❖ What is a Signal

We are all immersed in a sea of signals. All of us from the smallest living unit, a cell, to the most complex living organism(humans) are all time time receiving signals and are processing them. Survival of any living organism depends upon processing the signals appropriately. What is signal? To define this precisely is a difficult task. Anything which carries information is a sig- nal. In this course we will learn some of the mathematical representations of the signals, which has been found very useful in making information process- ing systems. Examples of signals are human voice, chirping of birds, smoke signals, gestures (sign language), fragrances of the flowers. Many of our body functions are regulated by chemical signals, blind people use sense of touch. Bees communicate by their dancing pattern.Some examples of modern high speed signals are the voltage charger in a telephone wire, the electromagnetic field emanating from a transmitting antenna,variation of light intensity in an optical fiber. Thus we see that there is an almost endless variety of signals and a large number of ways in which signals are carried from on place to another place.In this course we will adopt the following definition for the signal: A signal is a real (or complex) valued function of one or more real variable(s).When the function depends on a single variable, the signal is said to be one- dimensional. A speech signal, daily maximum temperature, annual rainfall at a place, are all examples of a one dimensional signal.When the function depends on two or more variables, the signal is said to be multidimensional. An image is representing the two dimensional signal,vertical and horizon- tal coordinates representing the two dimensions. Our physical world is four dimensional(three spatial and one temporal).

❖ What is signal processing

By processing we mean operating in some fashion on a signal to extract some useful information. For example when we hear same thing we use our ears and auditory path ways in the brain to extract the information. The signal is processed by a system. In the example mentioned above the system is biological in nature. We can use an electronic system to try to mimic this behavior. The signal processor may be an electronic system, a mechanical system or even it might be a computer program.The word digital in digital signal processing means that the processing is done either by a digital hardware or by a digital computer.

❖ Analog versus digital signal processing

The signal processing operations involved in many applications like communication systems, control systems, instrumentation, biomedical signal processing etc can be implemented in two different ways

(1) Analog or continuous time method and

(2) Digital or discrete time method.

The analog approach to signal processing was dominant for many years. The analog signal processing uses analog circuit elements such as resistors, ca-

pacitors, transistors, diodes etc. With the advent of digital computer and later microprocessor, the digital signal processing has become dominant now a days. The analog signal processing is based on natural ability of the analog system to solve differential equations the describe a physical system. The solution

are obtained in real time. In contrast digital signal processing relies on numerical calculations. The method may or may not give results in real time. The digital approach has two main advantages over analog approach

(1) Flexibility: Same hardware can be used to do various kind of signal processing operation,while in the core of analog signal processing one has todesign a system for each kind of operation.

(2) Repeatability: The same signal processing operation can be repeated again and again giving same results, while in analog systems there may be parameter variation due to change in temperature or supply voltage.

The choice between analog or digital signal processing depends on application. One has to compare design time,size and cost of the implementation.

❖ Classification of signals

As mentioned earlier, we will use the term signal to mean a real or complex valued function of real variable(s). Let us denote the signal by $x(t)$. The variable t is called independent variable and the value x of t as dependent variable. We say a signal is continuous time signal if the independent variable t takes values in an interval.

For example $t \varepsilon \ (-\infty, \infty), or\ t \ \varepsilon \ [0, \infty] or\ t \ \varepsilon \ [T_0, T_1]$

The independent variable t is referred to as time,even though it may not be actually time. For example in variation if pressure with height t refers above mean sea level.

When t takes a vales in a countable set the signal is called a discrete time signal. For example

$T \varepsilon \ \{0, T, 2T, 3T, 4T, ...\}$ or $t \ \varepsilon \ \{... -1, 0, 1, ...\}$ or $t \varepsilon \ \{1/2, 3/2, 5/2, 7/2, ...\}$ etc.

For convenience of presentation we use the notation x[n] to denote discrete time signal.

Let us pause here and clarify the notation a bit. When we write $x(t)$ it has two meanings. One is value of x at time t and the other is the pairs$(x(t), t)$ allowable value of t. By signal we mean the second interpretation. To keep this distinction we will use the following notation: $\{x(t)\}$ to denote the con- tinuous time signal. Here $\{x(t)\}$ is short notation for $\{x(t), t \ I\}$ where I is the set in which t takes the value. Similarly for discrete time signal we will use the notation $\{x[n]\}$, where $\{x[n]\}$ is short for $\{x[n], n \ I\}$. Note that in $\{x(t)\}$ and $\{x[n]\}$ are dummy variables ie. $\{x[n]\}$ and $\{x[t]\}$ refer to the same signal. Some books use the notation $x[\cdot]$ to denote $\{x[n]\}$ and $x[n]$ to denote value of x at time $n \cdot x[n]$ refers to the whole waveform,while $x[n]$ refers to a particular

value. Most of the books do not make this distinction clean and use $x[n]$ to denote signal and $x[n_0]$ to denote a particular value. As with independent variable t, the dependent variable x can take values in a continues set or in a countable set. When both the dependent and inde- pendent variable take value in intervals, the signal is called an analog signal. When both the dependent and independent variables take values in countable sets(two sets can be quite different) the signal is called Digital signal.When we use digital computers to do processing we are doing digital signal pro- cessing. But most of the theory is for discrete time signal processing where default variable is continuous. This is because of the mathematical simplic- ity of discrete time signal processing. Also digital signal processing tries to implement this as closely as possible. Thus what we study is mostly discrete time signal processing and what is really implemented is digital signal pro- cessing.

Exercise:

1.GIve examples of continues time signals.
2.Give examples of discrete time signals.
3.Give examples of signal where the independent variable is not time(one-dimensional).
4.Given examples of signal where we have one independent variable but dependent variable has more than one dimension.(This is sometimes called vector valued signal or multichannel signal).
5.Give examples of signals where dependent variable is discrete but independent variable are continues.

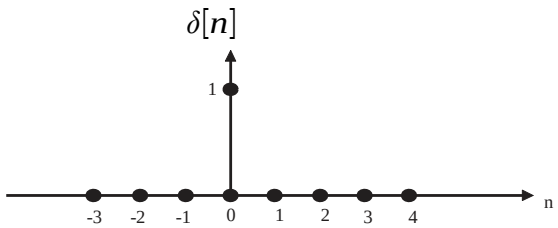
❖ Elementary signals

There are several elementary signals that feature prominently in the study of digital signals and digital signal processing.

(a)Unit sample sequence $\delta[n]$: Unit sample sequence is defined by

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n = 0 \end{cases}$$

Graphically this is as shown below.

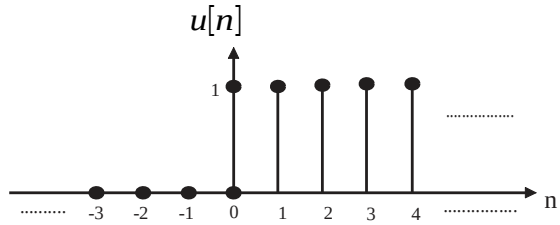


Unit sample sequence is also known as impulse sequence. This plays role akin to the impulse function $\delta(t)$ of continues time. The continues time impulse $\delta(t)$ is purely a mathematical construct while in discrete time we can actually generate the impulse sequence.

(b)Unit step sequence $u[n]$: Unit step sequence is defined by

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

Graphically this is as shown below

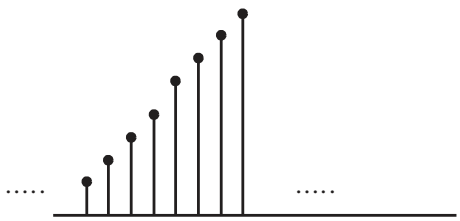


(c) Exponential sequence: The complex exponential signal or sequence $x[n]$ is defined by

$$x[n] = C\ \alpha^n$$

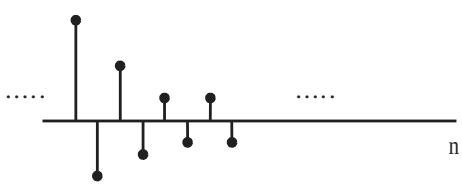
where C and α are, in general, complex numbers. Note that by writing $\alpha = e^{\beta}$, we can write the exponential sequence as $x[n] = c\ e^{\beta n}$.

Real exponential signals: If C and α are real, we can have one of the several type of behavior illustrated below

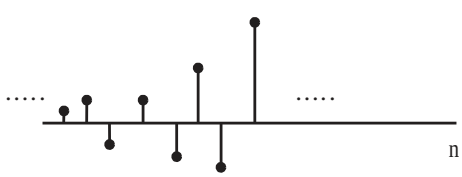


$\{x[n] = \alpha^n,\ \alpha > 1\}$

$\{x[n] = \alpha^n,\ 0 < \alpha < 1\}$



$\{x[n] = \alpha^n,\ -1 < \alpha < 0\}$



$\{x[n] = \alpha^n,\ \alpha < -1\}$

if $|\alpha| > 1$ the magnitude of the signals grows exponentially, whlie if $|\alpha| < 1$, we have decaying exponential. If α is positive all terms of $\{x[n]\}$ have same sign, but if α is negative the sign of terms in $\{x[n]\}$ alternates.

(d)Sinusoidal Signal: The sinusoidal signal $\{x[n]\}$ is defined by

$$x[n] = A\ \cos(w_0 n + \varphi)$$

Euler's relation allows us to relate complex exponentials and sinusoids.

$$e^{j w_0 n} = \cos w_0 n + j\ \sin w_0 n$$

and

$A \cos(w_0 n + \varphi) = 1/2\ \{\ A\, e^{j\varphi} e^{j w_0 n} + A\, e^{-j\varphi} e^{-j w_0 n})$

The general discrete time complex exponential can be written in terms of real exponential and sinusiodal signals.Specifically if we write c and α in polar for $C = |C|e^{j\theta}$ and $\alpha = |\alpha|e^{j w_0}$ then

$$C\,\alpha^n = |C||\alpha|^n \cos(w_0 n + \theta) + \boldsymbol{j}|C||\alpha|^n \sin(w_0 n + \theta)$$

Thus for $|\alpha| = 1$, the real and imaginary parts of a cmplex exponential sequence are sinusoidal. For $|\alpha| < 1$, they correspond to sinusoidal sequence multiplied by a decaying exponential, and for $|\alpha| > 1$ they correspond to sinusiodal sequence multiplied by a growing exponential.

❖ Generating Signals with MATLAB

MATLAB, acronym for MATrix LABoratory has become a very porplar soft- ware environment for complex based study of signals and systems. Here we give some sample programmes to generate the elementary signals discussed above. For details one should consider MATLAB manual or read help files. In MATLAB, ones(M,N) is an M-by-N matrix of ones, and zeros(M,N) is an M-by-N matrix of zeros. We may use those two matrices to generate impulse and step sequence.

The following is a program to generate and display impulse sequence.

```
>> % Program to generate and display impulse response sequence
>> n = −49 : 49;
>> delta = [zeros(1, 49), 1, zeros(1, 49)];
>> stem(n, delta)
```

Here >> indicates the MATLAB prompt to type in a command, stem(n,x) depicts the data contained in vector x as a discrete time signal at time values defined by n. One can add title and lable the axes by suitable commands. To generate step sequence we can use the following program

```
>> % Program to generate and display unit step function
>> n = −49 : 49;
>> u = [zeros(1, 49), ones(1, 50)];
>> stem(n, u);
```

We can use the following program to generate real exponential sequence

```
>> % Program to generate real exponential sequence
>> C = 1;
>> alpha = 0.8;
>> n = −10 : 10;
>> x = C ∗ alpha. ∧ n
>> stem(n, x)
```

Note that, in their program, the base alpha is a scalar but the exponent is a vector, hence use of the operator .∧ to denote element-by-element power.

Exercise: Experiment with this program by changing different values of alpha (real). Values of alpha greater then 1 will give growing exponential and less than 1 will give decaying exponentials.

❖ **Introduction to DSP**

A signal is any variable that carries information. Examples of the types of signals of interest are Speech (telephony, radio, everyday communication), Biomedical signals (EEG brain signals), Sound and music, Video and image,_ Radar signals (range and bearing).

Digital signal processing (DSP) is concerned with the digital representation of signals and the use of digital processors to analyse, modify, or extract information from signals. Many signals in DSP are derived from analogue signals which have been sampled at regular intervals and converted into digital form. The key advantages of DSP over analogue processing are Guaranteed accuracy (determined by the number of bits used), Perfect reproducibility, No drift in performance due to temperature or age, Takes advantage of advances in semiconductor technology, Greater exibility (can be reprogrammed without modifying hardware), Superior performance (linear phase response possible, and_ltering algorithms can be made adaptive), Sometimes information may already be in digital form. There are however (still) some disadvantages, Speed and cost (DSP design and hardware may be expensive, especially with high bandwidth signals) Finite word length problems (limited number of bits may cause degradation).

Application areas of DSP are considerable: _ Image processing (pattern recognition, robotic vision, image enhancement, facsimile, satellite weather map, animation), Instrumentation and control (spectrum analysis, position and rate control, noise reduction, data compression) _ Speech and audio (speech recognition, speech synthesis, text to Speech, digital audio, equalisation) Military (secure communication, radar processing, sonar processing, missile guidance) Telecommunications (echo cancellation, adaptive equalisation, spread spectrum, video conferencing, data communication) Biomedical (patient monitoring, scanners, EEG brain mappers, ECG analysis, X-ray storage and enhancement).

UNIT I
DISCRETE FOURIER TRANSFORM

1.1 Discrete-time signals

A discrete-time signal is represented as a sequence of numbers:

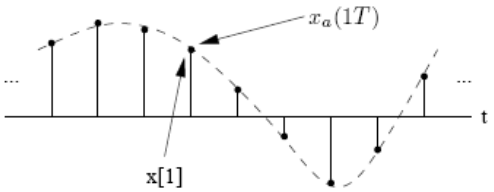
$$x = \{x[n]\}, \qquad -\infty < n < \infty.$$

Here n is an integer, and x[n] is the nth sample in the sequence. Discrete-time signals are often obtained by sampling continuous-time signals. In this case the nth sample of the sequence is equal to the value of the analogue signal xa(t) at time t = nT:

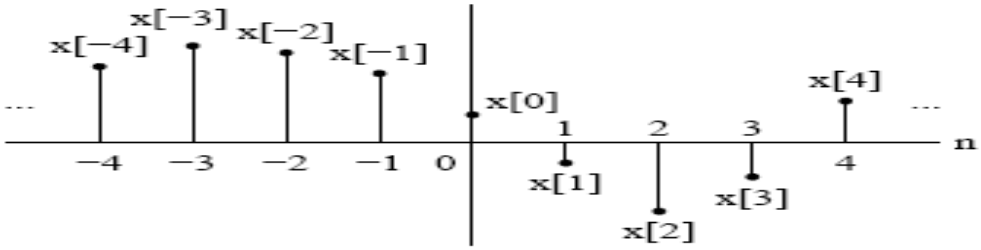
$$x[n] = x_a(nT), \qquad -\infty < n < \infty.$$

The sampling period is then equal to T, and the sampling frequency is fs = 1/T .

x[1]



For this reason, although x[n] is strictly the nth number in the sequence, we often refer to it as the nth sample. We also often refer to \the sequence x[n]" when we mean the entire sequence. Discrete-time signals are often depicted graphically as follows:

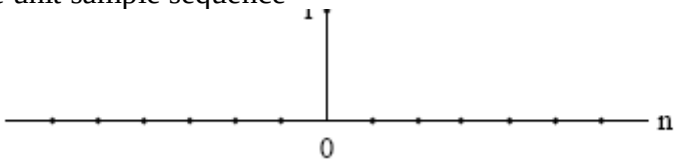


(This can be plotted using the MATLAB function stem.) The value x[n] is unde_ned for no integer values of n. Sequences can be manipulated in several ways. The sum and product of two sequences x[n] and y[n] are de_ned as the sample-by-sample sum and product respectively. Multiplication of x[n] by a is de_ned as the multiplication of each sample value by a. A sequence y[n] is a delayed or shifted version of x[n] if

$$y[n] = x[n - n_0],$$

with n0 an integer.

The unit sample sequence

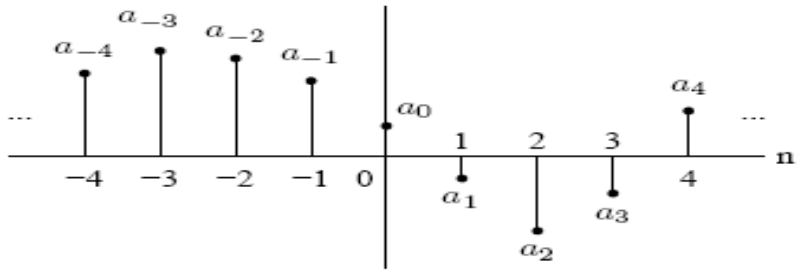


is defined as

$$\delta[n] = \begin{cases} 0 & n \neq 0 \\ 1 & n = 0. \end{cases}$$

This sequence is often referred to as a discrete-time impulse, or just impulse. It plays the same role for discrete-time signals as the Dirac delta function does for continuous-time signals. However, there are no mathematical complications in its definition.

An important aspect of the impulse sequence is that an arbitrary sequence can be represented as a sum of scaled, delayed impulses. For example, the

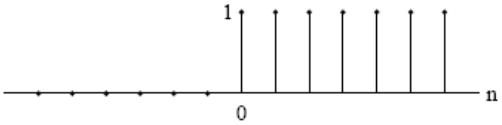


Sequence can be represented as

$$x[n] = a_{-4}\delta[n+4] + a_{-3}\delta[n+3] + a_{-2}\delta[n+2] + a_{-1}\delta[n+1] + a_0\delta[n] + a_1\delta[n-1] + a_2\delta[n-2] + a_3\delta[n-3] + a_4\delta[n-4].$$

In general, any sequence can be expressed as

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



The unit step sequence is defined as

$$u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0. \end{cases}$$

$$u[n] = \sum_{k=-\infty}^{n} \delta[k].$$

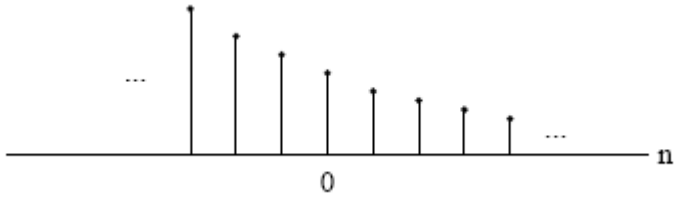
The unit step is related to the impulse by
Alternatively, this can be expressed as

$$u[n] = \delta[n] + \delta[n-1] + \delta[n-2] + \cdots = \sum_{k=0}^{\infty} \delta[n-k].$$

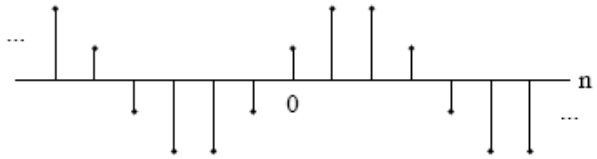
Conversely, the unit sample sequence can be expressed as the _rst backward difference of the unit step sequence

$$\delta[n] = u[n] - u[n-1].$$

Exponential sequences are important for analyzing and representing discrete-time systems. The general form is
$x[n] = A\alpha^n.$
If A and _ are real numbers then the sequence is real. If 0 < _ < 1 and A is positive, then the sequence values are positive and decrease with increasing n:



For □1 < _ < 0 the sequence alternates in sign, but decreases in magnitude. For j_j > 1 the sequence grows in magnitude as n increases.



A sinusoidal sequence has the form

$$x[n] = A\cos(\omega_0 n + \phi) \qquad \text{for all } n,$$

with A and ϕ real constants. The exponential sequence $A\alpha^n$ with complex $\alpha = |\alpha|e^{j\omega_0}$ and $A = |A|e^{j\phi}$ can be expressed as

$$\begin{aligned} x[n] = A\alpha^n &= |A|e^{j\phi}|\alpha|^n e^{j\omega_0 n} = |A||\alpha|^n e^{j(\omega_0 n+\phi)} \\ &= |A||\alpha|^n \cos(\omega_0 n + \phi) + j|A||\alpha|^n \sin(\omega_0 n + \phi), \end{aligned}$$

so the real and imaginary parts are exponentially weighted sinusoids

When $|\alpha| = 1$ the sequence is called the **complex exponential sequence**:

$$x[n] = |A|e^{j(\omega_0 n+\phi)} = |A|\cos(\omega_0 n + \phi) + j|A|\sin(\omega_0 n + \phi).$$

The frequency of this complex sinusoid is!0, and is measured in radians per sample. The phase of the signal is. The index n is always an integer. This leads to some important

Differences between the properties of discrete-time and continuous-time complex exponentials: $(\omega_0 + 2\pi)$.
Consider the complex exponential with frequency

$$x[n] = Ae^{j(\omega_0+2\pi)n} = Ae^{j\omega_0 n}e^{j2\pi n} = Ae^{j\omega_0 n}.$$

Thus the sequence for the complex exponential with frequency $(\omega_0 + 2\pi)$ is exactly the same as that for the complex exponential with frequency more generally; complex exponential sequences with frequencies $(\omega_0 + 2\pi)$ where r is an integer are indistinguishable

From one another. Similarly, for sinusoidal sequences

$$x[n] = A\cos[(\omega_0 + 2\pi r)n + \phi] = A\cos(\omega_0 n + \phi).$$

In the continuous-time case, sinusoidal and complex exponential sequences are always periodic. Discrete-time sequences are periodic (with period N) if x[n] = x[n + N] for all n: $x[n] = A\cos(\omega_0 n + \phi)$

$$A\cos(\omega_0 n + \phi) = A\cos(\omega_0 n + \omega_0 N + \phi),$$

Thus the discrete-time sinusoid is only periodic if which requires that

$$\omega_0 N = 2\pi k \qquad \text{for } k \text{ an integer.}$$

The same condition is required for the complex exponential

Sequence $Ce^{j\omega_0 n}$ to be periodic. The two factors just described can be combined to reach the conclusion that there are only N distinguishable frequencies for which the

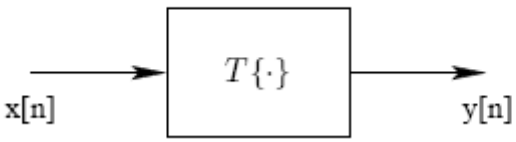
Corresponding sequences are periodic with period N. One such set is

$$\omega_k = \frac{2\pi k}{N}, \qquad k = 0, 1, \ldots, N-1.$$

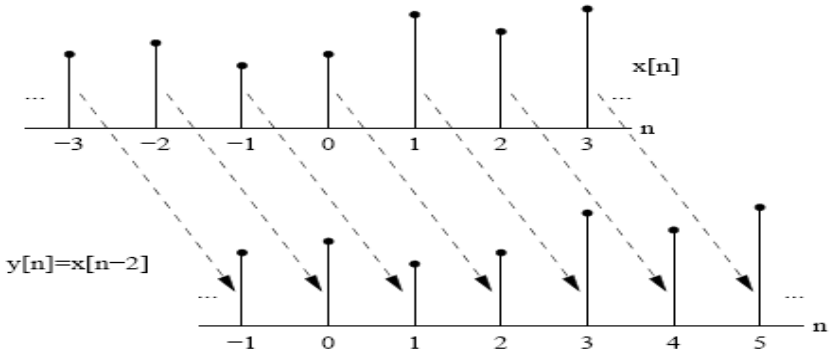
1.2 Discrete-time systems

A discrete-time system is de_ned as a transformation or mapping operator that maps an input signal x[n] to an output signal y[n]. This can be denoted as

$$y[n] = T\{x[n]\}.$$



Example: Ideal delay

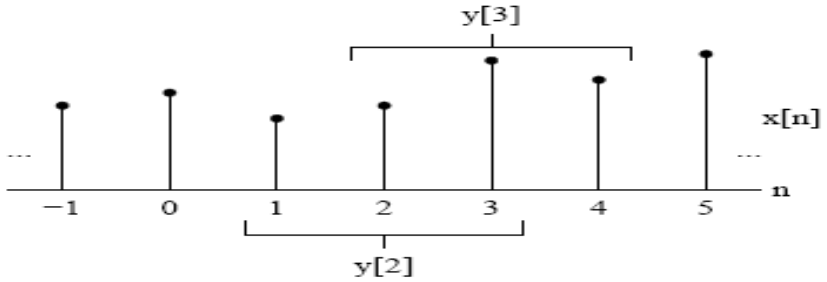


$y[n] = x[n - n_d]$:

Example: Moving average

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

For $M_1 = 1$ and $M_2 = 1$, the input sequence



yields an output with

$$\vdots$$

$$y[2] = \frac{1}{3}(x[1] + x[2] + x[3])$$

$$y[3] = \frac{1}{3}(x[2] + x[3] + x[4])$$

$$\vdots$$

Memoryless systems

A system is memory less if the output y[n] depends only on x[n] at the Same n. For example, y[n] = (x[n]) 2 is memory less, but the ideal delay $y[n] = x[n - n_d]$ is not unless $n_d = 0$.

Linear systems

A system is linear if the principle of superposition applies. Thus if y1[n] is the response of the system to the input x1[n], and y2[n] the response to x2[n], then linearity implies

Additivity:

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

Scaling:

$$T\{ax_1[n]\} = aT\{x_1[n]\} = ay_1[n].$$

These properties combine to form the general principle of superposition

$$T\{ax_1[n] + bx_2[n]\} = aT\{x_1[n]\} + bT\{x_2[n]\} = ay_1[n] + by_2[n].$$

In all cases a and b are arbitrary constants. This property generalises to many inputs, so the response of a linear system to $x[n] = \sum_k a_k x_k[n]$ will be $y[n] = \sum_k a_k y_k[n]$.

Time-invariant systems

A system is time invariant if times shift or delay of the input sequence Causes a corresponding shift in the output sequence. That is, if y[n] is the response to x[n], then y[n -n0] is the response to x[n -n0].

For example, the accumulator system

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

is time invariant, but the compressor system

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

for M a positive integer (which selects every Mth sample from a sequence) is not.

Causality

A system is causal if the output at n depends only on the input at n and earlier inputs. For example, the backward difference system

$$y[n] = x[n] - x[n-1]$$

is causal, but the forward difference system

$$y[n] = x[n+1] - x[n]$$

is not.

Stability

A system is stable if every bounded input sequence produces a bounded output sequence:

- **Bounded input:** $|x[n]| \leq B_x < \infty$
- **Bounded output:** $|y[n]| \leq B_y < \infty$.

For example, the accumulator

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

x[n]

is an example of an unbounded system, since its response to the unit

$$y[n] = \sum_{k=-\infty}^{n} u[n] = \begin{cases} 0 & n < 0 \\ n+1 & n \geq 0, \end{cases}$$

This has no _nite upper bound.

Linear time-invariant systems

If the linearity property is combined with the representation of a general sequence as a linear combination of delayed impulses, then it follows that a linear time-invariant (LTI) system can be completely characterized by its impulse response. Suppose hk[n] is the response of a linear system to the impulse h[n -k] at n = k. Since

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

the principle of superposition means that

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

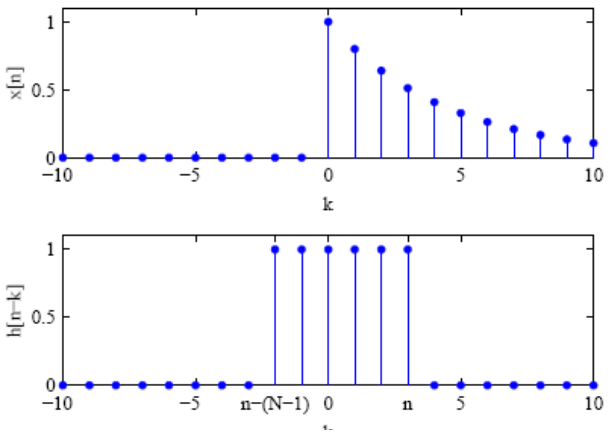
If the system is additionally time invariant, then the response to _[n -k] is h[n -k]. The previous equation then becomes

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

This expression is called the convolution sum. Therefore, a LTI system has the property that given h[n], we can _nd y[n] for any input x[n]. Alternatively, y[n] is the convolution of x[n] with h[n], denoted as follows:

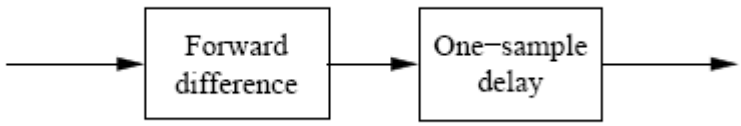
$$y[n] = x[n] * h[n].$$

The previous derivation suggests the interpretation that the input sample at n = k, represented by $x[k]\delta[n-k]$, is transformed by the system into an output sequence $x[k]h[n-k]$. For each k, these sequences are superimposed to yield the overall output sequence: A slightly different interpretation, however, leads to a convenient computational form: the nth value of the output, namely y[n], is obtained by multiplying the input sequence (expressed as a function of k) by the sequence with values h[n-k], and then summing all the values of the products x[k]h[n-k]. The key to this method is in understanding how to form the sequence h[n -k] for all values of n of interest. To this end, note that h[n -k] = h[- (k -n)]. The sequence h[-k] is seen to be equivalent to the sequence h[k] rejected around the origin



Since the sequences are non-overlapping for all negative n, the output must be zero y[n] = 0; n < 0:

Consider the system



1.3 Introduction to DFT

The discrete-time Fourier transform (DTFT) of a sequence is a continuous function of !, and repeats with period 2_. In practice we usually want to obtain the Fourier components using digital computation, and can only evaluate them for a discrete set of frequencies. The discrete Fourier transform (DFT) provides a means for achieving this. The DFT is itself a sequence, and it corresponds roughly to samples, equally

spaced in frequency, of the Fourier transform of the signal. The discrete Fourier transform of a length N signal x[n], n = 0; 1; : : : ;N -1 is given by

$$X[k] = \sum_{n=0} x[n]e^{-j(2\pi/N)kn}.$$

This is the analysis equation. The corresponding synthesis equation is

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j(2\pi/N)kn}.$$

When dealing with the DFT, it is common to define the complex quantity

$$W_N = e^{-j(2\pi/N)}.$$

With this notation the DFT analysis-synthesis pair becomes

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn}$$

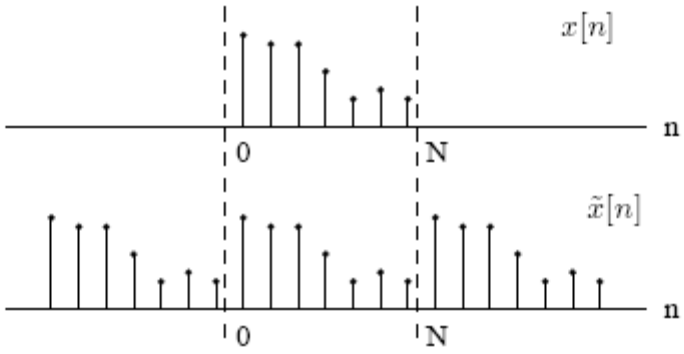
$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-kn}.$$

An important property of the DFT is that it is cyclic, with period N, both in the discrete-time and discrete-frequency domains. For example, for any integer r,

$$X[k+rN] = \sum_{n=0}^{N-1} x[n]W_N^{(k+rN)n} = \sum_{n=0}^{N-1} x[n]W_N^{kn}(W_N^N)^{rn}$$
$$= \sum_{n=0}^{N-1} x[n]W_N^{kn} = X[k],$$

since $W_N^N = e^{-j(2\pi/N)N} = e^{-j2\pi} = 1.$ Similarly, it is easy to show that x[n + rN] = x[n], implying periodicity of the synthesis equation. This is important | even though the DFT only depends on samples in the interval 0 to N -1, it is implicitly assumed that the signals repeat with period N in both the time and frequency domains. To this end, it is sometimes useful to de_ne the periodic extension of the signal x[n] to be To this end, it is sometimes useful to de_ne the periodic extension of the signal x[n] to be x[n] = x[n mod N] = x[((n))N]: Here n mod N and ((n))N are taken to mean n modulo N, which has the value of the remainder after n is divided by N. Alternatively, if n is written in the form n = kN + l for 0 < l < N, then n mod N = ((n))N = l:

$n \bmod N = ((n))_N = l.$



Similarly, the periodic extension of $X[k]$ is defined to be

$$\tilde{X}[k] = X[k \bmod N] = X[((k))_N].$$

It is sometimes better to reason in terms of these periodic extensions when dealing with the DFT. Specifically, if X[k] is the DFT of x[n], then the inverse DFT of X[k] is ~x[n]. The signals x[n] and ~x[n] are identical over the interval 0 to N □ 1, but may differ outside of this range. Similar statements can be made regarding the transform Xf[k].

Note:

1. $x(t)$ --- Continuous-time signal
   $X(f)$ --- Fourier Transform, frequency characteristics

   Can we find $X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt$

   if we don't have a mathematical equation for $x(t)$? No!

2. What can we do?
   (1) Sample $x(t)$ =>
   $x_0, x_1, \ldots, x_{N-1}$ over T (for example 1000 seconds)
   Sampling period (interval) Δt
   N (samples) over T => $\Delta t = T / N$
   Can we have infinite T and N? Impossible!

   (2) Discrete Fourier Transform (DFT):

   $$=> X_k = \sum_{n=0}^{N-1} x_n e^{-j2\pi kn/N} \qquad k = 1, 2, \ldots, N$$

   for the line spectrum at frequency $\omega_k = (2\pi)\frac{k}{T}$

3. Limited N and T =>

   limited frequency resolution $2\pi \frac{1}{T}$

   limited frequency band (from $-\infty$ to ∞ in Fourier transform to):

   $$0 \le \omega < 2\pi N / T$$

4. $x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{j2\pi kn/N}$ ---- periodic function (period N)

$x(t)$ --- general function

↓ sampling and inverse transform

x_n --- periodic function

5. X_k ($\omega_k = 2\pi\frac{k}{T}$ line spectrum)

$$X_k = \sum_{n=0}^{N-1} x_n e^{-j2\pi kn/N}$$ period function (period N)

1.4 Properties of the DFT

Many of the properties of the DFT are analogous to those of the discrete-time Fourier transform, with the notable exception that all shifts involved must be considered to be circular, or modulo N. Defining the DFT pairs $x[n] \overset{\mathcal{D}}{\longleftrightarrow} X[k]$, $x_1[n] \overset{\mathcal{D}}{\longleftrightarrow} X_1[k]$, and $x_2[n] \overset{\mathcal{D}}{\longleftrightarrow} X[k]$, the following are properties of the DFT:

Properties

1. Linearity : $Ax(n) + By(n) \leftrightarrow AX(k) + BX(k)$

2. Time Shift: $x(n-m) \leftrightarrow X(k)e^{-j2\pi km/N} = X(k)W_N{}^{k-m}$

3. Frequency Shift:

$$x(n)e^{j2\pi km/N} \leftrightarrow X(k-m)$$

4. Duality : $N^{-1}x(n) \leftrightarrow X(-k)$

why? $X(k) = \sum_{m=0}^{N-1} x(m)e^{-j2\pi mk/N}$

$DFT(X(n)) = \sum_{n=0}^{N-1} X(n)e^{-j2\pi nk/N}$

DFT of x(m)

$$x(-n) = x(N-n)$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{j2\pi k(N-n)/N}$$

$$e^{j2\pi k(N-n)/N} = e^{j2\pi kN/N}e^{-j2\pi kn/N}$$

$$x(-n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{-j2\pi kn/N}$$

$$\Rightarrow DFT(N^{-1}X(n)) = \frac{1}{N}\sum_{n=0}^{N-1} X(n)e^{-j2\pi nk/N} = x(-k)$$

5. Circular convolution

$$\sum_{m=0}^{N-1} x(m)y(n-m) = x(n)\mathrm{O}y(n) \leftrightarrow X(k)Y(k)$$ circular convolution

6. Multiplication

$$\underbrace{x(n)y(n)}_{\substack{new\ \ sequence \\ z(n)=x(n)y(n)}} \leftrightarrow N^{-1}\sum_{m=0}^{N-1} X(m)Y(k-m) = N^{-1}X(k)\mathrm{O}Y(k)$$

7.Parseval's Theorem

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

8.Transforms of even real functions:

$$x_{er}(n) \leftrightarrow X_{er}(k)$$

(the DFT of an even real sequence is even and real)

9. Transform of odd real functions:

$$x_{or}(n) \leftrightarrow jX_{oi}(k)$$

(the DFT of an odd real sequence is odd and imaginary)

10. $z(n) = x(n) + jy(n)$

$z(n) \leftrightarrow Z(k) = X(k) + jY(k)$

Example 1-1 :

$$z(n) = x(n) + jy(n) = e^{jn\pi/2}$$

$$= \cos(n\pi/2) + j\sin(n\pi/2) \qquad n = 0,1,2,3$$

Four – point DFT for *x(0), x(1), x(2), x(3)*:

X(0) = [x(0) + x(2)] + [x(1) + x(3)]
X(1) = [x(0) - x(2)] + (-j)[x(1) - x(3)]
X(2) = [x(0) + x(2)] - [x(1) + x(3)]
X(3) = [x(0) - x(2)] + j[x(1) - x(3)]

For $x(n) = \cos(n\pi/2)$ =>

$$\underbrace{x(0)=1 \quad x(1)=0 \quad x(2)=-1 \quad x(3)=0}_{\Downarrow}$$

$$X(0)=1 \quad X(1)=2 \quad X(2)=0 \quad X(3)=2$$

For $y(n) = \sin(n\pi/2)$ =>

$$\underbrace{y(0)=0 \quad y(1)=1 \quad y(2)=0 \quad y(3)=-1}_{\Downarrow}$$

$$Y(0)=1 \quad Y(1)=-j2 \quad Y(2)=0 \quad Y(3)=j2$$

$$\Rightarrow \begin{aligned} &Z(0)=0 \\ &Z(1)=2-j2 \\ &Z(2)=0 \\ &Z(3)=2+j2 \end{aligned}$$

Example 1-2

DFT of $x(n) = \delta(n)$:

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

Time-shift property

$$DFT[x(n-n_0)] = \sum_{n=0}^{N-1} \delta(n-n_0) W_N^{nk}$$

$$= W_N^{n_0 k} = e^{-j2\pi k n_0/N}$$

Example 1-3: Circular Convolution

$x_1(n) = 1 \quad x_2(n) = 1 \quad 0 \le n \le N-1$

Define $x_{3c}(n) = x_1(n) \text{O} x_2(n) = \sum_{m=0}^{N-1} x_1(m) x_2(n-m)$

$$X_1(k) = X_2(k) = \sum_{n=0}^{N-1} x_1(n) W_N{}^{nk} = \sum_{n=0}^{N-1} x_2(n) W_N{}^{nk}$$

$$= \sum_{n=0}^{N-1} W_N{}^{nk} = \begin{cases} N & k = 0 \\ 0 & k \neq 0 \end{cases}$$

$$\Rightarrow X_3(k) = X_1(k) X_2(k) = \begin{cases} N^2 & k = 0 \\ 0 & k \neq 0 \end{cases}$$

$$x_{3c}(n) = \frac{1}{N} \sum_{k=0}^{N-1} x_3(k) e^{j2\pi nk/N}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} N^2 \delta(k) e^{j2\pi nk/N}$$

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

$$\Rightarrow x_{3c}(n) = N$$

- **Symmetry:**

$$\begin{aligned} X[k] &= X^*[((-k))_N] \\ \mathrm{Re}\{X[k]\} &= \mathrm{Re}\{X[((-k))_N]\} \\ \mathrm{Im}\{X[k]\} &= -\mathrm{Im}\{X[((-k))_N]\} \\ |X[k]| &= |X[((-k))_N]| \\ \sphericalangle X[k] &= -\sphericalangle X[((-k))_N] \end{aligned}$$

- **Linearity:** $ax_1[n] + bx_2[n] \overset{\mathcal{D}}{\longleftrightarrow} aX_1[k] + bX_2[k]$.
- **Circular time shift:** $x[((n-m))_N] \overset{\mathcal{D}}{\longleftrightarrow} W_N^{km} X[k]$.
- **Circular convolution:**

$$\sum_{m=0}^{N-1} x_1[m] x_2[((n-m))_N] \overset{\mathcal{D}}{\longleftrightarrow} X_1[k] X_2[k].$$

  Circular convolution between two N-point signals is sometimes denoted by $x_1[n] \textcircled{N} x[n]$.

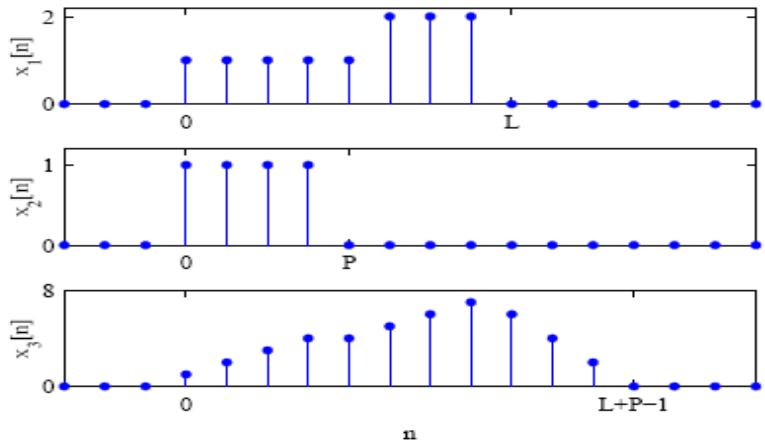
- **Modulation:**

$$x_1[n] x_2[n] \overset{\mathcal{D}}{\longleftrightarrow} \frac{1}{N} \sum_{l=0}^{N-1} X_1[l] X_2[((k-l))_N].$$

1.5 Convolution: Linear convolution of two finite-length sequences Consider a sequence x1[n] with length L points, and x2[n] with length P points. The linear convolution of the

$$x_3[n] = \sum_{m=-\infty}^{\infty} x_1[m]x_2[n-m],$$

is nonzero over a maximum length of $L + P - 1$ points:



Therefore $L + P - 1$ is the maximum length of $x_3[n]$ resulting from the linear convolution.

The N-point circular convolution of $x_1[n]$ and $x_2[n]$ is

$$x_1[n] \textcircled{N} x_2[n] = \sum_{m=0}^{N-1} x_1[m]x_2[((n-m))_N] = \sum_{m=0}^{N-1} x_1[m]\tilde{x}_2[n-m] :$$

sequences,

Therefore L + P □ 1 is the maximum length of x3[n] resulting from the Linear convolution.

1.6 Circular Convolution:

The N-point circular convolution of x1[n] and x2[n] is

$$x_1[n] \textcircled{N} x_2[n] = \sum_{m=0}^{N-1} x_1[m]x_2[((n-m))_N] = \sum_{m=0}^{N-1} x_1[m]\tilde{x}_2[n-m] :$$

It is easy to see that the circular convolution product will be equal to the linear convolution product on the interval 0 to N □ 1 as long as we choose N - L + P +1. The process of augmenting a sequence with zeros to make it of a required length is called zero padding.

1.7 Filtering methods based on DFT

1. DFT Algorithm

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

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

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

Properties of W_N^{m}:

(1) $W_N^{0} = (e^{-j2\pi/N})^0 = e^0 = 1, \qquad W_N^{N} = e^{-j2\pi} = 1$

(2) $W_N^{N+m} = W_N^{m}$

$$W_N^{N+m} = (e^{-j2\pi/N})^{N+m}$$

$$= (e^{-j2\pi/N})^{N}(e^{-j2\pi/N})^{m}$$

$$= 1\cdot(e^{-j2\pi/N})^{m} = W_N^{m}$$

(3) $W_N^{N/2} = e^{-j2\pi/(N/2)/N} = e^{-j\pi} = -1$

$W_N^{N/4} = e^{-j2\pi/(N/4)/N} = e^{-j\pi/2} = -j$

$W_N^{3N/4} = e^{-j2\pi/(3N/4)/N} = e^{-j3\pi/2} = j$

2. Examples

Example 1-5: Two-Point DFT

$x(0), x(1)$: $X(k) = \sum_{n=0}^{1} x(n)W_2^{nk}$ $\qquad k = 0,1$

$$X(0) = \sum_{n=0}^{1} x(n)W_2^{n0} = \sum_{n=0}^{1} x(n) = x(0) + x(1)$$

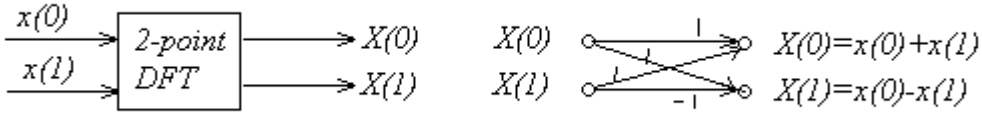
$$X(1) = \sum_{n=0}^{1} x(n)W_2^{n1} = \sum_{n=0}^{1} x(n)W_2^{n}$$

$$= x(0)W_2^{0} + x(1)W_2^{1}$$

$$= x(0) + x(1)W_2^{(1/2)2}$$

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

$$= x(0) - x(1)$$



Example 1-6: Generalization of derivation in example 2-3 to a four-point DFT

$x(0), x(1), x(2), x(3)$

$$X(k) = \sum_{n=0}^{3} x(n)W_4^{nk} \qquad k = 0,1,2,3,$$

$$X(0) = \sum_{n=0}^{3} x(n)W_4^{n0} = \sum_{n=0}^{3} x(n) = x(0) + x(1) + x(2) + x(3)$$

$$X(1) = \sum_{n=0}^{3} x(n)W_4^{n} = x(0)W_4^{0} + x(1)W_4^{1} + x(2)W_4^{2} + x(3)W_4^{3}$$

$$= x(0) - jx(1) - x(2) + jx(3)$$

$$X(2) = \sum_{n=0}^{3} x(n)W_4^{2n} = x(0)W_4^{0} + x(1)W_4^{2} + x(2)W_4^{4} + x(3)W_4^{6}$$

$$= x(0) + x(1)(-1) + x(2)(1) + x(3)W_4^{2}$$

$$= x(0) - x(1) + x(2) - x(3)$$

$$X(3) = \sum_{n=0}^{3} x(n)W_4^{3n} = x(0)W_4^{0} + x(1)W_4^{3} + x(2)W_4^{6} + x(3)W_4^{9}$$

$$= x(0) + x(1)W_4^{3} + x(2)(1)W_4^{2} + x(3)W_4^{1}$$

$$= x(0) + jx(1) + (-1)x(2) + (-j)x(3)$$

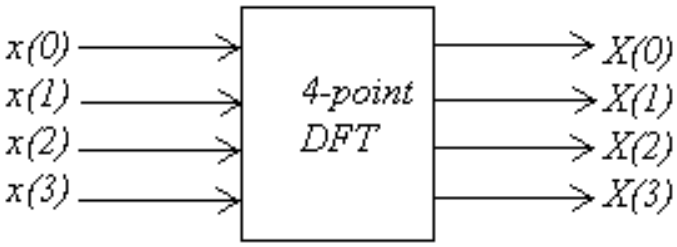
$$= x(0) + jx(1) - x(2) - jx(3)$$

$$X(0) = [x(0) + x(2)] + [x(1) + x(3)]$$

➔ $$X(1) = [x(0) - x(2)] + (-j)[x(1) - x(3)]$$

$$X(2) = [x(0) + x(2)] - [x(1) + x(3)]$$

$$X(3) = [x(0) - x(2)] + j[x(1) - x(3)]$$



Two – point DFT

If we denote $z(0) = x(0)$, $z(1) = x(2)$ => $Z(0) = z(0) + z(1) = x(0) + x(2)$

$Z(1) = z(0) - z(1) = x(0) - x(2)$

$v(0) = x(1)$, $v(1) = x(3)$ => $V(0) = v(0) + v(1) = x(1) + x(3)$

$V(1) = v(0) - v(1) = x(1) - x(3)$

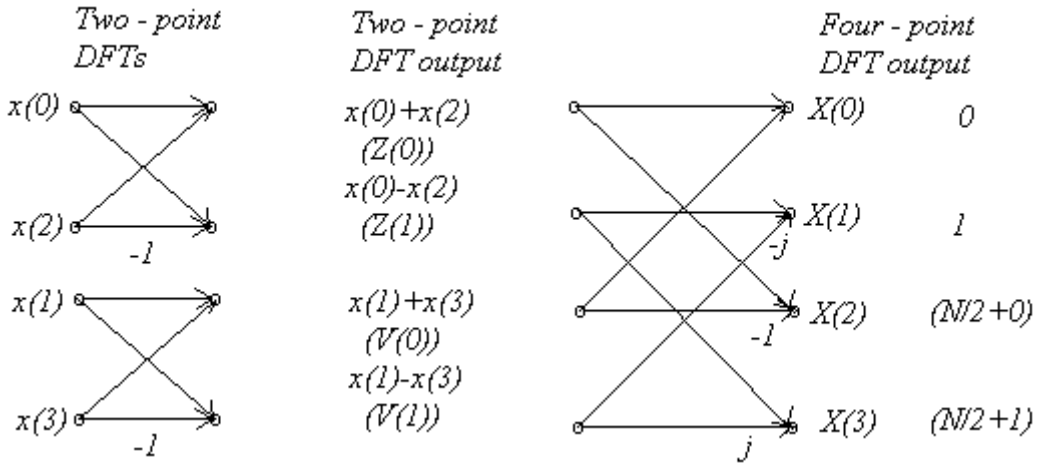
Four – point DFT **Two-point DFT**

➔ $X(0) = Z(0) + V(0)$

$X(1) = Z(1) + (-j)V(1)$

$X(2) = Z(0) - V(0)$

$X(3) = Z(1) + jV(1)$



➔ One Four – point DFT ➔Two Two – point DFT

1.8 FFT Algorithms:

Fast Fourier transforms

The widespread application of the DFT to convolution and spectrum analysis is due to the existence of fast algorithms for its implementation. The class of methods is referred to as fast Fourier transforms (FFTs). Consider a direct implementation of an 8-point DFT:

$$X[k] = \sum_{n=0}^{7} x[n] W_8^{kn}, \qquad k = 0, \ldots, 7.$$

If the factors W_8^{kn} have been calculated in advance (and perhaps stored in a lookup table), then the calculation of X[k] for each value of k requires 8 complex multiplications and 7 complex additions. The 8-point DFT therefore requires 8 * 8 multiplications and 8* 7 additions. For an N-point DFT these become N2 and N (N - 1) respectively. If N = 1024, then approximately one million complex multiplications and one million complex additions are required. The key to reducing the computational complexity lies in the observation that the same values of x[n] W_8^{kn} are effectively calculated many times as the computation proceeds | particularly if the transform is long. The conventional decomposition involves decimation-in-time, where at each stage a N-point transform is decomposed into two N=2-point transforms. That is, X[k] can be written as X[k] =N

$$\begin{aligned} X[k] &= \sum_{r=0}^{N/2-1} x[2r] W_N^{2rk} + \sum_{r=0}^{N/2-1} x[2r+1] W_N^{(2r+1)k} \\ &= \sum_{r=0}^{N/2-1} x[2r] (W_N^2)^{rk} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1] (W_N^2)^{rk}. \end{aligned}$$

Noting that $W_N^2 = W_{N/2}$ this becomes

$$\begin{aligned} X[k] &= \sum_{r=0}^{N/2-1} x[2r] (W_{N/2})^{rk} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1] (W_{N/2})^{rk} \\ &= G[k] + W_N^k H[k]. \end{aligned}$$

The original N-point DFT can therefore be expressed in terms of two N=2-point DFTs.

The N=2-point transforms can again be decomposed, and the process repeated until only 2-point transforms remain. In general this requires log2N stages of decomposition. Since each stage requires approximately N complex multiplications, the complexity of the resulting algorithm is of the order of N

log2 N. The difference between N2 and N log2 N complex multiplications can become considerable for large values of N. For example, if N = 2048 then N2=(N log2 N) _ 200. There are numerous variations of FFT algorithms, and all exploit the basic redundancy in the computation of the DFT. In almost all cases an Of the shelf implementation of the FFT will be sufficient | there is seldom any reason to implement a FFT yourself.

1.9 A Decimation-in-Time FFT Algorithm

$x(0), x(1), \ldots, x(N\text{-}1) \qquad N = 2^m$

$$\Rightarrow \begin{cases} g(0), g(1), \cdots, g(\frac{N}{2}-1) & -enen \quad \frac{N}{2} \quad point s \\ ((x(0), x(2), \cdots, x(N-2)) & (g(r) = x(2r)) \\ h(0), h(1), \cdots, h(\frac{N}{2}-1) & -odd \quad \frac{N}{2} \quad point s \\ ((x(1), x(3), \cdots, x(N-1)) & (h(r) = x(2r+1)) \end{cases}$$

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}$$

$$= \sum_{r=0}^{N/2-1} g(r) W_N^{k(2r)} + \sum_{r=0}^{N/2-1} h(r) W_N^{k(2r+1)} \qquad (k = 0,1,..., N-1)$$

$$= \sum_{r=0}^{N/2-1} g(r) W_N^{2kr} + W_N^k \sum_{r=0}^{N/2-1} h(r) W_N^{2kr}$$

$$W_N^{2kr} = (e^{-j2\pi./N})^{2kr} = (e^{-j2\pi/(.N/2)})^{kr} = W_{\frac{N}{2}}^{kr}$$

$$\Rightarrow X(k) = \sum_{r=0}^{N/2-1} g(r) W_{N/2}^{kr} + W_N^k \sum_{r=0}^{N/2-1} h(r) W_{N/2}^{kr}$$

$$= G(k) + W_N^k H(k)$$

(G(k): N/2 point DFT output (even indexed), H(k) : N/2 point DFT output (odd indexed))

$$X(k) = G(k) + W_N^k H(k) \qquad k = 0,1,..., N-1$$

$$G(k) = \sum_{r=0}^{N/2-1} g(r) W_{N/2}^{kr} = \sum_{r=0}^{N/2-1} x(2r) W_{N/2}^{kr}$$

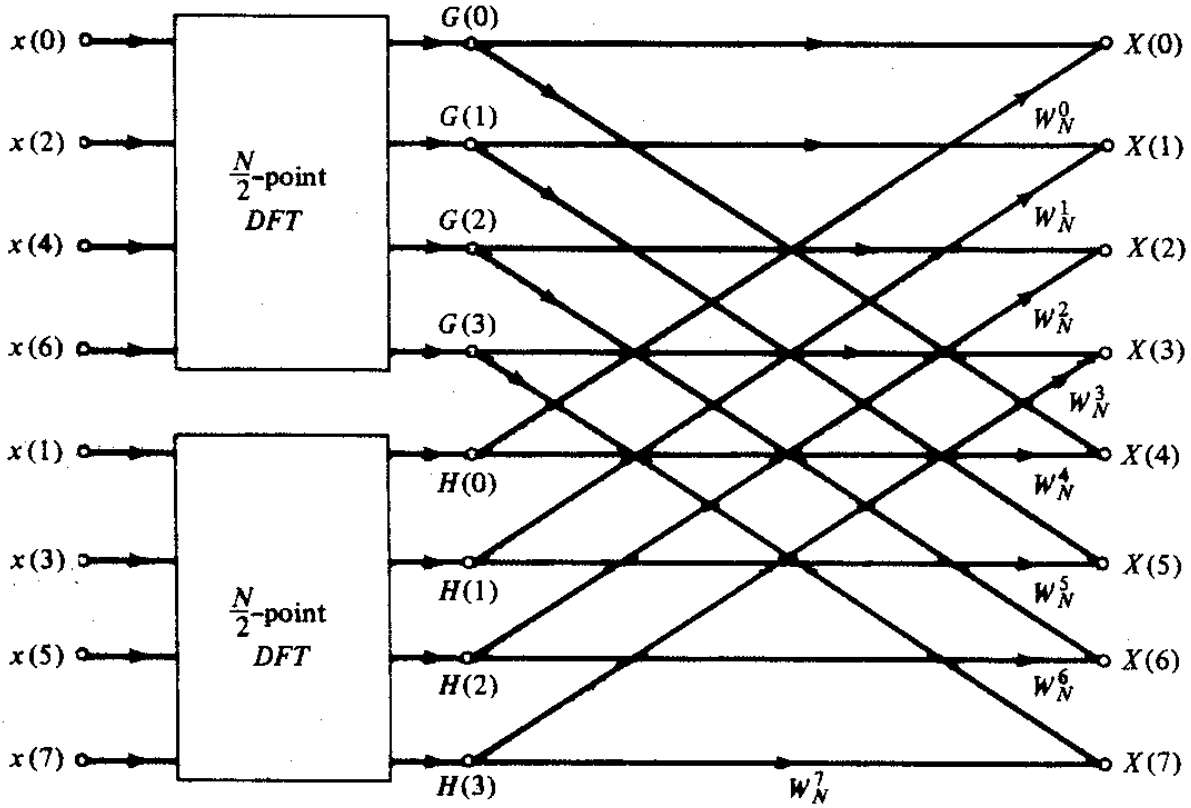
$$H(k) = \sum_{r=0}^{N/2-1} h(r) W_{N/2}^{kr} = \sum_{r=0}^{N/2-1} x(2r+1) W_{N/2}^{kr}$$

Question: $X(k)$ needs $G(k)$, $H(k)$, k=... N-1

How do we obtain $G(k)$, $H(k)$, for $k > N/2$-1 ?

$G(k) = G(N/2+k)$ $k <= N/2$-1

$H(k) = H(N/2+k)$ $k <= N/2$-1



(a) Result of one decimation of the time samples

Future Decimation

g(0), g(1), ..., g(N/2-1) $G(k)$

h(0), h(1), ..., h(N/2-1) $H(k)$

$$g(0), g(2), \cdots, g(\frac{N}{2}-2)$$

$$ge(0), ge(1), ...ge(\frac{N}{4}-1)$$

$$g(1), g(3), \cdots, g(\frac{N}{2}-1)$$

$$go(0), go(1), ...go(\frac{N}{4}-1)$$

$$G(k) = \sum_{r=0}^{N/2-1} g(r) W_{(N/2)}^{kr}$$

$$= \sum_{m=0}^{N/4-1} ge(m) W_{(N/4)}^{km} + W_{(N/2)}^{k} \sum_{m=0}^{N/4-1} go(m) W_{(N/4)}^{km}$$

$$= GE(k) + W_{(N/2)}^{k} Go(k)$$

even indexed g (N/4 point) odd indexed g (N/4 point)

$$W_{N/2}^{k} = W_N^{2k} \; ?$$

$$W_{N/2}^{k} = (e^{-j2\pi/(N/2)})^k$$

$$= (e^{-j2\pi 2/N})^k = (e^{-j2\pi/N})^{2k}$$

$$= W_N^{2k}$$

$$\Rightarrow G(k) = GE(k) + W_N^{2k} Go(k)$$

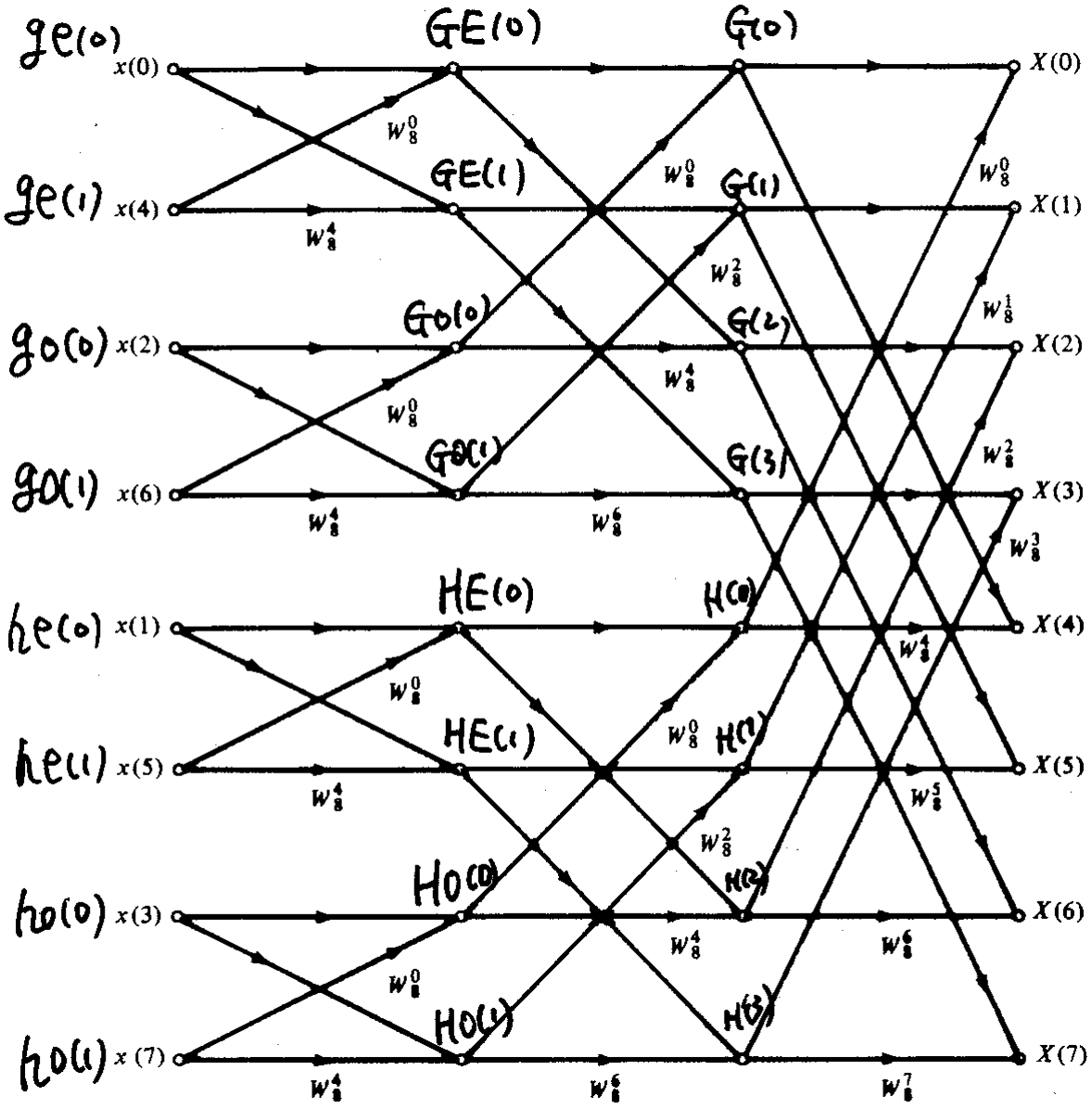
Similarly,

$$H(k) = HE(k) + W_N^{2k} Ho(k)$$

even indexed h (N/4 point) odd indexed h (N/4 point)

For 8 – point

$g(0)$	$g(1)$	$g(2)$	$g(3)$	$h(0)$	$h(1)$	$h(2)$	$h(3)$
↑	↑	↑	↑	↑	↑	↑	↑
$x(0)$	$x(2)$	$x(4)$	$x(6)$	$x(1)$	$x(3)$	$x(5)$	$x(7)$
↓		↓		↓		↓	
$ge(0)$	↓	$ge(1)$	↓	$he(0)$	↓	$he(1)$	↓
	$go(0)$		$go(1)$		$ho(0)$		$ho(1)$



1.10 Decimation-in-Frequency FFT Algorithm

$x(0), x(1), \ldots, x(N\text{-}1)$ $\qquad N = 2^m$

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

$$= \sum_{n=0}^{N/2-1} x(n) W_N^{nk} + \sum_{n=N/2}^{N-1} x(n) W_N^{nk}$$

let $m = n\text{-}N/2$ $\quad (n = N/2+m)$ $\qquad n = N/2 \Rightarrow m = N/2\text{-}N/2 = 0$

$n = N\text{-}1 \Rightarrow m = N\text{-}1\text{-}N/2 = N/2\text{-}1$

$$\Rightarrow X(k) = \sum_{n=0}^{N/2-1} x(n) W_N^{nk} + \sum_{m=0}^{N/2-1} x(N/2+m) W_N^{(N/2+m)k}$$

$$= \sum_{n=0}^{N/2-1} x(n) W_N^{nk} + \sum_{m=0}^{N/2-1} x(N/2+m) W_N^{mk} W_N^{\frac{N}{2}k}$$

$$W_N^{\frac{N}{2}} = -1 \Rightarrow W_N^{\frac{N}{2}k} = (-1)^k$$

$$X(k) = \sum_{n=0}^{N/2-1} x(n) W_N^{nk} + \sum_{m=0}^{N/2-1} (-1)^k x(N/2+m) W_N^{mk}$$

$$= \sum_{n=0}^{N/2-1} [x(n) + (-1)^k x(N/2+n)] W_N^{nk}$$

$$k : even \ (k = 2r) \Rightarrow X(k) = X(2r) = \sum_{n=0}^{N/2-1} [x(n) + x(N/2+n)] W_N^{2rn}$$

$$W_N^{2rn} = (e^{-j2\pi/N})^{2rn} = (e^{-j2\pi/(N/2)})^{rn} = W_{N/2}^{rn}$$

$$\Rightarrow X(k) = X(2r) = \sum_{n=0}^{N/2-1} [\underbrace{x(n) + x(N/2+n)}_{y(n)}] W_{N/2}^{rn}$$

N/2 point DFT

$$Y(r) = \sum_{n=0}^{N/2-1} y(n) W_{N/2}^{rn} \rightarrow Z(r)$$

$$k : odd \Rightarrow k = 2r+1$$

$$\Rightarrow X(k) = X(2r+1)$$

$$= \sum_{n=0}^{N/2-1} [x(n) - x(N/2+n)] W_N^{n(2r+1)}$$

$$= \sum_{n=0}^{N/2-1} \underbrace{[x(n) - x(N/2+n)] W_N^{n}}_{z(n)} W_N^{2rn}$$

$$= \sum_{n=0}^{N/2-1} z(n) W_N^{2rn}$$

$$= \sum_{n=0}^{N/2-1} z(n) W_{N/2}^{rn}$$

$$Z(r) = \sum_{n=0}^{N/2-1} z(n) W_{N/2}^{rn} \leftarrow \frac{N}{2} \text{ point DFT of } z(0), \cdots, z(\frac{N}{2}-1)$$

$X(k)$: N-point DFT of $x(0)$, …, $x(N)$ ➔ two $N/2$ point DFT

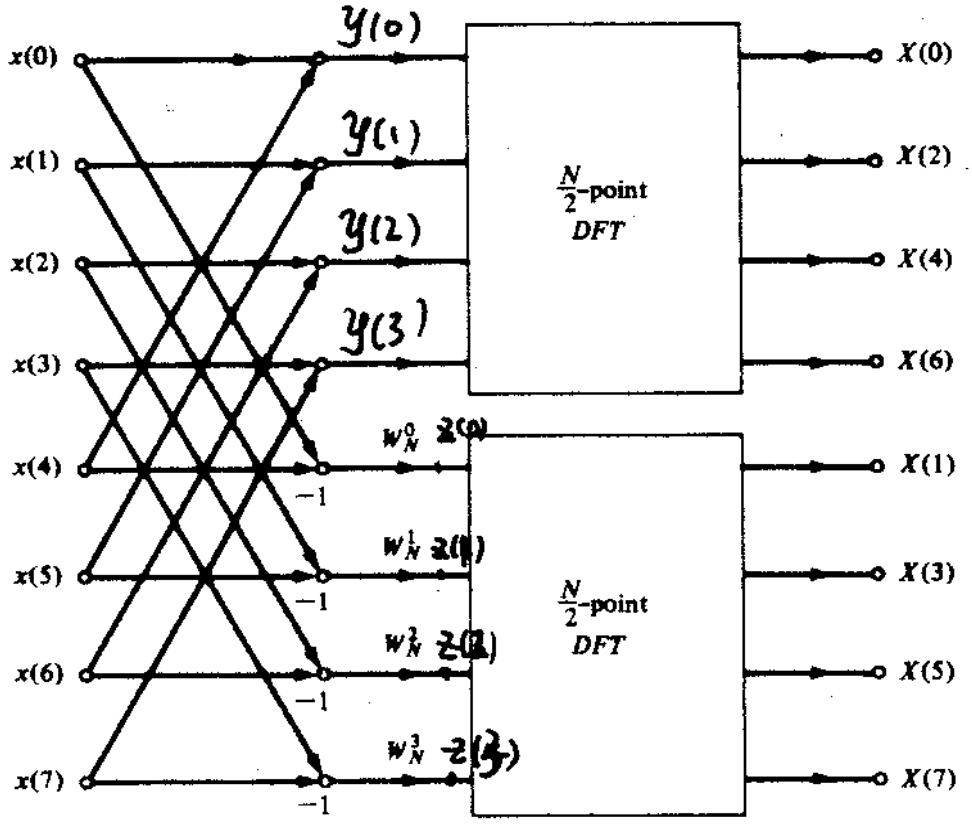


FIGURE 10-10. Flow graph after a single decimation.

One N/2 point DFT => two N/4 point DFT

… two point DFTs

Consider N/2 point DFT

y(0), y(1), …, y(N/2-1)

$$Y(k) = \sum_{n=0}^{N/2-1} y(n) W_{N/2}^{rn}$$

$$= \sum_{n=0}^{N/4-1} [y(n) + (-1)^k y(N/4+n)] W_{N/2}^{nk}$$

$$k : even \Rightarrow k = 2r$$

$$Y(k) = Y(2r) = \sum_{n=0}^{N/4-1} \underbrace{[y(n) + y(N/4+n)]}_{y1(n)} W_{N/2}^{nk}$$

$$Y1(r) = \sum_{n=0}^{N/4-1} y1(n) W_{N/4}^{nk} \leftarrow N/4 \quad point \quad DFt$$

$$k : odd \Rightarrow k = 2r+1$$

$$Y(k) = Y(2r+1) = \sum_{n=0}^{N/4-1} \underbrace{[y(n) - x(N/4+n)]W_{N/2}{}^{n}}_{y2(n)} W_{N/2}{}^{2rn}$$

$$Y2(r) = \sum_{n=0}^{N/4-1} y2(n) W_{N/4}{}^{rn} \leftarrow N/4 \quad point \quad DFT$$

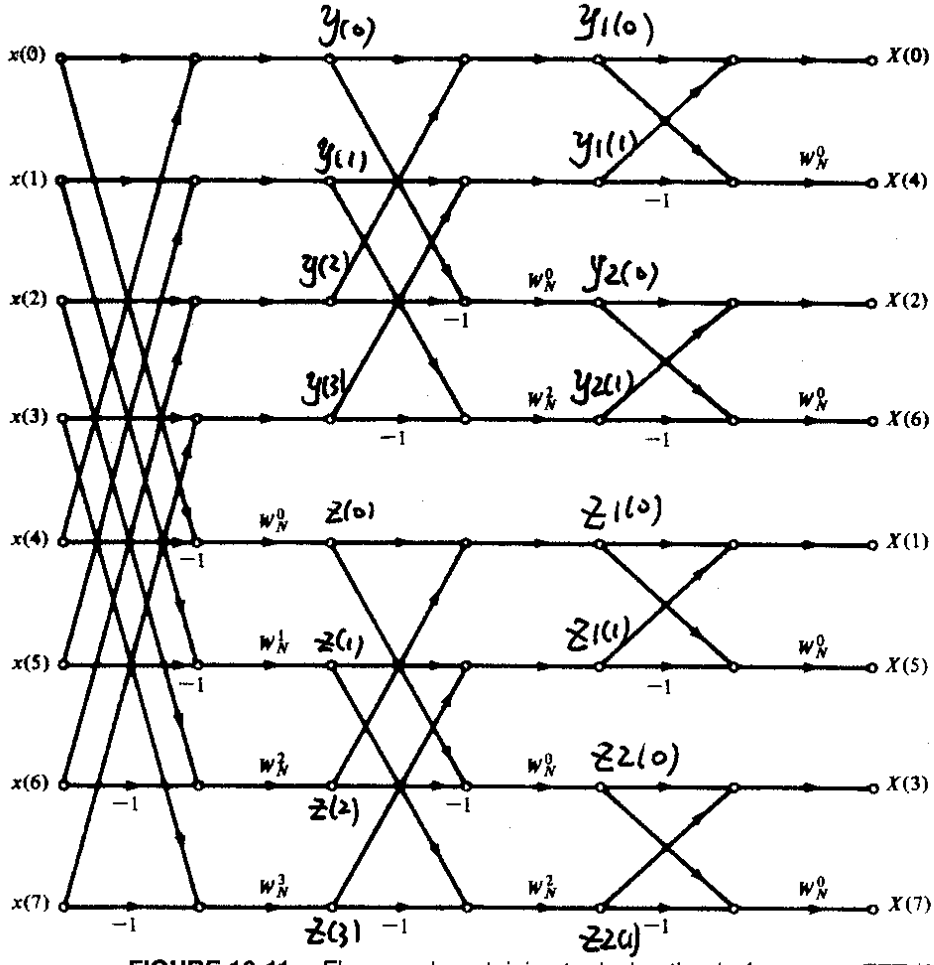


FIGURE 10-11. Flow graph pertaining to decimation-in-frequency FFT ($N = 8$).

Computation

N – point DFT : $4N(N-1)$ real multiplications
$4N(N-1)$ real additions

N – point FFT : $2N\log_2 N$ real multiplications

(N = 2^m) ***$3N\log_2 N$ real additions***

Computation ration

$$\frac{FFT's \quad computations}{DFT's \quad computations} = \frac{5\log_2 N}{8(N-1)}$$

$$N=2^{12}=4096 \Rightarrow \frac{5\times 12}{8\times 4095} = 0.18\%$$

TABLE 10-2
Symmetry Properties of W_N^m for $N = 8$

$m =$	W_8^m
0	1
1	$\frac{1-j}{\sqrt{2}}$
2	$-j$
3	$-\frac{1+j}{\sqrt{2}}$
4	$-1 = -W^0$
5	$-\frac{1-j}{\sqrt{2}} = -W^1$
6	$j = -W^2$
7	$\frac{1+j}{\sqrt{2}} = -W^3$

TABLE 10-3
Comparison of the Number of Real Multiplications for the DFT and FFT

Number of Points	DFT	FFT	Speed Factor
2	8	4	2
4	48	16	3
8	224	48	5
16	960	128	8
32	3,968	320	12
64	16,128	768	21
128	65,024	1,792	36
256	261,120	4,096	64
512	1,046,528	9,216	114
1,024	4,190,208	20,480	205
2,048	16,769,024	45,056	372
4,096	67,092,480	98,304	683

1.11Applications of FFT

Use of FFT in linear filtering.

1. Filtering

   x(0), …, x(N-1) FFT (DFT) =>
   X(0), … , X(1), … , X(N-1)

   X(k): Line spectrum at $\omega_k = \frac{2\pi k\Delta t}{T} = \frac{2\pi k}{N}$ $\quad (\omega_1 = \frac{2\pi}{T} \quad \Delta t = \frac{T}{N})$

   (Over T: x(0), …, x(N-1) are sampled.)

   Inverse DFT:

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

$$\hat{x}(n) = \sum_{k=0}^{N_0} X(k) e^{j2\pi nk/N}$$

   Frequencies with $\omega > \frac{2\pi N_0}{N}$ have been filtered!

   Example 1-10

$$x(n) = \cos \underbrace{\frac{\pi}{4} n}_{\omega_1 = \frac{2\pi}{8} (\frac{2\pi}{N})} + \cos \underbrace{\frac{\pi}{2} n}_{\omega_2 = (\frac{2\pi}{N}) 2 = \frac{4\pi}{8} = \frac{\pi}{2}} \qquad 0 \le n \le 7$$

x(0), x(1), …, x(7)

$$\Rightarrow \quad \underset{\uparrow}{X(0)}, \quad \underset{\uparrow}{X(1)}, \quad \underset{\uparrow}{X(2)}, \quad \cdots \quad \underset{\uparrow}{X(7)}$$

$$0 \quad non-zero \quad non-zero \quad 0 \quad 0$$

How to filter frequency higher than $\frac{\pi}{4}$?

2. Spectrum Analyzers
   Analog oscilloscopes => time-domain display
   Spectrum Analyzers: Data Storage, FFT

3. Energy Spectral Density
   x(0), …, x(N-1): its energy definition

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

   Parseval's Theorem

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

UNIT-II
IIR Filter design

2-1 Structures of IIR

1. Direct-Form Realization

$$H(z)=\frac{Y(z)}{X(z)}=\frac{\sum_{i=0}^{r}L_i z^{-i}}{1+\sum_{j=1}^{m}k_j z^{-j}}$$

$$\Rightarrow y(nT)=\sum_{i=0}^{r}L_i x(nT-iT)-\sum_{j=1}^{m}k_j y(nT-jT)$$

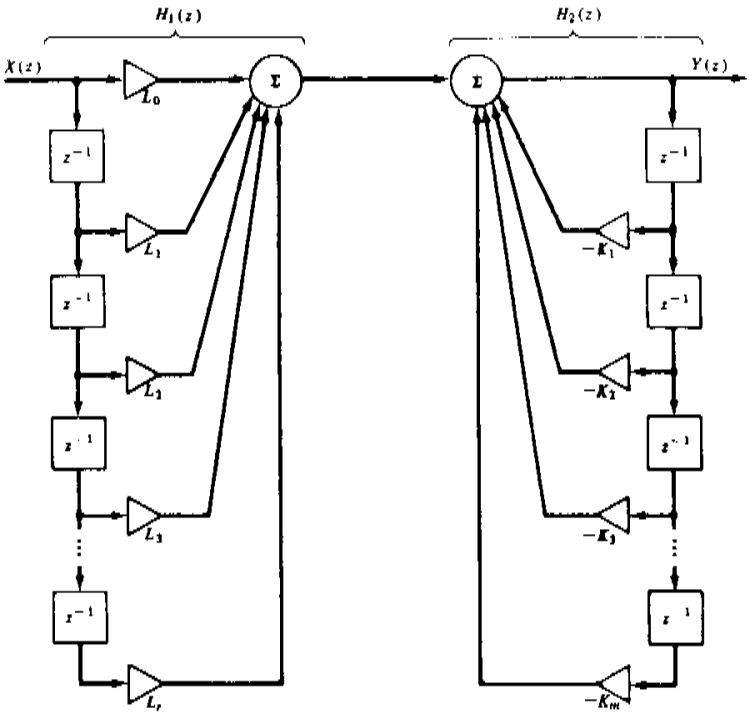


FIGURE 9-1. Direct Form I realization.

The function is realized!
What's the issue here?

Count how many memory elements we need!

Can we reduce this number?
If we can, what is the concern?

$$H(z)=\frac{\sum_{i=0}^{r}L_i z^{-i}}{1+\sum_{j=1}^{m}k_j z^{-i}}=\underbrace{\left(\sum_{i=0}^{r}L_i z^{-i}\right)}_{H_1(z)}\underbrace{\frac{1}{1+\sum_{j=1}^{m}k_j z^{-j}}}_{H_2(z)}$$

$$\Rightarrow Y(z)=H(z)X(z)=H_1(z)H_2(z)X(z)$$

Denote $V(z) = H_2(z)X(z) \Rightarrow Y(z) = H_1(z)V(z)$

Implement $H_2(z)$ and then $H_1(z)$?

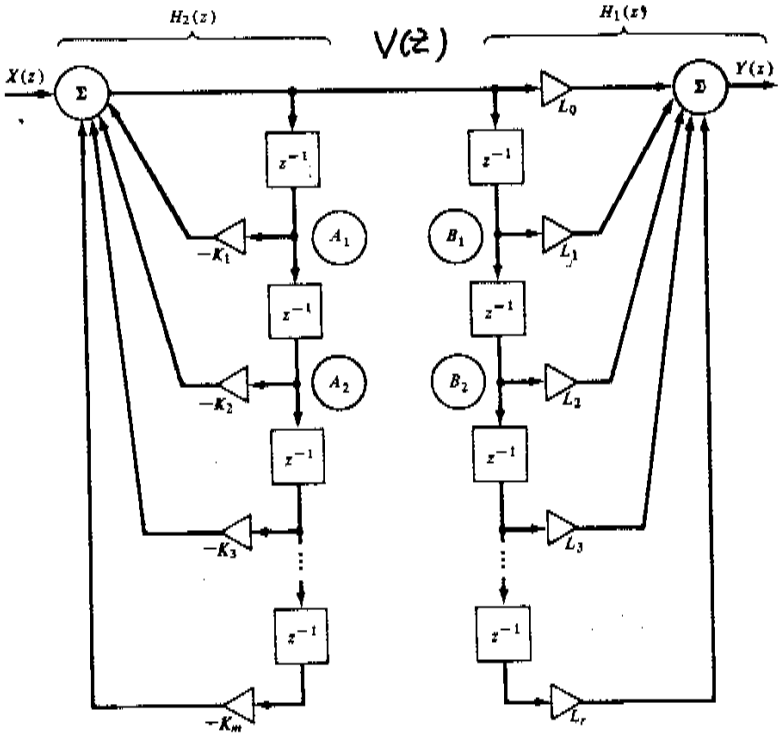


FIGURE 9-2. Rearrangement of Figure 9-1.

Why H_2 is implemented?

(1)

$$V(z) = X(z) - k_1 z^{-1}V(z) - \cdots - k_m z^{-m}V(z)$$

(2)

$$(1 + k_1 z^{-1} + \cdots + k_m z^{-m})V(z) = X(z)$$

$$V(z) = \frac{1}{1 + \sum_{j=1}^{m} k_j z^{-j}} X(z)$$

H_2 is realized!

Can you tell why H_1 is realized?

What can we see from this realization? Signals at A_j and B_j: always the same

➔ Direct Form II Realization

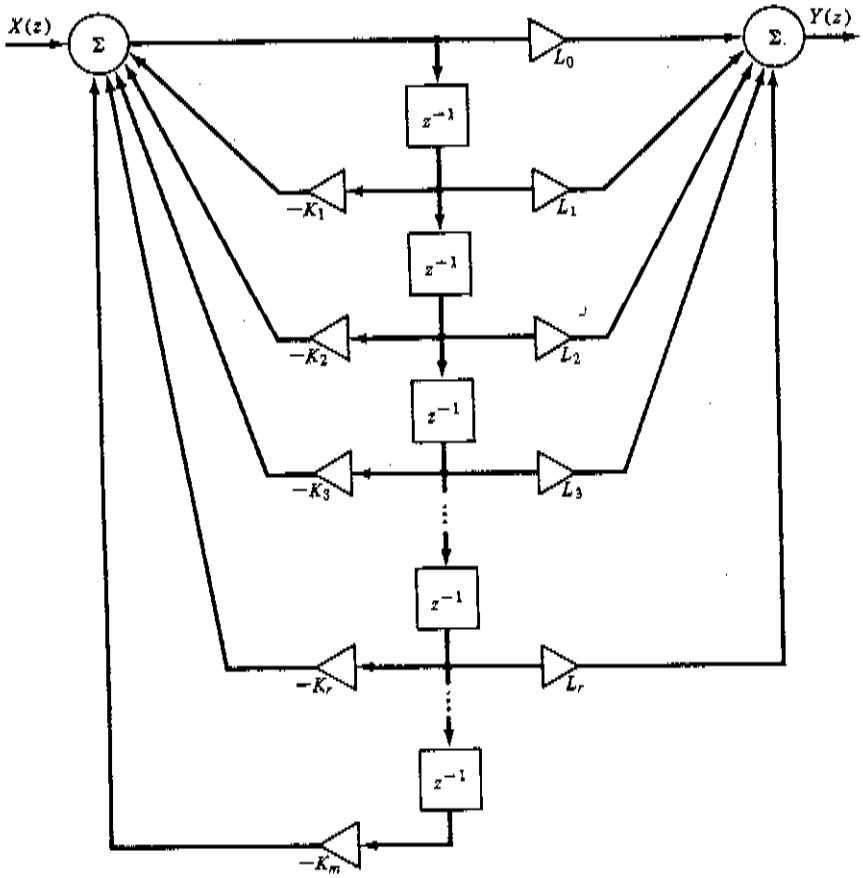
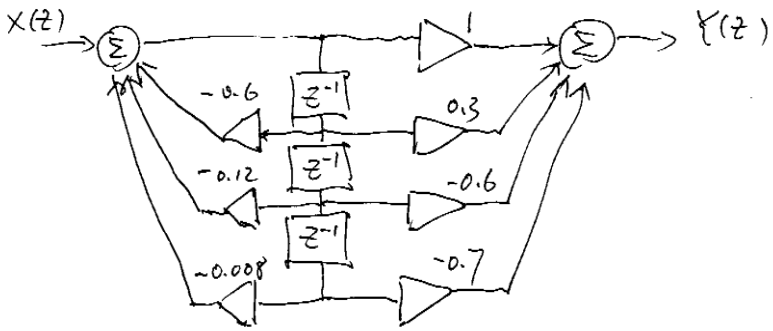


FIGURE 9-3. Direct Form II realization.

Example 2.1 $H(z)=\dfrac{1+0.3z^{-1}-0.6z^{-2}-0.7z^{-3}}{(1+0.2z^{-1})^3}$

Solution: $H(z)=\dfrac{1+0.3z^{-1}-0.6z^{-2}-0.7z^{-3}}{1+0.6z^{-1}+0.12z^{-2}+0.008z^{-3}}$

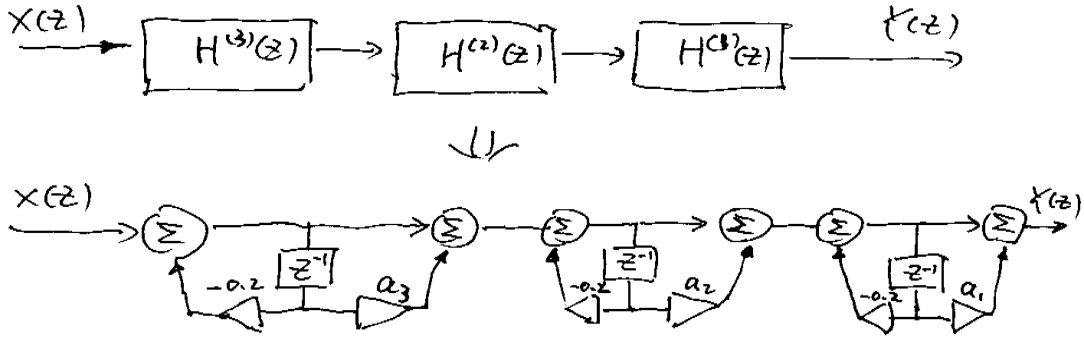


Important: H(z)=B(z)/A(z) (1) A: 1+…. (2) Coefficients in A: in the feedback channel

2. Cascade Realization

Factorize $1+0.3z^{-1}-0.6z^{-2}-0.7z^{-3}=(1-a_1z^{-1})(1-a_2z^{-2})(1-a_3z^{-3})$

$$H(z)=\frac{1+0.3z^{-1}-0.6z^{-2}-0.7z^{-3}}{(1+0.2z^{-1})^3}=\underbrace{\frac{1-a_1z^{-1}}{1+0.2z^{-1}}}_{H^{(1)}(z)}\cdot\underbrace{\frac{1-a_2z^{-1}}{1+0.2z^{-1}}}_{H^{(2)}(z)}\cdot\underbrace{\frac{1-a_3z^{-1}}{1+0.2z^{-1}}}_{H^{(3)}(z)}$$



General Form

$$H(z)=kz^{-M}\frac{\prod_{i=1}^{N_1}(1-a_iz^{-1})\prod_{j=1}^{N_2}(1-b_j^{*}z^{-1})(1+b_j^{*}z^{-1})}{\underbrace{\prod_{k=1}^{D_1}(1-c_kz^{-1})}_{\Downarrow}\underbrace{\prod_{l=1}^{D_2}(1-d_l^{*}z^{-1})(1+d_l^{*}z^{-1})}_{\Downarrow}}$$

$$\frac{1-q_iz^{-1}}{1-c_kz^{-1}} \qquad \frac{1-(b_j+b_j^{*})z^{-1}+b_jb_j^{*}z^{-2}}{1-(d_l+d_l^{*})z^{-1}+d_ld_l^{*}z^{-2}}$$

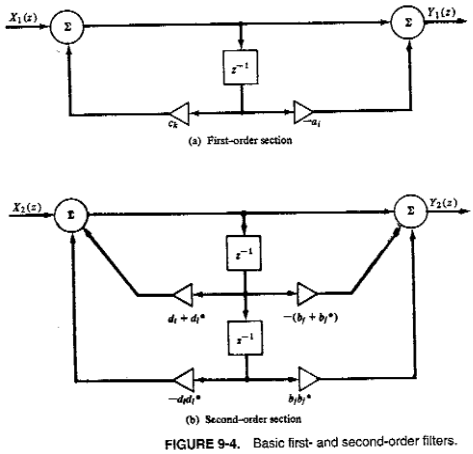


FIGURE 9-4. Basic first- and second-order filters.

Apply Direct II for each!

3. Parallel Realization (Simple Poles)

$$H(z)=\underbrace{\sum_{i=0}^{M}A_iz^{-i}}_{\text{If } r>m}+\underbrace{\sum_{k=1}^{D_1}B_k\frac{1}{1-C_kz^{-1}}}_{\text{realize the real poles}}+\underbrace{\sum_{l=1}^{D_2}C_l\frac{1-e_lz^{-1}}{(1-d_lz^{-1})(1-d_l^{*}z^{-1})}}_{\text{realize the complex congugate poles}}$$

Example 2-1

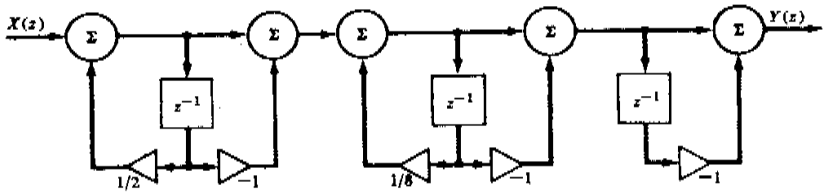
$$H(z)=\frac{(1-z^{-1})^3}{(1-\frac{1}{2}z^{-1})(1-\frac{1}{8}z^{-1})}$$

cascade and parallel realization!

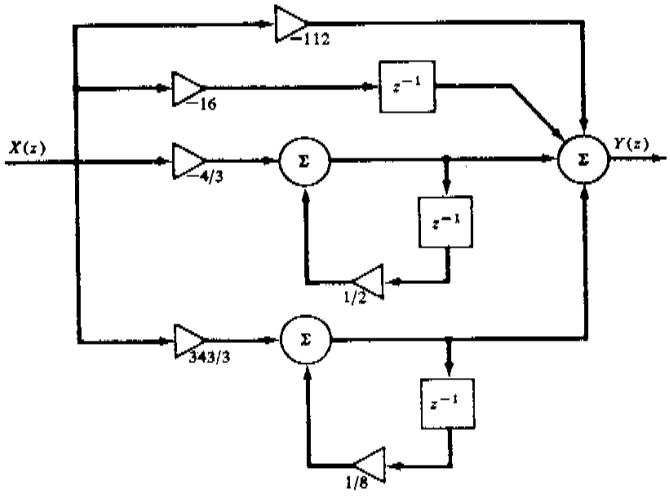
Solution:

(1) Cascade:

$$H(z)=\frac{1-z^{-1}}{(1-\frac{1}{2}z^{-1})}\frac{1-z^{-1}}{(1-\frac{1}{8}z^{-1})}(1-z^{-1})$$



(a) Cascade realization



(b) Parallel realization

FIGURE 9-5. Cascade and parallel realizations for Example 9-1.

(2) Parallel

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

In order to make deg(num)<deg(den)

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

Partial-Fraction Expansion for *s*

$$=\frac{A}{z^2}+\frac{B}{z}+\frac{C}{z-1/2}+\frac{D}{z-1/8}$$

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

$$C=\lim_{z\to 1/2}(z-1/2)\frac{H(z)}{z}=\lim_{z\to 1/2}\frac{(z-1)^3}{z^2(z-1/8)}=-\frac{4}{3}$$

$$D = \lim_{z\to 1/8}(z-1/8)\frac{H(z)}{z} = \lim_{z\to 1/8}\frac{(z-1)^3}{z^2(z-1/2)} = \frac{343}{3}$$

z = anything other than *0, ½, 1/8, 1* ➔ B = -112

For example, z=2

$$\lim_{z\to 0} H(z) = \lim_{z\to 0}(\frac{A}{z}+B)$$

$$\lim_{z\to 0} zH(z) = \lim_{z\to 0}(A+Bz)$$

$$\lim_{z\to 0}\frac{d(H(z)z)}{dz} = B$$

$$\Rightarrow B = \lim_{z\to 0}\frac{d}{dz}\left[\frac{(z-1)^3}{(z-\frac{1}{2})(z-\frac{1}{8})}\right]$$

$$= -112$$

$$\frac{H(z)}{z} = \frac{1^3}{4\times\frac{3}{2}\times\frac{15}{8}} = \frac{4}{45}$$

$$\frac{A}{z^2} = \frac{-16}{4} = -4$$

$$\frac{C}{z-\frac{1}{2}} = \frac{-4/3}{3/2} = -8/9$$

$$\frac{D}{z-\frac{1}{8}} = \frac{-343/3}{15/8} = \frac{343}{3}\frac{8}{15}$$

$$B = 2[\frac{4}{45}+4+\frac{8}{9}-\frac{343}{3}\times\frac{8}{15}] = -112$$

Example 2-2: System having a complex conjugate pole pair at $z = ae^{\pm j\theta}$

➔Transfer function

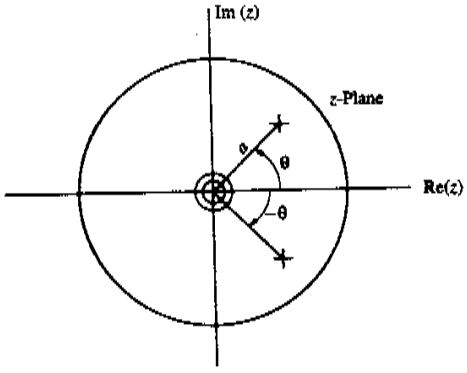
$$H(z) = \frac{z^2}{(z-ae^{+j\theta})(z-ae^{-j\theta})} = \frac{1}{(1-ae^{j\theta}z^{-1})(1-ae^{-j\theta}z^{-1})}$$

$$= \frac{z^2}{z^2-2a(\cos\theta)z+a^2} = \frac{1}{1-2a(\cos\theta)z^{-1}+a^2z^{-2}}$$

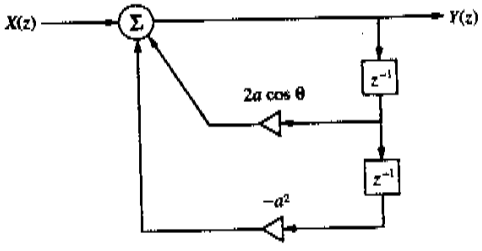
$$H(e^{j2\pi r}) = \frac{1}{1-2a(\cos\theta)e^{-j2\pi r}+a^2e^{-j4\pi r}}$$

How do we calculate the amplitude response $|H(e^{j2\pi r})|$ and $\angle H(e^{j2\pi r})$?

How the distance between the pole and the unit circle influence $|H|$ and $\angle H$?



(a) Pole-zero plot for transfer function



(b) Direct form implementation

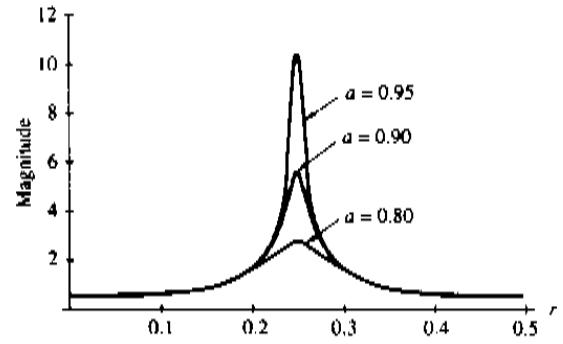
FIGURE 9-6. Pole-zero plot and implementation for Example 9-2.

How the distance between the pole and the unit circle influence $|H|$?

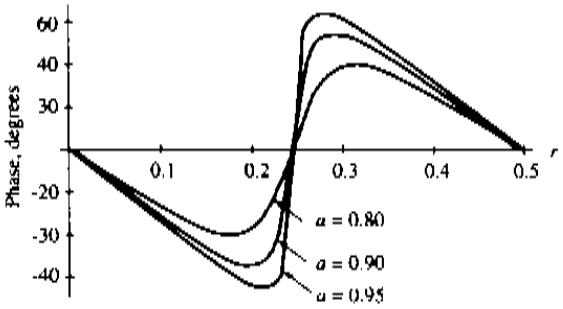
$1-a$

How the pole angle θ influence $|H|$ and $\angle H$?

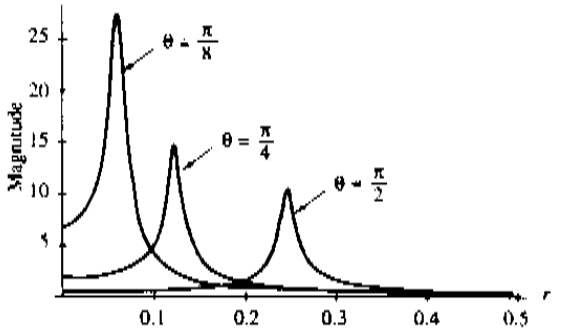
See Fig. 9-7



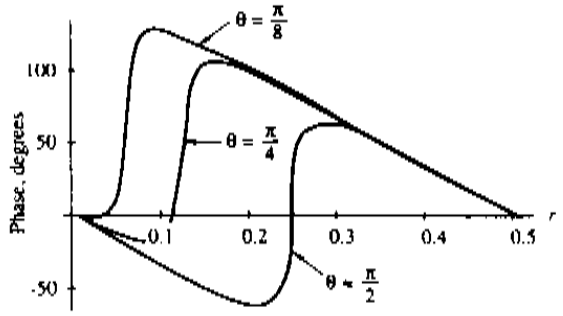
(a) Magnitude response for $\theta = \frac{\pi}{2}$ and varying a



(b) Phase response for $\theta = \frac{\pi}{2}$ and varying a



(c) Magnitude response for $a = 0.95$ and varying θ



(d) Phase response for $a = 0.95$ and varying θ

FIGURE 9-7. Magnitude and phase response for second-order filter.

2-3 Discrete-Time Integration

A method of Discrete-time system Design: Approximate continuous-time system

Integrator $y(t) = \int_0^t x(\tau)d\tau$ ← a simple system

↑ Output ↑ system input

Discrete-time approximation of this system: discrete-time Integrator

1. Rectangular Integration

$$y(t) = \int_0^t x(\tau)d\tau = \int_0^{t_0^-} x(\tau)d\tau + \int_{t_0}^t x(\tau)d\tau = y(t_0) + \int_{t_0}^t x(\tau)d\tau$$

change of y from t_0 to t : $\Delta y(t_0, t)$

$t = nT$, $t_0 = nT - T$

$$\Rightarrow y(nT) = y(nT - T) + \int_{nT-T}^{nT} x(\tau)d\tau$$

$$= y(nT - T) + \Delta y(nT - T, nT)$$

Constant $(\tau \in [nT - T, nT])$

T small enough => $x(\tau) \approx x(nT - T)$

$$\Rightarrow \int_{nT-T}^{nT} x(\tau)d\tau \approx \int_{nT-T}^{nT} x(nT - T)d\tau$$

Constant

$$= x(nT - T)\int_{nT-T}^{nT} d\tau$$

$$= Tx(nT - T)$$

$y(nT)$: System output

$x(nT)$: System input , to be integrated

$$y(nT) = y(nT - T) + Tx(nT - T)$$

A discrete-time integrator: rectangular integrator

$$\Rightarrow Y(z) = z^{-1}Y(z) + z^{-1}TX(z)$$

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

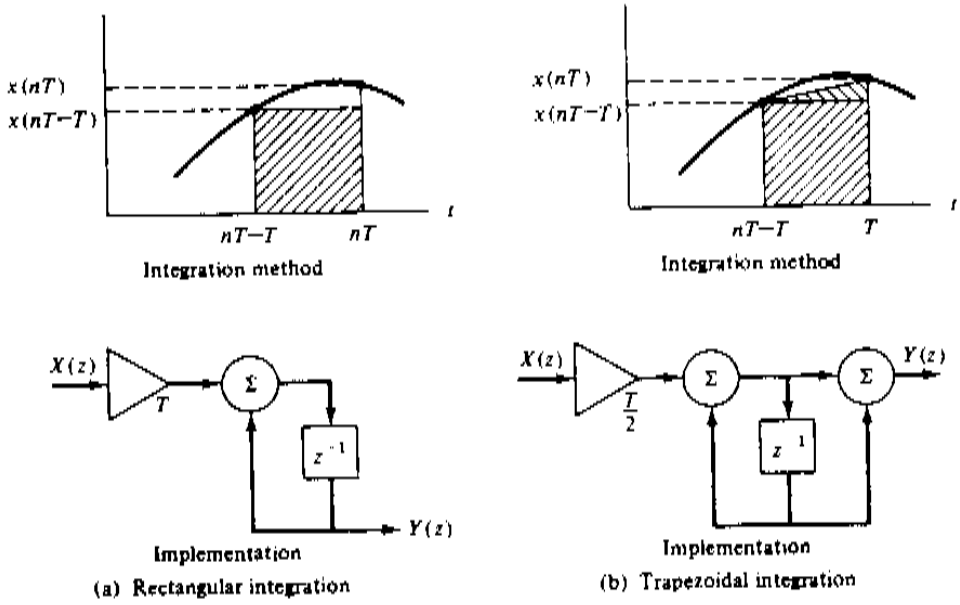


FIGURE 9-8. Rectangular and trapezoidal integration.

2. Trapezoidal Integration

Constants

$$\int_{nT-T}^{nT} x(\tau)d\tau = \int_{nT-T}^{nT} \frac{x(nT-T)+x(nT)}{2} d\tau$$

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

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

or $$y(nT) = y(nT-T) + Tx(nT-T) + \underbrace{\frac{T}{2}[x(nT)-x(nT-T)]}_{\substack{difference\\ between\\ two\\ dicrete\text{–}time\\ integrators}}$$

$$\Rightarrow Y(z) = z^{-1}Y(z) + \frac{T}{2}X(z) + \frac{T}{2}z^{-1}X(z)$$

$$H(z) = \frac{T}{2}\frac{1+z^{-1}}{1-z^{-1}}$$

3. Frequency Characteristics

2.4 Rectangular Integrator

$$H_r(z) = \frac{z^{-1}T}{1-z^{-1}}$$

Frequency Response $$H_r(e^{j\omega T}) = \frac{Te^{-j\omega T}}{1-e^{-j\omega T}} = \frac{Te^{-j\omega T/2}}{e^{j\omega T/2}-e^{-j\omega T/2}} = \frac{Te^{-j\omega T/2}}{2j\sin\omega T/2}$$

Or $H_r(e^{j2\pi r}) = \dfrac{Te^{-j\pi r}}{2j\sin\pi r}$ $\left(\begin{array}{l}\omega_s = 2\pi\frac{1}{T}\\ \omega/\omega_s = r\\ \Rightarrow \omega T = 2\pi r\end{array}\right)$

Amplitude Response

$$A_r(r) = \frac{T}{2\sin\pi r} \qquad 0 \le r \le \frac{1}{2} \quad (\sin\pi r \ge 0)$$

Phase Response

$$\Phi_r(r) = \angle e^{-j\pi r} - \angle j = -\pi r - \frac{\pi}{2} \qquad 0 \le r \le \frac{1}{2}$$

Trapezoidal Integrator

$$H_t(z) = \frac{T}{2}\frac{1+z^{-1}}{1-z^{-1}}$$

Frequency Response

$$H_t(e^{j2\pi r}) = \frac{T}{2}\frac{1+e^{-j2\pi r}}{1-e^{-j2\pi r}} = \frac{T}{2}\frac{e^{j\pi r}+e^{-j\pi r}}{e^{j\pi r}-e^{-j\pi r}}$$

$$= \frac{T}{2}\frac{2\cos\pi r}{2j\sin\pi r} = \frac{T}{2}\frac{\cos\pi r}{j\sin\pi r}$$

Amplitude: $A_t(r) = \dfrac{T\cos\pi r}{2\sin\pi r}$ $\qquad 0 \le r \le \frac{1}{2}$

Phase:

$$\Phi_t(r) = -\frac{\pi}{2} \qquad 0 \le r \le \frac{1}{2} \quad \left(\Rightarrow \begin{array}{l}\sin\pi r > 0\\ \cos\pi r \ge 0\end{array}\right)$$

2.4.2 Versus Ideal Integrator

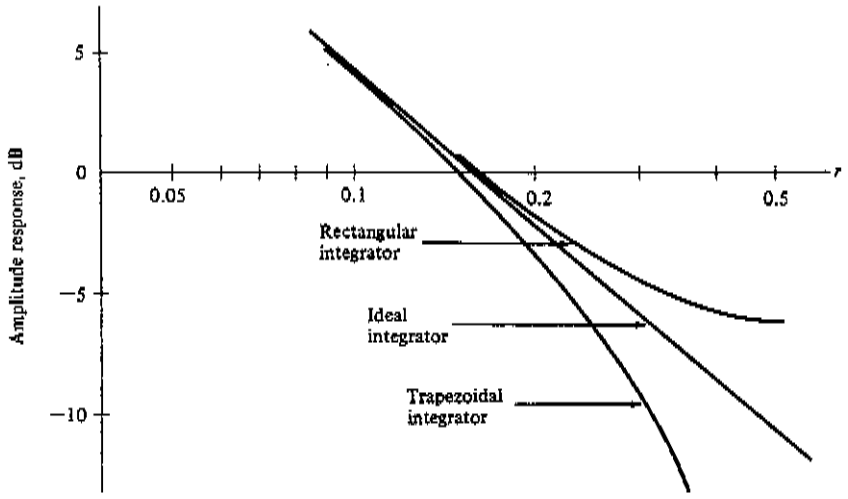
Ideal (continuous-time) Integrator

$$H(j\omega) = \frac{1}{j\omega} \qquad \omega = 2\pi f = 2\pi r f_s$$

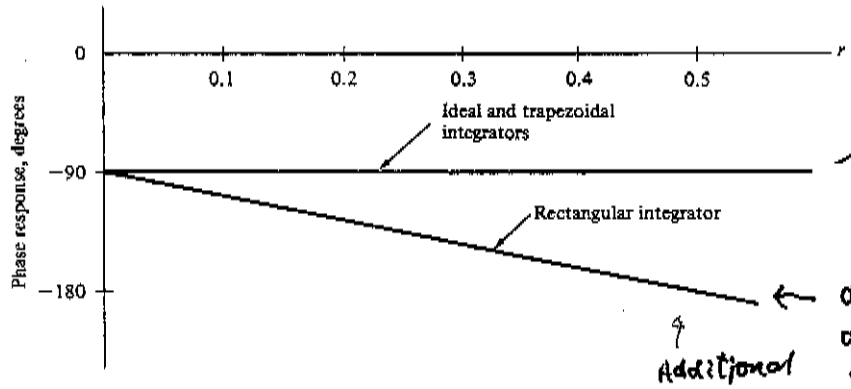
$$H(r) = \frac{1}{j2\pi r f_s}$$

$$A(r) = \frac{1}{2\pi r f_s} \qquad \Phi(r) = -\frac{\pi}{2}$$

when T=1 second (Different plots and relationships will result if T is different.)



(a) Amplitude response of rectangular and trapezoidal integrators



(b) Phase response of rectangular and trapezoidal integrators

delay the input by $\frac{2\pi}{4}$ ($\frac{T}{4}$) (should be)

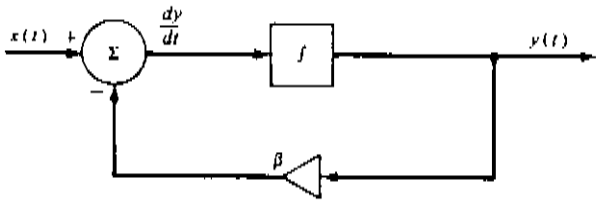
delay increases as the frequency of input increases

Additional delay up to 90°

FIGURE 9-9. Amplitude and phase responses of rectangular and trapezoidal integrators.

- Low Frequency Range $\omega T << 1$ $(2\pi r << 1)$

(Frequency of the input is much lower than the sampling frequency: It should be!)

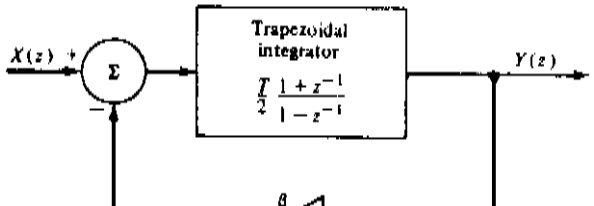


(a) Analog system

$$\frac{\cos \pi r}{\sin \pi r} \approx \frac{1}{\pi r} \Rightarrow A_t(r) \approx \frac{T}{2}\frac{1}{\pi r} = \frac{1}{2\pi r f_s}$$

$$\frac{1}{\sin \pi r} \approx \frac{1}{\pi r} \Rightarrow A_r(r) \approx \frac{T}{2\pi r} = \frac{1}{2\pi r f_s}$$

- High Frequency: Large error (should be)



Example 2-4 Differential equation (system)

$$\frac{dy}{dt} = x(t) - \beta y(t)$$

Determine a digital equivalent.

Solution

(1) Block Diagram of the original system

(2) An equivalent

(3) Transfer Function Derivation

$$Y(z)=\frac{T}{2}\frac{1+z^{-1}}{1-z^{-1}}(X(z)-\beta Y(z))$$

$$\left(1+\frac{T}{2}\beta\frac{1+z^{-1}}{1-z^{-1}}\right)Y(z)=\frac{T}{2}\frac{1+z^{-1}}{1-z^{-1}}X(z)$$

$$H(z)=\frac{Y(z)}{Z(z)}=\frac{T(1+z^{-1})}{(\beta T+2)+(\beta T-2)z^{-1}}$$

Design: 2-4 Find Equivalence of a given analog filter (IIR):
Including methods in Time Domain and Frequency Domain.
2-5 No analog prototype, from the desired frequency response: FIR
2-6 Computer-Aided Design

2.2 IIR Filter Design

- Normally done by transforming a continuous-time design to discrete-time
  - many "classical" continuous-time filters are known and coefficients
  - tabulated
    - Butterworth
    - Chebyshev
    - Elliptic, etc.
  - possible transformations are
    - Impulse invariant transformation
    - Bilinear transformation
- Butterworth Filters
  - maximally flat in both passband and stopband
    - first $2N-1$ derivatives of $|H(j\Omega)|^2$ are zero at $\Omega=0$ and $\Omega=\infty$

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

$= 1, \quad \text{for } \Omega = 0$

$= 1/\sqrt{2}, \quad \text{for } \Omega = \Omega_C$

$1/\Omega^N, \quad \text{for } \Omega >> \Omega_C$

- $2N$ poles of $H(s)H(-s)$ equally spaced around a circle in the s-plane of radius Ω_C symmetrically located with respect to both real and imaginary axes
  - poles of $H(s)$ selected to be the N poles on the left half plane of s
- coefficients of a continuous-time filter for specific order and cutoff frequency can be found
  - from analysis of above expression
  - from tables

2-3 Infinite Impulse Response (IIR) Filter Design

(Given $H(s) \rightarrow H_d(z)$)

2.3.1 A Synthesis in the Time-Domain: Invariant Design

1. Impulse – Invariant Design

(1) Design Principle

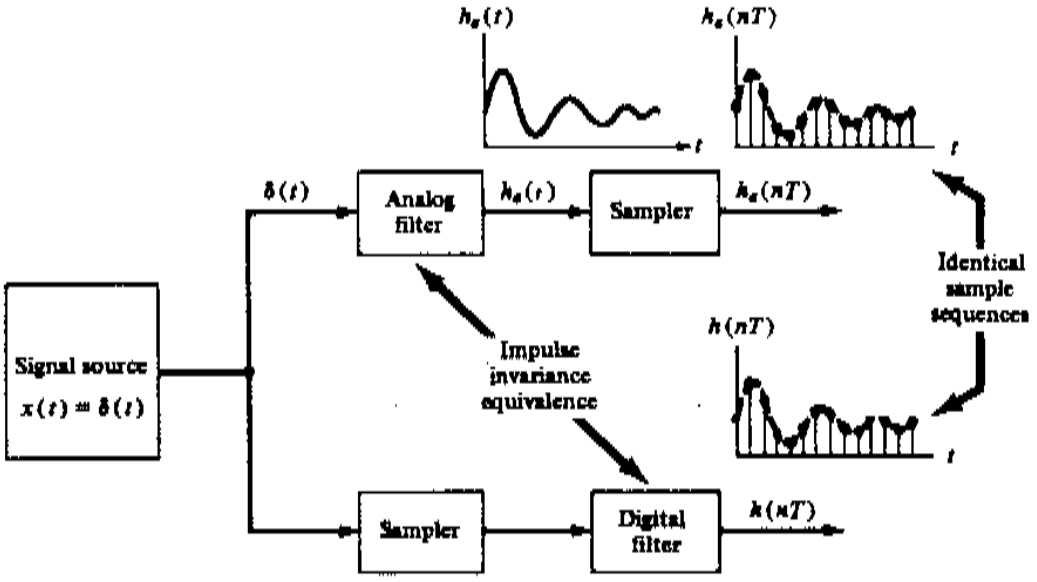


FIGURE 9-11. The concept of impulse invariance.

(2) Illustration of Design Mechanism (Not General Case)

Assume:

(1) Given analog filter (Transfer Function)

$$H_a(s) = \sum_{i=1}^{m} \frac{K_i}{s + s_i} \qquad \text{(a special case)}$$

(2) Sampling Period T (sample $h_a(t)$ to generate $h_a(nT)$)

Derivation:

(1) Impulse Response of analog filter

$$h_a(t) = L^{-1}(H_a(s)) = \sum_{i=1}^{m} k_i e^{-s_i t}$$

(2) $h_a(nT)$: sampled impulse response of analog filter

$$h_a(nT) = \sum_{i=1}^{m} k_i e^{-s_i nT} = \sum_{i=1}^{m} k_i (e^{-s_i T})^n$$

(3) z-transform of $h_a(nT)$

Sampled impulse response of analog filter

$$Z(h_a(nT)) = \sum_{n=0}^{\infty} h_a(nT) z^{-n} = \sum_{n=0}^{\infty} \sum_{i=1}^{m} k_i (e^{-s_i T})^n z^{-n}$$

$$= \sum_{i=1}^{m} k_i \sum_{n=0}^{\infty} (e^{-s_i T} z)^{-n} = \sum_{i=1}^{m} k_i \frac{1}{1 - e^{-s_i T} z^{-1}} = \sum_{i=1}^{m} \frac{k_i}{1 - e^{-s_i T} z^{-1}}$$

(4) Impulse-Invariant Design Principle

$$h(nT) \Leftrightarrow h_a(nT) \Rightarrow Z(h(nT)) \Leftrightarrow Z(h_a(nT))$$

$\uparrow$

Digital filter is so designed that its impulse response *h(nT)* equals the sampled impulse response of the analog filter $h_a(nT)$

Hence, digital filter must be designed such that

$$Z(h(nT)) \Leftrightarrow Z(h_a(nT)) = \sum_{i=1}^{m} \frac{k_i}{1 - e^{-s_i T} z^{-1}}$$

z-transfer function $H(z)$ of the digital filter. Of course, the z-transfer function of its impulse response.

$$\Rightarrow H(z) \Leftrightarrow \sum_{i=1}^{m} \frac{k_i}{1 - e^{-s_i T} z^{-1}}$$

(5) $TZ(h(nT))\Big|_{z=e^{j\omega T}} \xrightarrow{T \to 0} L(h_a(t))$ (scaling)

$$=> H(z) = T \sum_{i=1}^{m} \frac{k_i}{1 - e^{-s_i T} z^{-1}}$$

(3) Characteristics

(1) $H(z)\Big|_{z=e^{j\omega T}} \to H_a(j\omega)$ when $T \to 0$

$\uparrow$

frequency response of digital filter

(2) $T \neq 0$ $\quad H(z)\Big|_{z=e^{j\omega T}} \neq H_a(j\omega)$

(3) Design: Optimized for $T = 0$

Not for $T \neq 0$ (practical case) (due to the design principle)

(4) Realization: Parallel

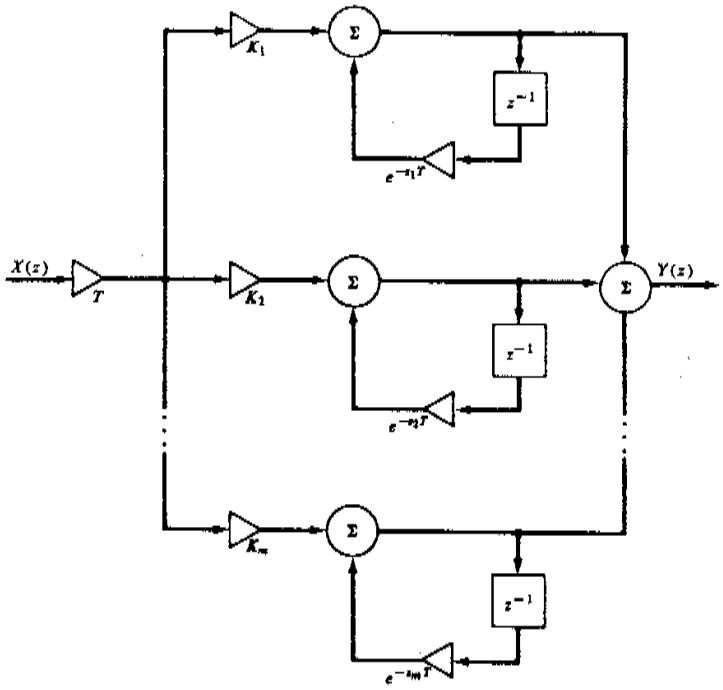


FIGURE 9-12. Impulse-invariant digital filter equivalent to (9-43).

(5) Design Example $H_a(s) = \dfrac{1}{s+1}$ $T = 2s$

Solution: $m = 1, K_1 = 1, s_1 = -1$

$$H(z) = T\sum_{i=1}^{m} \frac{K_i}{1 - e^{-s_i T} z^{-1}} = 2\frac{1}{1 - e^{2} z^{-1}} = \frac{2}{1 - e^{2} z^{-1}}$$

2. General Time – Invariant Synthesis

Design Principle

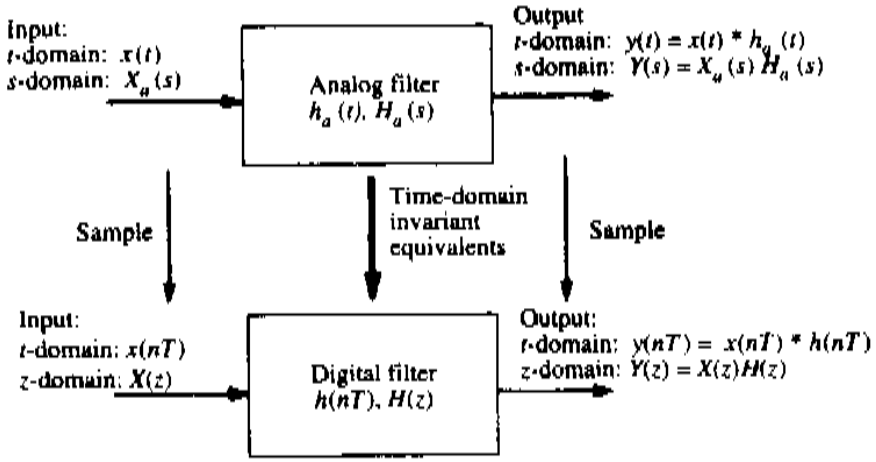


FIGURE 9-13. Time-domain invariance.

(1) Derivation

Given: $H_a(s)$

transfer function of analog filter

$X_a(s)$ Lapalce transform of input signal of analog Filter

T sampling period

Find $H(z)$ z-transfer function of digital filter

(1) Response of analog filter $x_a(t)$

$$y_a(t) = L^{-1}[H_a(s)X_a(s)]$$

(2) $y_a(nT)$ sampled signal of analog filter output

$$y_a(nT) = [L^{-1}[H_a(s)X_a(s)]\big|_{t=nT}$$

(3) z-transform of $y_a(nT)$

$$Z[y_a(nT)] = Z\{[L^{-1}[H_a(s)X_a(s)]\big|_{t=nT}\}$$

(4) Time – invariant Design Principle

$$y(nT) = y_a(nT) \Rightarrow Z(y(nT)) \Leftrightarrow Z[y_a(nT)]$$

↑

Digital filter is so designed
that its output equals the sampled
output of the analog filter

Incorporate the scaling :

$$\underbrace{Z[y(nT)]}_{H(z)X(z)} \Leftrightarrow \underbrace{G}_{G=T} Z[y_a(nT)]$$

↑

z-transfer function of digital filter T

(5) Design Equation

$$H(z) = \frac{G}{X(z)} Z\{[L^{-1}(H_a(s)X_a(s))]\big|_{t=nT}\}$$

special case $X(z)=1$, $X_a(s) = 1$ (impulse)

$$\Rightarrow H(z) = GZ\{[L^{-1}(H_a(s))]\big|_{t=nT}\}$$

(6) Design procedure

A: Find $L^{-1}[H_a(s)X_a(s)] = y_a(t)$ (output of analog filter)

B: Find $y_a(nT) = y_a(t)\big|_{t=nT}$

C: Find $Z(y_a(nT))$

D: $H(z) = GZ(y_a(nT))$

Example 3 $H_a(s) = \dfrac{0.5(s+4)}{(s+1)(s+2)}$

Find digital filter H(z) by impulse - invariance.

Solution of design:

(1) Find $L^{-1}[H_a(s)X_a(s)] = y_a(t)$

$$X_a(s) = 1$$

$$H_a(s) = \frac{0.5(s+4)}{(s+1)(s+2)} = \frac{1.5}{s+1} - \frac{1}{s+2}$$

$$y_a(t) = L^{-1}[H_a(s)X_a(s)] = 1.5e^{-t} - e^{-2t}$$

(2) Find $y_a(nT) = y_a(t)\big|_{t=nT}$

$$y_a(nT) = 1.5(e^{-T})^n - (e^{-2T})^n$$

(3) Find $Z(y_a(nT))$

$$Z(y_a(nT)) = \frac{1.5}{1-e^{-T}z^{-1}} - \frac{1}{1-e^{-2T}z^{-1}}$$

(4) Find z-transfer function of the digital filter

$$H(z) = GZ(y_a(nT))$$

$$= G\left[\frac{1.5}{1-e^{-T}z^{-1}} - \frac{1}{1-e^{-2T}z^{-1}}\right]$$

use G = T

$$H(z) = \frac{1.5T}{1-e^{-T}z^{-1}} - \frac{T}{1-e^{-2T}z^{-1}}$$

(5) Implementation

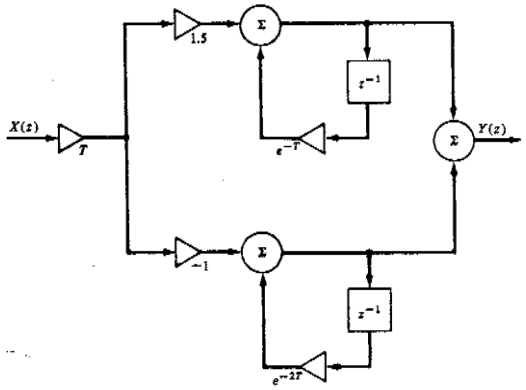


FIGURE 9-14. Impulse-invariant filter for Example 9-5.

Characteristics

(1) Frequency Response equations: analog and digital

$$\text{Analog}: H_a(j\omega) = \frac{0.5(j\omega+4)}{(1+j\omega)(2+j\omega)}$$

$$\text{Digital}: H(e^{j\omega T}) = \frac{1.5T}{1-e^{-T}e^{-j\omega T}} - \frac{T}{1-e^{-2T}e^{-j\omega T}}$$

(2) dc response comparison ($\omega = 0$)

Analog: $H_a(0) = \dfrac{0.5 \times 4}{1 \times 2} = 1$

Digital: $H(e^{j0}) = H(1) = \dfrac{1.5T}{1 - e^{-T}} - \dfrac{T}{1 - e^{-2T}}$

Varying with T (should be)

$$T \to 0: \quad e^{-T} \approx 1 - T, \quad e^{-2T} \approx 1 - 2T$$

$$H(1) \approx \frac{1.5T}{1 - (1 - T)} - \frac{T}{1 - (1 - 2T)} = 1$$

$T \neq 0$: for example $T = \dfrac{2\pi}{20} = 0.31416\,s$ $\qquad (f_s = 20)$

$$e^{-T} = 0.7304, \quad e^{-2T} = 0.5335$$

$H(1) = 1.0745$ good enough

(3) $|H_a(j\omega)|$ versus $|H(e^{j\omega T})|$: $T = \dfrac{2\pi}{20} = 0.31416\,\text{sec}\,ond$

Using normalized frequency $r = f / f_s = \omega / \omega_s$

$$|H(e^{j2\pi r})| = \frac{\pi}{10}\sqrt{\frac{0.25488 - 0.06983\cos 2\pi r}{2.74925 - 3.51275\cos 2\pi r + 0.77932\cos 4\pi r}}$$

$$|H_a(j\omega) = 0.5\sqrt{\frac{16 + \omega^2}{4 + 5\omega^2 + \omega^4}} \qquad \omega = 2\pi f = 2\pi f_s r = \frac{2\pi}{T} r$$

$$\Rightarrow H_a(r)$$

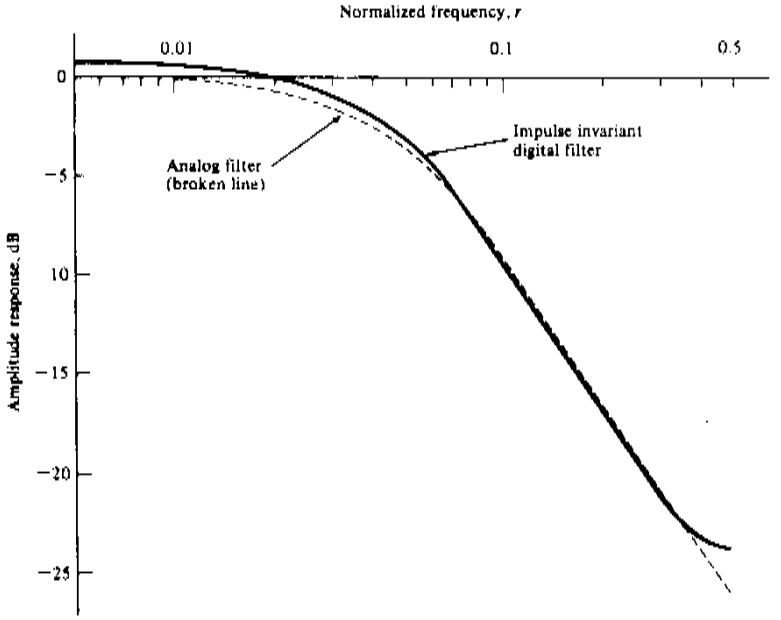


FIGURE 9-15. Amplitude response of impulse-invariant filter.

(4) $\angle H$ versus $\angle H_a$

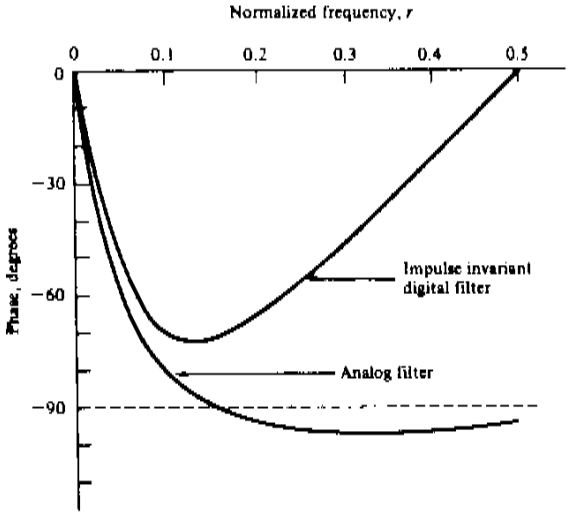


FIGURE 9-16. Phase response of impulse-invariant filter.

(5) Gain adjustment when $T \neq 0$

$T \neq 0$ => frequency response inequality

adjust G => $H(e^{j\omega T}) = H_a(j\omega)$ at a special ω

for example $\omega = 0$

If $G = T = 0.3142$ => $H(e^{j\omega T})\,|_{\omega=0} = 1.0745$

If selecting $G = T/1.0745$ => $H(e^{j\omega T})\,|_{\omega=0} = 1$

3. Step – invariance synthesis

$$X_a(s) = \frac{1}{s} \qquad X(z) = \frac{1}{1 - z^{-1}}$$

$$H(z) = \frac{G}{X(z)} Z\{[L^{-1}[H_a(s)X_a(s)]]|_{t=nT}\}$$

$$\Rightarrow H(z) = G(1-z^{-1})Z\{[L^{-1}[\frac{1}{s}H_a(s)]]|_{t=nT}\}$$

Example 9-6 $H_a(s) = \dfrac{0.5(s+4)}{(s+1)(s+2)}$. Find its step-invariant equivalent.

Solution of Design

$$\frac{1}{s}H_a(s) = \frac{0.5(s+4)}{s(s+1)(s+2)} = \frac{1}{s} - \frac{1.5}{s+1} + \frac{0.5}{s+2}$$

$$y_a(nT) = y_a(t)|_{t=nT} = L^{-1}[\frac{1}{s}H_a(s)]|_{t=nT} = 1 - 1.5e^{-nT} + 0.5e^{-2nT}$$

$$H(z) = G(1-z^{-1})Z[y_a(nT)]$$

$$= G(1-z^{-1})(\frac{1}{1-z^{-1}} - \frac{1.5}{1-e^{-T}z^{-1}} + \frac{0.5}{1-e^{-2T}z^{-1}}$$

Comparison with impulse-invariant equivalent.

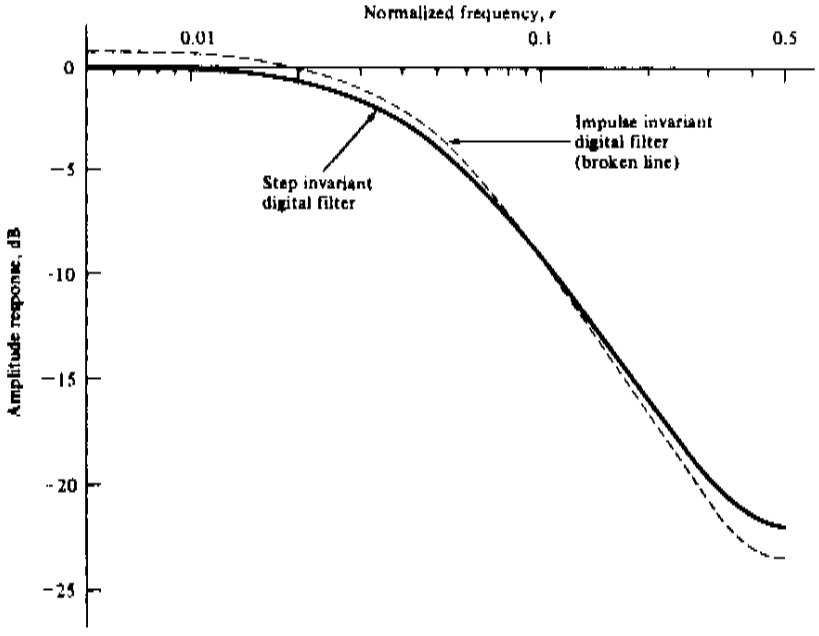


FIGURE 9-17. Amplitude response of step-invariant filter.

2-5 Design in the Frequency Domain --- The Bilinear z-transform

1. Motivation (problem in Time Domain Design)

$$x(t) \quad \leftrightarrow \quad x_s(nT)$$

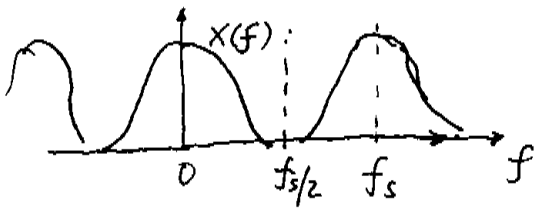
$$\updownarrow \qquad\qquad \updownarrow$$

$$X(f) \qquad X_s(f) = \sum_{n=-\infty}^{\infty} C_n X(f - nf_s)$$

$$= C_n X(f) + [C_{-1} X(f + f_s) + + [C_1 X(f - f_s)] + \cdots$$

Introduced by sampling, undesired!

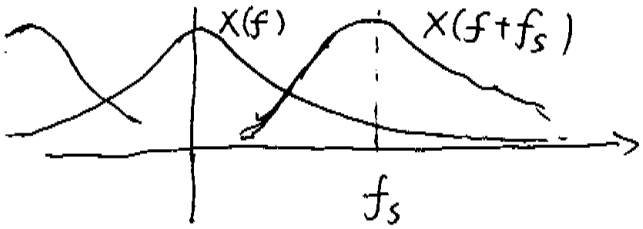
$x(t)$ bandlimited ($X(f = f_s/2) = 0$)



$$X(f) = X_s(f)$$

for $|f| < \frac{f_s}{2}$

X(t) not bandlimited



$$X(f) \neq X_s(f + f_s)$$

for $|f| < \frac{f_s}{2}$

Consider digital equivalent of an analogy filter $H_a(f)$: $H_a(j\omega) = H_a(j2\pi f))$

$H_a(f)$: bandlimited => can find a $H_d(z)$

$H_a(f)$: not bandlimited => can not find a $H_d(z)$ Such that $H_d(e^{j\omega T}) \neq H_a(\omega)$

2. Proposal: from ω axis to ω_1 axis

$\omega = \infty \rightarrow \omega_1 = 0.5\omega_s$ (ω_s: given sampling frequency)

(s plane to s_1 plane)

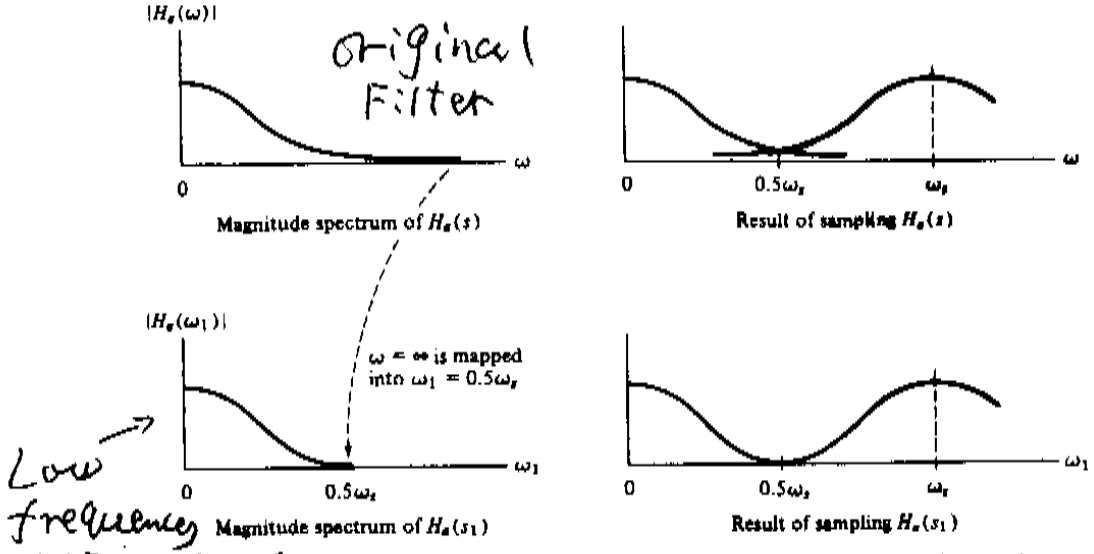


FIGURE 9-19. Effect of mapping the s-plane into the s_1-plane.

for example $H_a(\omega = 0.01\omega_s) \doteq H_a(\omega_1 = 0.01\omega_s)$

Observations: (1) Good accuracy in low frequencies

(2) Poor accuracy in high frequencies

(3) 100% Accuracy at $\omega = \omega_1 = \omega^*$

a given specific number such that $0.01\,\omega_s$

Is it okay to have poor accuracy in high frequencies? Yes! Input is bandlimited!

What do we mean by good, poor and 100% accuracy?

Assume (1) $H_a(\omega) = \dfrac{1}{j\omega + 1}$ (originally given analogy filter)

(2) The transform is $\omega = 0.915244 \tan \omega_1$

Then, $\dfrac{1}{j0.915\tan\omega_1 + 1}$ is a function of ω_1. Denote $\dfrac{1}{j0.915\tan\omega_1 + 1} = \tilde{H}_a(\omega_1)$.

Good accuracy:

$$H_a(\omega = 0.3) = \frac{1}{j0.3+1} \approx \frac{1}{j0.915\tan(0.3)+1} = \frac{1}{j0.28+1} = \tilde{H}_a(\omega_1 = 0.3)$$

Poor Accuracy

$$H_a(\omega = 1.5) = \frac{1}{j1.5+1} \neq \frac{1}{j0.915\tan(1.5)+1} = \frac{1}{j12.9+1} = \tilde{H}_a(\omega_1 = 1.5)$$

100% Accuracy (Equal)

$$H_a(\omega = 0.5) = \frac{1}{j0.5+1} = \frac{1}{j0.915\tan(0.5)+1} = \frac{1}{j0.5+1} = \tilde{H}_a(\omega_1 = 0.5)$$

Is $\tilde{H}(\omega_1)$ bandlimited? That is, can we find a ω_1 such that $\tilde{H}(\omega_1)$=0?

Yes, $\omega_1 = \pi/2$. ➔ We have no problem to find a digital equivalent $H_d(e^{j\omega_1 T}) \approx \tilde{H}_a(\omega_1)$ without aliasing!

Let's use $\omega^{(L)}$ as a number (for example 0.2) representing any low frequency,

Then, because $\tilde{H}_a(\omega^{(L)}=0.2)$ is a good approximation of $H_a(\omega^{(L)}=0.2)$,

$H_d(e^{j(\omega^{(L)}=0.2)T}) \approx H_a(\omega^{(L)}=0.2)$ should be a good approximation.

A digital filter can thus be designed for an analogy filter $H_a(\omega)$ which is not bandlimted!

Two Step Design Procedure:

Given: analogy filter $H_a(s)$

(1) Find an bandlimited analogy approximation ($\tilde{H}_a(s_1)$) for $H_a(s)$

(2) Design a digital equivalent $(H_d(z))$ for the bandlimited filter $\tilde{H}_a(s_1)$.

Because of the relationship between ($\tilde{H}_a(\omega_1)$) for $H_a(\omega)$,

$H_d(z)$ is also digital equivalent of $H_a(s)$.

The overlapping (aliasing) problem is avoided!

The designed digital filter can approximate $H_a(\omega)$ (for ω_1 and ω take the same value) at low frequency.

3. ω axis to ω_1 axis (s plane to s_1 plane) transformation

Requirement : $\omega = \infty \rightarrow \omega_1 = \dfrac{\omega_s}{2}$ (ω_s is given sampling frequency.)

Proposed transformation :

Variable in ω domain

Variable in ω_1 domain

$$\omega = C\tan\left[\frac{\omega_1}{\frac{1}{2}\omega_s}\frac{\pi}{2}\right] = C\tan\frac{\omega_1 T}{2}$$

Constant

Effect of C:

We want the transformation map

$\omega = \omega_r$ (for example, $\omega_r = 100 rad/s$) to $\omega_1 = \omega_r$ (ω^*)

=> $\omega_r = C\tan\dfrac{\omega_r T}{2} \Rightarrow C = \omega_r \cot\dfrac{\omega_r T}{2}$

i.e. when the sampling period T is given, C is the only parameter which determines what ω will be mapped into ω_1 axis with the same value.

Example: $H_a(s) = \dfrac{\omega_c^{\,2}}{s^2 + \sqrt{2}\omega_c s + \omega_c^{\,2}}$

$$H_a(j\omega) = \frac{\omega_c^{\,2}}{(\omega_c^{\,2} - \omega^2) + j\sqrt{2}\omega_c\omega}$$

not bandlimited

If we want to map $\omega = 2\pi \times 100$ to $\omega_1 = 2\pi \times 100$

$\Rightarrow \quad \omega_r = 200\pi$

$$C = 200\pi \cot\frac{200\pi T}{2}$$

Hence, for any given T or $\omega_s = 2\pi\frac{1}{T}$

$$H_a(jC\tan\frac{\omega_1 T}{2}) = H_a((j200\pi\cot\frac{200\pi T}{2})\tan\frac{\omega_1 T}{2})$$

$$= \frac{{\omega_c}^2}{({\omega_c}^2 - C^2\tan^2\frac{\omega_1 T}{2}) + j\sqrt{2}C\tan\frac{\omega_1 T}{2}\cdot\omega_c}$$

is bandlimited as a function of ω_1 by $|\omega_1| \le \frac{\omega_s}{2} = \frac{\pi}{T}$

$\tilde{H}_a(\omega_1) \stackrel{\Delta}{=} H_a(jC\tan\frac{\omega_1 T}{2}) \approx H_a(j\omega)$ when $\omega_1 = \omega$ at low frequencies.

Further $\tilde{H}_a(\omega_1 = \omega_r = 200\pi) = H_a(j\omega = j\omega_r = j200\pi)$

Exactly holds!

How to select ω_r or sampling frequency ω_s at which $\omega_1 = \omega$?

(1) $\frac{\omega_r T}{2}$ should be small?

why? $T = \frac{1}{f_s} = \frac{2\pi}{\omega_s}$ $\qquad \frac{\omega_r T}{2} << 1 \Rightarrow \frac{\omega_r}{\omega_s}\pi << 1$ or $\omega_r << \omega_s$,

(The accuracy should be good at low frequencies)

(2) When ω_r is given or determined by application, ω_s should be large enough such that $\omega_r << \omega_s$ to ensure the accuracy in the frequency range including ω_r

When $\frac{\omega_r T}{2} << 1$:

$C = \omega_r \frac{2}{\omega_r T} = \frac{2}{T}$ $\qquad$ since $\quad \cot x \approx \frac{1}{x}$ $\quad$ for small x.

4. Design of Digital Filter using bilinear z-transform

- A procedure: (1) $H_a(j\omega) \Rightarrow \tilde{H}_a(\omega_1)$

(not bandlimited, original analog) (bandlimited, analog)

or $H_a(s) \Rightarrow \tilde{H}_a(s_1)$

(2) $H_d(z) \Rightarrow \tilde{H}_a(s_1)\big|_{e^{s_1 T}=z}$

↑ (Transfer function of digital filter)

↑ replace $e^{s_1 T}$ by z

* Question: Can we directly obtain $H_d(z)$ from $H_a(s)$? Yes! (But how?)

- Bilinear z-transform

Preparation : (1) $\tan x = \dfrac{\sin x}{\cos x} = -j\dfrac{e^{jx}-e^{-jx}}{e^{jx}+e^{-jx}}$

(2) $\omega = C\tan\dfrac{\omega_1 T}{2}$

Hence, $\omega = C\tan\dfrac{\omega_1 T}{2} = C(-j)\dfrac{e^{j\frac{\omega_1 T}{2}}-e^{-j\frac{\omega_1 T}{2}}}{e^{j\frac{\omega_1 T}{2}}+e^{-j\frac{\omega_1 T}{2}}}$

Replace $j\omega$ by s , $j\omega_1$ by s_1 $(\omega = s/j = -js, \qquad \omega_1 = -js_1)$

$$-js = -jC\frac{e^{s_1 T/2}-e^{-s_1 T/2}}{e^{s_1 T/2}+e^{-s_1 T/2}}$$

$$\Rightarrow s = C\frac{e^{s_1 T/2}-e^{-s_1 T/2}}{e^{s_1 T/2}+e^{-s_1 T/2}} = C\frac{1-e^{-s_1 T}}{1+e^{-s_1 T}}$$

Replace $e^{s_1 T} = z$ for digital filter

$s = C\dfrac{1-z^{-1}}{1+z^{-1}} \Leftarrow$ direct transformation from s to z (bypass s_1)

inear z – nsformation

Example $H_a(s) = \dfrac{\omega_c{}^2}{s^2+\sqrt{2}\omega_c s+\omega_c{}^2}$

Digital Filter

$$H_d(z) = \frac{\omega_c{}^2}{C^2\dfrac{(1-z^{-1})^2}{(1+z^{-1})^2}+\sqrt{2}\omega_c\dfrac{1-z^{-1}}{1+z^{-1}}+\omega_c{}^2}$$

$$= \frac{\omega_c{}^2(1+z^{-1})^2}{C^2(1-z^{-1})^2+\sqrt{2}\omega_c(1-z^{-2})+\omega_c{}^2(1+z^{-1})^2}$$

C: only undetermined parameter in the digital filter.

To determine C: (1) T (ω_s)

(2) ω_r (related to the frequency range of interest)

Example : $H_a(s) = \dfrac{\omega_c^{\,2}}{s^2 + \sqrt{2}\omega_c s + \omega_c^{\,2}}$

ω_c : break frequency

Take $\omega_r = \omega_c$

Consider $f_c = 500\,Hz$ $(\omega_c = 2\pi \times 500)$

$f_s = 2000\,Hz$ $(T = \dfrac{1}{f_s} = 0.0005\,\text{sec})$

➔ C determined => $H_d(z)$ determined

➔ $H_d(z) = \dfrac{0.292893 + 0.585786z^{-1} + 0.292893z^{-2}}{1 + 0.171573z^{-2}}$

➔ $H_d(e^{j2\pi r})$, $|H_d(e^{j2\pi r})|$, $\angle H_d(e^{j2\pi r})$

To compare the frequency response with the original analog filter H_a :

$$H_a(j\omega) = H_a(j2\pi f) = H_a(j2\pi r f_s) \overset{f_s = 4f_c}{=} H_a(j8\pi r f_c)$$

(replace s by $j8\pi r f_c$ *in* $H_a(s)$)

➔ $|H_a|$, $\angle H_a$

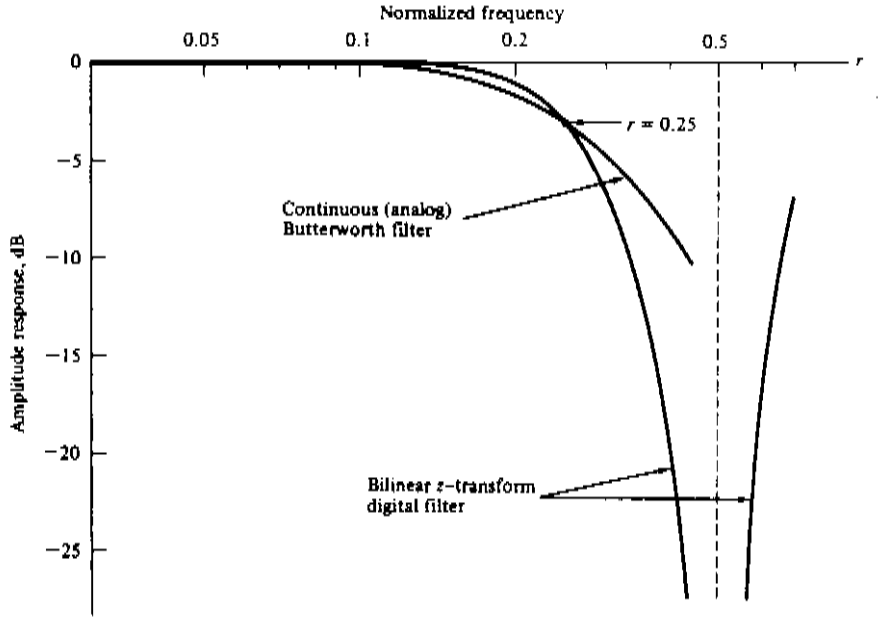


FIGURE 9-20. Amplitude response of bilinear z-transform filter.

Too low f_s => poor accuracy in f_c.

2-6 Bilinear z-Transform Bandpass Filter

1. Construction Mechanism

(1) From an analog low-pass filter $H_a(s)$

to analog bandpass filter $H_a(\frac{s^2+\omega_c^{\ 2}}{s\omega_b})$

i.e., replace s by $\frac{s^2+\omega_c^{\ 2}}{s\omega_b}$ to form a bandpass filter

in the low pass filter

For example $H_a(s)=\frac{1}{s+1}$ low-pass

$\frac{1}{\frac{s^2+\omega_c^{\ 2}}{s\omega_b}+1}$ band-pass

Why? Original low-pass

$H_a(j\omega)$

↑

Low => High Gain

High => Low Gain

After Replacement

$$H_a(\frac{-\omega^2+\omega_c^{\ 2}}{j\omega\omega_b})=H_a(j\frac{\omega^2-\omega_c^{\ 2}}{\omega\omega_b})$$

high $\omega \Rightarrow \frac{\omega^2-\omega_c^{\ 2}}{\omega\omega_b}\approx\frac{\omega^2}{\omega\omega_b}=\frac{\omega}{\omega_b}$ => high => low gain

low $\omega \Rightarrow \frac{\omega^2-\omega_c^{\ 2}}{\omega\omega_b}\approx-\frac{\omega_c^{\ 2}/\omega_b}{\omega}$ => high => low gain

(2) From analog to digital

Replace s in $H_a(\frac{s^2+\omega_c^{\ 2}}{s\omega_b})$ by $C\frac{1-z^{-1}}{1+z^{-1}}$

➔ $H_d(z^{-1})$

for example

$$\frac{1}{\frac{s^2+\omega_c{}^2}{s\omega_b}+1} \Rightarrow \frac{1}{\frac{C^2(1-z^{-1})^2}{(1+z^{-1})^2}+\omega_c{}^2}{\frac{1-z^{-1}}{1+z^{-1}}\omega_b}+1}$$

digital bandpass filter

2. Bilinear z-transform equation

Analog Low-pass		Bandpass (analog)
s	$\Rightarrow$	$\frac{s^2+\omega_c{}^2}{s\omega_b}$
	$\Rightarrow$	$\frac{\left[C\frac{1-z^{-1}}{1+z^{-1}}\right]^2+\omega_c{}^2}{C\left[\frac{1-z^{-1}}{1+z^{-1}}\right]\omega_b}$
		$=\frac{C^2(1-z^{-1})^2+\omega_c{}^2(1+z^{-1})^2}{C\omega_b(1-z^{-2})}$

Direct Transformation s (in low-pass)		
	$\Rightarrow$	$\frac{C^2(1-z^{-1})^2+\omega_c{}^2(1+z^{-1})^2}{C\omega_b(1-z^{-2})}$
		$=\frac{C^2+\omega_c{}^2}{C\omega_b}\frac{1-2\left[\frac{C^2-\omega_c{}^2}{C^2+\omega_c{}^2}\right]z^{-1}+z^{-2}}{1-z^{-2}}$

s (in low-pass)		
	$\Rightarrow$	$A\frac{1-Bz^{-1}+z^{-2}}{1-z^{-2}}$

$$A=\frac{C^2+\omega_c{}^2}{C\omega_b}$$

with

$$\mathbf{B}=2\frac{C^2-\omega_c{}^2}{C^2+\omega_c{}^2}$$

3. How to select (C,ω_c,ω_b) for bandpass filter (design)

Important parameters of bandpass

(1) center frequency ω_c

(2) ω_u upper critical frequency

(3) ω_l low critical frequency

Selection of ω_c for bandpass: $\omega_c{}^2 = \omega_u \omega_l$

Design of (C, ω_c, ω_b)

We want $\omega_u = C \tan \frac{\omega_u T}{2}$, Also want $\omega_l = C \tan \frac{\omega_l T}{2}$

one parameter C => impossible

⇨ solution $\omega_c{}^2 \Rightarrow C^2 \tan \frac{\omega_u T}{2} \tan \frac{\omega_l T}{2}$

⇨ bandwidth $\omega_b \Rightarrow C \tan \frac{\omega_u T}{2} - C \tan \frac{\omega_l T}{2}$

Hence, A and B can be determined to perform the transform.

4. Convenient design equation

$$A = \frac{C^2 + C^2 \tan \frac{\omega_u T}{2} \tan \frac{\omega_l T}{2}}{C^2 (\tan \frac{\omega_u T}{2} - \tan \frac{\omega_l T}{2})}$$

$$= \frac{1 + \tan \frac{\omega_u T}{2} \tan \frac{\omega_l T}{2}}{\tan \frac{\omega_u T}{2} - \tan \frac{\omega_l T}{2}}$$

why no C?

$$I \xrightarrow[\tan(x-y) = \frac{\tan x - \tan y}{1 + \tan x \tan y}]{\text{trigonomatric identity}} \cot[(\omega_u - \omega_l)\frac{T}{2}]$$

$$= 2 \frac{1 - \tan \frac{\omega_u T}{2} \tan \frac{\omega_l T}{2}}{1 + \tan \frac{\omega_u T}{2} \tan \frac{\omega_l T}{2}}$$

$$\frac{1-\tan x\tan y}{1+\tan x\tan y}=\frac{\cos(x+y)}{\cos(x-y)} \Rightarrow \quad 2\frac{\cos(\omega_u+\omega_l)\frac{T}{2}}{\cos(\omega_u-\omega_l)\frac{T}{2}}$$

In low-pass $\nearrow$ $s \Rightarrow A\dfrac{1-Bz^{-1}+z^{-2}}{1-z^{-2}}$

Example : $\dfrac{1}{s+\alpha} \Rightarrow \dfrac{1}{A\dfrac{1-Bz^{-1}+z^{-2}}{1-z^{-2}}+\alpha} = \dfrac{1-z^{-2}}{A(1-Bz^{-1}+z^{-2})+\alpha(1-z^{-2})}$

5. In the normalized frequency

Reference frequency: sampling frequency f_s (ω_s)

$$r_u \overset{\Delta}{=} \frac{\omega_u}{\omega_s} = \frac{f_u}{f_s} \qquad r_l \overset{\Delta}{=} \frac{\omega_l}{\omega_s} = \frac{f_l}{f_s}$$

$$\Rightarrow \frac{\omega_u T}{2} = \frac{2\pi f_u \frac{1}{f_s}}{2} = \pi r_u, \quad \frac{\omega_l T}{2} = \pi r_l$$

$$\Rightarrow \begin{cases} A = \cot \pi(r_u - r_l) \\ B = 2\dfrac{\cos \pi(r_u + r_l)}{\cos \pi(r_u - r_l)} \end{cases}$$

In low-pass $\nearrow$ s $\Rightarrow A\dfrac{1-Bz^{-1}+z^{-2}}{1-z^{-2}}$

Example : Lowpass $H_a(s) = \dfrac{1}{s+1}$

Transfer function of bandpass digital filter

$$H_d(z)\frac{1}{A\dfrac{1-Bz^{-1}+z^{-2}}{1-z^{-2}}+1}$$

A and B? Determined by design requirements.

$\begin{cases} f_u = 1000\,Hz \\ f_l = 500\,Hz \end{cases}$ sampling frequency $f_s = 5000Hz$

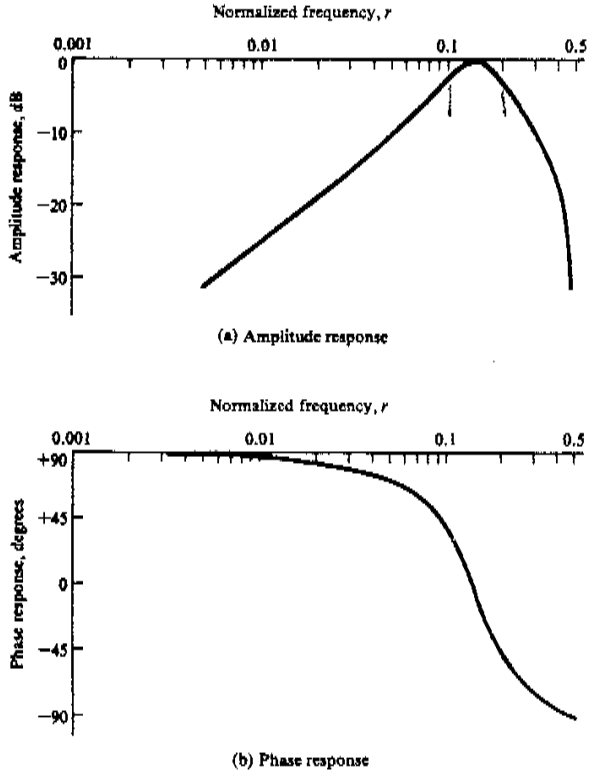
$\Rightarrow \begin{cases} r_u = f_u / f_s = 0.2 \\ r_l = f_l / f_s = 0.1 \end{cases}$

$$A = \cot \pi(0.2 - 0.1) = 3.0776835$$

$$\Rightarrow \quad B = 2\frac{\cot \pi(0.2 + 0.1)}{\cot \pi(0.2 - 0.1)} = 1.2360680$$

$$\Rightarrow \quad H(z) = \frac{1 - z^{-2}}{4.0776835 - 3.8042261\, z^{-1} + 2.0776835\, z^{-2}}$$

$H(e^{j2\pi r})$, $|H(e^{j2\pi r})|$, $\angle H(e^{j2\pi r})$



(a) Amplitude response

(b) Phase response

FIGURE 9-23. Amplitude and phase responses for example bandpass filter.

UNIT-III
FIR FILTER DESIGN

Design of Finite-Duration Impulse Response (FIR) Digital Filter

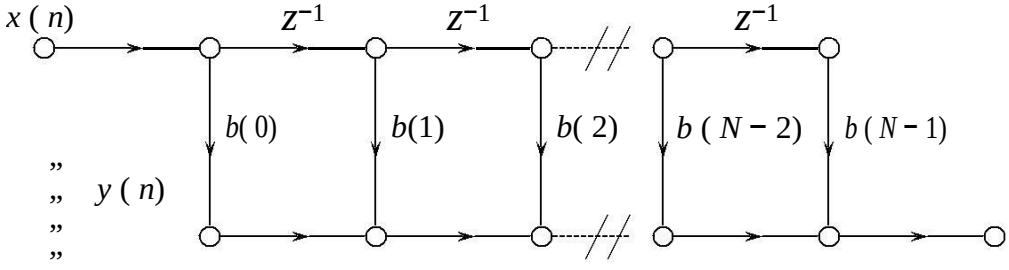
Direct Design of Digital Filters with no analog prototype.
Can we also do this for IIR? Yes!

3.1 Structures of FIR :

- System function $H(z) = \sum_{k=0}^{M} b_k z^{-k}$
- Difference equation $y(n) = \sum_{k=0}^{M} b_k x(n-k)$

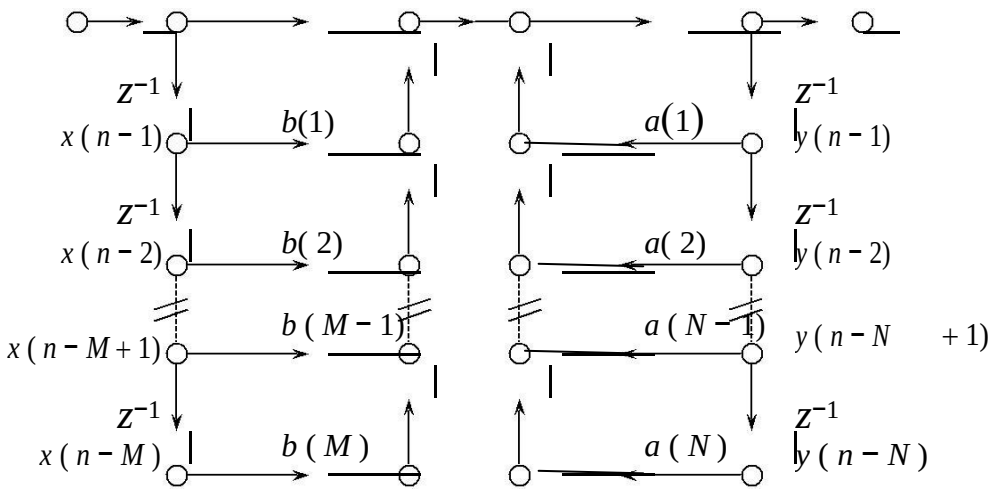
  - Output is weighted sum of current and previous M inputs
  - The filter has order M
  - The filter has $M+1$ taps, i.e. impulse response is of length $M+1$
  - $H(z)$ is a polynomial in z^{-1} of order M
  - $H(z)$ has M poles at $z = 0$ and M zeros at positions in the z-plane determined by the coefficients b_k

Direct Form FIR Filter



  - Also known as moving average (MA) and non-recursive

- Direct Form 1 IIR Filter (for $M=N$)

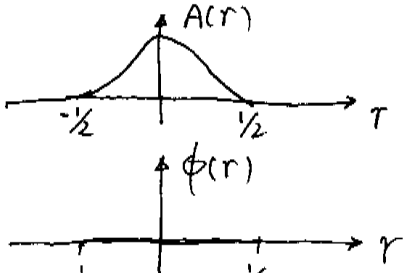


3.2 Linear Phase FIR Filters

- For causal linear phase FIR filters, the coefficients are symmetric $h(n) = h(N-1-n)$
  - Linear phase filters do not introduce any phase distortion
    - they only introduce delay
    - The delay introduced is $(N-1)/2$ samples
  - Zeros occur in mirror image pairs
    - if z_0 is a zero, then $1/z_0$ is also a zero
  - Symmetry leads to efficient implementations
    - $N/2$ multiplications (N even) or $(N+1)/2$ multiplications (N odd) per output sample instead of N for the general case.

3.2 -A A few questions

1. How are the specifications given?

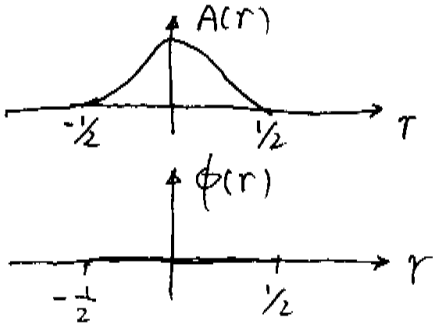


By given $A(\omega)$ and $\phi(\omega)$

2. What is the form of FIR digital filter?
   Difference Equation $y(nT) = \sum_k h(kT)x(nT - kT)$
   (What is T ? sampling period)

   Transfer function $H(z) = \sum_k h(kT)z^{-k}$

3. How to select T ?
4. After T is fixed, can we define the normalized frequency r and $A(r)$ and $\phi(r)$? Yes!



Can we then find the desired frequency response
$H(r) = A(r)e^{j\phi(r)}$? Yes!

5. Why must $H(r)$ be a periodic function for digital filter?
   $H(r) = H(n + r)$? Why? What is its period?

$T_r = 1$

6. Can $H(r)$ be expressed in Fourier Series ? Yes!
How?

ω_0 : sampling frequency? No! T_0 : period of $x(t)$

See general formula :

$$x(t) = \sum_{n=-\infty}^{\infty} X_n e^{jn\omega_0 t} = \sum_{n=-\infty}^{\infty} X_n e^{jn2\pi f_0 t} = \sum_{n=-\infty}^{\infty} X_n e^{jn\frac{2\pi}{T_0}t}$$

$$X_n = \frac{1}{T_0}\int_{T_0} x(t)e^{-jn\omega_0 t}dt = \frac{1}{T_0}\int_{-T_0/2}^{T_0/2} x(t)e^{-jn\frac{2\pi}{T_0}t}dt$$

In our case for $H(r)$:

$$\left.\begin{array}{l} x \to H \\ t \to r \\ T_0 \to 1 \end{array}\right\} \Rightarrow \begin{array}{l} H(r) = \sum_{n=-\infty}^{\infty} X_n e^{jn2\pi r} \\ X_n = \int_{-1/2}^{1/2} H(r)e^{-jn2\pi r}dr \end{array}$$

What does this mean? Every desired frequency response $H(r)$ of digital filter can be expressed into Fourier Series ! Further, the coefficients of the Fourier series can be calculated using $H(r)$!

3-2 -B Design principle

$$H(r) = \sum_{n=-\infty}^{\infty} X_n e^{jn2\pi r}$$

Denote $h_d(nT) = X_{-n} \Rightarrow \begin{cases} H(r) = \sum_{n=-\infty}^{\infty} h_d(nT)e^{-jn2\pi r} \\ h_d(nT) = X_{-n} = \int_{-1/2}^{1/2} H(r)e^{jn2\pi r}dr \end{cases}$

Consider a filter with transfer function $\sum_{n=-\infty}^{\infty} h_d(nT)z^{-n}$

What's its frequency response ?

$$\sum_{n=-\infty}^{\infty} h_d(nT)(e^{j2\pi r})^{-n} = \sum_{n=-\infty}^{\infty} h_d(nT)e^{-j2\pi rn} = H(r)$$

given specification of digital filter's frequency response!

3-3 C Design Procedure

(1) Given $H(r)$

(2) Find ***H(r)*'s Fourier series** $H(r)=\sum_{n=-\infty}^{\infty} h_d(nT)e^{-j2\pi rn}$

where $h_d(nT)=\int_{-1/2}^{1/2} H(r)e^{jn2\pi r}dr$

(3) Designed filter's transfer function

$$\sum_{n=-\infty}^{\infty} h_d(nT)z^{-n}$$

What's $h_d(nT)$? Impulse response!

Example 3-1: $H(r)=\frac{1}{2}(1+\cos 2\pi r)$

Solution :

(1) Given $H(r)$: done

(2) Find $H(r)$'s Fourier series

$$H(r)=\frac{1}{2}(1+\cos 2\pi r)=\sum_{n=-\infty}^{\infty} h_d(nT)e^{-jn2\pi r}$$

where $h_d(nT)=\frac{1}{2}\int_{-1/2}^{1/2}(1+\cos 2\pi r)e^{jn2\pi r}dr$

$n = 0$

$$h_d(0)=\frac{1}{2}\int_{-1/2}^{1/2}(1+\cos 2\pi r)dr=\frac{1}{2}\int_{-1/2}^{1/2}dr+\frac{1}{2}\int_{-1/2}^{1/2}\cos 2\pi r dr=\frac{1}{2}\quad n\neq 0$$

$$h_d(nT)=\frac{1}{2}\int_{-1/2}^{1/2}e^{jn2\pi r}dr+\frac{1}{2}\int_{-1/2}^{1/2}\frac{e^{j2\pi r}+e^{-j2\pi r}}{2}e^{j2\pi nr}dr$$

$$=\frac{1}{4}\int_{-1/2}^{1/2}e^{j2\pi(n+1)r}dr+\frac{1}{4}\int_{-1/2}^{1/2}e^{j2\pi(n-1)r}dr$$

$$\Rightarrow h_d(T)=\frac{1}{4}\int_{-1/2}^{1/2}dr=\frac{1}{4}$$

$$h_d(-T)=\frac{1}{4}\int_{-1/2}^{1/2}dr=\frac{1}{4}$$

$$h_d(nT)=0 \qquad n\neq 0,\pm 1$$

➔
$$\begin{cases} H(r)=\sum_{n=-\infty}^{\infty} h_d(nT)e^{-j2\pi nr}=\frac{1}{4}e^{-j2\pi r}+\frac{1}{2}+\frac{1}{4}e^{-j2\pi r} \\ h_d(0)=\frac{1}{2}, \qquad h_d(\pm T)=\frac{1}{4}, \quad h_d(nT)=0, \quad |n|\geq 2 \end{cases}$$

(3) Digital Filter

$$\sum_{n=-\infty}^{\infty} h_d(nT)z^{-n}=h_d(-T)z^{1}+h_d(0)+h_d(T)z^{-1}=\frac{1}{4}z+\frac{1}{2}+\frac{1}{4}z^{-1}$$

3.4-D Practical Issues : Infinite number of terms and non-causal

(1) $H_{nc}(z) = \sum_{n=-\infty}^{\infty} h_d(nT) z^{-n}$ 2M+1 terms

Truncation => $H_{nc}(z) = \sum_{n=-M}^{M} h_d(nT) z^{-n} (= \sum_{n=-\infty}^{\infty} w_r(n) h_d(nT) z^{-n})$

Rectangular window function

$$w_r(n) = \begin{cases} 1 & |n| \le M \\ 0 & |n| > M \end{cases}$$

Truncation ⇔ window

Effect of Truncation (windowing):

Time Domain: Multiplication (h and w)

Frequency Domain: Convolution

$$w_r(e^{j2\pi r}) = \sum_{n=-M}^{M} e^{-j2\pi nr} = \frac{\sin \pi(2M+1)r}{\sin \pi r}$$

(After Truncation: The desired frequency H_r

⇨ $H_r w_r$ frequency response of truncated filter)

The effect will be seen in examples!

(2) Causal Filters:

$$H_c(z) = z^{-M} \sum_{n=-M}^{M} h_d(nT) z^{-n} = \sum_{n=-M}^{M} h_d(nT) z^{-(M+n)}$$

$k = n+M$ $H_c(z) = \sum_{k=0}^{2M} h_d(kT - MT) z^{-k}$

Define $L_k = h_d(kT - Mt)$ ➔ $H_c(z) = \sum_{k=0}^{2M} L_k z^{-k}$

Relationship: $H_c(z) = H_{nc}(z) z^{-M}$

Frequency Response $H_c(e^{j2\pi r}) = H_{nc}(e^{j2\pi r}) e^{-j2\pi Mr} \Rightarrow \begin{cases} A_c(r) = A_{nc}(r) \\ \phi_c(r) = \phi_{nc}(r) - 2\pi Mr \end{cases}$

Design Examples

Hamming window: $w_h(n) = \begin{cases} 0.54 + 0.46 \cos \frac{n\pi}{M} & |n| \le M \\ 0 & |n| > M \end{cases}$

Example 3-2 Design a digital differentiator

Step1 : Assign $H(r)$

$H(r)$ should be the frequency response of the analog differentiator

$H(s) = s$

=> Desired $H(r) = j\omega\big|_{\substack{\omega=2\pi f \\ f=rf_s}} = j2\pi\ f_s r$

Step2 : Calculate $h_d(nT)$

$$h_d(nT) = \int_{-1/2}^{1/2} H(r)e^{j2\pi nr}dr = \int_{-1/2}^{1/2} (j2\pi\ f_s r)e^{j2\pi nr}dr$$

$$\int_{-1/2}^{1/2} H(r)e^{j2\pi nr}dr = \int_{-1/2}^{1/2} (j2\pi\ f_s r)e^{j2\pi nr}dr$$

$$= \frac{f_s}{n}\int_{-1/2}^{1/2} j2\pi\, nre^{j2\pi nr}dr$$

$$= \frac{f_s}{n}\int_{-1/2}^{1/2} rde^{j2\pi nr}r$$

$\int_a^b udv = uv\Big|_a^b - \int_a^b vdu$

$u = r, v = e^{j2\pi nr}$

$$= \frac{f_s}{n}\left[re^{j2\pi nr}\Big|_{r=-1/2}^{r=1/2} - \int_{-1/2}^{1/2} e^{j2\pi nr}d(r)\right]$$

$$= \frac{f_s}{n}\left[re^{j2\pi nr} - \frac{1}{j2\pi n}e^{j2\pi nr}\right]_{r=-1/2}^{r=1/2}$$

$$= \frac{f_s}{n}\left[\frac{1}{2} - \frac{1}{j2\pi n}\right]e^{j\pi n} - \frac{f_s}{n}\left[-\frac{1}{2} - \frac{1}{j2\pi n}\right]e^{-j\pi n}$$

$$= \frac{f_s}{2n}\left[e^{j\pi n} + e^{-j\pi n}\right] - \frac{f_s}{j2\pi n^2}\left[e^{j\pi n} - e^{-j\pi n}\right]$$

$$= \frac{f_s}{2n}2\cos\pi n - \frac{f_s}{j2\pi n^2}\left[2j\sin\pi n\right]$$

$$h_d(nT) = \begin{cases} \dfrac{f_s}{n}(-1)^n - \dfrac{f_s 2j}{j2\pi n^2}\sin\pi n = \dfrac{f_s}{n}(-1)^n & n \neq 0 \\ \dfrac{f_s}{n} - \dfrac{f_s}{\pi n^2}\sin\pi n = \dfrac{f_s[\pi n - \sin\pi n]}{\pi n^2} & \end{cases}$$

0

$$\Rightarrow f_s \lim_{n\to 0} \frac{\frac{d}{dn}[\pi n - \sin \pi n]}{\frac{d\pi n^2}{dn}}$$

$$= \frac{f_s}{2} \lim_{n\to 0} \frac{\pi - \pi \cos \pi n}{\pi n}$$

$$= \frac{f_s}{2} \lim_{n\to 0} \frac{d(\pi - \pi \cos \pi n)/dn}{d(\pi n)/dn}$$

$$= \frac{f_s}{2} \lim_{n\to 0} \frac{\pi^2 \sin \pi n}{\pi} = 0 \qquad n = 0$$

i.e., $h_d(nT) = \begin{cases} \frac{f_s}{n}(-1)^n & n \neq 1 \\ 0 & n = 0 \end{cases}$

Step 3: Construct nc filter with hamming window (M=7)

TABLE 9-1
Filter Weights for FIR Differentiator

n	Unit Pulse Response with Rectangular Window, $h(nT)$	Hamming Window Function, $w_h(nT)$	Unit Pulse Response with Hamming Window, $w_h(nT)h(nT)$
−7	−0.142857	0.08	+0.011429
−6	−0.166667	0.125554	−0.020926
−5	+0.2	0.253195	+0.050639
−4	−0.25	0.437640	−0.109410
−3	+0.333333	0.642360	+0.214120
−2	−0.5	0.826805	−0.413403
−1	+1.0	0.954446	+0.954446
0	0	1.0	0
1	−1.0	0.954446	0.954446
2	+0.5	0.826805	+0.413403
3	−0.333333	0.642360	−0.214120
4	+0.25	0.437640	+0.109410
5	−0.2	0.253195	−0.050639
6	+0.166667	0.125554	+0.020926
7	−0.142857	0.08	−0.011429

$$H(z) = \sum_{n=-7}^{n=-1} \frac{f_s}{n}(-1)^n w_h(n) z^{-n} + \sum_{n=1}^{n=7} \frac{f_s}{n}(-1)^n w_h(n) z^{-n}$$

$$H_c(z) = z^{-7} H(z)$$

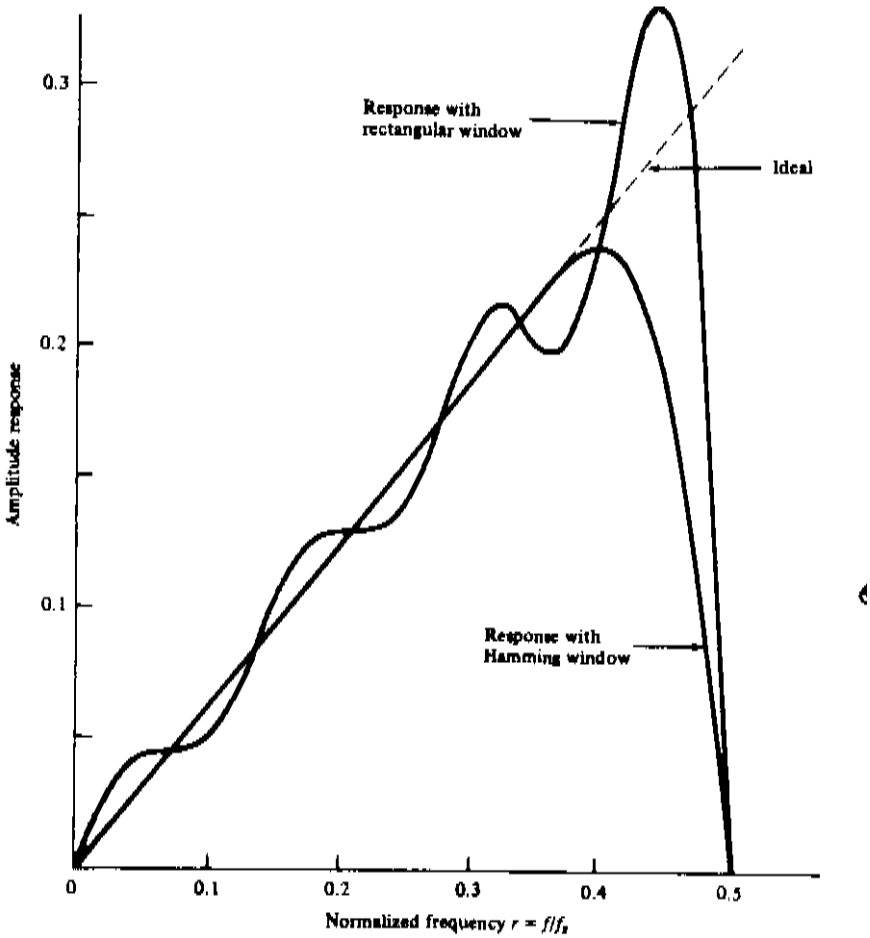


FIGURE 9-29. Amplitude response of digital differentiator.

Example 3-3: Desired low-pass FIR digital filter characteristic

$$H(r) = \begin{cases} 1 & |r| \le 0.15 \\ 0 & 0.15 < |r| \le 0.5 \end{cases}$$

$$h_d(nT) = \int_{-0.15}^{0.15} e^{j2\pi nr} dr = \frac{1}{j2\pi n}(e^{j0.3\pi} - e^{-j0.3\pi}) = \frac{1}{\pi n}\sin 0.3\pi n$$

$$h_d(0T) = \lim_{\pi n \to 0} \frac{d(\sin 0.3\pi n)/d(\pi n)}{d(\pi n)/d(\pi n)} = \lim_{\pi n \to 0} \frac{0.3\cos 0.3\pi n}{1} = 0.3$$

$$\Rightarrow \begin{cases} h_d(0) = 0.3 \\ h_d(nT) = \dfrac{\sin 0.3\pi n}{\pi n} \qquad n \neq 0 \end{cases}$$

TABLE 9-2
Filter Weights for FIR Low-Pass Filter ($f_c = 0.15 f_s$)

n	Unit Pulse Response with Rectangular Window, $h(nT)$	Hamming Window Function, $w_h(nT)$	Unit Pulse Response with Hamming Window, $w_h(nT)h(nT)$
−8	0.037841	0.08	0.003027
−7	0.014052	0.115015	0.001616
−6	−0.031183	0.214731	0.006696
−5	−0.063662	0.363966	−0.023171
−4	−0.046774	0.54	−0.025258
−3	0.032788	0.716034	0.023477
−2	0.151365	0.865269	0.130972
1	0.257518	0.964985	0.248501
0	0.3	1.0	0.3
1	0.257518	0.964985	0.248501
2	0.151365	0.865269	0.130972
3	0.032788	0.716034	0.023477
4	−0.046774	0.54	−0.025258
5	−0.063662	0.363966	−0.023171
6	−0.031183	0.214731	−0.006696
7	0.014052	0.115015	0.001616
8	0.037841	0.08	0.003027

NC filter with 17 weight's window: $H_{NC}(z) = \sum_{n=-8}^{8} h_d(nT) w_h(n) z^{-n}$, $H_C = z^{-8} H_{NC}(z)$

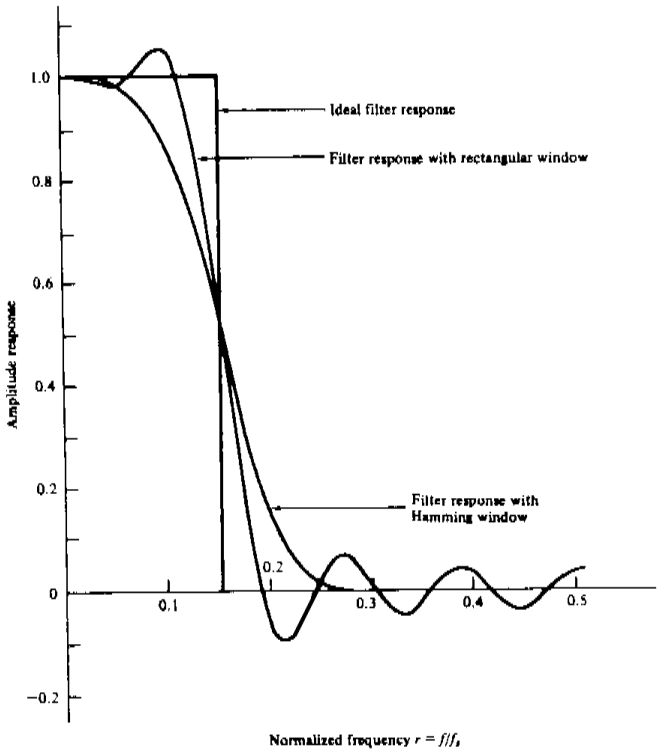


FIGURE 9-31. Amplitude response of digital low-pass filter. (The negative portions are shown negative for convenience.)

Example 3-4 (90^o phase shifter)

$$H(r)=\begin{cases}-j & 0<r<0.5\\ j & -0.5<r<0\end{cases}$$

$$h_d(nT)=\int_{-1/2}^{0} je^{j2\pi nr}dr+\int_{0}^{1/2}(-j)e^{j2\pi nr}dr$$

$$=\frac{1}{2\pi n}(1-e^{-jn\pi})-\frac{1}{2\pi n}(e^{jn\pi}-1)$$

$$=-\frac{1}{2\pi n}(e^{jn\pi}+e^{-jn\pi})+\frac{1}{\pi n}=-\frac{\cos\pi n}{\pi n}+\frac{1}{\pi n}$$

$$n=0 \Rightarrow\ h_d(0)=\lim_{\pi n\to 0}\frac{1-\cos\pi n}{\pi n}=\lim_{\pi n\to 0}\frac{\sin\pi n}{1}=0$$

$$\Rightarrow\ \begin{cases}h_d(0)=0\\ h_d(nT)=\dfrac{1}{\pi n}[1-(-1)^n] & n\neq 0\end{cases}$$

$$=\begin{cases}\dfrac{2}{\pi n} & n=odd\\ 0 & n\neq 0,\quad n=even\end{cases}$$

$$\Rightarrow h_d(nT)=\begin{cases}\dfrac{2}{\pi n} & n=odd\\ 0 & n=even\end{cases}$$

Filter: $H_{NC}(z)=\sum_{n=-7}^{7}h_d(n)w_h(n)z^{-n}$ $\qquad M=7$

$$H_C=z^{-7}H_{NC}(z)$$

TABLE 9-3
Filter Weights for FIR 90° Phase Shifter

n	Unit Pulse Response with Rectangular Window, $h(nT)$	Hamming Window Function, $w_h(nT)$	Unit Pulse Response with Hamming Window, $w_h(nT)h(nT)$
−7	−0.090946	0.08	−0.007276
−6	0	0.125554	0
−5	−0.127324	0.253195	−0.032238
−4	0	0.437640	0
−3	−0.212207	0.642360	−0.136313
−2	0	0.826805	0
−1	−0.636620	0.954446	0.607619
0	0	1.0	0
1	0.636620	0.954446	0.607619
2	0	0.826805	0
3	0.212207	0.642360	0.136313
4	0	0.437640	0
5	0.127324	0.253195	0.032238
6	0	0.125554	0
7	0.090946	0.08	0.007276

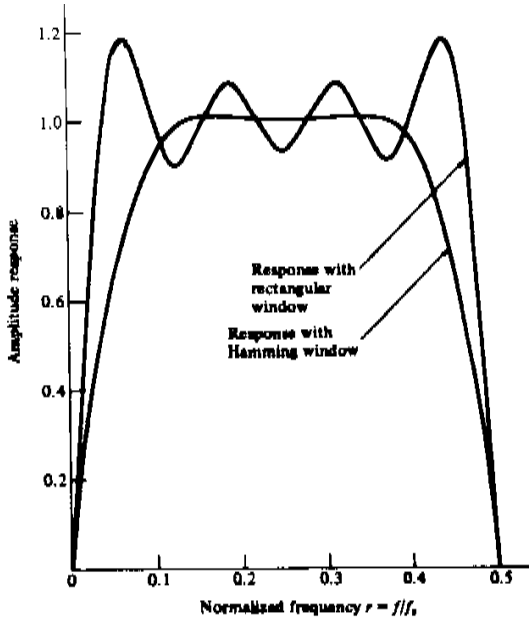


Fig. Amplitude response of digital 90 degree phase shifter

UNIT 4

FINITE WORD LENGTH EFFECTS

- The digital signal processing algorithms is realized either with special purpose hardware or as programs for general purpose digital computer.
- In both cases, the numbers and coefficients are stored in finite length registers.
- Coefficients and numbers are quantized by truncation and rounding off when they are stored.
- Errors due to quantization
  - Input quantization error (in A/D conversion process)
  - Product quantization error (in multiplier)
  - Coefficient quantization error (in filter design)

4.1 Number representation:

A number N is represented by finite series as

$$N = \sum_{i=n_1}^{n} c_i r^i$$

r = 10 for decimal representation

$$30.285 = \sum_{i=3}^{1} c_i 10^i$$

$$= 3 \times 10^1 + 0 \times 10^0 + 2 \times 10^{-1} + 8 \times 10^{-2} + 5 \times 10^{-3}$$

r = 2 for binary representation

$$110.010 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3}$$

$$= (6.25)_{10}$$

1. Convert the decimal number 30.275 to binary form.

Integer part					Remainder		Fractional part						Integer part	Binary number
30	/	2	=	15	0	↑	0.275	x	2	=	0.550		0	↓
15	/	2	=	7	1		0.55	x	2	=	1.10		1	
7	/	2	=	3	1		0.1	x	2	=	0.2		0	
3	/	2	=	1	1		0.2	x	2	=	0.4		0	
1	/	2	=	0	1		0.4	x	2	=	0.8		0	
					Binary number		0.8	x	2	=	1.6		1	
							0.6	x	2	=	1.2		1	
							0.2	x	2	=	0.4		0	

$$(30.275)_{10} = (11110.01000110)_{10}$$

Types of number of representation.

- Fixed point representation
- Floating point representation
- Block floating point representation

4.1.1 Fixed point representation:

- The position of the binary point is fixed
- The negative numbers are represented in Sign magnitude form
  - One's complement form
  - Two's complement form

Sign magnitude form:

Sign	Magnitude

- MSB is set to 1 to represent the negative sign
- Zero has two representations
- With 'b' bits only 2^b-1 numbers can be represented
- Eg. $(1.75)_{10} = (01.110000)_2$
  $(-1.75)_{10} = (11.110000)_2$

One's complement form:

- Positive numbers are represented as in sign magnitude form
- Negative number is represented by complementing all the bits of the positive number
- Eg. $(0.875)_{10} = (0.111000)_2$
  $(-0.875)_{10} = (1.000111)_2$ – one's complement form
- The same is obtained by subtracting the magnitude from $2-2^{-b}$, b is the number of bits (without sign bit)
- In the above example b=6 therefore $2-2^{-6} = 10.000000 - 0.000001 = 1.111111$
- Now for $(-0.875)_{10}$ 1.111111
  -0.111000
  =1.000111 (one's complement)
- Also the magnitude for the negative number is given by

$$1 - \sum_{i=1}^{b} c_i 2^{-i} - 2^b$$

- For $(-0.875)_{10} = (1.000111)_2$
  $= 1 - (0 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} + 1 \times 2^{-5} + 1 \times 2^{-6}) - 2^{-6}$
  $= 1 - (2^{-4} + 2^{-5} + 2^{-6}) - 2^{-6}$
  $= (0.875)_{10}$

Two'scomplement form:.

- Positive numbers are represented as in sign magnitude form
- Negative number is represented in two's complement form of the positive number
- Eg. $(0.875)_{10} = (0.111000)_2$
  $(-0.875)_{10} = (1.000111)_2$ – one's complement form
  $= +0.000001 \quad +1$
  $= (1.001000)_2$ – two's complement form
- The same is obtained by subtracting the magnitude from 2
- Now for $(-0.875)_{10}$ 10.000000 (2)
  -0.111000 (+0.875)

= 1.001000 (-0.875 in two's complement form)

- The magnitude for the negative number is given by

$$1 - \sum_{i=1}^{b} c_i 2^{-i}$$

- For $(-0.875)_{10} = (1.001000)_2$

$= 1 - (0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 0 \times 2^{-5} + 0 \times 2^{-6})$

$= 1 - 2^{-3}$

$= (0.875)_{10}$

Addition of two fixed point numbers:

- The two numbers are added bit by b it starting from right, with carry being added to the next bit.
- Eg., $(0.5)_{10} + (0.125)_{10}$

$(0.5)_{10} = 0.100$

$+(0.125)_{10} = +0.001$

$= 0.101 = (0.625)_{10}$

- When two number of 'b' bits are added and the sum cannot be represented by 'b' bits an overflow is said to occur.
- Eg., $(0.5)_{10} + (0.625)_{10}$

$(0.5)_{10} = 0.100$

$+(0.625)_{10} = +0.101$

$=1.125 =$ 1.001 But which is $(-0.125)_{10}$ in sign magnitude

- In general, the addition of fixed point numbers causes an overflow

Subtraction of two fixed point numbers:

- $(0.5)_{10} - (0.25)_{10}$

$(0.5)_{10} = 0.100$

$-(0.25)_{10} = 0.010 = 1.101$ (1'S) + 0.001 = +1.110(2'S)

= 10.010

= 0.010 (neglect carry) = $(0.25)_{10}$

- $(0.25)_{10} - (0.5)_{10}$

$(0.25)_{10} = 0.010$

$-(0.5)_{10} = 0.100 = 1.011$ (1'S) + 0.001 = +1.100 (2'S)

= 1.110(No carry, result is negative take 2's complement)

= 0.001(1's complement)

= +0.001 (+1)

= $0.010(-0.25)_{10}$

Multiplication in fixed point:

- Sign and magnitude components are separated
- The magnitude of the numbers are multiplied then sign of the product is determined and applied to result
- With 'b' bit multiplicand and 'b' bit multiplier the product may contain 2b bits
- If $b = b_i + b_f$, where b_i represents integer part and b_f represents the fraction part then the product may contain $2b_i + 2b_f$ bits
- Multiplication of the two fraction results in a fraction and overflow can never occur

4.1.2 Floating point representation:

- The number is represented as $F = 2^{c} \cdot M$. where M called mantissa is a fraction such that ½ ≤M≤1 and 'c' called exponent can be either positive or negative
- Eg., $(4.5)_{10} = (100.1)_2 = 2^{011} \times 0.1001$
  $(1.5)_{10} = (1.1)_2 = 2^{001} \times 0.1100$
  $(6.5)_{10} = (110.1)_2 = 2^{011} \times 0.1101$
  $04.625)_{10} = (0.1010)_2 = 2^{000} \times 0.1010$
- For negative numbers the sign of the floating point number is obtained from the first bit of mantissa

Multiplication:

- If $F_1 = 2^{c1} \cdot M_1$ and $F_2 = 2^{c2} \cdot M_2$ then $F_3 = F_1 \times F_2 = (M_1 \times M_2) \cdot 2^{(c1+c2)}$

- Eg., $(1.5)_{10} = 2^{001} \times 0.1100$
  $(1.25)_{10} = 2^{001} \times 0.1010$
  $(1.5)_{10} \times (1.25)_{10} = 2^{001} \times 0.1100 \times 2^{001} \times 0.1010$
  $= 2^{(001+001)} \times (0.1100 \times 0.1010)$
  $= 2^{010} \times 0.01111$
- Addition and subtraction of two floating point numbers are more difficult than addition and subtraction of two fixed point numbers
- To carry out addition, first adjust the exponent of the smaller number until it matches with the exponent of the larger number
- The mantissa are then added or subtracted

- Eg., $(3)_{10} + (0.125)_{10}$
  $(3)_{10} = 2^{010} \times 0.110000$
  $(0.125)_{10} = 2^{000} \times 0.001000 = 2^{010} \times 0.000010$ (adjust the exponent of smaller number)
  $(3)_{10} + (0.125)_{10} = 2^{010} (0.110000 + 0.000010)$
  $= 2^{010} \times 0.110010$

Comparison of fixed point and floating point arithmetic:

Fixed point arithmetic	Floating point arithmetic
Fast operation	Slow operation
Relatively economical	More expensive because of hardware
Small dynamic range	Increased dynamic range
Roundoff errors occur only for addition	Roundoff errors can occur with both addition and multiplication
Overflow occurs in addition	Overflow dost not arise
Used in small computers	Used in larger, general purpose computers

4.1.3 Block floating point numbers:

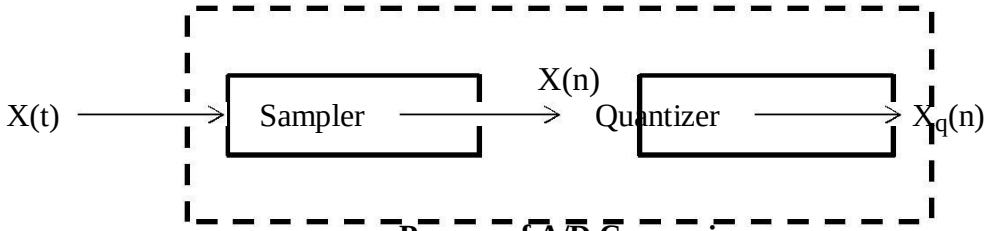
- A compromise between fixed and floating point systems is the block floating point

arithmetic

- The set of signals to be handled is divided into blocks
- Each block have the same value of exponent
- The arithmetic operations within the block uses fixed point arithmetic and only one exponent per block is stored, thus saving memory
- Suitable for FFT flow graphs and digital audio applications

4.2 Quantization noise:

- For most of the engineering applications the input signal is continuous in time or analog waveform.
- This signal is to be converted into digital by using ADC



Process of A/D Conversion

- First the signal x(t) is sampled at regular intervals t=nT where n=0,1,2….. to create a sequence x(n). This is done by sampler.
- Then numeric equivalent of each sample is expressed by a finite number of bits giving the sequence $x_q(n)$.
- The difference signal $e(n) = x_q(n)-x(n)$ is called quantization error or A/D conversion noise
- Assume a sinusoidal signal varying between +1 and -1 having dynamic range 2.
- If ADC is used to convert the sinusoidal signal it employs (b+1) bits including sign bit. Then the number of levels available for quantizing x(n) is 2^{b+1}.
- Thus the interval between successive levels is $q = \frac{2}{2^{b+1}} = 2^{-b}$ where q is known as quantization step size.
- The common methods of quantization are
  - Truncation
  - Rounding

4.3 Truncation:

It is a process of discarding all bits less significant than LSB that is retained.

e.g. 0.00110011 = 0.0011 (8 bits to 4 bits)
1.01001001 = 1.0100

4.4 Rounding:

Rounding of a number of *b* bits is accomplished by choosing the rounded result as the *b* bit number closest to the original number unrounded.

e.g. 0.11010 = 0.110 or 0.111
0.110111111 = 0.11011111 or 0.11100000

- Rounding up or down will have negligible effect on accuracy of computation

4.5 Error due to truncation and rounding:

If the quantization method is truncation, the number is appropriated by the nearest level that does not exceed it. In this case, the error x_T-x is negative or zero where x_T is truncation value of x.

The error made by truncating a number to *b* bits following the binary point satisfies the inequality

$$0 \geq x_T - x > -2^{-b} \qquad \text{----------1}$$

e.g. $(0.12890625)_{10} = (0.00100001)_2$

$x_T = (0.0010)_2 =$

Truncate to 4 bits $(0.125)_{10}$

Now the error $(x_T-x) = -0.00390625$ which is $> -2^{-b} = -2^{-4} = -0.0625$ satisfy the inequality

- Equation 1 holds good for sign magnitude, one's complement and two's complement if x>0

By considering two's complement representation the magnitude of the negative number is

$$x = 1 - \sum_{i=1}^{b} c_i \, 2^{-i}$$

If we truncate to N bits then

$$x_T = 1 - \sum c_i \, 2^{-i} \quad {}_{i=1}$$

The change in magnitude

$$x_T - x = 1 - \sum_{i=N}^{b} c_i \, 2^{-i}$$

$$\geq 0$$

- Therefore, due to truncation the change in the magnitude is positive, which implied that error is negative and satisfy the in equality $0 \geq x_T - x > -2^{-b}$

For one's complement representation the magnitude of the negative number with b bits is given by

$$x = 1 - \sum_{i=1}^{b} c_i \, 2^{-i} - 2^{-b}$$

When the number is truncated to N bits, then

$$x_T = 1 - \sum c_i \, 2^{-i} - 2^{-N} \quad {}_{i=1}$$

The change in magnitude due to truncation is

$$x_T - x = 1 - \sum_{i=1}^{N} c_i \, 2^{-i} - (2^{-N} - 2^{-b})$$

$$< 0$$

- Therefore,the. magnitude decreases with truncation which implies the error is positive

and satisfy the in equality $0 \le x_T - x < 2^{-b}$

- The above equation holds for sign magnitude representation also

In floating point systems the effect of truncation is visible only in the mantissa

If $x = 2^c M$ then $x_T = 2^c M_T$

Error $e = x_T - x = 2^c(M_T - M)$

- With two's complement representation of mantissa we have

$$0 \ge M_T - X > -2^{-b}$$

$$0 \ge e > -2^{-b}2^c \text{ ---------- 2}$$

- Let's define relative error $\varepsilon = \frac{x_T - x}{x} = \frac{e}{x}$
- The equation 2 becomes

$$0 \ge \varepsilon x > -2^{-b}2^c$$

Or $0 \ge \varepsilon 2^c M > -2^{-b}2^c$

Or $0 \ge \varepsilon M > -2^{-b}$

- If M = 1/2 the relative error is maximum. Therefore, $0 \ge \varepsilon > -2 \cdot 2^{-b}$
- If M = -1/2 the relative error range is $0 \le \varepsilon < 2 \cdot 2^{-b}$

In one's complement representation, the error for truncation of the values of the mantissa is

$$0 \ge M_T - X > -2^{-b}$$

$$0 \ge e > -2^{-b}2^c$$

With $e = \varepsilon x = \varepsilon 2^c M$ and M=1/2 we get the maximum range of the relative error for positive mantissa is

$$0 \ge \varepsilon > -2 \cdot 2^{-b}$$

For negative mantissa, value of error is

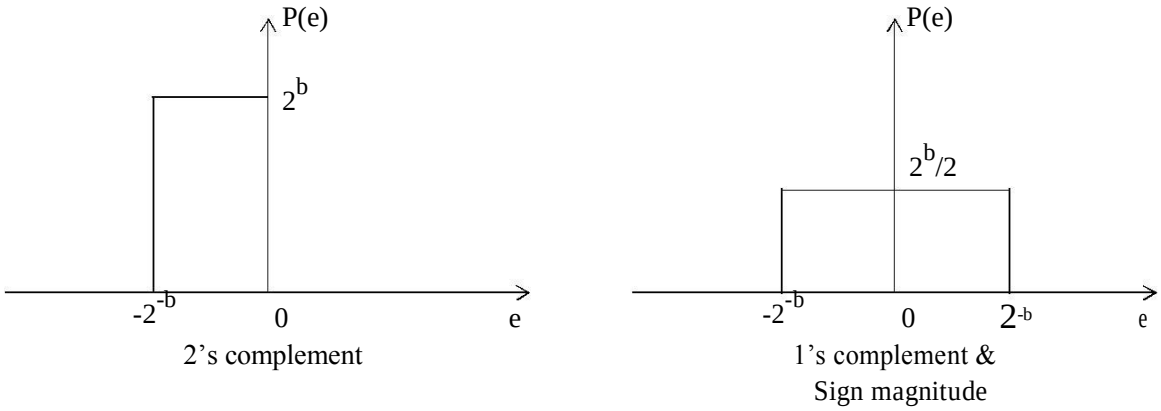
$$0 \le M_T - X < 2^{-b}$$

Or $0 \le e < 2^c 2^{-b}$

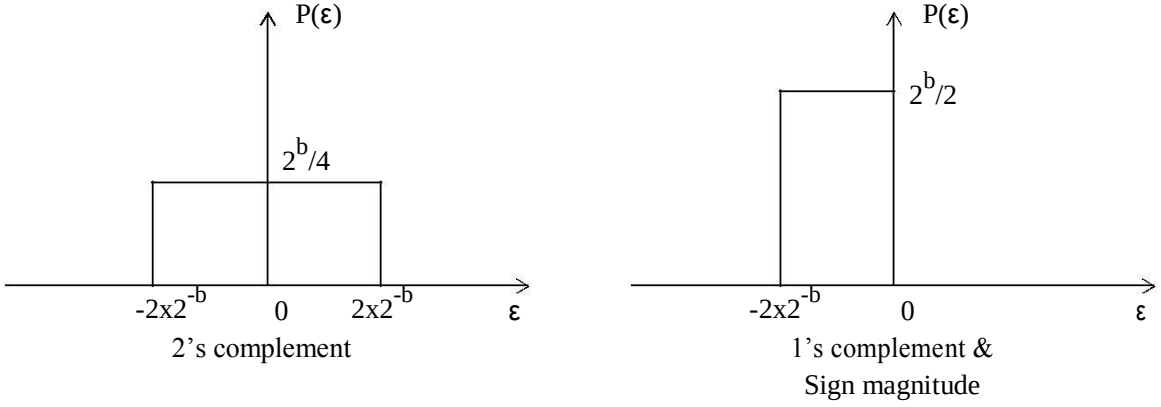
With M=-1/2 the maximum range of the relative error negative mantissa is $0 \ge \varepsilon > -2 \cdot 2^{-b}$, which is same as positive mantissa.

The probability density function for p(e) for truncation of fixed point and floating point numbers are

Fixed Point



Floating Point



In fixed point arithmetic the error due to rounding a number to 'b' bits produces an error $e=x_R-x$ which satisfies the in equality

$$\frac{-2^{-b}}{2} \le x_R - x \le \frac{2^{-b}}{2} \quad \text{---------- 3}$$

This is because with rounding, if the value lies half way between two levels, it can be approximated either nearest higher level or nearest lower level. The above equation satisfied regardless of whether sign magnitude, 1's complement or 2's complement is used for negative numbers.

In floating point arithmetic, only mantissa is affected by quantization

If $x = 2^C M$ and $x_R = 2^C M_R$ then error $e = x_R - x = 2^C(M_R - M)$. But for rounding

$$-\frac{2^{-b}}{2} \le M_R - M \le \frac{2^{-b}}{2}$$

$$-2^c \frac{2^{-b}}{2} \le x_R - x \le 2^c \frac{2^{-b}}{2}$$

$$-2^c \frac{2^{-b}}{2} \le \varepsilon x \le 2^c \frac{2^{-b}}{2}$$

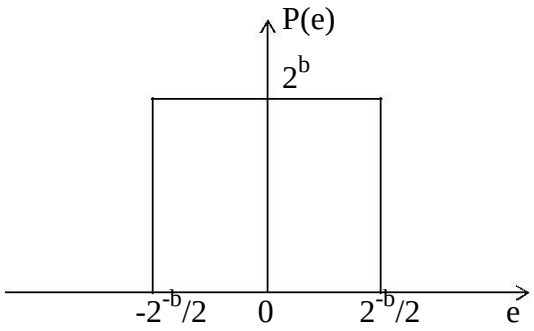
$$-2^c \frac{2^{-b}}{2} \le \varepsilon 2^c M \le 2^c \frac{2^{-b}}{2}$$

$$-\frac{2^{-b}}{2} \le \varepsilon M \le \frac{2^{-b}}{2}$$

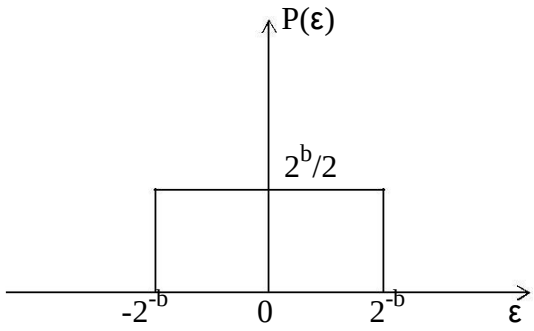
The mantissa satisfy ½<M<1. If M=1/2 we get the maximum range of relative error

$$-2^{-b} \le \varepsilon \le 2^{-b}$$

The probability density function for rounding is as follows



Fixed point



Floating point

Input quantization error:

- The quantization error arises when a continuous signal is converted into digital value.
- The quantization error is given by

$$e(n) = x_q(n) - x(n)$$

where $x_q(n)$ = sampled quantized

value x(n) = sampled
unquantized value

- Depending upon the way in which x(n) is quantized the distributions of quantization noise will differ. If rounding of a number is used to get $x_q(n)$ then the error signal satisfies the relation

$$-q/2 \leq e(n) \leq q/2$$

- Because the quantized signal may be greater or less than actual signal

let $x(n) = (0.7)_{10}$ =

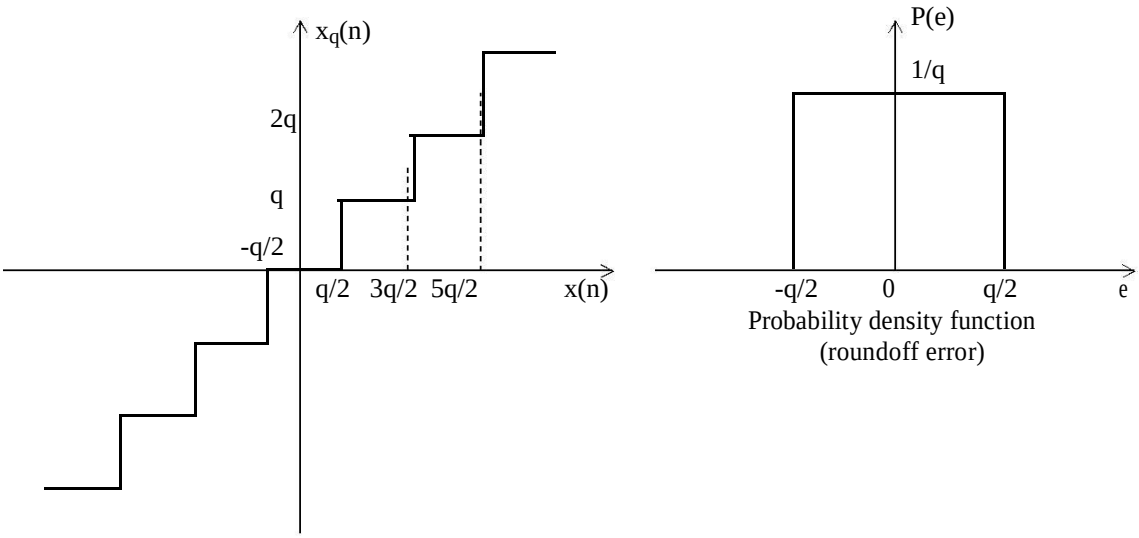
- Eg., $(0.10110011\ldots)_2$

After rounding x(n) to 3 bits

$x_q(n) = (0.110)_2$ =

$(0.75)_{10}$

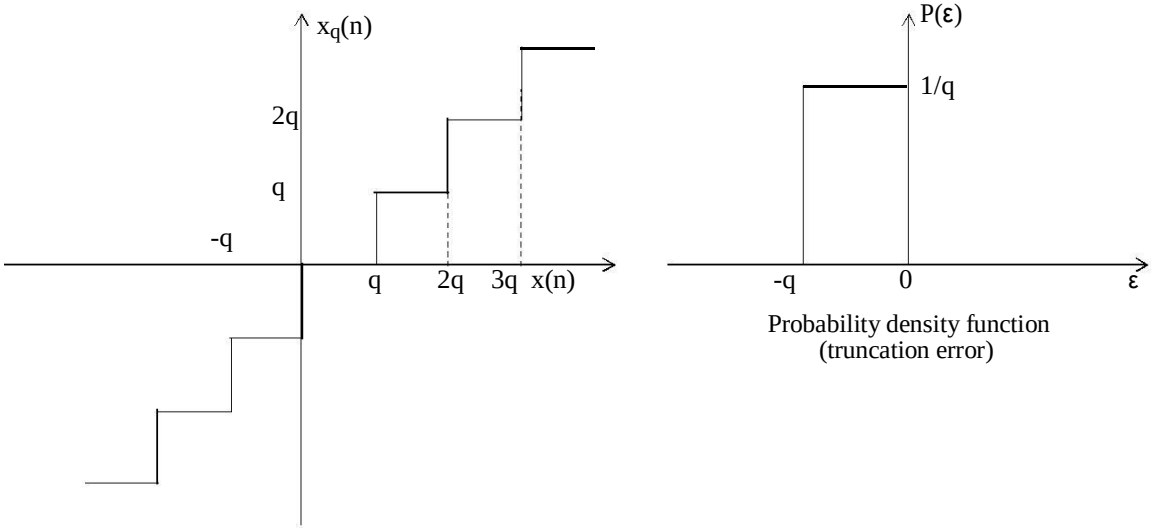
- Now the error e(n) = 0.75 – 0.7 = 0.05 which satisfies the inequality



Quantizer characteristics (rounding)

- The other type of quantization can be obtained by truncation. In truncation, the signal is represented by the highest quantization level that is not greater than the signal
- In two's complement truncation, the error e(n) is always negative and satisfied the inequality

$$-q \leq e(n) < 0$$



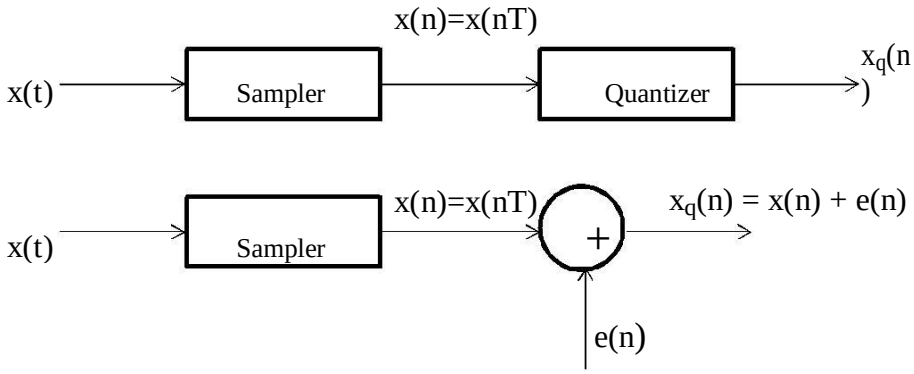
Quantizer characteristics (2's complement truncation)

- The quantization error mean value is zero for rounding and –q/2 for 2's complement truncation

Digital **of** processing of analog signals the quantization error is commonly viewed as an additive noise signal

i.e. $x_q(n) = x(n) + e(n)$

Quantization noise model



- Therefore, the A/D converter output is the sum of the input signal x(n) and the error signal e(n)

If the rounding is used for quantization then the quantization error $e(n)=x_q(n) - x(n)$ is bounded by $-q/2 \le e(n) \le q/2$. In most cases, assume that A/D conversion error e(n) has the following properties

- The error sequence e(n) is a sample sequence of a stationary random process
- The error sequence is uncorrelated with x(n) and other signals in the system
- The error is a white noise process with uniform amplitude probability distribution over the range of quantization error

In case of rounding the e(n) lies between –q/2 and q/2 with equal probability. The variance of e(n) is given by

$$\sigma_e^2 = E[e^2(n)] - E^2[e(n)]$$

Where $E[e^2(n)]$ is the average of $e^2(n)$

$E[e(n)]$ is mean value of e(n)

$$\sigma_e^2 = \int_{-\infty}^{\infty} e^2(n)\, p(e)de - (0)^2$$

$$\sigma_e^2 = \int_{-\frac{q}{2}}^{\frac{q}{2}} e^2(n) \cdot \frac{1}{q}\, de$$

$$\sigma_e^2 = \frac{1}{q}\int_{-\frac{q}{2}}^{\frac{q}{2}} e^2(n) \cdot de$$

$$\sigma_e^2 = \frac{1}{q}\left(\frac{e^3(n)}{3}\right)_{-\frac{q}{2}}^{\frac{q}{2}}$$

$$\sigma_e^2 = \frac{1}{q}\left(\frac{q^3}{24} + \frac{q^3}{24}\right)$$

$$\sigma_e^2 = \frac{2q^3}{24q} = \frac{q^2}{12}$$

Sub $q = 2^{-b}$

$$\sigma_e^2 = \frac{\left(2^{-b}\right)^2}{12}$$

$$\sigma_e^2 = \frac{2^{-2b}}{12}$$

In case of two's complement truncation the e(n) lies between 0 and –q having mean value of –q/2. The variance or power of the error signal e(n) is given by

$$\sigma_e^2 = \int_{-q}^{0} e^2(n)\, p(e)de - \left(-\frac{q}{2}\right)^2$$

$$\sigma_e^2 = \int_{-q}^{0} e^2(n) \cdot \frac{1}{q}\, de - \frac{q^2}{2}$$

$$\sigma_e^2 = \frac{1}{q}\left(\frac{e^3(n)}{3}\right)_{-q}^{0} - \frac{q^2}{2}$$

$$\sigma_e^2 = \frac{1}{q}\int_{-q/2}^{q/2} e^2 de = \frac{1}{q}\left[\frac{e^3}{3}\right]_{-q/2}^{q/2}$$

$$\sigma_e^2 = \frac{4q^2 - 3q^2}{12} = \frac{q^2}{12}$$

$subq = 2^{-b}$

$$\sigma_e^2 = \frac{\left(2^{-b}\right)^2}{12}$$

$$\sigma_e^2 = \frac{2^{-2b}}{12}$$

- In both cases the value of $\sigma_e^2 = \frac{2^{-2b}}{12}$, which is also known as the steady state noise power due to input quantization.

- If the input signal is x(n) and its variance is σ_x^2 then the ratio of signal power to noise power which is known as signal to noise ratio for rounding is

$$\frac{\sigma_x^2}{\sigma_e^2} = \frac{\sigma_x^2}{2^{-2b}/12} = 12(2^{2b}\sigma_x^2)$$

When expressed in a log scale SNR in dB

$$= 10\log_{10}\frac{\sigma_x^2}{\sigma_e^2}$$

$$= 10\log_{10} 12(2^{2b}\sigma_x^2)$$

$$= 10\log_{10} 12 + 10\log_{10} 2^{2b} + 10\log_{10}\sigma_x^2)$$

$$= 10.73 + 10\,x2b\,x\log_{10} 2 + 10\log_{10}\sigma_x^2)$$

$$= 10.79 + 6.02b + 10\log_{10}\sigma_x^2$$

- From the above equation it is known that the SNR increases approximately 6dB for each bit added to the register length
- If the input signal is Ax(n) instead of x(n) where 0 < A < 1, then the variance is $A^2\sigma_x^2$. Hence

$$\text{SNR} = 10\log_{10}\frac{A^2\sigma_x^2}{\sigma_e^2}$$

$$= 10.8 + 6b + 10\log_{10}\sigma_x^2 + 20\log_{10} A$$

If $A = \frac{1}{4}$ then

σ_x

$$\text{SNR} = 10.8 + 6b + 10\log_{10}\sigma_x^2 + 20\log_{10} A$$
$$= 10.8 + 6b + 10\log_{10}\sigma_x^2 + 10\log_{10} A^2$$
$$= 10.8 + 6b + 10\log_{10}\sigma_x^2 + 10\log_{10}\frac{1}{16\sigma_x^2}$$
$$= 10.8 + 6b + 10\log_{10}\sigma_x^2 + 10\log_{10} 2^{-4}\sigma_x^{-2}$$
$$= 10.8 + 6b + 10\log_{10}\sigma_x^2 + 10\log_{10} 2^{-4} + 10\log_{10}\sigma_x^{-2}$$
$$= 6b - 1.24 dB$$

- Thus to obtain SNR ≥ 80dB required b = 14 bits.

4.8 LIMIT CYCLE OSCILLATIONS

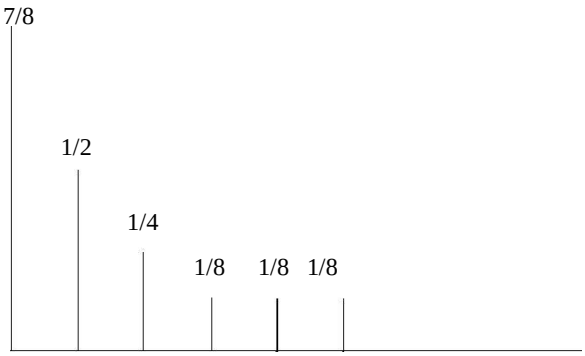
4.8.1 Zero input limit cycle oscillations:

When a stable IIR filter is excited by a finite input sequence, that is constant, the output will ideally decay to zero. However, the non-linearities due to the finite-precision arithmetic operations often cause periodic oscillations in the recursive systems are called zero input limit cycle oscillations.

Consider a first order IIR filter with difference equation $y(n) = x(n) + \alpha y(n-1)$

Let's assume $\alpha = ½$ and the data register length is 3 bits plus a sign bit. If the input is $x(n) = \begin{cases} 0.875 & \text{for } n = 0, \\ 0 & \text{otherwise}. \end{cases}$ and rounding is applied after the arithmetic operation. Here Q[·] represents the rounding operations.

n	*x(n)*	*y(n-1)*	$\alpha y(n-1)$	$Q[\alpha y(n-1)]$	$y(n) = x(n) + Q[\alpha y(n-1)]$
0	0.875	0	0	0.000	7/8
1	0	7/8	7/16	0.100	1/2
2	0	1/2	1/4	0.010	1/4
3	0	1/4	1/8	0.001	1/8
4	0	1/8	1/16	0.001	1/8
5	0	1/8	1/16	0.001	1/8



From the above table it is found that for n ≥ 3 the output remains constant and gives 1/8 as steady output causing limit cycle behavior.

Round a value in the above table:

7/16 = 0.4375

0.4375 x 2 = 0.875
0.875 x 2 = 1.75
0..75 x.
2 = 1.5
0.5 x 2 = 1.1
$(0.4375)_{10}$ = $(0.0111)_2$

After rounding to 3 bits
= $(0.100)_2$
= $(0.5)_{10}$

Let's assume α = -1/2

n	*x(n)*	*y(n-1)*	*α y(n-1)*	*Q[α y(n-1)]*	*y(n) = x(n) + Q[αy(n −1)]*
0	0.875	0	0	0.000	7/8
1	0	7/8	-7/16	1.100	-1/2
2	0	-1/2	1/4	0.010	1/4
3	0	1/4	-1/8	1.001	-1/8
4	0	-1/8	1/16	0.001	1/8
5	0	1/8	-1/16	1.001	-1/8
6	0	-1/8	1/16	0.001	1/8

When α = -1/2 the output oscillates between 0.125 to -0.125

Dead band:

- The limit cycle occur as a result of quantization effects in the multiplications
- The amplitudes of the output during a limit cycle are confined to a range of values that is called the dead band of the filter

Let's consider a single pole IIR system whose difference equation is given by

$$y(n) = \alpha y(n-1) + x(n) \qquad n > 0$$

After rounding the product term

$$y_q(n) = Q[\alpha y(n-1)] + x(n)$$

During the limit cycle oscillations

$$Q[\alpha y(n-1)] = y(n-1) \quad for \alpha > 0$$
$$= -y(n-1) \quad for \alpha < 0$$

By definition of roundin

$$|Q[\alpha y(n-1)] - \alpha y(n-1)| \le 2^{-b} 2$$

$$|y(n-1) - \alpha y(n-1)| \le 2^{-b} 2$$

$$|y(n-1)[1-\alpha]| \le 2^{-b} 2$$

$$2^{-b} / 2\, y(n-1) \le \overline{1-|\alpha|}$$

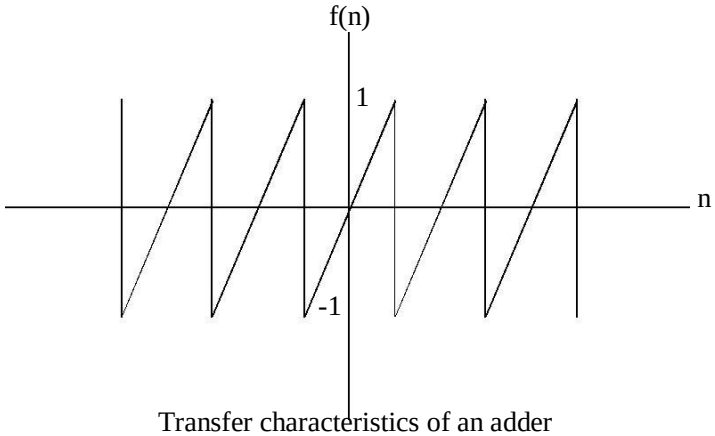
The above equation defines the dead band for the given first order filter.

4.8.2 *Overflow limit cycle oscillations:*

- In addition to limit cycle oscillations caused by rounding the result of multiplications, there are several types of oscillations caused by addition, which makes the filter output oscillates between maximum and minimum amplitudes such limit cycles have referred to as overflow oscillations.
- An overflow in addition of two or more binary numbers occurs when the sum exceeds the word size available in the digital implementation of the system.
- Let's consider two positive number n_1 and n_2

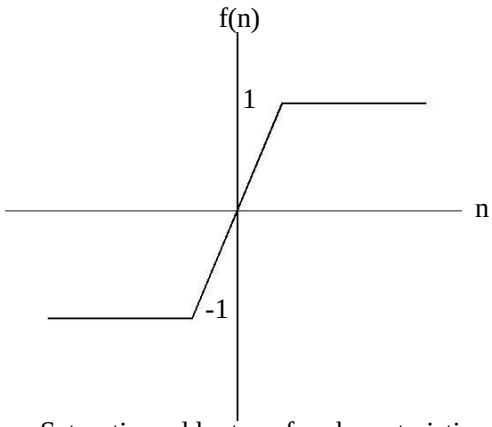
n_1	=	$(7/8)_{10}$	=	$(0.111)_2$	
n_2	=	$(6/8)_{10}$	=	$(0.110)_2$	
$n_1 + n_2$ =				$(1.101)_2$	(-5/8 in sign magnitude, but actual total is 13/8)

- In the above example, when two positive numbers are added the sum is wrongly interrupted as a negative number



Transfer characteristics of an adder

- The overflow occurs if the total input is out of range, this problem can be eliminated by modifying adder characteristics

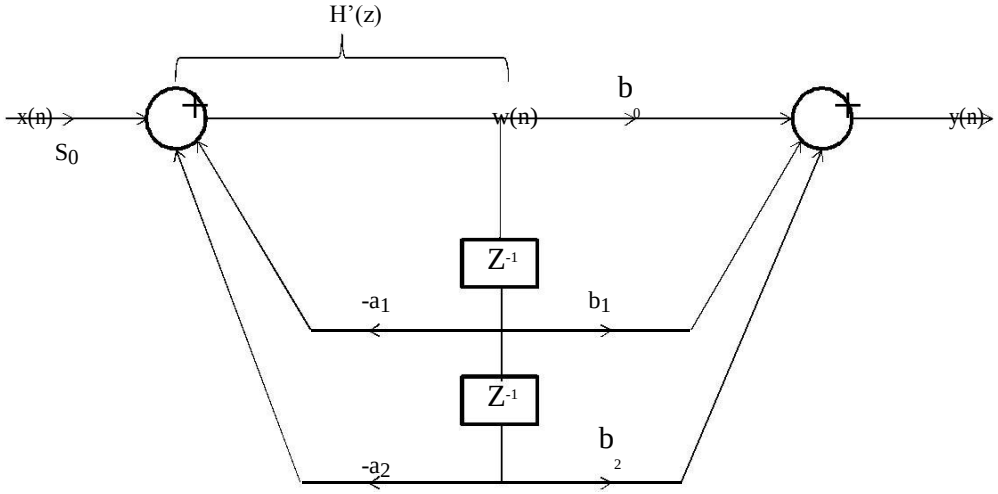


Saturation adder transfer characteristics

- When an overflow is detected, the sum of adder is set equal to the maximum value

4.9 **SIGNAL SCALING**

- The saturation arithmetic eliminates limit cycles due to overflow, but it causes undesirable signal distortion due to the non linearity of the clipper.
- In order to limit the amount of non-linear distortion, it is important to scale the input signal and the unit sample response between the input and any internal summing node in the system such that overflow becomes a rare event.
- Let's consider a second order IIR filter. A scale factor S_0 is introduced between the input x(n) and the adder 1, to prevent overflow at the output of adder 1.



Realization of second order IIR filter

- Now the overall input output transfer function is

$$H(z) = S_0 \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

$$= S_0 \frac{N(z)}{D(z)}$$

$$H'(z) = \frac{W(z)}{X(z)} = \frac{S_0}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{S_0}{D(z)}$$

- If the instantaneous energy in the output sequence w(n) is less than the finite energy in the input sequence then, there will not be any overflow

$$W(z) = \frac{S_0 X(z)}{D(z)} = S_0 S(z)D(z) \qquad where S(z) = \frac{1}{D(z)}$$

We have

$$w(n) = \frac{S_0}{2\pi} \int S(e^{j\theta}) X(e^{j\theta}) (e^{jn\theta}) d\theta$$

Which gives

$$w^2(n) = \frac{S_0^2}{4\pi^2} \left| \int S(e^{j\theta}) X(e^{j\theta}) (e^{jn\theta}) d\theta \right|^2$$

Using Schwartz in equality

$$w^2(n) \le S_0^2 \frac{1}{2\pi} \int_{2\pi} \left| S(e^{j\theta}) \right|^2 d\theta \frac{1}{2\pi} \int_{2\pi} \left| X(e^{j\theta}) \right|^2 d\theta$$

Applying Parsavel's theorem

$$w^2(n) \le S_0^2 \sum_{n=0}^{\infty} x^2(n) \frac{1}{2\pi} \int_{2\pi} \left| S(e^{j\theta}) \right|^2 d\theta \quad 1$$

We know

$$z = e^{j\theta}$$

Differentiate with respect to θ

$$\frac{dz}{d\theta} = je^{j\theta}$$

$$dz = je^{j\theta} d\theta$$

$$d\theta = \frac{dz}{je^{j\theta}} = \frac{dz}{jz} \quad \text{----------2}$$

Substitute equation 2 in equation 1

$$w^2(n) \le S_0^2 \sum_{n=0}^{\infty} x^2(n) \frac{1}{2\pi j} \int_c |S(z)|^2 z^{-1} dz$$

$$\le S_0^2 \sum_{n=0}^{\infty} x^2(n) \frac{1}{2\pi j} \int_c S(z)S(z^{-1})z^{-1} dz$$

$$w^2(n) \le \sum_{n=0}^{\infty} x^2(n) \quad \text{when } S_0^2 \frac{1}{2\pi j} \int_c S(z)S(z^{-1})z^{-1} dz = 1$$

Therefore, $S_0^2 = \dfrac{1}{\frac{1}{2\pi j} \int_c S(z)S(z^{-1})z^{-1} dz}$

$$= \frac{1}{\frac{1}{2\pi j} \int_c \frac{z^{-1} dz}{D(z)D(z^{-1})}}$$

$$= \frac{1}{I}$$

Where $I = \dfrac{1}{2\pi j} \int_c \dfrac{z^{-1} dz}{D(z)D(z^{-1})}$

UNIT 5
DSP APPLICATIONS

5.1 MULTIRATE SIGNAL PROCESSING

There is a requirement to process the various signals at different sampling rate e.g., Teletype, Facsimile, speech and video, etc., The discrete time systems that process data at more than one sampling rate are known as multirate systems.

Example:

- High quality data acquisition and storage
- Audio and video signal processing
- Speech processing
- Narrow band filtering for ECG/EEG
- Transmultiplexers

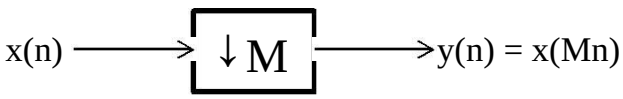
Sampling rate conversion can be done in i) analog domain and ii) digital domain. In analog domain using DAC the signal is converted into analog and then filtering is applied. Then the analog signal is converted back to digital using ADC. In digital domain all processing is done with signal in digital form.

In the first method, the new sampling rate doesn't have any relationship with old sampling rate. But major disadvantage is the signal distortion. So the digital domain sampling rate conversion is preferred even then the new sampling rate depends on the old sampling rate.

The two basic operations in multirate signal processing are decimation and interpolation. Decimation reduces that sampling rate, whereas interpolation increases the sampling rate.

Down sampling:

The sampling rate of a discrete time signal x(n) an be reduced by a factor M by taking every Mth value of the signal.



A downsampler

The output signal y(n) is a downsampled signal of the input signal x(n) and can be represented by

y(n) = x(Mn)

Example:

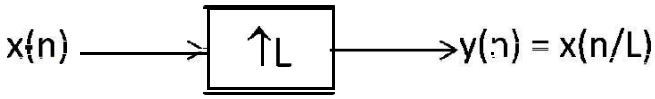
x(n) = {1, -1, 2, 4, 0, 3, 2, 1, 5, ….} if M = 2

y(n) = {1, 2, 0, 2, 5, ….}

Up sampling:

The sampling rate of a discrete time signal can be increased by a factor L by placing L-1 equally spaced zeros between e ach pair of samples. Mathematically, upsamplin g is represented by

$$y(n) = \begin{cases} x\left(\frac{n}{L}\right) & n = 0, \pm L, \pm 2L, \ldots \\ 0 & \text{otherwise} \end{cases}$$



Upsampler

Example:

x(n) = {1, 2, 4, -2, 3, 2, 1, …..}

if L = 2

y(n) = x(n/2) = {1, 0, 2, 0, 4, 0, -2, 0, 3, 0, 2, 0, 1, …..}

In practice, the zero valued samples inserted by upsampler are replaced with appropriate non-zero values using some typee f filtering process. This process is called interppolation.

5.2 Polyphase structure of Decimator:

The transfer function H(z) of the polyphase FIR filter is decomposed into M branches given by

$$H(z) = \sum_{m=0}^{M-1} z^{-m}\, p_m(z^M)$$

Where $p_m(z) = \sum_{n=0}^{N+1} h(Mn + m) z^{-n}$

The Z transform of an infiniite sequence is given by

$$H(z) = \sum_{n=}^{\infty} h(n) z^{-n}$$

In this case H(z) an be decomposed into M-branches as

$$H(z) = \sum_{m=0}^{M-1} z^{-m}\, p_m(z^M)$$

Where $p_m(z) = \sum_{r=}^{\infty} h(rM+m)z^{-r}$

$$H(z) = \sum_{m=0}^{M-1}\sum_{r=}^{\infty} z^{-m}\, h(rM+m)z^{-rM}$$

$$H(z) = \sum_{m=0}^{M-1}\sum_{r=}^{\infty} h(rM+m)z^{-(rM+m)}$$

$let\ h(Mn+m) = p_m(r)$

$$H(z) = \sum_{m=0}^{M-1}\sum_{r=}^{\infty} p_m(r)z^{-(rM+m)}$$

$$Y(z) = \sum_{m=0}^{M-1}\sum_{r=}^{\infty} p_m(r)X(z)z^{-(rM+m)}$$

$$y(n) = \sum_{m=0}^{M-1}\sum_{r=}^{\infty} p_m(r)x[n-(rM+m)]$$

$let\ x_m(r) = x(rM+m)$

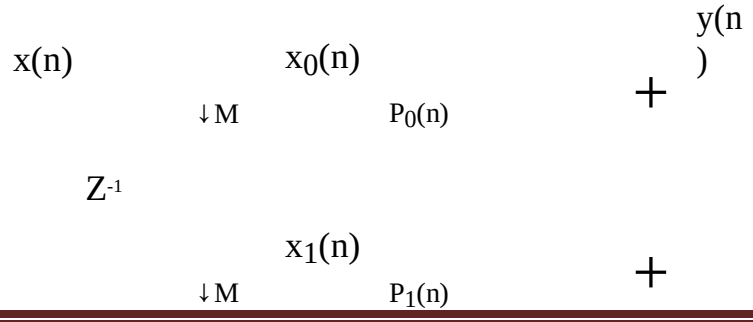
$$y(n) = \sum_{m=0}^{M-1}\sum_{r=}^{\infty} p_m(r)x_m(n-r)$$

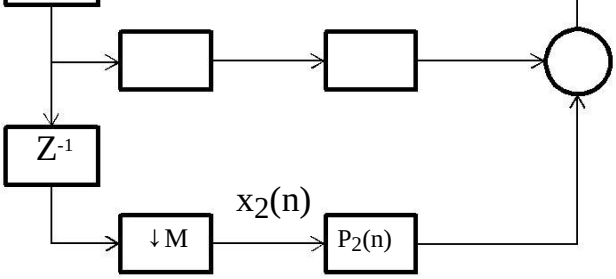
$$y(n) = \sum_{m=0}^{M-1} p_m(n) * x_m(n)$$

$$y(n) = \sum_{m=0}^{M-1} y_m(n)$$

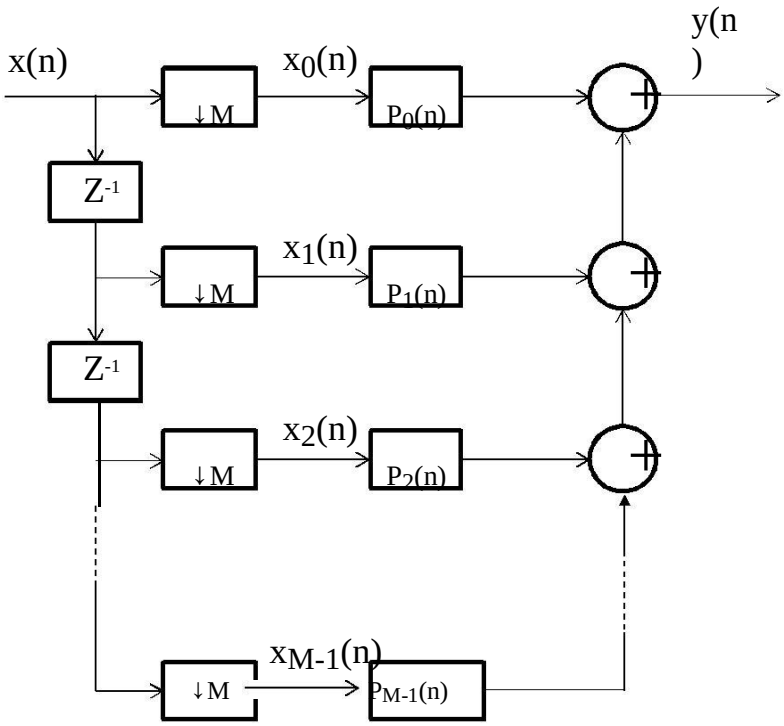
Where $y_m(n) = p_m(n) * x_m(n)$

The operation $p_m(n) * x_m(n)$ is known as polyphase convolution, and the overall process is polyphase filtering. $x_m(n)$ is obtained first delaying x(n) by M units then downsampling by a factor M. Next $y_m(n)$ can be obtained by convolving $x_m(n)$ with $p_m(n)$.



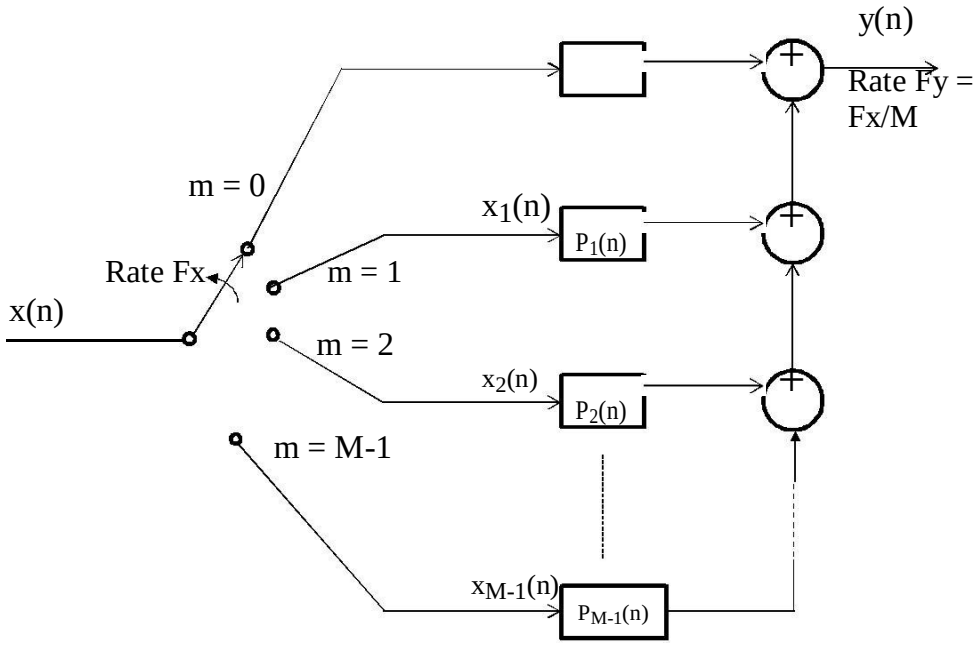


Polyphase structure of a 3 branch decimator



Polyphasestructure of a M branch decimator

The splitting of x(n) into the low rate sub sequence $x_0(n),x_1(n)……..x_{M-1}(n)$ is often represented by a commutator. The input values x(n) enter the delay chain at high rate. Then the M downsampler sends the group of M input values to M filters at time n=mM.

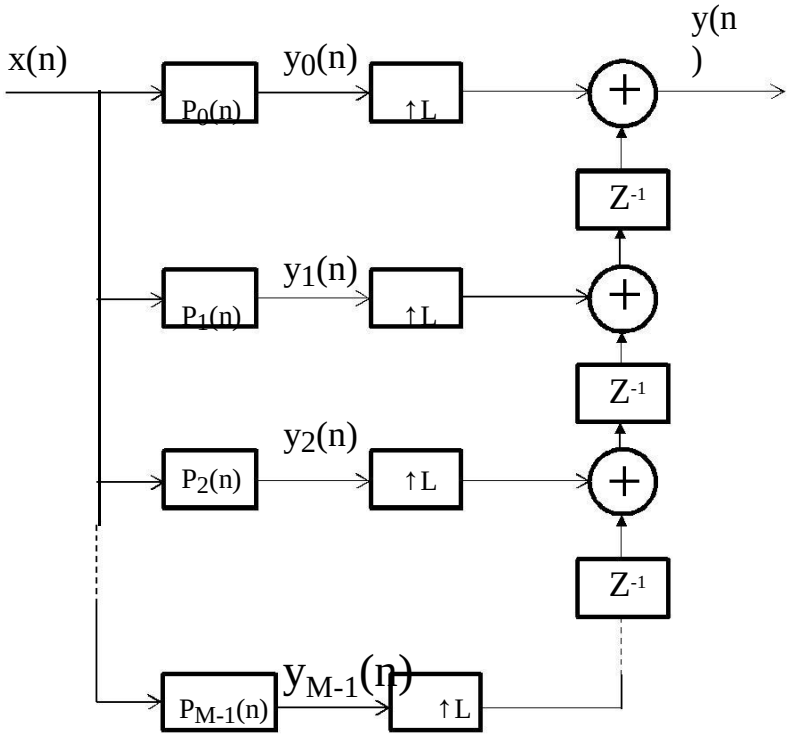

y(n)
Rate Fy = Fx/M
m = 0
x1(n)
P1(n)
Rate Fx
m = 1
x(n)
m = 2
x2(n)
P2(n)
m = M-1
xM-1(n)
PM-1(n)

Polyphase decimator with a commutator

To produce the output y(0), the commutator must rotate in counter-clockwise direction starting from m = M-1…… m=2, m=1, m=0 and give the input values x(-M+1)…..x(-2), x(-1), x(0) to the filters pM-1(n)…….$p_2(n)$,$p_1(n)$,$p_0(n)$.

5.3 Polyphase structure of Interpolator:

By transposing the decimator structure, we can obtain the polyphase structure for interpolator, which consists of a set of L sub filters connected in parallel.



Polyphasestructure of a M branch Interpolator

Here the polyphase components of impulse response are give by

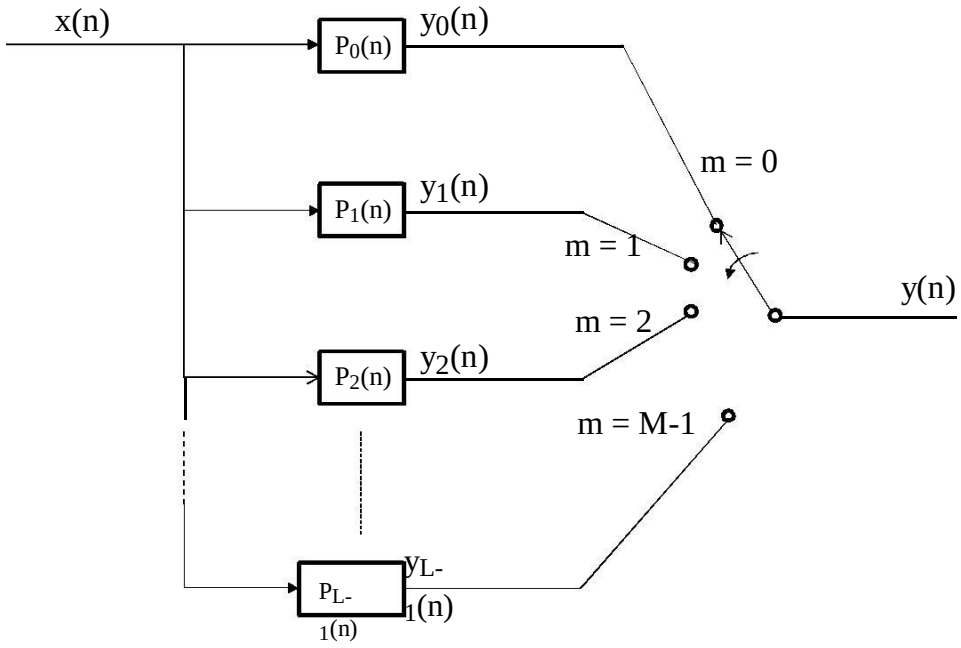
$P_m(n) = h(nL+m)$ m = 0, 1, 2 …… L – 1

Where h(n) is the impulse response of anti-imaging filter. The output of L sub filters can be represented as

$$y_m(n) = x(n)p_m(n) \quad m = 0,1,2.....L-1$$

By upsampling with a factor L and adding a delay z^{-m} the polyphase components are produced from $y_m(n)$. These polyphase components are all added together to produce the output signal y(n)

The output y(n) also can be obtained by combining the signals $x_m(n)$ using a commutator as shown below



Polyphase interpolator with a commutator

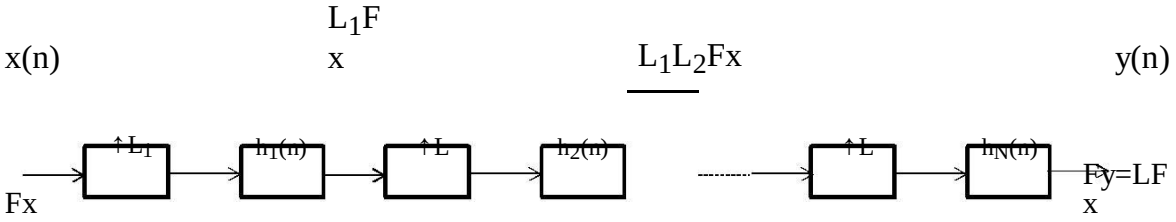
5.4 Multistage implementation of sampling rate conversion:

If the decimation factor M and/or interpolation factor L are much larger than unity, the implementation of sampling rate conversion in a single stage is computationally inefficient. Therefore for performing sampling rate conversion for either M>>1 and/or L>>1 the multistage implementation is preferred.

If the interpolation factor L>>1, then express L into a product of positive integers as

$$„ = \prod_{i=1}^{N} L_i$$

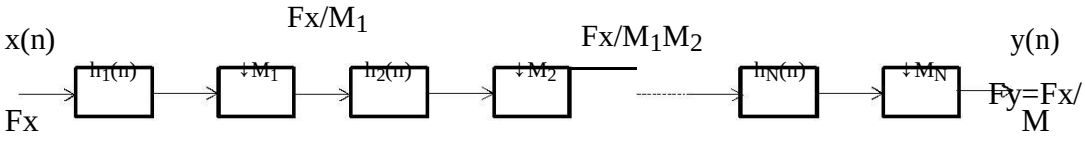
Then each interpolator L_i is implemented and cascaded to get N stages of interpolation and filtering.



Similarly if the decimation factor M>>1 then express M into a product of positive integers as

$$M = \prod_{i=1}^{N} M_i$$

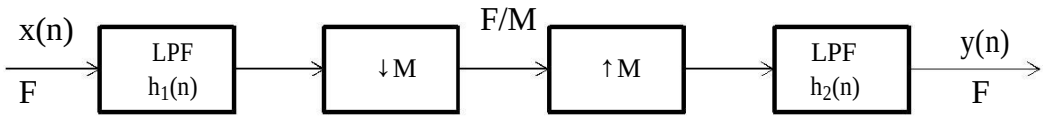
Each decimator M_i is implemented and cascaded to get N stages of filtering and decimators.



Implementation of narrowband LPF:

A narrowband LPF is characterized by a narrow passband and a narrow transition band. It requires a very large number of coefficients. Due to high value of N it is susceptible to finite word length effects. In addition the number of computations and memory locations required are very

high. So multirate approach of designing LPF overcomes this problem.



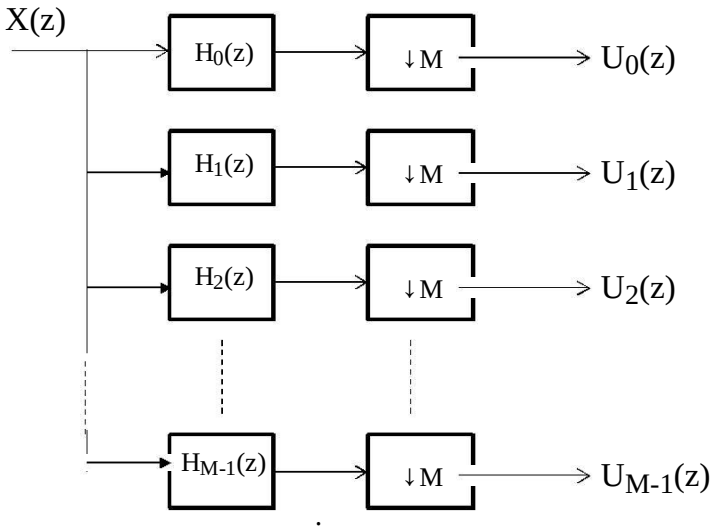
In the above diagram, the interpolator and decimator are in cascade. The filters $h_1(n)$ and $h_2(n)$ in the decimator and interpolator are lowpass filters. The sampling frequency of the input sequence is first reduced by a factor M then lowpass filtering is performed. Finally the original sampling frequency of the filtered data is obtained using interpolator.

To meet the desired specifications of a narrow band LPF, the filters $h_1(n)$ and $h_2(n)$ are identical, with passband ripple $\delta p/2$ and stopband ripple δs.

5.5 Filter bank:

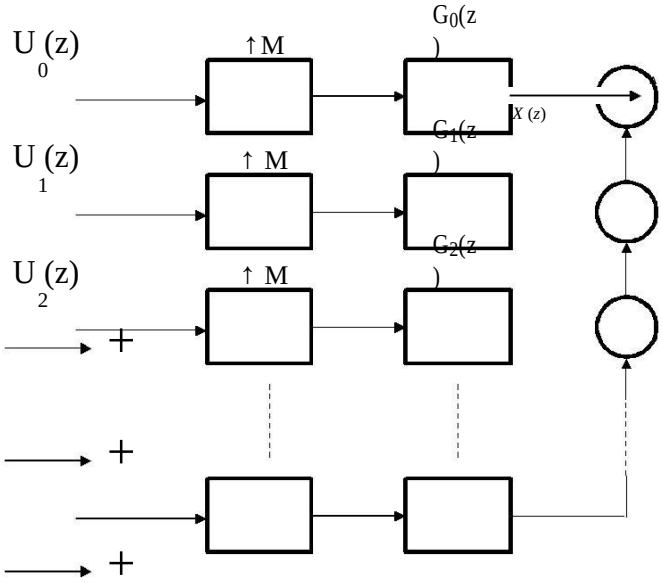
- Analysis filter bank
- Synthesis filter bank

Analysis filter bank:



2. It consists of M sub-filters. The individual sub-filter $H_k(z)$ is known as analysis bank.
3. All the sub-filters are equally spaced in frequency and each have the same band width.
4. The spectrum of the input signal $X(e^{j\omega})$ lies in the range $0 \leq \omega \leq \pi$.
5. The filter bank splits the signal into number of subbands each having a band width of π/M.
6. The filter $H_0(z)$ is lowpass, $H_1(z)$ to $H_{M-2}(z)$ are bandpass and $H_{M-1}(z)$ is highpass.
7. As the spectrum of signal is band limited to π/M, the sampling rate can be reduced by a factor M. The downsampling moves all the subband signals into the baseband range $0 \leq \omega \leq \pi/2$.

Analysis filter bank:



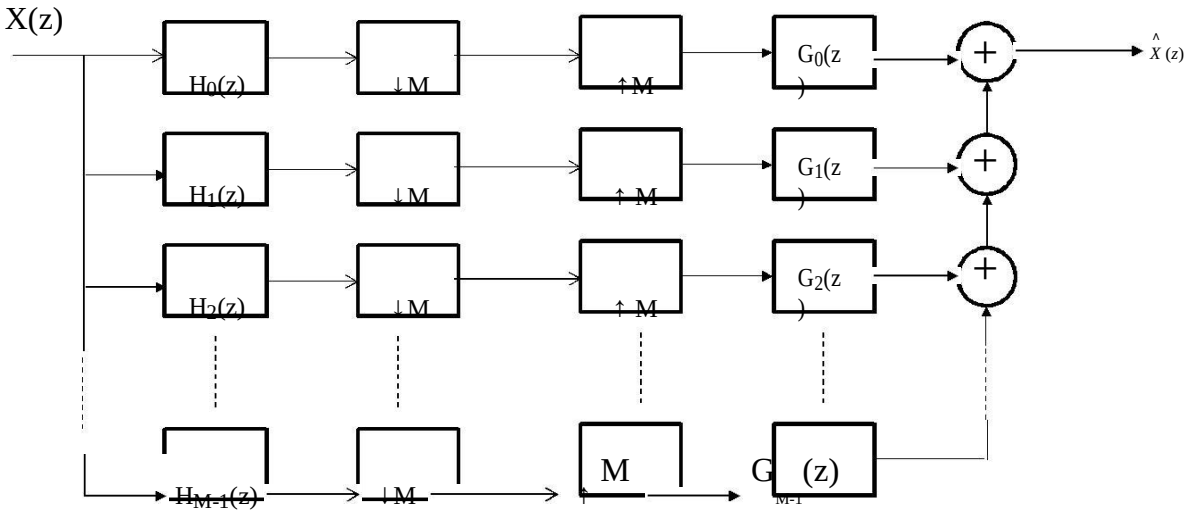
$U_{M-1}(z)$ ↑ M $G_{M-1}(z)$

The M channel synthesis filter bank is dual of M channel analysis filter bank. In this case $U_m(z)$ is fed to an upsampler. The upsampling process produces the signal $U_m(z^M)$. These signals are

applied to filters $G_m(z)$ and finally added to get the output signal $\hat{X}(z)$. The filters $G_0(z)$ to $G_{M-1}(z)$ have the same characteristics as the analysis filters $H_0(z)$ to $H_{M-1}(z)$.

Subband coding filter bank:

If we combine the analysis filter band and synthesis filter band we obtain an M-channel subband coding filter bank.



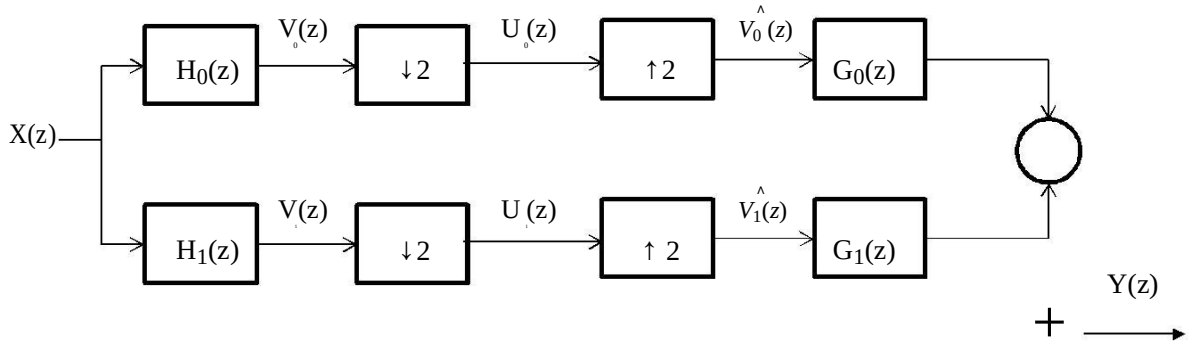
The analysis filter band splits the broadband input signal x(n) into M non-overlapping frequency band signals $X_0(z)$, $X_1(z)$……$X_{M-1}(z)$ of equal bandwidth. These outputs are coded and

transmitted. The synthesis filter bank is used to reconstruct output signal approximate the original signal. It has application in speech signal processing.

Quadrature Mirror Filter (QMF) Bank:

^

X (*z*)which should

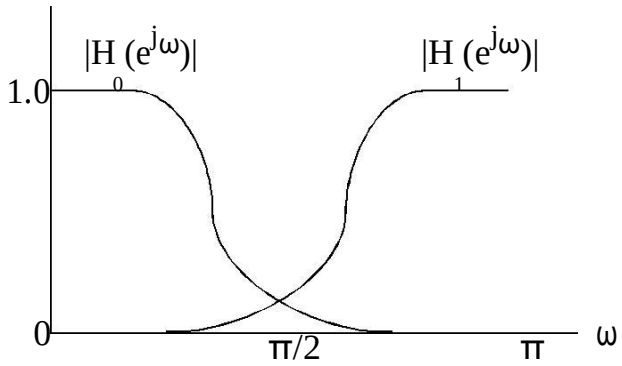


It is a two-channel subband coding filter bank with complementary frequency responses. It consists of two sections

1. Analysis section
2. Synthesis section

Analysis Section:

- The analysis section is a two channel analysis filter bank
- The signal x(n) is fed to a LPF $H_0(z)$ and a HPF $H_1(z)$ simultaneously. Hence the input signal x(n) is decomposed into high frequency component and low frequency component
- Since the normalized frequency range is ω = 0 and ω = π, the cut off frequency of HPF and LPF are chosen as π/2.



The output of low pass and high pass filters are

$$V_0(z) = X(Z)H_0(z) \text{ and}$$
$$V_1(z) = X(Z)H_1(z) \quad \ldots\ldots\ldots\ldots\ldots 1$$

Down sampling with M = 2, yields the subband signals

$$U_0(z) = \frac{1}{2}[V_0(z^{\frac{1}{2}}) + V_0(-z^{\frac{1}{2}}) \text{ and}$$
$$U_1(z) = \frac{1}{2}[V_1(z^{\frac{1}{2}}) + V_1(-z^{\frac{1}{2}}) \quad \ldots\ldots\ldots\ldots\ldots 2$$

Substitute equation 1 in equation 2

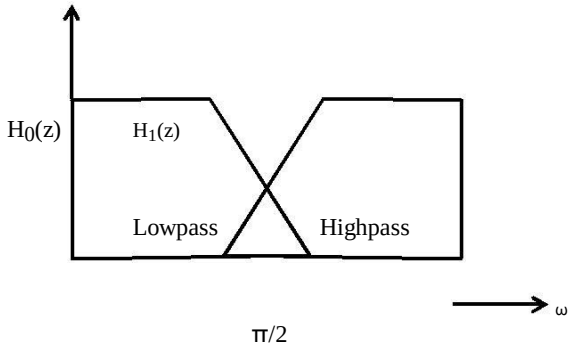
$$U_0(z) = \frac{1}{2}[X(z^{\frac{1}{2}})H_0(z^{\frac{1}{2}}) + X(-z^{\frac{1}{2}})H_0(-z^{\frac{1}{2}}) \text{ and}$$

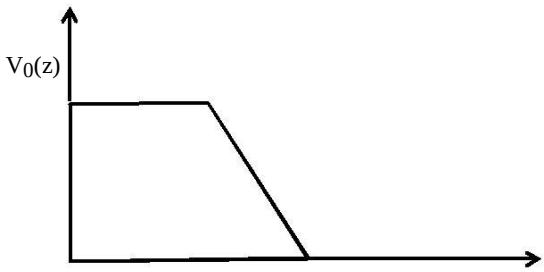
$$U1(z) = \frac{1}{2}[X(z^{\frac{1}{2}})H_1(z^{\frac{1}{2}}) + X(-z^{\frac{1}{2}})H_1(-z^{\frac{1}{2}})$$

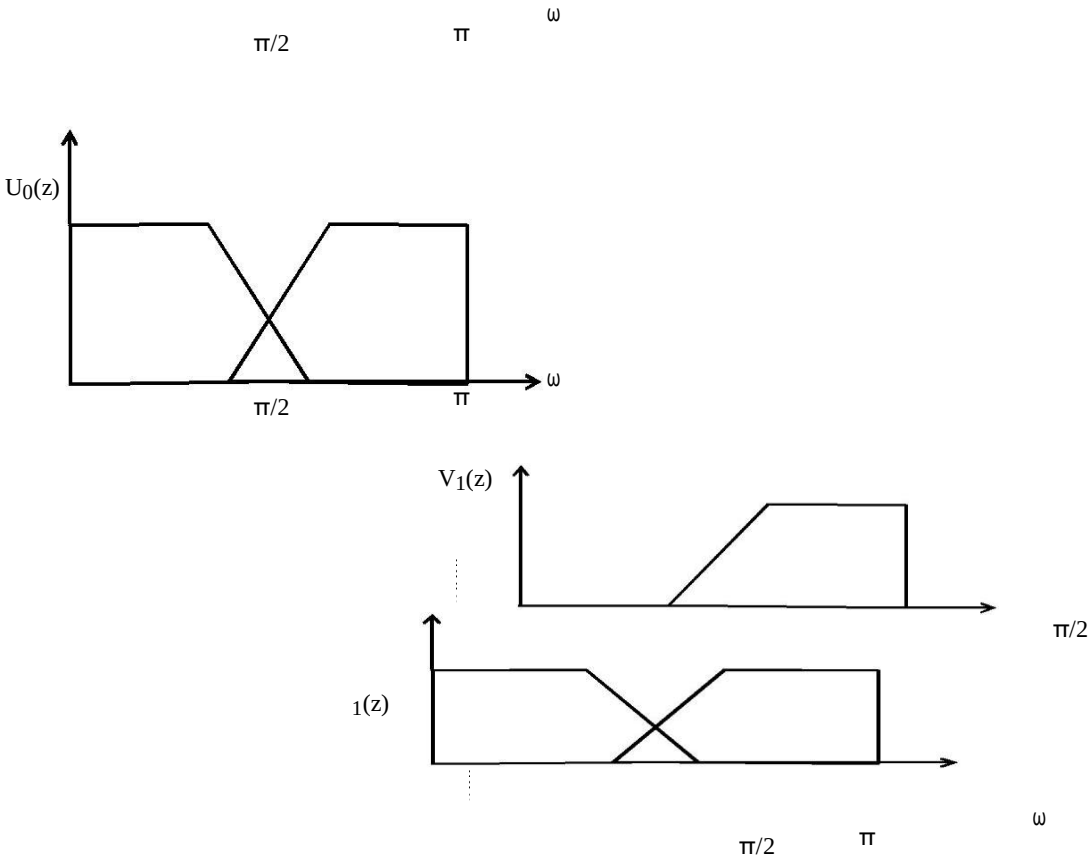
In matrix form

$$U_{\quad}(z) \qquad \frac{1}{\quad} \qquad \frac{1}{\quad} \qquad \frac{1}{\quad}$$

$$\begin{bmatrix} 0 \\ \end{bmatrix} = \frac{1}{2} \begin{bmatrix} H_0(z^{1/2}) & H_0(-z^{1/2}) \\ H_1(z^{1/2}) & H_1(-z^{1/2}) \end{bmatrix} \begin{bmatrix} X(z^{1/2}) \\ X(-z^{1/2}) \end{bmatrix} \quad \ldots\ldots\ldots\ldots\ldots \ldots 3$$


H0(z)
H1(z)
Lowpass
Highpass
ω
π/2


V0(z)



Frequency response characteristics of signals

x(n) is a white noise input signal. The frequency spectra of $V_0(z)$ have two components one is the original spectrum that depends on $X(z^{1/2})$ lies in the baseband and the other is periodic repetition that is function of $X(z^{-1/2})$. The high pass signal $U_1(z)$ drops into the baseband $0 \leq \omega \leq \pi$ and is reversed in frequency. Since the filtered signals are not properly band limited to π, alias signals appear in baseband.

Synthesis section:

The signals $U_0(z)$ and $U_1(z)$ are fed to the synthesis filter bank. Here the signals $U_0(z)$ and $U_1(z)$ are upsampled and then passed through two filters $G_0(z)$ and $G_1(z)$ respectively. The filter $G_0(z)$ is a lowpass filter and eliminates the image spectrum of $U_0(z)$ in the range $\pi/2 \leq \omega \leq \pi$. Meanwhile the highpass filter $G_1(z)$ eliminates most of the image spectra in the range $0 \leq \omega \leq \pi/2$. As the frequency range of the two signals $U_0(z)$ and $U_1(z)$ overlap, the image spectra is not completely eliminated.

The reconstructed output of the filter bank is

$$Y(z) = G_0(z)V_0(z) + G_1(z)\hat{V}_1(z)$$

$$Y(z) = G_0(z)U_0(z^2) + G_1(z)U_1(z^2) \quad \ldots\ldots\ldots\ldots\ldots\ldots 4$$

Where $\hat{V}_0(z) = U_0(z_2)$

$$\hat{V}_1(z) = U_1(z^2)$$

Equation 4 can be written in matrix form as

$$Y(z) = \begin{bmatrix} G_0(z) & G_1(z) \end{bmatrix} \begin{matrix} U_0(z^2) \\ U_1(z^2) \end{matrix}$$

From equation 3

$$U_0(z^2) = \frac{1}{2}\left[H_0(z)\ H_0(-z)\right] X(z)$$

$$U_1(z^2) = \frac{1}{2}\left[H_1(z)\ H_1(-z)\right] X(-z)$$

$$Y(z) = \begin{bmatrix} G_0(z) & G_1(z) \end{bmatrix} \frac{1}{2} \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix} \begin{bmatrix} X(z) \\ X(-z) \end{bmatrix}$$

$$Y(z) = \frac{1}{2}[G_0(z)H_0(z) + G_1(z)H_1(z)]X(z) + \frac{1}{2}[G_0(z)H_0(-z) + G_1(z)H_1(-z)]X(-z)$$

$$Y(z) = T(z)X(z) + A(z)X(-z) \quad \ldots\ldots\ldots\ldots 5$$

Where

$$T(z) = \frac{1}{2}[G_0(z)H_0(z) + G_1(z)H_1(z)]$$

$$A(z) = \frac{1}{2}[G_0(z)H_0(-z) + G_1(z)H_1(-z)]$$

The function T(z) describes the transfer function of the filter and is called distortion transfer function. The function A(z) is due to aliasing components.

Alias free filter bank:

To obtain an alias free filter bank, we can choose the synthesis filter such that A(z) = 0.

$$\text{i.e., } A(z) = \frac{1}{2}[G_0(z)H_0(-z) + G_1(z)H_1(-z)] = 0$$

$$G_0(z)H_0(-z) + G_1(z)H_1(-z) = 0$$

A simple sufficient condition for alias cancellation is

$$G_0(z) = H_1(-z) \quad \textit{and}$$

$$G_1(z) = -H_0(-z)$$

Then equation 5 becomes

$$Y(z) = T(z)X(z)$$

Substituting $z = e^{j\omega}$ yields

$$Y(e^{j\omega}) = T(e^{j\omega})X(e^{j\omega})$$

$$= |T(e^{j\omega})|\, e^{j\theta(\omega)} X(e^{j\omega})$$

If $|T(e^{j\omega})|$ is constant for all 'ω', there is no amplitude distortion. This condition is satisfied when $T(e^{j\omega})$ is an all pass filter. In same way, if $T(e^{j\omega})$ have linear phase there is no phase distortion. This condition is satisfied when $\theta(\omega) = \alpha\omega + \beta$ for constant α and β. Therefore $T(e^{j\omega})$ need to be a linear phase all pass filter to avoid any magnitude or phase distortion.

If an alias free QMF bank has no amplitude and phase distortion then it is called a perfect reconstruction (PR) QMF bank. In such a case

$Y(z) = kz^{-l} X(z)$ and the output y(n) = kx(n-l).

i.e., the reconstructed output of a PRQMF bank is a scaled, delayed replica of the output.

5.8 ADAPTIVE FILTER

An **adaptive filter** is a filter that ***self-adjusts*** its transfer function according to an optimizing algorithm. Because of the complexity of the optimizing algorithms, most adaptive filters are

digital filters that perform digital signal processing and adapt their performance based on the input signal. By way of contrast, a non-adaptive filter has static filter coefficients (which collectively form the transfer function).

For some applications, adaptive coefficients are required since some parameters of the desired processing operation (for instance, the properties of some noise signal) are not known in advance. In these situations it is common to employ an adaptive filter, which uses feedback to refine the values of the filter coefficients and hence its frequency response.

Generally speaking, the adapting process involves the use of a cost function, which is a criterion for optimum performance of the filter (for example, minimizing the noise component of the input), to feed an algorithm, which determines how to modify the filter coefficients to minimize the cost on the next iteration.

As the power of digital signal processors has increased, adaptive filters have become much more common and are now routinely used in devices such as mobile phones and other communication devices, camcorders and digital cameras, and medical monitoring equipment.

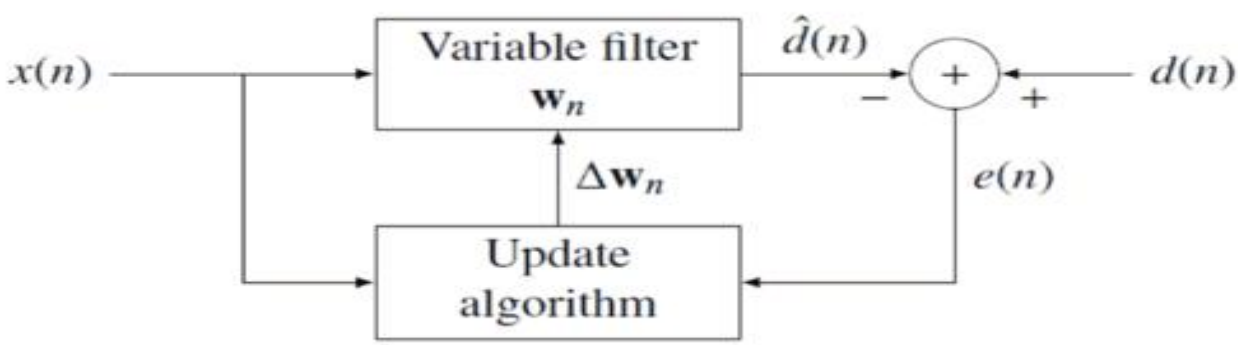
Example

Suppose a hospital is recording a heart beat (an ECG), which is being corrupted by a 50 Hz noise (the frequency coming from the power supply in many countries).One way to remove the noise is to filter the signal with a notch filter at 50 Hz. However, due to slight variations in the power supply to the hospital, the exact frequency of the power supply might (hypothetically) wander between 47 Hz and 53 Hz. A static filter would need to remove all the frequencies between 47 and 53 Hz, which could excessively degrade the quality of the ECG since the heart beat would also likely have frequency components in the rejected range.

To circumvent this potential loss of information, an adaptive filter could be used. The adaptive filter would take input both from the patient and from the power supply directly and would thus be able to track the actual frequency of the noise as it fluctuates. Such an adaptive technique generally allows for a filter with a smaller rejection range, which means, in our case, that the quality of the output signal is more accurate for medical diagnoses.

Block diagram

The block diagram, shown in the following figure, serves as a foundation for particular adaptive filter realisations, such as Least Mean Squares (LMS) and Recursive Least Squares (RLS). The idea behind the block diagram is that a variable filter extracts an estimate of the desired signal.



To start the discussion of the block diagram we take the following assumptions:

- The input signal is the sum of a desired signal $d(n)$ and interfering noise $v(n)$

  $x(n) = d(n) + v(n)$

- The variable filter has a Finite Impulse Response (FIR) structure. For such structures the impulse response is equal to the filter coefficients. The coefficients for a filter of order p are defined as

$$\mathbf{w}_n = [w_n(0), w_n(1), \ldots, w_n(p)]^T$$

.

- The error signal or cost function is the difference between the desired and the estimated signal

$$e(n) = d(n) - \hat{d}(n)$$

The variable filter estimates the desired signal by convolving the input signal with the impulse response. In vector notation this is expressed as

$$\hat{d}(n) = \mathbf{w}_n * \mathbf{x}(n)$$

where

$$\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-p)]^T$$

is an input signal vector. Moreover, the variable filter updates the filter coefficients at every time instant

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \Delta\mathbf{w}_n$$

where $\Delta\mathbf{w}_n$ is a correction factor for the filter coefficients. The adaptive algorithm generates this correction factor based on the input and error signals. LMS and RLS define two different coefficient update algorithms.

5.9 Applications of adaptive filters

- Noise cancellation
- Signal prediction
- Adaptive feedback cancellation
- Echo cancellation

Active noise control

Active noise control (ANC) (also known as **noise cancellation**, **active noise reduction (ANR)** or **antinoise**) is a method for reducing unwanted sound.

Explanation

Sound is a pressure wave, which consists of a compression phase and a rarefaction phase. A noise-cancellation speaker emits a sound wave with the same amplitude but with inverted phase (also known as antiphase) to the original sound. The waves combine to form a new wave, in a process called interference, and effectively cancel each other out - an effect which is called phase cancellation. Depending on the circumstances and the method used, the resulting soundwave may be so faint as to be inaudible to human ears.

A noise-cancellation speaker may be co-located with the sound source to be attenuated. In this case it must have the same audio power level as the source of the unwanted sound. Alternatively, the transducer emitting the cancellation signal may be located at the location where sound attenuation is wanted (e.g. the user's ear). This requires a much lower power level for cancellation but is effective only for a single user. Noise cancellation at other locations is more difficult as the three dimensional wavefronts of the unwanted sound and the cancellation signal could match and create alternating zones of constructive and destructive interference. In small enclosed spaces (e.g. the passenger compartment of a car) such global cancellation can be achieved via multiple speakers and feedback microphones, and measurement of the modal responses of the enclosure.

Modern active noise control is achieved through the use of a computer, which analyzes the waveform of the background aural or nonaural noise, then generates a signal reversed waveform to cancel it out by interference. This waveform has identical or directly proportional amplitude to the waveform of the original noise, but its signal is inverted. This creates the destructive interference that reduces the amplitude of the perceived noise.

The active methods (this) differs from passive noise control methods (soundproofing) in that a powered system is involved, rather than unpowered methods such as insulation, sound-absorbing ceiling tiles or muffler.

The advantages of active noise control methods compared to passive ones are that they are generally:

- More effective at low frequencies.
- Less bulky.
- Able to block noise selectively.

The first patent for a noise control system was granted to inventor Paul Lueg in 1934 U.S. Patent 2,043,416, describing how to cancel sinusoidal tones in ducts by phase-advancing the wave and canceling arbitrary sounds in the region around a loudspeaker by inverting the polarity. By the 1950s, systems were created to cancel the noise in helicopter and airplane cockpits including those patented by Lawrence J. Fogel in the 1950s and 1960s such as U.S. Patent 2,866,848, U.S. Patent 2,920,138, U.S. Patent 2,966,549 and Canadian patent 631,136. In 1986, Dick Rutan and Jeana Yeager used prototype headsets built by Bose in their around-the-world flight.[1][2]

Applications

Applications can be "1-dimensional" or 3-dimensional, depending on the type of zone to protect. Periodic sounds, even complex ones, are easier to cancel than random sounds due to the repetition in the wave form.

Protection of a "1-dimension zone" is easier and requires only one or two microphones and speakers to be effective. Several commercial applications have been successful: noise-cancelling headphones, active mufflers, and the control of noise in air conditioning ducts. The term "1-dimension" refers to a simple pistonic relationship between the noise and the active speaker (mechanical noise reduction) or between the active speaker and the listener (headphones).

Protection of a 3-dimension zone requires many microphones and speakers, making it less cost-effective. Each of the speakers tends to interfere with nearby speakers, reducing the system's overall performance. Noise reduction is more easily achieved with a single listener remaining stationary in a three-dimensional space but if there are multiple listeners or if the single listener moves throughout the space then the noise reduction challenge is made much more difficult. High frequency waves are difficult to reduce in three dimensions due to their relatively short audio wavelength in air. Sinusoidal noise at approximately 1000 Hz is double the distance of the average person's left ear to the right ear; such a noise coming directly from the front will be easily reduced by an active system but coming from the side will tend to cancel at one ear while being reinforced at the other, making the noise louder, not softer. High frequency sounds above 1000 Hz tend to cancel and reinforce unpredictably from many directions. In sum, the most effective noise reduction in three dimensions involves low frequency sounds. Commercial applications of 3-D noise reduction include the protection of aircraft cabins and car interiors, but in these situations, protection is mainly limited to the cancellation of repetitive (or periodic) noise such as engine-, propeller- or rotor-induced noise.

Antinoise is used to reduce noise at the working environment with ear plugs. Bigger noise cancellation systems are used for ship engines or tunnels. An engine's cyclic nature makes FFT analysis and the noise canceling easier to apply.

The application of active noise reduction produced by engines has various benefits:

- The operation of the engines is more convenient for personnel.
- Noise reduction eliminates vibrations that cause material wearout and increased fuel consumption.
- Quieting of submarines.

Linear prediction

Linear prediction is a mathematical operation where future values of a discrete-time signal are estimated as a linear function of previous samples.

In digital signal processing, linear prediction is often called linear predictive coding (LPC) and can thus be viewed as a subset of filter theory. In system analysis (a subfield of mathematics), linear prediction can be viewed as a part of mathematical modelling or optimization.

The prediction model

The most common representation is

$$\widehat{x}(n) = -\sum_{i=1}^{p} a_i x(n-i)$$

where $\widehat{x}(n)$ is the predicted signal value, $x(n-i)$ the previous observed values, and a_i the predictor coefficients. The error generated by this estimate is

$$e(n) = x(n) - \widehat{x}(n)$$

where $x(n)$ is the true signal value.

These equations are valid for all types of (one-dimensional) linear prediction. The differences are found in the way the parameters a_i are chosen.

For multi-dimensional signals the error metric is often defined as

$$e(n) = \|x(n) - \widehat{x}(n)\|$$

where $\|\cdot\|$ is a suitable chosen vector norm.

Estimating the parameters

The most common choice in optimization of parameters a_i is the root mean square criterion which is also called the autocorrelation criterion. In this method we minimize the expected value of the squared error $E[e^2(n)]$, which yields the equation

$$\sum_{i=1}^{p} a_i R(i-j) = -R(j),$$

for $1 \le j \le p$, where R is the autocorrelation of signal x_n, defined as

$$R(i) = E\{x(n)x(n-i)\},$$

and E is the expected value. In the multi-dimensional case this corresponds to minimizing the L_2 norm.

The above equations are called the normal equations or Yule-Walker equations. In matrix form the equations can be equivalently written as

$$Ra = -r,$$

where the autocorrelation matrix R is a symmetric, Toeplitz matrix with elements $r_{i,j} = R(i - j)$, vector r is the autocorrelation vector $r_j = R(j)$, and vector a is the parameter vector.

Another, more general, approach is to minimize

$$e(n) = x(n) - \hat{x}(n) = x(n) - \sum_{i=1}^{p} a_i x(n-i) = \sum_{i=0}^{p} a_i x(n-i)$$

where we usually constrain the parameters a_i with $a_0 = -1$ to avoid the trivial solution. This constraint yields the same predictor as above but the normal equations are then

$$Ra = [1, 0, \ldots, 0]^{\mathrm{T}}$$

where the index i ranges from 0 to p, and R is a $(p + 1) \times (p + 1)$ matrix.

Optimization of the parameters is a wide topic and a large number of other approaches have been proposed.

Still, the autocorrelation method is the most common and it is used, for example, for speech coding in the GSM standard.

Solution of the matrix equation $Ra = r$ is computationally a relatively expensive process. The Gauss algorithm for matrix inversion is probably the oldest solution but this approach does not efficiently use the symmetry of R and r. A faster algorithm is the Levinson recursion proposed by Norman Levinson in 1947, which recursively calculates the solution. Later, Delsarte et al. proposed an improvement to this algorithm called the split Levinson recursion which requires about half the number of multiplications and divisions. It uses a special symmetrical property of parameter vectors on subsequent recursion levels.

Echo cancellation

The term **echo cancellation** is used in telephony to describe the process of removing echo from a voice communication in order to improve voice quality on a telephone call. In addition to improving subjective quality, this process increases the capacity achieved through silence suppression by preventing echo from traveling across a network.

Two sources of echo have primary relevance in telephony: **acoustic echo** and **hybrid echo**.

Echo cancellation involves first recognizing the originally transmitted signal that re-appears, with some delay, in the transmitted or received signal. Once the echo is

recognized, it can be removed by 'subtracting' it from the transmitted or received signal. This technique is generally implemented using a digital signal processor (DSP), but can also be implemented in software. Echo cancellation is done using either echo suppressors or echo cancellers, or in some cases both.

Acoustic echo

Acoustic echo arises when sound from a loudspeaker—for example, the earpiece of a telephone handset—is picked up by the microphone in the same room—for example, the mic in the very same handset. The problem exists in any communications scenario where there is a speaker and a microphone. Examples of acoustic echo are found in everyday surroundings such as:

- Hands-free car phone systems
- A standard telephone or cellphone in speakerphone or hands-free mode
- Dedicated standalone "conference phones"
- Installed room systems which use ceiling speakers and microphones on the table
- Physical coupling (vibrations of the loudspeaker transfer to the microphone via the handset casing)

In most of these cases, direct sound from the loudspeaker (not the person at the far end, otherwise referred to as the Talker) enters the microphone almost unaltered. This is called direct acoustic path echo. The difficulties in cancelling acoustic echo stem from the alteration of the original sound by the ambient space. This colours the sound that re-enters the microphone. These changes can include certain frequencies being absorbed by soft furnishings, and reflection of different frequencies at varying strength. These secondary reflections are not strictly referred to as echo, but rather are "reverb".

Acoustic echo is heard by the far end talkers in a conversation. So if a person in Room A talks, they will hear their voice bounce around in Room B. This sound needs to be cancelled, or it will get sent back to its origin. Due to the slight round-trip transmission delay, this acoustic echo is very distracting.

Acoustic Echo Cancellation

Since invention at AT&T Bell Labs[1] echo cancellation algorithms have been improved and honed. Like all echo cancelling processes, these first algorithms were designed to anticipate the signal which would inevitably re-enter the transmission path, and cancel it out.

The Acoustic Echo Cancellation (AEC) process works as follows:

1. A far-end signal is delivered to the system.
2. The far-end signal is reproduced by the speaker in the room.
3. A microphone also in the room picks up the resulting direct path sound, and consequent reverberant sound as a near-end signal.
4. The far-end signal is filtered and delayed to resemble the near-end signal.
5. The filtered far-end signal is subtracted from the near-end signal.
6. The resultant signal represents sounds present in the room excluding any direct or reverberated sound produced by the speaker.

Challenges for AEC (Acoustic Echo Cancellation)

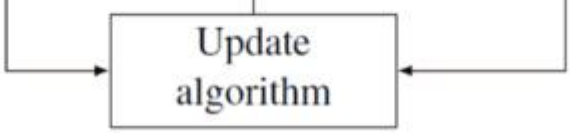
The primary challenge for an echo canceler is determining the nature of the filtering to be applied to the far-end signal such that it resembles the resultant near-end signal. The filter is essentially a model of the speaker, microphone and the room's acoustical attributes.

To configure the filter, early echo cancellation systems required training with impulse or pink noise, and some used this as the only model of the acoustic space. Later systems used this training only as a basis to start from, and the canceller then adapted from that point on. By using the far-end signal as the stimulus, modern systems can 'converge' from nothing to 55 dB of cancellation in around 200 ms.

Full Bandwidth Cancellation

Until recently echo cancellation only needed to apply to the voice bandwidth of telephone circuits. PSTN calls transmit frequencies between 300 Hz and 3 kHz, the range required for human speech intelligibility.

Videoconferencing is one area where full bandwidth audio is transceived. In this case, specialised products are employed to perform echo cancellation.


Update
algorithm