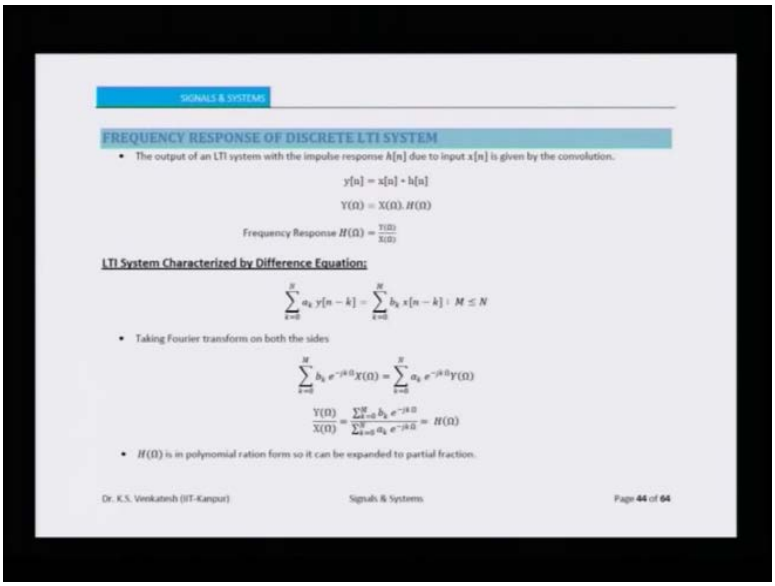
Signals and System
Prof. K. S. Venkatesh
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

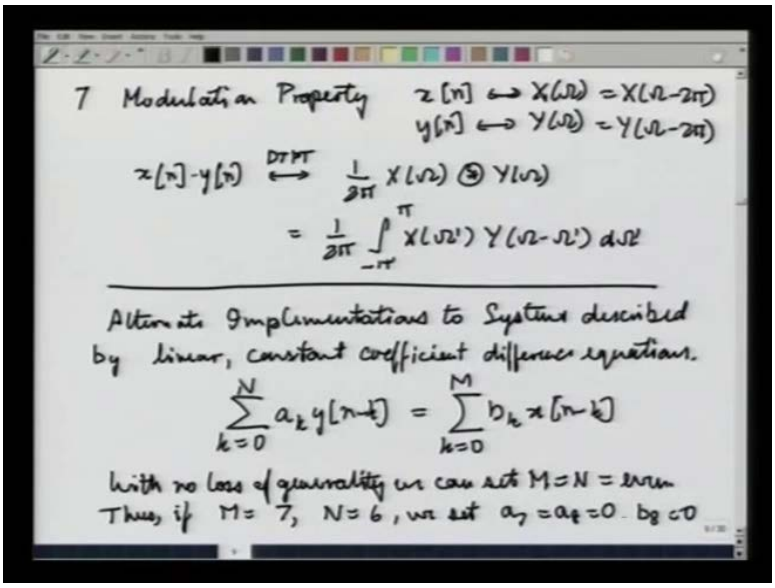
Lecture - 35
Frequency Response of Discrete LTI System

(Refer Slide Time: 00:14)



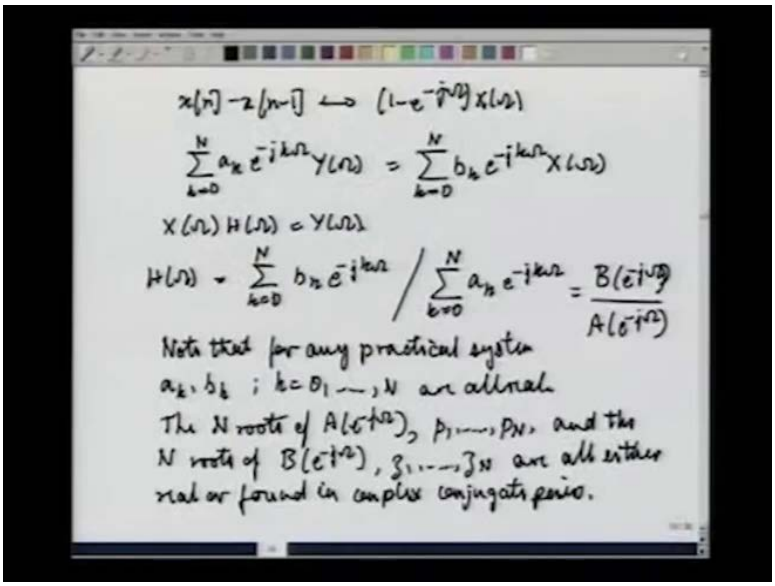
Now, move on to certain issues similar to what we discussed in the case of the continuous time period transform, namely the issue of finding alternate implementations to the systems described by in this case difference equations, there corresponding to differential equations. In short, we want to find constant, linear constant coefficient difference equations recall that the standard form of the linear constant coefficient n eth order linear constant coefficient difference equation was the following. k equals 0 to N a k y n minus k given equal to be summation over k equal to 0 to M d k x n minus k. By a series of arguments of the sort, we adopted in the case of the differential equations study for alternate implementations after this introduction of the continuous time period transform.

(Refer Slide Time: 00:49)



Here to we will argue that with no loss of generality we can set M equals N equals even. Thus for example, if M equals 7 N equals 6 we set a 7 equals a 8 equal to 0, b 8 equal to 0 and we get what we want. We get 8 as the order of both sides of this difference equation. As with the study of difference differential equations in the context of the continuous time period transform, here to we begin by applying the d t f t to both sides of the difference equation.

(Refer Slide Time: 04:32)



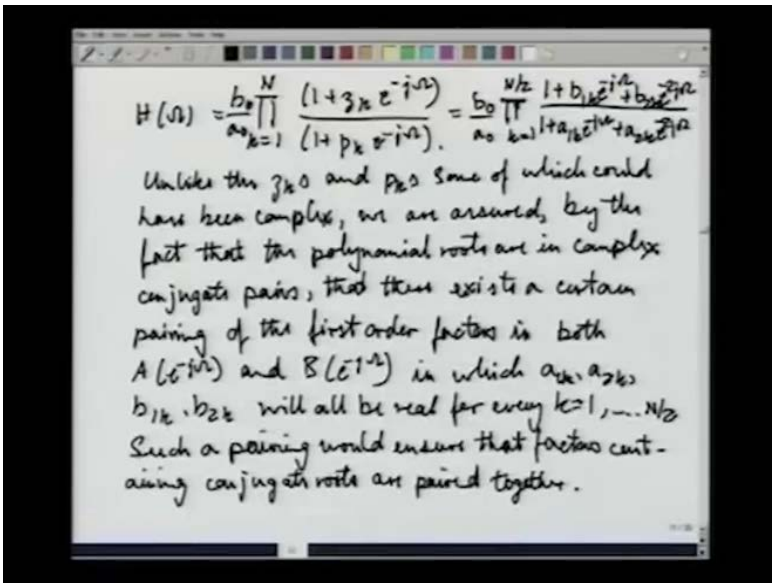
So, taking the DTFT of both sides and using the fact that x n minus x n minus 1 transforms to 1 minus e to the minus j omega x omega. That is that is the Fourier transform of the first difference. We will say that for the entire difference equation, we get summation k equals 0 to M to N a k e to the minus j k omega times y omega equals summation k equals 0 to n again b k e to the minus j k omega x omega. Now, invoking the convolution property which says that x omega convolved x omega times h omega equals y omega.

We can write h omega equals, so we get it as a ratio of polynomials in e to the minus j omega with coefficients a k and b k, which I will briefly write as capital B e to the minus j omega divided by capital A e to the minus j omega. This is what you need, this is what you get? Now, note that for any practical system a k and b k for k equals 0 to n are all real, we only have real coefficients. Now, when all the coefficients are real if you look at this two polynomials b e to the minus j omega and a e to the j omega and examine the roots the n roots of these two polynomials, what do we expect to find?

So, the n roots of A e to the minus j omega and B of e to the minus j omega are both are all well. Let me give their names the n roots of A to the e to the j omega, which we will call p 1 onto p N and the n roots of B of e to the minus j omega namely z 1 to z N are all either real or found in complex conjugate pairs. We said this we said exactly the same thing when we were studying the differential equations under the CTFT and the reasons in both cases are also the same. It is simply that if the roots are not all real or found in complex conjugate pairs.

Then when you re expand the factorized form where the roots are evident to us. It will no longer be the case that the coefficients a k and b k will be purely real as they expect to be as they are expected to be. In short is a k and b k have to be all real there is no choice, but for these coefficients for these roots to be real are in complex conjugate pairs. So, writing it in factorized form we will write as follows.

(Refer Slide Time: 10:25)



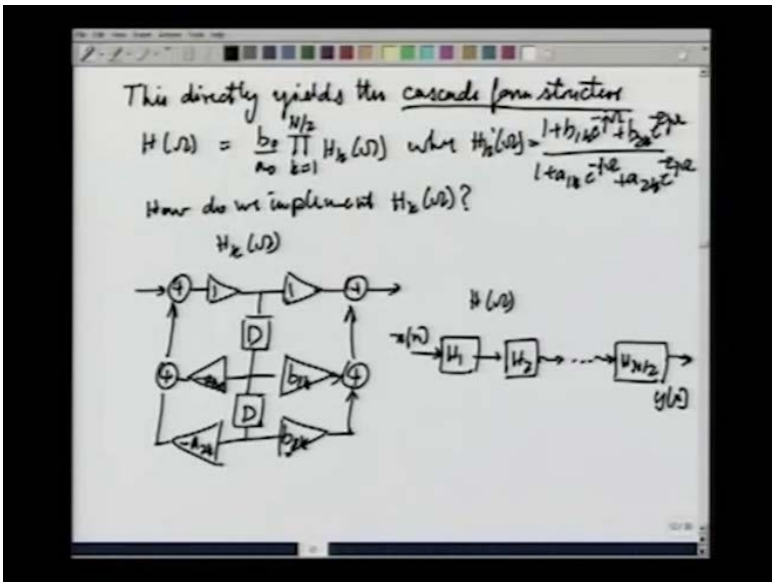
H omega equals product k equals 1 to N multiplied outside b 0 by a 0 1 plus z k e to the minus j omega divided by 1 plus p k e to the minus j omega. But having these factors in first order factors in the form of first order factors is not particularly convenient, because the sum of the z case and p case as we said might be complex. We do not know how to implement complex coefficients in the real world system. So, we resort to living with a second order a set of second order factors just like again what we practised in the context of the differential equations under the CTFT.

So, we will rewrite this as follows b 0 by a 0 product k equals 1 to N by 2, because now there are only N by 2 second order factors. We will write one1 plus, I will say b 1 k e to the minus j omega plus b 2 k e to the minus 2 j omega divided by 1 plus a 1 k e to the minus j omega plus a 2 k e to the minus 2 j omega. This is what we get now unlike with the z case and the p case, which could have been complex or other some of which could have been complex.

We are assured by the fact that the polynomial roots are in complex conjugate pairs; that there exists a certain pairing of the first order factor of the first order factors in both A e to the minus j omega and B e to the minus j omega in which a 1 k a 1 k a 2 k b 1 k b 2 k will all be real for every k k going from 0 to N by 2 sorry 1 to N by 2? Indeed in this that pairing would be characterized by the fact that conjugate roots factors containing conjugate roots would be paired together.

Such a pairing would ensure that factors containing conjugate roots are paired together. If you did that you would get this kind of decomposition into second order factors with real coefficients. So, with this formulation available, we have only a few steps to go to demonstrate, two new kinds of alternate implementations; the first is a cascade form structure.

(Refer Slide Time: 17:25)

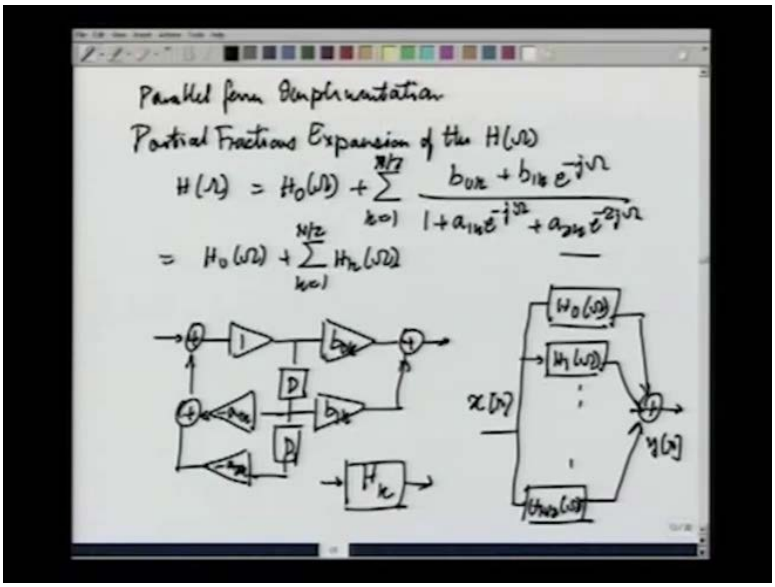


This directly yields the cascade form structure H omega equals b 0 by a 0 pi k equals 1 to product k equals 1 to N by 2 of H k of omega, where H k of omega equals 1 plus b 1 k e to the minus j omega plus b 2 k e to the minus 2 j omega over 1 plus a 1 k e to the minus j omega plus a 2 k e to the minus 2 j omega. Now, how do we implement? Just this much? How do we implement H k of omega H k of omega? Remember that these are discrete systems and there is no problem employing delay blocks the counter part of delay blocks in the continuous time system case was the differentiator.

We did not want to have a differentiator because of several practical reasons like its tendency to amplify noise, but here. Therefore, there we had to use integrators here there is not an issue, so we can easily implement h k of omega as follows you getting the input here we will write b 2 k here we write b 1 k. So, we had all this on this side on this side we have minus a 1 k and here we have minus a 2 k this gives you H k and H omega will then be implemented as x n going into H 1 followed by H 2 of omega.

Finally, followed by H N by 2 to get y N, this is the complete description of the cascade form implementation where the complete details. Now, as with the case continuous time period transform alternatives implementations, we here also have the parallel form implementation.

(Refer Slide Time: 21:46)



The parallel form implementation results from carrying out a partial fraction expansion of the polynomial rational form H omega fine. This is how we get the formulation required for the parallel form implementation. It comes out as follows H omega equals H 0 of omega, which is the quotient term plus summation k equals 1 to N by 2 of b 0 k plus b 1 k e to the minus j omega over the same second order factor containing real coefficients found in the denominator of the cascade form namely 1 plus a 1 k e to the minus j omega plus a 2 k e to the minus 2 j omega.

So, in order to carry out this kind of an implementation in parallel form, we construct each of these partial components found in the summation. There are totally n by two of them and finally, add it to x 0 of omega and the whole thing will give you the complete system H omega. So, in order to implement just this one particular H k of omega, I will rewrite this as summation H H 0 of omega plus summation k equals 1 to N by 2 of H k of omega.

So, what does what form does h k of omega take very similar to the previous case, we can use delays instead of summers. And so you get b 0 k 1 delay and another delay b 1 k

minus a 1 k, and here you have minus a 2 k; this is h k, the overall system would be implemented by taking x t, sorry x N, and passing it to several parallel blocks starting with H 0 of omega H 1 of omega. So on until you have here H N by 2 of omega and then adding all these up to get y N, this is the parallel form implementation again.

It is worth pointing out that the parallel form implementation and the cascade form implementation differ considerably from the direct form one and direct form two implementations. They would not have been possible to cons eve of unless, we had a mechanism to factorize the n eth order differential equation into smaller second order forms and that became possible only by the invention.

In this case of DTFT as earlier, we had said that the similar alternate implementations of the differential equations became possible only with the invention of the period transform, the continuous time period transform. So, in both cases this new transforms have paved the way for these new kinds of implementations. That last discussion, that last remark completes our discussion of the discrete time period transform.