left(a e^{-j2\pi k/N}\right)^n$$

$$= \frac{1-\left(a e^{-j2\pi k/N}\right)^N}{1-a e^{-j2\pi k/N}}$$

$$X(k) = \frac{1-a^N}{1-a e^{-j2\pi k/N}},\ k = 0, 1, \ldots, N-1$$

Example Derive the DFT of the sample data sequence $x(n) = \{1, 1, 2, 2, 3, 3\}$ and compute the corresponding amplitude and phase spectrum.

Solution The N-point DFT of a finite duration sequence $x(n)$ is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi nk/N},\ k = 0, 1, \ldots, N-1.$$

For $k = 0$

$$X(0) = \sum_{n=0}^{5} x(n)\, e^{-j2\pi(0)n/6} = \sum_{n=0}^{5} x(n) = 1+1+2+2+3+3 = 12$$

For $k = 1$

$$X(1) = \sum_{n=0}^{5} x(n)\, e^{-j2\pi(1)n/6}$$

$$= \sum_{n=0}^{5} x(n)\, e^{-j\pi n/3}$$

$$= 1 + e^{-j\pi/3} + 2\,e^{-j2\pi/3} + 2\,e^{-j\pi} + 3\,e^{-j4\pi/3} + 3\,e^{-j5\pi/3}$$

$$= 1 + 0.5 - j\,0.866 + 2\,(-0.5 - j\,0.866) + 2\,(-1) + 3\,(-0.5 + j\,0.866) + 3(0.5 + j\,0.866)$$

$$= -1.5 + j\,2.598$$

For $k = 2$

$$X(2) = \sum_{n=0}^{5} x(n)\, e^{-j2\pi(2)n/6}$$

$$= \sum_{n=0}^{5} x(n)\, e^{-2j\pi n/3}$$

$$= 1 + e^{-j2\pi/3} + 2e^{-j4\pi/3} + 2e^{-j2\pi} + 3e^{-j8\pi/3} + 3e^{-j10\pi/3}$$

$$= 1 + (-0.5) - j\,0.866 + 2(-0.5 + j\,0.866) + 2(1) + 3\,(-0.5 - j\,0.866) + 3(-0.5 + j\,0.866)$$

$$= -1.5 + j\,0.866$$

For $k = 3$

$$X(3) = \sum_{n=0}^{5} x(n)\, e^{-j2\pi(3)n/6}$$

$$= \sum_{n=0}^{5} x(n)\, e^{-j\pi n}$$

$$= 1 + e^{-j\pi} + 2e^{-j2\pi} + 2e^{-j3\pi} + 3e^{-j4\pi} + 3e^{-j5\pi}$$
$$= 1 - 1 + 2(1) + 2(-1) + 3(1) + 3(-1) = 0$$

For $k = 4$

$$X(4) = \sum_{n=0}^{5} x(n)\, e^{-j2\pi(4)n/6}$$
$$= \sum_{n=0}^{5} x(n)\, e^{-j4\pi n/3}$$
$$= 1 + e^{-j4\pi/3} + 2e^{-j8\pi/3} + 2e^{-j4\pi} + 3e^{-j16\pi/3} + 3e^{-j20\pi/3}$$
$$= 1 + (-0.5 + j\,0.866) + 2(-0.5 - j\,0.866) + 2(1) + 3(-0.5 + j\,0.866) + 3(-0.5 - j\,0.866)$$
$$= -1.5 - j\,0.866$$

For $k = 5$

$$X(5) = \sum_{n=0}^{5} x(n)\, e^{-j2\pi(5)n/6}$$
$$= \sum_{n=0}^{5} x(n)\, e^{-j5\pi n/3}$$
$$= 1 + e^{-j5\pi/3} + 2e^{-j10\pi/3} + 2e^{-j5\pi} + 3e^{-j20\pi/3} + 3e^{-j25\pi/3}$$
$$= 1 + (-0.5 + j\,0.866) + 2(-0.5 + j\,0.866) + 2(-1) + 3(-0.5 - j\,0.866) + 3(0.5 - j\,0.866)$$
$$= -1.5 - j2.598$$

$$X(k) = \{12, -1.5 + j2.598, -1.5 + j0.866, 0, -1.5 - j\,0.866, -1.5 - j\,2.598\}$$

The corresponding amplitude spectrum is given by

$$|X(k)| = \left\{\sqrt{12 \times 12}, \sqrt{(-1.5)^2 + (-2.598)^2}, \sqrt{(-1.5)^2 + (0.866)^2}, 0, \sqrt{(-1.5)^2 + (-0.866)^2}, \sqrt{(-1.5)^2 + (-2.598)^2}\right\}$$
$$= \{12, 2.999, 1.732, 0, 1.732, 2.999\}$$

and the corresponding phase spectrum is given by

$$\angle X(k) = \left\{\tan^{-1}(0), \tan^{-1}\left(\frac{2.598}{-1.5}\right), \tan^{-1}\left(\frac{0.866}{-1.5}\right), \tan^{-1}(0)\ \tan^{-1}\left(\frac{-0.866}{-1.5}\right), \tan^{-1}\left(\frac{-2.598}{-1.5}\right)\right\}$$
$$= \left\{0, -\frac{\pi}{3}, -\frac{\pi}{6}, 0, \frac{\pi}{6}, \frac{\pi}{3}\right\}$$

Example Find the inverse DFT of $X(k) = \{1, 2, 3, 4\}$.

Solution The inverse DFT is defined as

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, e^{j2\pi nk/N},\ n = 0, 1, 2, 3, \ldots, N-1.$$

Given $N = 4$, $x(n) = \frac{1}{4} \sum_{k=0}^{3} X(k)\, e^{j2\pi nk/N}$, $n = 0, 1, 2, 3$

When $n = 0$

$$x(0) = \frac{1}{4} \sum_{k=0}^{3} X(k)\, e^{j\pi(0)k/2}$$

$$= \frac{1}{4}(1+2+3+4) = \frac{5}{2}$$

When $n = 1$

$$x(1) = \frac{1}{4} \sum_{k=0}^{3} X(k)\, e^{j\pi(1)k/2}$$

$$= \frac{1}{4}\left(1 + 2e^{j\pi/2} + 3e^{j\pi} + 4e^{j3\pi/2}\right)$$

$$= \frac{1}{4}(1 + 2(j) + 3(-1) + 4(-j))$$

$$= \frac{1}{4}(-2 - j2) = -\frac{1}{2} - j\frac{1}{2}$$

When $n = 2$

$$x(2) = \frac{1}{4} \sum_{k=0}^{3} X(k)\, e^{j\pi k}$$

$$= \frac{1}{4}\left(1 + 2e^{j\pi} + 3e^{j2\pi} + 4e^{j3\pi}\right)$$

$$= \frac{1}{4}(1 + 2(-1) + 3(1) + 4(-1))$$

$$= \frac{1}{4}(-2) = -1/2$$

When $n = 3$

$$x(3) = \frac{1}{4} \sum_{k=0}^{3} X(k)\, e^{j3\pi k/2}$$

$$= \frac{1}{4}\left(1 + 2e^{j3\pi/2} + 3e^{j3\pi} + 4e^{j9\pi/2}\right)$$

$$= \frac{1}{4}(1 + 2(-j) + 3(-1) + 4j)$$

$$= \frac{1}{4}(-2 + 2j) = -\frac{1}{2} + j\frac{1}{2}$$

$$x(n) = \left\{\frac{5}{2}, -\frac{1}{2} - j\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2} + j\frac{1}{2}\right\}$$

Properties of the DFT

The properties of the DFT are useful in the practical techniques for processing signals. The various properties are given below.

Periodicity

If $X(k)$ is an N-point DFT of $x(n)$, then

$x(n + N) = x(n)$ for all n

$X(k + N) = X(k)$ for all k

Linearity

If $X_1(k)$ and $X_2(k)$ are the N-point DFTs of $x_1(n)$ and $x_2(n)$ respectively, and a and b are arbitrary constants either real or complex-valued, then

$$a x_1(n) + b x_2(n) \xleftrightarrow[N]{DFT} aX_1(k) + bX_2(k)$$

Time Reversal of a Sequence

If $\quad x(n) \xleftrightarrow[N]{DFT} X(k)$, then

$$x(-n, (\text{mod } N)) = x(N - n) \xleftrightarrow[N]{DFT} X(-k, (\text{mod } N)) = X(N - k)$$

Hence, when the N-point sequence in time is reversed, it is equivalent to reversing the DFT values.

Circular Time Shift

If $\quad x(n) \xleftrightarrow[N]{DFT} X(k)$, then

$$x(n - l, (\text{mod } N)) \xleftrightarrow[N]{DFT} X(k)e^{-j2\pi kl/N}$$

Shifting of the sequence by l units in the time-domain is equivalent to multiplication of $e^{-j2\pi kl/N}$ in the frequency-domain.

Circular Frequency Shift

If $\quad x(n) \xleftrightarrow[N]{DFT} X(k)$, then

$$x(n)e^{j2\pi ln/N} \xleftrightarrow[N]{DFT} X(k - l, (\text{mod } N))$$

Hence, when the sequence $x(n)$ is multiplied by the complex exponential sequence $e^{j2\pi ln/N}$, it is equivalent to circular shift of the DFT by l units in the frequency domain.

Complex Conjugate Property

If $$x(n) \xleftrightarrow[N]{DFT} X(k), \text{ then}$$

$$x^*(n) \xleftrightarrow[N]{DFT} X^*(-k, (\text{mod } N)) = X^*(N-k)$$

Circular Convolution

If $$x_1(n) \xleftrightarrow[N]{DFT} X_1(k) \quad \text{and} \quad x_2(n) \xleftrightarrow[N]{DFT} X_2(k), \text{ then}$$

$$x_1(n) \circledN x_2(n) \xleftrightarrow[N]{DFT} X_1(k)\, X_2(k)$$

where $x_1(n) \circledN x_2(n)$ denotes the circular convolution of the sequence $x_1(n)$ and $x_2(n)$ defined as

$$x_3(n) = \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m, (\text{mod} N))$$

$$= \sum_{m=0}^{N-1} x_2(m)\, x_1(n-m, (\text{mod} N))$$

Multiplication of Two Sequences

If $$x_1(n) \xleftrightarrow[N]{DFT} X_1(k) \quad \text{and} \quad x_2(n) \xleftrightarrow[N]{DFT} X_2(k), \text{ then}$$

$$x_1(n)\, x_2(n) \xleftrightarrow[N]{DFT} \frac{1}{N} X_1(k) \circledN X_2(k)$$

Parseval's Theorem

For complex-valued sequences $x(n)$ and $y(n)$,

if $$x(n) \xleftrightarrow[N]{DFT} X(k) \quad \text{and} \quad y(n) \xleftrightarrow[N]{DFT} Y(k), \text{ then}$$

$$\sum_{n=0}^{N-1} x(n)\, y^*(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, Y^*(k)$$

If $y(n) = x(n)$, then the above equation reduces to

$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2$$

This expression relates the energy in the finite duration sequence $x(n)$ to the power in the frequency components $X(k)$.

Methods of Circular Convolution:

Generally, there are two methods, which are adopted to perform circular convolution and they are –

(1) Concentric circle method (2) Matrix multiplication method.

Concentric Circle Method:

Let $x_1(n)$ and $x_2(n)$ be two given sequences. The steps followed for circular convolution of $x_1(n)$ and $x_2(n)$ are

- **Take two concentric circles. Plot N samples of $x_1(n)$ on the circumference of the outer circle (maintaining equal distance successive points) in anti-clockwise direction.**
- **For plotting $x_2(n)$,plot N samples of $x_2(n)$ in clockwise direction on the inner circle, starting sample placed at the same point as 0^{th} sample of $x_1(n)$**
- **Multiply corresponding samples on the two circles and add them to get output.**
- **Rotate the inner circle anti-clockwise with one sample at a time.**

Matrix Multiplication Method:

Matrix method represents the two given sequence $x_1(n)$ and $x_2(n)$ in matrix form.

- **One of the given sequences is repeated via circular shift of one sample at a time to form a N X N matrix.**
- **The other sequence is represented as column matrix.**

The multiplication of two matrices gives the result of circular convolution

SECTIONED CONVOLUTION:

Suppose, the input sequence x(n) of long duration is to be processed with a system having finite duration impulse response by convolving the two sequences. Since, the linear filtering performed via DFT involves operation on a fixed size data block, the input sequence is divided into different fixed size data block before processing. The successive blocks are then processed one at a time and the results are combined to produce the net result. As the convolution is performed by dividing the long input sequence into different fixed size sections, it is called sectioned convolution. A long input sequence is segmented to fixed size blocks, prior to FIR filter processing. Two methods are used to evaluate the discrete convolution.

(1) Overlap-save method (2) Overlap-add method

Overlap Save Method:

Overlap–save is the traditional name for an efficient way to evaluate the discrete convolution between a very long signal x(n) and a finite impulse response FIR filter h(n).

1. **Insert M - 1 zeros at the beginning of the input sequence x(n).**
2. **Break the padded input signal into overlapping blocks $x_m(n)$ of length N = L + M - 1 where the overlap length is M -1.**

3. **Zero pad h(n) to be of length N = L + M - 1.**
4. **Take N-DFT of h(n) to give H(k), k = 0, 1,2,.........N - 1.**

5. For each block m:

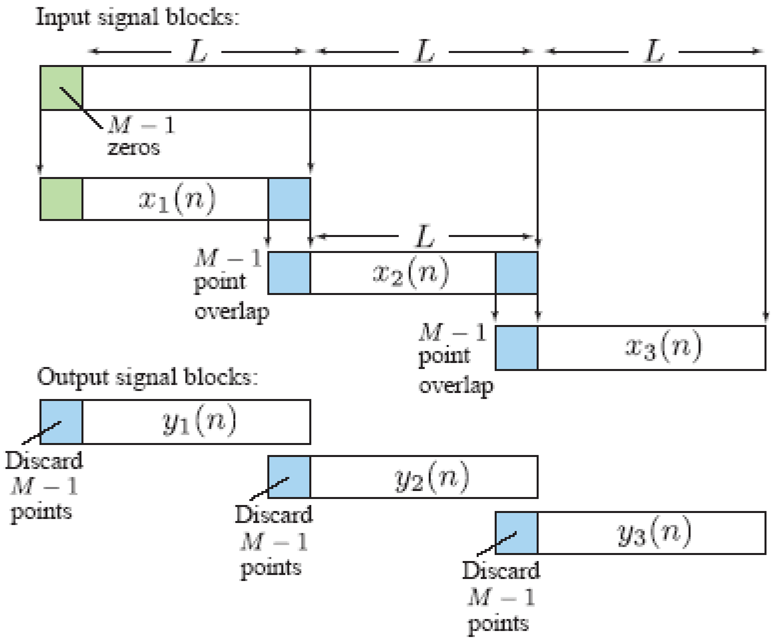
Take N-DFT of $x_m(n)$ to give Xm(k), k = 0, 1,2,… N - 1.

5.2 Multiply: Ym(k) = Xm(k) . H(k), k = 0, 1,2,……………..N - 1.

Take N-IDFT of Ym(k) to give $y_m(n)$,n = 0, 1,2,……………..N - 1.

Discard the first M - 1 points of each output block $y_m(n)$

6. Form y(n) by appending the remaining (i.e., last) L samples of each block



Overlap Add Method:

Given below are the steps to find out the discrete convolution using Overlap method:

1. Break the input signal x(n) into non-overlapping blocks $x_m(n)$ of length L.
2. Zero pad h(n) to be of length N = L + M - 1.
3. Take N-DFT of h(n) to give H(k), k = 0, 1,2,……………..N - 1.
4. For each block m:

Zero pad $x_m(n)$ to be of length N = L + M - 1.

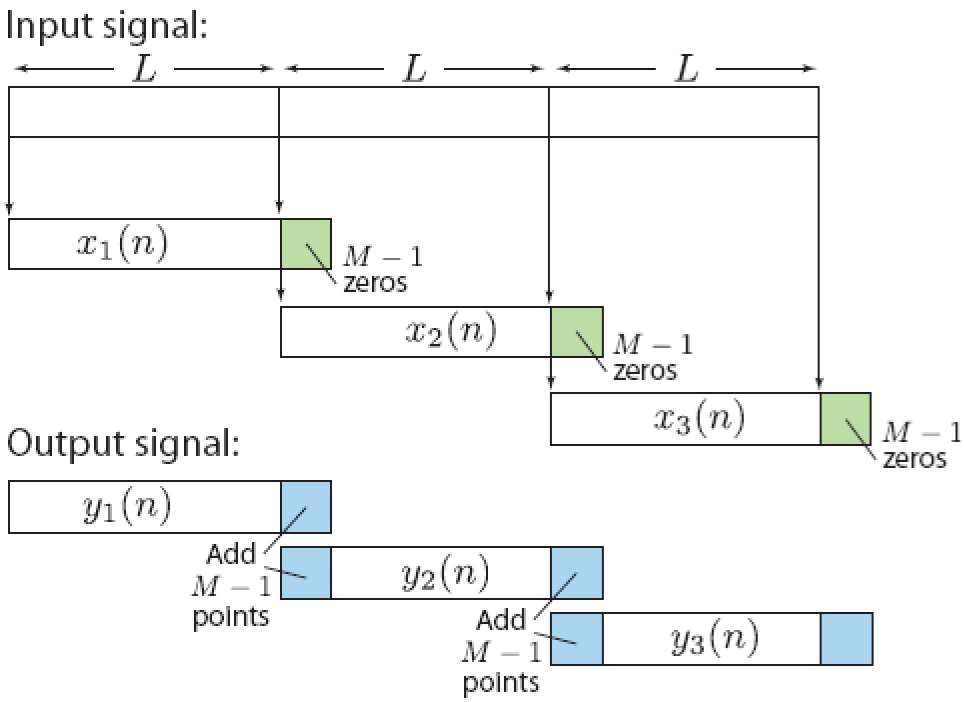
Take N-DFT of $x_m(n)$to give $X_m(k)$, k = 0, 1,2,……………..N

- 1. 4.3 Multiply: $Y_m(k) = X_m(k).H(k)$, k = 0, 1,2,

……………..N - 1.

4.4 Take N-IDFT of $Y_m(k)$ to give $y_m(n)$, n = 0, 1,2,……………..N - 1.

5. Form y(n) by overlapping the last M - 1 samples of $y_m(n)$ with the first M -1 samples of $y_{m+1}(n)$ and adding the result.



FAST FOURIER TRANSFORM (FFT)

The fast Fourier transform (FFT) is an algorithm that efficiently computes the discrete Fourier transform (DFT). The DFT of a sequence $\{x(n)\}$ of length N is given by a complex-valued sequence $\{X(k)\}$

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi nk/N}, \; 0 \le k \le N-1.$$

Let W_N be the complex-valued phase factor, which is an N th root of unity expressed by

$$W_N = e^{-j2\pi/N}$$

Hence $X(k)$ becomes

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}, \;\; 0 \le k \le N-1$$

Similarly, IDFT becomes

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, W_N^{-nk}, \;\; 0 \le n \le N-1$$

From the above equations, it is evident that for each value of k, the direct computation of $X(k)$ involves N complex multiplications ($4N$ real multiplications) and $N-1$ complex additions ($4N-2$ real additions). Hence, to compute all N values of DFT, N^2 complex multiplications and $N(N-1)$ complex additions are required. The DFT and IDFT involve the same type of computations.

Decimation-in-Time (DIT) Algorithm

In this case, let us assume that $x(n)$ represents a sequence of N values, where N is an integer power of 2, that is, $N = 2^L$. The given sequence is decimated (broken) into two $\frac{N}{2}$ point sequences consisting of the even numbered values of $x(n)$ and the odd numbered values of $x(n)$.

The N-point DFT of sequence $x(n)$ is given by

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}, \quad 0 \le k \le N-1$$

Breaking $x(n)$ into its even and odd numbered values, we obtain

$$X(k) = \sum_{n=0,\, n \text{ even}}^{N-1} x(n)\, W_N^{nk} + \sum_{n=0,\, n \text{ odd}}^{N-1} x(n)\, W_N^{nk}$$

Substituting $n = 2r$ for n even and $n = 2r + 1$ for n odd , we have

$$X(k) = \sum_{r=0}^{(N/2-1)} x(2r)\, W_N^{2rk} + \sum_{r=0}^{(N/2-1)} x(2r+1)\, W_N^{(2r+1)k}$$

$$= \sum_{r=0}^{(N/2-1)} x(2r)\, (W_N^2)^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x(2r+1)\, (W_N^2)^{rk}$$

Here, $W_N^2 = [e^{-j(2\pi/N)}]^2 = e^{-j(2\pi/(N/2))} = W_{N/2}$

Therefore, Eq. can be written as

$$X(k) = \sum_{r=0}^{(N/2-1)} x(2r)\, W_{N/2}^{rk} + W_N^k \sum_{r=0}^{(N/2-1)} x(2r+1)\, W_{N/2}^{rk}$$

$$= G(k) + W_N^k \cdot H(k), \quad k = 0, 1, \ldots \frac{N}{2} - 1$$

where $G(k)$ and $H(k)$ are the $N/2$-point DFTs of the even and odd numbered sequences respectively. Here, each of the sums is computed for $0 \le k \le \frac{N}{2} - 1$ since $G(k)$ and $H(k)$ are considered periodic with period $N/2$.

Therefore,

$$X(k) = \begin{cases} G(k) + W_N^k\, H(k), & 0 \le k \le \frac{N}{2} - 1 \\ G\left(k + \frac{N}{2}\right) + W_N^{(k+N/2)}\, H\left(k + \frac{N}{2}\right), & \frac{N}{2} \le k \le N-1 \end{cases}$$

Using the symmetry property of $W_N^{k+N/2} = -W_N^k$,

$$X(k) = \begin{cases} G(k) + W_N^k\, H(k), & 0 \le k \le \frac{N}{2} - 1 \\ G(k + N/2) - W_N^k\, H(k + N/2), & \frac{N}{2} \le k \le N-1 \end{cases}$$

Figure shows the flow graph of the decimation-in-time decomposition of an 8-point ($N = 8$) DFT computation into two 4-point DFT computations. Here the branches entering a node are added to produce the node variable. If no coefficient is indicated, it means that the branch transmittance is equal to one. For other branches, the transmittance is an integer power of W_N.

Then $X(0)$ is obtained by multiplying $H(0)$ by W_N^0 and adding the product to $G(0)$. $X(1)$ is obtained by multiplying $H(1)$ by W_N^1 and adding that result to $G(1)$. For $X(4)$, $H(4)$ is multiplied by W_N^4 and the result is

added to $G(4)$. But, since $G(k)$ and $H(k)$ are both periodic in k with period 4, $H(4) = H(0)$ and $G(4) = G(0)$. Therefore, $X(4)$ is obtained by multiplying $H(0)$ by W_N^4 and adding the result to $G(0)$.

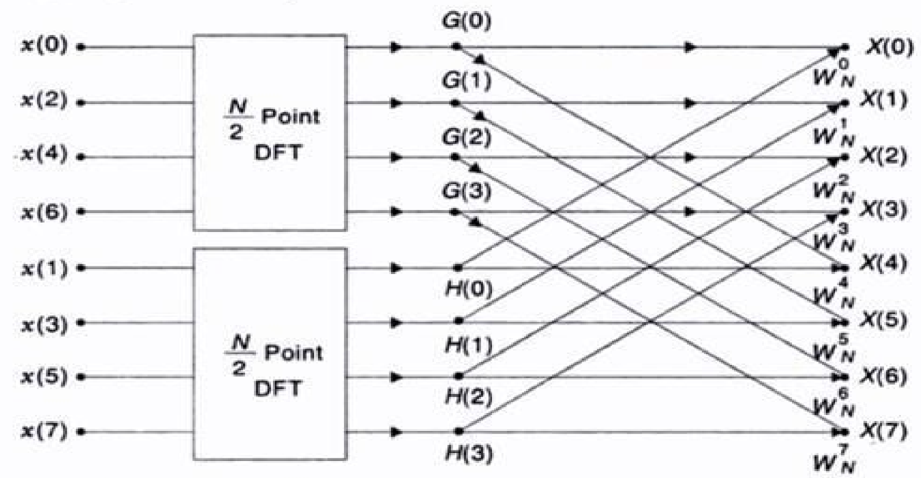


Fig. *Flow Graph of the First Stage Decimation-In-Time FFT Algorithm for N = 8*

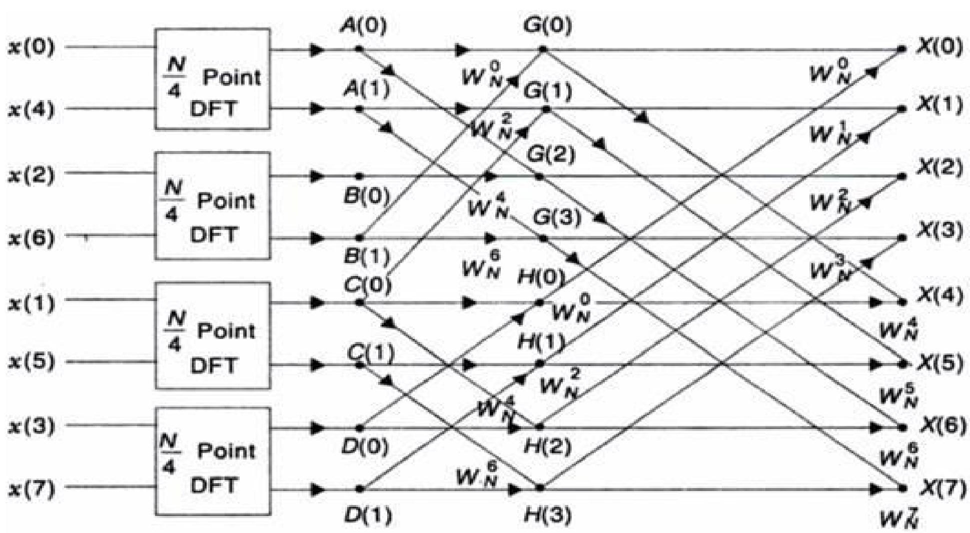


Fig. *Flow Graph of the Second Stage Decimation-in-time FFT Algorithm for N = 8*

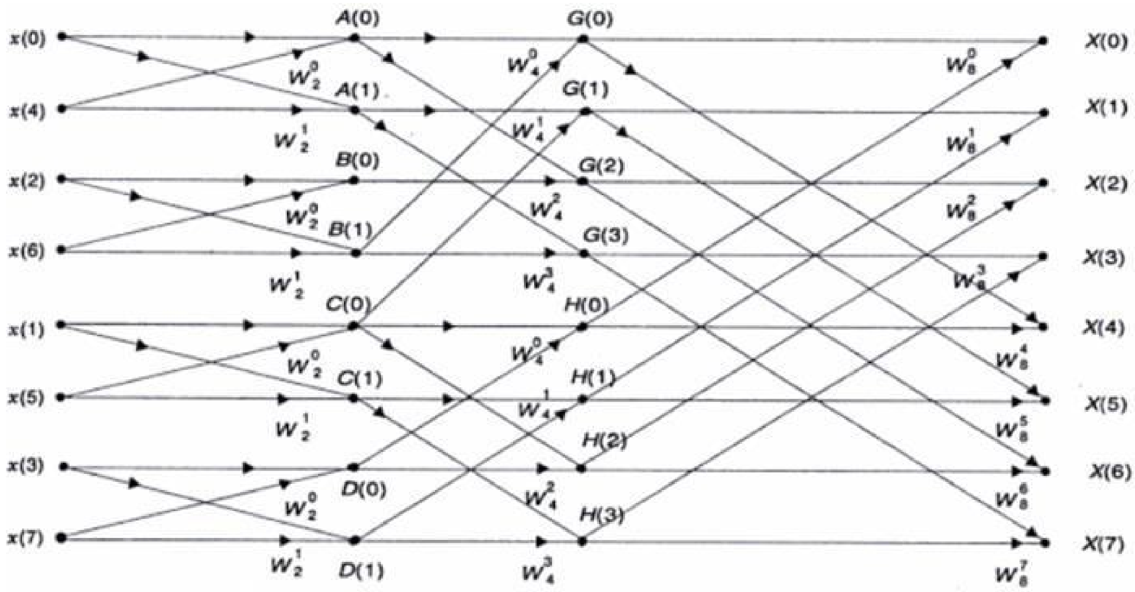


Fig. *The Flow-Graph of the Decimation-in-time FFT Algorithm for N = 8*

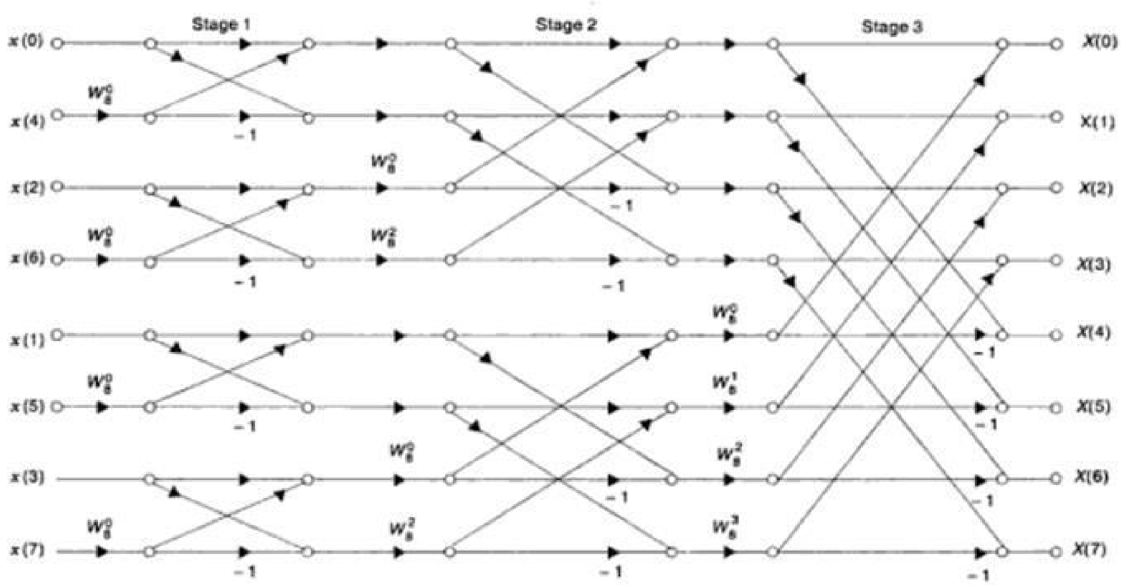


Fig. *Reduced Flow-Graph for an 8-Point DIT FFT*

Example **Given $x(n) = \{1, 2, 3, 4, 4, 3, 2, 1\}$, find $X(k)$ using DIT FFT algorithm.**

Solution **We know that $W_N^k = e^{-j\left(\frac{2\pi}{N}\right)k}$. Given $N = 8$.**

Hence, $W_8^0 = e^{-j\left(\frac{2\pi}{8}\right)0} = 1$

$$W_8^1 = e^{-j\left(\frac{2\pi}{8}\right)} = \cos \pi/4 - j \sin \pi/4 = 0.707 - j\,0.707$$
$$W_8^2 = e^{-j\left(\frac{2\pi}{8}\right)2} = \cos \pi/2 - j \sin \pi/2 = -j$$
$$W_8^3 = e^{-j\left(\frac{2\pi}{8}\right)3} = \cos 3\pi/4 - j \sin 3\pi/4 = -\,0.707 - j\,0.707$$

Using DIT FFT algorithm, we can find $X(k)$ from the given sequence $x(n)$ as shown in Fig.

Therefore, $X(k) = \{20, -5.828 - j\,2.414, 0, 0.172 - j\,0.414, 0, -0.172 + j\,0.414, 0, -5.828 + j\,2.414\}$

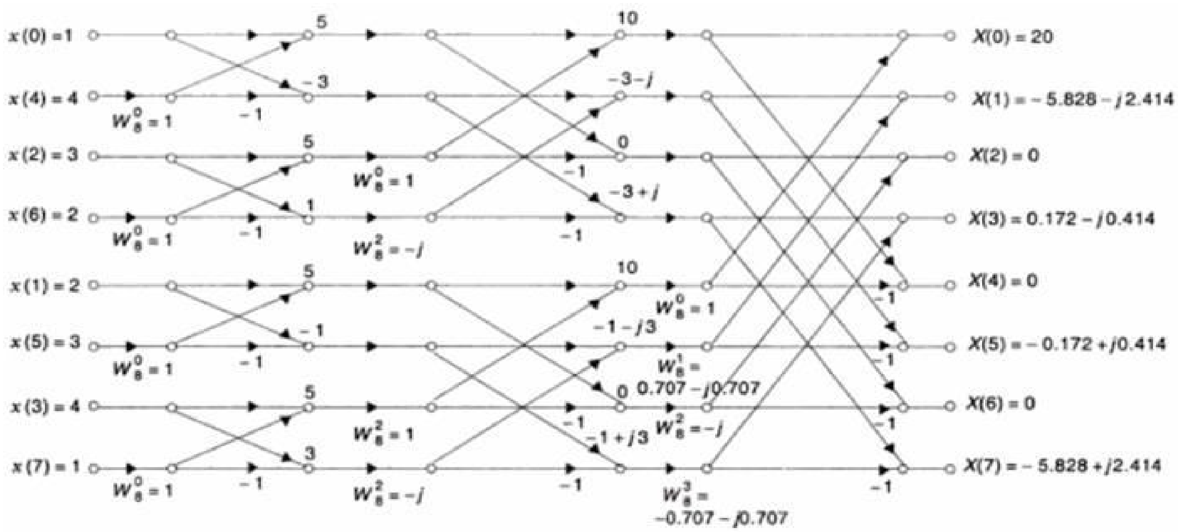


Fig.

Example Given $x(n) = \{0, 1, 2, 3\}$, find $X(k)$ using DIT FFT algorithm.

Solution Given $N = 4$

$$W_N^k = e^{-j\left(\frac{2\pi}{N}\right)k}$$
$$W_4^0 = 1 \text{ and } W_4^1 = e^{-j\pi/2} = -j$$

Using DIT FFT algorithm, we can find $X(k)$ from the given sequence $x(n)$ as shown in Fig.

Therefore, $X(k) = \{6, -2 + j2, -2, -2 - j\,2\}$

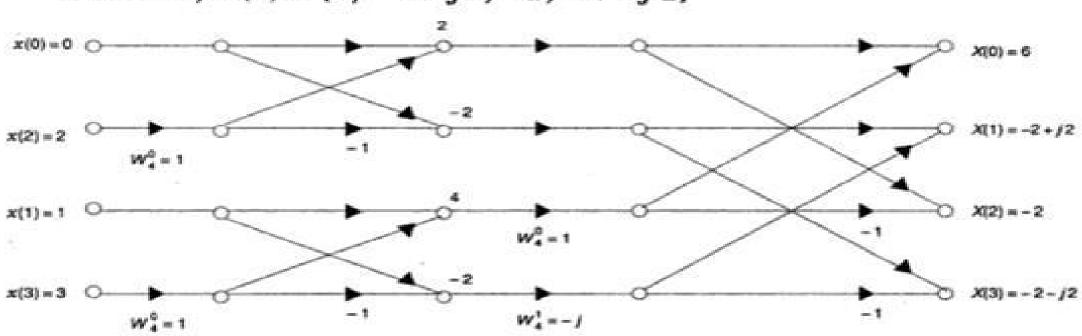


Fig.

Decimation-in-Frequency (DIF) Algorithms

The decimation-in-time FFT algorithm decomposes the DFT by sequentially splitting input samples $x(n)$ in the time domain into sets of smaller and smaller subsequences and then forms a weighted combination of the DFTs of these subsequences. Another algorithm called decimation-in-frequency FFT decomposes the DFT by recursively splitting the sequence elements $X(k)$ in the frequency domain into sets of smaller and smaller subsequences. To derive the decimation-in-frequency FFT algorithm for N, a power of 2, the input sequence $x(n)$ is divided into the first half and the last half of the points

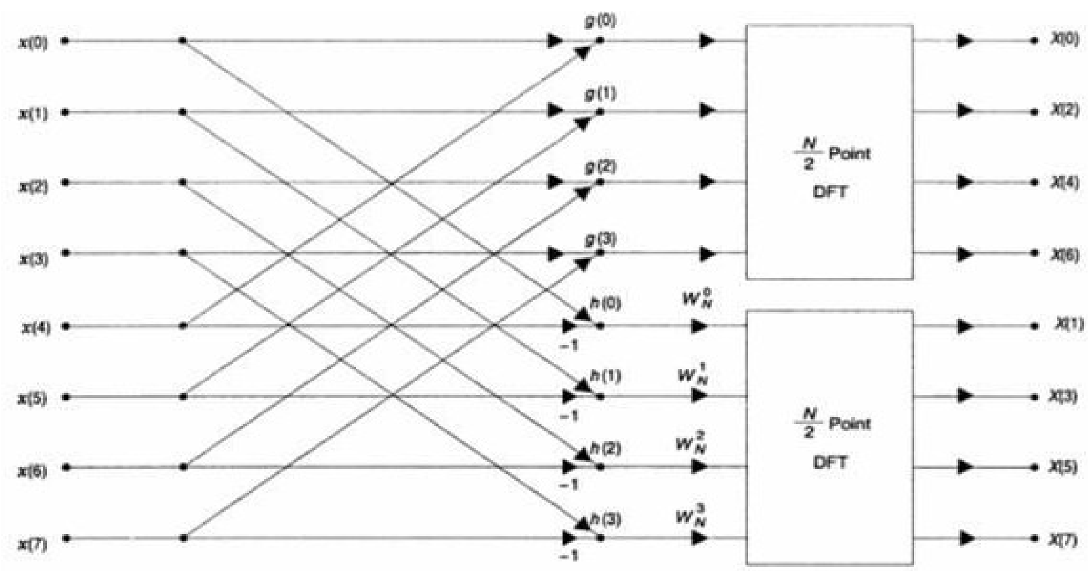


Fig. *Flow Graph of the First Stage of Decimation-In-Frequency FFT for N = 8*

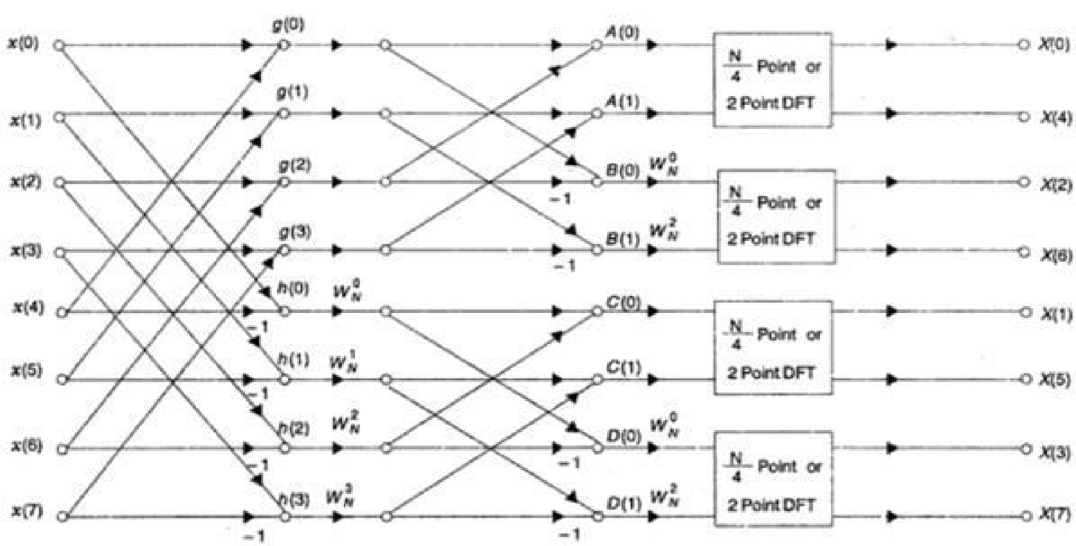


Fig. *Flow Graph of the Second Stage of Decimation-In-Frequency FFT for N = 8*

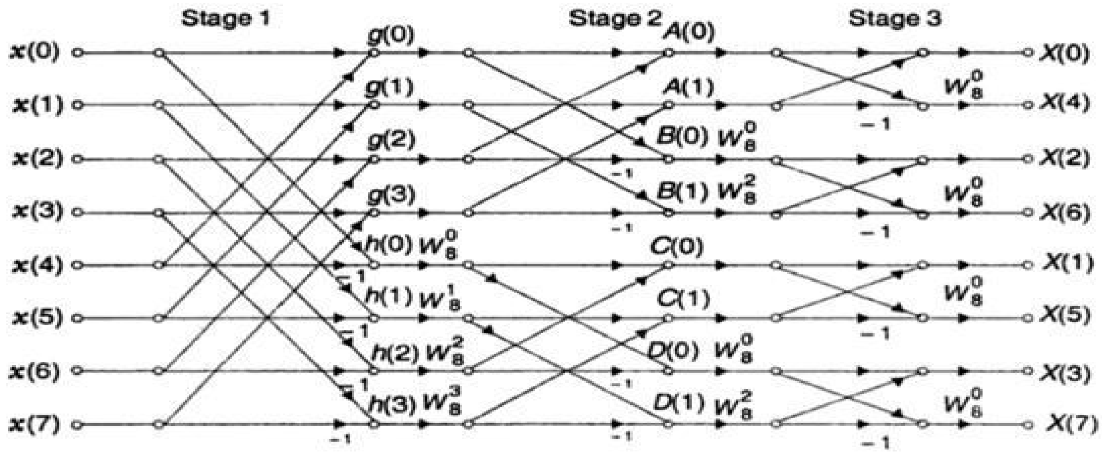


Fig. *Reduced Flow Graph of Final Stage DIF FFT for N = 8*

Example Compute the DFTs of the sequence $x(n) = \cos \frac{n\pi}{2}$, where $N = 4$, using DIF FFT algorithm.

Solution Given $N = 4$ and $x(n) = \{1, 0, -1, 0\}$

$$W_N^k = e^{-j\left(\frac{2\pi}{N}\right)k}$$
$$W_4^0 = 1 \text{ and } W_4^1 = e^{-j\pi/2} = -j$$

Using DIF FFT algorithm, we can find $X(k)$ from the given sequence $x(n)$ as shown in Fig.

Therefore, $X(k) = \{0, 2, 0, 2\}$

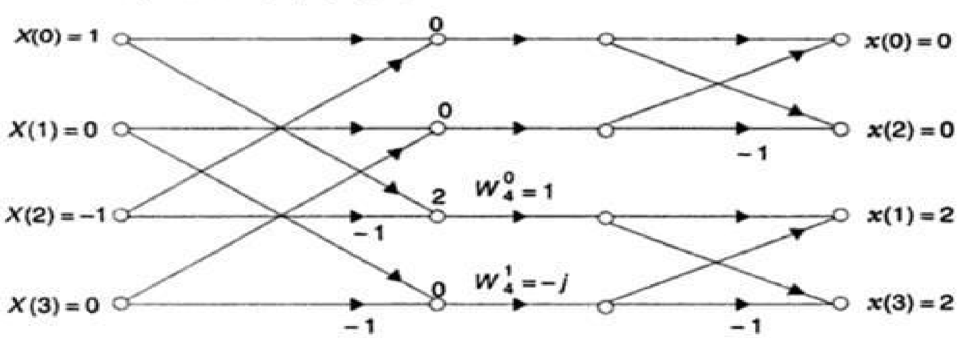


Fig.

Example Given $x(n) = \{1, 2, 3, 4, 4, 3, 2, 1\}$, find $X(k)$ using DIF FFT algorithm.

Solution Given $N = 8$.

We know that $W_N^k = e^{-j\left(\frac{2\pi}{N}\right)k}$.

Hence, $W_8^0 = 1$, $W_8^1 = 0.707 - j\,0.707$

$W_8^2 = -j$, $W_8^3 = -0.707 - j\,0.707$

Using DIF FFT algorithm, we can find $X(k)$ from the given sequence $x(n)$ as shown in Fig.

Hence, $X(k) = \{20, -5.828 - j\,2.414, 0, -0.172 - j\,0.414, 0, -0.172 + j\,0.414, 0, -5.828 + j\,2.414\}$

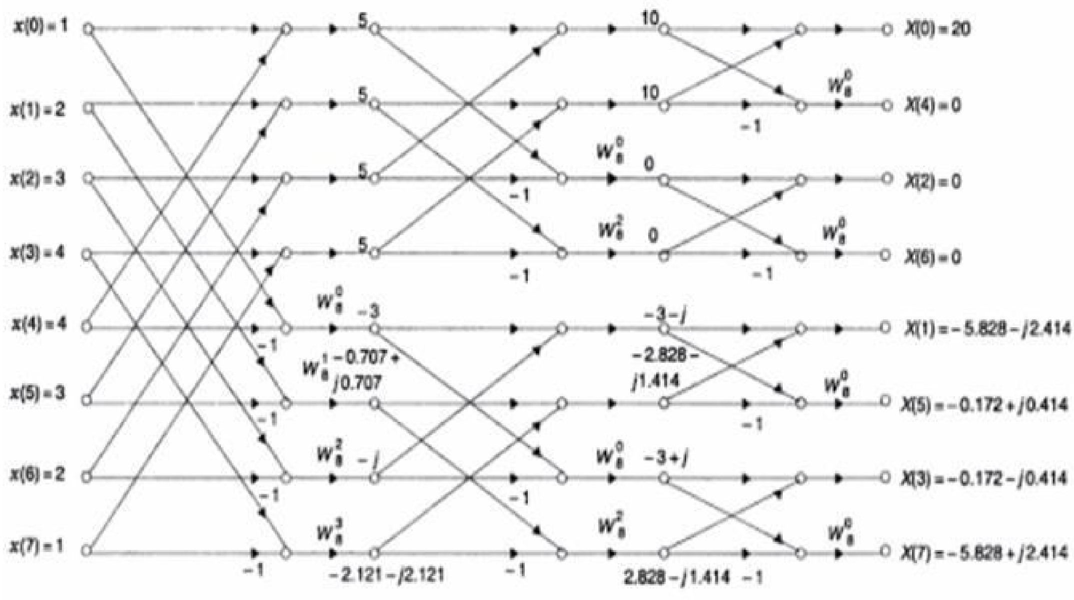


Fig.

INVERSE FFT:

An FFT algorithm can be used to compute the IDFT if the output is divided by N and the "twiddle factors are negative powers of W_N, i.e. powers of W_N^{-1} is used instead of powers of W_N. Therefore, an IFFT flow graph can be obtained from an FFT flow graph by replacing all the $x(n)$ by $X(k)$, dividing the input data by N, or dividing each stage by 2 when N is a power of 2, and changing the exponents of W_N to negative values.

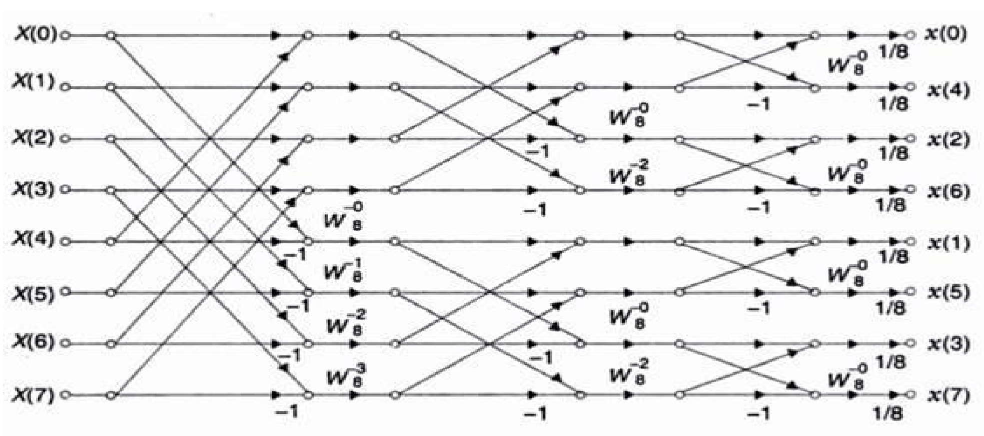


Fig. *Flow Graph of an IDFT Computation*

Example Use the 4-point inverse FFT and verify the DFT results {6, $-2 + j2$, -2, $-2 - j2$} obtained in Example 6.18 for the given input sequence {0, 1, 2, 3}.

Solution We know that $W_N^k = e^{-j\left(\frac{2\pi}{N}\right)k}$. Hence,

$W_4^{-0} = 1$ and $W_4^{-1} = e^{j\pi/2} = j$

Using IFFT algorithm, we can find the input sequence $x(n)$ from the given DFT sequence $X(k)$ as shown in Fig.

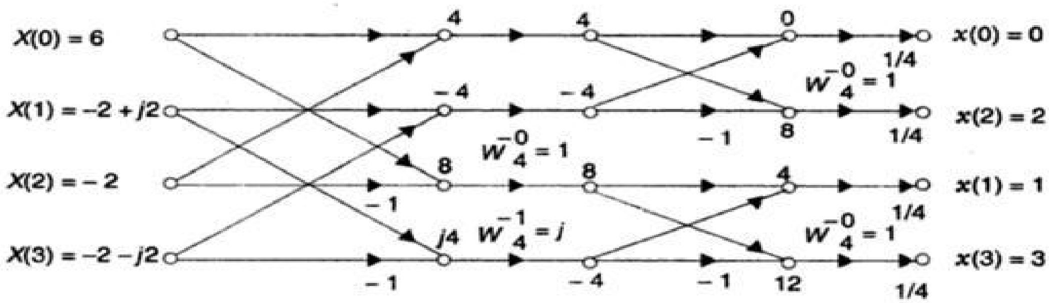


Fig.

Hence, $x(n) = \{0, 1, 2, 3\}$

SELECTION OF THE FILTER TYPE

The selection of the digital filter type *i.e.*, whether an IIR and FIR digital filter to be employed ; depends on the nature of the problem and on the specification of the desired frequency response. For example, FIR filters are used in filtering problems where there is a requirement for a linear phase characteristic within the passband of the filter. When linear phase is not a requirement, either an IIR or FIR filter can be used. However, in most cases, the order (N_{FIR}) of an FIR filter is considerably higher than the order (N_{IIR}) of an equivalent IIR filter meeting the same magnitude specifications. It has been shown that for most practical filter specifications, the ratio N_{FIR}/N_{IIR} is typically of order of ten or more and as a result, IIR filter is usually computationally more efficient.

In this chapter we shall discuss techniques for designing IIR filters from the analog filters, with the restriction that the filters be realizable and, of course, stable. There are four different methods which are available under IIR filter design, these are,

1. Impulse invariance method
2. Bilinear transformation method
3. Matched z-transform technique
4. Approximation of derivatives.

We shall concentrate only the first two methods.

IIR Filter Design by Impulse Invariance

A technique for digitizing an analog filter is called impulse invariance transformation. The objective of this method is to develop an IIR filter transfer function whose impulse response is the sampled version of the impulse response of the analog filter. The main idea behind this technique is to preserve the frequency response characteristics of the analog filter. In the consequence of the result, the frequency response of the digital filter is an aliased version of the frequency response of the corresponding analog filter.

To develop the necessary design formula for impulse invariance method, consider a causal and stable "analog" transfer function $H_a(s)$. Its impulse response $h_a(t)$ is given by inverse Laplace transform of $H_a(s)$, *i.e.*,

$$h_a(t) = L^{-1}\{H_a(s)\} \qquad \ldots(1)$$

In this method, we require that unit sample response $h(n)$ of the desired causal digital transfer function $H(z)$ be given by the sampled version of $h_a(t)$ sampled at uniform interval of T seconds.

i.e.,
$$h(n) = h_a(nT) \quad n = 0, 1, 2 \qquad \ldots(2)$$

where T is the sampling period

To investigate the mapping of points between z-plane and s-plane implied by the sampling process, the z-transform is related to the Laplace transform of $h_a(t)$ as

$$H(z)\big|_{z=e^{sT}} = Z\{h(n)\} = Z\{h_a(nT)\} \qquad \ldots(3)$$

$$= \frac{1}{T}\sum_{k=-\infty}^{\infty} H_a\left(s + j\frac{2\pi k}{T}\right) \qquad \ldots(4)$$

where,
$$H(z) = \sum_{n=0}^{\infty} h(n)\, z^{-n} \qquad \ldots(5)$$

and
$$H(z)\big|_{z=e^{sT}} = \sum_{n=0}^{\infty} h(n) e^{-sTn} \qquad \ldots(6)$$

where $$s = \sigma + j\Omega$$

Let us examine the transform $z = e^{ST}$ of eqn. (4) which can be written alternatively as,

$$z = e^{sT}$$

For $$s = \sigma + j\Omega$$

$$z = \gamma e^{j\omega} = e^{sT} = e^{\sigma T} = e^{j\Omega T}.$$

This then implies $$\Omega = e^{\sigma T}$$

$$\omega = \Omega T$$

where Ω is analog frequency and

ω is frequency in digital domain.

9.2.1.1 Development of the transformation

To explore the effect of the impulse invariance design method on the characteristics of resultant filter, let us consider the system function of the analog filter in the partial fraction form. Assume that the poles of the analog filter are distinct *i.e.,*

$$H_a(s) = \sum_{k=1}^{N} \frac{A_k}{s - p_k} \qquad ...(\ 7)$$

where $\{A_k\}$ are the co-efficients in the partial fraction expansion and

p_k are the poles of the analog filter. The impulse response $h_a(t)$ corresponding to eqn. (7) has the form

$$h_a(t) = \sum_{k=1}^{N} A_k e^{p_k t} u_a(t) \qquad ...(\ 8)$$

where $u_a(t)$ is the continuous time step function.

If we sample $h_a(t)$ periodically at $t = nT$, we have

$$h(n) = h_a(nT) = \sum_{k=1}^{N} A_k e^{p_k nT} u_a(nT) \qquad ...(\ 9)$$

Now, the system function H(z) of the digital filter is the z-transform of this sequence and is defined by

$$H(z) = z\{h(n)\}.$$

$$H(z) = \sum_{n=0}^{\infty} h(n)\, z^{-n} \qquad ...(\ 10)$$

Using eqn. (10) the system function becomes

$$H(z) = \sum_{n=0}^{\infty} \sum_{k=1}^{N} A_k\, e^{p_k nT}\, z^{-n} \qquad ...(\ 11)$$

$$= \sum_{k=1}^{N} \sum_{k=0}^{\infty} (e^{p_k T}\, z^{-1})^n \qquad ...(\ 12)$$

$$H(z) = \sum_{k=1}^{N} A_k \cdot \frac{1}{1 - e^{p_k T}\, z^{-1}} \qquad ...(\ 13)$$

provided that $|e^{p_k T}| < 1$, which is always satisfied if $p_k < 0$, indicating that $H_a(s)$ is a stable transfer function. From the eqn. (13) we observe that the digital filter has poles at

$$z_k = e^{p_k T} \quad k = 1, 2, ... N$$

Comparing the expression (13) and (7), we see that the impulse invariance transformation is accomplished by the mapping.

$$\frac{1}{s-p_k} \longrightarrow \frac{1}{1-e^{p_k T} z^{-1}} \qquad \text{...(14)}$$

$$\frac{1}{s+p_k} \longrightarrow \frac{1}{1-e^{-p_k T} z^{-1}}.$$

Problem 1. *For the analog transfer function* $H_a(s) = \dfrac{2}{(s+1)(s+2)}$ *determine the H(z) using impulse invariance method.*

Sol.
$$H_a(s) = \frac{2}{(s+1)(s+2)} = \frac{2}{s+1} - \frac{2}{s+2}$$

Using the impulse invariance transformation of eqn. (14), the digital filter transfer function

$$H(z) = \frac{Y(z)}{X(z)} = \frac{2}{1-e^{-T}z^{-1}} - \frac{2}{1-e^{-2T}z^{-1}} = \frac{2e^{-T}(1-e^{-T})z^{-1}}{(1-e^{-T}z^{-1})(1-e^{-2T}z^{-1})}.$$

Problem 2. *Convert the analog filter with system function*

$$H_a(s) = \frac{s+2}{(s+1)(s+3)}$$

into the digital IIR filter by means of the impulse invariance method.

Sol. The partial-fraction expansion of $H_a(s)$ is given as

$$H_a(s) = \frac{s+2}{(s+1)(s+3)} = \frac{\frac{1}{2}}{s+1} + \frac{\frac{1}{2}}{s+3}$$

Using eqn. (14) the corresponding digital filter is then

$$H(z) = \frac{1}{2}\left[\frac{1}{1-e^{-T}z^{-1}} + \frac{1}{1-e^{-3T}z^{-1}}\right]$$

$$= \frac{1}{2}\,\frac{2-z^{-1}(e^{-3T}+e^{-T})}{(1-e^{-T}z^{-1})(1-e^{-3T}z^{-1})}$$

$$= \frac{1}{2}\,\frac{[2-z^{-1}e^{-2T}(e^{-T}+e^{+T})]}{(1-e^{-T}z^{-1})(1-e^{-3T}z^{-1})}$$

$$H(z) = \frac{1-z^{-1}e^{-2T}\cosh T}{(1-e^{-T}z^{-1})(1-e^{-3T}z^{-1})}$$

It should be noted that zero of H(z) at $z = e^{-2T}\cosh T$ is not obtained by transforming the zero at $s = -z$ into a zero at $z = e^{-2T}$.

Problem *Apply the impulse invariant method to obtain the digital filter from the second order analog filter*

$$H_A(s) = \frac{s+a}{(s+a)^2+b^2}.$$

Sol. The analog filter transfer function is

$$H_A(s) = \frac{s+a}{(s+a+jb)(s+a-jb)}$$

Inverse Laplace transforming,

$$h_A(t) = \begin{cases} e^{-at}\cos(bt), & t \geq 0. \\ 0, & \text{otherwise.} \end{cases}$$

Sampling this function produces

$$h(nT_s) = \begin{cases} e^{-anT_s}\cos(bnT_s), & n \geq 0. \\ 0, & \text{otherwise.} \end{cases}$$

The z-transform of $h(nT_s)$, is equal to

$$H(z) = \sum_{n=0}^{\infty} e^{-anT_s}\cos(b_n T_s)\, z^{-n}$$

$$H(z) = \sum_{n=0}^{\infty} [e^{-aT_s}\cos(bT_s)\, z^{-1}]^n$$

$$H(z) = \frac{1 - e^{-aT}\cos(bT_s)\, z^{-1}}{(1 - e^{-(a+jb)T_s}\, z^{-1})(1 - e^{-(a-jb)T_s}\, z^{-1})}.$$

Problem *Using impulse invariance method with $T = 1$ sec determine*

$$H(z) \text{ if } H(s) = \frac{1}{s^2 + \sqrt{2}s + 1}.$$

Sol. Given that

$$H(s) = \frac{1}{s^2 + \sqrt{2}s + 1}$$

$$h(t) = L^{-1}(H(s)) = L^{-1}\left[\frac{1}{s^2 + \sqrt{2}s + 1}\right]$$

$$= L^{-1}\left[\frac{1}{\left(s + \frac{1}{\sqrt{2}}\right)^2 + \left(\frac{1}{\sqrt{2}}\right)^2}\right]$$

$$= L^{-1}\left[\sqrt{2}\,.\,\frac{\frac{1}{\sqrt{2}}}{\left(s + \frac{1}{\sqrt{2}}\right)^2 + \left(\frac{1}{\sqrt{2}}\right)^2}\right]$$

$$= \sqrt{2}\, L^{-1}\left[\frac{\frac{1}{\sqrt{2}}}{\left(s + \frac{1}{\sqrt{2}}\right)^2 + \left(\frac{1}{\sqrt{2}}\right)^2}\right]$$

$$= \sqrt{2}\, e^{-t/\sqrt{2}} \sin t/\sqrt{2}.$$

Let $\quad t = n\mathrm{T}$

$$h(n\mathrm{T}) = \sqrt{2}\, e^{-n\mathrm{T}/\sqrt{2}} \sin n\mathrm{T}/\sqrt{2}.$$

If $\quad \mathrm{T} = 1$ sec.

$$h(n) = \sqrt{2}\, e^{-n/\sqrt{2}} \sin n/\sqrt{2}.$$

$$\mathrm{H}(z) = z[h(n)] = \sqrt{2}\left[\frac{e^{-1/\sqrt{2}} z^{-1} \sin 1/\sqrt{2}}{1 - 2e^{-1/\sqrt{2}} z^{-1} \cos 1/\sqrt{2} + e^{-\sqrt{2}} z^{-2}}\right]$$

$$= \frac{0.453z^{-1}}{1 - 0.7497z^{-1} + 0.2432z^{-2}}.$$

IIR FILTER DESIGN BY THE BILINEAR TRANSFORMATION

The IIR filter design using (i) approximation of derivatives method and (ii) the impulse invariant method are appropriate for the design of low-pass filters and bandpass filters whose resonant frequencies are low. These techniques are not suitable for high-pass or band-reject filters. This limitation is overcome in the mapping technique called the ***bilinear transformation***. This transformation is a one-to-one mapping from the s-domain to the z-domain. That is, the bilinear transformation is a conformal mapping that transforms the $j\,\Omega$-axis into the unit circle in the z-plane only once, thus avoiding aliasing of frequency components. Also, the transformation of a stable analog filter results in a stable digital filter as all the poles in the left half of the s-plane are mapped onto points inside the unit circle of the z-domain. The bilinear transformation is obtained by using the trapezoidal formula for numerical integration. Let the system function of the analog filter be

$$H(s) = \frac{b}{s + a} \tag{2}$$

The differential equation describing the analog filter can be obtained from Eq. 2 as shown below.

$$H(s) = \frac{Y(s)}{X(s)} = \frac{b}{s + a}$$

$$sY(s) + aY(s) = bX(s)$$

Taking inverse Laplace transform,

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} + a\,y(t) = b\,x(t) \tag{3}$$

Eq. 3 is integrated between the limits $(nT - T)$ and nT

$$\int_{nT-T}^{nT} \frac{\mathrm{d}y(t)}{\mathrm{d}t}\,\mathrm{d}t + a\int_{nT-T}^{nT} y(t)\,\mathrm{d}t = b\int_{nT-T}^{nT} x(t)\,\mathrm{d}t \qquad 4$$

The trapezoidal rule for numeric integration is given by

$$\int_{nT-T}^{nT} a(t)\,\mathrm{d}t = \frac{T}{2}\,[a(nT) + a(nT - T)] \qquad 5$$

Applying Eq. 5 in Eq. 4 we get

$$y(nT) - y(nT-T) + \frac{aT}{2}y(nT) + \frac{aT}{2}y(nT-T) = \frac{bT}{2}x(nT) + \frac{bT}{2}x(nT-T)$$

Taking z-transform, the system function of the digital filter is,

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b}{\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + a} \qquad 6$$

Comparing Eqs. 2 and 6 we get,

$$s = \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) = \frac{2}{T}\left(\frac{z-1}{z+1}\right) \qquad 7$$

The general characteristic of the mapping $z = e^{sT}$ can be obtained by substituting $s = \sigma + j\,\Omega$ and expressing the complex variable z in the polar form as $z = re^{j\omega}$ in Eq. 7

$$s = \frac{2}{T}\left(\frac{z-1}{z+1}\right) = \frac{2}{T}\left(\frac{re^{j\omega} - 1}{re^{j\omega} + 1}\right)$$

$$= \frac{2}{T}\left(\frac{r^2 - 1}{1 + r^2 + 2r\cos\omega} + j\frac{2r\sin\omega}{1 + r^2 + 2r\cos\omega}\right)$$

Therefore,

$$\sigma = \frac{2}{T}\left(\frac{r^2 - 1}{1 + r^2 + 2r\cos\omega}\right) \qquad 8$$

$$\Omega = \frac{2}{T}\left(\frac{2r\sin\omega}{1 + r^2 + 2r\cos\omega}\right) \qquad 9$$

From Eq. 8 , it can be noted that if $r < 1$, then $\sigma < 0$, and if $r > 1$, then $\sigma > 0$. Thus, the left-half of the s-plane maps onto the points inside the unit circle in the z-plane and the transformation results in a stable digital system. Consider Eq. 9 for unity magnitude ($r = 1$), σ is zero. In this case,

$$\Omega = \frac{2}{T}\left(\frac{\sin\omega}{1+\cos\omega}\right)$$

$$= \frac{2}{T}\left(\frac{2\sin\omega/2\cos\omega/2}{\cos^2\omega/2+\sin^2\omega/2+\cos^2\omega/2-\sin^2\omega/2}\right)$$

$$\Omega = \frac{2}{T}\tan\frac{\omega}{2} \qquad 10$$

or equivalently,

$$\omega = 2\tan^{-1}\frac{\Omega T}{2} \qquad 11$$

Equation 11 gives the relationship between the frequencies in the two domains and this is shown in Fig. It can be noted that the entire range in Ω is mapped only once into the range $-\pi \le \omega \le \pi$. However, as seen in Fig. , the mapping is non-linear and the lower frequencies in analog domain are expanded in the digital domain, whereas the higher frequencies are compressed. This is due to the non-linearity of the arc tangent function and usually called as ***frequency warping***.

The warping effect can be eliminated by prewarping the analog filter. The effect of non-linear compression at high frequencies can be compensated by prewarping. When the desired magnitude response is piece-wise constant over frequency, this compression can be compensated by introducing a suitable prescaling or prewarping the critical frequencies by using the formula,

$$\Omega = \left(\frac{2}{T}\right)\tan\left(\frac{w}{2}\right)$$

$|_T| \quad |_2|$

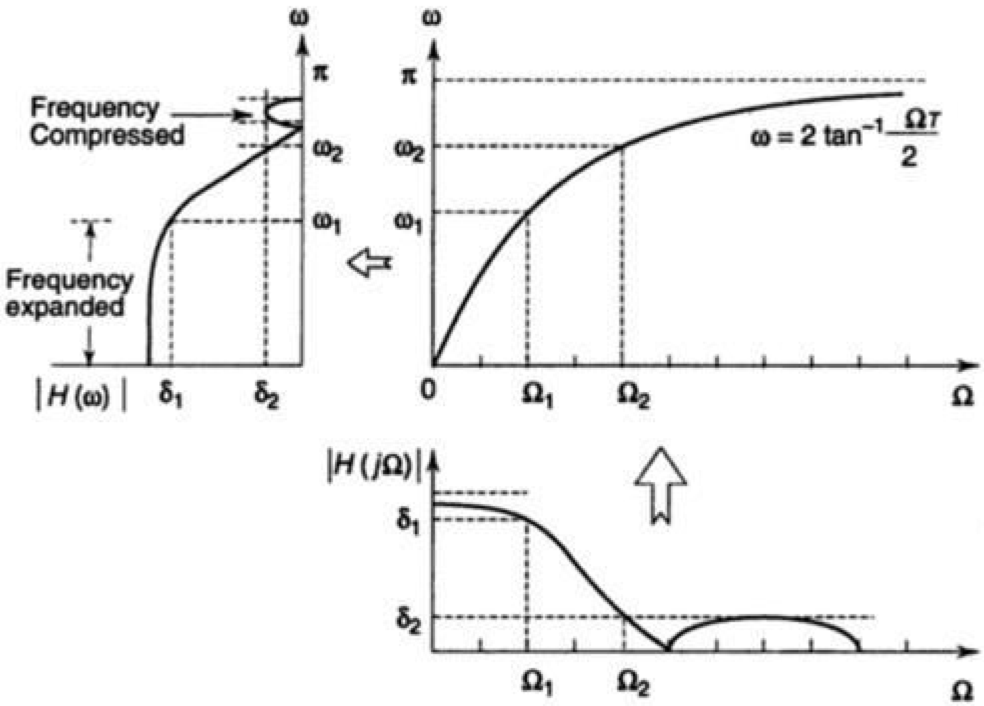


Fig. *Relationship Between ω and Ω as Given in Eq.* 11

Problem *Convert the analog filter with system function.*

$H_a(s) = \dfrac{s+0.1}{(s+0.1)^2+16}$ *into a digital IIR filter by means of bilinear transformation. Resonant frequency of a digital filter is given as* $w_r = \dfrac{\pi}{2}$.

Sol. (*i*) We first note that the analog filter $H_a(s)$ has a resonant frequency.

$$\Omega_r = \sqrt{16} = 4.$$

(*ii*) Let us find T

$$\Omega = \frac{2}{T}\tan\frac{w}{2}$$

$$4 = \frac{2}{T}\tan\frac{\pi}{4}$$

$$T = \frac{1}{2}.$$

(*iii*) Now map

$$S = \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) = 4\left(\frac{1-z^{-1}}{1+z^{-1}}\right)$$

By substituting values of s into H(s), we have,

$$H(z) = H_a(s)\Big|_{s=4\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

$$H(z) = \frac{4\left(\frac{1-z^{-1}}{1+z^{-1}}\right)+0.01}{\left[4\left(\frac{1-z^{-1}}{1+z^{-1}}\right)+0.01\right]^2+16}$$

$$= \frac{0.128 + 0.006z^{-1} - 0.122z^{-1}}{1 + 0.0006z^{-1} + 0.975z^{-2}} = \frac{0.128 + 0.006z^{-1} - 0.122z^{-1}}{1 + 0.975z^{-2}}$$

$$= \frac{(z+1)(z-0.95)}{(z - 0.987e^{-j\pi/2})(z - 0.987e^{j\pi/2})}$$

This filter has a pole $P_{1,2} = 0.98\, ze^{\pm j\pi/2}$ and zeros at $z_{1,2} = -1, 0.95$.

Problem *A first order Butterworth low pass transfer function with a 3dB cut off frequency at Ω_c is given by*

$$H_a(s) = \frac{\Omega_c}{s + \Omega_c}.$$

Design a single pole low pass with 3dB bandwidth of $0.2\,\pi$ using the bilinear transformation.

Sol. $\Omega_c = \frac{2}{T} \tan \frac{\omega c}{2}$.

Given that $\omega_c = 0.2\,\pi$

$$\Omega_c = \frac{2}{T} \tan \frac{0.2\,\pi}{2} = \frac{2}{T} \tan 0.1\,\pi = \frac{0.65}{T}.$$

The analog filter has a system function,

$$H_a(s) = \frac{\Omega_c}{s + \Omega_c} = \frac{0.65/T}{s + \frac{0.65}{T}}$$

Now
$$H(z) = H_a(s)\Big|_{s = \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} = \frac{0.65/T}{\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + \frac{0.65}{T}}$$

$$H(z) = \frac{0.65(1+z^{-1})}{2 - 2z^{-1} + 0.65} = \frac{0.245(1+z^{-1})}{(1 - 0.509z^{-1})}$$

The frequency response of the digital filter is

$$H(\omega) = \frac{0.245(1 + e^{-j\omega})}{1 - 0.509\, e^{-j\omega}}.$$

Thus at $\omega = 0$, $H(0) = 1$ and at $\omega = 0.2\,\pi$,

$|H(0.2\,\omega)| = 0.707$, which is a desired response.

Problem *Obtain $H(z)$ from $H_a(s)$ when $T = 1$ sec and $H_a(s) = \frac{s^3}{(s+1)(s^2+s+1)}$.*

Sol. Given that, $H_a(s) = \frac{s^3}{(s+1)(s^2+s+1)}$.

Put $s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}$ in $H_a(s)$ to get $H(z)$.

$$H(z) = \frac{\left[\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)\right]^3}{\left[\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + 1\right]\left[\left(\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)\right)^2 + \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + 1\right]}$$

$$= \frac{8(1-z^{-1})^3}{\left[2\left(1-z^{-1}\right)+T(1+z^{-1})\right]\left[4(1-z^{-1})^2+2T(1-z^{-1})(1+z^{-1})+T^2(1+z^{-1})^2\right]}$$

But T = 1 sec.

$$H(z) = \frac{8(1-z^{-1})^3}{(3-z^{-1})(7-6z^{-1}+3z^{-2})}$$

$$H(z) = \frac{2.67(z^{-1}-1)^3}{(z^{-2}-2z^{-1}+2.33)(z^{-1}-3)}.$$

Problem *Design a digital Butterworth filter satisfying the constraints*

$$0.707 \le |H(e^{j\omega})| \le 1 \quad \text{for } 0 \le \omega \le \frac{\pi}{2}$$

$$|H(e^{j\omega})| \le 0.2 \quad \text{for} \quad \frac{3\pi}{4} \le \omega \le \pi.$$

with T = 1 sec using The bilinear transformation

Sol. Bilinear transformation

Given that $\dfrac{1}{\sqrt{1+\varepsilon^2}} = 0.707$; $\dfrac{1}{\sqrt{1+\lambda^2}} = 0.2$, $\omega_p = \dfrac{\pi}{2}$; $\omega_s = \dfrac{3\pi}{4}$

The analog frequency ratio is

$$\frac{\Omega_s}{\Omega_p} = \frac{\frac{2}{T}\tan\frac{\omega_s}{2}}{\frac{2}{T}\tan\frac{\omega_p}{2}} = \frac{\tan\frac{3\pi}{8}}{\tan\frac{\pi}{4}} = 2.414$$

The order of the filter,

$$N \ge \frac{\log \lambda/\varepsilon}{\log \frac{\Omega_s}{\Omega_p}}.$$

From the given data $\lambda = 4.898$, $\varepsilon = 1$,

So, $$N \ge \frac{\log 4.898}{\log 2.414} = 1.803.$$

Rounding N to nearest higher value we get N = 2.

We know $$\Omega_c = \frac{\Omega_p}{(\varepsilon)^{1/N}} = \Omega_p \qquad (\because \ \varepsilon = 1)$$

$$= \frac{2}{T}\tan\frac{\omega_p}{2} = 2\tan\frac{\pi}{4} = 2 \text{ rad/sec}.$$

The transfer function of second order normalised Butterworth filter is,

$$H(s) = \frac{1}{s^2+\sqrt{2}s+1}.$$

$H_a(s)$ for $\Omega_c = 2$ rad/sec can be obtained by substituting

$s \to s/2$ in $H(s)$

i.e.,
$$H_a(s) = \frac{1}{(s/2)^2 + \sqrt{2}(s/2) + 1} = \frac{4}{s^2 + 2.828s + 4}.$$

By using bilinear transformation H(*z*) can be obtained as

$$H(z) = H(s)\Big|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}.$$

Thus
$$H(z) = \left.\frac{4}{s^2 + 2.828s + 4}\right|_{s=2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} \qquad (\because \quad T = 1 \text{ sec})$$

$$= \frac{4(1+z^{-1})^2}{4(1-z^{-1})^2 + 2.828\,(1-z^{-2}) + 4(1+z^{-1})^2}$$

$$= \frac{0.2929(1+z^{-1})^2}{1 + 0.1716z^{-2}}.$$