Digital Signal Processing

Prof. S. C. Dutta Roy

Department of Electrical Engineering

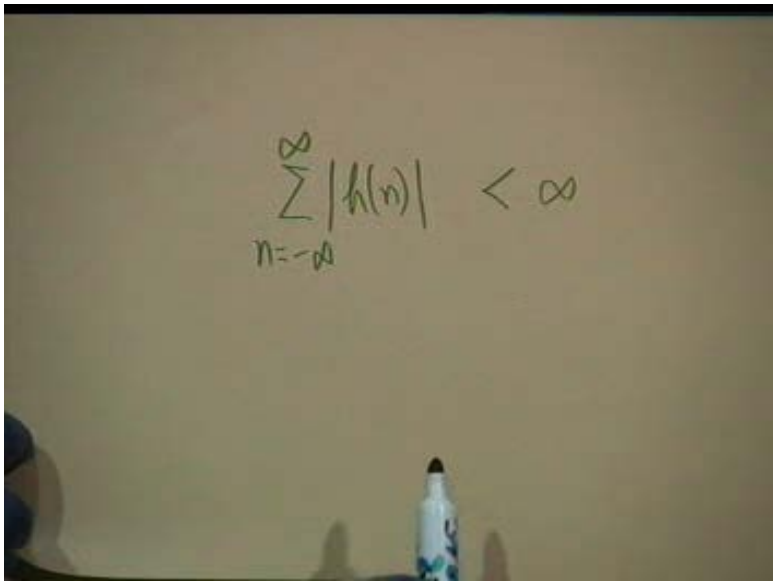
Indian Institute of Technology, Delhi

Lecture - 7

FIR AND IIR Recursive and non recursive Digital Systems

This is the 7th lecture on DSP and we propose to discuss FIR and IIR types of digital systems, and recursive and non recursive realizations and we wish to gain an entry to the frequency domain. In the last lecture, we had reviewed the trick for convolution that I talked to you about in lecture 5. Then I said it can be done also by polynomial multiplication and we saw that the results are the same; they have to be the same. We discussed what an inverse system is and we gave an example of an inverse system. Then we also mentioned that parallel interconnection adds up the impulse responses. It is very important in practice because parallel processing speeds up the operation. So for fast DSP, you have to use parallel decomposition. Then we talked about stability in terms of h(n) and we said that the system is stable if the impulse response is absolutely summable. The limits are from – infinity to + infinity and we ended up in discussing difference equations which are very similar to differential equations in the continuous time domain. Then we said that the solution to a difference equation consists of two parts: the complementary part and the particular solution. And I also made a strong point about the common mistake that people do to evaluate the constants of the solution. The constants should be evaluated only after adding the particular solution. The particular solution is of the same form as that of the excitation except for cases where the excitation also contains an eigenvalue of the system. The eigenvalues are the roots of the characteristic equation or the zeros of the characteristic polynomial. We discussed the cases of distinct roots, and repeated roots, and then we took an example of an excitation containing one of the eigenvalues.

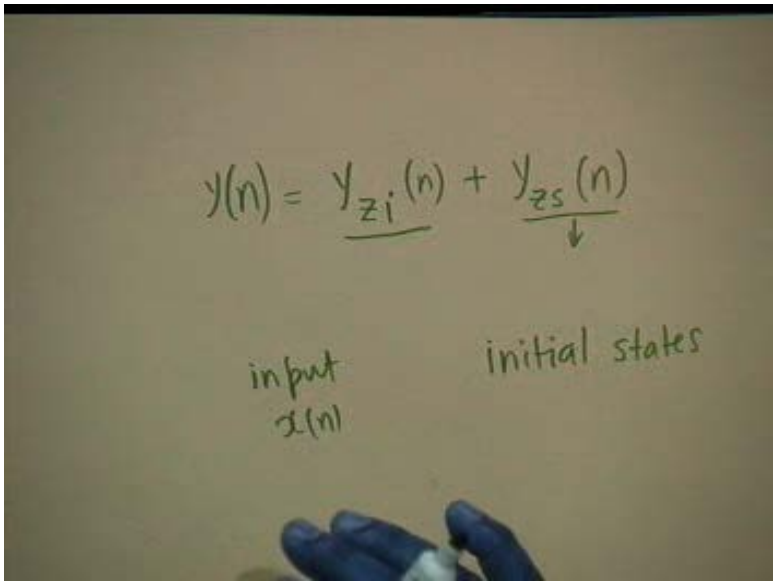
(Refer Slide Time: 02:33 - 04:14)



There is another method that is followed in practice, particularly in systems engineering because of its physical interpretation. This other way is to view it as a problem of superposition. The response of a system depends on the input as well as the initial conditions. Initial conditions are also known as initial states of the system. That is, if you specify initial conditions, then you know what is the condition of the system before the excitation has been applied. The superposition principle says that if it is a linear system, then the response would be due to each excitation applied separately and then you add up all of them. Now the excitation in a difference equation or a digital system consists of two parts, namely the input x(n) and the initial states. Therefore the output response y(n) shall be the sum of two components: zero input response plus the component due to zero state. This is another way of looking at the complete solution. However, the zero input solution is not the same as the complementary function, there is a difference. The solution under zero state conditions is not the same as the particular solution; one has to distinguish between the two. I want you to follow this carefully and then I will tell you which procedure I prefer. The output under zero input condition is the solution to the equation with x(n) = 0. Therefore the form of the solution shall be the same as the form of the complementary function. The difference is that this solution now shall be totally determined from the initial conditions. Call this solution as $y_{zi}(n)$.

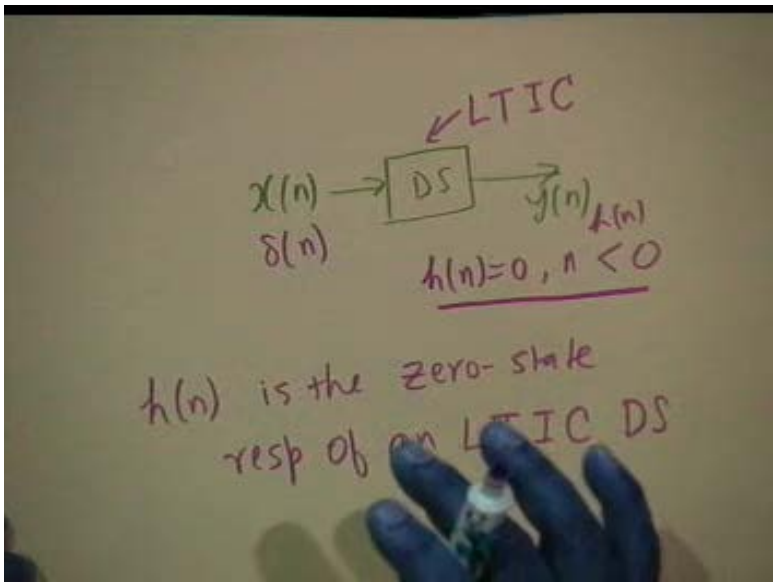
In our previous procedure, I warned you that the initial conditions have to be evoked only after adding the complementary function to the particular solution. In this alternative procedure, the zero input condition is the output with x(n) = 0 with the constants determined from the initial conditions. Naturally this will be different from the complementary function. Zero state condition says that initial conditions are 0. You find the solution to the equation, which will consist of a complementary part and a particular solution. And then evaluate the constants by putting initial conditions equal to 0. Call this solution as $y_{zs}(n)$. $y_{zi}(n)$ is simply the complementary function with constants determined from the initial conditions, $y_{zs}(n)$ is the total solution to the equation that is complementary function plus particular solution with constants determined from zero initial conditions. These are two different procedures for finding the solution to a difference equation. This is what the system theorists prefer because both zero state conditions and zero input conditions are realizable in practice and in an experimental situation, this is to be preferred. As far as theoretical computation is concerned, I prefer the previous procedure because in the alternative procedure, you have to find the constants twice. In the former procedure, the constants are evaluated only once.

(Refer Slide Time: 04:29 - 09:24)



But you should understand what we mean by zero state and what we mean by zero input condition. You take any of the examples that we did last time and you work out according to this procedure and then verify that the ultimate solution is the same. It has to be the same, there is no other alternative. But as far as computation or calculation on paper is concerned we prefer the previous procedure because the constants have to be determined only once. As an example of zero state response, you know that if I feed an impulse function δ(n) to a digital system, which is linear, time invariant and causal, then h(n) = 0, for n less than 0. It means that h(n) is the zero state response of the system. h(n) = 0 for n less than 0 means that the system is initially relaxed. There are no initial conditions; h(n) is defined like that. To find out h(n), you must first reduce the system to one with zero initial state. All initial conditions are made equal to 0. So we say that h(n) is the zero state response of an LTICDS, that is Linear Time Invariant Causal Digital System. h(n) is an example of zero state response.

(Refer Slide Time: 10:10 - 12:01)

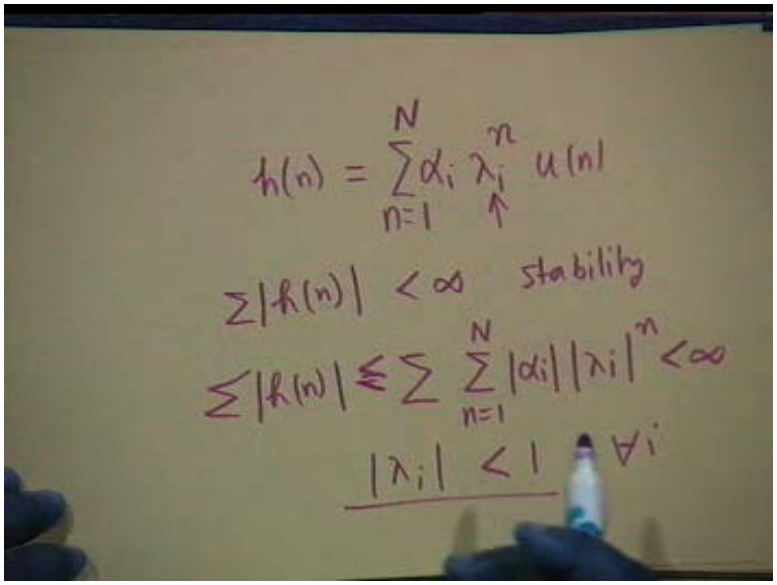


Therefore the form of h(n) shall be $\sum_{i=1}^{N} \alpha_i \lambda_i^n$ u(n), if it is a Nth order system. It shall be of the same form as the complementary function because $\delta(n) = 0$ for n greater than 0. Now stability demands that summation absolute value of h(n) should be less than infinity. If this summation is

to be less than infinity then what is the condition on these eigenvalues? We have $\sum_{n=-\infty}^{\infty} |h(n)|$ $\leq \sum_{n=-\infty}^{n=+\infty} \sum_{i=1}^{N} |\alpha_i \lambda_i^n|$.

Now, magnitude of alpha$_i$ should obviously be finite. If α_i is not finite, then your system to start with is unstable. So α_i is finite and therefore λ_i^n should be less than infinity for all i and n. How can you guarantee this? The condition is that $|\lambda_i|$ must be less than 1, for all i. If λ_i = 2, which you got earlier in an example, then the system is unstable. We took a first order system where the eigenvalue was – 0.5; that was a stable system. But if one of the roots of the characteristic equation exceeds unity in magnitude, the system is unstable. As you shall see later, these eigenvalues are also the poles of the system in a complex plane. So the poles should be confined to be within a unit circle.

(Refer Slide Time: 12:11 - 15:40)

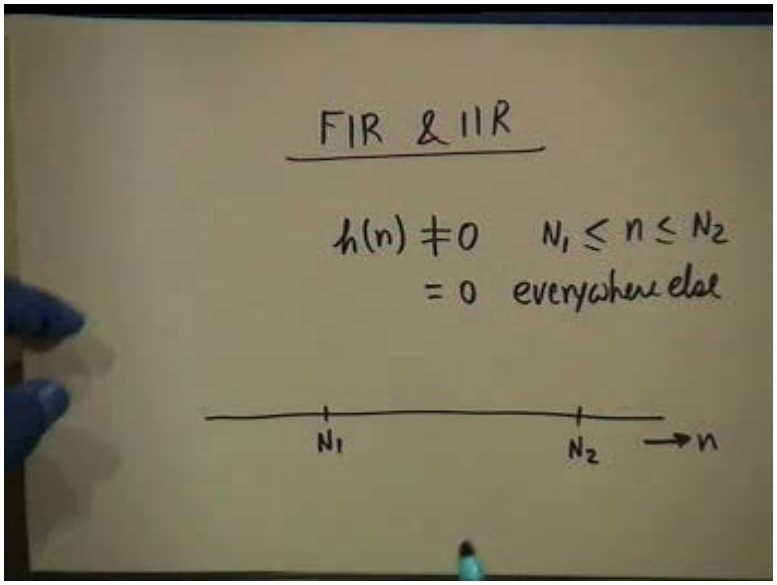


We shall come to this later after we introduce the z - transform. But right at this stage when we are working in the time domain, you know that the roots of the characteristic equation must be

such that they are bounded in magnitude by unity. Even $\lambda_i = 1$ is not permitted, because it is marginally stable. We cannot work with a marginally stable system because you never know, due to some small perturbations, the system may go wild.

Now we introduce the terms FIR and IIR. Finite impulse response and infinite impulse response refer to the length of the impulse response. An FIR system has h(n) which is non zero for $N_1 \leq n \leq N_2$ and it is zero everywhere else. That is, our range of vision on the n axis is limited on both sides.

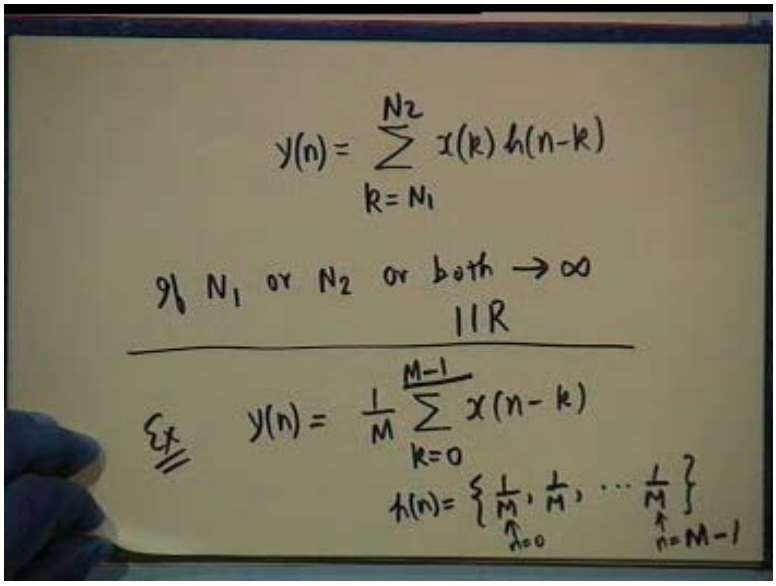
(Refer Slide Time: 16:22 - 17:12)



In a finite impulse response or FIR system, the output would be given by the convolution summation $\sum x(k)h(n-k)$. Of course, we are talking of a linear time invariant system; k shall go from N_1 to N_2; so the limits are finite and there are only $N_2 - N_1 + 1$ number of samples in the impulse response. On the other hand, if N_1 or N_2 or both go to infinity then the system becomes IIR. If the length of the impulse response goes to infinity, it is an infinite impulse response system. The moving average system $y(n) = \frac{1}{M}\sum_{k=0}^{M-1} x(n-k)$ is an FIR system. What is the

sequence h(n)? The samples are: h(n) = {1/M, 1/M,……1/M} starting at n = 0 and ending at n = M – 1. All the samples are equal. It is a rectangular gate u(n) – u(n – M), multiplied by 1/M.

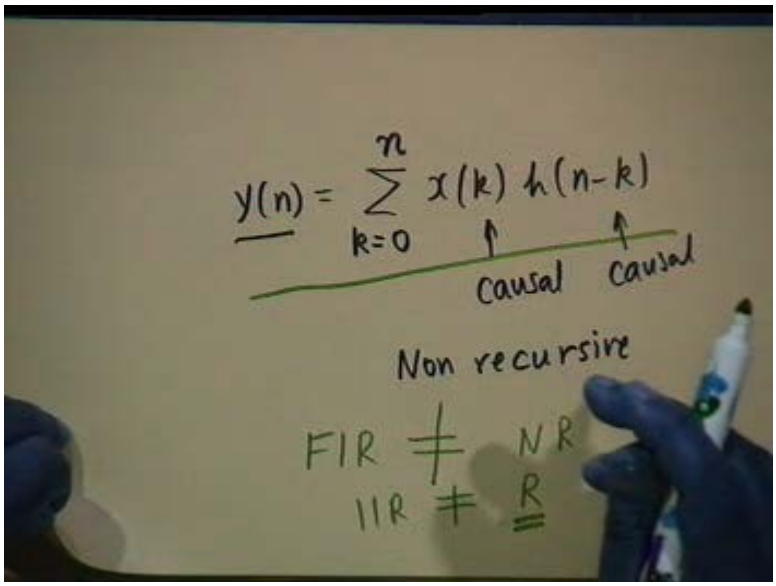
(Refer Slide Time: 17:16 - 19:50)



Whether it is FIR or IIR, the output can always be written as a convolution summation, that is, it can always be written as $\sum$ [x(k)h(n – k)]. The limits shall depend on the nature of x(k) and h(k). For example, if x(k) and h(k) are causal, that is the system is causal, and the input is also causal, then k goes from 0 to n. Now in the computation of this, whether it is FIR or IIR, we are only using the present value of the input and the past values of input. We are not using any past value of the output. Therefore this computation is non recursive. The most natural way of computing FIR filters is non recursive; convolution summation shows that IIR filters can also be computed non recursively although, in general, IIR filters are more convenient to compute recursively. In other words, FIR is not necessarily non recursive and IIR is not necessarily recursive. Recursive and non recursive are the two terms which are used to describe the process of computation: whether you require feedback or you do not require feedback. Recursive computation requires feedback and non recursive does not require feedback.

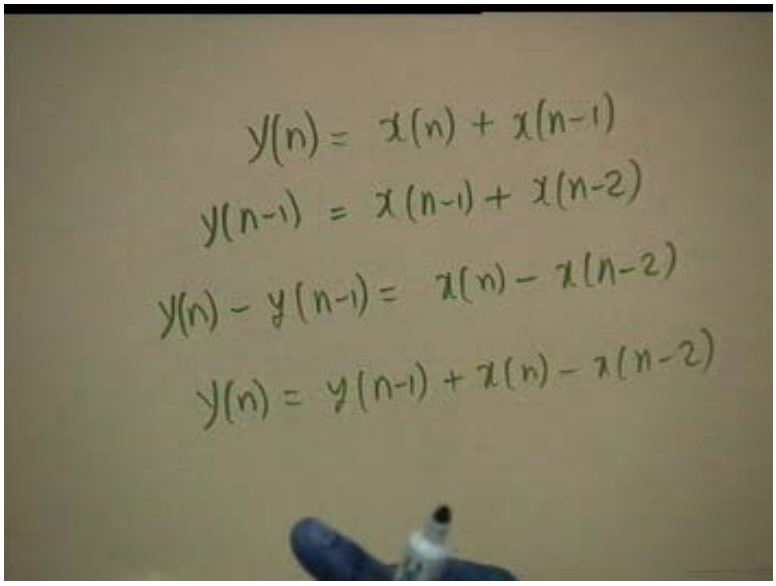
As I have shown, the convolution operation does not make a distinction between FIR and IIR. Therefore IIR filters can also be computed non recursively although, in general, it is more convenient to compute it recursively. I have shown examples of recursive computation earlier.

(Refer Slide Time: 20:03 - 22:40)



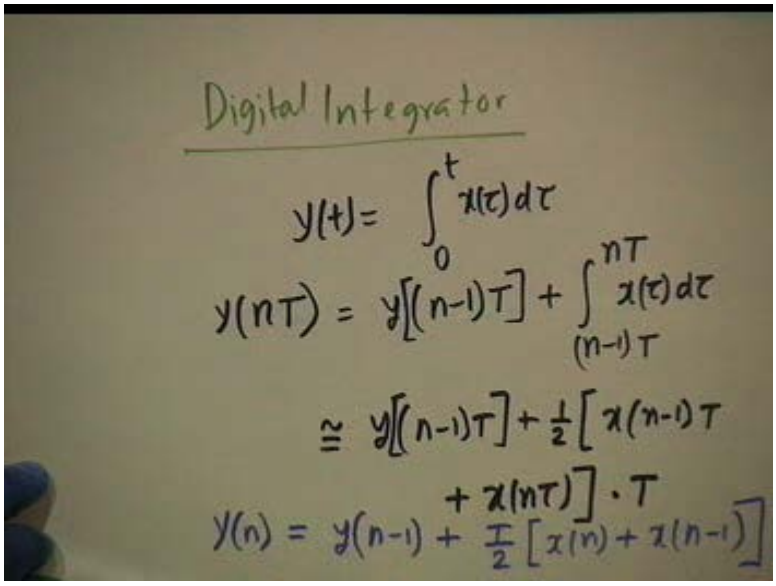
What about an FIR filter? Can it be computed recursively? Let us take an FIR system $y(n) = x(n) + x(n – 1)$. Obviously the computation is non recursive. But I can also write $y(n – 1) = x(n – 1) + x(n – 2)$ and I have the freedom to write $y(n) – y(n – 1) = x(n) – x(n – 2)$. That is, $y(n) = y(n – 1) + x(n) – x(n – 2)$ and this is recursive computation. I require the present value of the input, the value of input two samples earlier, and the last output sample. This is computation with feedback. Therefore FIR filters can be computed recursively or non recursively. IIR filters also can be computed either recursively or non recursively. The two terms should not be confused with each other.

(Refer Slide Time: 23:01 - 24:17)



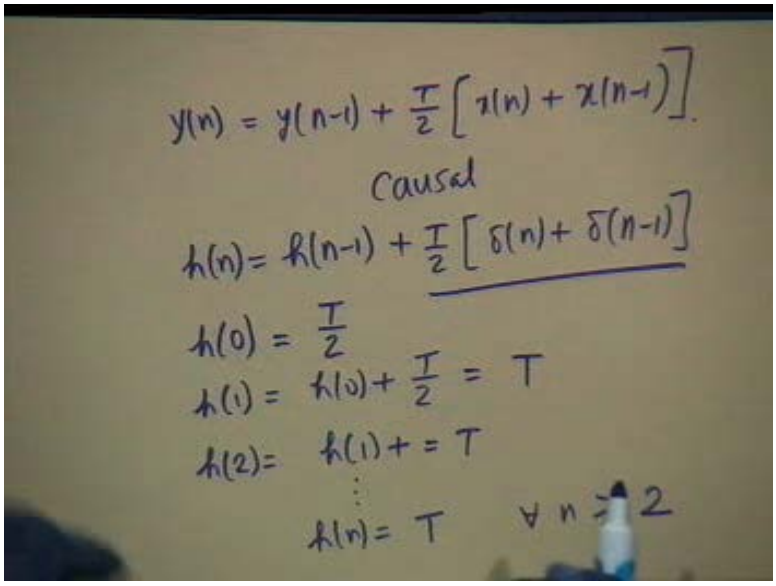
Take an example of a digital integrator. What is an analog integrator? Analog integrator is defined by y(t) = $\int_0^t x(\tau)\ d\tau$. We assume that x(t) is causal. Now if I wish to do it in the sample domain, we can write y(nT) = y[(n – 1)T] + $\int_{(n-1)T}^{nT} x(\tau)d\tau$. This is the exact equation; I have integrated up to n – 1 times T, then I have integrated between the interval (n – 1)T to nT. Now, I have to approximate $\int_{(n-1)T}^{nT} x(\tau)d\tau$ because x(t) is known only at (n – 1)T and nT. I do not know the value in between, it is not defined for a digital signal. Therefore I can approximate y(nT) as y[(n – 1)T] + ½[x(n – 1)T + x(nT)]T. This second term is T multiplied by average value of x(t) between the two limits. This is how integration is carried out by a digital computer. A digital computer handles only digital numbers. This is called a numerical approximation procedure. There are many algorithms for numerical integration; this is one of them, perhaps the simplest. You just take the average value and then multiply by the interval. If I translate this in the digital domain, I shall drop T from the arguments and write y(n) = y(n – 1) + (T/2)[x(n) + x(n – 1)].

(Refer Slide Time: 24:43 - 27:36)



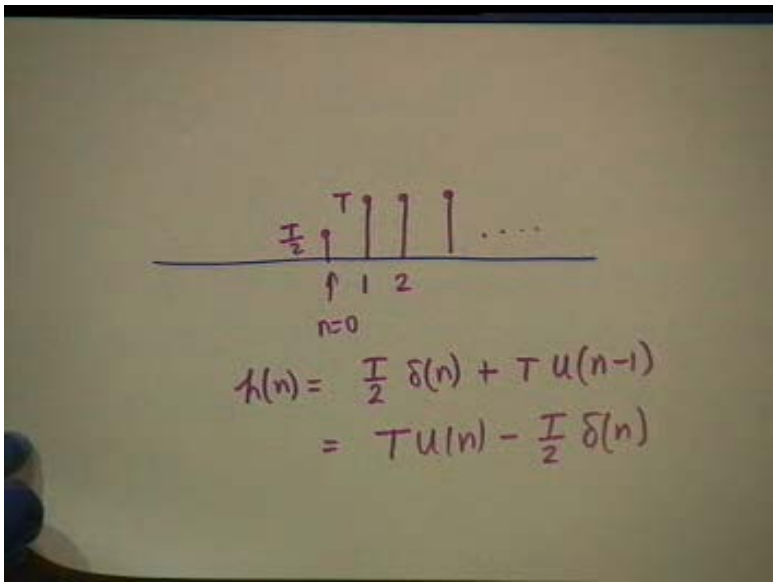
Let me write it again y(n) = y(n – 1) + (T/2)[x(n) + x(n – 1)]. From this equation can you guess whether it is an FIR or an IIR system? It is not obvious, the only thing I know that it is being computed recursively because I require a y(n – 1). Incidentally, this is what forms the basis of the so called bilinear transformation which we shall discuss at a later date. But let us assume that the system is causal. Causal means h(n) = 0 for n less than 0. Now h(n) = h(n – 1) + (T/2)[δ (n) + δ(n – 1). Because h(n) = 0, n ⟨ 0, we start computing from n = 0. Since h(– 1) = 0; δ(0) = 1, and δ(– 1) = 0, h(0) is simply (T/2). What is h(1)? h(1) = h(0) + T/2 = T; h(2) = h(1) + a term which does not contribute anything beyond n = 1 and therefore it is simply equal to T and this continues. Thus h(n) = T for all n greater than equal to 2, continuing to n = infinity. Therefore the impulse response is of infinite length.

(Refer Slide Time: 27:43 - 30:11)



If you so desire, you can write this as h(n) = (T/2) δ(n) + Tu(n – 1). Is there some other way I can express this? Yes, you can write h(n) = Tu(n) – (T/2) δ(n). There are many ways of expressing the same result; you can use the suitable one.

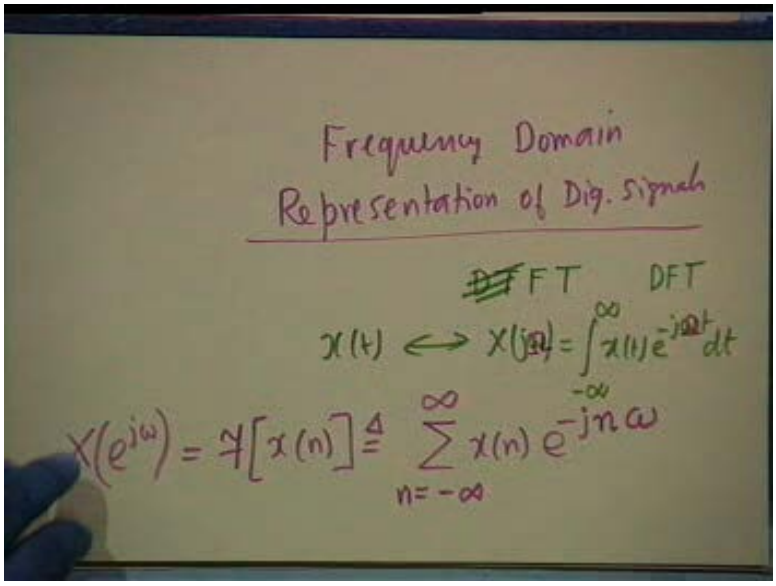
(Refer Slide Time: 30:17 - 31:16)



This is an infinite impulse response system. Recursion and non recursion have nothing to do with the length of the impulse response. A system with any length of the impulse response can be computed recursively as well as non recursively.

Now it is time to enter into the frequency domain representation of digital signals. In the analog domain, there are various kinds of transforms which takes t, the time variable, to a frequency variable. The latter can be ω if it is Fourier transform or the complex variable s if it is Laplace transform. In the digital domain, the Fourier transform is the most important one, while corresponding to the Laplace transform, we have the z transform. We shall talk about both but it is important to recognize that we use the same term, viz. Fourier transform for both analog and digital signals. Most of the text books write this as discrete time Fourier transform (DTFT). I find this pretty confusing because we shall also introduce what is DFT. DTFT and DFT are very close to each other; so instead of discrete time Fourier transform, I shall simply call it as Fourier transform. DFT is something else we shall define at a later date. The Fourier transform of an analog signal x(t), as you know, is $X(j\Omega) = \int_{-\infty}^{\infty} x(t) e^{-j\Omega t}\, dt$. We have used capital Ω instead of the usual small ω, because small ω is a symbol we have reserved for normalized digital frequency. This involves integration. In the digital signal domain, the Fourier transform of a sequence x(n) is very simply defined as the summation $x(n)\, e^{-j\omega n}$; here ω is normalized digital frequency; n, in general, goes from – infinity to + infinity. This is the definition of the Fourier transform. You see that the FT is a series in $e^{j\omega}$. Hence the argument of X is taken as $e^{j\omega}$. Note this difference in notation.

(Refer Slide Time: 34:41 - 35:14)

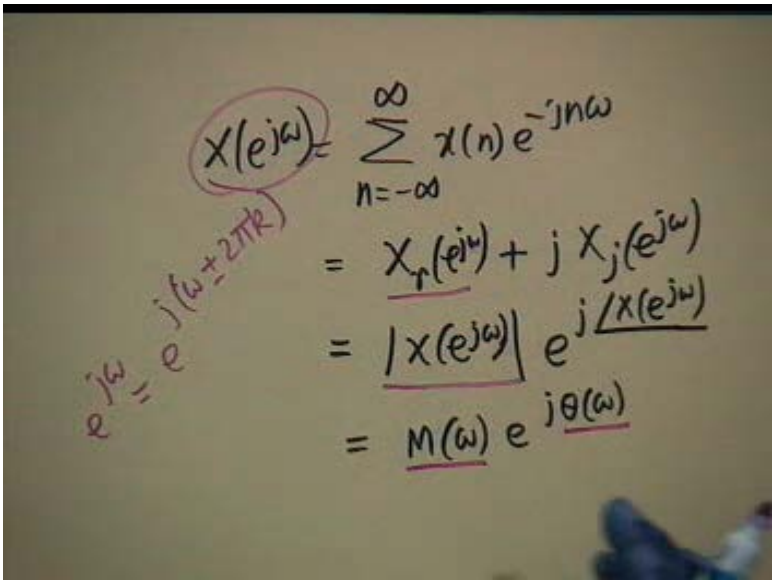


Let me write this again: $X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\varpi n}$; it is a much simpler expression than the analog FT. In general $X(e^{j\omega})$ is a complex quantity because $e^{j\omega}$ is $\cos\omega + j\sin\omega$. Therefore I can write $X(e^{j\omega})$ as a real part (X_r) plus an imaginary part jX_j. Both X_r and X_j are real quantities. I am not writing the argument, which is $e^{j\omega}$. $X(e^{j\omega})$ can also be written in terms of magnitude and phase, that is you can write this as $|X| e^{j\theta}$ where θ is the angle of X. Let us write this as $Me^{j\theta}$. If you look at the definition carefully, you observe that $X(e^{j\omega})$ is a continuous function of ω. Although x(n) is a discrete signal, it exists only at n = 0, ± 1, ± 2 etc, in the frequency domain it is a continuous function of ω. It also shows that if ω is increased or decreased by 2π or any multiple of 2π, the function does not change because $e^{j(\omega \pm 2\Pi k)} = e^{j\varpi}$. Therefore $X(e^{j\omega})$ is a periodic function of period 2π and this strengthens our earlier statement that our range of vision is from – π to + π. We do not have to go beyond that because we have a periodic function.

The other thing one should remember is that in the real part, if ω changes to – ω, it does not change. It is an even function of ω because the magnitude is even and cosΘ is an even function. The real part is nothing but magnitude M multiplied by cosΘ. The magnitude is even, while the phase is odd; if you change ω to – ω, sign of the phase changes. The magnitude and real part are

even functions. The imaginary part and the phase are odd functions. Further, notice that Θ is also periodic with a period of 2π. If theta is incremented by 2π neither cosine changes nor sine changes and therefore, while M (ω) can be found uniquely, there exists a fuzziness or uncertainty about the actual value of Θ. If you find Θ = – 45°, it could also be 360° – 45° or 720° – 45° or – 360° – 45°. Therefore about the phase, there is an uncertainty. We cannot work with uncertainty in the laboratory, in hardware, therefore we decide once for all that we shall only consider the principal value. Principal value is the value which lies between – π and + π.

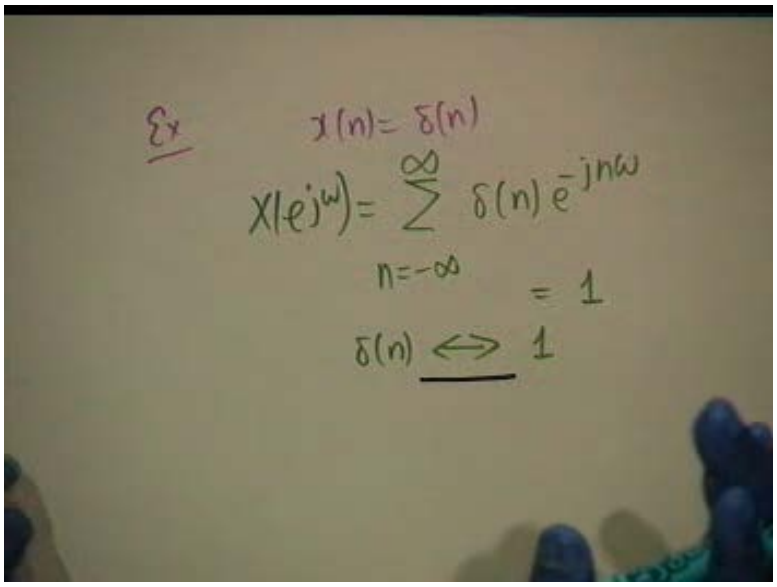
(Refer Slide Time: 35:20 - 40:29)



Why does one go to the frequency domain at all? Frequency domain is complex. There is a real part and there is an imaginary part. Why does one complicate life? It is because it simplifies life in some other fashion. Processing of a signal in time domain is much more complicated than processing in frequency domain. For example, convolution requires summation of multiplication of two signals; on the other hand, convolution operation is equivalent to multiplication in the frequency domain. You do not have to do this graphical shifting etc. So one goes to the frequency domain and intentionally complicates life in order to be able to simplify life in some other fashion. Be it design, or analysis or synthesis, everything is much simpler in the frequency domain than in the time domain.

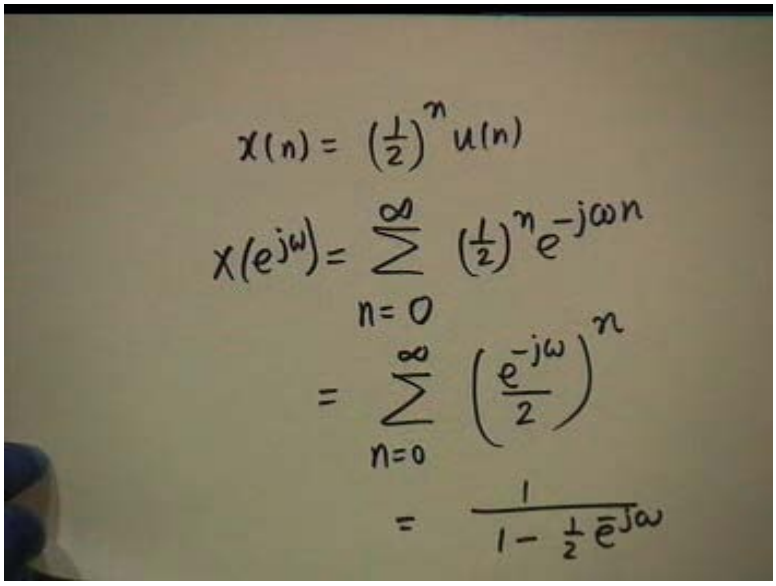
We take an example of Fourier transform. The simplest signal is $x(n) = \delta(n)$ which is an impulse function. Now $\delta(n)$ exists at $n = 0$ only and therefore this summation in $X(e^{j\omega})$ shall be replaced by a single term and that is equal to 1. So the Fourier transform of $\delta(n)$ is 1. We write this as $\delta(n) \Leftrightarrow 1$. Notice the symbol I have introduced here: a double sided arrow; it means that the Fourier transform is unique. In other words, if the Fourier transform is given, you should be able to go back to the time domain uniquely; there should be no arbitrariness, no confusion and no uncertainty. It must be a one to one representation, otherwise it is invalid. This is why we insisted that phase should be viewed only in the principal domain, otherwise there is uncertainty and we cannot work with uncertainty.

(Refer Slide Time: 41:36 - 43: 10)



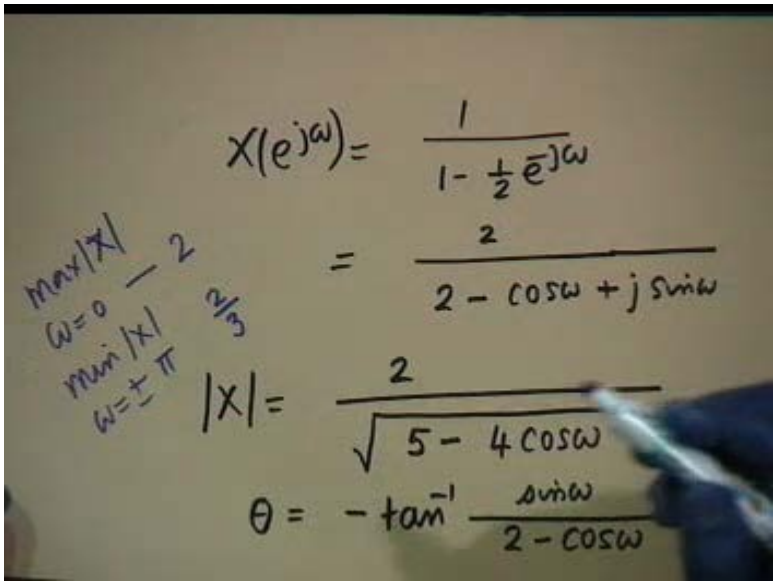
Let us take a second example: $x(n) = (1/2)^n u(n)$. The Fourier transform is $X(e^{j\omega}) = \text{summation}[(1/2)^n e^{-j n \omega}]$ where $n = 0$ to infinity which I can write as summation $([e^{-j\omega}/2]^n)$ where $n = 0$ to infinity. Does this summation converge? If it does not, if it diverges, then the Fourier transform does not exist. The existence is guaranteed here because the magnitude of $(e^{-j\omega})$ is 1. The sum is $1/(1 - [e^{-j\omega}/2])$. So the Fourier transform does exist and this is the value.

(Refer Slide Time: 43:18 - 44:23)



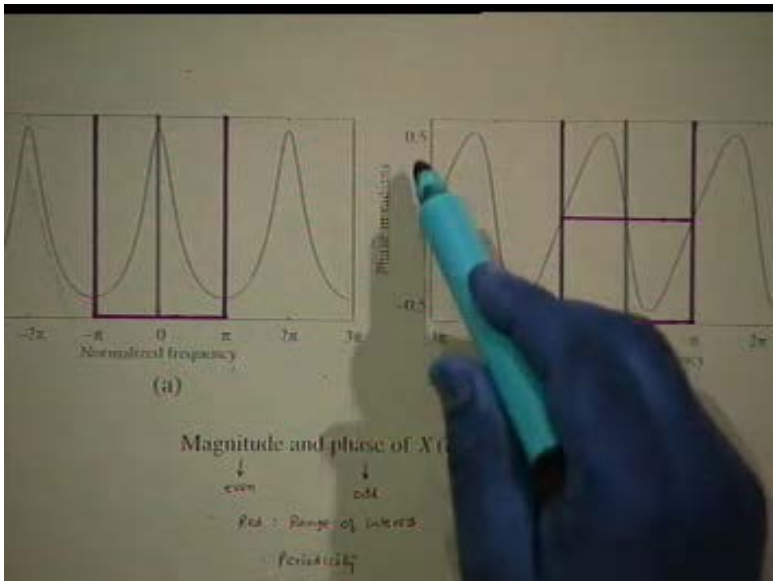
For $X(e^{j\omega}) = 1/(1 - [e^{-j\omega}/2])$, I can find the magnitude and phase of this function. I can write this as $2/[2 - \text{cosine}(\omega) + j\text{sine}(\omega)]$. So the magnitude I can write as magnitude of the numerator divided by the magnitude of the denominator and the magnitude of the numerator is 2. The magnitude of the denominator would be square root of $[5 - 4\text{cosine}(\omega)]$. This is obvious. And the phase is equal to the phase of the numerator, which is 0, – the phase of the denominator, which is $\tan^{-1}$ of imaginary part / real part, that is $\tan^{-1}$ {sine $(\omega)/[2 - \text{cosine } (\omega)]$}. If you look at these expressions carefully, X magnitude is obviously maximum when cosine (ω) is 1. You want the denominator to be as small as possible, so the maximum occurs at $\omega = 0$, the value is 2. What is the minimum? The minimum occurs at $\omega = \pm \pi$, so it will be 2/3. Thus the magnitude varies between 2 and 0.666. What can we say about the phase? What is the phase at $\omega = 0$? It is 0 because X(0) is a real quantity. So the phase starts from 0 at $\omega = 0$.

(Refer Slide Time: 44:32 - 47:02)



I have got here a plot of the magnitude and phase of this quantity $1/(1–0.5\ e^{-j\omega})$. This picture is worth 1000 words, if not more. Look at the picture; I have highlighted the important parts with colour. The magnitude is periodic and the range of vision is this red marked region $– \pi$ to $+ \pi$. In fact if you plot from 0 to π, this is good enough because |X| is an even function and the range $–\pi$ to 0 can be easily taken care of. So we shall basically confine to the range 0 to π. And if you look at the range 0 to π what can you say about the spectrum of the signal? The signal $x(n) = (1/2)^n u(n)$ is clearly a low pass signal. The phase, on the other hand, is odd. The phase from 0 to $– \pi$ is the negative of the phase from 0 to $+ \pi$. The phase is an odd function. It is not sinusoidal, it is a slanted sinusoid. It can be of any arbitrary shape but there is a maximum and there is a minimum. Again in phase we confine to $– \pi$ to $+ \pi$ and since we know phase is an odd function of frequency, we simply plot from 0 to π. We remove all uncertainties by confining the phase value also between π and $– \pi$.

(Refer Slide Time: 47:15 - 49:28)

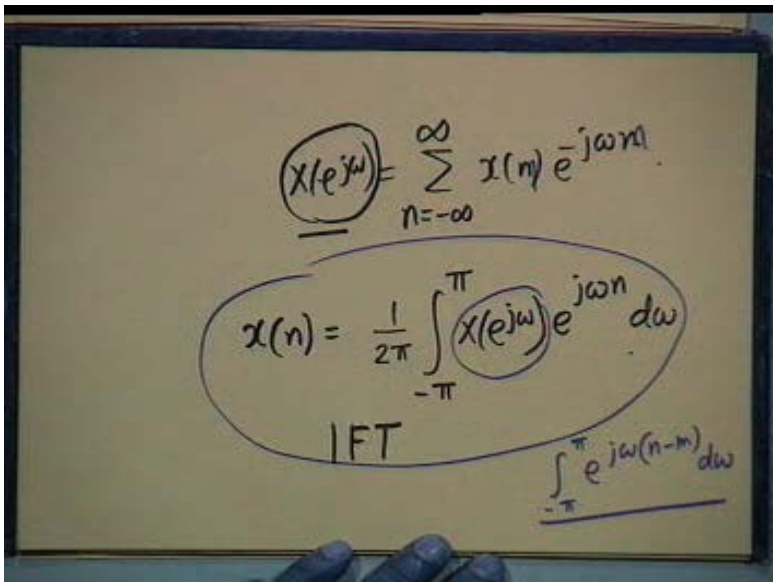


Now I come back to definition of the Fourier Transform, $X(e^{j\omega})$ = summation $x(n)\ e^{-j\ n\ \omega}$ where n = – infinity to + infinity. I have already made the point that if the Fourier transform is to be of any use then it must be unique and one to one. That is, given x(n), you should be able to find $X(e^{j\omega})$. There cannot be two answers. In a similar manner, if the signal is given in the frequency domain, you should be able to go back to x(n). And if you look at the summation, and recall what Fourier series is, you would recognize that X is a Fourier series, not of a time function, but of a frequency domain function. It is periodic of period 2π and it is expressed as the sum of exponentials with coefficient x(n). Therefore x(n) must be given by the formula for evaluation of Fourier coefficients that is $x(n) = [1/(2\pi)]$ integral (over one period $-\pi$ to $+\pi$) $X(e^{j\omega})\ e^{jn\omega}$ integrated over d ω. So this is the inverse Fourier transform relationship. Now, the two are not independent of each other. This is another point I want to make very clearly and very loudly. The two are not independent of each other because we require one to one transformation. In other words, if X is given, x(n) can be derived and if x(n) is given, X can be derived; so both are not matters of definition. Only one of them can be defined, the other should follow. The derivation is extremely simple; what you do is to substitute for X from its definition and then evaluate. You will get an integral of the form $-\pi$ to $\pi\ e^{j\omega\ (n\ -\ m)}\ d\omega$. If you substitute for the $X(e^{j\omega})$ you cannot

retain small n because this summation and integration are independent of each other. The integration contains $e^{jn\omega}$, so you must change this dummy variable n to m.

$e^{j\,(n\,-\,m)\,\omega}$, as you know, consists of cosine and sine; so for one period the cosine integral will become 0 and integral sine will also become 0, except in the case n = m. In that case, the integral simply becomes x(n) 2π and divided by 2π, it becomes x(n).

(Refer Slide Time: 50:35 - 53:55)



In conclusion, recursive is not necessarily IIR, non recursive is not necessarily FIR and vice versa. Why do we go to Fourier transform? We go to Fourier transform, although it is complex, because analysis, design and synthesis become much simpler in the frequency domain. But the frequency domain has to be handled with care because of uncertainty in Θ, the phase. There is no uncertainty in magnitude but Θ can be multi-valued because addition or subtraction of 2π does not matter. The third point is, in Fourier transform relationship, Fourier transform and Inverse Fourier transform are not independent of each other, one depends on the other. It has to be so for it to be useful to us. If it has to be useful then the transformation must be unique. That is, given one, you can find the other. So, one is a matter of definition, the forward transform or the reverse transform. How did I find IFT? By comparison to Fourier series. That was also somewhat

unconventional because Fourier series is usually written as periodic function in time. Now here we identify a periodic function in frequency. So it was not difficult for us to find out what the inverse Fourier transform is. We shall look into more details next time.