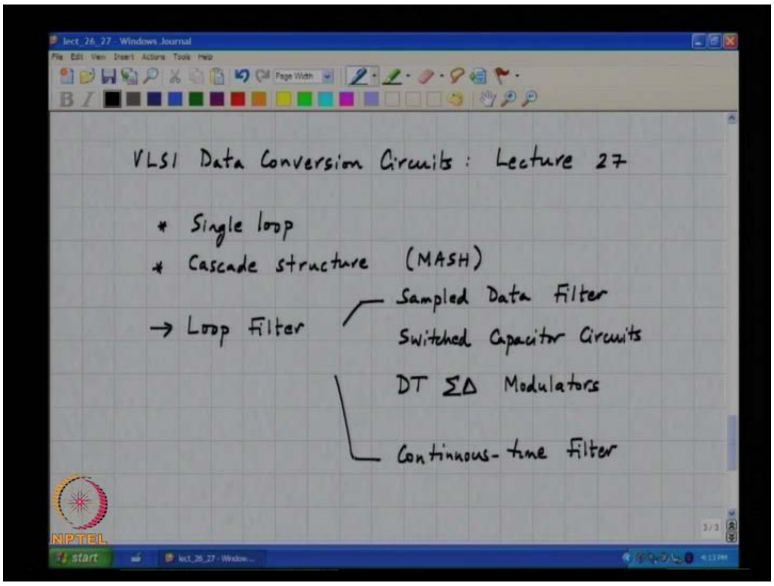
VLSI Data Conversion Circuits
Prof. Shanthi Pavan
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture - 27
Continuous-Time Delta Sigma Modulation

VLSI data conversion circuits – lecture twenty seven.

(Refer Slide Time: 00:14)



So far we have seen several ways of realizing a noise transfer function: one is the so-called single loop structure; the other, which we saw in the last lecture, is the cascade structure or the multistage noise shaping architecture called the MASH. The cascade structure as we saw depended on matching, because we use a second loop to digitize the error introduced by the first loop and subtract it from the output of the first loop. And, this is done in the digital domain. So, one needs the transfer function of the digital cancelation filter to match the noise transfer function of the first loop, which through its inherent nature is analog. Therefore, you end up with artifacts caused by mismatch; and, you may not get the order of noise shaping that you originally hope to get, because of the leakage of the noise of the first loop into the output. Is this clear? So, the next thing is to see what one has to do to implement the various building blocks of a modulator loop. So far we have seen the various so-called system level aspects of the entire loop. We know how to derive the loop transfer function; we know what the quantizer does; and, that is A to D and D to A; and, we know that, the errors in the A to D thresholds do not really matter; whereas, the errors in the D to A thresholds can be very problematic; and, so on and so forth.

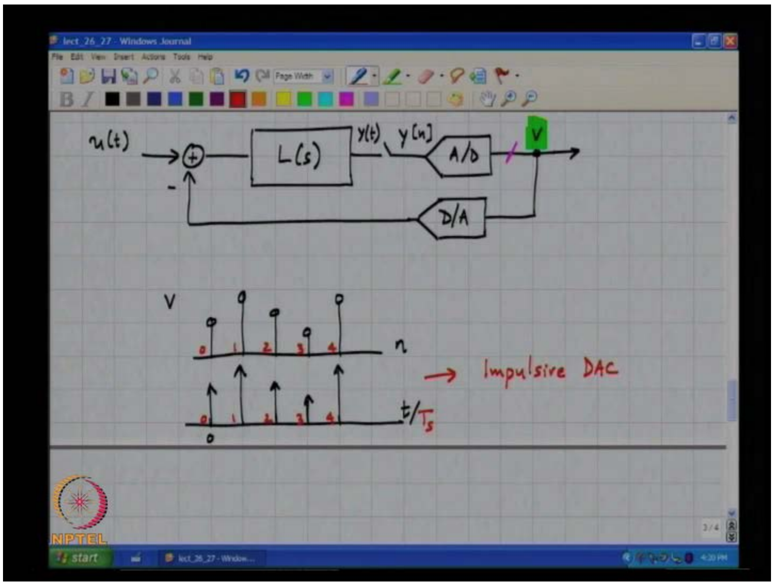
Now, the time has come to start understanding how to realize each of these?

Student: Blocks

Building blocks in circuit form, so that we can get a or make a working delta sigma modulator. So, the first thing we look at is the loop filter. The loop filter as you know takes the output of the quantizer as one of its inputs; the input would be digitized as another of it inputs and does some processing and gives you an output sequence. Now, if the input has been already sampled; correct? Then, the filter is working off of two sequences: one is v and the other one is u; the levels are still continuous. So, this is a discrete-time analog signal. And therefore, the filter that processes such signals is what is called a sampled data filter. And, they are often implemented using what are called switched capacitor circuits. And, since the signals inside are all discrete time and analog in nature, these are what are called discrete-time sigma delta modulators.

So, I suppose most of you are not familiar with the switch capacitor circuits. So, we will presumably come back to this mode of realizing the loop filter. Later on in the course, since you are doing analog IC design parallel, perhaps towards the end of the semester, we can get back to this mode of realizing the loop filter. Another way of realizing the loop filter is to do this in continuous time. So, in other words, the loop filter can be realized as a continuous time structure. And, the output of interest is not the entire waveform of the loop filter output, but only its samples.

(Refer Slide Time: 05:45)



In other words, the modulator structure looks like this. Do you understand? As far as the loop is concerned, you can fool the loop into thinking that, what is there. In fact, here is a discrete-time filter, because you are anyway sampling the output of L of s. So, as long as the samples match, what you would have gotten if there was a true discrete-time filter L of z; the loop would not know so to speak.

Student: Now, that u t is not sampled one.

Now, u t is not a sampled waveform; u of t is?

Student: Continuous

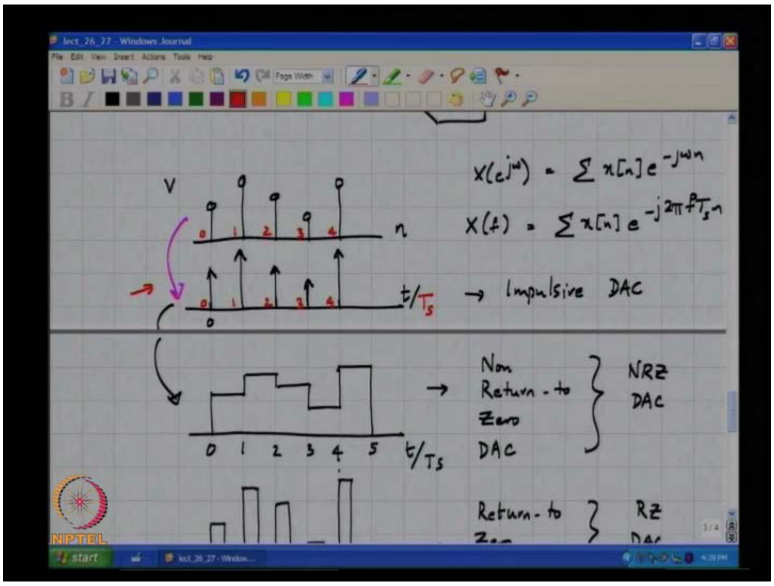
Continuous time. There are several advantages to doing this as we will find out going forward. And, this way of realizing the loop filter results in what is called continuous-time delta sigma modulation. So, what this L of s is trying to do therefore, is to mimic the operation of a?

Student: ((Refer Slide Time: 08:16))

Discrete-time transfer function, where the sampled output of L of s must be the same as that would be obtained when the sequence v was pushed into a discrete time filter. We will get into the details a little later on. But, at least intuitively you should see that, this should be possible. How you do it is another question. But, hopefully by fooling around with L of s, you can make it look like some desired L of z.

Now, let us refresh ourselves. Please note that, v is a digital sequence. And, to convert it into an analog waveform, you need a D to A converter. And, if you go back to the very beginning of this course, a discrete-time sequence is related. The sequence v is a discrete-time sequence with discrete levels also mind you, because it is the output of an A to D converter. The sequence v here is that digital sequence with finite resister length. So, these levels will be whatever – quantized. Now, D to A conversion is the process at least conceptually of converting this into a sequence, which is continuous time. But, instead of discrete time impulses, you have continuous-time impulses. This was n; this must be t. And, what is the periodicity of these samples? The sampling rate of the D to A converter. So, this is… In fact, instead of calling this, if I call this 0, 1, 2, 3, 4 and so on; I can still call this 0, 1, 2, 3, 4 and replace t with t by T s. So, in principle, this is what a D to A converter will do. It will take a discrete-time sequence and convert it into a continuous-time waveform. This is an analog waveform; it is so turns out that, its levels are quantized, because the input is that way. There are a couple of… Such a DAC is what is called an impulsive DAC, because the output is a sequence of dirac delta functions. And, if you have impulses, you know that, it is most likely not practical to have such a DAC waveform. In practice, what one will do is to have a pulse shape associated with a DAC.

(Refer Slide Time: 12:22)



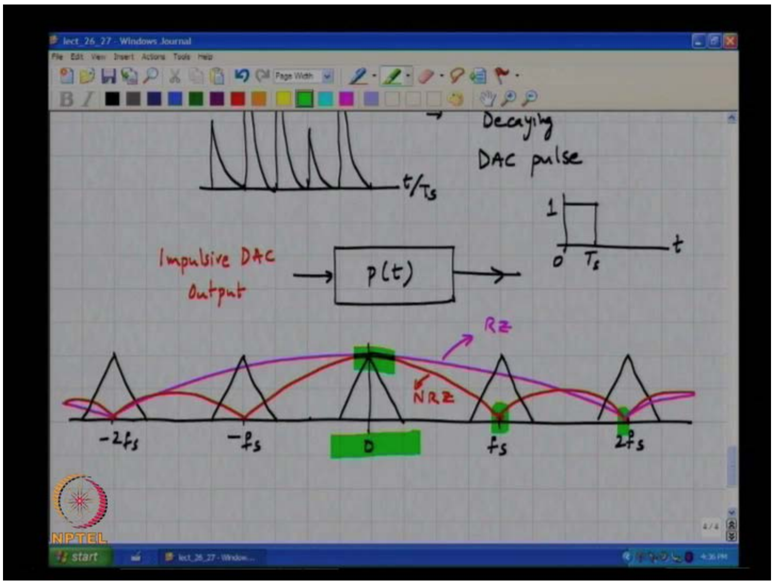
In other words, what you do is not send out impulses proportional to the input sequence, but some multiple of a known pulse shape. So, for example… So, this is what is called a non-return-to zero DAC. Everybody understands that, a DAC takes an input sequence and converts it into an analog output waveform. So, the only point of contention is what the pulse shape of the DAC is. So, it can either be impulsive or it can have the favorite pulse shape. Commonly used pulse shapes are the following: one is what is called the non-return-to zero DAC, where instead of putting an impulse, which we know is very difficult to generate. We have a rectangular pulse, where the height of the pulse stays constant throughout the sampling period. So, such a DAC is called – in short, it is called an NRZ DAC. NRZ is standing for non-return-to zero, because the waveform does not return to 0.

Now, given that, there is a name called a non-return-to zero DAC. It must also follow that.

Student: ((Refer Slide Time: 14:38))

There must be something called a return-to zero DAC, where the waveform actually returns to zero in the middle of the clock cycle. For example… This is what is called the return-to zero DAC or in short, the RZ DAC. And, as I said, you can choose the favorite pulse shape; NRZ and RZ DACS are commonly used pulse shapes presumably because they are easy to generate.

(Refer Slide Time: 16:09)



Now, another pulse shape that one can think of is the so-called exponentially decaying pulse shape. So, you have something like this. So, one obvious question is what is the difference between all these waveforms? Or, why would one want to do one versus the other? First of all, can you comment on how one can relate the spectrum of the discrete-time sequence to that of this impulsive DAC output. Clearly, the output continuous-time signal is representation of the discrete-time sequence. So, their spectrum must be related. Given the discrete-time spectrum, we should be able to determine the continuous-time output spectrum. So, can somebody tell me what that relationship will be?

Student: ((Refer Slide Time: 18:10))

Yes.

Student: ((Refer Slide Time: 18:20))

So, it is very straightforward. X of e to the j omega is nothing but sigma x of n e to the minus j omega n. And, X of f is nothing but sigma x of n e to the minus j 2 pi f T s times n; correct? So, as you can see, it is simply a matter of taking the spectrum of the discrete-time sequence and scaling the frequency access. Instead of omega… Omega equal to 2 pi specifically becomes f equals 1 by T s or f s. So, the continuous-time fourier transform of this impulsive sequence will look like will have images repeated at multiples of the sampling rate. Is that clear? Now, how can you relate the spectrum of this to the spectrum of energy DAC output?

Student: The precession of ((Refer Slide Time: 19:56)) Instead of taking z, it will be integral of that or another way is through say some sort of ((Refer Slide Time: 20:21)) From which we have to pass one impulse; we will get the…

Correct. Since it should… Please note that, this… Once you know this spectrum… Or, in other words, once you know the spectrum of the impulsive DAC, finding the spectral properties of the NRZ DAC or the RZ DAC or the exponentially decaying DAC are all very straightforward. They are equivalent to taking the impulsive DAC waveform; in principle, taking the impulsive DAC output and passing it through a filter, whose impulse response is the DAC pulse shape and we get the… So, for example, the impulsive DAC output spectrum let us say is something like this. At multiples of f s, we must have the spectrum repeated with the same strength. That makes sense because the impulse has got… The impulsive pulse has a fourier transform, which is flat. Correct?

Now, after passing this through an NRZ pulse shape, what is p of t for an NRZ pulse? It is 1 for 0 less than t less than T s and 0 elsewhere. Correct? So, what do you think is the fourier transform of the

Student: ((Refer Slide Time: 22:57))

It is T s sink f T s. So, the sink will have null set.

Student: ((Refer Slide Time: 23:08))

Multiples of 1 by T s or f s. So, if you want to find the spectrum of the DAC, whose pulse shape is an NRZ pulse; then, you take this spectrum and multiply it by sink square and in the log… You will have basically nulls if you want to just find the fourier transform, take this and multiply it by sink for n. For a DAC, whose pulse shape is a return-to zero pulse, how will it look like?

Student: Spectrum will be of span ((Refer Slide Time: 24:18))

So, a return-to zero pulse shape. What is the p of t?

Student: ((Refer Slide Time: 24:25))

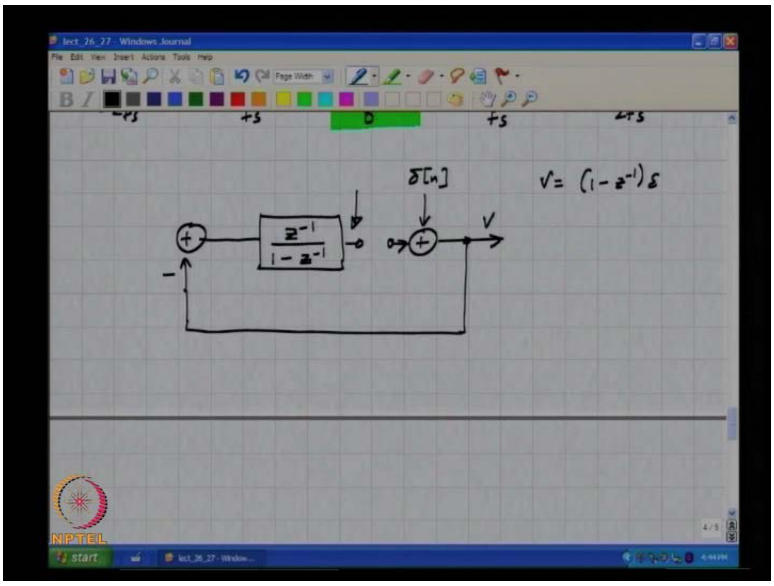
If you want it to have the same DC gain as the NRZ pulse, then you must have a height of two, but extends to only half the clock period. So, that way, you will have a pulse, whose DC gain is the same; and, whose frequency response will look like… So, this is the RZ spectrum. This is the NRZ spectrum. And, as you can see from this spectral picture, the RZ DAC spectrum has got lot more high frequency energy than the NRZ DAC spectrum. And, why does that make intuitive sense? Width is less. So… I mean one… See the amount of high frequency energy is proportional to the number of jumps you have in the… Or, the magnitude of the jumps you have in the waveform. Correct? So, you can see that, clearly, in every clock sample, there is a guaranteed two jumps of equal and opposite magnitude. So, even if the input to the DAC was DC, the output will be a bunch of rectangular pulses. So, there will be high frequency energy.

Whereas, if you had an NRZ DAC and the input was purely DC, the output would be free of DC, because there is a null at multiples of f s. If the input only consisted of DC, the nulls will take care of – will remove all of the high frequency images and you will see only DC spectrum. And, that makes sense because there will be no jumps in the time domain waveform at all if you use an NRZ DAC with a DC input. Is this clear? So, for an exponentially shaped DAC… I am not going to do the math here, but straightforward to find the fourier transform of the pulse shape and compute the spectrum. So, all that I want to say is that, the choice of pulse shape within the signal band at low frequencies; whether you choose NRZ or RZ, at low frequencies, it does not matter. That make sense because in the average, whether you use an NRZ DAC or an RZ DAC, the average will be the same, because one has a value of 1 for full clock period; the other one has got value 2 for half the clock period. So, on the average, it is the same thing. But, they have different high frequency properties; and, you can all relate them to the spectrum of the original discrete-time sequence that generated this waveform. Do you understand?

So, as far as we are concerned, when we talk about a DAC, we need to know the sampling rate and the pulse shape. And, as we discussed some time back, an ideal DAC will be such that, if the code here was 1 and the code here was say 2, the ratio of the two steps will be exactly 2. Correct? But, another non-ideality is the fact that, these steps are all not the ideal steps; there will be small differences because of mismatch. And, that is a non-ideality. But, even an ideal DAC is associated with the sampling rate and a pulse shape. The pulse shape can be either an NRZ or an RZ or an exponential decaying pulse shape or a sinusoidal shaped pulse. The choice of pulse shape is yours; but, you must have a pulse shape.

Now, we want to build a sigma delta loop, where we are trying to make the loop believe that, the filter is discrete time. Do you understand? So, the question is how do we do this? So, let us take an example of a first order loop as an illustration.

(Refer Slide Time: 30:19)

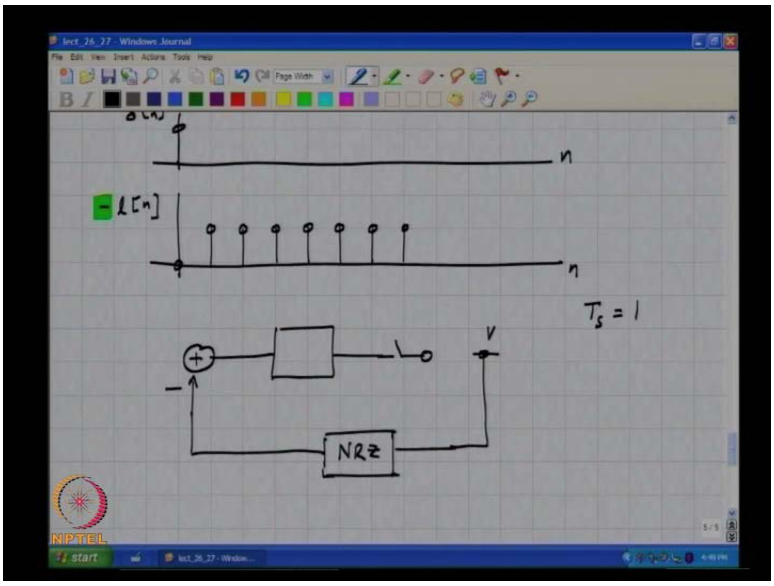


So, let us now consider a first order discrete time delta sigma loop and see how one can use a continuous-time filter to make the loop believe that, it is indeed dealing with a discrete-time loop filter. So, all I have done is… This is a familiar discrete-time loop. As far as the loop is concerned, we want to make sure that, the transfer function from E to V here is 1 minus?

Student: z inverse.

z inverse. So, V must be 1 minus z inverse times E. And, if the gain of the loop filter is high enough, the STF will be 1 at DC. Correct? So, for the time being, let us not worry about the STF; we want make sure… See the feedback loop is in this way. Correct? So, if you want to realize the same noise transfer function with continuous-time circuitry, we should focus on this whole path; that is, you break the loop here for example; you will get the same noise transfer function if you have the same loop gain. Correct? And, how will you measure loop gain? In this example, I will break the loop like this, for example; I will apply an impulse here and look at the output sequence, which comes up.

(Refer Slide Time: 32:49)

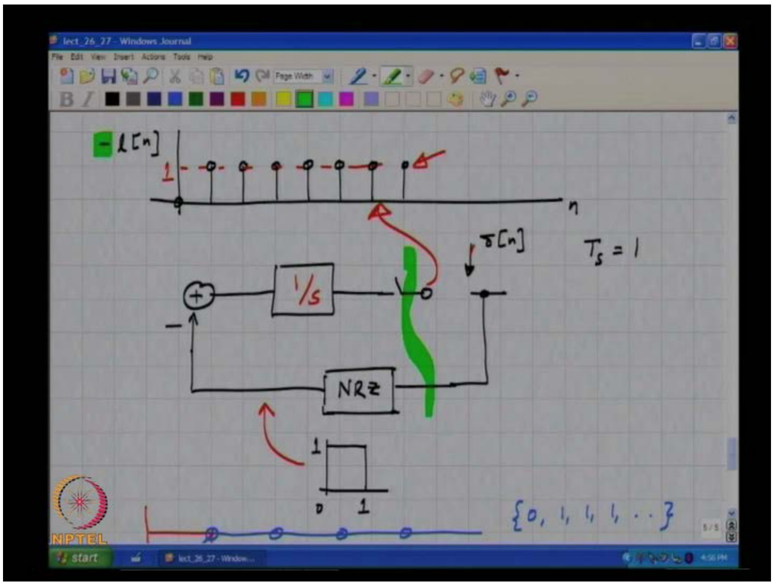


So, if I do that, what happens? The input sequence is an impulse; and, the output sequence… How does it look like? 0, 1, 1, 1 and so on. Does it make sense? Now, what we want to do is to do this entire thing in continuous time, where the output of the loop filter is sampled. And, let me call this minus l of n, which is the actual thing that comes up. The sequence that comes out here is minus u of n minus 1. So, I am going to avoid drawing negative samples; I will simply call this minus l of N. Now, the loop filter has a transfer function z inverse by 1 minus z inverse. So, if you want to do this in continuous time, what is going to happen?

Let us assume a DAC pulse shape. So, let us see what we need to put in the loop filter. There is clearly a sign inversion here; and, we have the pulse shape of the DAC to worry about; the output of the loop filter is sampled and fed back through a quantization error. Is this clear? Now, how would we break the loop? We break the loop… We finally, want the transfer function from E to V to be 1 minus z inverse. Please note that, these are all only… E is a sampled sequence; V is also a sampled sequence. So, we do not really care what happens in the middle; but, if we break the loop here and inject an impulse, we should see if this was claimed to be a… Somebody claimed this is a first order modulator. When I broke the loop here at E; apply an impulse – discrete-time impulse on one side; the sequence which comes back at the other end must be exactly u of n minus 1. Is this clear? Do you understand?

So, let us take an example of an NRZ pulse shaped for the DAC. We need to talk about some pulse shape instead of leaving it as p of t. Let us call this an NRZ DAC. And, to make matter simple, let us assume that the sampling rate is 1 hertz. We can always scale it to whatever we want later.

(Refer Slide Time: 37:39)



So, in other words, this pulse shape p of t will be 1 all the way from 0 to 1 second. What are we expecting if this loop filter is doing its job properly and mimicking a discrete time integrator properly? What sequence are we expecting here?

Student: ((Refer Slide Time: 38:05))

We are expecting the output samples to look like this. Is this clear? So, if I put an impulse here. So, here I must inject a discrete-time impulse. At the output of the DAC, what will that result in?

Student: Single rectangular pulse

It will result in a?

Student: Single rectangular pulse

It will result in a single rectangular pulse, which is… It will result in this pulse basically. Correct? And, the sampled output of the loop filter must look like this. Do you understand? So, what do you think should be there in the box? What is this? What is that level finally?

Student: 1

1. So, what do you think must be there in the box in order that you get a waveform, whose samples will be 1 when excited by a pulse as shown here?

Student: Integrator

It should be an?

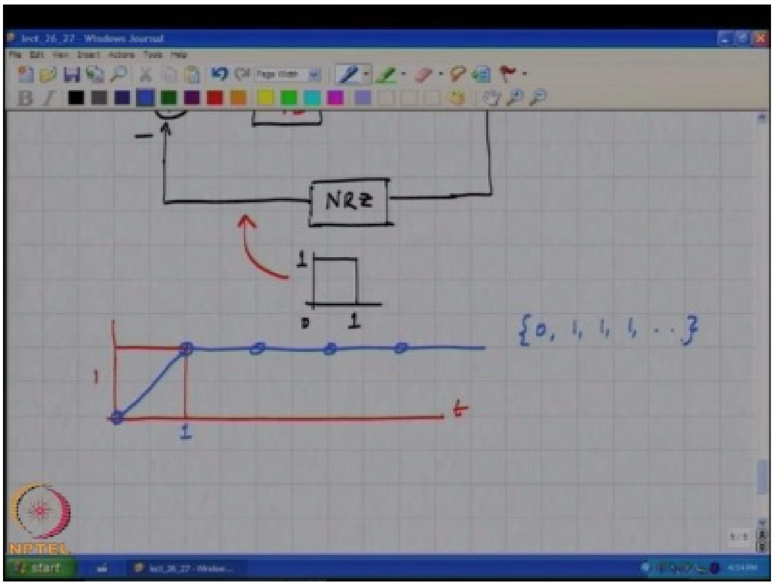
Student: ((Refer Slide Time: 40:27))

It should be an?

Student: Integrator

It could be an integrator. Do you understand? So, a continuous-time integrator for example, will give you…

(Refer Slide Time: 40:56)



If the input to the continuous-time integrator was like this, the output would be… How will the output look like? How will the output look like when this is integrated?

Student: It should be ramp ((Refer Slide Time: 41:27))

It will be a ramp. At the end of 1 second, the output will be 1; after which, it will be?

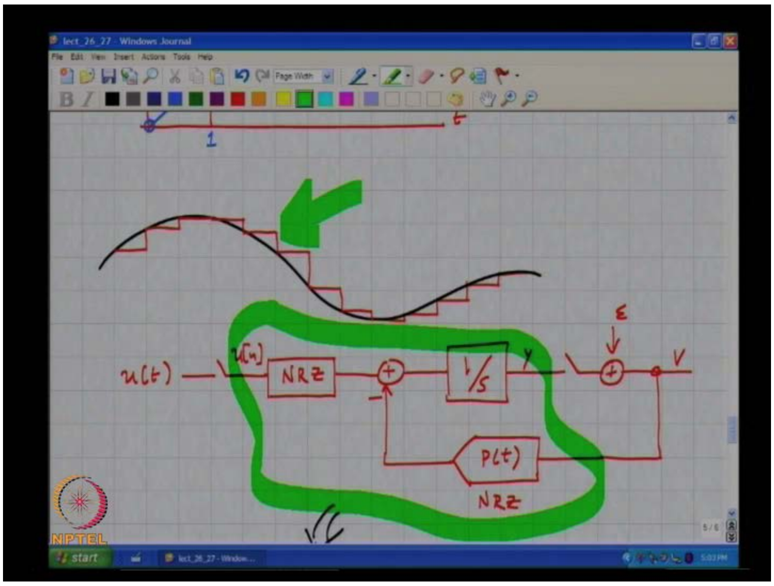
Student: Constant

Constant. And, if you sample this at the sampling rate, which is 1 hertz; how will it look like? The sampled output sequence looks like 0, 1, 1, 1 and so on. So, in other words, this part or rather this part here, where the input is at discrete-time sequence and the output is also a discrete-time sequence is completely indistinguishable from the situation, where we had a truly discrete-time filter, which directly processed the input sequence without having any continuous-time circuit; which means that, an integrator, which is being driven by an NRZ DAC is mimicking the role of a discrete-time integrator. Is this clear? So, since the input sequence delta of n and the output sequence coming out of the loop filter after it is gone through the loop are identical to the discrete-time case, it must follow that, the noise transfer function when I close a loop like this must be?

Student: 1 minus z inverse

Must be 1 minus z inverse, because as far as the loop is concerned, it does not know that, all this integration is being done in continuous time and you are sampling the output of the continuous-time filter, rather than taking the samples and actually integrating them in discrete time. Is this clear?

(Refer Slide Time: 44:16)

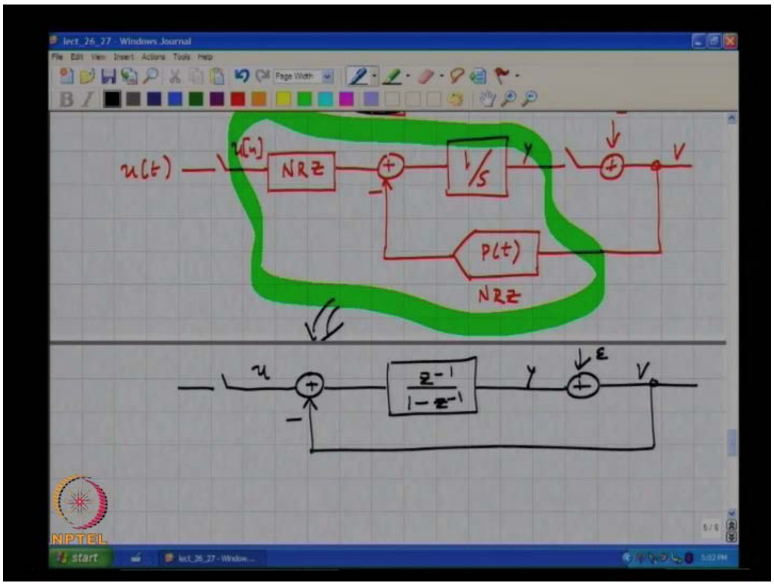


Now, the next thing is to worry about what happens to the input. Correct? So, let us see that. The input is some low frequency waveform like this. And, if the oversampling ratio is very large, this is a grossly exaggerated picture, where the oversampling ratio is not very large. You can see that, it is kind of one cycle is occupying perhaps maybe 10 or 12 units of time. So, if we have a highly oversampled signal, the periodicity of the signal will be many many many samples. So, what I want to point out is that, whether you have the signal like this or whether you say that it is some signal like this; really does not make much of a difference you understand. So, as far as the stepped signal approximation is concerned, you can think of it as taking u of t; sampling it with an impulse sampler; and then, passing it through a filter, whose pulse shape is also an NRZ pulse. This is an NRZ DAC pulse. This is 1 by s; this is the output. This is the quantization noise; and, this is V. Rather let me say… Is this clear? So, as long as the input signal is of lower frequency, you can think of this as being filtered by a… I mean impulse sampling it and having an NRZ pulse shape to the input also, so that that will then allow us to represent this whole thing as what?

Student: ((Refer Slide Time: 48:18))

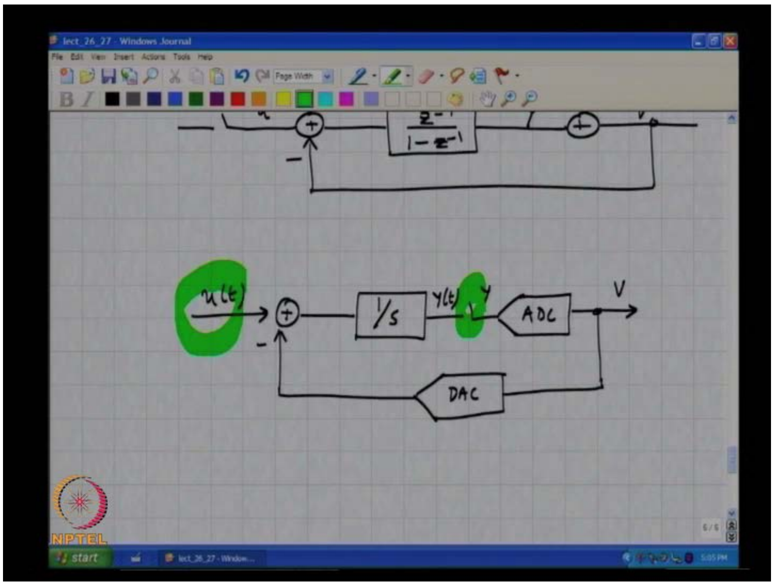
As a 2 input discrete-time integrator.

(Refer Slide Time: 48:25)



There is no sampler at all here. Is this clear? So, what we have essentially done is taken a continuous-time loop filter. In this case, finding what the loop filter must be turned out to be a trivial affair; we are able to guess. With more complicated loop filter structures, that may not be possible. So, we need to figure out a systematic way of finding the l of s; which when driven by a DAC with a known pulse shape p of t and its output is sampled at f s will result in the same sequence that one would have got when one used a discrete-time loop filter. Do you understand? I mean this kind of ad-hocing will not probably work for higher order more complicated noise transfer functions. Do you understand? But, before we go there, we will also kind of see what happens when u of t is not really a low frequency signal. The assumption so far has been that, the input to the sigma-delta loop is oversampled; which means that, some sampling has taken place before; which means that, before we sampled, we know we are going to put an anti-aliasing filter, which is going to cut off stuff at multiples of the sampling rate. But, here no such thing is happening. This idea is just a way of understanding the equivalence between a discrete-time modulator and a continuous-time one. But, the true modulator is only this. And, here is where we…

(Refer Slide Time: 51:22)



Let me draw that. The true modulator is only this – a DAC with some pulse shape. This is an ADC; this is V; this is Y; this is y of t; this is u of t. So, the sampling – please notice is happening inside the loop at the output of the loop filter. So, we have no intention of sampling before the signal enters the loop. So, one question is do I really need to put an anti-aliasing filter at all. So, in other words, if this input signal, which is anyway being sampled at the output of the loop filter inside consists of high frequency components along with the desired low frequency component; one question is clearly, the high frequency component cannot be represented as samples of u of t with the rectangular pulse shape. So, we have to wonder what will happen to this loop if the input consist not of a low frequency signal, but frequencies, which can potentially alias to DC, because we have not done any anti-aliasing here. So, we would be curious to figure out what happens if the input was say a tone at frequency f s, which in discrete-time delta sigma modulator would result in…

If you did not put an anti-aliasing filter, a tone at f s would alias to DC and show up in the signal band. If the same thing happens here, then we will say that is too bad; it is nice that we are able to realize the loop filter using continuous-time circuitry; but, we still need an anti-aliasing filter upfront to kill any signal components at multiples of the sampling rate. So, we will see what happens to this – how this loop behaves when a tone at multiples of the sampling rate is put. We will do that in the next class.