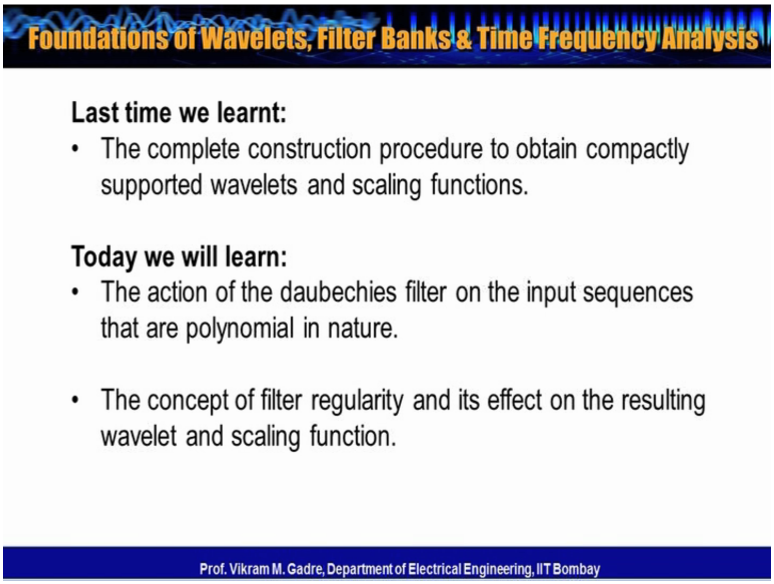
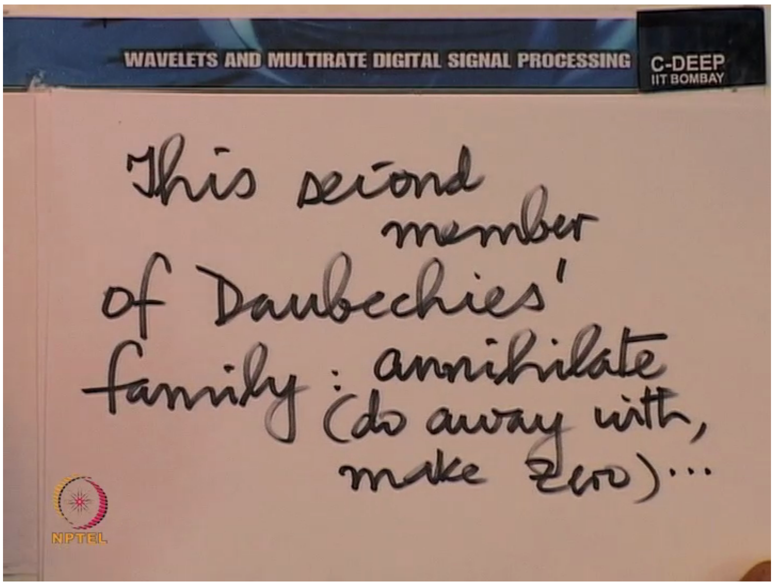
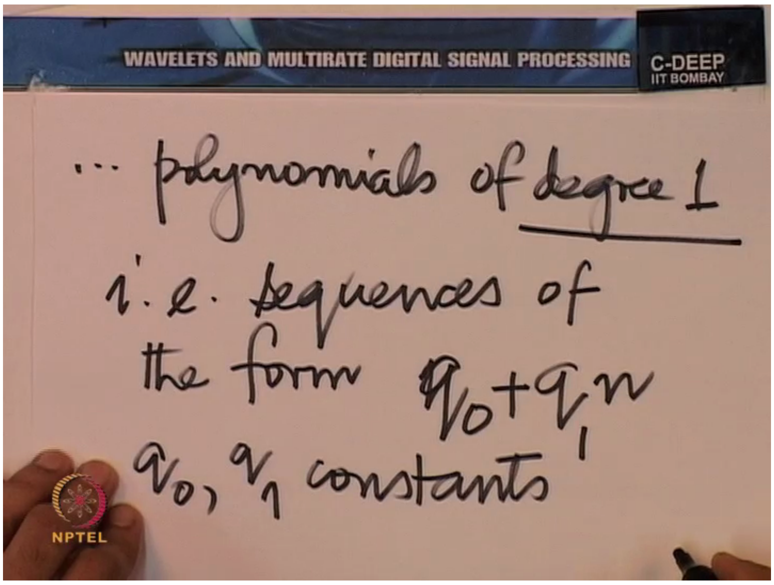
**Foundations of Wavelets, Filter Banks and Time Frequency Analysis.**
**Professor Vikram M. Gadre.**
**Department Of Electrical Engineering.**
**Indian Institute Of Technology Bombay.**
**Week-5.**
**Lecture -14.3.**
**Second Member of Daubeshies Family.**

(Refer Slide Time: 0:20)



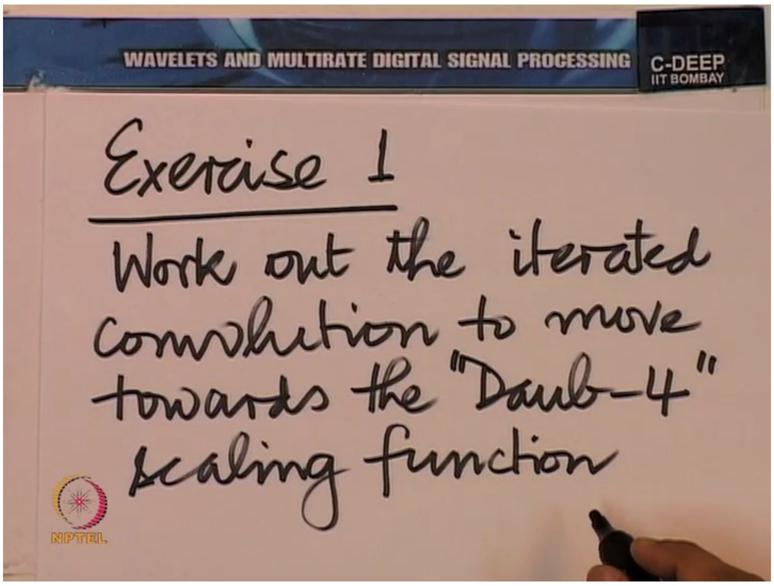



The 2nd member, this 2nd member would annihilate or kill or do away with or make zero whatever you wish to call that, annihilate polynomials of degree 1. Essentially sequences of the form some say Q0 + Q1n, Q0 and Q1 are constants. What I mean what I mean by that is the following. If you look at the overall sequence being given to the filter bank at the input and if you would think of that sequence as possessing one component of this kind and the remaining essentially a residual component, only the residual component would come out on the high pass branch.

This polynomial component would only be present in the lowpass branch. So the other way of saying it is a few more smoother terms are retained on the lowpass branch and removed from the high pass branch. Another way of saying it is, well, the high pass branch becomes even more and more high pass. You know, so what I am saying effectively is that it really behaves more as the high pass filter than does the case of haar of length 2, Daubeshies of length 2 is the Haar. And that does not behave as well as the high pass filter as does this.

In fact I would like to put down 2 exercises for the class now with this discussion that we have. And I strongly recommend as a part of this course that students work out both of these exercises to understand what I am saying. Working out these exercises does mean using a computer, it would help to use a computer to evaluate the expressions finely, to get a good feel, it cannot be done by hand but it is worth doing.

(Refer Slide Time: 3:27)



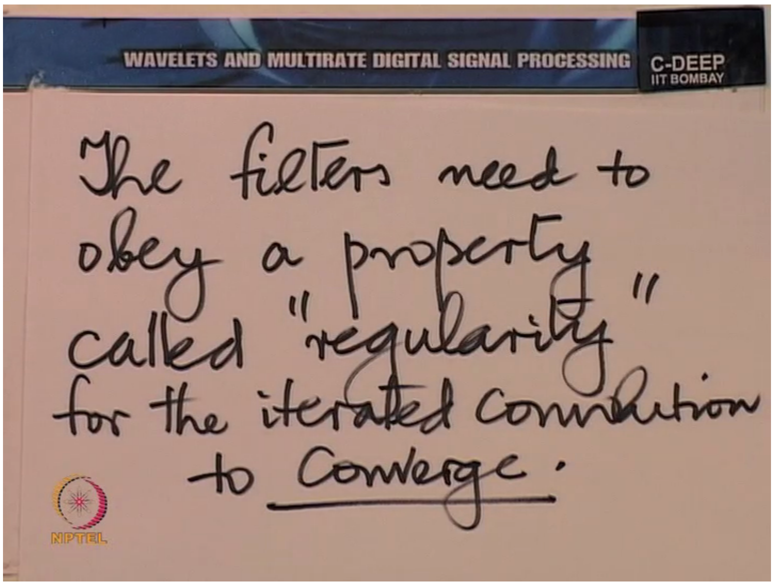

So the exercise is as follows, I shall explain the exercise here, exercise 1. Work out the iteration to move towards the Daub-4 as they call it. You know this is a nomenclature that we would like to introduce here. Daub-4 means the Daubeshies filter bank with the filters of length 4. So here what we have just been talking about is the Daub-4 set of filters or Daub-4 filter bank. So we have already explained how to carry out the iterative convolution but it would be worth actually carrying it out to move towards a Daub-4 scaling function. One would notice that the function that emerges is a continuous function, but it would not be expressible in what is called close form.

So one would not be able to express it as some sin t or sin e raised to power t, something of that kind but it would be continuous function, it would converge to a continuous function. And you know this is something important here. We have just set up some kind of a filter bank and we started iteratively convolving the impulse response with its own compressed versions. What is it that guarantees that when you carry this iteration to infinity, there is going to be some semblance of convergence in that process? Nothing inherently. So if we take an arbitrary filter bank like this with an H0, H1, G0 and G1 and if you were to take the H0 and carry out an iterative convolution like this, you might land up with what is called the fractal function.

In fact when I say fractal function, the word function is a misnomer. It would mean that that iterative convolution processor would not converge to a function at all, or at least definitely not to a continuous function, that could very well happen. In the Haar case we had a neat beautiful rectangular Pulse to which it converged. In the Daubeshies 4 case against, we are going to converge towards a continuous function I am assuring you even before you carry out this exercise. We will soon see that for the higher-order members of the Daubeshies family, you would again converge to continuous functions.

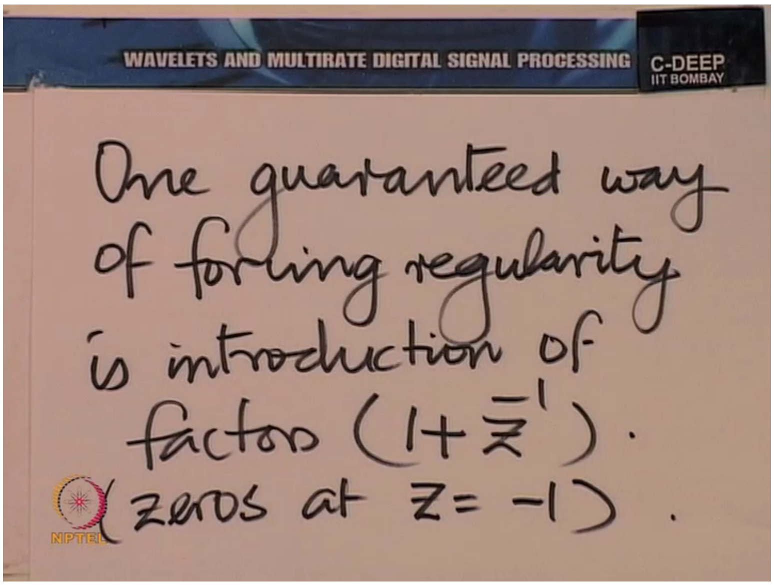
However if you just picked some arbitrary lowpass filter and started iterating it like this, maybe even a lowpass filter which satisfies that orthogonality to even translates that we have asked for. There is no guarantee that any such arbitrary filter would converge in this iterated convolution. So what is it about the filter which allows convergence? In the literature on wavelets, they speak of this as a property called regularity.

(Refer Slide Time: 7:18)




So we say that the filters need to obey for the iterated convolution to converge. Converge do what? Well, converge to a function which is either continuous or at least continuous almost everywhere, except for an isolated finite number of points. So you know if you really want to look at it that way, the Haar Scaling function is not continuous but it is only 2 points at which it is discontinuous, not infinite number of points. And we do not want that situation of this iteration taking us to quote unquote function or objects which has infinite points of discontinuance, that is what we are trying to say.

(Refer Slide Time: 8:43)



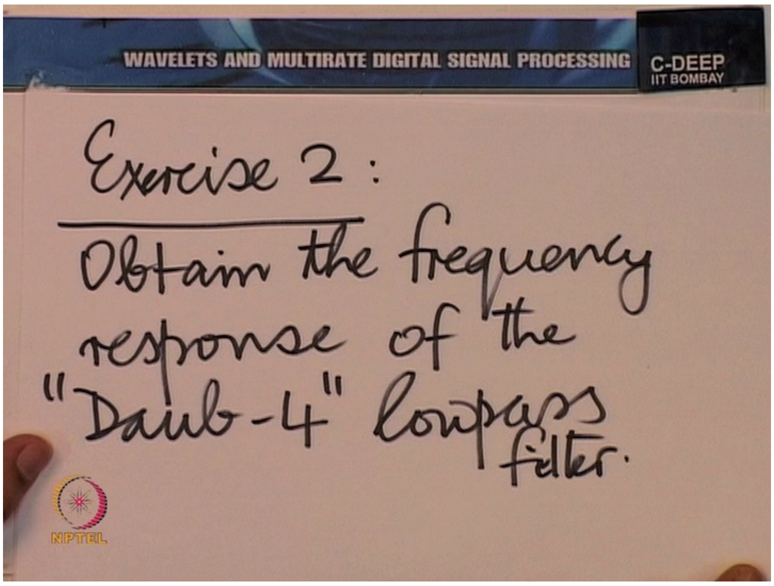

So whatever it be, this regularity in this case comes from the presence of zeros. So one guaranteed way of forcing regularity is introduction of factors. So the more zeros you have at Z equal to -1, of course in the lowpass filter and therefore correspondingly they would have zeros at +1 in the high pass filter. Again to give a physical significance, when you put Z equal to -1, you are talking about e raised to the power J Omega being equal to e raised to the power J + - pie, so make a being equal to pie, the extreme frequency, extreme high-frequency.

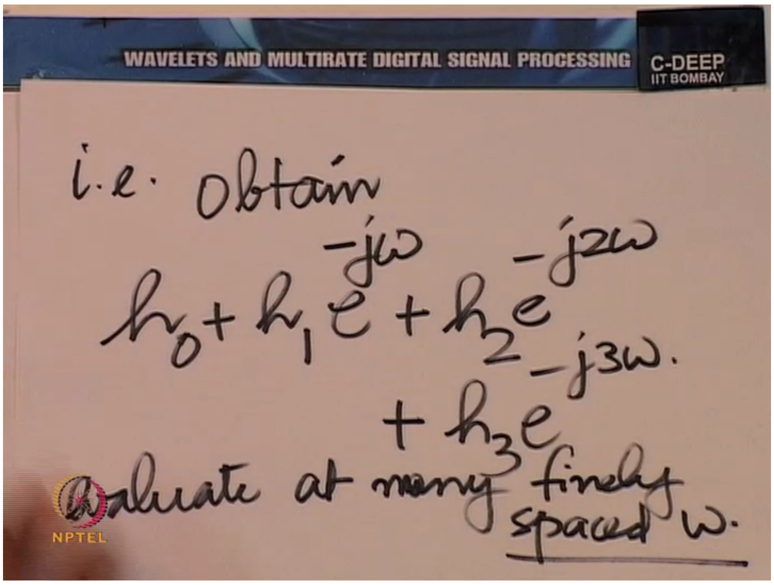
So in the lowpass filter we are saying but 0 at the extreme high-frequency and correspondingly therefore in the high pass filter we put zeros that the extreme low frequency namely 0, Omega equal to 0. Omega equal to 0 corresponds to the request to +1, e raised to the power J0 is one, simple. So one way to force regularity is to put 0 that -1 and that is what we are doing in the Daubeshies family. Haar 1 zero, Daub-4 2 zeros, Daub-6, that is the next member of the family, length 6 would have 3 zeros and so on so forth.

And you know what we say, we say the higher you go in the Daubeshies family in terms of length, the more regular your filters are, the more regular. What that means is the functions to which we converge on iterative convolution becomes smoother and smoother, they have more and more derivatives that are continuous. So if you look at Daub-4, its differentialability in the traditional sense is under scanner, it is continuous but as far as differentialability goes, there are issues. But when you go to higher-order Daubeshies, that is also taken care of, the functions becomes smoother and smoother.

And now you can see how to do this for higher-order higher-order Daubeshies filters. I mean whether it is length 6 or length 8 or length 10, we know exactly how to carry out an iterative convolution. Put all those impulses together, the impulse response coefficients as impulses uniformly spaced, squeeze that set of impulses by factor of 2, convolve it with the 1st set, again squeeze by factor of 2 convolve it again and this can continue and continue and continue.

(Refer Slide Time: 12:12)

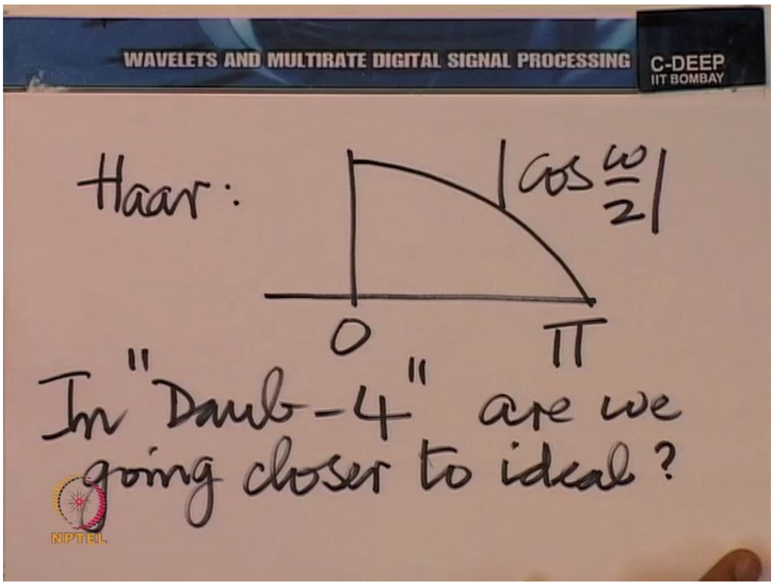


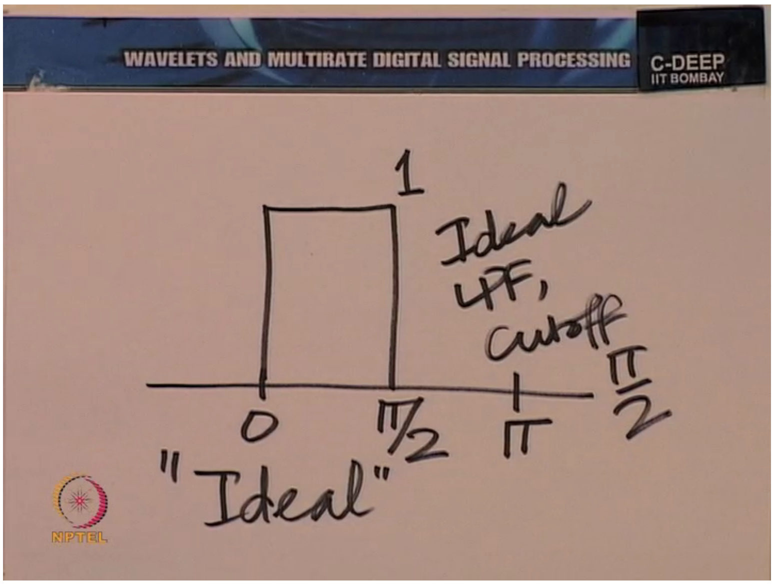



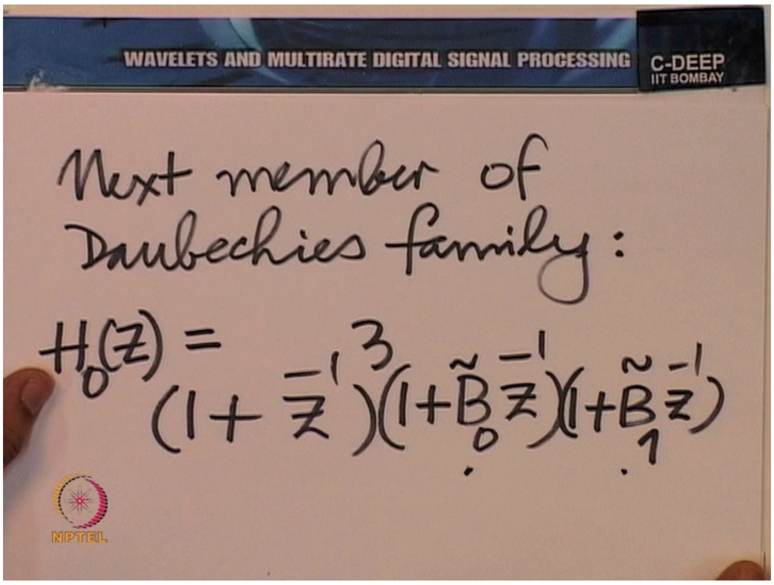

So, I leave it as I said as an exercise to complete this iterative convolution. So I repeat the exercise which all of us must do. Work out the iterated convolution to move towards the Daub-4 scaling function. The 2nd exercise which I would like to ask the class to do is the following, obtain the frequency response of Daub-4 lowpass filter and of course therefore also of the high pass. So just for completeness, let me write down the expression for the frequency response.

What I am saying is obtain H0 + H1 e raised to the power J Omega + H2, well, e raised the power - J Omega, H2 e raised to the power - J2 omega, + H3 e raised to the power - J 3 omega. Evaluate at many finely spaced omega, so where you know between 0 and pie, one could take 1000 points and evaluate this expression to get a feel of the frequency response. And the idea is to compare it with the frequency response of the Haar. So what we are specifically looking for is this.

(Refer Slide Time: 14:10)



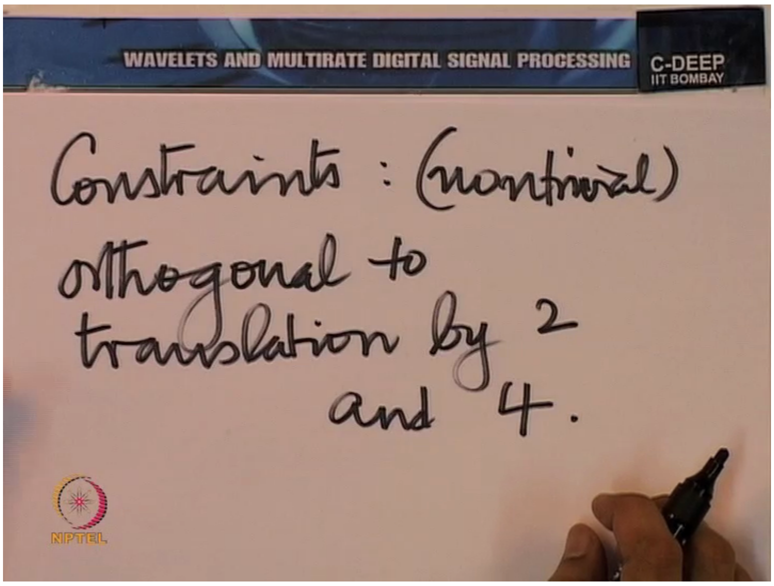



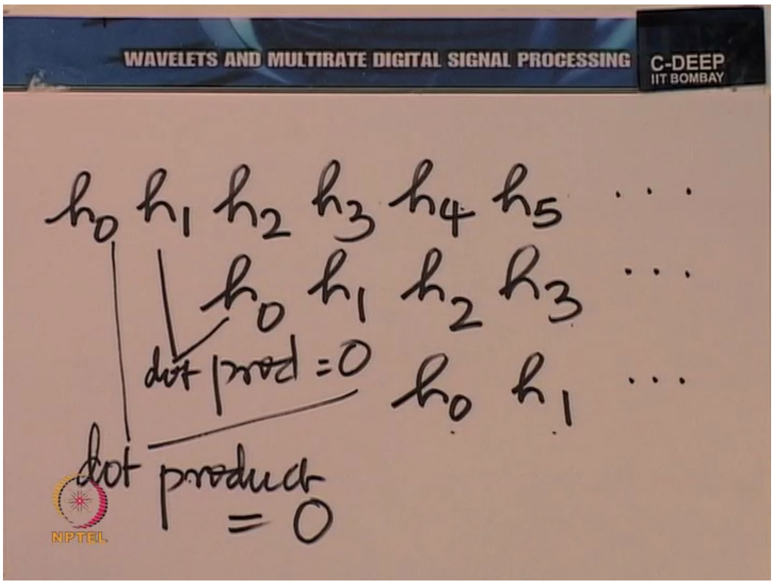



Haar gave us essentially this cos Omega by2 kind of response between 0 and pie. In Daub-4, are we going closer to ideal? What ideal are we talking about? You remember what the ideal was, the idea was essentially this, this is the ideal, an ideal lowpass filter with a cut-off of pie by 2. Now let me tell you how to build the next member of the Daubeshies family. To do that then we would put one more 0 at Z equal to -1 in the lowpass filter. So the next member of Daubeshies family.

Essentially take H0Z to be of the following form 1+ Z inverse, the whole cube and there would be 2 more degrees of freedom or you could call it B0 tilde if you like, just to distinguish it from the B0 that we have calculated here, B1 tilde to Z inverse, remember that this member would have a lowpass filter of length 5, I mean sorry, of degree 5 and therefore length 6. And since it is degree 5 or length 6, 3 of the zeros are constrained, 2 of them are free, so you have 2 free parameters here, B0 tilde and B1 tilde. And what are the constraints in this case?

(Refer Slide Time: 16:38)

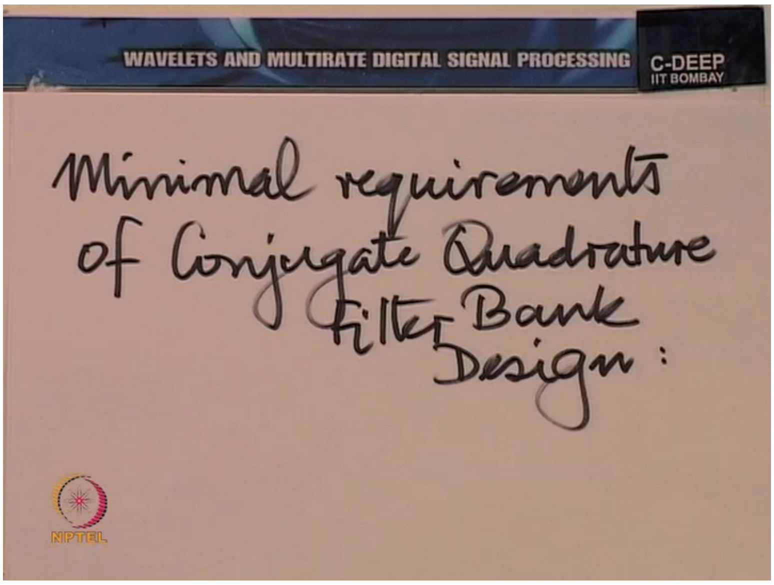


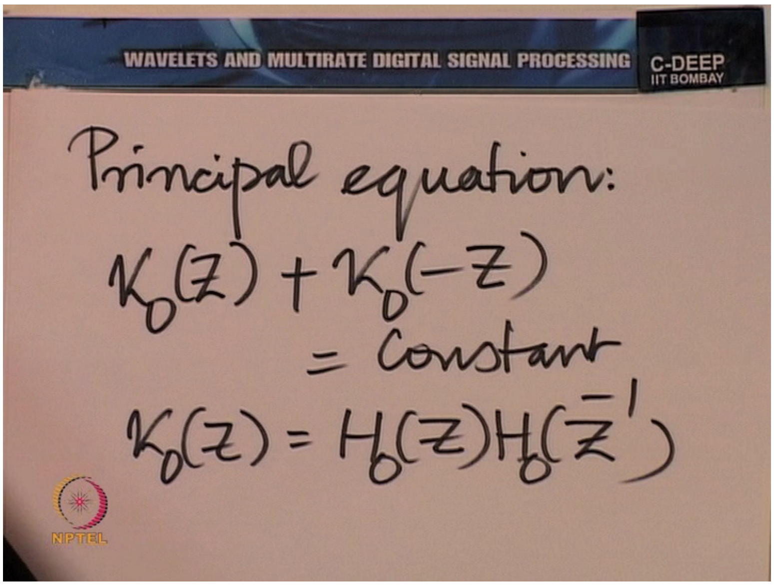



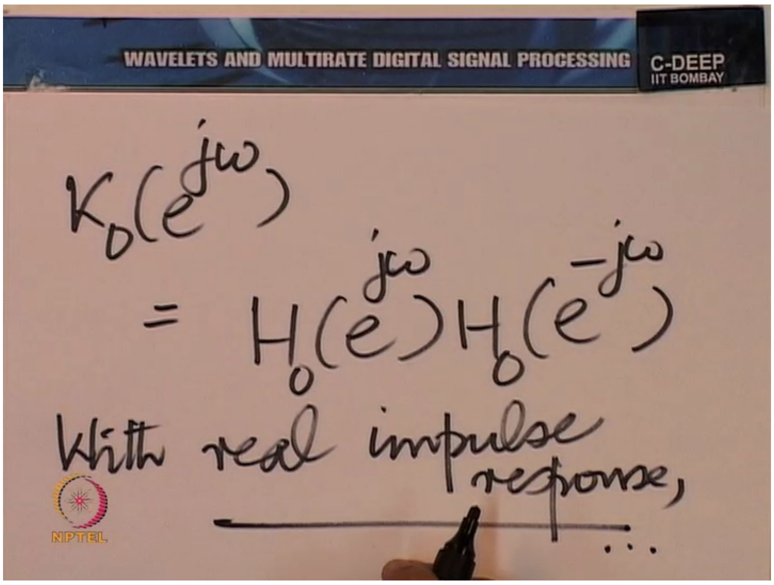

The constraints are, I mean the nontrivial ones I mean. Orthogonality to translation by 2 and 4. In other words let me put it down explicitly you would have H0, H1, H2, 3, 4, 5, translation by 2 would mean this, rest of them are zeros, we do not need to bother, translation by 4 would mean this. So take the dot product of these 2 and put it equal to 0. Take the dot product of these 2 and put it equal to 0, these are the 2 constraints. So I will read them off, H0 times H2 + H1 times H3 + H2 times H4 + H3 time H5 is 0 in this. And H0 times H4 + H1 times H5 is 0 in this.

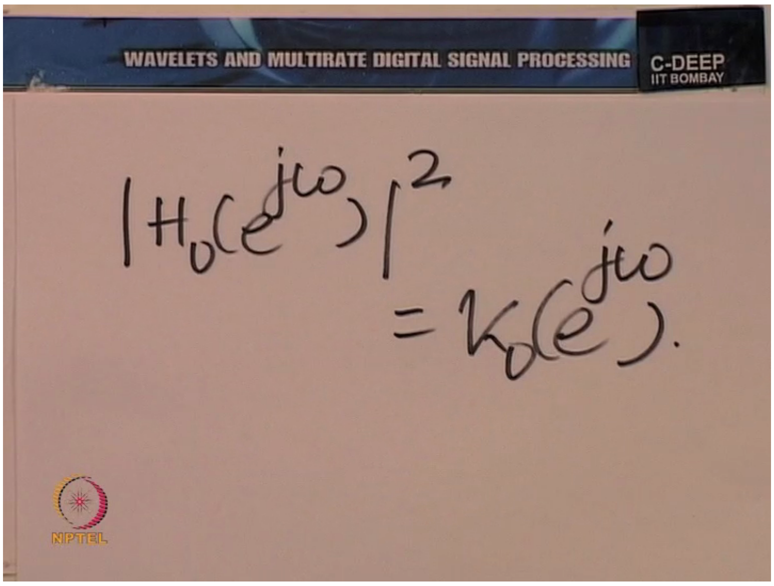
And these are the only 2 nontrivial constraints that we have, 2 constraints, 2 free parameters one can determine them, simple quadratic equations, this time they are simultaneous quantity equations involved, little more work but doable. And one can keep doing this for higher-order members now. By the way there are different ways of building this family of Daubeshies filters, this is one way. There are more convenient ways to or what might be seen as convenient ways by some, it is not our objective to dwell on those methods in this lecture but rather to make a more general remarks now about this class of filter banks that we are talking about, namely the conjugate quadrature filters.

(Refer Slide Time: 19:09)





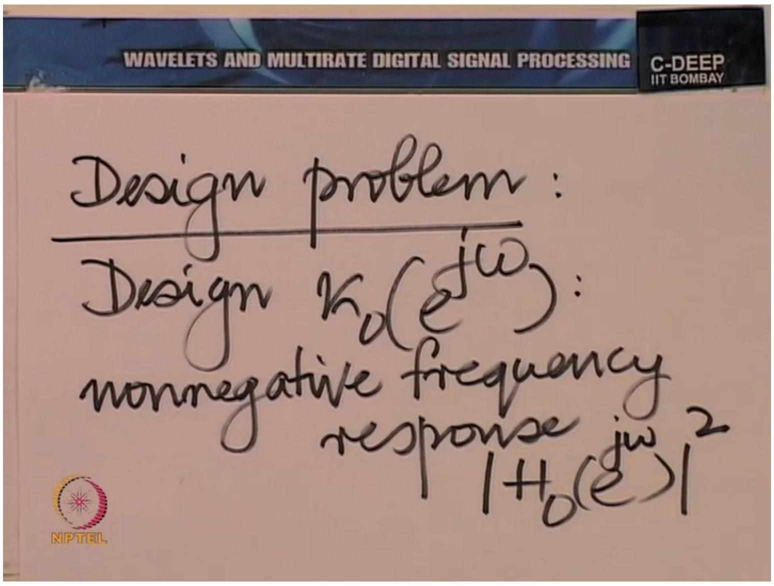

So we wish to put down what we call the minimum requirements of design in a conjugate quadrature filter bank. Incidentally you might wonder where this name comes from, conjugate quadrature. Actually it is the quadrature word which is important there. The quadrature word comes in some sense from the idea of 90 degrees shift. So you know in a certain sense of what we have done is to relate the high pass filter and the lowpass filter frequency responses by a shift of pie on the frequency axis. So notionally what we are saying is a lowpass filter with cut-off of pie by 2 aspires to become a high pass filter with cut-off pie by 2 in this relationship, replacing Z by - Z essentially.

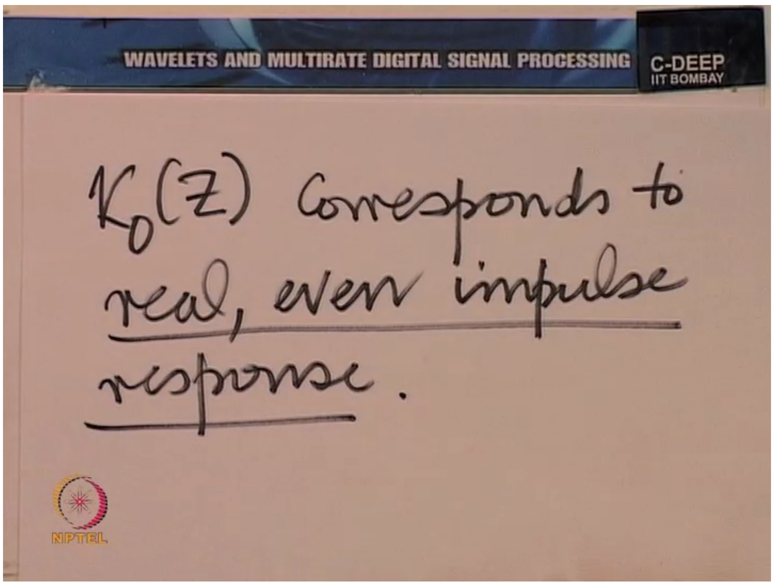
So replacement of Z by - Z to relate the low and high pass filter brings what is called the quadrature relationship, that about the name. But anyway, so what we have is essentially the following equation. The principal equation governing the quadrature filter bank is this. Kappa

0Z + Kappa 0 - Z is a constant where Kappa 0Z is H0Z H0Z inverse. And therefore if you look at it, the frequency domain says Kappa 0 e raised to the power J Omega H0 e raised to the power J omega H0 e raised to the power - J omega essentially.

And with the real impulse response, what do we have? With the real impulse response we have mod H0 e raised to the power J omega squared is Kappa 0 e raised to the power J Omega. So in other words now we have a very clear design problem before us. Designing a conjugate quadrature filter bank is equivalent to designing essentially one filter, Kappa 0 e raised to the power J Omega.

(Refer Slide Time: 22:37)



Principal equation:

$$K_0(Z) + K_0(-Z) = \text{Constant}$$

$$K_0(Z) = H_0(Z) H_0(Z^{-1})$$



WAVELETS AND MULTIRATE DIGITAL SIGNAL PROCESSING C-DEEP IIT BOMBAY

Design problem:

Design $K_0(e^{j\omega})$: nonnegative frequency (lowpass $\pi/2$) response $|H_0(e^{j\omega})|^2$



WAVELETS AND MULTIRATE DIGITAL SIGNAL PROCESSING C-DEEP IIT BOMBAY

$K_0(Z)$ corresponds to real, even impulse response with even samples = 0 except $0^{th}$.

So the design problem is, design Kappa 0 e raised to the power J Omega and if you look at Kappa 0 e raised to the power J omega, it is a nonnegative frequency response as you can see. Mod H0 e raised to the power J omega the whole squared. And this non-negativity can only come from an even, real and even response. So we are saying Kappa 0Z corresponds to a real and even impulse response with a constraint that even samples must be 0, except the 0th. So you know this equation that we had here Kappa 0Z + Kappa 0 - Z is a constant essentially says they even samples other than the $0^{th}$ sample are all 0.

So you are trying to design a lowpass filter, so in fact we should qualify this further, nonnegative lowpass frequency response. And what kind of a lowpass frequency response, with a cut-off pie by 2. So now we have the design problem very clear. Design an even impulse response, a lowpass aspiring to be a lowpass with cut-off pie by 2, nonnegative with the constraints that the even samples of the impulse response are 0 except for the $0^{th}$ sample. There are many different ways in which one can design finite impulse response filters and any optimisation which allows us to design Kappa 0 with these constraints is acceptable to design Kappa 0 and once we have Kappa 0, then you look at its roots.

For each root you have reciprocal roots, H0 Z, H0 Z inverse. Out of each pair of reciprocal roots, put one in H0 and the other one automatically goes in H0Z inverse. This is the general strategy to design conjugate quadrature filters and the Daubeshies family is just one of many such families. So with this then we have put down the whole family of multiresolution analysis or filter banks for you. In the next lecture we shall ask what is it that we are looking for in these families, in other words is there some fundamental limit, is there some fundamental 2 domain requirement that we are trying to seek and fulfil? Thank you.