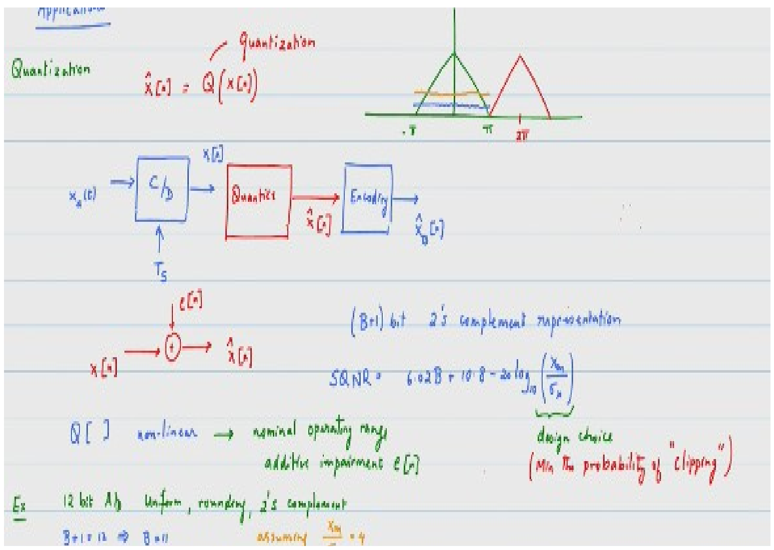
Multirate Digital Signal Processing
Prof. David Koilpillai
Department of Electrical Engineering
Indian Institute of Technology- Madras

Lecture - 40
Some More Applications of MDSP

Great, good morning. Let us begin a last lecture of the course we will today we will cover the applications of Multirate signal processing and let me begin with a quick review of what we had covered in the last lecture.

(Refer Slide Time: 00:28)



We are trying to look at quantization and whether any multi rate techniques could help us in the aspects of quantization, the conventional way is that you sample at or just above Nyquist a quantize the signals based on the number of bits of precision that you want. And then apply the coding scheme and then give out a value that is proportional to the quantized value, so this particular model gives us a way of characterizing the quantization noise.

And that would be as an adaptive impairment, so you think of quantization as a additive process where you add a error term or a noise term that gets added that gives you a perturbation of the un quantized signal, now yesterday we went through the lecture and we showed that when you have a B+1 bit representation, B+1 bit 2s complement representation, that is one of these standard forms that is used, and we derive the signal to quantization noise ratio.

The representation, so the SQNR can be expressed in the following way it was 6.02B+10.8-20 logarithm base 10 of the ratio of the maximum or full scale divided by the standard deviation of the input signal. So one observation is that this is a design choice, you have to choose how much you want to choose your full scale, design choice, and the design choice is controlled by your desire to minimize the probability of clipping.

So, you want to minimize the probability of clipping, the reason is probability, the clipping gives you a very nonlinear a distortion of very severe form of distortion, probability of clipping, you want to capture most of your signals within the range there for which you have designed your A to D, now clipping is nonlinear, is quantization nonlinear as well? You quantize A you get a certain value if you quantize B you get a certain value.

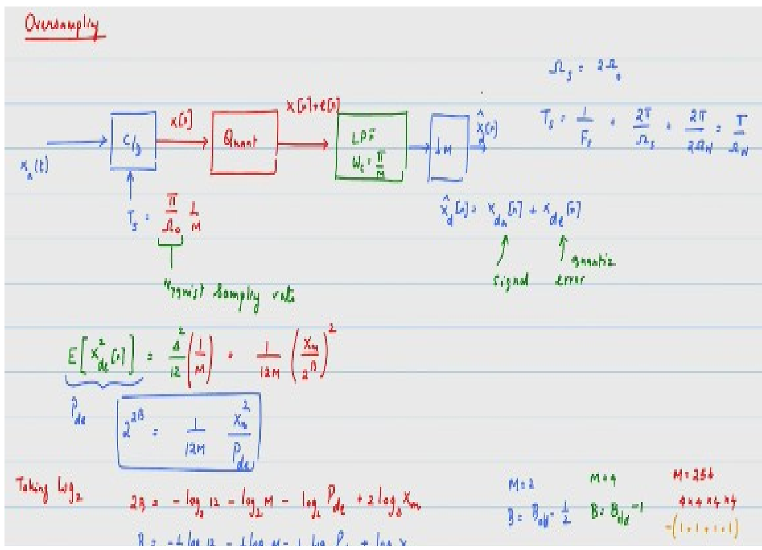
Add them together and quantize you will get the something totally different, so it is not linear so quantization itself is a nonlinear operation, so quantization is to begin with nonlinear, but we have come up with a linearized model where we said it is going to be in the range of operation in the, what we call as the nominal operating range that is you are not in the clipping zone, we have made it in additive impairment.

So, basically it is something we can analyse, there is a properties of linearity, super position all those would hold, so in the nominal operating range we have modeled it as a additive impairment and like, noise like impairment and which is very advantageous because in communication this is something, we do all the time and we have several tools by which we can assess the impact of these impairments.

So, this is the broad framework that we have worked with. I thought I could write down equation or just do one example just to strengthen what we had already done. Example for the, so supposing you are told that you have a 12 bit A to D, uniform 2’s compliment uniform quantization rounding using 2s complement comma 2s compliment, what is the expected SNR SQNR. First thing to keep in mind is that B+1=12.

The number of bits totally available to you is = 12 implies B for all the formula has to be B = 11 so the approximate SQNR from our expression will be 6.0 2* 11-1.25 if I have chosen the, this is assuming that we have chosen the full scale as by sigma X=4 then you will get this 1.25, if not you would have to recalculate that, so this comes out to be about 64.97dB, we can say that it is about 65dB that is what you will get a 12 bit quantizer with 2s compliment, uniform step levels and so to add to this we said that you could in, just make sure you are okay with this 65 dB that is roughly the range that you will get.

(Refer Slide Time: 06:35)



Now look at the oversampling element, over sampling element says I am going to do oversampling by a factor of M at the input, so the quantization will happen like before, basically quantization noise gets spread out and then we use a low pass filter with the appropriate down sampling to get the get back to the Nyquist rate or as close to Nyquist rate. And we said that there is a advantage that we can we can get with the over sampling rate.

And that over sampling rate based expression, yesterday we derived, let me just write that down for you so that we can build on this equation. So in this particular case the quantization noise expected value of Xde of n whole squared comes out to be 1 by 12 of the original expression Delta squared by M was the original expression and because of the, sorry delta squared by 12 was the original expression and 1/M is the modified expression.

Okay so 1/M is the new advantage or the new factors that we have brought in so effectively this is = 1/12 M Xm/2 power B whole squared, which you can a rewrite the equations. Yesterday we did a expression in terms of log based 10. I am going to do it slightly differently so just sort of get a different perspective. I am going to take logarithm based 2 for this expression so 2 power 2B= basically rewrite this.

It is 1/12 M Xm squared divided by, if I call this the power of the signal de, so this is divided by Pde this is a new equivalent representation of the previous expression now taking log base 2 logarithm base 2; I get 2B= – log base 2 of 12 – log base 2 of M- log base 2 Pde+2 log base 2 of X of m, and I am going to take across the other side just for – 1/2 log base 2 of 12- 1/2 log base 2 of M – 1/2log base 2 of + one half and 2 will cancel log base 2 of X of M.

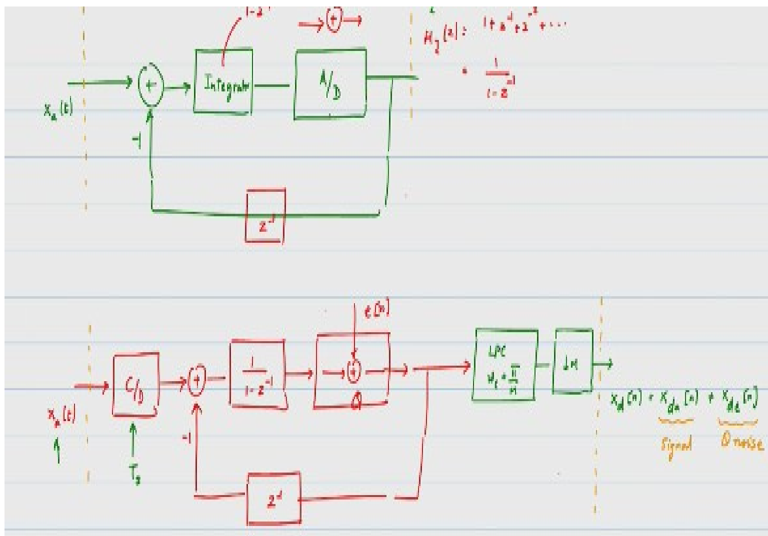
Okay now I am going to put a tick mark on those things that are constants that is a constant this is the constant this is the constant so basically, I want to relate B and M given this scenario the relationship between B and M says B = -1/2log M, so if I set them equal to 2, log base 2 will be = 1 B will be = the new value of B will be B = B old–1/2, correct whatever was the expression we call it as B old then – 1/2, so this is what we said yesterday.

That a doubling of the frequency translates into a reduction of 1/2 bit so B is the number of bits of a precision that you need to achieve a particular error variance and if you have M=1 you can go back to the original equation now if you have the new, with the oversampling, it becomes so similarly you can you can substitute, and you can find out what is the corresponding level, so M = 4 basically you will get B old – 1.

And every multiple of 4 subsequently you will get an additional reduction of 1 bit, so again what we are trying to work towards is; so if I ask you to tell me what is the, what would you get if M was = 256 of course you can plug in and do it, but you can write it as 4 times 4 times 4 times 4 times 4, 4 power 4 each of these would contribute 1 bit this is 1 bit 1 bit 1 bit 1 bit, add all of these together with the overall minus sign.

So, B will be = to B old – 4 bits you have reduced your precision requirements by 4 because you have increased your oversampling by a factor of 256. Now it turns out that this is good it is good to know that multi rate can give you a benefit of course we want to ask the question can we do a little bit more so that was where we ended the last class.

(Refer Slide Time: 12:31)



So, we said that as long as the input error is white, this is the best that we can do the only way that we could do is to color the input signal and we said that this would be a way to do the coloration and our goal today is to actually to derive this result and to build on that. The overall picture that we that we want to have is this block is very important for us in terms of our understanding.

The lower block is the equivalent discrete time representation, so this is a C to D block will assume that we would have done an oversampling by some factor so which means that at the other end we would, this would not be the we will have to do the down sampling, incorporate the down sampling so the down sampling will be a low pass filter with Omega C=Pi/M whatever has been your oversampling at this point.

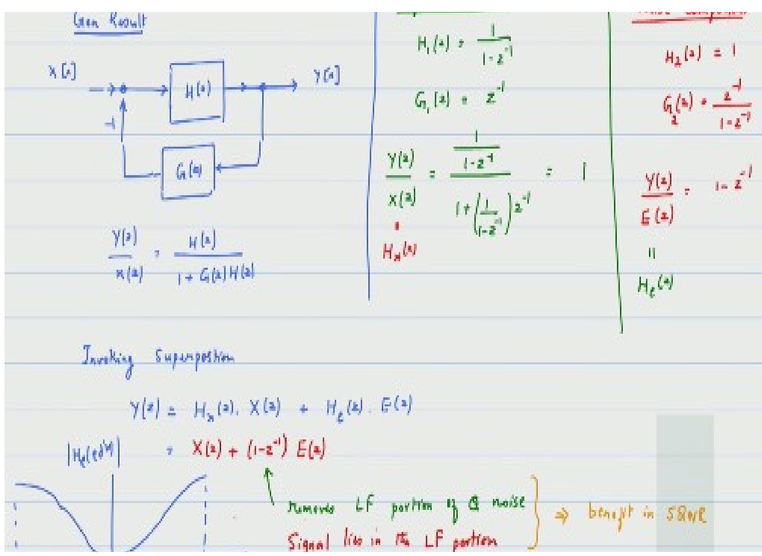
So, if you have over sampled by a factor of Ts, then we would do the corresponding down sampling by a factor of M and this is my output signal since its comes out of a down sampler, the notation used by Oppenheim Schafer that I have used the same; is a combination of Xda for n, ‘a’ stands for the input signal, the analog signal that means its signal component, the other part is Xde of n whatever is the error introduced by that.

So, just an indication this represents the signal component at the output other one in the quantisation noise component at the output so now an A to D converter : is it an analog device is it a digital device or is it a mixed signal device? What does the, input is an analog output is a discrete time signal so obviously it has to be a mixed signal device now where is the mixed signal boundary for us in this case?

So, if you were to look at it as the input at this point where the analog is taken in and then at the output the discrete time comes out or if you have to take it in so essentially what we are doing is we are doing some modifications inside the mixed signal portion of it, we will do a discrete time equivalent and then say that okay ultimately it is not just purely discrete time it is going to be a mixed signal element.

So, for the purposes of analysis we are using the lower block diagram and we will build on that.

(Refer Slide Time: 15:43)



So, the task that I had given you was to obtain the transfer function between the input and the output so let us write down a general result which helps us uniformly and so if I have the following setup where I have H of Z as my forward transfer function and this is an additive node input is X of n and I have a transfer function G on the lower branch which is the feedback branch.

And you would have seen this many times in the control discussion that the feed forward transfer function, feedback transfer function and the output is Y of n, so the overall transfer function Y of Z divided by X of Z is given by H of Z divided by 1+G of Z H of Z but if the - sign was not there you will get 1 -G of Z H of Z so this is the general form, so now go back to this figure and do a transfer function between the input signal.

Let this be the point at which we are interested in X of n to the output Y of n, we have not done the down sampling we have not done the filtering post band, basically just this portion of the transfer function so what we will do is for the signal component, here is the signal component, please keep the diagram as your reference and verify that the forward transfer function H1 of Z is given by 1 over 1-Z inverse.

And the feedback portion is given by G1of Z which is Z inverse the feedback portion so the signal transfer function Y of Z divided by X of Z is given by 1/ 1-Z inverse divided by 1 +1 / 1-Z inverse times Z inverse which gives us a transfer function which =1 that is a signal component, now move over to the noise component apply the same technique obtain the forward transfer function H2 of Z in this case, notice that there is no other gain term for transfer function is 1, G of Z the feedback portion it should be back to the point where the input signal entered the diagram so this will be = G2 of Z which is Z inverse divided by 1- Z inverse and Y of Z divided by E of Z, assuming there is no input X. I am just doing the analysis for this comes out to be 1– Z inverse just apply that, okay now we are ready to invoke super position.

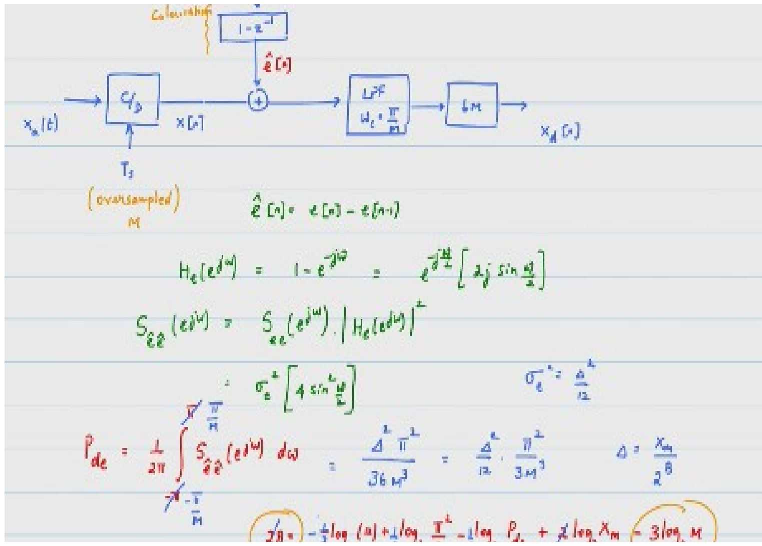
So, invoking superposition between the two linear inputs, one is X of n and the other one is E of n so the output must be the output due to each of those inputs so Y of Z, if I call this as H subscript x of Z and this 1 as H subscript e of Z so Y of Z = Hx of Z times X of Z+ He of Z times E of Z that comes out to be X of Z + 1 -Z inverse E of Z; so the input output between the whatever was the structure that we have implemented says that there is no the transfer function between the input X and output is 1, so there is no modification of the input signal but the error signal got multiplied by 1- Z inverse so what type of filter is this? If you sketch this 1-Z inverse basically says that a 0 at omega = 0 so it looks like a bowl, so this is frequency this is Pi this is 0, –Pi this is magnitude He e of j omega magnitude, so if I multiply my error signal has got a spectrum which is flat.

If you multiply by this transfer function the resultant now that is going to be showing up at my output is the original flat spectrum multiplied by a high pass spectrum which means the energies in the low frequencies have been eliminated and the high frequencies components are still left, there so the insight is that this type of a transfer function removes the low frequency noise components

Low frequency portion of the quantization noise, important to also highlight; the signal is lying in the low frequency portion so which means that effective signal to quantization noise ratio is now going to improve, signal lies in the low frequency portion of the spectrum so the combination of these two says I am going to see a benefit in SQNR. So how much of a benefit and how will we leverage it that is something we will like to quickly quantify and so is this okay?

What we have done; basically analyzing the transfer functions showing that the output is an input X of N+ a transfer function times E of Z

(Refer Slide Time: 22:53)



So here is an equivalent diagram if you would like to draw it down, draw, so the continuous time signal X a of t which passes through C to D converter oversampled let me just put a little note there that it is over sampled and let me also indicate oversampling factor as M. Then at this point it becomes a discrete time signal X of n over sampled. I need to add noise to it previously it was a white noise signal.

We now have the white noise components still present e of n that is still the old assumptions, you know sample to sample uncorrelated, but it is passing through a filter 1-Z inverse which is going to introduce the colouration so this is where the colouration is going to occur. Colouration as I mentioned, also means that there is sample to sample dependence which is what is built in basically there is memory into the; so we will call this as e hat of n.

Now these 2 get added together and it passes on and then eventually will go through a low pass filter with cut off omega C= Pi/ M then down sample by a factor of M and that will be our output X d of M; sort of a simplified diagram but it is very helpful to visualize what is happening. Now very quickly what is the type of colouration that we have introduced e of n = e of n - e of n -1 like a first order difference is what you are doing.

So, if I were to write this as a filter He e of j omega this will be =1– e power - j omega, do a little bit of simplification e power - j omega/ 2. What you will get within the bracket is 2j sin omega/2 and we have a result which tells us that the spectrum of e hat; power spectrum e of j omega will be = the power spectrum of the input signal as S subscript ee e of j omega times modulus of He e of j omega magnitude squared.

So when I pass white noise through a filter, the resultant power spectrum comes out to be the magnitude square of the filter because the power spectrum of the, so this is = sigma e squared modulus 4 times sin squared omega by 2. Basically it is the magnitude square, so now if I want to find out the total power; the power spectrum. So if I wanted to find the total power at the output Pde, the general formula is 1/2 Pi integral - Pi 2 Pi.

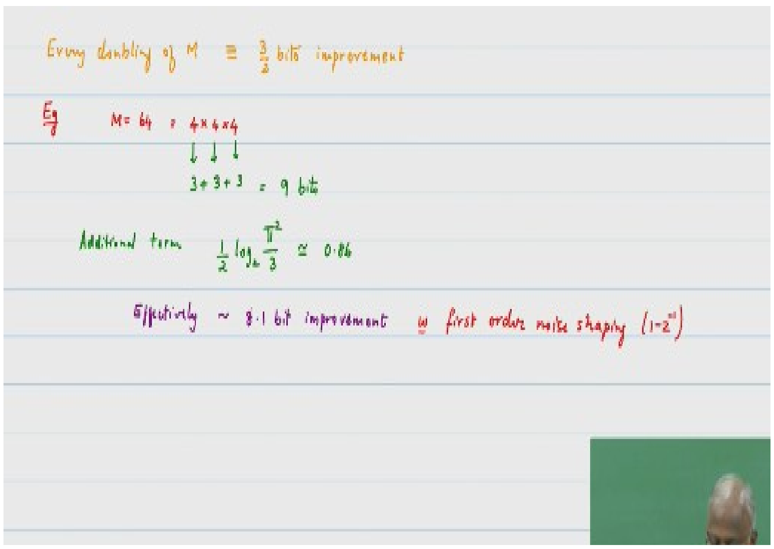
The power spectrum of S e hat e hat e of j omega d omega but Pde is not measured here it is measured at the output, at the output what did we do? We already restricted it to Pi/M so actually these limits are not correct this should be – Pi/ M to Pi/ M so please substitute for S ee do the integration and verify that what comes out at this point is delta squared by Pi sigma e squared = delta squared by 12.

So, that is where the delta squared is coming so delta squared Pi squared/36 M cubed, so please do verify this result just a very simple integral, M cubed so maybe it is helpful for us to write it as a delta squared/12, delta squared/12*Pi squared/3M cubed. Now delta squared, delta squared itself, delta itself =X of M subscript M / 2 raise to B and now if you write the log base 2 equation that we did in the beginning of this lecture, what you will get is 2 B = 2 times logarithm, logarithm based, log Base 2 of 12 with a - sign – there is a Pi squared/ 3 –, no there is a + log base 2 of Pi squared /3 just make sure I do not make a mistake in this equation, because yes there is a + log base 2 Pi squared / 3 and of course the other 2 parts - log base 2 of P de and a + 2 times log base 2 of Xm. I hope you can look at it and just verify there is a -3 log base 2 of M -3 log base 2 of M.

And just take the 1/2 to the other, this 2 to the other side so all of these things will get a factor of 1/2 so this becomes -1/2, this becomes 1 half, this becomes one half, this cancels, this becomes -3/ 2. The key result; what is the link between B and M? Previously it was - 1 half a log of log base 2 of M now it has become -3/2 so for every doubling of the input of the sampling rate you are not going to get a half bit improvement.

You are going to get 3/2 bit improvement. So that is a significant jump because that tells me that actually we can also start to see a fairly significant advantage and that is something that we can definitely leverage so the additional term, the benefit that we get is that it is going to be 3/2.

(Refer Slide Time: 31:05)



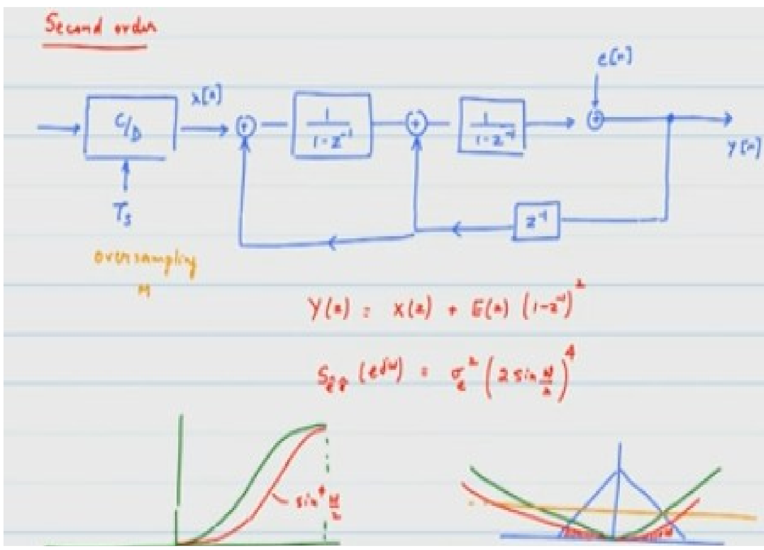
So every doubling of M, doubling of M will correspond to a improvement of 3/2 bits improvement, so it is as if you had 3/2 additional bits in terms of the quantisation so, example : if I have M = 64 which is 4 times 4 times 4 what I should expect is that first, of a factor of 4 gives me 3 bits + 3 bits + 3 bits previously they were all contributing 1 1 1 now they are all contributing 3 3 3 that should be = 9 bits of improvement correct?

That is what you would get, except that if you go back and look at the equation a little bit more carefully there is a term that was not there before and that term is actually, M is reducing the number of bits needed and this term was actually increasing the number of bits needed. So, basically go back and look at that 1 half log base 2 Pi squared by 3, that additional term do not overlook that additional term is 1 half of logarithm based 2 pi squared/3.

That comes out to approximately .86, so you gained 9 bits you lost .86 bits so effectively what you will get is a if you take it as .9, effectively it is a 8.1 bit improvement and you achieved this with a simple first order noise shaping of the type 1 – Z inverse. Very simple, of course then you say why not second order? Why not 3rd order? Why not 4 th order? Yes, that is what you should ask that is a right question to ask.

So, let us do everyone is okay with this? The improvement but there is a scale factor that kind of pulls down the gain a little bit but the net gain is still very substantial so you achieved a very significant, I mean you can go from 8 bits to 16 bits just by increasing the sampling rate.

(Refer Slide Time: 33:50)



So can we do 2nd order noise shaping? I will leave this as an exercise for you to verify but basically the structure is like this. You pass it through a C to D with oversampling, followed by the A to D conversion with the noise shaping so the noise shaping now is going to be in the following form, + sign 1 – z inverse that is the first integrator so if that gave you a first order transfer function, you expect that there should be a second integrator that will give you a second order transfer function, which is actually correct. There are two in series, with a + sign with the error entering into the picture and of course now we are to design the feedback taps correctly. The feedback taps are Z inverse so that looks exactly like a first order right? The latter part looks like a 1 st order quantizer embedded inside a 2nd order so basically you have a embedded inside another feedback mechanism so effectively.

So, if you take this to be Y of n and this to be X of n like before please go through the analysis and obtain the transfer function and please verify that what you get is Y of Z=X of Z again no change in the input signals component +E of Z*1–Z inverse whole squared so which then says that my modified spectrum S e hat e hat of e of j omega will be equal to sigma e squared that is the white noise.

Now what is inside the bracket is 2 times sin omega/2 previously it was a factor of 2, now it is raised to the power 4 so just to give an intuitive feel. In one case you saw sin squared omega/2 in the other case you are going to see sin power 4 so if this is sin squared omega/2 then this is going to be sin power 4 because it is less than 1 so it is going to be good rise much slower but eventually it will reach the same, so this is sin power 4 omega/2 all the way from 0 to Pi.

So, effectively what did you do to noise? You pushed it out further so in the region of interest it was further reduced so if you were to draw the spectrum if this is the input spectrum the 1^{st} order noise shaping push the noise out with a certain slope, you did not, you cannot remove the noise all you have done is you have sort of a distributed it in a non-uniform way, the uniform distribution would have been something like that.

What you did was you did it in a non uniform fashion and therefore it got pushed out, these are, the 2nd order noise shaping does it even further so effectively what you see within the signal portion of it is going to be advantageous for us, so basically what is the portion of the noise that is going to affect your now is only this portion which is substantially less than what you would have to deal with if you had either gone with the no noise shaping or with the 1st order noise shaping and this is where now why not 3rd order 4th order 5th order. it just you know get this thing as flat as possible. Control theory people will tell you unstable, which means you are the region in which this thing will, there are certain inputs for which you may become unstable so up to second order we can guarantee stability once you go beyond 2nd order there are restrictions in terms of stability.

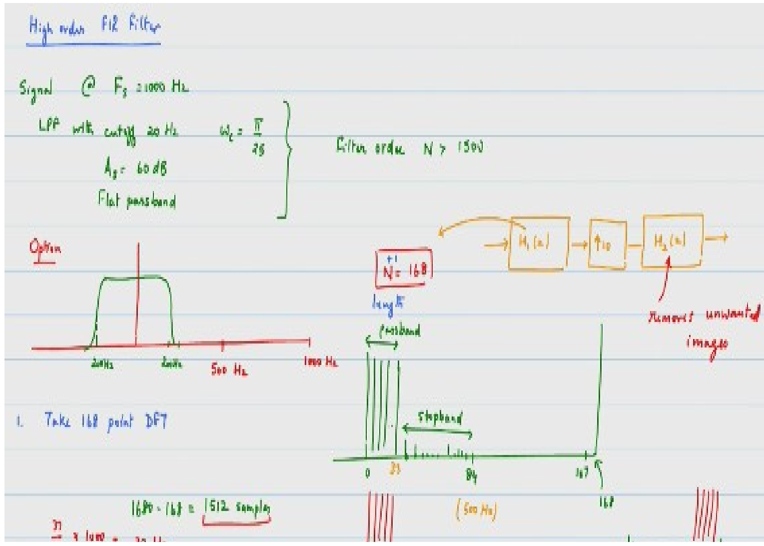
So you have to also keep in mind the issue of stability, so n >2 we need to keep the, you have to study the stability and you also have to restrict your input signal to a sharp, more a shorter range so again what is the, to study and ensure stability. So, we say that up to second order we can do the noise shaping with the guarantee of stability so then what people said was if I still want to get higher order noise shaping what will I do?

I do second order noise shaping and then on that I will do a second order of noise shaping so basically shaped noise and basically you can do so basically it is a cascaded second order noise shaping which you can do of course and you can achieve the benefit of that. So let me just sort of say that that is a start at least initiating point for you to start thinking about some very interesting applications of a multi rate signal processing.

Okay of course there are many more applications I had a thought of at least 2 more applications but given the time constraint maybe I will just restrict myself to 1. **“Professor – student conversation starts**” yes (()) (40:01) yeah, say that again, correct so yeah so the way you feed your, the point at which you feed, where you take the second order noise shaping is not at the output of the signal. So, you basically take the error and do a shaping.

So, basically you keep working on the error signal so the way you construct the higher order is not on the input signal itself but on the error so you take the difference take the error and then try to shape it even further okay good “**Professor - student conversation ends”.**

(Refer Slide Time: 40:53)



So, let us very quickly look at just one more example supposing I have to design a high order FIR filter, again these are the cases where you will encounter very often, so let me just give you a construct for you the scenario. The signal is a signal sample at thousand Hertz and you have been asked to design a low pass filter with a cut off of 20 hertz, pretty sharp cut off so omega C is equal to Pi/25.

And you are asked to design a filter with good stop band attenuation: 60 dB stop band attenuation and you want to have a flat pass band in the range 0 to 20 hertz so basically you want to have and you cannot just start dropping all the way from the omega = 0 hertz. So, you want to have a flat pass band as much as possible you want to have a good stop on attenuation and this is a tough filter design problem.

So, you give these to your math lab filter design problem it will say that the filter order typically will be N>1500 and it will say out of memory and it will return back saying I cannot design this filter; so here is the trick or multi rate signal processing to the rescue so you tell the program well do not worry about it. I designed something which is a little bit easier for us to do so this is 1000 hertz sampling frequency a half the sampling frequency is 500 hertz.

And we say that okay design a filter with a cut off around 200 hertz, with reasonable flat pass band and you do a cut off over here so this is around 200 hertz you design you write down the specifications and the algorithm does a calculation and let us say that it came up with an estimate that it is 168 and basically you are allowed a reasonable transition brand you are not made it you did not make it very tight.

So it can come up with a filter order equal to 168, but this is not the filter you want what you want is a filter that will cut off at 20 hertz. You remember the IFIR technique, what did we do? We will up sample by a factor of 10 which means that it will compress the spectrum from 200 hertz to 20 hertz your pass band will become 10 times sharper, the transition band, so effectively the strategy that we had was that you will design this filter H1 of Z which is corresponding to this 168 tap and then you would follow it up with an upsampling by a factor of 10 and then you would have another filter which will remove the images when you sampled by a factor of 10. This is the one that removes the images, it removes the unwanted images now this is the underlying technique that we need to do, but here is a very clever way to do it.

And I will leave you to sort of go through and make sure you are so first step you want you have got your filter out that is 168 lengths the filter length is 168, N is order and N plus one equal to 168. The length is 168 so this is length. Take a 168 point DFT. What will that give you? It will give you the frequency response of the filter with values from 0 to 167. The last one will be 167 and 168 of course will map back to 0, the DFT coefficients.

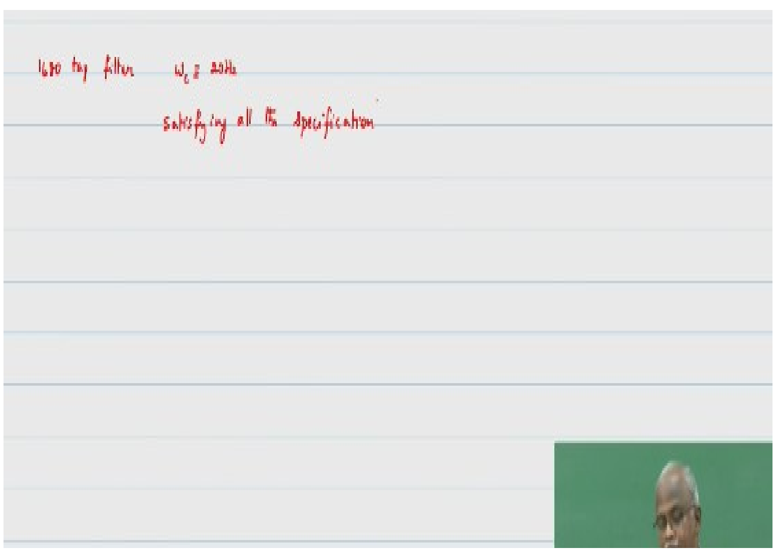
Now this is the sampling frequency that is 1000 hertz now at what frequency you will get the 200 hertz cut off, so 500 hertz basically corresponds to 84 and 168 by 2, that is 500 hertz I write the frequencies in orange 500 hertz, so 200 hertz will be approximately, you divide that by two point five, somewhere around 33 so your filter response we have these sorts of things up to 33, around 33 and then you expect to see something some small values.

You get to see some small values that is your stop band this is what you would see 168 points you took out 168 point FFT. These are alll, the pass band, so this corresponds to pass band and the rest is the stop band, now what I want you to do is pay close attention to what we are doing. so basically take these samples from 0 to 84, 0 to 80 take these samples. Take those 0 to 83 that is the 84 samples 0 to 83 and the last 84 to 167.

But leave a gap in between and the gap that is in between is exactly 1680 -168, that is =1512 samples, these 1512 samples are all the, I am going to set them to be equal to 0, visually okay? What did I do? I took a 200 hertz cut off filter with a sampling rate of 100 hertz I designed a computed the filter of 168 taps took the DFT. The DFT will have sort of mirror image symmetry and I have separated it out and inserted some number of zeros.

And exactly so that I will now get, this will become sample number 1680 this will become sample number 0 so effectively what we see is these samples and then towards the end you will see again these samples now, 33/1680 into 100 hertz is = what? 20 Hertz right? Exactly, one 10th of what it was before because you now expanded it by a factor of 10 from 168 we went to 1680 so which means that 33: if it corresponded to 200, the same 33 will now correspond to a 20 hertz so what you do is take the inverse DFT of this, but now it is not 168 point inverse DFT it is 1680 point inverse DFT

(Refer Slide Time: 49:29)



And what you will get is 1680 tap filter with omega c corresponding to 20 hertz and satisfying all of the requirements, all the specifications the stop band attenuation you already satisfied right you when you designed the 168 tap you already specified. So effectively what is the trick involved, previously what did we do? You up sampled in the time domain you inserted 9 zeros and then you passed it through a filter in the frequency domain what was it going to do?

It is going to have these 10 copies of the signal and this low pass filter would remove this unwanted symbols. What does, what did you do here? You did not even allow those copies to come in, it is the same as saying the copies came in when I inserted the 0s but I interpolated with the ideal low pass filter, there is no trace of any small bump or anything. Everything is 0 perfectly is 0 so which means that this is effectively like interpolating with an ideal filter.

If I did an ideal filter that is what I will get, it did not cost you anything basically it was one inverse DFT and a design and you incorporated 2 techniques that we have learned, one is the interpolated FIR technique the 2nd one is design using the FFT to your advantage so that you do not have to actually do the physical interpolation. It sorts of comes out as an inherent advantage into the system.

So, this is a starting point for us to start saying you know the whole tool kit of multi rate signal processing has got a very rich ways of helping us in all of our designs. So, you always try to see up sampling will compress the spectrum I can exploit, the down sampling we will go this way, so you kind of always keep looking at ways in which you can exploit the techniques that you have learned to your advantage.

So, in a nutshell what the course is all about is to say that the signal processing is built on a foundation where you have a fixed sampling rate, to that tool box you add the flexibility to change the sampling rate input-output and, what are the implications when you down sample? What are the implications when you up sample? When you insert 0s? what is; So, once you have this additional tool box then you start to see that okay.

In Communications, and for this application I need a filter here is an advantage, I can take advantage of it. If I did not have this notion of multi rate signal processing I would at this point give up and say well 1500 tap filter I cannot design right and or you would have to go through some extremely complicated process to design it, but whereas this one says hey I can design it through a 2 step process which was sort of almost trivial.

And will give you the advantage that we are looking for. Thank you very much. That is the end of the course and we really thank you for the way in which you have participated and wish you all the very best.