Digital Signal Processing

Prof. S.C. Dutta Roy

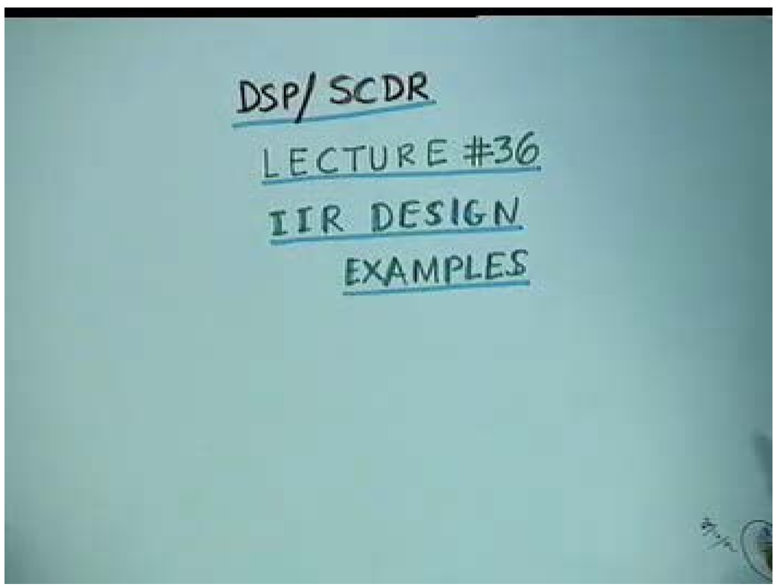
Department of Electrical Engineering

Indian Institute of Technology, Delhi

Lecture - 36

IIR Design Examples

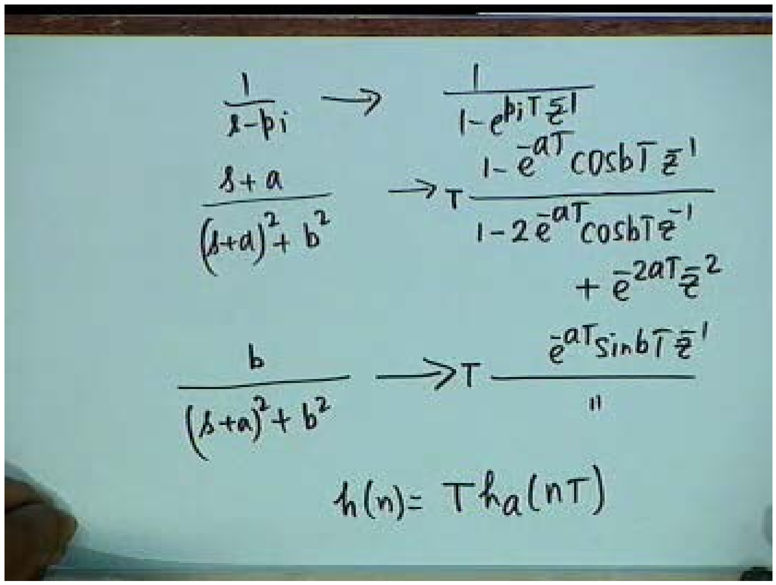
(Refer Slide Time: 01:06 min)



This is the 36th lecture and today we will continue our IIR design example by both Bilinear and Impulse Invariant Transformations. Later on, we shall divorce the Impulse Invariant Transformation with the understanding that you go to impulse invariance only when the requirement is not very stiff. The advantage of IIT is its simplicity but the disadvantage is that there is always some aliasing and the sampling frequency usually has to be very high to be able to contain the aliasing distortion within permissible limits. But in common with Bilinear Transformation, it is a stable transformation. It is guaranteed that a stable analog filter shall transform to a stable digital filter. We also showed that impulse invariance does not necessarily imply step invariance; they are generally different. I gave you the rule for transforming an analog filter to a digital filter by impulse invariance.

(Refer Slide Time: 02:46 - 04:35 min)

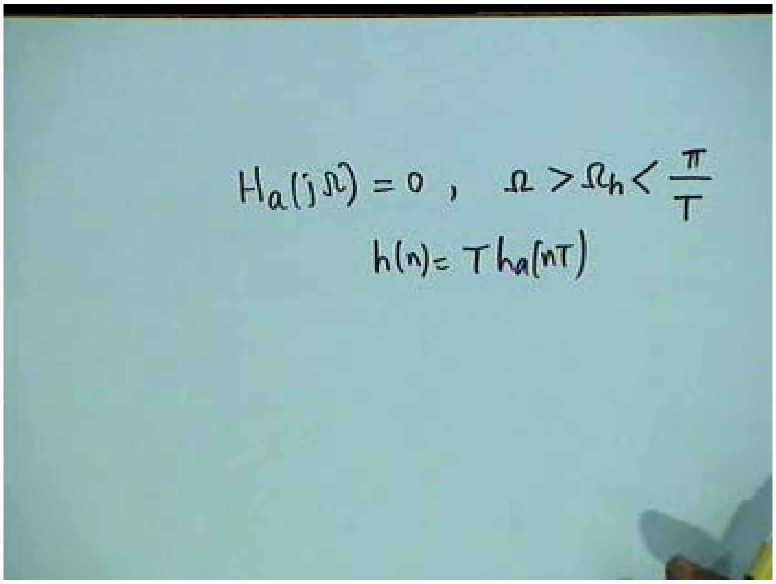
For a simple pole in $H_a(s)$, the term $1/(s – p_i)$ goes to $T/(1 – e^{p_iT} z^{-1})$.



We also gave two relations which we mostly use for complex poles. If I have $(s + a)/[(s + a)^2 + b^2]$, then in IIT, it transforms to $T (1 – e^{aT} \cos bT z^{-1} /(1 – 2 e^{-aT} \cos bTz^{-1} + e^{-2aT}z^{-2})$. The other relationship is that $b/[(s + a)^2 + b^2]$ transforms to $e^{-aT} \sin bT z^{-1}$ divided by the same denominator. The multiplication by T comes because we take $h(n) = T \times h_a(nT)$. This multiplication is needed; otherwise there is a problem of overflow.

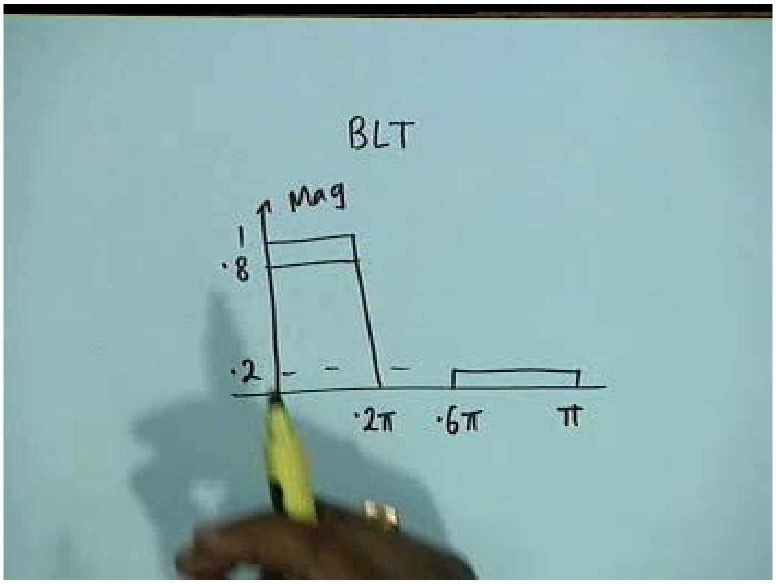
These are the only three things that you have to remember. Usually you will not get a case of repeated poles, but if repeated poles come, then the design effort may not have any advantage over Bilinear Transformation. It is for the simplicity in design that you are attracted to IIT. If you get a multiple pole, you better use the design effort with BLT and get a better filter. With the normalization $h(n) = Th_a(nT)$ and band limited transfer function i.e. $H_a(j\Omega) = 0$ for Ω greater than Ω_h which is less than π/T, your IIT digital filter shall work satisfactorily.

(Refer Slide Time: 05:22 - 06:10 min)



Here the sampling theorem must be obeyed. You cannot have $H_a(j\Omega) = 0$ over a band of frequencies; you have to put a tolerance may be 10^{-2} or 10^{-3}. That is the limitation of Impulse Invariant Transformation. But as I said, in very simple situations, like you require in ECG filtering or a vision signal filtering where the requirements are not too stiff, you can use impulse invariance and the industries use this even now. On the other hand, in Bilinear Transformation, there is absolutely no aliasing.

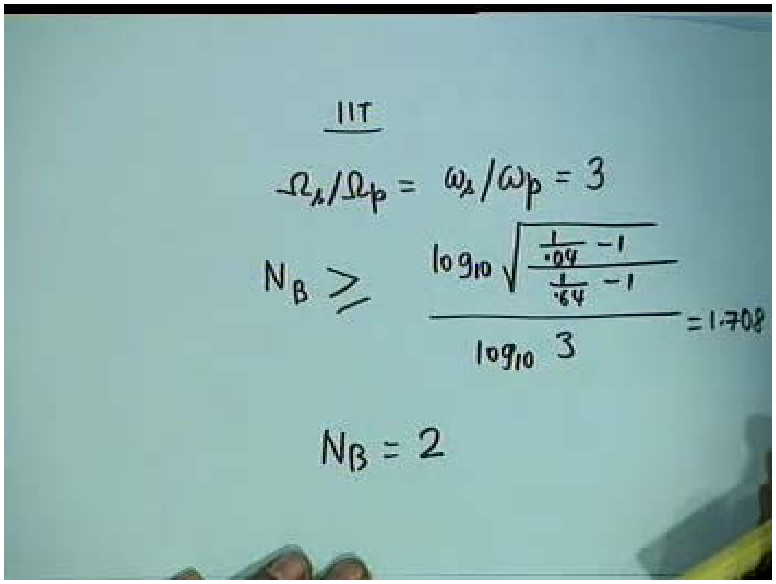
(Refer Slide Time: 06:31 - 07:59 min)



The only disadvantage is that the frequency scale, an infinite one in the analog domain, is compressed into a finite scale. So, necessarily there will be crowding at one end and thinning at the other end; this is called warping. But warping is of no concern because by pre-warping we can take care of that. If we anticipate a distortion, then we pre-distort so the pre-distortion combined with distortion gives you a perfect result. It only complicates the design because you have to calculate tangents and tan inverses.

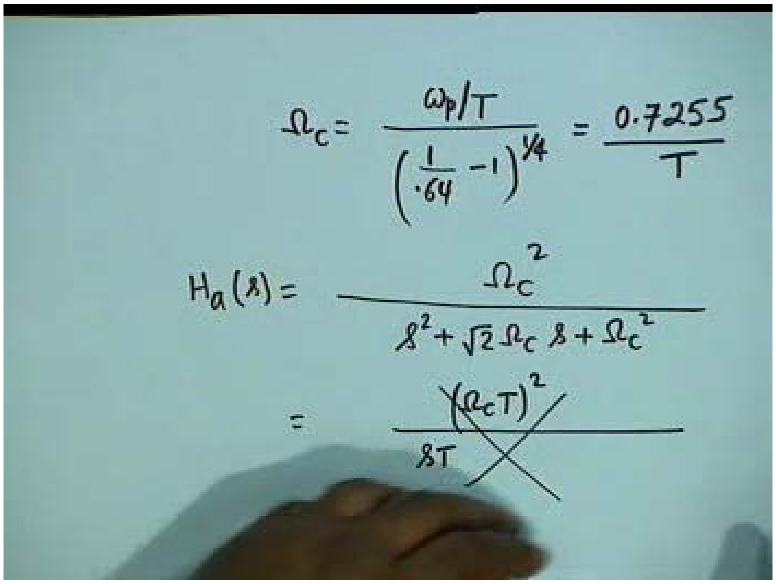
And then we took a simple example. The specification was to have the magnitude between 0.8 and 1 in the passband and $\leq$ 0.2 in the stopband. Also, $\omega_p = 0.2\pi$ and $\omega_s = 0.6\ \pi$. We made a design by a second order Butterworth filter using Bilinear Transformation. Today we shall redesign for the same specs by impulse invariance technique. If you want to design this filter by IIT, the only difference is that Ω_s/Ω_p, that is the transition ratio for the corresponding analog filter, shall be simply the ratio ω_s/ω_p. In BLT, it was the ratio of tangents, now it is simply a linear relationship and this ratio = 3.

(Refer Slide Time: 08:12 - 09:35 min)



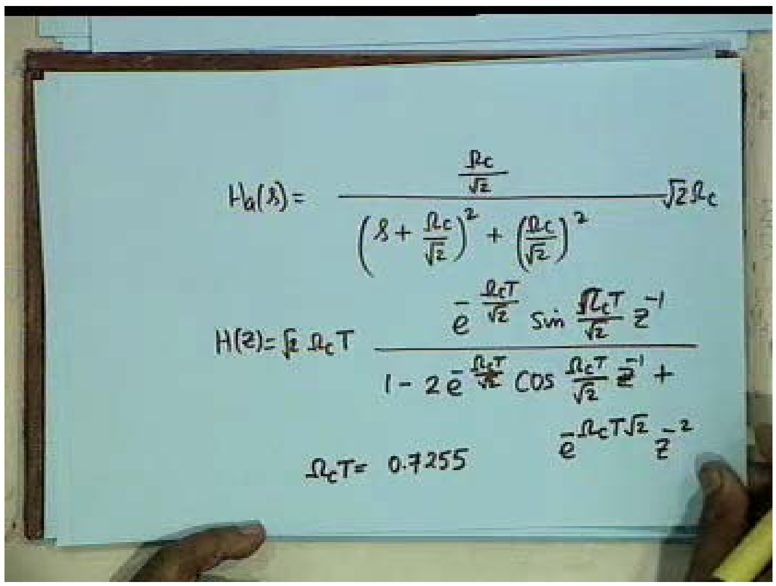
We still continue to use Butterworth. Here calculation gives $N_B \geq 1.708$ and therefore N_B is still = 2. In order to design a Butterworth filter, you also require the value of the 3 dB cutoff frequency Ω_c. here it is $(\Omega_p/T)/(1/0.64 - 1)^{1/4}$ (instead of the tangent) and this calculates out as 0.7255/T.

(Refer Slide Time: 09:39 - 11:29 min)



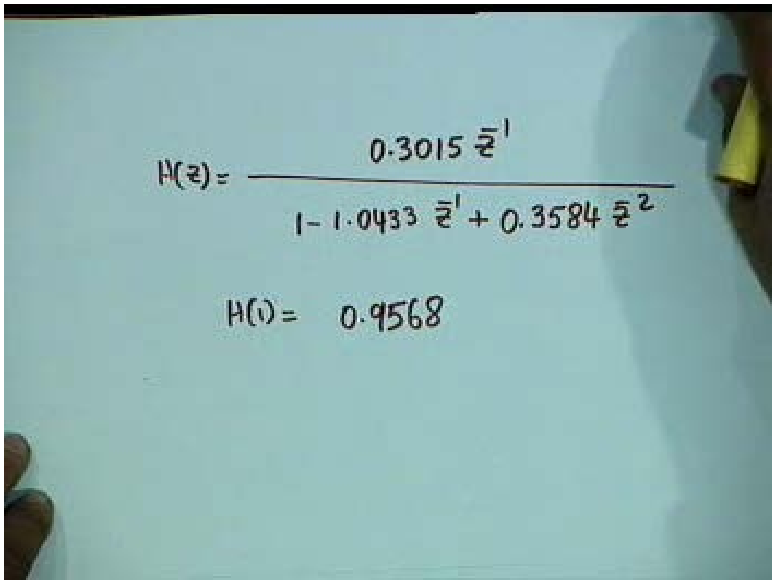
Now I can immediately write $H_a(s) = \Omega_c^2/(s^2 + \sqrt{2}\ \Omega_c s + \Omega_c^2)$. What we have to do is to decompose the denominator into the form $(s + a)^2 + b^2$. Also, we have to adjust the numerator constant to be equal to b, multiplied by another constant. The denominator is obviously $(s + \Omega_c/\sqrt{2})^2 + (\Omega_c/\sqrt{2})^2$.

(Refer Slide Time: 11:37 - 14:42 min)



To bring $\Omega_c/\sqrt{2}$ in the numerator we write $\Omega_c^2 = \sqrt{2}\Omega_c.(\Omega_c/\sqrt{2})$. My digital filter H(z) shall be T multiplied by the transformed form of $H_a(s)$. So in the numerator you get $\sqrt{2}\ \Omega_c T\ e^{-\Omega_c T/\sqrt{2}} \times \sin(\Omega_c T/\sqrt{2})\, z^{-1}$, while the denominator is $1 - 2\, e^{-\Omega_c T/\sqrt{2}} \cos(\Omega_c T/\sqrt{2}) z^{-1} + e^{-\Omega_c T\sqrt{2}} z^{-2}$. Now you know the value of $\Omega_c T$, which we found as 0.7255. Putting this value, the final result is that the numerator of H(z) is simply $0.3015\, z^{-1}$. In the Butterworth design by BLT, we had $(1+ z^{-1})^2$ in the numerator; you see the difference. So at $\omega = \pi$, BLT Butterworth was guaranteed to give 0 but there is no guarantee here. Here it will be a small quantity, but not 0 because of aliasing. The denominator is $1 - 1.0433\, z^{-1} + 0.3584\, z^{-2}$. You must keep as many digits as permitted by your calculator, because of the possibility of errors due to quantization of numbers.

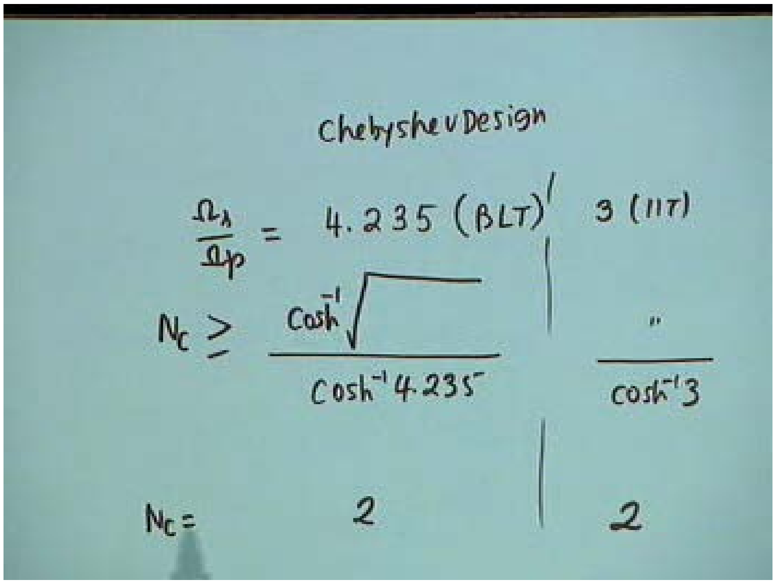
(Refer Slide Time: 14:47 - 16:26 min)



You should keep as many digits as possible because the filter design is extremely sensitive to quantization or to truncation errors. For example, the dc gain H(1) should have been 1, but here it is 0.9568 which is quite a substantial deviation, more than 4 % from the unity value.

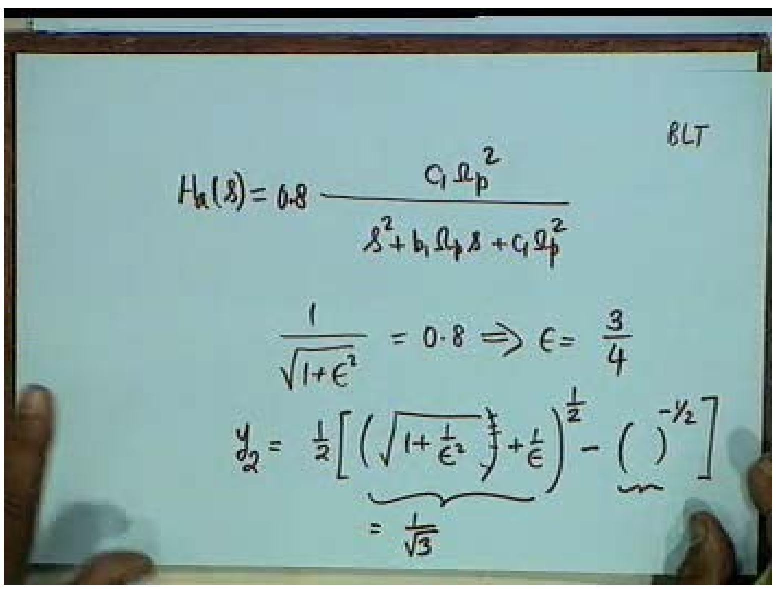
Let us look at Chebyshev design now for the same example. Let us take a short cut because we have already calculated the values of transition ratio Ω_s/Ω_p and some other expressions. For Bilinear Transformation Ω_s/Ω_p is 4.235 and for Impulse Invariant Transformation, the ratio is 3. It turns out that in both cases, N_c calculates out to 2. The example was so chosen that the order is not changed.

(Refer Slide Time: 16:46 - 17:52 min)



With N_c = 2, your transfer function will have a multiplication by T in IIT, not in BLT. The transfer function $H_a(s)$ have a multiplying factor $1/\sqrt{(1 + \varepsilon^2)}$ for $H_a(0)$ to be the correct value.

(Refer Slide Time: 18:08 - 23:34 min)

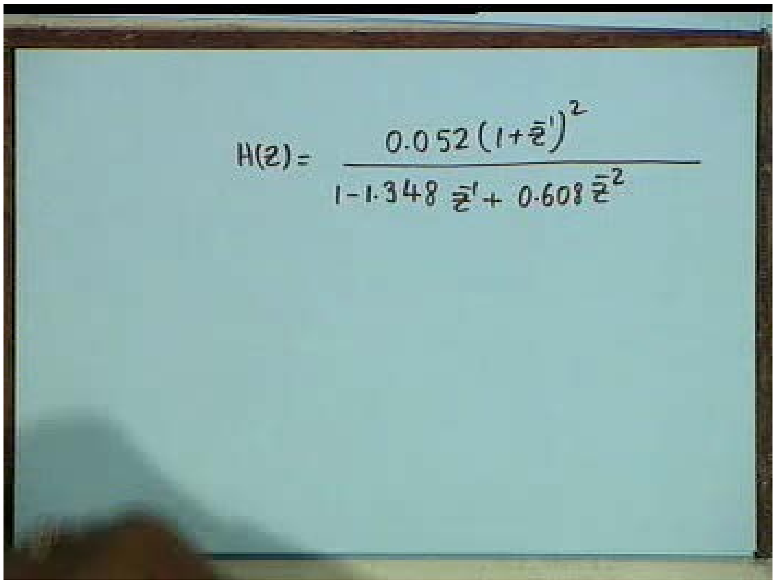


The magnitude of even order Chebyshev designs start from the lower end of the pass band tolerance and therefore you must find out $1/\sqrt{(1 + \varepsilon^2)}$. This is given as 0.8, and from this, ε is calculated (required for finding out y_2, c_1 and c_0) as 3/4. Once again, as long as possible, keep numbers as fractions; then it will not give rise to truncation error. The transfer function is $C_1\Omega_p^2 \times 0.8/(s^2 + b_1\Omega_p s + c_1\Omega_p^2)$. In order to calculate b_1 and c_1 you require the value of y_2. And y_2 is $(1/2)[\sqrt{(1 + 1/\varepsilon^2)} + 1/\varepsilon)^{1/2} – (\sqrt{(1 + 1/\varepsilon^2)} + 1/\varepsilon)^{-(1/2)}]$.

If you substitute the value of ε, y_2 calculates out to $1/\sqrt{3}$ and therefore b_1 which is $= 2y_2 \sin(\pi/4)$ comes out as $\sqrt{2/3}$; $c_1 = y_2^2 + \cos^2 \pi/4$ and this comes out as 5/6. We also require Ω_p which is (2/T) tangent of $0.1\pi = (2/T) \times 0.3249$. What we require is $\Omega_p T/2$ and that is 0.3249. The transfer function can be written after multiplying both numerator and denominator by $T^2/4$, and recognizing that $(sT/2) = (1– z^{-1})/(1+ z^{-1})$. The result is independent of T, so you do not have to bring T anywhere.

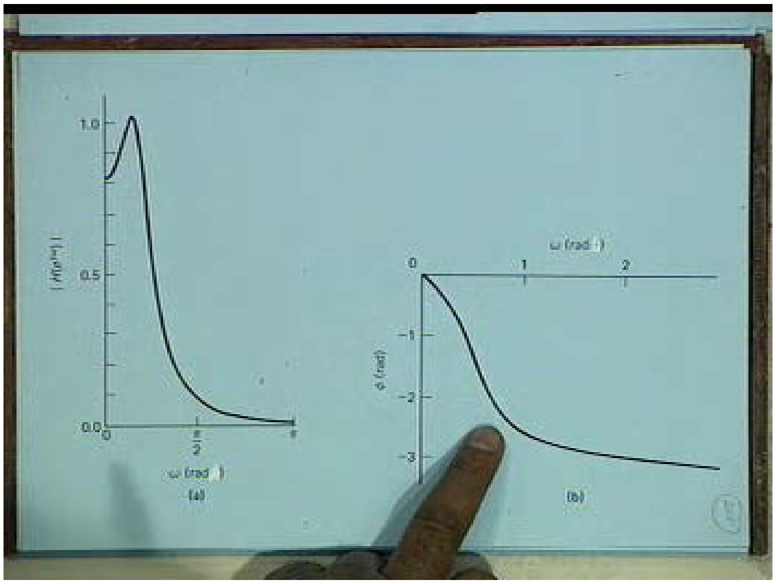
Later on you can assume T = 2 and calculate out as if that multiplying factor is not there. After skipping some of the intermediate computations, the final result is $0.052\ (1+ z^{-1})^2/(1 – 1.348\, z^{-1} + 0.608\, z^{-2})$.

(Refer Slide Time: 24:30 - 25:00 min)



The magnitude response of this transfer function, as shown in the next figure, does not start from 0.8; it is slightly raised. The maximum also is not 1. This happens because we computed only up to three places of decimal.

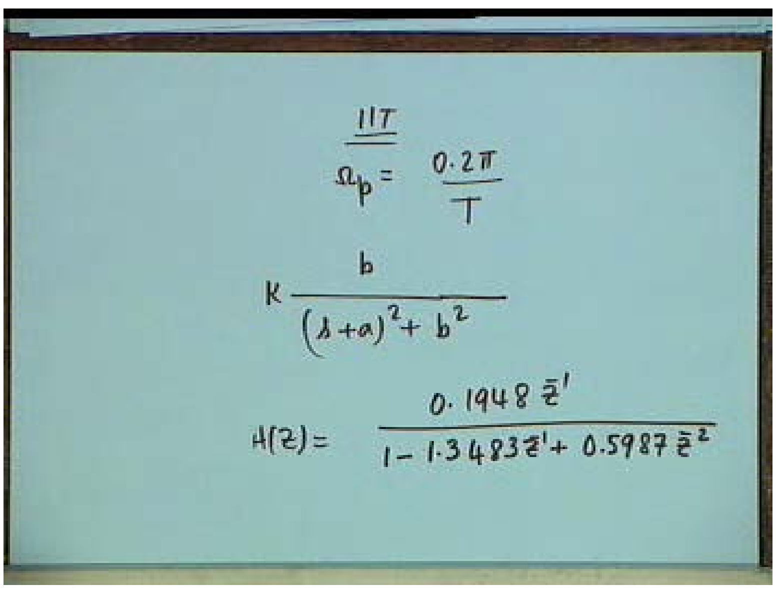
(Refer Slide Time: 25:13 - 31:17 min)



This is the effect of truncation error; even up to three places of decimal is not good enough. As far as phase is concerned, it is approximately linear up to about 0.8; radians and then the nonlinearity starts. So if your signal processing or pass band is confined to this limit, then you can be reasonably certain that there will be no delay distortion. If you go beyond this then you have to use an all pass filter for equalization.

For IIT realization of this example, we have already calculated the order which is two, and if we use IIT the only change will be that Ω_p, which is required in the transfer function, will now be $0.2\pi/T$, and not (2/T) tangent of $\omega_p/2$. The same transfer function in the form $Kb/[(s + a)^2 + b^2]$ holds here too.

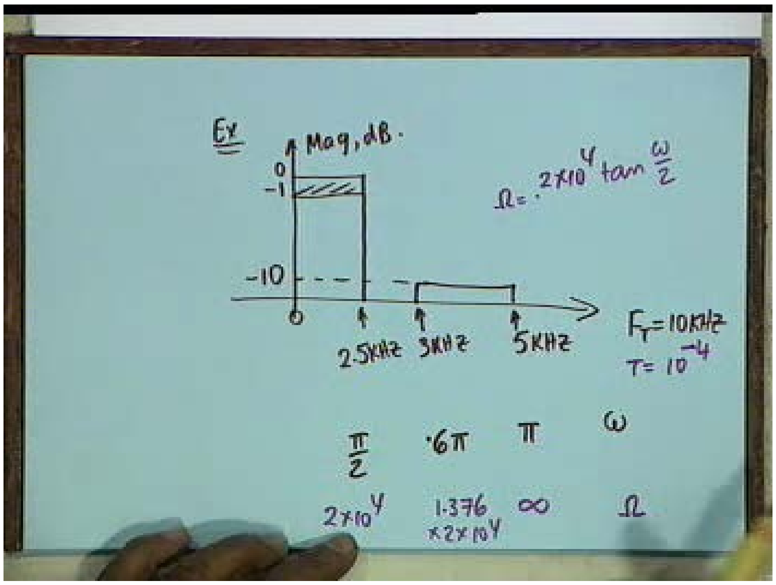
(Refer Slide Time: 26:49 - 29:51 min)



The only thing we have to do is to change $\Omega_p T$ to 0.2π. Once you do this, you appeal to the formula and write the transfer function H(z) as $K \times e^{-aT} \sin bT\ z^{-1}$ divided by the quadratic denominator. The final result is $0.1948\,z^{-1}/(1 - 1.3483\,z^{-1} + 0.5987\,z^{-2})$. Next, you have to put $z = e^{j\omega}$, and plot its magnitude and phase. It will be seen that it starts from 0.8 and the maximum is also 1 because at very low values of frequency, IIT works perfectly alright. The deviation starts at higher values of digital frequency and you would see that at π, it is considerably raised; this is the effect of aliasing. If you had increased the sampling frequency, it would have come closer to zero.

As far as the phase response is concerned there is nothing much to choose; again up to about 0.8, it is approximately linear. For any design, although your emphasis is on the magnitude response, the desirable results may not be obtained because of extraordinarily large delay distortion. If you are doing the magnitude calculation on MATLAB, there is no reason why you cannot use it to find the phase response also. Next we take a higher order design example. We now require 1 dB tolerance in the pass band that extends from 0 to 2.5 kilo Hertz.

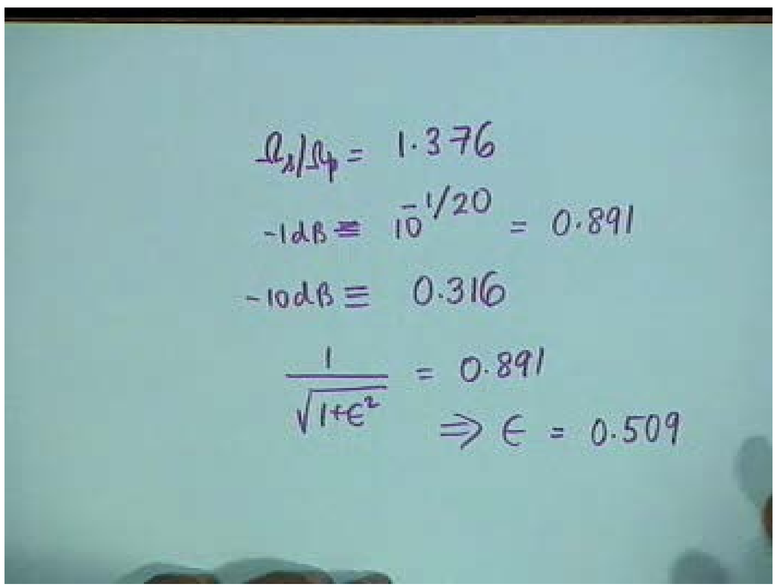
(Refer Slide Time: 31:46 - 35:50 min)



Beyond 3 kilo Hertz, the magnitude response must be down by at least 10 dB. One could expect that this would be satisfied by a sufficiently low order filter but you must also see that the transition band is pretty narrow. The stopband end is 5 kilo Hertz which means that the sampling frequency, F_T must be 10 kilo Hertz.

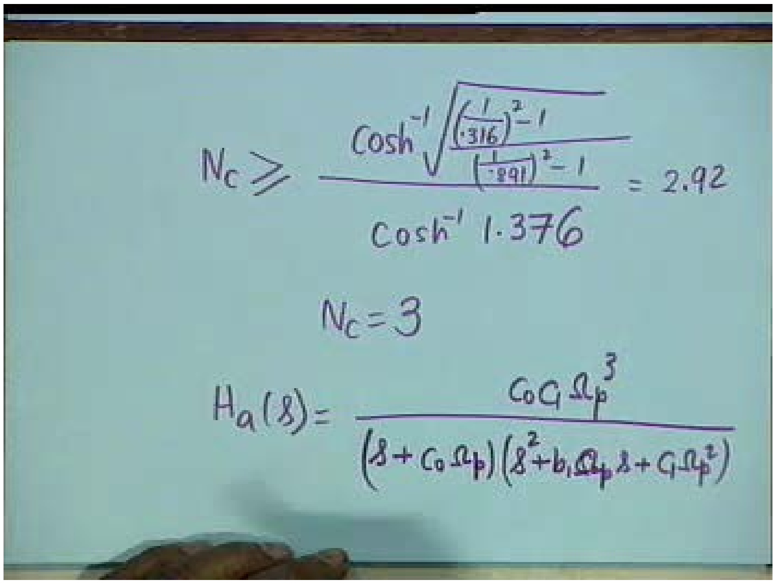
The first thing you do is to convert these specifications in terms of ω, the normalized digital frequency. 5 kilo Hertz must correspond to π; 2.5 kilo Hertz must correspond to $\pi/2$ and 3 kilo Hertz must correspond to 0.6 π. Then the next thing to do is to convert ω's to Ω's and we will work this out for Bilinear Transformation only. Therefore $\Omega = 2/T$ $(= 2 \times 10^4)$ tangent of $\omega/2$. Obviously π corresponds to infinity, 0.6 pi corresponds to $1.376 \times 2 \times 10^4$ (with a little bit of practice we can even omit 2×10^4). We could assume T = 2, because the final transfer function is independent of T. So Ω_s/Ω_p, tangent of $\pi/4$ being unity, is 1.376, which is sufficient to design the filter.

(Refer Slide Time: 35:56 - 37:23 min)



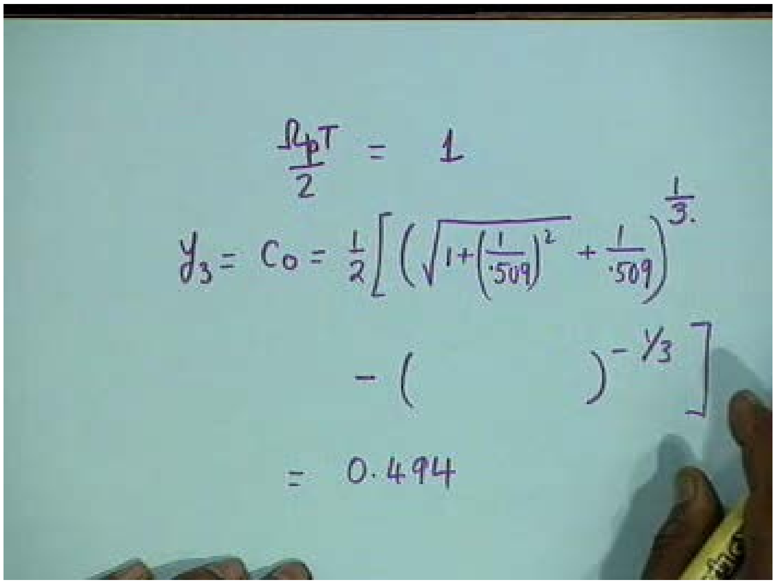
I must convert tolerances given in dB to corresponding ratios. – 1dB is equivalent to $10^{-1/20}$ and this comes as 0.891. – 10dB will correspond to $10^{-1/2} = 1/\sqrt{10}$ and this comes as 0.316. Obviously, $1/\sqrt{(1 + \varepsilon^2)}$, if I am asking for a Chebyshev design, is 0.891 and therefore I can calculate ε which is required for calculating y_N. I do not know yet what is N but ε comes out as 0.509. I truncated at the third place. Once we have found ε, the other parameter is N_c and $N_c \geq \cosh^{-1} [((1/0.316)^2 - 1)/((1/0.891)^2 - 1)]/\cosh^{-1}1.376$. This calculates out to 2.92. Use the log natural to calculate $\cosh^{-1}$. Do not use those simplified procedures I showed you earlier because digital filter design is prone to so many errors at many stages.

(Refer Slide Time: 37:31 - 39:57 min)



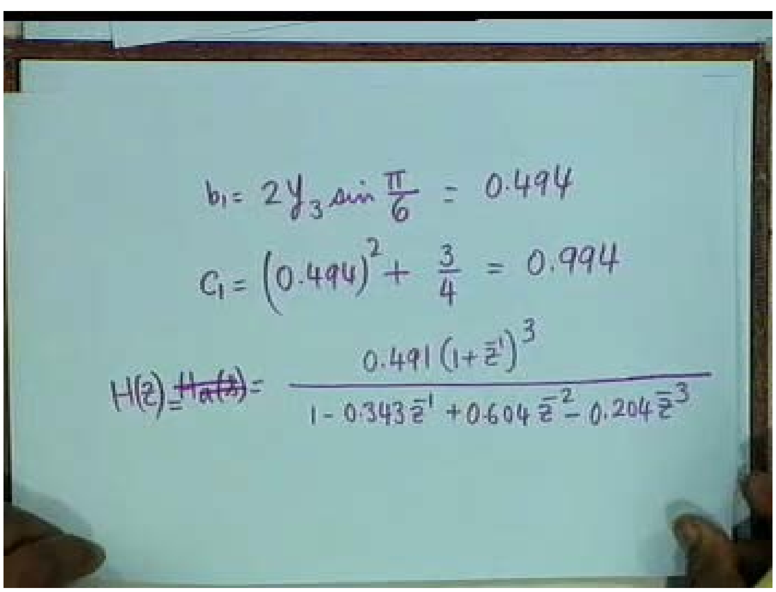
For example, if this came out as 2.99 then you do not know whether it is 2.99 or 3.01. If it comes out as 2.001 then you are in deeper trouble so you will have to use as many digits as are permitted by the calculator to calculate this. Here $N_c = 3$ and once I have got N_c, I can write down the expression for $H_a(s)$. Since it is third order; the magnitude starts from 1 at $\Omega = 0$ and therefore the denominator is $(s + c_0\Omega_p) \times (s^2 + b_1\Omega_p s + c_1\Omega_p^2)$ and the numerator will be $c_0c_1\Omega_p^3$. I know the value of Ω_p but you require only $\Omega_pT/2$ and that is equal to 1. We have to calculate y_3 which is the same as c_0.

(Refer Slide Time: 40:14 - 41:24 min)



y_3 = ½[(√(1 + $(1/0.509)^2$) + $(1/0.509))^{1/3}$ – (√(1 + $(1/0.509)^2$) + $(1/0.509))^{-1/3}$]; my calculation gives this as 0.494. Now you have to calculate b_1 and c_1 where b_1 is $2y_3 \sin\pi/6$, which simplifies to 0.494. And c_1 is $(0.494)^2 + \cos^2 \pi/6 = 0.994$.

(Refer Slide Time: 41:28 - 44:08 min)

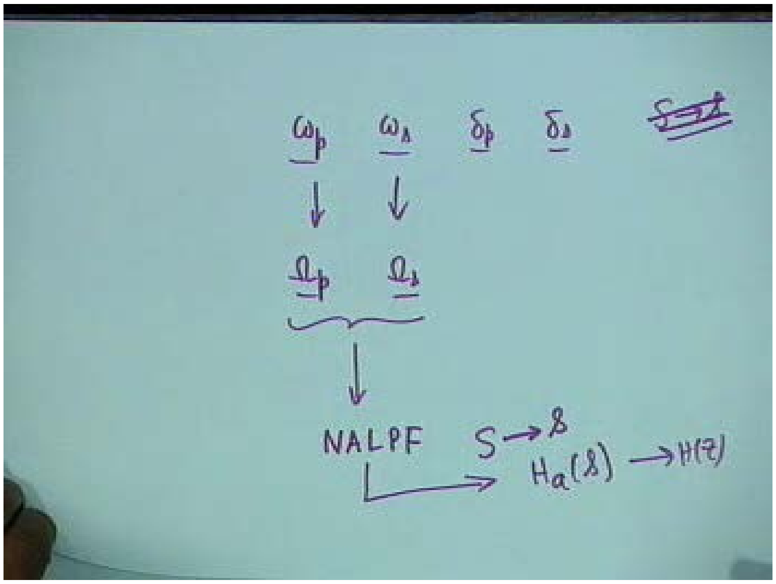


It is very close to 1 but not exactly 1. You can now substitute $\Omega_p T/2$ and sT/2 and so on. The final result is 0.491 $(1 + z^{-1})^3$ (if it does not come then you have made a mistake) in the numerator and in the denominator, it is $1 - 0.343z^{-1} + 0.604z^{-2} - 0.204z^{-3}$.

Now it is up to you to put $z = e^{j\omega}$, find its magnitude response and see how much is the deviation because of truncation upto the third place of decimal. You can also see, if you use a MATLAB, how the error gradually diminishes by extending the number of digits to the right. We shall close this example here.

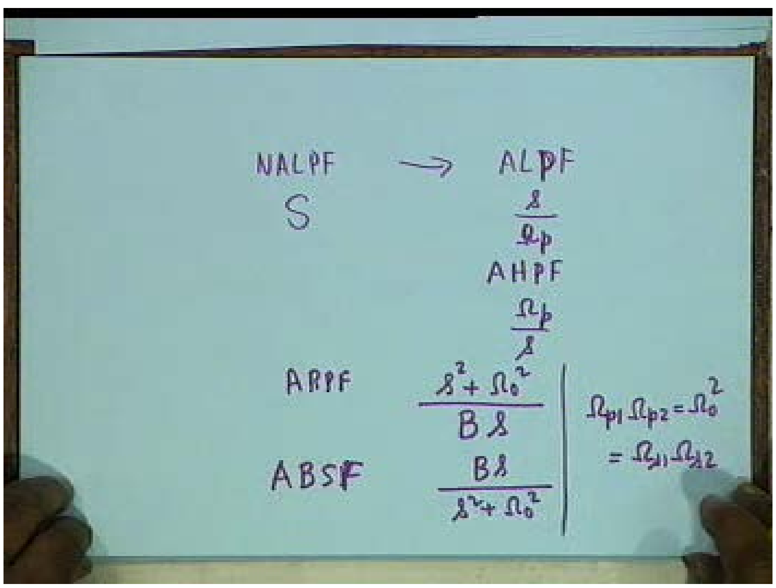
So far we have confined to only low pass filter. How do you design other kinds of filters like high pass, band pass and band stop? In general, the specs are ω_p's, ω_s's, δ_p's, and δ_s's. Multiple values of each spec are needed. For example, even a simple band pass filter has ω_{p1} and ω_{p2}, ω_{s1} and ω_{s2}, δ_p, and δ_{s1} and δ_{s2}.

(Refer Slide Time: 46:14 - 48:49 min)



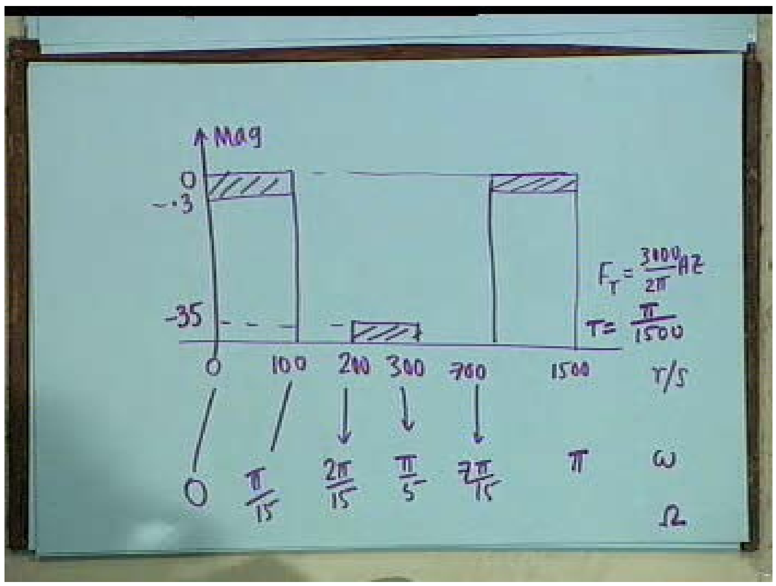
In multi-band pass filter, δ_p may be different in different pass bands; then you choose the minimum tolerance. But if it is a stiffer requirement, then you may have to use a different kind of transformation from S to s; simple band pass or band stop transformation may not suffice. You may have to find your own transformation. What you do is you convert ω_p and ω_s into Ω_p and Ω_s. δ_p and δ_s remain the same and from these, you find out the Normalized Analog Low Pass Filter. For the Normalized Analog Low Pass Filter, δ_p and δ_s are known, Ω_p the end of the pass band, is 1. So all that you have to find out is the transition ratio, for which you require Ω_s, the beginning of the stop band of the normalized filter. From this Normalized Analog Low Pass Filter, you use the transformation S to s and therefore you get $H_a(s)$. Then you obtain the digital filter by using the Bilinear Transformation: $s = (2/T)\,(1-z^{-1})/(1+z^{-1})$. Now let us recall the transformations. If in the Normalized Analog Low Pass Filter the frequency variable is S, then from here to a de-normalized LPF, we simply put $S = s/\Omega_p$. If it is analog high pass filter, then we use $S = \Omega_p/s$. Here Ω_p is the required cutoff frequency in the transformed filter.

(Refer Slide Time: 48:57 - 51:02 min)



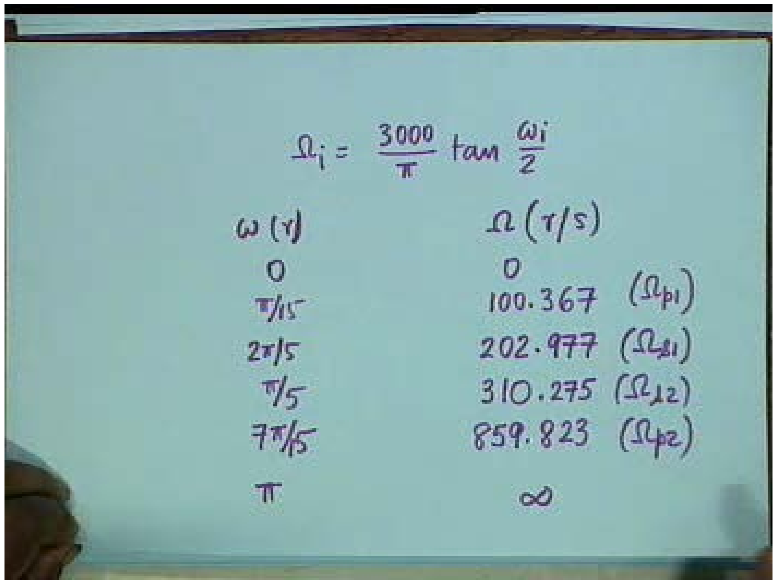
If it is analog band pass filter, then we use $s = (s^2 + \Omega_0^2)/(Bs)$. If it is analog band stop filter, then we use the reciprocal of this, which is $Bs/(s^2 + \Omega_0^2)$ where the specs may not ensure geometric symmetry. But you have to ensure that the product $\Omega_{p1}\,\Omega_{p2}$, which is equal to Ω_0^2 should be equal to Ω_{s1} times Ω_{s2}. If it does not happen, then for a bandpass case, either decrease Ω_{s2} or increase Ω_{s1} but do not do the reverse. Now, in order to illustrate the process we shall completely work out a fairly complicated example. We take a band stop filter design of fairly stiff tolerances. The tolerance is 0.3 decibel in the pass band and the pass band extends from 0 to 100 radians per second. From 200 to 300 rad/sec, the attenuation must be at least 35 decibels. For the other pass band, fortunately the tolerance is the same, and it extends from 700 to 1500 rad/sec which means that the sampling frequency F_T = 3000 radians per second.

(Refer Slide Time: 51:26 - 54:02 min)



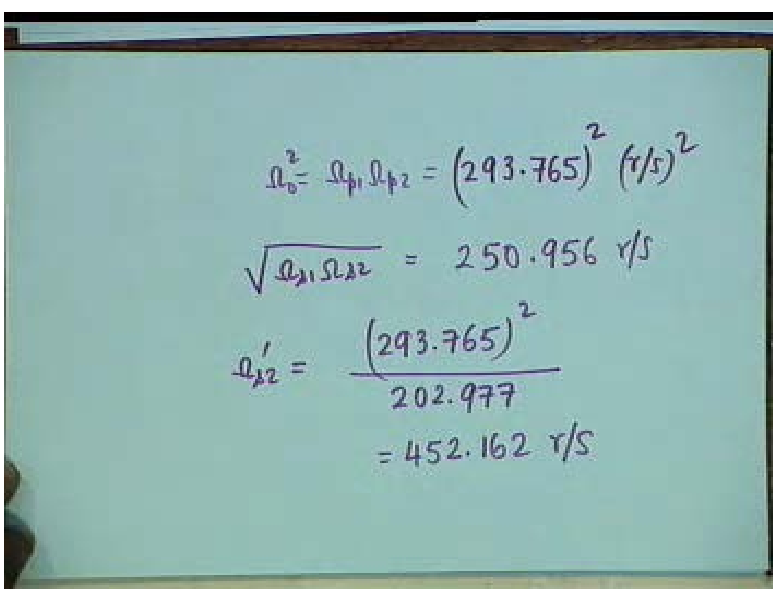
So 3000/(2π) Hertz is the sampling frequency. And T is π/1500. First thing to do is to convert the specs to normalized digital filter specs. And ω would be π at 1500 rad/sec, and 7π/15, 3π/15 (which is π/5), 2π/15 and π/15 would correspond to 700, 300, 200 and 100 rad/sec. The next step is to calculate the Ω for the analog filter corresponding to pass and stopband edges, by using Ω_i = (2/T) tangent of $\omega_i/2$. My calculation gives π/15 converting to 100.367 (this is Ω_{p1}), and 2π/15 converting to 202.977 (this is Ω_{s1}).

(Refer Slide Time: 54:09 - 57:17 min)



Similarly, $\pi/5$ converts 310.275 (this is Ω_{s2}) and $7\pi/15$ converts to 859.823 (this is Ω_{p2}). Now we should calculate Ω_0^2 from Ω_{p1} and Ω_{p2} and this comes out as $(293.765)^2$ (radian per second)^2. Then you also have to find out the square root of Ω_{s1} Ω_{s2} and see whether geometric symmetry exists or not. And if you put down the values my calculation gives 250.956 radians per second.

(Refer Slide Time: 57:24 - 59:01 min)



Obviously geometric symmetry does not exist but you have to bring geometric symmetry and Ω_{s2} has to be increased. So change Ω_{s2} to $\Omega_{s2}' = (293.765)^2/202.977$ and this comes out as 452.162 radians per second. After you have done this, now you are in business and if you want to take a stock of this situation you draw the analog filter characteristic. Incidentally when you do this, convert 0.3 dB to a fraction; by the usual calculation, this comes to 0.966, a pretty stiff tolerance and this frequency now corresponds to 100.367 in radians per second. Ω_{s1} remains the same at 202.977 but Ω_{s2} has now changed to 452.162. The stopband tolerance is 35 decibel, which comes as 0.0178, and Ω_{p2} is 859.823. We shall proceed next time starting from this.