Digital Signal Processing

Prof. S. C. Dutta Roy

Department of Electrical Engineering

Indian Institute of Technology, Delhi

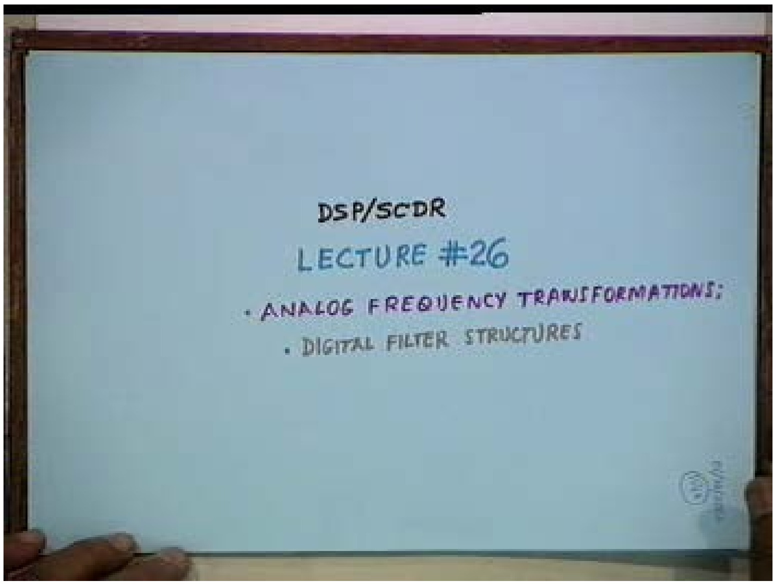
Lecture - 26

Analog Frequency Transformations:

Digital Filter Structures

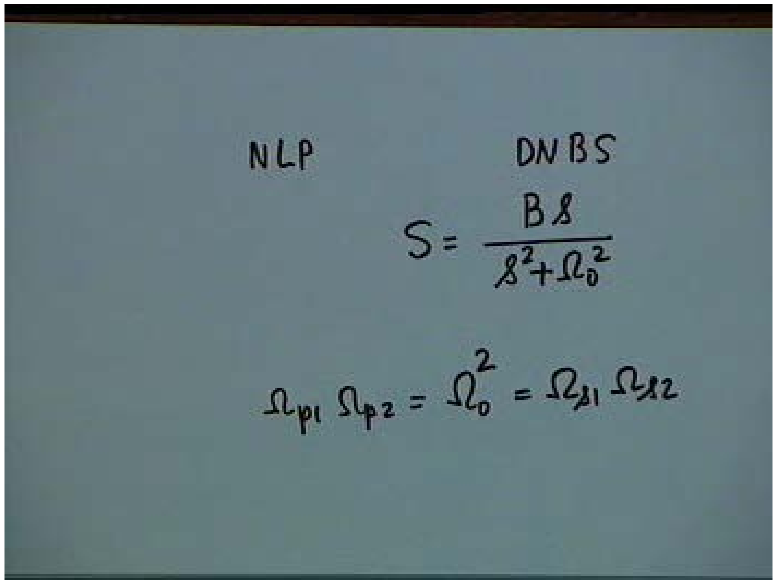
This is the 26th lecture and we will continue our discussions on analog frequency transformations.

(Refer Slide Time: 1:09 - 1:24)



We illustrate by a fairly involved example and you must follow the steps carefully. And then we move to the topic digital filter structures. In the last lecture, we discussed Chebyshev low pass filter design for a given specification and we illustrated by a couple of examples. We then talked about analog frequency transformations.

(Refer Slide Time: 01:47 - 5:53)

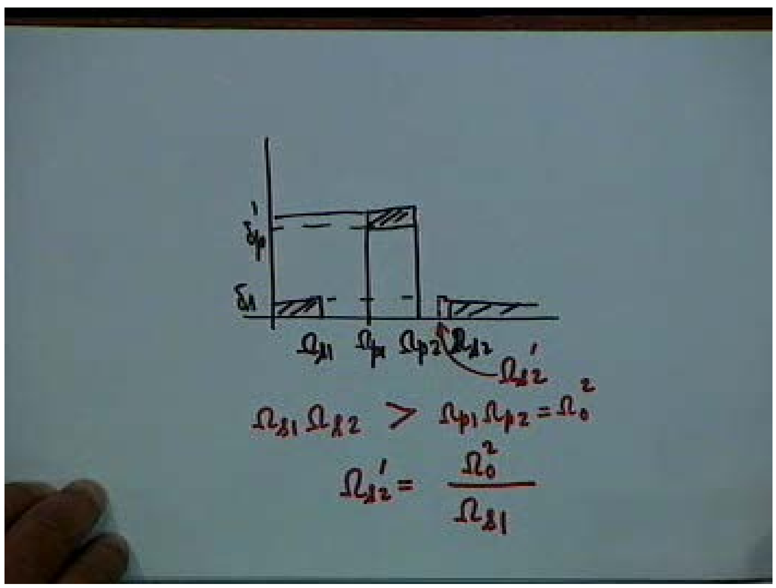


We also assumed that S is the complex frequency variable of the normalized low pass filter (NLPF) in which the passband is defined by 1 $\leq$ magnitude $\leq$ δp, and it extends from zero to 1 radian per second. The stopband starts somewhere at Ω_s with a tolerance of δs. If these magnitude characteristics are satisfied by a transfer function in the complex S domain, then this normalized low pass filter can be transformed to a de-normalized high pass filter by substituting S by Ω_p/s.

On the other hand, if it is simply a de-normalized low pass filter, where only the cutoff frequency changes, obviously S has to be replaced by s/Ω_p. Then for de-normalized band pass filter S has to be replaced by $(s^2 + \Omega_0^2)$ Q/(Ω_0s). This can be written as $(s^2 + \Omega_0^2)$/(Bs) where B is the bandwidth. This bandwidth has the same tolerance as in the normalized low pass filter.

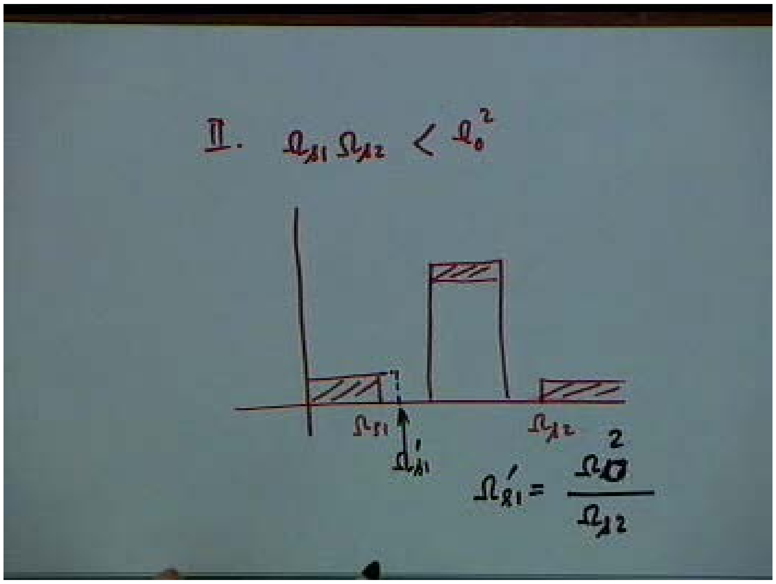
In other words, this 1 radian per second in NLPF transforms to Ω_{p1} and Ω_{p2} but the tolerances remain the same. On the other hand, if the transformation is to be from normalized low pass to de-normalized band stop filter, then the transformation required is simply the reciprocal of the band pass transformation i.e. Bs/$(s^2 + \Omega_0^2)$. Given a normalized low pass filter, you can always transform it. Suppose it is the reverse transformation, which is usually the case. If the specifications are those of the de-normalized low pass or de-normalized high pass, there is no problem. The problem arises with band pass and band stop because of the constraint that they have to be geometrically symmetrical. In other words, $\Omega_{p1}\Omega_{p2} = \Omega_0^2$. Usually Ω_0 is not specified. What are specified are Ω_{p1} and Ω_{p2} and Ωs_1 and Ωs_2. The product of Ω_{p1} and Ω_{p2} must be equal to $\Omega s_1 \Omega s_2$. This geometric symmetry is inherent in the transformation as I showed in the last lecture. And therefore given specifications may not satisfy this. In that case, you have to adjust the stopband edges on one side or both sides if you so desire so that passband specifications are met exactly and stopband is over satisfied.

(Refer Slide Time: 06:21 - 8:22)



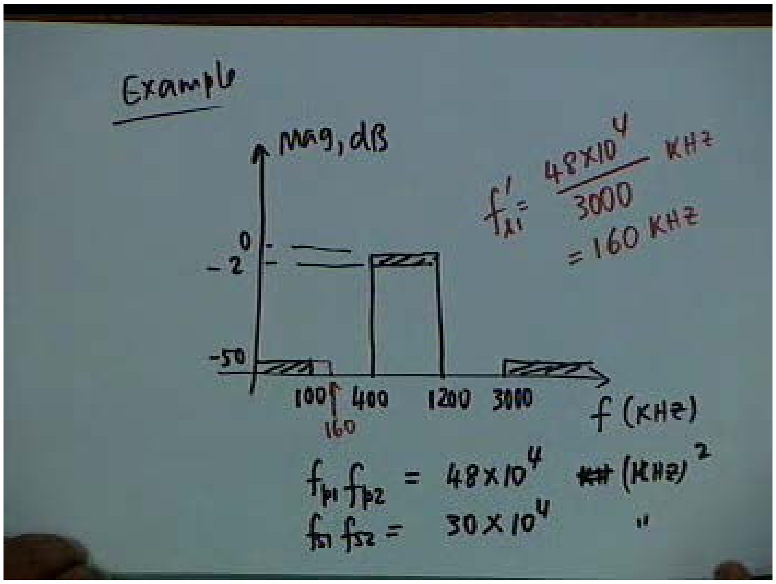
For example, let the specifications on tolerances of a band pass filter be δs and δp, and let the band edges be Ω_{p1}, Ω_{p2} and Ω_{s1}, Ω_{s2}. Let $\Omega_{s1}\Omega_{s2} > \Omega_{p1}\Omega_{p2}$ which is of course equal to Ω_0^2. What you have to do is make the left hand side smaller and therefore you have to shift Ω_{s2} to a lower value. You cannot reduce Ω_{s1} because then the stopband specifications will not be met. On the other hand, you can reduce Ω_{s2} by shifting it to Ω'_{s2} such that $\Omega_{s2} = \Omega_0^2/\Omega_{s1}$. Therefore if you design this filter after these adjustments, the stopband on the right hand side will be over satisfied. The stopband on the left hand side is exactly satisfied and the passband will be exactly satisfied. This is the story if $\Omega_{s1}\Omega_{s2} > \Omega_0^2$.

(Refer Slide Time: 08:27 - 10:17)



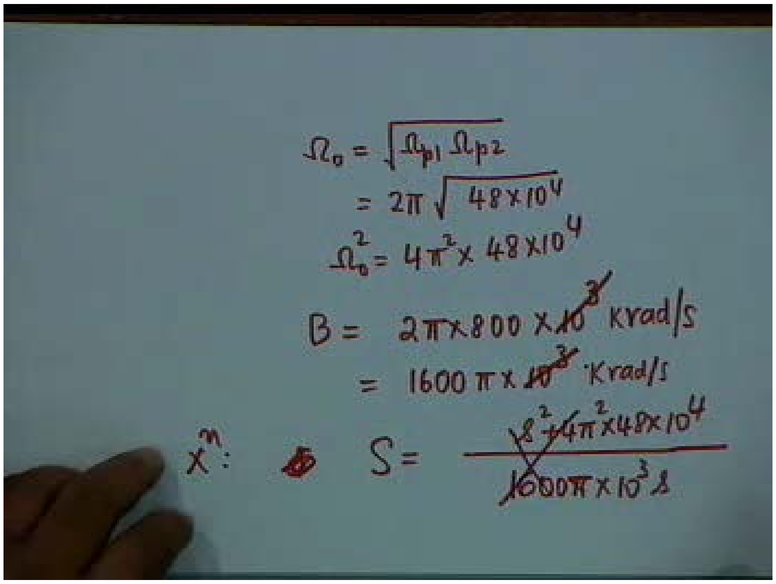
If $\Omega s_1 \Omega s_2 < \Omega_0^2$ then obviously you will have to increase Ωs_1. Therefore what you have to do is leave Ωs_2 intact. So you increase Ωs_1 to $\Omega' s_1$ in such a manner that $\Omega' s_1 = \Omega_0^2 / \Omega s_2$. This modification has to be made, otherwise the filter shall not satisfy the specifications. There is a further complication. After you have adjusted the edges of the stopband, how do you find the specifications of the normalized low pass filter corresponding to this? We know the tolerances δp in passband and δs in stopband. But what is the stopband edge? It is Ωs. That is to be found out.

(Refer Slide Time: 10:28 - 13:20)



A practical example is shown in the figure where the magnitude is given versus frequency in kHz and not in radians. The tolerance scheme is this: the passband tolerance is 2dB and the stopband tolerance is – 50 dB. The figure is not to scale. The edges of the passband are specified as 400 and 1200. The edges of stopband are specified as 100 and 3000. Now obviously to test geometric symmetry, you do not have to go to $\Omega_{p1}\Omega_{p2}$, but you can test with the frequency in kHz itself. In this case, $f_{p1}\ f_{p2}$ is 48×10^4 (kHz) whereas $f_{s1}f_{s2} = 30 \times 10^4$ (kHz)^2; so $f_{s1}f_{s2} < f_{p1}f_{p2}$. In other words, f_{s1} has to be increased to f'_{s1} such that $f'_{s1}\ f_{s2}$ shall equal 48×10^4. This gives f'_{s1}= 160 kHz. Now with these specifications you have to design the filter. The bandwidth is obviously 800 kHz. Now let us find the transformation.

(Refer Slide Time: 13:26 - 17:09)



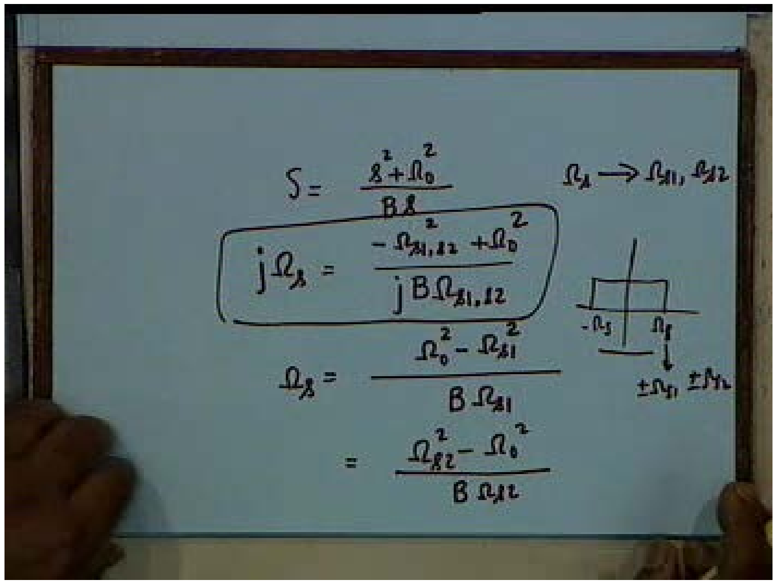
What is Ω_0?

$\Omega_0 = \sqrt{(\Omega_{p1}\Omega_{p2})} = 2\pi \times\sqrt{(48 \times 10^4)}$. We need only Ω_0^2. We do not require Ω_0 so we need not find it out. Here $\Omega_0^2 = 4\pi^2 \times 48 \times 10^4$. We require the bandwidth $B = 800 \times 2\pi \times 10^3 = 1600\pi \times 10^3$. Therefore our transformation is $S = (s^2 + 4\pi^2 \times 48 \times 10^4)/(1600\pi \times 10^3 s)$.

Our normalized low pass filter should have a tolerance of 2 dB and stopband tolerance of – 50 dB. The passband extends to 1 radian per second and what is Ωs? To find Ωs, you have to appeal to the basic relationship. What is the basic relationship?

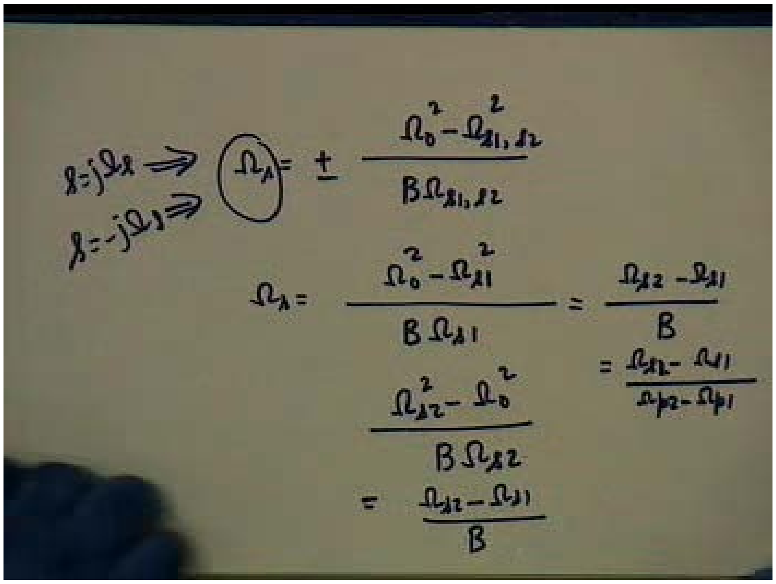
Here $S = (s^2 + \Omega_0^2)/(Bs)$.

(Refer Slide Time: 19:34 to 23:34)

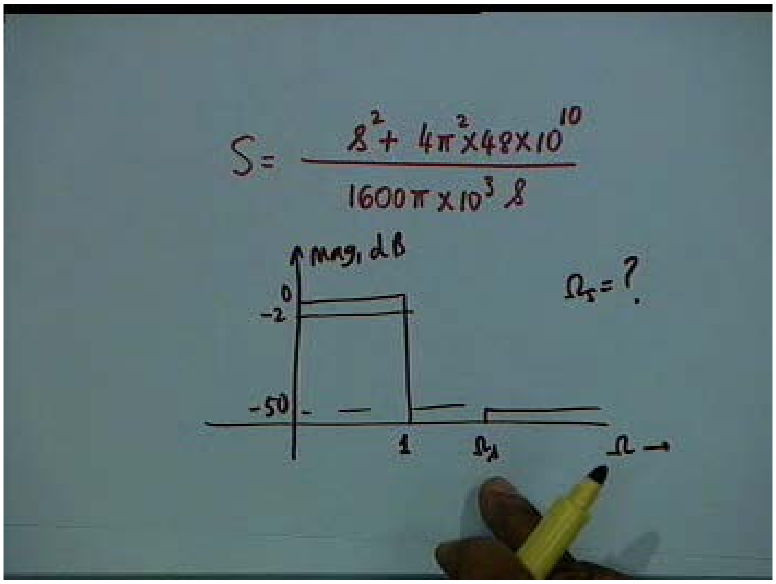


And we are trying to find out Ω_s so if I put $S = j\Omega_s$ then on the right hand side we shall have $s = j\Omega_{s1}$ or $j\Omega_{s2}$ because Ω_s corresponds to Ω_{s1} as well as Ω_{s2}. The stopband edge of the normalized low pass filter must correspond to the two stopband edges of the de-normalized band pass filter. In general, I shall have $\pm\ j\Omega_s = (-\ \Omega_{s1,s2}^2 + \Omega_0^2)/(\pm jB\ \Omega_{s1,s2})$ where the signs may not correspond to each other. The positive value of Ω_s is obtained as $(\Omega_0^2 - \Omega_{s1}^2)/(B\Omega_{s1})$ or $(\Omega_{s2}^2 - \Omega_0^2)/(B\Omega_{s2})$. The two expressions are identical because $\Omega_0^2 = \Omega_{s2}\ \Omega_{s1}$. Thus $\Omega_s = (\Omega_{s2} - \Omega_{s1})/(\Omega_{p2} - \Omega_{p1})$. Several text books have found incorrect expressions for Ω_s.

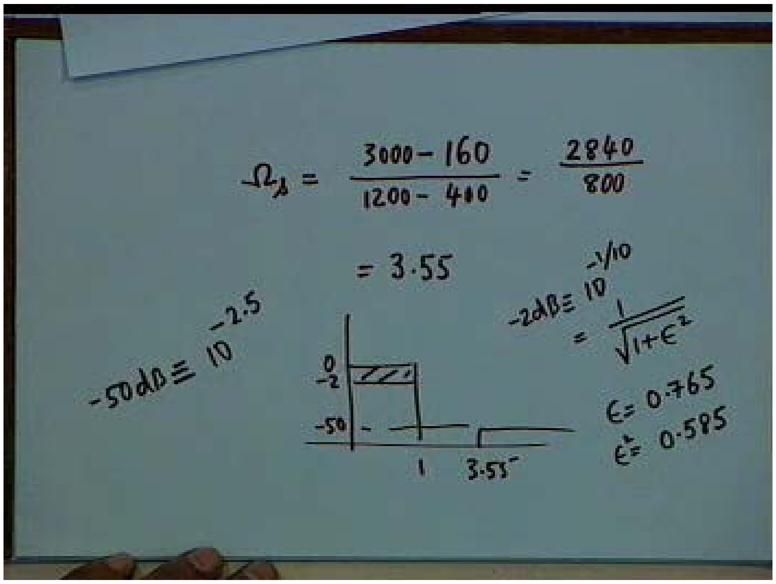
(Refer Slide Time: 23:41 - 27:12)



(Refer Slide Time: 24:57 to 25:08)

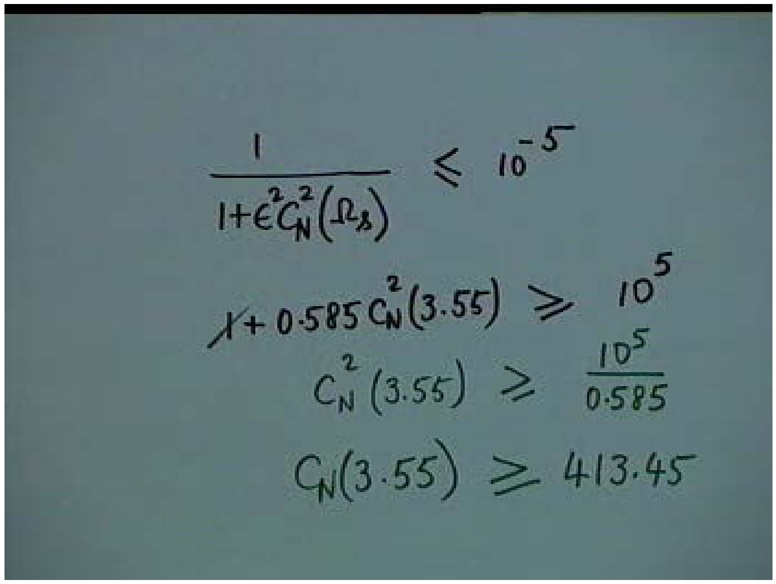


(Refer Slide Time: 27:18 - 29:52)



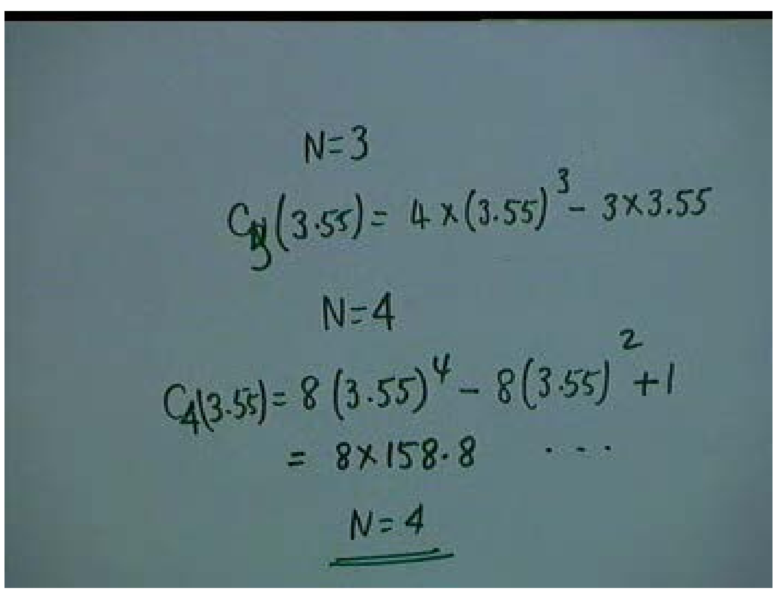
Therefore in our case, the normalized analog low pass filter should have an Ω_s which is equal to $(3000 – 160)/(1200 – 400) = 2840/800 = 3.55$. We have to design a low pass filter with the specs shown in the figure. To proceed further, the first thing that I have to do is to convert – 2dB in terms of ratio which is $10^{-.1}$ and if I want a Chebyshev design, for example, then this should be equal to $1/\sqrt{(1 + \epsilon^2)}$. My calculation shows that $\epsilon = 0.765$. I do require ϵ^2 also which is calculated as 0.585; – 50dB in terms of ratio is equivalent to $10^{-2.5}$ and therefore to find the order, instead of using $\cosh^{-1}$ formula, let us use the basic formula. This gives $1 + 0.585\ C_N^2\ (0.355) \geq 10^5$.

(Refer Slide Time: 29:56 to 32:12)



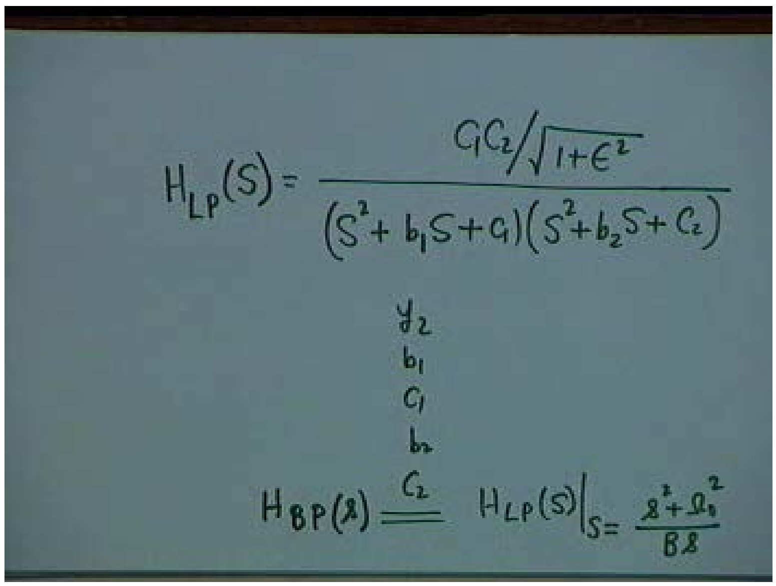
We can ignore this 1 on the left hand side as compared to 10^5. Then I should have C_N^2 (3.55) ≥ $10^5/0.585$ = 413.45. Now you take some starting point and iterate.

(Refer Slide Time: 32:15 - 33:43)



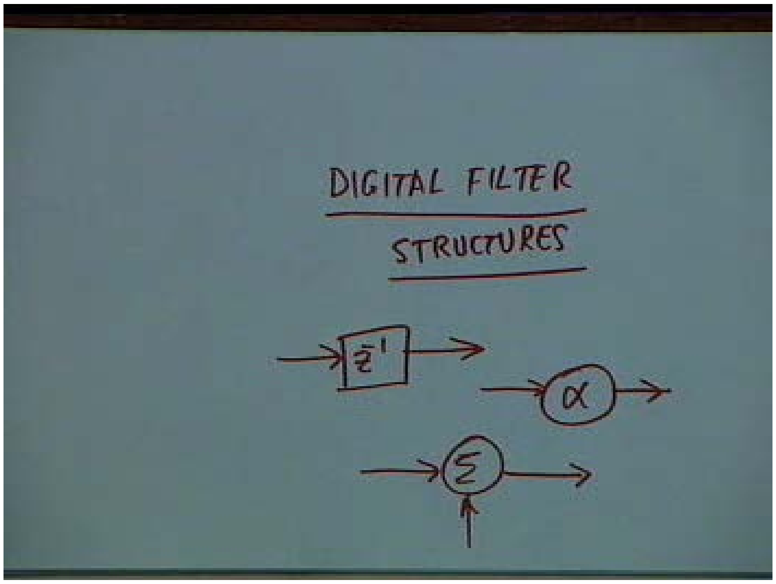
The starting point I took was N = 3 then C_3 (3.55) = $4(3.55)^3 - 3(3.55)$. Even if you replace 3.55 by 4, the right hand side does not make 400 and therefore N = 3 does not suffice. Take N = 4 and that suffices. Fourth order Chebyshev low pass filter shall satisfy these specifications on the normalized low pass filter.

(Refer Slide Time: 33:45 - 35:57)



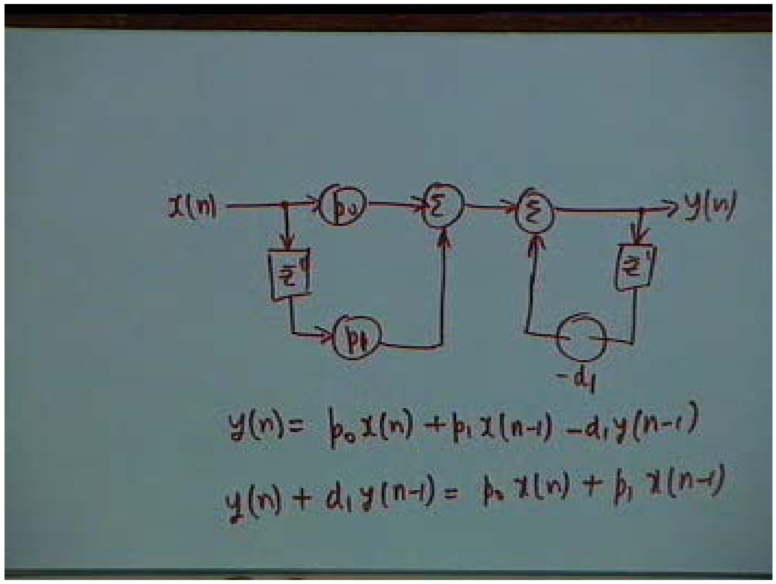
And the next step is to write down the low pass filter transfer function H_{LP} as a function of S; since it is fourth order, in the denominator, we shall have two quadratics $(S^2 + b_1S + c_1)$ $(S^2 + b_2S + c_2)$. In the numerator we shall have c_1 $c_2/\sqrt{(1 + \epsilon^2)}$. We shall have to do this normalization. And now you have to calculate y_2, b_1, c_1, b_2, c_2 and then substitute in this and get the normalized low pass transfer function in the S domain. Finally your band pass transfer function in the s domain shall be obtained by replacing S by $(s^2 + \Omega_0^2)/(Bs)$ and that is the end of the design. It is a fairly elaborate process and you can make a mistake at almost every step. Similarly you can design band stop filter with arbitrary specifications.

(Refer Slide Time: 36:24 - 37:53)



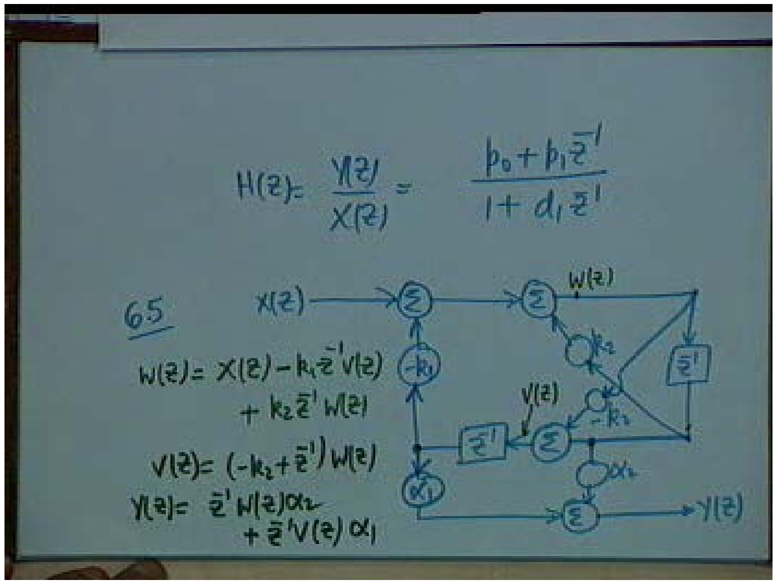
Our next topic would be digital filter structures. As I have talked about earlier, digital filters require only three basic blocks: one is the delay which is denoted by z^{-1}, one is the scalar multiplication and the third is addition or subtraction. In order to respect the actual hardware process of addition or subtraction, we do not use more than two signals at any adder. Now, if you are given a structure you can always analyze it to find the transfer function. The problem here is: given the transfer function, how do you find the structure? If you can find one, is this unique? No it is not unique. Synthesis problems are always characterized by the fact that if one solution exists, there exists an indefinite number of solutions. But a solution may or may not exist to a synthesis problem. On the other hand, to an analysis problem, a solution, however complicated the problem is, shall exist. We shall illustrate by two examples of analysis and then we shall go to synthesis.

(Refer Slide Time: 38:00 - 39:55)



You are given a structure like the one shown in the figure. While analyzing it, I shall also point out some of the disciplines that we have observed. If an arrow is missing, your structure does not make any sense. Analysis is rather easy; you can see that y(n) gets two signals $[p_0x(n) + p_1x(n – 1)]$ and $– d_1y(n – 1)$. So it is a recursive structure because it requires a feedback. Therefore I get $y(n) + d_1 y(n – 1) = p_0 x(n) + p_1 x(n – 1)$ and all that I have to do is to take the z transform now on both sides. The result is $H(z) = Y(z)/X(z) = (p_0 + p_1 z^{-1})/(1 + d_1 z^{-1})$.

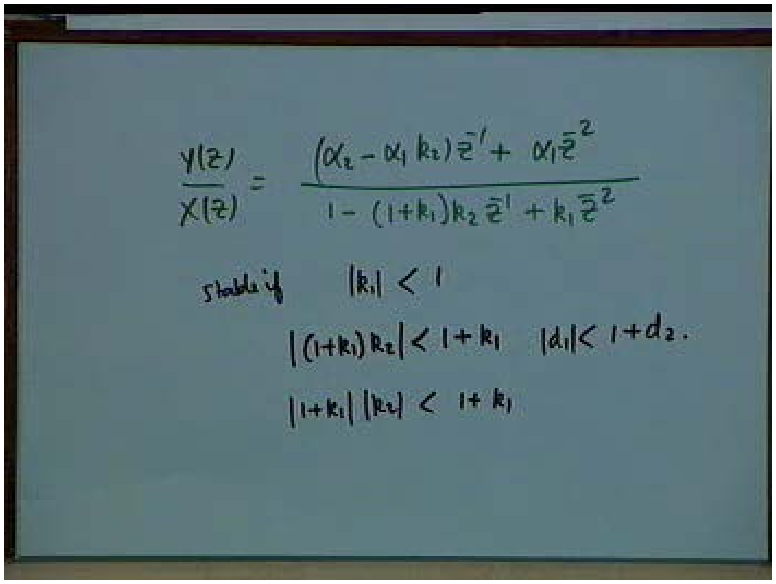
(Refer Slide Time: 39:58 - 46:02)



Obviously the structure that was asked to be analyzed is not canonic in delays. This is a first order filter and therefore it should require only one delay and not more than that. But it is canonic in multipliers because we have three coefficients and three multipliers. So the structure we analyzed is non-canonic in delays but canonic in multipliers.

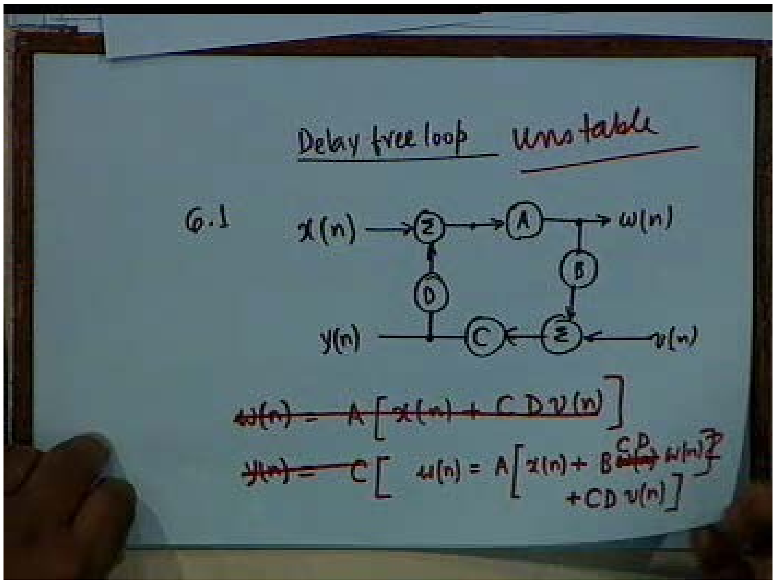
We will show how to obtain a canonic structure a little later but before that let us take another example. This is problem 6.5 from Mitra and is shown in the Figure. In analysis, it is a good idea, except at trivial nodes, to indicate some intermediate signals. We indicate signal at the output of the second adder on the top line as W(z). I also call the output of the adder in the middle line as V(z). I do not have to indicate any other because all other signals can be expressed in terms of W and V. You notice that we can write $W(z) = X(z) – Vk_1 z^{-1} + k_2 z^{-1}W$. I have expressed W in terms of X, V and W so, W can be expressed in terms of X and V. Now V(z) is $(– k_2 + z^{-1})W$. Finally, $Y(z) = z^{-1} W(z) \alpha_2 + z^{-1} V(z) \alpha_1$. So we have three relations and three unknowns Y, V, and W. And as V and W are related, all that you have to do now is to eliminate W. And in whatever way you do the algebra you can verify whether the calculation gives Y(z)/X(z), the transfer function, as $((\alpha_2 – \alpha_1 k_2)z^{-1} + \alpha_1 z^{-2})/(1 – (1 + k_1) k_2 z^{-1} + k_1 z^{-2})$.

(Refer Slide Time: 46:11 - 48:47)



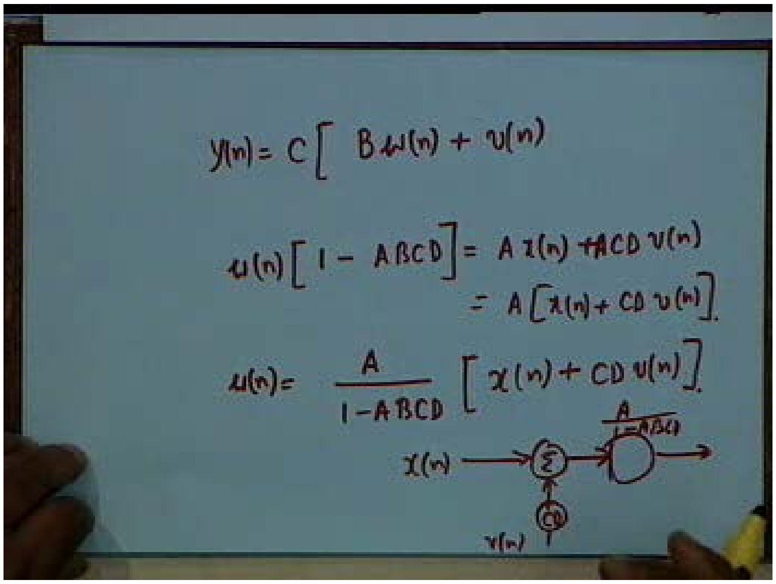
As it is a second order filter, it need not be always stable. What are the conditions of stability in this case? The coefficient k_1 should be less than 1 in magnitude; what was the other condition? $|d_1|$ should be less than $|1 + d_2|$, therefore $| 1 + k_1 \; k_2| < |1 + k_1|$. Can you make further simplification of this? No, we have to find out by taking actual numbers. Can you draw a map similar to the stability triangle? We had drawn the stability triangle d_2 versus d_1. Can you draw here k_1 versus k_2? Would it be a triangle? It is just a question arising at this point in time and is left to you.

(Refer Slide Time: 49:02 - 54:27)



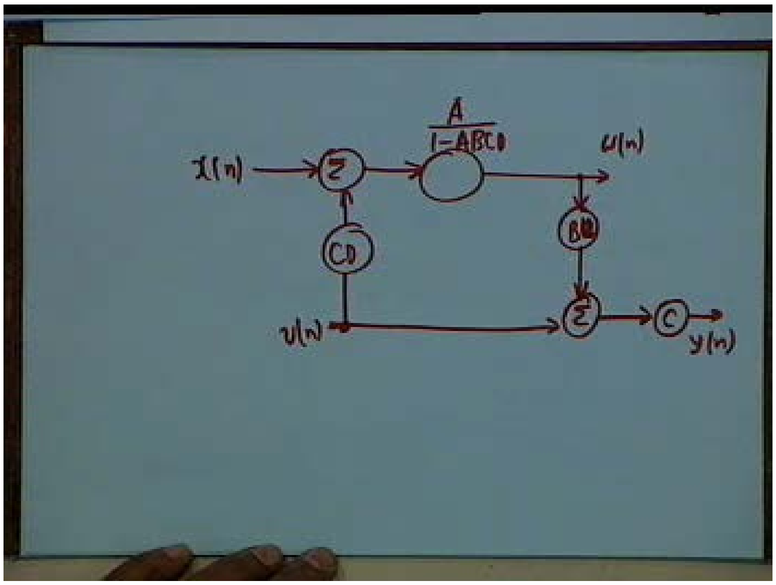
Another important thing in analysis and synthesis of digital signal filter structures is the so called Delay Free Loop. To illustrate this I have taken another example, problem 6.1 in Mitra, where in some part of the digital filter structure you get a flow diagram like the one shown in the Figure. I will state the problem later. There is no delay in the loop so y(n) will either increase or decrease continuously and in no time the signal either builds up to infinity or goes down to 0. Such a loop cannot be implemented in practice. For a loop, you must have a delay. Delay free loops are unstable. We either get nothing from the output or we get an infinite output. Infinity of course we cannot get because number of bits is limited and therefore you will get a saturated output. For stability, a necessary condition is that there must be a delay somewhere. If by mistake such a delay free loop occurs in your realization, you can always avoid it by an equivalent structure. Here, for example, you can write w(n) = A [x(n) + w(n) (BCD) w(n) + CD v(n)]. In other words, you can write w(n) in terms of only x(n) + CD v(n). Then you can write the other equation y(n) as C times [Bw(n) + v(n)].

(Refer Slide Time: 54:30 - 56:17)



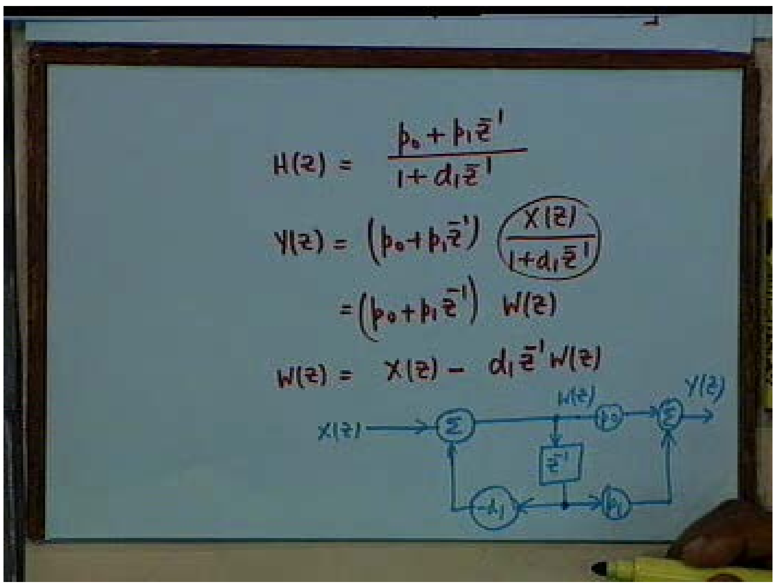
So y(n) = C (Bw(n) + v(n)). One of the equations was w(n) (1 – ABCD) = Ax(n) + ACD v(n) which can be written as A[x(n) + CD v(n)]. Therefore w(n) = [A/(1 – ABCD)] (x(n) + CD v(n)), similarly y(n) can be calculated.

(Refer Slide Time: 56:18 - 58:00)



The final realization is shown in the Figure. Now you see that in this diagram, there is no loop. Whenever a loop occurs due to a mistake on your part or somebody has given you the structure like that then you break the loop by writing equations and making sure that these equations are implemented without recursion. There is no recursion here and there is no feedback loop without a delay. So delay free loops should be avoided.

(Refer Slide Time: 58:07 - 1:00:57)



Now we consider the question of canonic delay structure. We recall the first example; (p_0 + p_1 z^{-1})/(1 + d_1 z^{-1}) was delay non-canonic. In order to make it canonic, we represent this as (p_0 + p_1 z^{-1}) W(z), where W|z| = X(z)/(1 + d_1 z^{-1}). Then W(z) = X(z) – d_1 z^{-1} w(z). I have written it in a particular fashion, so that the realization is obvious, as shown in the Figure. To construct Y(z), all you have to do is multiply W(z) by p_0 and add it to W(z) z^{-1} p_1. And you have used only one delay. This is the delay as well as the multiplier canonic structure.