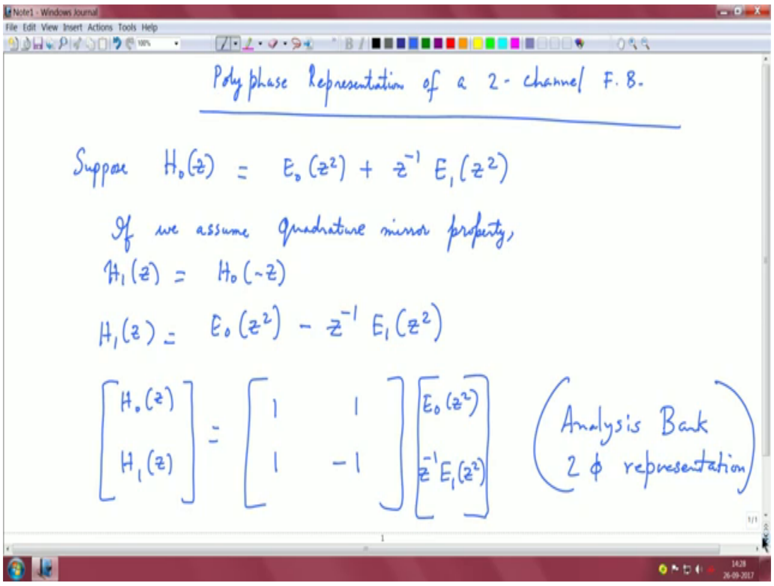
Mathematical Methods and Techniques in Signal Processing - I
Prof. Shayan Srinivasa Garani
Department of Electronic Systems Engineering
Indian Institute of Science, Bangalore

Lecture – 48
Polyphase representation of 2-channel filter banks, signal flow graphs and perfect reconstruction

So, let us consider the polyphase representation of a 2 channel filter bank right. So, let us start with a base filter.

(Refer Slide Time: 00:23)



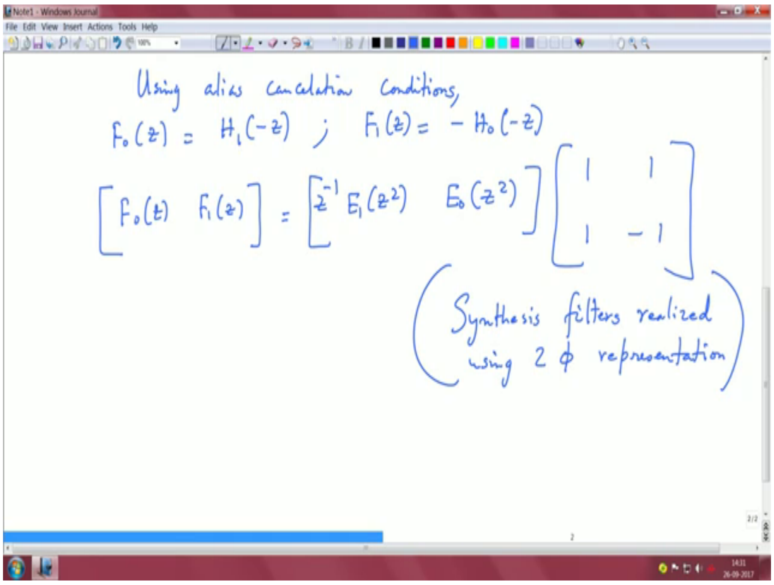
H naught of Z and we can write this as E naught of Z square plus Z power minus 1 that is a delay E 1 of Z square. Now, this is basically a 2 phase representation for the filter H naught of Z.

If the assumed quadrature mirror property then H 1 of Z is basically H naught of minus Z. So, I can write H 1 of Z in terms of polyphase representation as E naught of Z square minus a delay with E 1 of Z square right. I am just replacing Z with minus Z in H naught right and that that is representation polyphase representation for the filter H 1.

So, now, I can stack the filters H naught of Z H 1 of Z, and combine them together as a column vector and express this in matrix form using the poly phase components. So, I have E naught of Z square then I have Z power minus 1 which is a delay E 1 of Z square.

So, I have 1 1 1 minus 1. So, this is the representation polyphase representation for the analysis bank. So, this is the analysis bank representation. And we can do this similarly for the synthesis bank. Basically we will use the conditions for the alias cancellation using alias cancellation conditions.

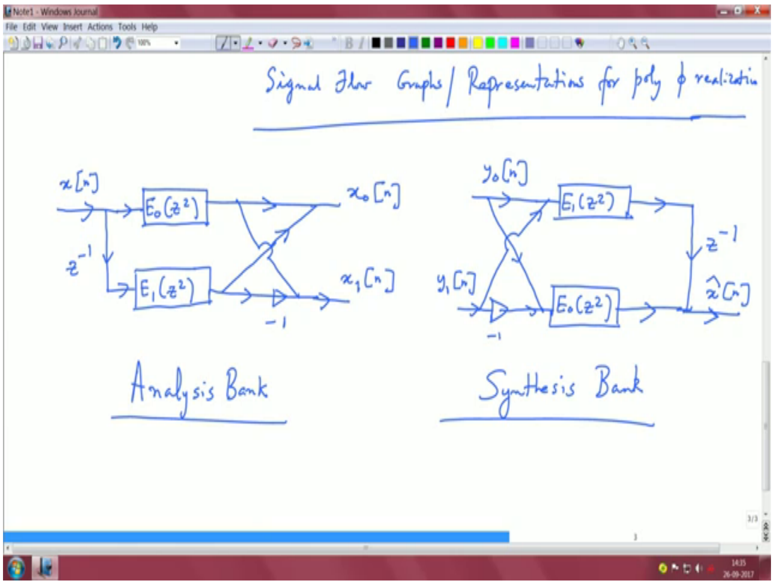
(Refer Slide Time: 03:29)



We have F naught of Z is H 1 of minus Z and F one of Z is minus H naught of minus Z; so using this representation for the synthesis filters. So, we can express them as follows F naught of Z F 1 of Z stacked in this row format is basically a delay E 1 of Z square, E naught of Z square with 1 1 1 minus 1. So, this is basically the synthesis filters realized using 2 phase representation, ok.

So, we has though the analysis and synthesis filters that can be realized using polyphase representation and we will see this is going to be very very useful for us because when we have to do multi rate operations and realizing these filter banks through possibly generalizations from 2 phase to m phase then we will find these representations very useful. And one of the things that you might have noticed is the form of representation right for the analysis somehow I wrote it in this form where I stacked them in the column format and synthesis. I did this in the row format and the hint would be the type 1 and type 2 phase representations that I would want type 1 polyphase for analysis and type 2 polyphase for the synthesis that is the sort of idea why we have stacked these filters in this form.

(Refer Slide Time: 06:20)



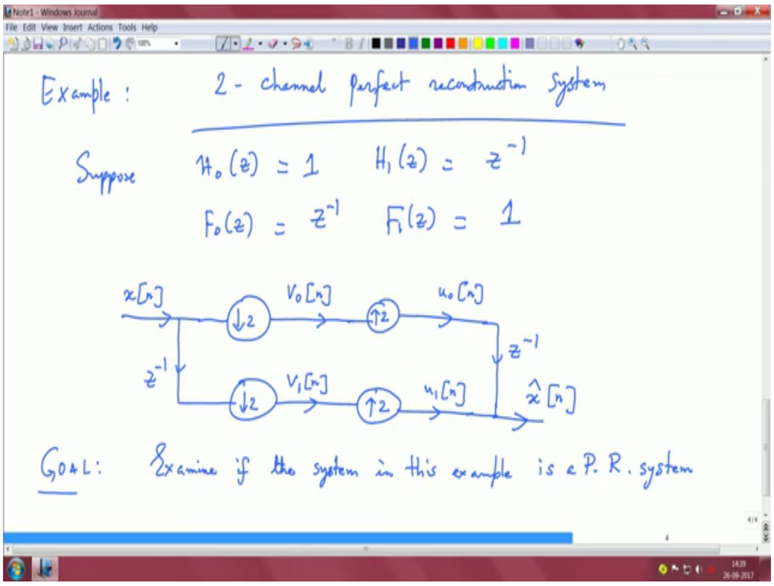
So, let us look at the signal flow graphs for this for realizing these filters for a polyphase realization. So, we will start with analysis bank. So, we have x of n this goes through the filter E naught of Z square, then I have a delay element this is filtered through the filter E 1 of Z square.

Now, I take this component here add them up with this branch I get x naught n then I have a gain of minus 1 here I add from the first polyphase component and I get x 1 of n. So, basically this is sort of a two band representation two sub bands we have and signals x naught and x 1 that are realized through polyphase representation, so this is the analysis bank.

Similarly, we have the synthesis bank. We have y naught n coming in this is filtered through the filter E 1 of Z square we also have a filter E naught of Z square in the parallel branch and y 1 it a gain of minus 1 will be fed here and then we take the component from y naught add it to this branch and here we directly take this sub band signal and add it here. So, we get, so basically two different sub band signals, one is the sum other is a difference goes through each of the filters and then the output of the filters are combined using a delay element to get you back your reconstructed signal x hat this is the synthesis part. So, all we have done is just the polyphase representation. We have not used any down sampling corrupt sampling right we have to bring in those components as well if we were to realize the entire two channel filter bank this is just a polyphase representation, ok.

Now, once we have understood the basic components we can connect all the dots before we wrap up with this slide just a small detail and arrow here has to be pointing in this direction. But before we begin the generalization I would like you to go through a simple example of a 2-channel perfect reconstruction system.

(Refer Slide Time: 11:11)



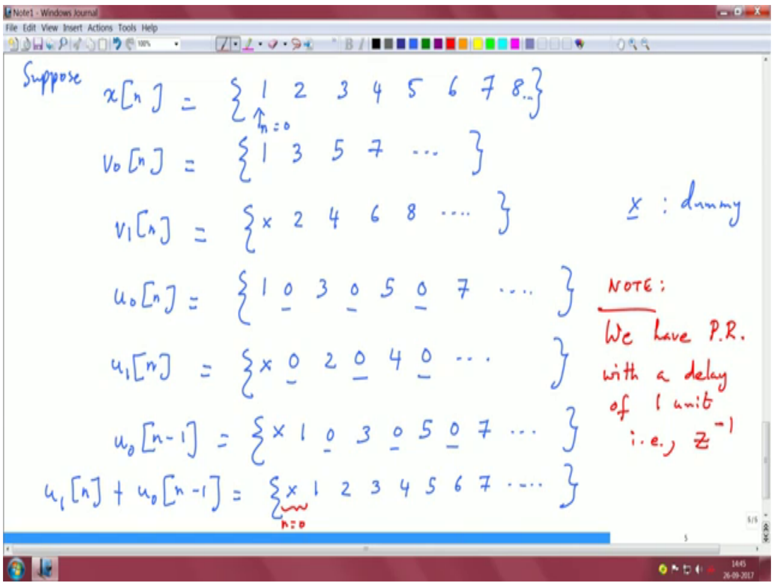
We will see an example. So, that you know what is happening purely in the time domain. So, suppose we have the analysis filters as follows H naught of Z is 1 absolutely no filtering is just again, H 1 of Z is a unit delay, F naught of Z is a delay, F 1 of Z is just a unit gain ok. So, we can sketch this we can sketch the schematic of the 2 channel system along with the down samplers and up samplers. So, we have x of n, then this goes through a down sampler because as unit gain here right H naught is 1. So, therefore, I am just not putting any filter here.

Now, in the other branch we have a delay which is corresponding to this filter H 1 of Z this goes through a down sampler at this point we have v naught of n and v 1 of n. Let us assume that we do not do any quantization etcetera. So, basically we just up sample by 2, on each of the two branches. We get u naught of n and u 1 of n and in the synthesis stage F naught is this delay. So, therefore, I have to put a delay element here because this is a filter and second branch this is just unity gain nothing. So, therefore, I am just leaving this branch as is so now, this is our x hat of l.

So, let us examine if the system above or I would say system in this example is perfect reconstruction is a perfect reconstruction system. Now, you may even wonder what is happening here, basically if I give you a signal I am just delaying it I am considering the even parts in one side other is the odd parts and then I am just expanding them and combining them using delay.

So, I think intuitively this should just be a perfect reconstruction system right, but let us just work out through an example figure out all the interim points v naught v 1 u naught u 1 etcetera and then see what is really happening.

(Refer Slide Time: 15:13)



So, let us take for granted that x of n is suppose x of n is a ramp maybe I just take about 8 values 1 2 3 4 5 6 7 8 dot dot dot, ok. Now, v naught of n is the down sample signal right. So, basically we have only the odd components 1 3 5 7 so on because we are considering only the alternative samples here alternate samples.

Now, v 1 of n is basically you take a delay element and then you down sample right. So, basically at the first step you may have some value then you land up with 2 4 6 8 dot dot dot right. So, basically if you reduce a delay the reference some something here which would have been some element in the past before time 0. Let us say that variable is x right you will have x if you delay that you have x 1 2 3 so on. Now, the down sample you get x 2 4 6 8, clear. So, this is a dummy variable indicating some value that could have been possibly before time equals 0. So, this is basically time equals 0, ok

Now, we have v 1 of n maybe in the interest of I will call this as n just to be consistent. So, n equal 0 n equals minus 1 we had some x here. So, therefore, and then when you down sample you get x 2 4 6 8. Now, u naught of n is the up sample version of v naught of n. So, therefore, we have to insert 0s. So, you have 1 0 3 0 5 0 7 these are all insertions, then u 1 of n is basically x 0 2 0 4 0 dot dot where these are the insertion points right, at every even time step I am just inserting these or time step I am inserting these.

Now, you have a delay element for u naught of n right because u naught goes through the filter F naught of Z and therefore, you get u naught of n minus 1 and u naught of n minus 1 is may be some element in the past I call some x right because is a delay therefore, something before in equals 0 is what I am going to get then I am going to delay it. So, I have 1 0 3 0 5 0 7 dot dot dot and then I take u 1 of n and I add it with u naught of n minus 1. I get some value I call it as dummy x then I have 1 2 3 4 5 6 7 I just can reconstruct by sequence, so this x is dummy. So, do not worry this is some variable x. So, therefore, this is x 1 this is x 2 and do not get confused with all that. So, x is basically a dummy variable for just please holding the value at time step n equals minus 1 and before then, ok.

So, we now have, we are reconstructing the sequence back, but with a delay. So, note we have perfect reconstruction with a delay of 1 unit and this is the delay and to say n equals 0 here and; that means, I am I am shifting the original sequence by 1 unit, ok.

So, now, if we were the real question is this is a very simple filter bank two channel filter bank satisfying the perfect reconstruction property and of course, you have considered we have considered only delays as some of our filters. But in practice it may not be the case because you may have to filter out certain components in the low pass some other components in the high pass, and then and then come up with reconstruction filters.

So, it is not very straightforward like the way we looked at the delay filter bank because the filters can be quite sophisticate. So, then it will be very useful for us to think what are the conditions for perfect reconstruction property. How do we design synthesis filters such that the overall transfer function is of the form C Z power minus n naught, where C is some amplitude which is not 0 and the output comes to you with the delay of n naught units ok. So, the design of such synthesis filters that can give you perfect reconstruction, I think this is of great interest.

So, we will stop here.