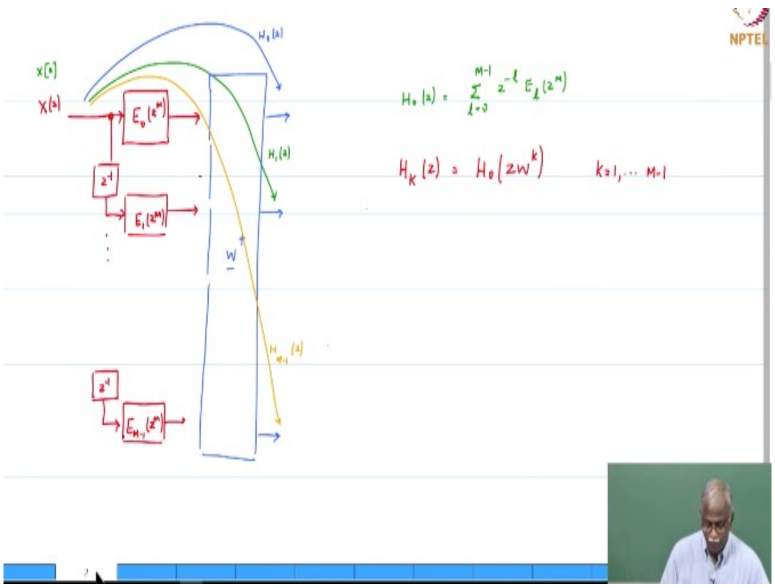
Multirate Digital Signal Processing
Prof. David Koilpillai
Department of Electrical Engineering
Indian Institute of Technology – Madras

Lecture – 19 (Part-1)
Transmultiplexer and Maximally Decimated Filterbanks – Part 1

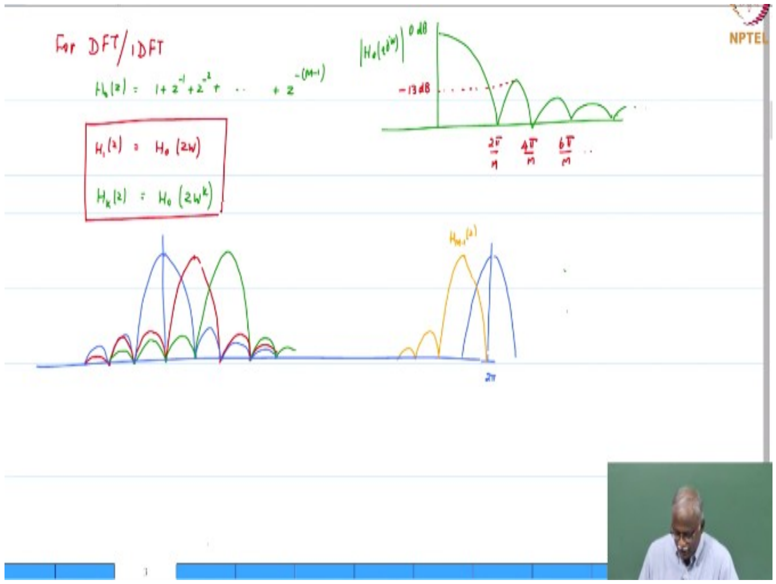
Good morning. We begin lecture 19. And in today's lecture, we will introduce the concept of transmultiplexes and also the topic of maximally decimated filterbank. So before that a quick review of the concepts that we have covered in the last class, so that we can build on the tools that we have for multirate signal processing.

(Refer Slide Time: 00:33)



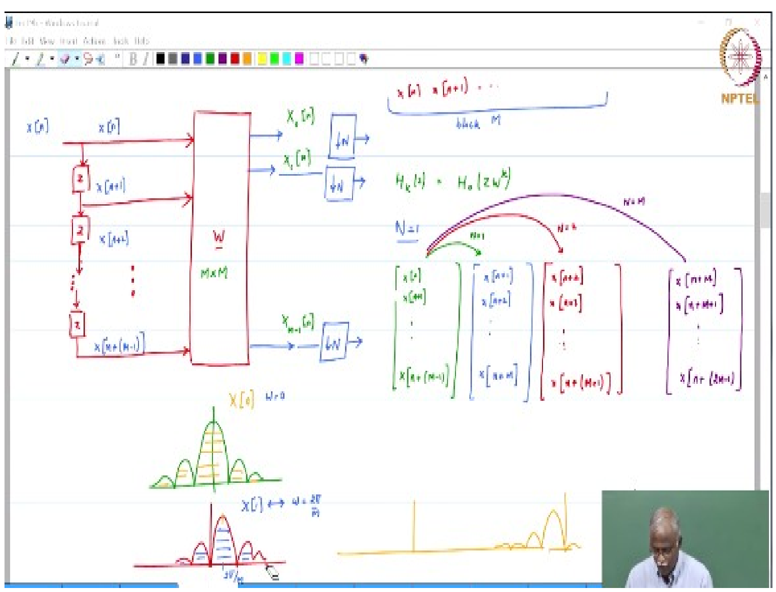
The general framework of a DFT filterbank, so keep this as a useful reference point so this would be a DFT filterbank and key elements are the fact that we have done the polyphase decomposition and have shown that the inverse DFT is the one that appears here. So we call it the DFT filterbank. We come back to this particular concept several times but right now in this particular discussion specific discussion we are interested in the concept in the context of the spectral analysis.

(Refer Slide Time: 01:16)



And therefore we said that if we set these polyphase components to be equal to one each of these polyphase components then we actually get a filterbank where the underlying filter is a rectangular window and it turns out and we showed that this is pretty much what we are doing when we take the DFT or the IDFT for that matter. So basically we have a bank of filters which are giving you readings which can be interpreted as the spectral analysis of the input signal, okay.

(Refer Slide Time: 01:53)



So we went so far as to say that the basic structure of the DFT that we compute, you take a stream which is in the form of x of n, x of n+1 you take a block M and then you compute its DFT and that block can be generated by means of passing the input signal through a series of advanced operators so you get x of n, x of n+1 all the way to x of n+M-1 that becomes your first block.

You take the DFT of it you should actually label this as the DFT coefficients but it is another sequence uppercase X, we call it X sub 0, X sub 1 you can think of it as the DFT coefficients I mean but basically it is mapping from this input vector to the corresponding output vector, okay. Now one element that I just want to add on to that in terms of concept is to introduce a down sampling by a factor of N; N is not the same as M. N and M are different.

Basically, this is a M by M DFT matrix. So this is a M by M matrix, okay. So I am taking an n point DFT but I am introducing a down sampling by a factor of N and very deliberately chosen to introduce that, okay. So my first vector is as follows is, x of n; x of n+1… x of n+M-1 that is my first vector. Now without ignoring the down sampling if I just asked you what is the next instant of time. What is the vector that is being introduced?

You can write down, let me use a different color so at the next time instant the vector for which you will be taking the DFT if you do not do anything in this structure will be x of n+1; x of n+2 because all of the samples would have moved by 1 unit. This the last sample will be x of n+M, okay and then so on so forth and maybe just write one more just for; so things become clearer when we actually do the explanation.

So next one would have been x of n+2; x of n+3…x of n+M+1, okay. Now please pay attention. If I set N=1 down sampling by a factor of 1 that means it is not doing anything, right. It basically allows all these samples to go. So if I set N=1 then my next vector is going to be this one, so this corresponds to N=1 if this is my first vector. Now if I set N=2 basically move the down sampler all the way to the left.

So that means x of n will be the first sample then x of n+1 will get removed x of n+2 plus will become the next. So basically this becomes the N=2 case you will skip one of those vectors, okay. Now when does it become a something of interest I would like to look at the case where

N=M, if I take N=M then that is the non-overlapping DFT, if you remember the non-overlapping one the next one will be x of n+M; x of n+M+1…x of n+2M-1. Okay.

So actually the structure that we have developed is completely flexible. You can choose to have any level degree of overlap that you want all the way from just shifting the next vector by just one sample or by M samples or by any other factor that you want. So this structure I just wanted to leave with you the thought that this structure is actually a very powerful structure. It is one that we can use quite extensively to analyze the various DFT computations that we that we do.

And it is a very flexible structure and it gives us all of the types of overlap. See sometimes you want to have a 50% overlap between the successive vectors so which means that you would choose N=M/2; M/2 will give you 50% overlap and of course you can have anything in between you can get you can have variations on that as well, okay well. Short-time Fourier transform, the question is, is this a structure that is used for? So the two things that; let me just introduce a blank sheet and then write down some to answer that question.

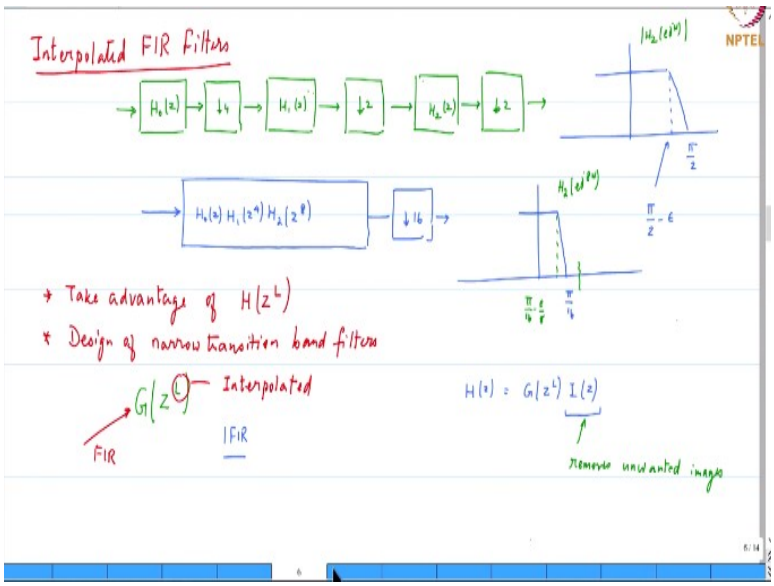
(Refer Slide Time: 07:27)

Okay, so short time Fourier transform has the following; short-time Fourier transform. One is you must decide on your segment length of a segment. So length of data, length of data, okay. That is the first one that you have to decide. The second one would be the window type. They prefer to window the data before you compute the short-time Fourier transform.

So to answer your question what you would have to do is determine the size of the dataset which means that the M dimension has to get fixed. And then you have to apply the windowing function the constants and then you take the FFT. So with these multipliers which will be the let us call them as W0 to WM-1 that is your windowing coefficients then you get the short-time Fourier transform.

So otherwise what you are doing is just a block Fourier transform processing, right. So good always feel free to ask a question so that is a short-time Fourier transform.

(Refer Slide Time: 08:33)



Okay, now the second concept that we had mentioned in the context of multirate applications is the notion of interpolated FIR. And interpolated FIR comes from the observation that if I have some sort of a cascaded implementation again I use the example I repeated the example from last time so that it would be familiar.

The equivalent filter can be written as H0 of z; H1 of z for basically moving all the down samplers all the way to the right and applying the noble identities. We get H2 z power 8. We focused in on what happens when you have H of Z power L. Basically, if you were to take H2 of Z and then if you call this as, let me just label this; if this is mod H2 e of j Omega okay. Now this one is H2 e of j 8 Omega, okay. I have shown only up to I am not shown the whole thing, okay.

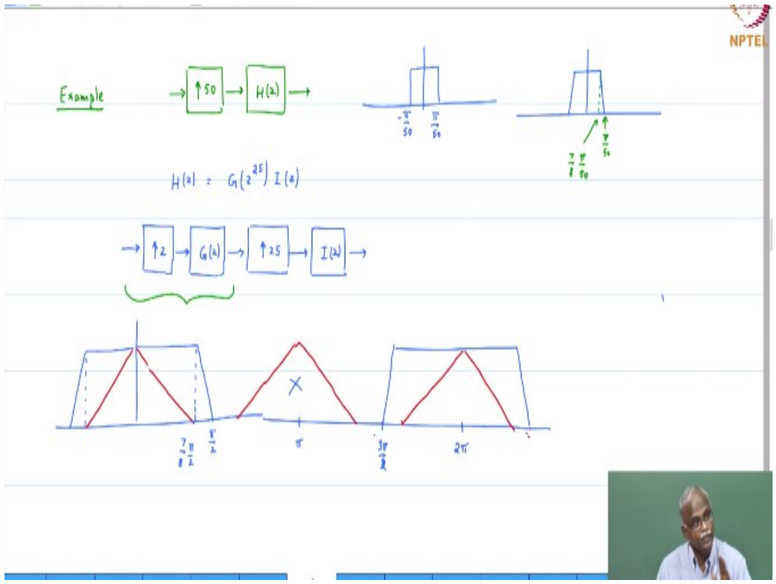
Basically it stops at pi/16 just to; so up to Pi/16 this is what is shown. So what we find is that when we do this Z power L where L is some integer you find that the spectrum gets compressed and that gives you an advantage because that also compresses the transition band any practical filter will have a finite transition band and this is a way to take advantage of the fact but it will generate images.

We have to get rid of the images and therefore the concept is that you will take advantage of the interpolation to get the sharp transitions and then introduce an additional filter. So basically we saw the example where you can design a filter H of Z as G of Z raised to the power L followed by an interpolator, okay. So now this; these are called by the way this type of design called Interpolated FIR Filters. Maybe I should write this is G.

So the term the name Interpolated FIR comes from the fact that if G of Z is an FIR Filter then if I do G of Z power L that Z power L comes as if I was doing interpolation of the impulse response. So FIR comes from the basic inherent property of the filter itself. The Z power L gives me the term interpolated so that is how we get the name Interpolated FIR. But Interpolate FIR by itself cannot satisfy the filter requirements.

You would have to have an additional filter here to get rid of the unwanted images, so this one is the one that removes the removes unwanted images. Okay. So maybe it is good to sort of reinforce this, unwanted images, let me just ask you to do a quick example.

(Refer Slide Time: 11:47)



The example that we would like to look at in the context of the Interpolated FIR technique is to have is to do an up sampling by a factor of 50 followed by a filter that would get rid of the images H of z. So first and foremost if this is the requirement I want to know what H of z ideally should look like. A brick wall filter from –pi/50 to pi/50. I have drawn it; you know with the reasonable amount of width. But in any scale that you actually do with if you draw it to scale you will find that pi/50 is a very narrow filter, okay.

So now if I cannot design ideal filters I have to go for practical filters so then I say that okay. No, actually what I have to design will have a finite transition. So this cut-off has to be pi/50, this is pi/50. But it has got a finite transition bands, so the transition band also has got where the passband ends. Let us say that I have taken it to be 7/8 pi/15. That is my specifications, please run it through some filter order estimation formulae.

I am sure you have done and assume some reasonable ripple for the passband and stopband. You will find that this will give you a very high order filter because the transition band is very sharp. Now of course that means you will have to have a lot of multipliers. On the other hand, if I were to take the design or implement the design in the following way where I say that H of z I would like to take advantage z power 25 and times I of z, okay.

If I were to design it in this form, then the actual implementation will look like moving the interpolate; the up sampling appropriately. I will implement 50 as an up sampling by 2 and up sampling by 25, so applying the noble identity this becomes G of z followed by up sampling by a factor of 25 then followed by I of z, okay. So we have implemented this. Now the question is, is there an advantage to designing it this way; will there be fewer multipliers involved and if so how is the whole system going to work together, okay.

So now look at this portion of the; this portion the first part of it. So this first one should be up sampling by a factor of 2; so if I were to look at this particular signal then it is a it is up sampling by a factor of 2 so that G of z should actually cut-off at pi/ 2, right G of z should cut-off at pi/2. And keep in mind that we are basically going to retain the same passband relationship so this would be 7/8 times pi/2.

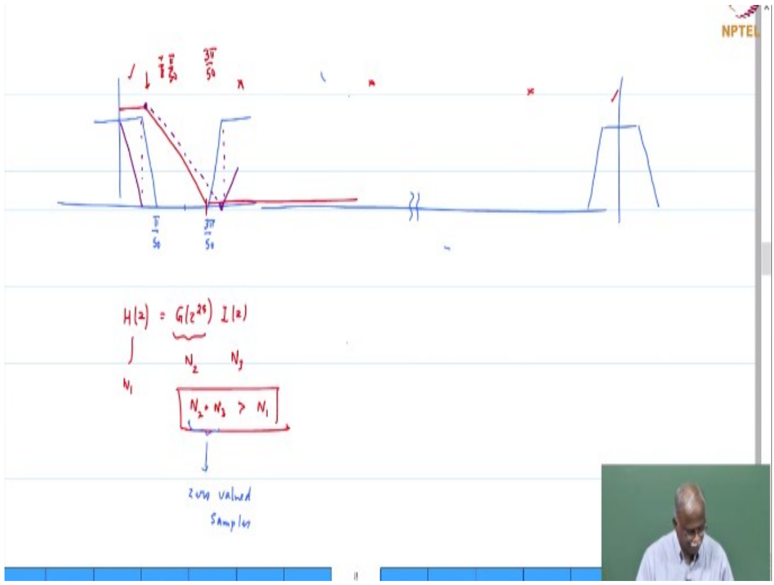
And of course if you were to draw it on the other side again you will see this is the behaviour, okay. Now this is pi 3pi/2 and 2pi. And of course you should have a corresponding image on the other side. Okay. So the reason we are spending a little bit extra time on this is to just to make sure that the advantages of Interpolate FIR method are actually visible from this aspect. Now if this is the scenario that we are trying to do then in all likelihood we always want the signal not to be distorted.

So in all likelihood the signal was probably in this range in all likely. So let us assume that that is the range so that its signal does not get distorted. So when I up sample by a factor of 2 and I pass it through this filter G of z, I want to remove the image; when I do the up sampling by a factor of 2 the signal will produce another image. Okay, now this; how did this image get killed because the blue filter removed it.

So the blue line actually kills this portion. So up; the input signal is the red spectrum up sampled by a factor of 2 you will have an image at pi apply a low-pass filter with pi/2 and 7/8 Pi/2 again this is a reasonable is not a very sharp filter and this will kill the image here, okay. Now the next stage of up sampling, what does it do to my G of z? G of z becomes maps to pi/50 and the passband automatically gets mapped to 7/8 times pi/50.

So without doing anything I have got the desired response and the key point is the; I of z has to remove certain images; we just have to make sure where the images are coming. So let me just do one more diagram and let us complete the discussion.

(Refer Slide Time: 17:53)



So after the up sampling by a factor of 50 there is a cut-off there is first image at pi/50. Where is the next image? 3pi/2 is the next frequency edge that divided by 50 so 3pi/50 is going to be the next edge. So let me just show it as this here. So this is going to be 3pi/50 that is the up sampling by a factor of 25 is going to produce images and it is going to produce several images, okay. And we just want to sort of leave the all of these and then go all the way to the last image.

So this has to be kept; this has to be retained; everything in between has to be killed, right. So I of z has to be a filter that does not distort you in the passband and gets rid of everything beyond the stopband. Okay. So and of course it will be it is a low-pass filter so basically it kills all of the other images.

So the requirements for I of z the passband edge is 7/8 times pi/50. The stopband edge is 3pi/50 again there is a much wider transition band than the original filter that we would have had to design. And basically this would be the requirements for I of z. So G of z power 25 times I of z will become my equivalent H of z.

But because I am using it to do the up sampling actually the poly; the interpolated FIR actually has a significant advantage because I will implement it in this fashion and also the filter requirements of the two filters G of z and I of z are relatively milder than the requirements on H of z directly, so one question, so basically if I design it in this fashion if I design H of z directly I will get a certain filter order N1 okay.

So this comes out to be eventually a filter order of N2; this is filter order N3, G of z power 25. So when I add N2+N3 it will be greater than N1 because this filter; is the cascade of two filters and end up. But the key advantage is the number there are lots of 0s in this in this implementation, right. So basically there are lot of 0 valued samples and that is where we get the advantage. 0 valued samples in the impulse response.

Now we have; if you do it in the polyphase and the multirate implementation anyway you have taken advantage of that. But just to keep in mind that the computational load becomes actually lower if you go to the IFIR, the in filter order itself will not be lower but the computational load actually turns out to be lower. Okay. Any questions? You have to make sure that like for example the filters G and I can be designed to be linear phase, so it will not affect or tamper with the phase of the signal.

So usually in our filter designs if you are able to maintain linear phase that is the best that we can do and so that really does not pose a problem for us. Now just as a passing comment. Where does the signal actually lie, the next copy of the signal? The next copy of the signal actually lies here, right. If you go back and look at it because the signal is preserved in the passband so the signal is preserved in the passband.

So the first copy of the signal is here the second copy of the signal. So in principle you could have designed your transition band to start here and end here. So you could have designed it slightly wider and therefore a lower order filter as well, because ultimately what you want to kill is the signal presence of the signal the filters are not the ultimate ones that you have to worry about.

It is where the signal lies as long as you have filter has got the response of the purple line it will remove the unwanted images and you will get the desired frequency response. So again several you know simple intuitive insights that you can learn from here. Hopefully, if there is a chance that if you have to actually design a very narrow filter you will not do a direct filter design you will do and Interpolated FIR design.

And thereby take a lot of advantages especially if sampling rate change is going to be involved as part of the as part of the computation. Okay. So with that we would like to now introduce the new concept by the way review of the cascaded integrated comb filters, basic filter is of this type. You can cascade them to get a lower attenuation but we have to keep in mind that the passband is not flat.

So for example the I of z that we have designed here cannot be a cascaded integrated design integrated comb filter unless you have designed G of z to have the pre-compensation. If you have designed G of z to have the pre-compensation, then I of z can be a multiplier less filter and then have huge advantages. So again this is a class of filters that are used quite extensively in ASIC design because primarily because they are multiplierless filters.