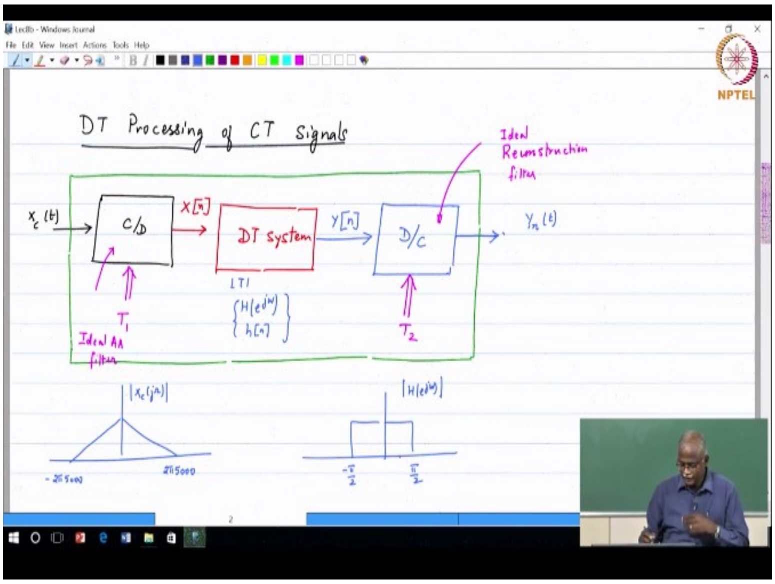
Multirate Digital Signal Processing
Prof. David Koilpillai
Department of Electrical Engineering
Indian Institute of Technology – Madras

Lecture – 08 (Part-1)
Upsampler and Downsampler – Continued - Part 1

Good morning. We begin lecture 8, we will do an example first. This is to sort of refresh our understanding, it is probably an example that is similar to what you have done in the assignment sheet but I thought it will be interesting for us to discuss it especially since we have spent a fair amount of time discussing sampling and reconstruction and what we are going to take away from this example is our understanding that when we change the sampling rate.

There are some things that we have to be very careful about and that is what I hope today's example will help us.

(Refer Slide Time: 00:50)



Now this is a block diagram that you are familiar with. This is when we do discrete-time processing of continuous-time signals. One very key element that is in this figure which enables us to get the linear time-invariant system for the continuous-time system is that you have the following. You have Ts and you have Ts okay, the same sampling period that you have used at the input is what produces at the output.

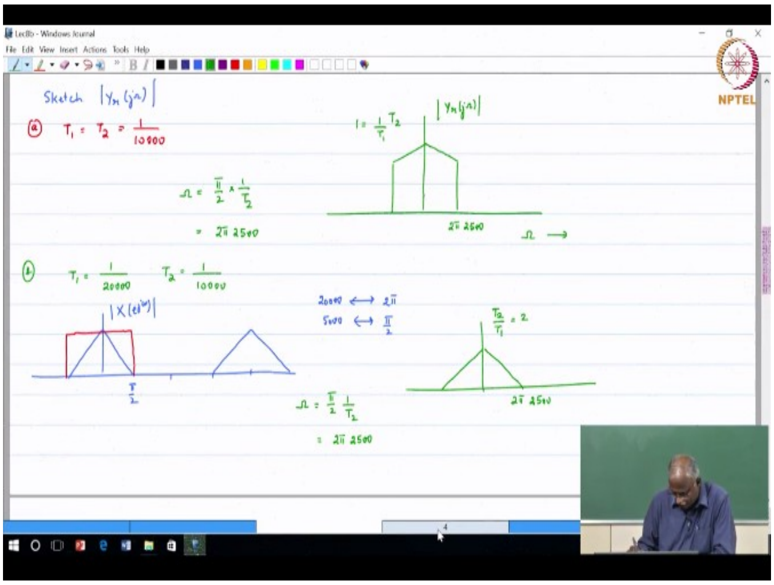
Because if you have different sampling periods then you will lose the property of linearity okay. So we are deliberately going to violate that and see what is the impact and that is the purpose of this first lecture example. So very quickly I am going to use a sampling period T1 and a sampling period T2. Again, they can be equal, they may not be equal. So this is the and we will assume the following.

That in the C to D converter, there is an ideal anti-aliasing filter, so that means it will not allow aliasing provided I have satisfied Nyquist criterion and this block the D to C block, I have an ideal reconstruction filter, those are assumptions that we will make and we will see why those were important and what sort of insight that it gives. So I do not want to bring in any effects of filtering, practical filtering.

Assume that ideal reconstruction filter, the both ends we have ideal filters that are present that means they are brick-wall filters okay. So here is the problem statement. I have a signal which is band limited, it is band limited to 2pi times 5,000 hertz okay, 10,000 pi radians per second. So this is -2pi times 5,000. It is a band limited signal so and the discrete-time system is a low-pass filter with –pi/2 to pi/2 cutoff okay.

And basically this is Xc of j omega magnitude response, H magnitude H e of j omega, this is the magnitude response of the discrete-time filter okay. So keep this picture in mind.

(Refer Slide Time: 03:10)



I have to shift over to the next screen and continue the example. So the task is to sketch, sketch Yr of j omega magnitude under the different cases that we will have to work with. The first one is probably a very straightforward one where we say that T1=T2=1 by 10,000 okay. I have up to 5,000 hertz; I am doing a 10 kilohertz sampling reconstruction at 10 kilohertz, so therefore there is an ideal anti-aliasing filter.

So no aliasing and the reconstruction filter does the perfect job for me. So just draw the sketch of the output spectrum, let me draw the output spectrum and just make sure that you are comfortable with that. If there is a doubt, we will answer it. Otherwise, we will move on from there. So the output spectrum because it passed through a low-pass filter will have the following shape.

It will basically some portion of its spectrum would have been removed by the low-pass filter okay and the cutoff frequencies would be it would have been pi by 2 in the discrete time domain. So pi by 2 in the discrete time domain and the continuous time frequency would be this divided by Ts 1 by T2 and this would be 2pi times 2,500. So some portion of the spectrum got removed. This in the continuous time this is 2pi times 2500.

This is omega axis, this is Yr of j omega okay and any scaling of the spectrum. I would have gotten a 1 over T1 when I did the sampling, I got a T2 when I got the reconstruction, T1 equal to T2, so therefore this is equal to 1 okay. No scaling of the spectrum okay. The first interesting case comes now, when I have T1 equal to 1 by 20,000 and T2 equal to 1 by 10,000. It is a multirate system. Basically, I have deliberately introduced a sampling rate change.

Just want to get a feel for what is happening in terms of the continuous-time signal. Again, this is more of an example just to explore and develop the insights okay. So first step is what is the discrete-time spectrum? Discrete-time spectrum says I have signal up to 5000 hertz, I have sampled it at 20,000 hertz. Basically, the spectral content is from -5000 to 5000. I have sampled it at 20,000 kilohertz.

So 20,000 kilohertz approximately is 2pi, so if I were to look at 5000 that becomes 2pi by 4, so basically this would map to pi by 2. So this corresponds to pi by 2, this is pi, 3pi by 2 and 2pi and this is the next copy of the spectrum. So this is the discrete-time spectrum of X of e of j omega okay. Now this went through a discrete-time system with the low-pass filter which was a low-pass filter which was going from –pi by 2 to pi by 2.

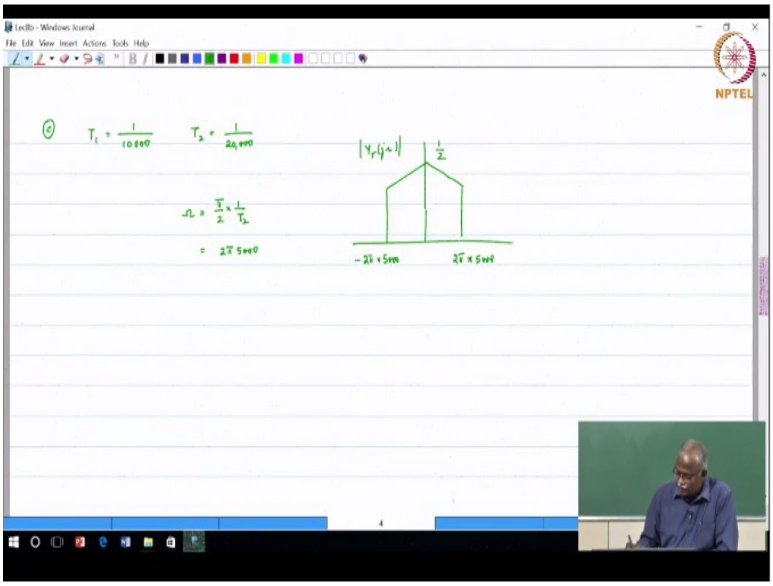
Notice that it did not do anything to the spectrum, basically the spectrum went through and then it passed on to the reconstruction filter, the reconstruction filter as before does the reconstruction process removing the spectrum, so this time around the entire signal seems to have appeared on the output. So the discrete time frequency is pi by 2 and I have to convert it into a continuous time frequency, so into 1 over T2.

So when I do this, this becomes 2pi, T2 is 10,000, so it becomes 2pi times 2500 okay. So this becomes 2pi times 2500. Now what happened? I gave a signal which was up to 5000 hertz, it got digitized, sampled. It went through a low-pass filter and then came out and I reconstructed it and looks like everything was there as before except that there has been a scaling or a warping of my frequency axis okay.

Now this is a very important element, if we do not have the same sampling rate at both ends, there will be some kind of a warping of the spectrum either you can have a stretching or you can have a compression and in this case it turns out to be that the frequencies looks like it got compressed and this is typically what happens if you sample at one rate and playback at another rate there will be some warping of the spectrum.

Now is there any scaling? So there is a 1 over T1 scaling at the sampling point and then there is a T2 by T1 part and T2 is 2 times because T2 is 1 by 10,000. So this would be equal to a factor of 2. So I got a scale factor of 2 and then a spectrum but and the axis on the spectrum is not what I sent in at the input but this is a result of that warping that has happened okay. Just one more variant of this and we will conclude this.

(Refer Slide Time: 09:10)



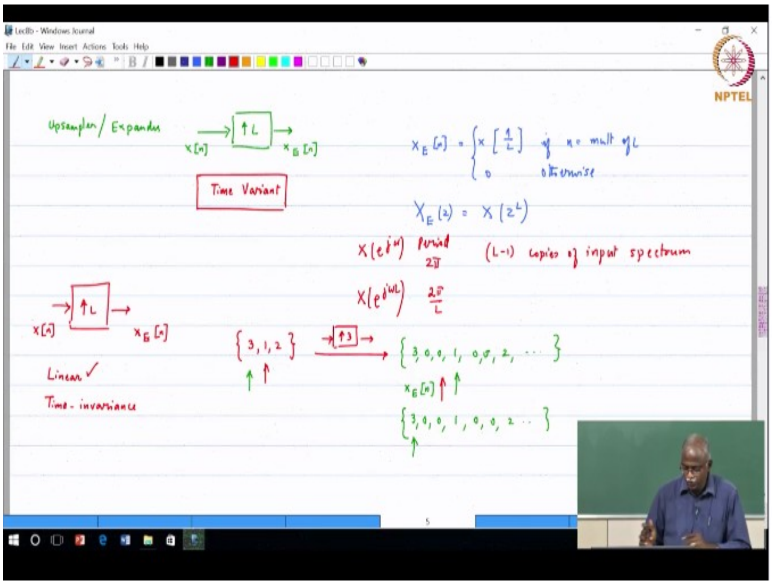
The part C, this one the input sampling rate is Nyquist 10,000, output sampling period is 1 by 20,000 so the reverse scenario and of course if I sample it at Nyquist rate, the low-pass filter will remove some portion of the spectrum. So the output spectrum is going to look as follows. It is going to have this shape okay and in the discrete time this cutoff point was pi by 2, the continuous time frequency was omega into 1 over T2.

So this will come out to be 2pi times 5,000 okay. So this spectrum I do not need to do a dashed line, I will just use a solid line is 2pi times 5,000. This is -2 pi times 5,000 okay. So again another sort of interesting outcome from this example is that the filtering did happen. The low-pass filter actually removed some portion of the spectrum. I pass it to the reconstruction.

But because my reconstruction assumes a different sampling rate, my frequency axis is now actually distorted and the interpretation comes out saying that you have frequencies all the way from -5000 hertz to +5000 hertz which is not correct because it is the interpretation is correct; however, that was not the intend because when I passed the signal through I wanted to remove some portion of the spectrum.

So again keep in mind that the sampling rate plays an important role, it is something that always by the way this will have a scale factor of 1 by 2 so and this would be mod Y reconstructed of j omega okay and it is some very interesting things and these can actually be leveraged into our advantage whenever there is a requirement and so the flexibility of the sampling period is something that we want to keep in mind always. So now I spend a minute or so on the review of yesterday's lecture.

(Refer Slide Time: 11:27)



So the primary contribution of yesterday's lecture was the upsampler or the sampling rate expander and the notation that we have is a box with a up arrow, upward pointing arrow and an integer L basically which means that I am doing an increase by an integer sampling factor L and if this is x of n, this is xE of n expanded version, we wrote down the time domain relationship xE of n is x of n by L if n is equal to a multiple of L, 0 otherwise okay that is the time domain relationship.

And we also derived that XE of Z is X of Z power L. We said that this basically introduces L-1 copies of the spectrum, L-1 copies of the input spectrum. There is also a scaling of the frequency axis because XE of j omega the period for this one is 2pi and for XE power j omega L the period is 2pi by L. So basically in the range 0 to 2pi you will have L copies of the spectrum, one original copy and L-1 additional copies okay.

So this is the general case that the new periodicity and from this we can derive. Now the interesting observations about this are twofold. The first one is whether the system satisfies linearity and time invariance. So the question is I have the upsampler block, upsampling by a factor of L and this is x of n, this is xE of n. Now the question is if so whether it satisfies linearity.

If I scale the input, output would get scaled, so there is no issue about that. If I add 2 signals, the output will be the sum of those 2 signals with the appropriate zeros inserted. So linearity is very straightforward for us to verify okay. Now the time invariance property probably easy to see but also need to be careful to show it in a more systematic fashion.

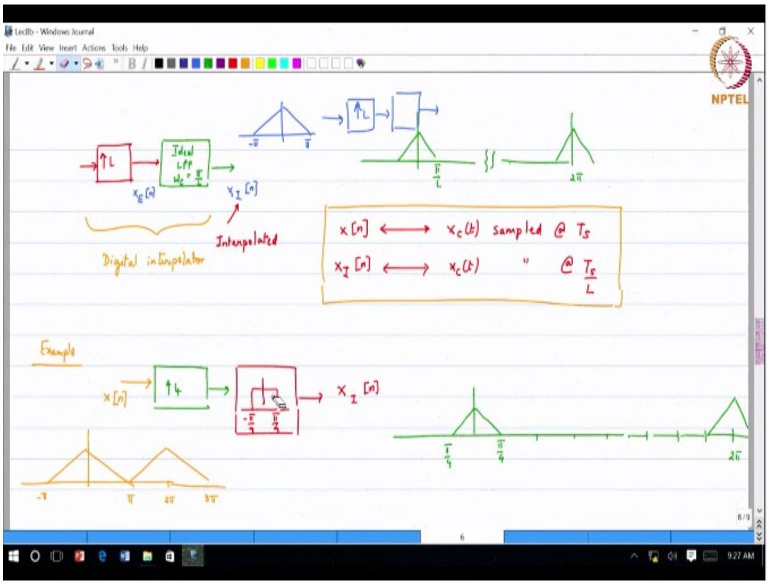
So if I have sequence 3, 1, 2 with this being my origin and I pass it through an upsampler by a factor of 3 okay. If I pass it through an upsampler, so the new sequence becomes 3, 0, 0, 1, 0, 0, 2, dot dot dot, my origin remains the same, I does not change but I have inserted the zeros, so this is xE of n, am I right? I have inserted 2 zeros between every set of samples that I have at the input.

Now if I want to delay my input by one unit of time, I apply this then the sequence that I get at the other side will be 3, 0, 0, 1, 0, 0, 2 dot dot dot with this being my origin right if I shifted my input by one unit of time okay. Now clearly that is not the same as shifting the output by one unit of time because if I shifted the output by one unit of time, it would point to that direction.

So shifting the input does not produce the same shift on the output side, again it is something that is probably obvious that you can but be careful I mean also just be comfortable to be able to show that yes linearity holds, the shift invariance does not hold and these are all, so basically this is a time varying system, so time variant system, probably the first entry point for you to start looking at time variant systems because up to now more or less the entire DSP was based on time invariance.

And of course we used that part a lot; LTI is very important for us but the fact that the time variance has now come in place and plays an important role okay.

(Refer Slide Time: 16:30)



So the second aspect that I want to highlight with in terms of the upsampler is the following. So I have the upsampler by a factor of L and yesterday we also talked about the following that if we put insert a discrete-time filter which is an ideal low-pass filter, ideal low-pass filter with omega c cutoff at pi over L. Then, what I get is a system where the spectra, this is limited to pi over L and then I have the repetition of the spectrum at 2pi okay.

So basically the copies of the spectrum assuming the input had if the input was of this form – pi to pi after the upsampling by a factor of L, upsampling by a factor of L followed by the filter okay. So the important element in this one is if we called this as xE of n, basically you have inserted a zero valued samples. I would like to call this as xI of n, xI of n, I stands for interpolated.

Because there are no more zero valued samples, all of these samples are non-zero interpolated signal okay. So if x of n your input signal corresponded to a continuous-time signal xc of t sampled at Ts then xI of n corresponds to xc of t sampled at Ts by L. Basically, you are sampling it at a faster rate but that is only with the low-pass filter that is present okay. So the combination of these two has got a name.

This is what we call as a digital interpolator. Notice that I did not go to the continuous time domain at all but I still talk about something called interpolation and the sampling rate being changed and other things. Where is the underlying framework? The underlying framework basically lies in the box that is on the right hand side which says that if I view my input signal as a continuous time signal band-limited which was sampled at Ts, the output of this combination of the upsampling block.

And the low-pass filtering looks like the same continuous time signal sampled at Ts by L. So again what I would like you to do is as an example just try out the following, make sure that you are getting all of the axes, the points on the spectra correctly. So I have an input signal x of n which has the following spectrum –pi to pi okay. So which means that you have to be you have to actually draw the periodicity property also shows okay.

So this point would be 2pi, this point would be 2pi then I would have 3pi and so on. So basically if I take x of n, pass it through an upsampler by a factor of 4 and then pass it through a low-pass filter with cutoff pi by 4, -pi by 4 to pi by 4 okay, ideal low-pass filter cutoff from –pi over 4 to pi over 4, this will produce an output which is xI of n and the spectrum should be limited to pi by 4 okay.

So basically whatever was the input spectrum, it got compressed by a factor of 4 or in other words with respect to the sampling frequency this is pi by 2, 3pi by 2, pi and here comes the copy of the spectrum okay. So in other words you have effectively created a scenario where it looks like a sampling rate as if you sampled the original signal at a 4 times the sampling rate, 4 times the Nyquist rate which will produce for you the following spectrum okay.

Please go through make sure that you are comfortable with this representation. If I went to the continuous-time, then I would have to worry about the gain but if my low-pass filter that was that if it had a gain of 1 that means it did not change anything right. So basically I inserted samples which means I replicated the spectrum then I pass it through a low-pass filter, so we need to be a little bit careful about the height of what we draw as the spectrum.

If you made this 4 which is what will give you a sinc function with the peak value equal to 1. In the time domain, if you want the sinc function to have a value of 1 then this would have a factor of 4 in which case you will end up scaling by a factor of 4 but if so just be careful about how the how the filter is specified. Again, it is a matter of a scale factor but it is very important that you brought it up.

Whether you want to normalize it to 1 in the frequency domain or you want to normalize it to 1 in the time domain depends on that, that will be reflected in the reconstructed in the output signal okay but just you make sure you verify that the height of the low-pass filter, the scale factor will get reflected in the impulse response that you will see in the sinc function and the factor of 4 in this case will actually give you a sinc function which has value equal to 1.