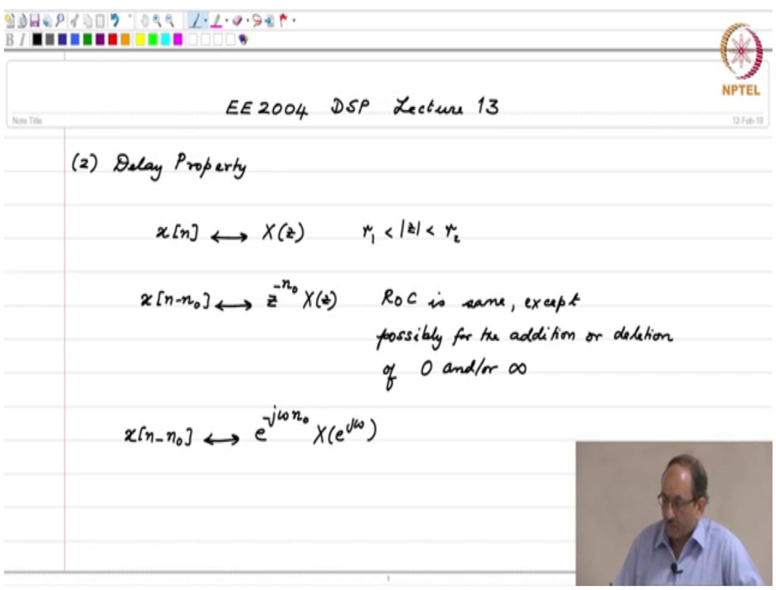
Digital Signal Processing

Prof. C.S. Ramalingam
Department Electrical Engineering
Indian Institute of Technology, Madras

Lecture 28:
Properties of the Z - transform (2)
-delay property

We have been looking at Z-transform properties and we looked at the very first property namely, linearity. We will now look at the next one, and this is the delay property.

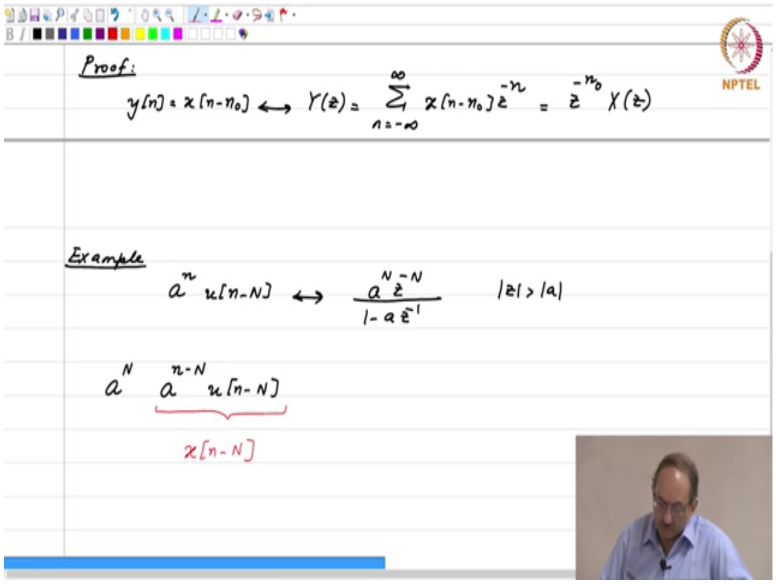
(Refer Slide Time: 00:27)



So, we have $x[n]$ having Z-transform $X(z)$ and in general its RoC is of the form $r_1 < |z| < r_2$. We want to see how the Z-transform of a related sequence namely $x[n-n_0]$ is related to the original sequence's Z-transform and the property is $x[n-n_0] \longleftrightarrow z^{-n_0}X(z)$. And, the RoC is same except possibly for the addition or deletion of 0 and or ∞.

And, the intuition behind this is, if you have a causal sequence and then if you delayed by some samples to the left, so that you have no samples corresponding to negative indices, then in the earlier case, 0 is not part of the RoC, but ∞ is. If you make it non-causal, then ∞ gets eliminated from the RoC. And, the corresponding DTFT property is this, all you need to do is you need to replace z by $e^{j\omega}$; therefore, this becomes $x[n-n_0] \longleftrightarrow e^{-j\omega n_0}X(e^{j\omega})$.

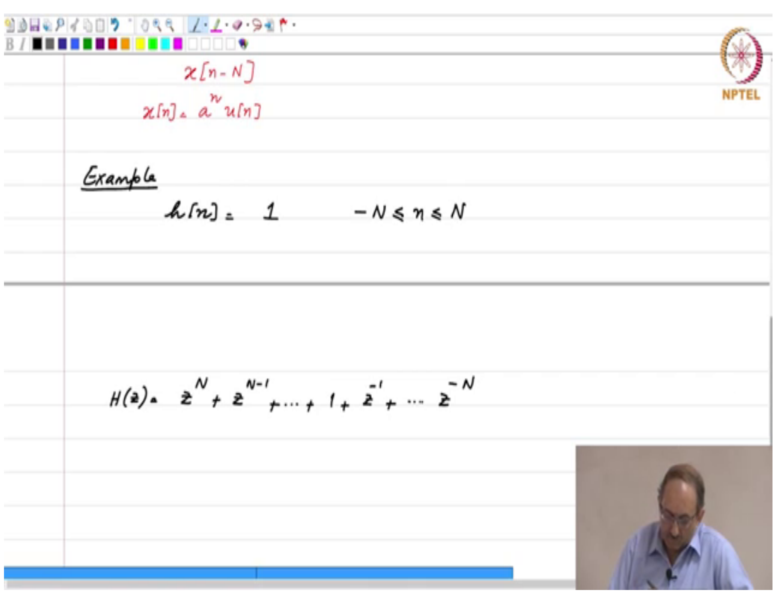
(Refer Slide Time: 03:14)



And, that is a simple application of this. Remember, when we look at the linearity property, we had given $a^n u[n-N]$ as one of the sequences and we had mentioned that the transform of this was $\frac{a^N z^{-N}}{1-az^{-1}}$, $|z| > |a|$. You can of course, very easily get this from first principles, but using the delay property this becomes really easy.

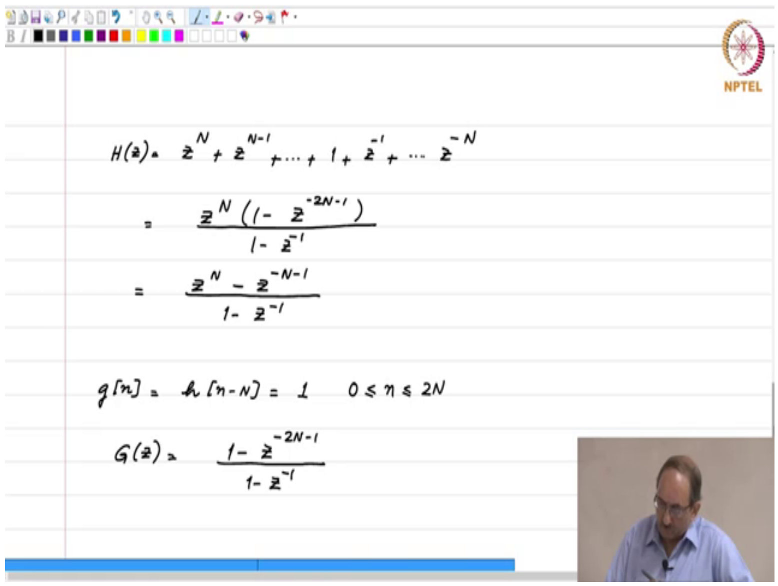
By the way, the proof of the delay properties just almost one line. So, if $y[n] = x[n-n_0]$, the Z-transform of the shifted sequence is $Y(z) = \sum_{n=-\infty}^{\infty} x[n-n_0]z^{-n_0}$ and then all you need to do is put $l = n - n_0$ and immediately this simplifies to $z^{-n_0}X(z)$. Just one line proof. And to derive the transform of this, you can write $a^n u[n-N]$ as $a^N.a^{n-N}u[n-N]$. So, nothing has changed.

(Refer Slide Time: 05:39)



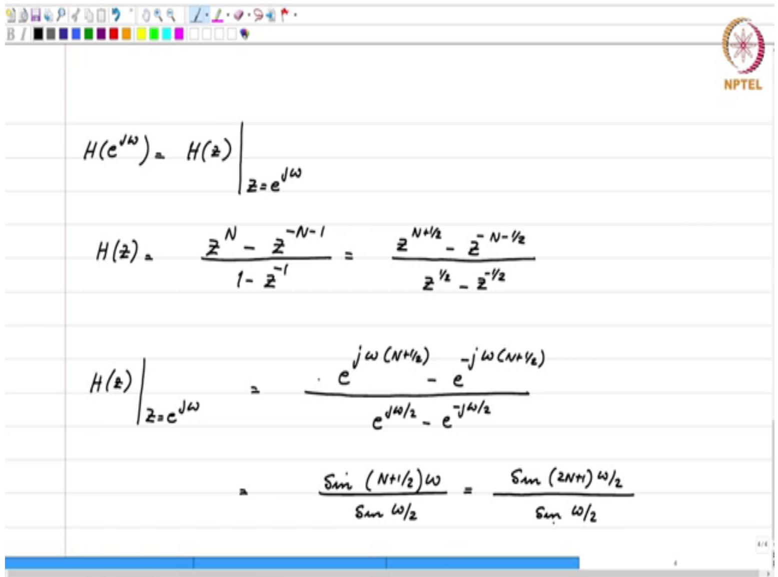
So, this you can think of $a^{n-N}u[n-N]$ as $x[n-N]$, where $x[n]$ is $a^n u[n]$ and hence you are able to see that this is nothing, but a^N and $x[n-N]$ has Z-transform $z^{-N}.X(z)$, $X(z)$ of course, is $\dfrac{1}{1-az^{-1}}$. So, the delay property helps you get this transform immediately, $a^n u[n-N] \longleftrightarrow \dfrac{a^N z^{-N}}{1-az^{-1}}$. Another example, suppose if you had $h[n] = 1, -N \leq n \leq N$, $H(z)$ of course is $z^N + z^{N-1} + \ldots + 1 + z^{-1} + \ldots + z^{-N}$ and this can be written as $\dfrac{a(1-r^N)}{1-r}$, $r = z^{-1}$.

(Refer Slide Time: 07:07)



And this is nothing, but $\dfrac{z^N - z^{-N-1}}{1-z^{-1}}$. And, if you had $g[n] = h[n-N]$. This of course, is 1 now for $0 \leq n \leq 2n$. All we have done is, we have shifted this non-causal sequence by N samples, so that it now begins at $n = 0$. And, $G(z)$ is, all you need to do is you need to multiply the previous transform by Z^{-N}; therefore, this is nothing, but $\dfrac{1-z^{-2N-1}}{1-z^{-1}}$.

(Refer Slide Time: 08:51)



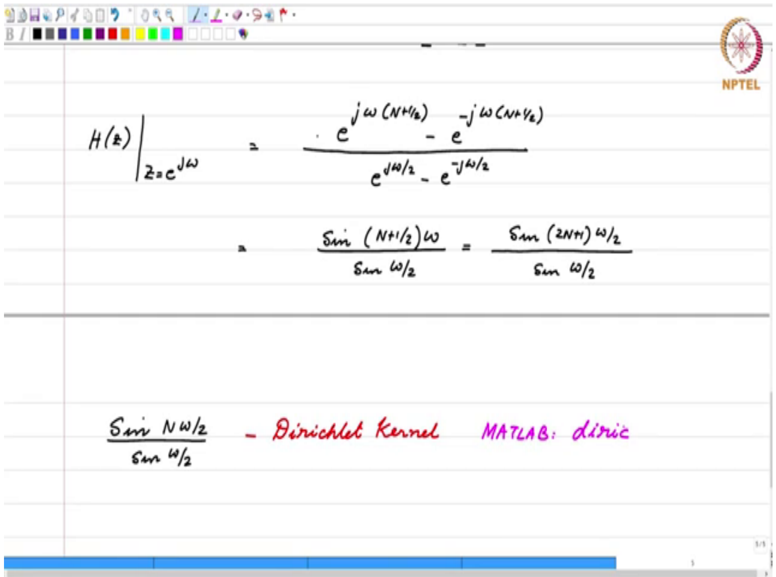
We can also look at the corresponding Fourier transforms. So, this is nothing but $H(z)$ evaluated at $z = e^{j\omega}$. You can do this, because this after all is a finite duration sequence and the RoC does indeed contain the unit circle, because the RoC is the entire z-plane except 0 and ∞. And, for this particular example, you can write the given $H(z)$ is $\frac{z^{N} - z^{-N-1}}{1 - z^{-1}}$.

Let me multiply the numerator and the denominator by $z^{1/2}$. So, this becomes $z^{N+(1/2)} - z^{-N-(1/2)}$ and denominator if you multiply by $z^{1/2}$ this becomes, $z^{1/2} - z^{-1/2}$. And the reason for doing this is, we want the DTFT of the given sequence, therefore, now if you replace z by $e^{j\omega}$, this becomes $\frac{e^{j\omega(N+(1/2))} - e^{-j\omega(N+(1/2))}}{e^{j\omega/2} - e^{()}}$.

Student: Minus.

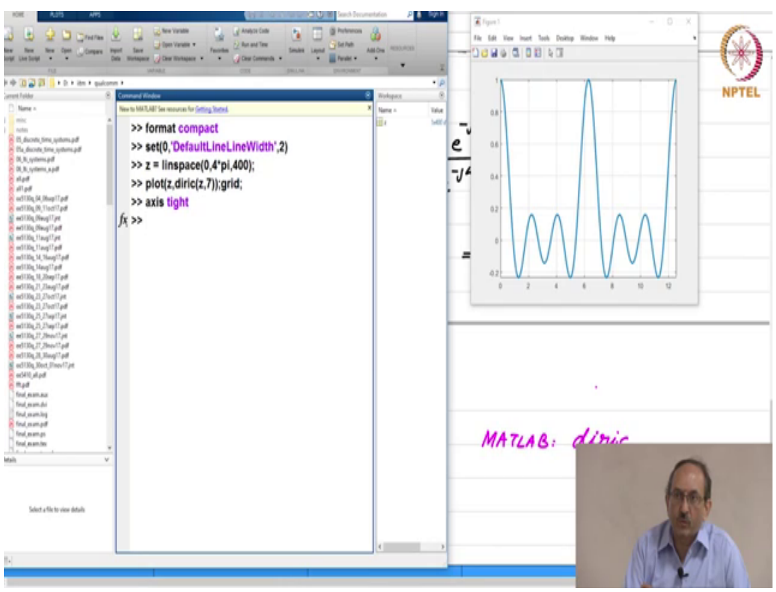
$H(e^{j\omega}) = \frac{e^{j\omega(N+(1/2))} - e^{-j\omega(N+(1/2))}}{e^{j\omega/2} - e^{-j\omega/2}}$. And this is nothing, but $\frac{\sin\big((N+(1/2))\omega\big)}{\sin(\omega/2)}$. And this is also commonly written as $\frac{\sin\big((2N+1)\omega/2\big)}{\sin(\omega/2)}$.

(Refer Slide Time: 11:40)



And, this function $\dfrac{\sin\left(N\omega/2\right)}{\sin(\omega/2)}$, this is called as the Dirichlet Kernel and MATLAB, you can look up the command *diric*, that evaluates this function.

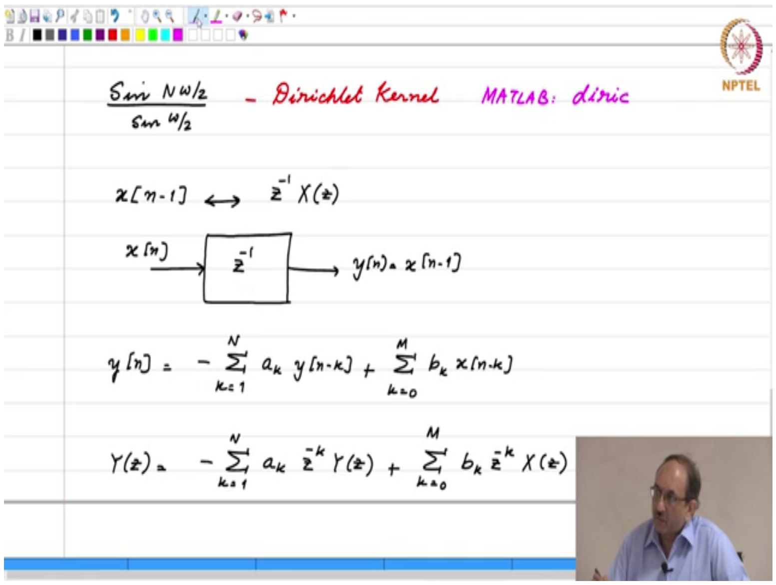
(Refer Slide Time: 12:32)



And, suppose I set z to be, linspace generates points between (a, b) and you can specify the number of points you want between a and b. And then you can plot z which is 0 to 4π and I am plotting $diric(z, 7)$. So, here $N = 7$. So, this is what this function is.

And, clearly this function is the Fourier transform of a sequence and this is 2π periodic. So, here I have plotted from 0 to 4π, so that is why you see two periods here. So, what was happening at 0 gets repeated at 2π. And MATLAB, if you say axis tight, it will plot the axis just limiting itself to the data range. So, this is the Dirichlet Kernel. So, this is $\dfrac{\sin\left(N\omega/2\right)}{\sin(\omega/2)}$. In continuous-time Fourier transform, you have come across the sinc function $\dfrac{\sin \pi t}{\pi t}$, and later when we do sampling, we will relate these two functions.

(Refer Slide Time: 15:06)

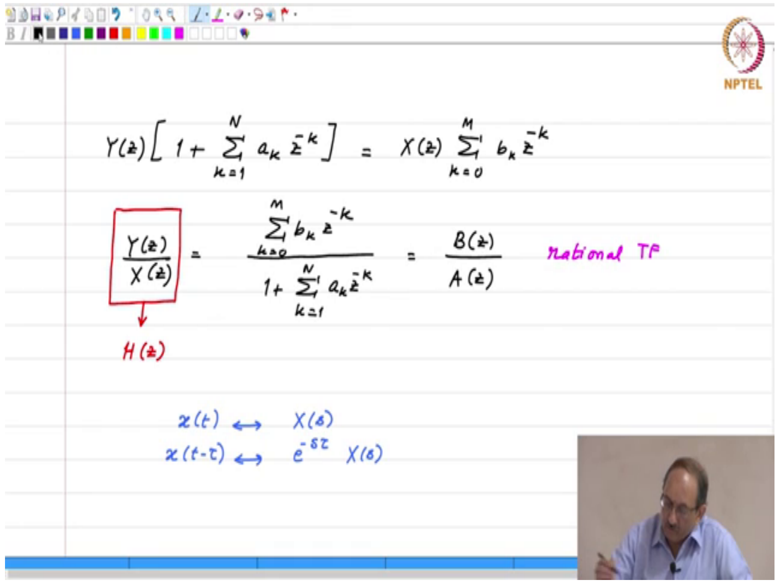


So, from the delay property, it is easy to see that $x[n-1] \longleftrightarrow z^{-1}X(z)$ and this property is used in block diagrams. Suppose, you had $x[n]$, and if the block diagram is labeled as z^{-1}, then this is a simple delay. Therefore, this $y[n]$ is nothing but $x[n-1]$. Instead of z^{-1}, if you had z^{-N}, then the delay will be by N samples.

So, if in block diagrams, if you see things like z^{-1} or z^{-k} and so on, then that stands for a delay by so many samples. And, if in $x[n-n_0]$, if n_0 were negative, then this advances the input rather than delaying it. So, n_0 can either be positive or negative.

Now, this delay property also can be used to point out one important consequence of systems described by linear constant coefficient difference equations. Therefore, suppose you had $y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k]]$; similar to what was happening in the continuous-time case where you took a system with LCCDE. But in this case, its differential equations, then you transformed it using the Laplace transform. So, here we will do something similar. We will transform this into the z domain. This after all is the input-output relationship.

(Refer Slide Time: 17:38)



So, now if you transform it to the z-domain, this becomes, $y[n-k]$ is nothing but $z^{-k}Y(z)$ i.e., $Y(z) = -\sum_{k=1}^{N} a_k z^{-k} Y(z) + \sum_{k=0}^{M} b_k z^{-k} X(z)$. Now, you can collect all terms involving $Y(z)$. So, this becomes $Y(z)\left[1 + \sum_{k=1}^{N} a_k z^{-k}\right] = X(z) \sum_{k=0}^{M} b_k z^{-k}$.

And, when I had introduced this difference equation, one of the questions raised was, why do you have a minus sign here? Right. And the only reason why a minus sign is used is, so that when you take it to the other side, this becomes plus. Otherwise, if this were a minus, you have to carry the minus sign throughout all the terms, that is only reason. It is just a cosmetic a notation.

And, this of course, can be written as $\frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$ and this is of the form $\frac{B(z)}{A(z)}$. And, just as in the continuous-time case, where you called $\frac{Y(s)}{X(s)}$ as $H(s)$ which was the system transfer function. Here, $H(z) = \frac{Y(z)}{X(z)}$ is the discrete-time system transfer function in the z-domain. And, the important consequence of this is, this is now a rational transfer function.

So, systems described by linear constant coefficient difference equations give rise to transfer function that are rational and you can actually work these steps backwards. Given a system with rational transfer function, there is always a corresponding LCCDE associated with it in the time domain. Therefore, LCCDE in the time domain gives rise to systems with rational transfer function, systems with rational transfer functions have a corresponding time domain implementation that involves linear constant coefficient difference equations, which is the exact counter part of LCCDE in continuous-time, where D stands for differential there giving rise to systems with rational transfer function in the Laplace domain.

So, now we are ready to answer the question as to why we are enamored by this class. This question has been raised more than once. The reason why this class is of so much importance to us is, this is the only class that can be realized in practice. In continuous-time case, if you had a system with rational transfer function $H(s)$ which is of the form $\dfrac{B(s)}{A(s)}$.

Suppose, the transfer function is something that you want, say for example, it is a low pass filter, you need to actually realize this in practice. And, any $\dfrac{B(s)}{A(s)}$ can always be realized in practice using R, L and C. Given an RLC circuit, what do you have been doing in that course is, you would replace R by R, L by sL and C by $1/sC$ and then you will write equations and then you will be able to come up with the system transfer function.

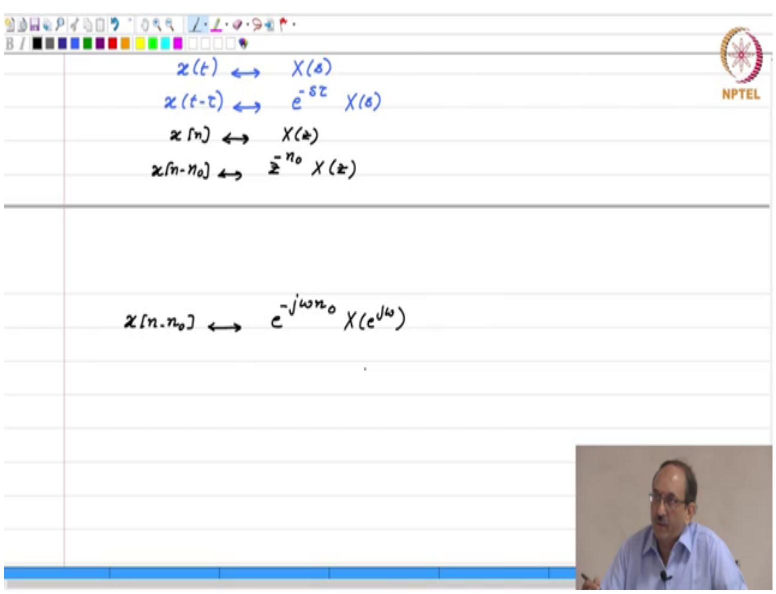
The converse also is true. Given an arbitrary transfer function of the form $\dfrac{B(s)}{A(s)}$, there exists an RLC realization of this. So, there is no use in having a nice transfer function that in terms of design does what you want it to do, but if you cannot realize it this is absolutely useless. But, if you did a course on network synthesis, you can always realize this in practice.

Similarly, here given a rational transfer function in the z-domain, it corresponds to a linear constant coefficient difference equation. And, in terms of realization, all you need are multiplier coefficients and delay elements. Once you have multiplier coefficients and delay elements, you can realize this difference equation in practice, all right. And this is the reason why this class is so important.

The other icing on the cake is it is not only just merely realizable, but it also gives rise to transfer functions that are extremely useful in practice. There is no point if this is realizable, but does not produce too many useful transfer functions. It does provide a large variety of transfer function that are extremely useful in practice and it can also be realized. And somebody made a remark when I had talked about $y(t) = x(t - \tau)$, right. I think you made the remark about this one.

Now, let us see the differences between this continuous-time and discrete-time. You had $x(t) \longleftrightarrow X(s)$. And, if you look at $x(t - \tau)$, this is $e^{-s\tau}X(s)$. So, the delay element in continuous-time, that transfer function is $e^{-s\tau}$.

(Refer Slide Time: 25:14)



On the other hand, if you had $x[n] \longleftrightarrow X(z)$, $x[n - n_0] \longleftrightarrow z^{-n_0}X(z)$. Here the delay element has transfer function z^{-n_0}, ok. So, if you compare these two, is there any point that kind of jumps out at you? Suppose, your original system had, if this $X(s)$ were rational, what about $e^{-s\tau}X(s)$ thing? Is it still rational or takes it out of that class? It is no longer a rational Laplace transform, whereas, here if $X(z)$ were rational of the form $\dfrac{P(z)}{Q(z)}$, this continues to be rational.

Student: Ok.

All right. So, a delay element in discrete-time which is z^{-n_0} is indeed a rational transfer function whereas, the corresponding counterpart in continuous-time is not. So, this is very important to keep in mind that the only class that can be realized in practice are the class of rational transfer functions. In continuous-time, you realize them using R, L and C, in discrete-time for realization, you need multiplier and delay elements. You can realize systems with rational transfer function.

Now, continue with the a delay property, let us look at this for the DTFT. So, this was $x[n-n_0] \longleftrightarrow e^{-j\omega n_0}X(e^{j\omega})$. One thing about the DTFT that is different from the Z-transform is, the Z-transform is a complex function of a complex variable. So, really you need four-dimensions to plot it, so we cannot visualize it.

On the other hand, the DTFT, the independent variable is omega which is real valued quantity. And, ω you need to let it vary only from 0 to 2π or between $-\pi$ and π because this is periodic. Therefore, the DTFT is a complex function of a real variable therefore, you can actually plot it in 3D; you need one-dimension for ω for the independent variable and two-dimensions for the dependent variable, real part and imaginary part.

Typically, what you have been using when you plot the Fourier transform in the continuous-time case is, you have been plotting magnitude versus frequency, and phase versus frequency. There also, the continuous-time Fourier transform is a complex function of a real variable. They are the real variable Ω went from $-\infty$ to ∞, ok.

Now, if you look at this. So, you have some original transform $X(e^{j\omega})$ and then when you shift this, it is getting multiplied by $e^{-j\omega n_0}$. So, if you want to view this geometrically, what is happening here? So, what does this do? This is a complex number and it is being multiplied by $e^{-j\omega n_0}$. So, what does it do in the complex plane?

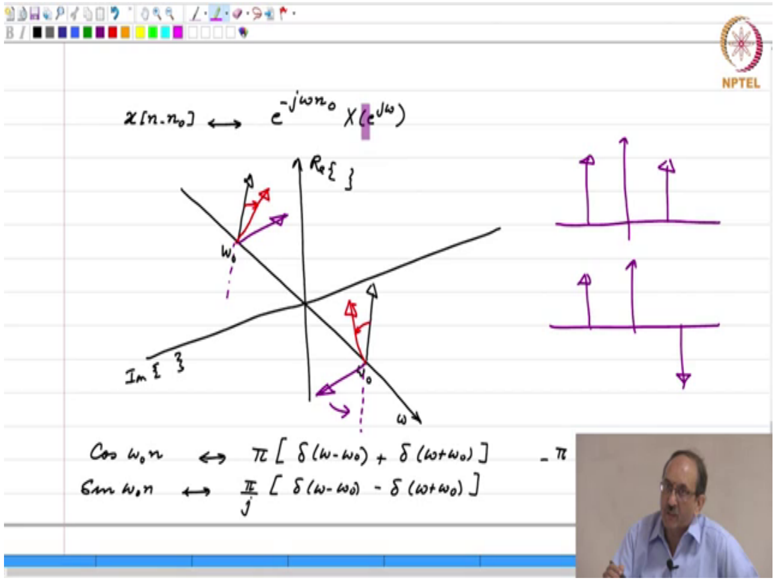
Student: Phase (Refer Time: 29:41).

Phase modulation not quite, geometrically what is happening?

Student: Rotate.

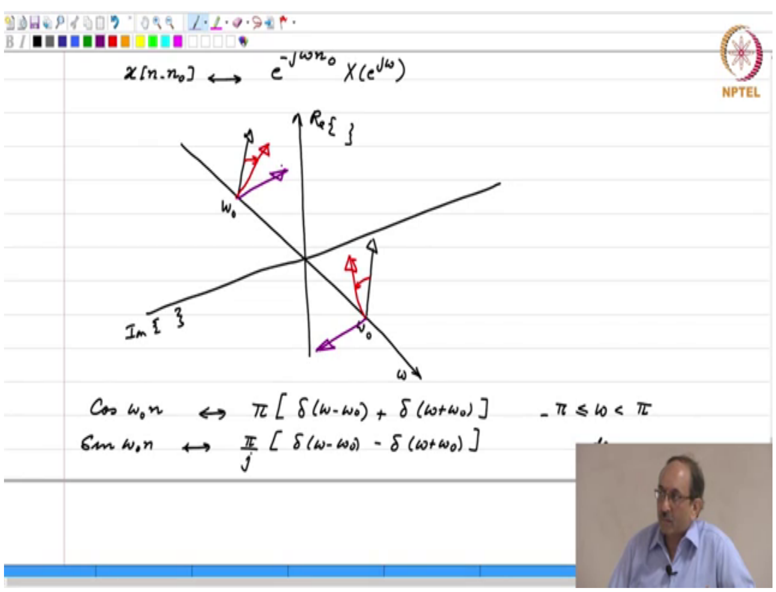
Very good, it rotates, right. So, the geometric interpretation of multiplication of two complex numbers is the magnitudes get multiplied and the phase angles get added, right. In this case, if one of the complex numbers is $e^{j\omega}$. All it does is, it cannot alter the magnitude, but merely rotate the complex number that it is multiplying, all right. So, now we can get a feel for what is happening here.

(Refer Slide Time: 30:23)



So, let us now plot the Fourier transform. Sequence we are interested in is this, later we will be looking at $\cos(\omega_0 n)$, we will formally derive the Fourier transform of this, but it in terms of equation this appears similar to what we had seen in continuous-time. So, this is $\cos(\omega_0 n) \longleftrightarrow \pi\Big[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)\Big]$.

(Refer Slide Time: 31:01)



And, $\sin(\omega_0 n) \longleftrightarrow \dfrac{\pi}{j}\Big[\delta(\omega-\omega_0)-\delta(\omega+\omega_0)\Big]$. So, this looks frighteningly similar to its continuous-time counterpart expression. The only difference of course, here is?

Student: (Refer Time: 31:27).

Very good. So, in the continuous-time case, you had an impulse at $+\Omega_0$ and $-\Omega_0$ and there you must have used the Ω notation. You may have also used the ω notation. But since here we want to compare and contrast CTFT and the DTFT, for the CTFT, we will use the Ω notation and ω for the DTFT.

But, in the CTFT case, you had only one impulse at $+\Omega_0$ and another impulse at $-\Omega_0$. In the DTFT, because by definition it is periodic, when you write something like this the periodicity is clearly not explicit. In fact, it is not periodic the way it is written. Therefore, what do you need to qualify this with to make sure that it is the DTFT?

Student: (Refer Time: 32:19).

You have to specify the range over which this is valid. So, this is between this.

Student: (Refer Time: 32:29).

Note that, in the case like $a^n u[n]$, its DTFT is $\dfrac{1}{1-ae^{-j\omega}}$. All you needed to do there was, you replace z by $e^{j\omega}$, therefore, $\dfrac{1}{1-az^{-1}}$ became $\dfrac{1}{1-ae^{-j\omega}}$. There you do not need to specify the range because by the very nature of the expression, if you replace ω by $\omega + 2\pi$, the periodicity is there in the form that it is written.

Now, coming back to this. So, let us call this as the ω axis and this is the real part and this is the imaginary part, ok. Or, the let me make this as the real part and this the imaginary part and this is omega. Now, for $\cos(\omega_0 n)$, this is the DTFT. So, it has an impulse at ω_0 and at $-\omega_0$. Therefore, at ω_0, I have an impulse; at $-\omega_0$, I have an impulse.

Now, what I am going to do is, I am, cosine; and remember this transform is purely real valued, so that is why the transform lies completely in the real part plane. And now, when I take cosine and shift it, then I start to add phase and remember the geometric interpretation is a rotation, all right. And hence, when I am rotating the this 3D plane, this impulse starts to rotate like this. So, now, this comes like this; so now, this is rotated by an angle like this. This impulse starts to rotate in this direction, right.

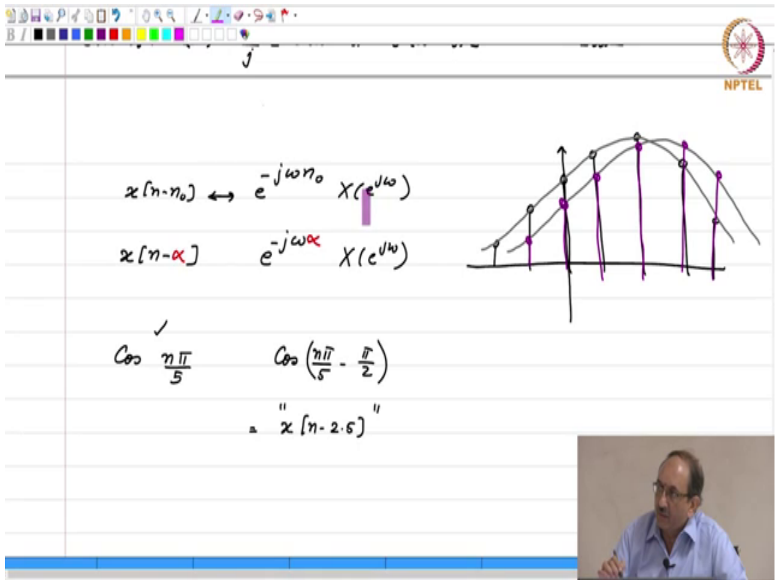
So, if these are the two impulses, then when I start to shift, this rotates like this, this rotates like this, ok. And now I keep shifting, I shift it enough, so that cosine has now become sine. So, now, what will happen is this impulse will be like this; this impulse will be like this, all right. So, the two impulses that were like this, they are now completely real. 90 degree shift, this will become like this, this will become like this. They are now completely lying in the imaginary plane, right. So, that again you see from this expression.

Therefore, the transform that you are used to seeing. So, the transform is, for cosine, you would have seen impulses like this. And for sine, you would have seen impulses like this, correct. So, this is nothing, but the view of this 3D picture as viewed in this direction. After phase shifting enough, so that it may it has become sine. The picture that you see here is nothing, but this 3D picture viewed from the top.

Now, if you keep shifting further, this sin will become −cosine. Therefore, what will happen is, this keeps further rotating and the impulses will be completely pointing downwards. Again, you shift further, −cosine will become −sine. Again, you keep shifting further, −sine will again, it will become cosine. So, when you have shifted by enough sample, so that cosine comes back to cosine, these two impulses would have rotated one complete rotation, all right. So, this is the geometric picture associated with the shift, all right. So, this kind of adds inside to the algebraic expressions.

So, any multiplication by a factor that is $e^{-j\omega n_0}$, merely twists the transform in the 3D complex plane. And clearly, the magnitude does not change because its mere rotation, magnitude is related to the length, length does not change if you rotate, ok.

(Refer Slide Time: 38:42)



$x[n-n_0] \longleftrightarrow e^{-j\omega n_0}X(e^{j\omega})$. Then, extending this, somebody had this idea in mind, instead of n_0, the person used α and now this became $n-\alpha$. So, the point about this is, it is not mere change of notation, no α can be any quantity not necessarily an integer, all right. So, once you have this delay property, it suggests that you can delay a sequence by fractional amount.

Now, what is the interpretation of this? The interpretation of this is as follows. So, we have a sequence like this, associated with this sequence, an underlying envelope continuous-time signal. So, you can think of something like this. And, because the underlying signal is continuous, suppose you take this and then delay the underlying continuous-time envelop by some amount, ok. Suppose you delay this by some amount, the point is the underlying continuous-time envelope can be delayed by any amount, not necessarily integer number of samples.

Now, suppose you resample the shifted continuous-times signal, but the point at which you sample remain the same, you sample exactly the same points. Therefore, now you will sample it here, here, here, here and so on. You can think of now the new samples, which are exactly the same locations the as the old, except now this in your mind, you can think of this as taking the original underlying continuous-time signal shifting it by not necessarily an amount that is integer multiple of the sampling period. You are shifting it by say 2.5 samples and then you are resampling it. So, you will get the newly marked samples.

So, you can interpret these new samples as taking the old ones and shifting it by fractional amount. So, this delay property helps you to bring in the notion of fractional shift because $e^{j\omega n_0}$; n_0 need not necessarily be an integer. So, what we will do is.

Student: Sir.

Yes.

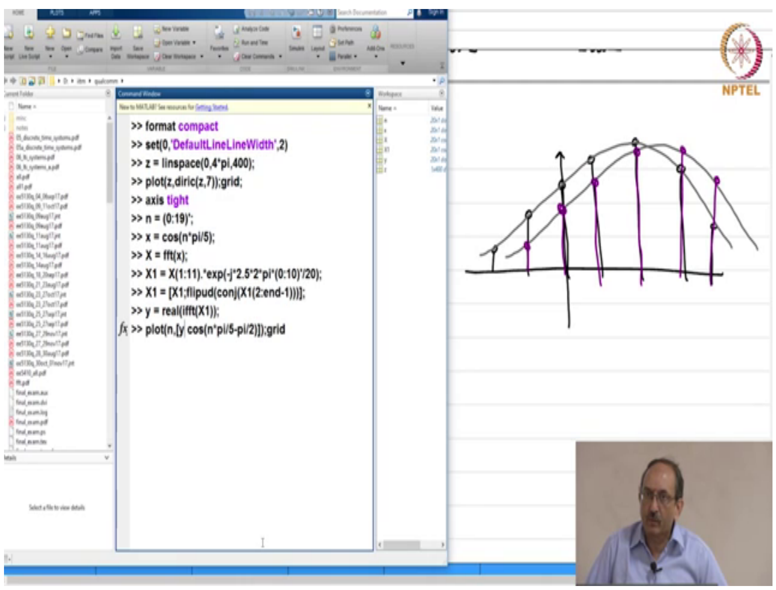
Student: Continuous part of the (Refer Time: 42:58) need not be the (Refer Time: 42:59).

Yes. So, the assumption is you have sampled it adequately, so that these samples represent an underlying continuous-time signal. So, that is the assumption that you are making. Through a discrete set of points, an arbitrary number of curves can pass through, but if you assumed that the signals band limited which is what you have studied in sampling in signals and systems, the so unique correspondence between samples and the underlying signal, continuous-time signal. So, that is the assumption here. So, what we will do is, remember we had $\cos\left(\frac{n\pi}{5}\right)$ and then we also looked at $\cos\left(\frac{n\pi}{5} - \frac{\pi}{2}\right)$ and this was $x[n-2.5]$.

So, now what we will do is we will take $\cos\left(\frac{n\pi}{5}\right)$, apply this property and then obtain this signal by letting α to be -2.5, and we will do this in MATLAB. And, the only thing that at this point you do not know is, right now ω is continuous whereas, on a computer implementation you have to implement it as samples. That is the transform cannot exist on a continuum of values from 0 to 2π, it has to necessarily be sampled. And, that we will come back to towards the last part of the course.

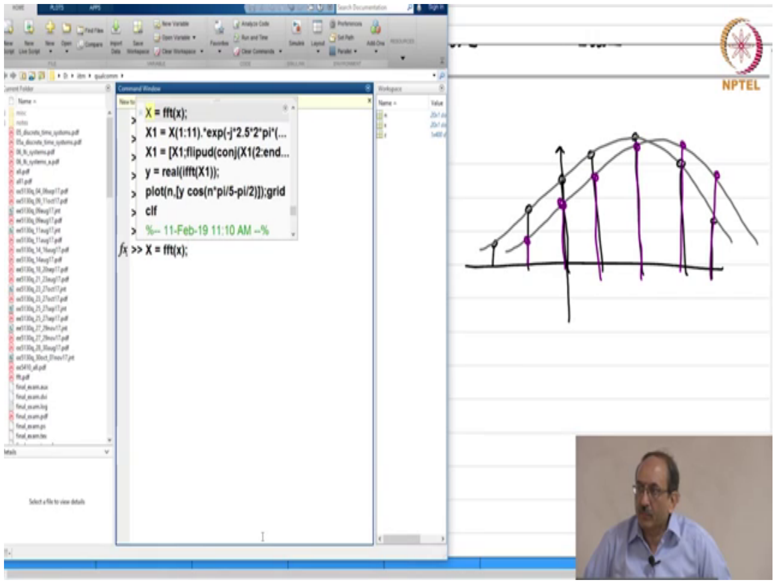
So, only that step may not be clear to you, other than that, all I am going to do is I am going to implement this and transform again cannot be a continuum of values. So, transform also has to be machine implementable. This multiplication also has to be machine implementable. So, the connection between what is being done in MATLAB and the DTFT, this is exactly what I am going to implement, ok.

(Refer Slide Time: 45:11)



So, I have N points from 0 to 19. So, I have taken 20 samples and then this is my signal $\cos\left(\frac{n\pi}{5}\right)$.

(Refer Slide Time: 45:35)



And, I am going to compute the transform. I need $X(e^{j\omega})$, that is obtained by this operation $fft(x)$. So, we later, we will talk about the DFT and we will talk about the FFT. So, this the machine implementation of the DTFT, ok. So, this is the transform, ok.

Now, I need to take the transform and multiply by $e^{-j2.5\omega}$, all right. And remember, we will see symmetry properties later. Similar to the continuous-time case, if the signal were real valued, the transform is symmetric in the CTFT. So, if you know the transforms from 0 to ∞, you know the transforms from 0 to $-\infty$. Similarly, here if you know the transform from 0 to π, you can recreate the transform from π to 2π, so that property will also exploit. We will formally state that property later. I am using that in advance now. So, now, I have this; so this is what I am doing here.

I have taken the first 11 samples. Remember, the sequence is 20 points, 0 to 19 contains 20 points. Later, we will see that the transform also contains 20 points. I am only taking the first half so, I am taking the transform from 1 to 11. That is, this is 0 to $N/2$, $N = 20$ therefore, $N/2 = 10$, 0 to 10. MATLAB is one based index therefore, I am taking 1 to 11. And $e^{-j()}$, here I have 2.5 and ω, I have discretized. So, this is what the discretization of the ω is; $2\pi * (0 : 10)/20$. This corresponds to the discretization of the ω variable. Details will become clearer, when you do the DFT and the FFT.

So, I have now created $e^{-j\alpha\omega}X(e^{j\omega})$. This is what I have created in this step. But, I have created this only from 0 to π because I have taken only one half of these spectrum. Now, the second half of the spectrum is nothing, but the complex conjugate of the first part. And time reversed, again we will see this later.

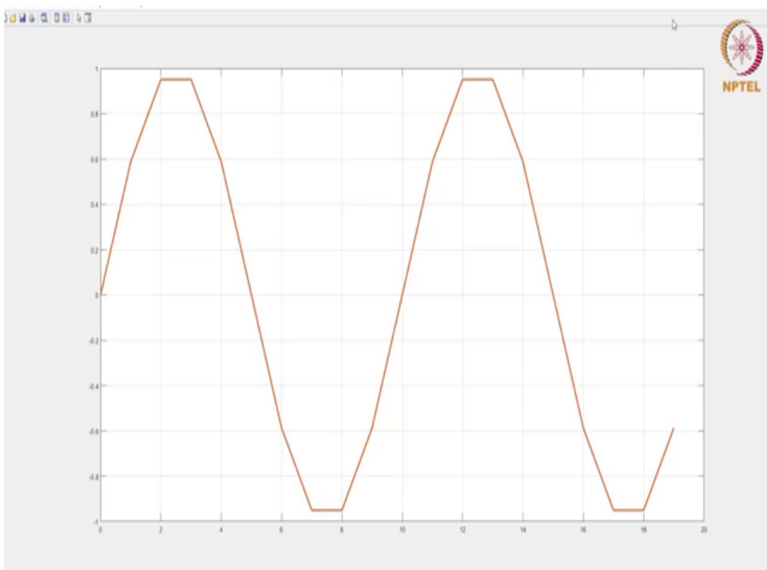
So, here, the $X1$ vector that was created earlier is there as it is. So, here what I have done is, the flip up down time reverses the vector, you will see the need for time reversal later and complex conjugation, conjugate does the complex conjugate part. And I am taking the first half, I am complex conjugating it and time reversing it.

But in the first part, I have this sample value at $\omega = 0$ and at $\omega = \pi$, those need not be repeated, because they are common. Therefore, that is why I am taking only samples from 2 to $end - 1$. Sampled number 1 would correspond to $\omega = 0$ which need not be repeated. Sampled number end, which is the last vector corresponds to π again which need not be repeated. So, this is the reason why I am taking only samples from 2 to $end - 1$, ok. And then I need to compute the inverse transform.

So, the $X1$ vector that was created earlier, I am now doing the inverse transform. Earlier I had used fft that was the counterpart for the DTFT in terms of machine implementation. The inverse DTFT is accomplished by $ifft$. And if things were ideal, because of the symmetry of the transform, the inverse transform would be real. But due to machine precision effects, the transform is not expected to be real, but the real part will have some very small machine ϵ values. To get rid of the small imaginary part, I am taking the real part.

And now I need to plot. So, here what I have plotted is, n from 0 to 19 and this is y; y was obtained by taking $\cos\left(\frac{n\pi}{5}\right)$ and then shifting it by 2.5 samples. So, if we have fractionally shifted the original sequence and we are now comparing it to the ground truth. Ground truth is really $\cos\left(\frac{n\pi}{5} - \frac{\pi}{2}\right)$. Remember, $\cos\left(\frac{n\pi}{5} - \frac{\pi}{2}\right)$ is $x[n - 2.5]$, whatever that means. So, this is the ground truth, and y is taking $\cos\left(\frac{n\pi}{5}\right)$ and shifting it in the frequency domain by applying a phase factor of 2.5. There you go, so one curve is right on top of each other.

(Refer Slide Time: 50:54)



So, the two curves here, one was y which has been obtained by shifting by 2.5 samples of $x = \cos\left(\frac{n\pi}{5}\right)$. I have shifted by 2.5 samples using this procedure, using the delay property. The other reference is really $\cos\left(\frac{n\pi}{5} - \frac{\pi}{2}\right)$, those lie right on top of each other. So, you can actually fractionally shift using the delay property.