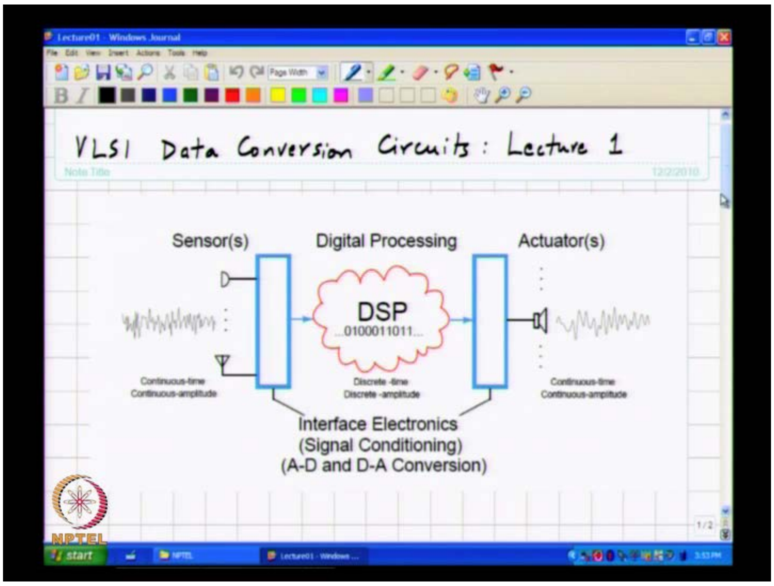
VLSI Data Conversion Circuits
Prof. Shanthi Pavan
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture - 1
Introduction to Data Conversion

(Refer Slide Time: 00:11)



Good evening, this is VLSI data conversion circuits the first lecture; in this lecture I will cover or rather I will give you an overview of the entire course and the significant topics that I wish to cover, as well as policies regarding assignments and submission and so on. The motivation for this course is the following. So, if you look at this diagram on the screen; any electronic system today can be roughly broken up into you know these 3 parts, we have the sensors which are signals that we want to sense. This could be voice signals which are sensed by microphone, this could be means RF signals that could be sensed by the antennas in your mobile phone.

These could be pressure sensors you know what have it, all right. Now, for various reasons mostly to do with programmability and as well as cost and the time it takes to make a system; it makes sense to process these signals in a digital fashion, right. So, even though the signals that naturally occur are analog in nature, which are coming from sensors of various kinds; we want to be able to store and process them digitally. So, this is what we eventually need to get to, and once we process this data right, whichever I mean whatever way we want amplification, filtering, decoding, whatever once we have processed this data, we need to be able to sense the output again, right.

So, if you are either watching a t v, where the output is optically in nature. You are listening to a song the output is audio in nature. So, we need to be able to convert these signals which are process digitally back into the analog domain. So, what we mean by analog and digital domains? One thing we must notice is that the input signals right, the sensors the signals that we want to sense are continuous both in time and in amplitude. On the other hand; the processing engine is digital now, what do we understand by a digital system.

A digital system as you know it most of these digital systems are synchronous state machines right, which means; what they understand data only at edges of a clock, right. You all done digital I C design, where you have learnt you know how to design the systems. But the bottom line is that the underlying mechanism is that signals only make sense in these systems at 1 edge of the clock, be it the rising edge or the falling edge or a combination of the 2, it does not matter. But the fundamental thing is that signals only make sense at discrete instance of times, in other words; time is discrete as far as the DSP is concerned.

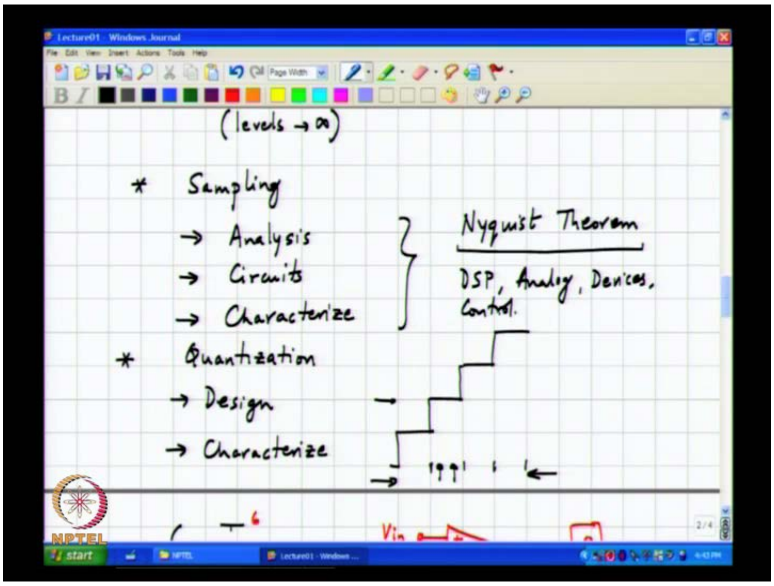
Further the register lens that you have in a digital system, are also finite, correct. So, you might have a 32 bit processor you may have a 64 bit processor. But it is still a finite number which basically means; that at a given instant of time, right. The value in the register can take any value from all zeroes to all ones, which means; that if you have n bit register you have 2 to the n possible levels represented by that n bit digital work. In other words; as far as the digital domain is concerned not, only is time quantized because of the nature of the clock and because you are only interested in values of the signal at edges of the clock, amplitude is also quantized. Because you have a register with a finite number of bits, which means; that if you have an n bit register, you can only represent 2 to the n levels correct.

So, that is what I have put down here, real world signals are continuous in both time and amplitude whereas; the processing is done in the digital domain, where neither time is continuous nor is amplitude. And, similarly; once we have done this processing we need to able to take whatever process data we have, and be able to convert it back into a signal which is both continuous in time and amplitude, correct. So, that basically means that you cannot directly hook the sensors into a digital state machine, they must be some interface electronics, which interfaces between on the input side the sensors and the digital state machine, correct. And on the output side between the digital state machine and the actuators, the actuators are systems which convert electrical inputs or digital inputs into physical quantities, right.

So, for example, loudspeakers and actuator which takes which takes an analog input an electrical input and converts it into acoustic output. And, similarly; an LED is something which takes an electrical input and converts it into and optical output and so on, right. So, in order to be able to interface with the real world we need electronics, whose job from a signal processing point of view is to take a signal which is both continuous in time and an amplitude and convert it into something which is discrete both in time and amplitude. And, this is to be done without loss of information or at least with a minimal loss information, you understand.

So, that is where the signal conditioning electronics comes in, and signal conditioning is a general term used to not just say not only do this continuous time, continuous amplitude to discrete time discrete amplitude conversion. It also in many cases it turns out that the signal that you get is very small in power, right. And that is not enough to be able to drive subsequent stages. So, it is not simply directly converting the small signal into something which is digital, it is often very difficult to do or sometimes impossible. So, you need to amplify the signals first perhaps do some filtering things like that before you actually end up converting it from an analog signal to a digital signal.

(Refer Slide Time: 07:46)



So, all this comes under what we call, what is commonly called? Interface electronics and. So, clearly if we need to convert from continuous time, continuous amplitude we need to eventually get to discrete time discrete amplitude. There are apparently a many number of routes you can take to do this correct. So, one commonly taken route is first you quantize time. So, what do you do is; you take a continuous time, continuous amplitude signal and convert it into something which has got a discrete time, but; where the amplitude is continuous, right. So, this operation is called sampling, right. Please feel free to stop me, if you have any questions. So, once you finish sampling; you have with you a sequence a discrete time sequence, where all the, I mean the amplitude is still continuous, right.

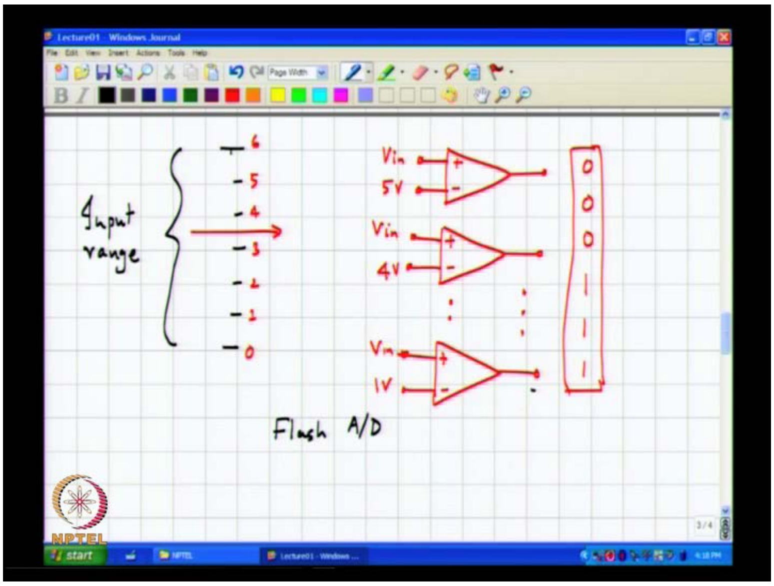
Then, you can take this continuous level signal which is already been sampled and then quantize it, right. Which is basically; discretizing the making the amplitude levels also discrete, just like you made time discrete. So, this is called quantization, right. So, in other words; here, the number of levels is infinity, number of levels of amplitude is infinite. The moment you quantize you have a finite number of levels, once you have finite number of levels it is possible to assign a unique digital word to each level. And, therefore; I mean if you have a larger number of levels it stands to reason that you need a larger length of the digital world which will quantize the I mean which will represent that the number of levels that you have ok.

So, in order to be able to do this with minimal information loss when you have to bear in mind a couple of things; so, since analog to digital conversion consists of both; quantization and time, as well as quantization and amplitude. The first I mean what we can do is to start looking at we definitely need to look at the sampling operation, right. So, let us try and understand what sampling is all about? You get some feel for spectrum before sampling spectrum after sampling and so on, ok. Then we will so, once we figure out what it is all about from a mathematical point of view then, we start designing circuits which do, which perform this operation, right.

And, one thing you must be aware of by now is that whatever you think about when you have a mathematical idealisation, when you build something practical it is very likely that you know what you think will happen almost often does not happen, right. So, there are going to be artefacts which are occurring, because of non idealities in the elements that you use to make the sample and hold. So, even though in on paper it is very easy to write x(t) and then, when you sample it is x of k times t or t s, right. When you build or when you design a real sample and hold you will find that there all sorts of artefacts. In other words; we not only must know how what the math means right, we also must be aware of what circuit non idealities are added in this whole process.

In other words; we must be able to understand and characterize a sample and hold, once we have done this. We will move onto quantization and if you think about it at some level quantization is basically a very simple thing, right. You have some continuous level and then, you are basically trying to figure out in which if you divide the input range into a whole bunch of bins, right. Quantization is simply nothing but finding, which bin the input lies in, right.

(Refer Slide Time: 14:08)



For example; if this was the input range, and we decided to break this input range into say 6 levels. Now, all that we need to do is if the input was some value here, we need to figure out in which of these ranges the input is that is equivalently quantization, correct. So, it is in some sense you can think of it as a search problem, where the input lies in several bins, and you find out which bin I mean which of the bins is the right answer. So, this is nothing but a classic search thing, and it is no wonder that common search algorithms can be recast and hardware as an analog to digital converter, where what it is doing is trying to figure out which of these bins the input lies.

So, what I am trying to say is that the there are many ways of potentially doing quantization; just like there are many ways of doing search. So, which will boil down to many different hardware architectures to do this task, ok? So, you will have I mean one dumb thing to do, if you want to figure out which of these bins the input lies in is to compare the input simultaneously with all these thresholds, correct. So, for example; so, let us say this is 0, and this is the thresholds are 1 2 3 4 5 and 6, we have some unknown input which lies in this range. We would like to figure out which of these bins or which of these amplitude ranges the input lies in 1 very straight forward way is to say if I have a contraction or a box which can tell me if the input is less than or greater than a threshold, right.

And, this kind of box is a is called is what is called a comparator, whose output is a logical 1 if V in is greater than in this case 5 volts, and is a logical 0 if the input is less than 5 volts. Now, one way of doing accomplishing A to D conversion of the quantization operation is to simultaneously compare the input against all these thresholds, right. And, what do you think will the output of these comparators will be, everything will be 1 or 0, but is there a more specific pattern, do you think you will be able to expect. ((Refer Time: 18:03)) here there are how many comparators are there 5 comparators here, right. But is there do you think this output sequence of these the digital output sequence of these comparators, do you think it can be any arbitrary 5 bit word or is there some pattern to this.

Student: Some pattern.

Clearly because the input lies in only 1 bin at any given time it must follow that all the comparators, where the input is greater than, their thresholds must all fire a logical 1. All the comparators whose thresholds are higher than the input will have an output which is a logical 0. So, in general you should expect to see a string of one's followed by a string of zeroes, right. Depending on the height of or depending on the number of one's consecutive ones in the string you will be able to figure out, where in this range the input is located, correct. Now, one of you may say this is a very bad way of using resources because for example; if I want to digitize something or divide up the input range into not 6 bins, but into say 1024 bins right, which corresponds to how many bits? 1024 levels is nothing but 10 bits right, which is a very reasonable resolution to have in practice ok.

Now, if you want to do something like this, how many comparators will you approximately need?

Students: 124.

You will need roughly 1 give or take 1 at the beginning or end, your roughly condemned using 2 to the n comparators, if you want to achieve if n bit resolution, correct. So, if you have 10 bits you need 1000 comparators roughly. And, why I mean 1 of you might simply say this is a terrible ways of resources, because I know for sure that if the input is greater than 512 for example; that I do not really if my five hundred twelfth comparator says the input is greater than 512, it hardly make sense to sit and go and compare the input with all the levels below 512, you understand. I mean it only make sense for me to now look in into.

Student: Above to 500.

Above 512 all the way up to 1024, you understand. But this cannot be done in 1 single shot this has to be done iteratively. If I have fewer comparators then, I have to use them probably many more times than being able to do this in a single shot. Which is what I can accomplish; if there were, if all the comparators were there then, in single shot I will be able to right away tell you where the input lies, because; there is a lot of parallelism in this network right, because at many guys doing the same job, all right. And, you know you get speed at the expense of hardware complex. So, there are many situations where speed is important.

So, this architecture where you have you compare the input simultaneously against a whole bunch of references right, which as you can see is capable of giving you the answer very very quickly, right. This kind of approach is what is called a flash analog to digital converter, because here only it works in a flash. You must understand; however, that before I do this I must have a sample and hold, right. Somehow build the sample and hold into the comparator, sampling and quantization are both and are both necessary to make a complete A to D converter. Now, it might be possible to somehow combine the sampling and the quantization in some clever way right, but fundamentally somewhere along the line both these operations must be occurring.

Student: Sir this can sample and I had a series of comparators it will still work it out.

Well, it will give you an output, I mean; the question is if I did not have a sample and hold, but I had this array of comparators; I put in a continuous time input can any of you comment on what happens to the output is the output still a digital representation as we know it.

Student: It will keep on varying rigid.

I mean it. So, well it will keep on varying sure so, well.

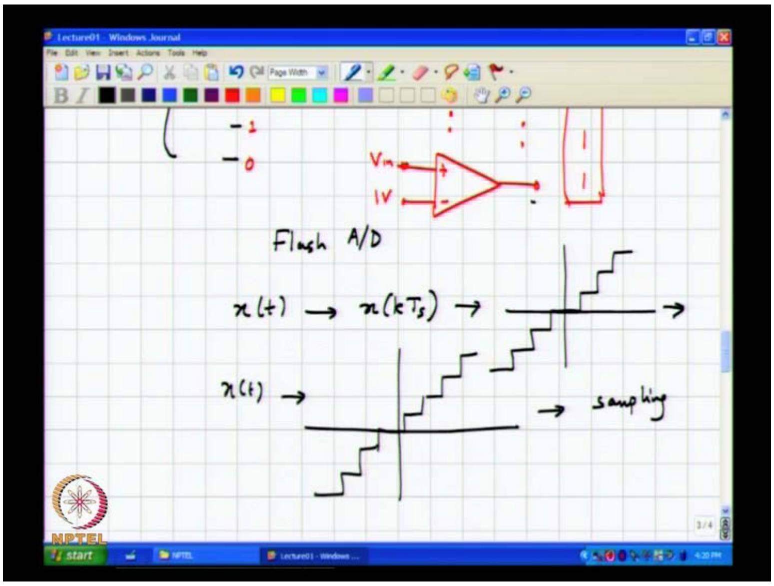
Student: Create digital thing will not be digital it will be continuation at clock pulse we have to get.

Correct. So, even though you can get discrete amplitude outputs by not having a sample and hold, right. That is not still a digital signal as we know it, as far as we are concerned a digital signal is one which is both discrete in time and amplitude. Here, even though the amplitude is discrete time is not, because every time assuming we have ideal comparators; every time they input crosses certain threshold right, which can happen at any time correct I mean that time is not quantized. So, even though the output is, has got discrete levels it is still a time is still a continuous. So, it is not a digital signal in the sense of what we have discussed, which where we require the signal to be both discrete in time and amplitude, you understand yes.

Student: In that case can we sample pull at the output of the component.

Well, not really because this is clearly please note that quantization of amplitude is a non-linear operation right whereas, sampling is a linear operation. So, interchanging the order of the operations does not result in, does not potentially result in the same system. There you can find systems, where it is that is indeed true right, but in general that need not be the case and in a practical flash A to D converter it turns out that interchanging the sampling and quantization. It does not work quite as well the other one, I mean in principle right.

(Refer Slide Time: 25:33)



If you sample and in this particular case; if I sample x(t) and if I get x of k times t s is the sampled version then, if I pass this through some kind of non-linear function like this. It is the same as doing applying the non-linear function first and then, sampling the output of the result ok. So, this is mathematically it is equivalent; however, when you try and make this in practice it turns out that it is much easier to deal with signals which are sampled and held, right. And, to process them and figure out which amplitude bin they lie in rather than do the opposite which is you know quantized and then, sampled.

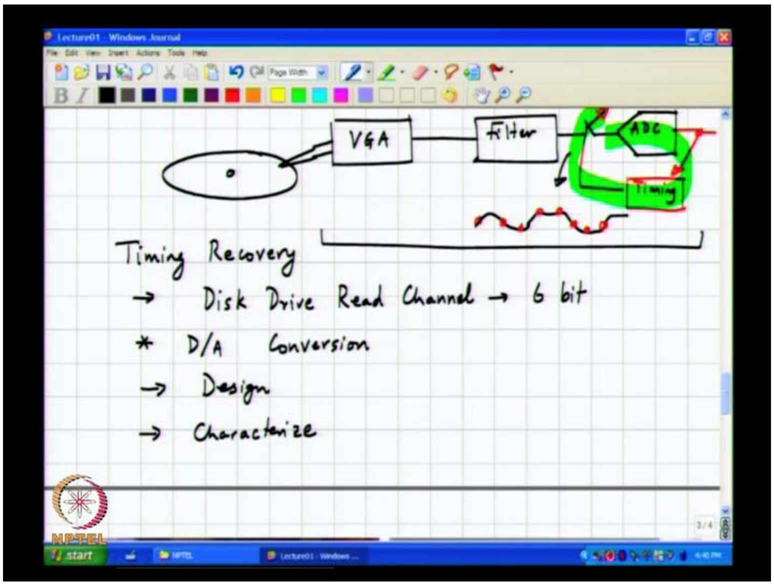
And, in general while in this particular case because this we are modelling the quantization process as memory less and pretty much ideal. You can neglect I mean the order of the sampling and quantization does not make a difference in practice these models are not really true, when real circuits are implemented. And, so, you get very different performance when you interchange the sampling and the quantization operations, right. But this is still a useful artefact if you want I mean; if you want to evaluate spectrum of signals and, so on. This is a useful thing to know because even though this is physically might not be feasible mathematically it is still an identity, which means that if you are finding it difficult to work with sample signals and then, quantizing them. You can work with a continuous time signal which is being quantized and then, sample it by spectrum.

So, mathematically since these 2 are interchangeable; we can still which is a still good thing to know, all right. So, as I said this way of using many comparators all with different thresholds simultaneously is one way of doing flash A to D conversion. And, this is a very often used in systems, where you really need speed and you need the output value instantly. So, what kind of systems do you think will need the output right away, any examples?

Student: ((Refer Time: 29:00))

Correct. So, whenever the A to D converter is part of a control system right a feedback system, you cannot effort to have too much latency, from your basic analog circuits class, you know that whenever you insert delay inside a feedback loop, your signing up for potential disaster in the form of instability. So, flash converters are often used in systems which just cannot tolerate delay, right. Where, you need absolutely minimal latency through the system.

(Refer Slide Time: 29:55)



A very good example of such systems were flash A to D converters are routinely used is the system that goes into reading data of your hard disk drives, ok. So, it turns out that if you kind of look at the signal processing chain of a disk drive system; that is you have a disk on it magnetic material is there every time. You are saving a file, you are basically writing on to the changing the orientation of the magnetic material on the disk, right. When you want to read it back, this disk is spinning at a rapid speed and you want to be able to record as much data as possible on a given physical say; for a given physical size of the disk, right. So, this disk is rotating at a rapid speed and you have a head which is sensing the direction of magnetization of that material on the disk and producing a small signal. From that small signal you are you should be eventually be able to figure out whether you written a 1 or a 0 ok.

So, it turns out that since; the signal is small first you basically need a variable gain amplifier because the signal strength can depend on the location in the disk, it can depend on the quality of the read head it can depend on how old the disk is and so on, right. So, you do not want them, you do not want to have a fixed gain you want to be able to vary the gain depending on the signal strength. So, that you get a descent signal strength at the output of the amplifier, ok.

So, there is also, some other non-linear processing that happens, but for the time being let us just assume that; a simple model to illustrate the basic idea. So, the variable gain amplifier goes to some signal conditioning circuitry which consists of a filter, the output of the filter is then, sampled with an A D C. Please note that this is for all practical purposes a digital communication system, right. What is being written on to the disk is supposed to be a sequence of ones and zeros. Now, what signal you see here, may have no resemblance to 1 0 sequence at all. The job of the entire system is to figure out given this wave form; what is the best sequence of ones and zeroes which is most likely to have resulted in this particular read out wave form, do you understand.

This is a classic problem in communication and the techniques that are used are also I mean all within codes communication techniques, right. Those who heard of the beta algorithm, and things like that which are ways of recovering, you know what sequence was transmitted given a received sequence, this is exactly the same problem except that the medium now is different, it is a magnetic medium and that is all there is good, ok. But from a fundamental problem point of view; this is the same system that people have been working on for a lot of time. So, this wave form here will probably look at I mean will look something like this. So, from samples of this wave form; one needs to figure out what data was written on the disk.

So, clearly you know one thing that you can see from this poorly drawn picture is that the time at which you sample has a significant bearing on what you eventually recover. For example; if you sample the output of this at the zero crossings of this wave form it will seem as if, it will look like a constant zero, correct. So, in other words there is a lot of sanctity to when you close this at what phase you close this sampling switch. So, there are, what are called timing recovery algorithms which look at this digital sequence, and figure out if the time at which this switch is closed is the right time. So, this as you can see is the feedback system, because; there is a feedback loop like this.

So, this is the classic timing recovery problem in communication systems, especially in hard disk drives since; the data rates at which you write and you read are very high. The sampling rates here are today of the order of several giga bits per seconds ok. So, today when you read data of a hard drive, you are mostly likely reading of what a giga byte per second, which means; that all this has to happen very quickly and if this timing recovery loop has to be stable you must make sure that the latency in the feedback loop is very small. And, if you want very small latency an obvious choice is the flash A to D converter, where we understand that it is very wasteful of hardware, because we seem to be doing a lot of redundant things, right.

But I mean if you cannot if you have to do it, if you need to get things answers quickly it is not surprising that you are not very efficient with the use of resources, all right. So, a disk drive this whole thing here right, as I mean the output of the A to D converter then goes into a digital engine, which implements all the algorithms which go and figure out what the sequences is. As far as the analog front end is concerned is a variable gain amplifier there is a filter there is an A to D converter and a timing recovery loop ok. So, this is a one of the most ubiquitous applications of flash A to D converters; that I can think of, right.

And, this all these electronics is called the disk drive, read channel. It describes the channel all the way from the magnetic medium to the end of the entire chain, including the D S P, all right. So, flash A to D's are often used in systems which require high speed and low to medium resolution. So, in a disk drive read channel typically you will have a 6 bit flash converter, 6 bits means 64 levels, right. 64 bins, which means; 63 comparators and 63 does not seem like too bad a number, all right. Whereas, if you wanted a 10 bit converter now, the number of comparators would have to become 1024 which is a large number times which is16 times larger than 64, you understand.

So, if the resolution is low then, while it is true that doing a flash is not a particularly good use of resources. We recognize in the need for speed, it does not seem like a bad idea and sometimes is the only way that you can make it things work, all right. So, this is you know 1 with encode straight forward way of implementing; anybody when you ask them how you plan to do it this seems like a very simple idea that comes to mind ok. The design of the flash converters is not a simple, simply because at these speeds making circuits work is a challenge, though on paper it seems like a trivial exercise, right. As we go forward we will see that there are whole bunch of design issues that one needs to think about before one can actually make a flash converter work, all right.

So, what is not there in terms of concept is compensated for a simply in terms of the speed at which the circuits need to operate, all right. So, basically one can this is one way of implementing a quantizer, and you can think of many other ways when, once we understand that this is basically a search algorithm. One can think of a whole bunch of ways of implementing the search, all of them will I mean; will vary in details of how efficient they are how fast they are and so on. And, it seems like a reasonable trade off that if you want to do a job very efficiently then, from hardware efficient say hardware and power efficiency point of view then, it follows that you will take time over. Which I mean it seems like a reasonable trade off, when do you reduce the use of hardware when you reuse the same hardware over and over again.

That is why; when you have a small amount of hardware to do the same job as before. The only way you can think of making it work is or making do with little hardware is to reuse the same hardware over and over again. So, some kind of iterative way of doing things so, examples like this are. So, called success of approximation converters which might also kind of vaguely studied in your earlier classes. And, another way of reusing the same hardware over and over again is what is called the delta sigma converter or the oversampled data converters family. So, do not worry about the details right now.

So, all that I wanted to emphasize was that there are many ways of potentially converting of figuring out in which level or which bin the input signal lies. And, a clearly we want to be able to realise a quantizer whose input output transfer curve looks like this. In other words; we want to chop up the input into a whole bunch of equal bins, right. So, the input is lies between 0 and 1 I need an output of 1, if it lies between 1 and 2 I need an output of 2, and so on. In practice circuits non idealities are likely to make the actual curve look different. If 1 obvious non ideality is that all the bin widths you want, you started off with the ideal of wanting all the bin widths to be the same. In practice because of non idealities the bin widths will be different.

So, it is not just simply enough to say I will get; if I will get 10 wires out with digital data. I must make sure that; this data is represents the input to some degree of accuracy. And, clearly multiple inputs right, can give the same output because you are because the very nature of quantization. So, there is clearly a loss of information when you quantize, you understand. So, this so, the size of the widths of the bins, and so on are likely to influence the quality of the quantized signal. So, we need to be able to understand not only how we are going to go and design a quantizer. We also need to understand, how we characterize a quantizer, all right.

On the other front, on the other side; we need to figure out the opposite side of the story that is given a discrete time discrete amplitude sequence how do we convert it into a continuous time you know; obviously, if the input levels are discrete. Any effort we make to make the output level continuous will necessary cause I mean you know it is like taking the same information and trying to generate something in the middle ok. So, this is the opposite of what we did on the input side and this is what is called D to A conversion. Where and, you know these things are at work every time you listen to music from an electronic box or whatever. And, in many precision instruments it is often desired to make all sorts of signals, and signal shapes, and it is very easily done.

If all these things are stored digitally and then you read them out, and you know go through a digital to analog converter then, generate the whole thing in an analog fashion, ok. Another application of high speed D to A conversion is what is called direct digital synthesis. If you want to generate a sign wave for example; one way of doing it is to build an oscillator this is a completely analog effect. Another way of doing it is to store the values of a sinusoid inside ROM and then, read out the values of the ROM and pass them into a digital to analog converter, right. So, the advantage of doing direct digital synthesis is at any time you want to change the frequency of operation it can be done on the fly without I mean which would be kind of difficult to do with an oscillator, ok.

So, these are you know several applications of digital to analog conversion. So, we need to figure out a given that there are so many ways of designing an A to D converter. It is not surprising that there will be as many ways if not more of designing a doing the opposite thing or designing D to A converter. And, then we need to be able to figure out how to characterize the output signal, which comes out of a D to A converter. And so, coming back to sampling; the way this is basically the overview of all the topics we will be covering in the course.

The first may be 10 to 15 lectures; we will deal with sampling analysis of sampling. How we are going to design the circuits, how we are going to sit and characterize the circuits once you have designed them. And, make sure that they are good enough. And so, from a purely mathematical point of view sampling can be an operation, where you do not lose any information. And, what is that enabling theorem that is telling you that you can do this.

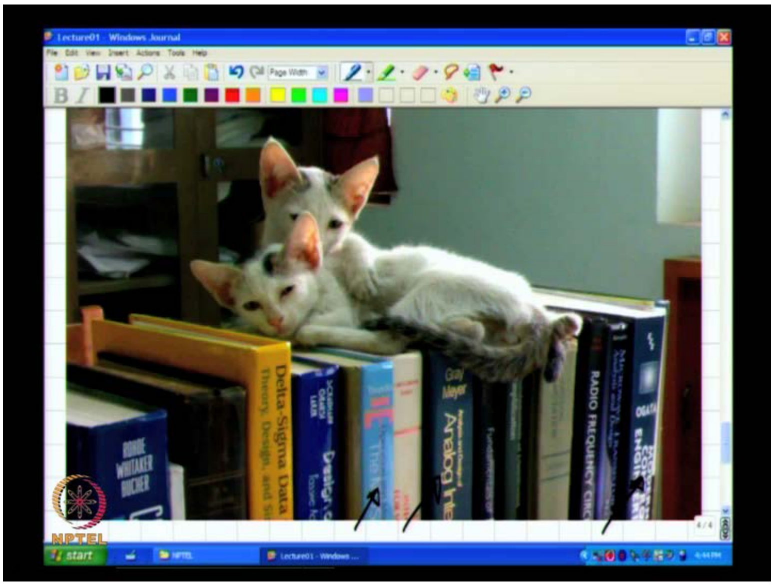
Student: Nyquist theorem.

The Nyquist theorem is the, which is telling you that; if you have a signal with the bandwidth B then, in principle you can reconstruct it perfectly as long as you have samples of it at a rate higher than 2 B, all right. So, we will sit and first mathematically understand sampling lot of this is most likely to be review material because I am sure your aware of all these from your D S P classes and so on. And, then we will get to the circuits and characterization. Given that you are dealing with continuous time signals discrete time signals quantization and so on, right. The kind of background required for work in the in this area is it is a multi pronged thing you need a bit of D S P to understand signals, which are quantized, which are sampled and so on, right.

Of course you need all the standard analog circuits background that you built up in your previous courses. You need some devices a little bit to understand finally, if you do not understand the transistor properly, it is not very likely that you will be able to sit and use it efficiently to design circuits. So, we will need to brush up if our knowledge on of basic device models and so on, right. And, a lot of this stuff, basically; the circuits background requires you know some knowledge of control in the sense of loop gain, stability and plot’s and that kind of thing.

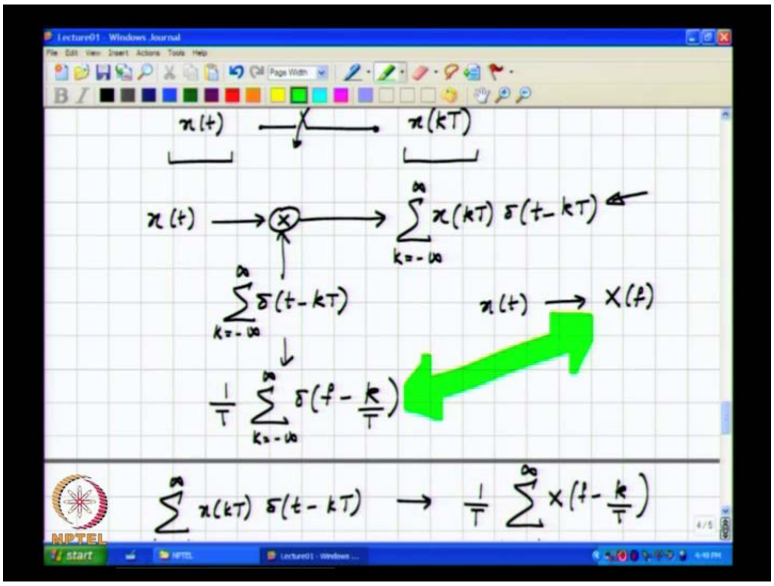
So, the; this is this will become especially important when we cover whatever; what we will call what are called over sample A to D converters right, which are based on feedback. So, that is why I like to show this picture ok.

(Refer Slide Time: 51:04)



So, if you want to be a good data converter engineer, right. You need to be like these 2 cats as we can see are sitting on a whole bunch of books, the meat is analog, right. Some this book is operation and modelling on the mass transistor. This is gray Meyer analog integrated circuits; this is control by ogata and so on, right. So, it is you have to be fairly well versed with a range of areas, it is not a very narrow, it is not a narrow field and that is what makes it fun right. So, ok.

(Refer Slide Time: 52:08)

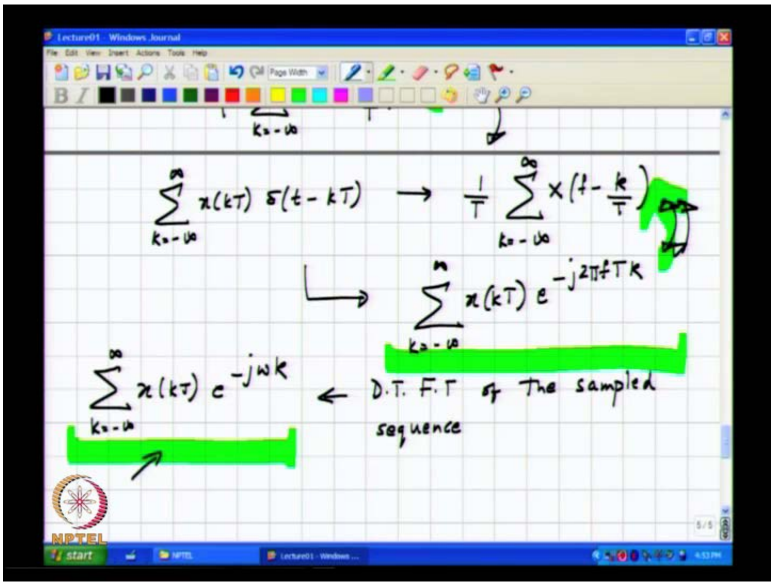


So, let us start with sampling. So, if you have a signal x (t) and you sample this at the rate I mean every T seconds. The sample signal you will get the samples are x (k) times t where k is an integer ranging from minus infinity to infinity. What will be interested in mathematically is how does, how do the properties of this signal relate to the properties of the input signal that we are sampling presumably there must be some relation between the 2 right.

So, to derive this relationship we proceed as follows; we first generate a continuous time signal right, which is the x (t) multiplied by a Dirac delta train. And, K ranging from minus infinity to infinity, and this gives what will be the output of this when I multiply a Dirac delta train with x (t) what should I get? I will get x of K times T delta of t minus k T sum over the infinity, all right. So, now, let us try and figure out the spectrum of this signal in 2 different ways; do you think this is, is this continuous time or discrete time? It is continuous time, this is still a; this is a continuous time signal; it is just that it is a periodic repetition of the continuous time I mean Dirac impulse.

It is a periodic extension of the Dirac impulse, and if x(t) transforms to x (f), all right. Then, what do we see, what do what is the Fourier transform of this. It is another impulse train it is 1 by T sigma k equal to minus infinity to infinity delta of f minus K by T, all right. Now, if you multiply 2 signals what happens I mean how can you relate the spectrum of the product to the spectrum of the 2 individual signals convolution. So, one way of finding the spectrum of this signal which is the continuous time signal multiplied by the Dirac impulse train, is to convolve the spectra of the input signal with the spectrum of to convolve the spectra of the input signal, and the spectrum of this Dirac train right, which happens to be this character here.

(Refer Slide Time: 56:15)



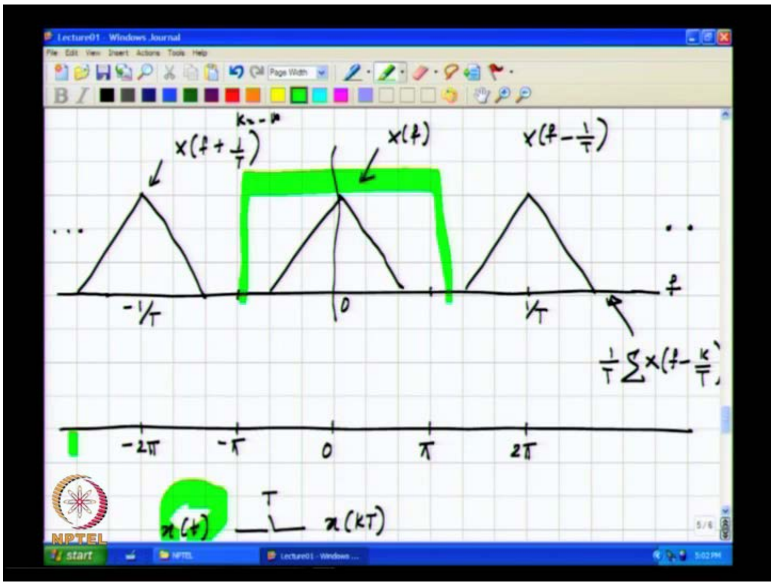
Now, that results in x of k t delta of t minus K T, therefore; transforms to based on, right. So, x (f) if I convolve x (f) with this, the result is this. Is that clear? On the other hand we can find the spectrum of this in other way, right. And, what is that it is you can think of this signal has the sum of impulse; whose strengths are x of K T. And, therefore; the spectrum of this can also be written as sigma K equal to minus infinity to infinity, x of K T, e to the minus j 2 pi f T times K, correct. Because the spectrum of a delayed impulse is nothing but e to the minus j 2 pi f times whatever delay there is, and the strength of each impulse is x of K T.

So, you now just you simply need to add up the Fourier transform of the each of these delayed impulses. So, in other words; these 2 are the, are one and the same thing, it is just that we found the spectra in 2 different ways. Now, the discrete time sequence is x of K T, right. And, I can find it is I am trying to relate the properties of the discrete time sequence to the properties of the continuous time signal. So, the discrete time sequence has a Fourier transform, a discrete time Fourier transform e to the minus j omega K, correct. So, this is the discrete time Fourier transform of the sample sequence, right. So, now, all that we need to do is to connect these 2 things up we want this, but we know that this equals this.

So, what is the connection you need to make, right. You need this character and you know this right, which basically means that since; I know sigma x of k t e to the minus j

2 pi f T times K, if I need to find sigma x of K T e to the minus j omega K it is simply a matter of replacing omega with 2 pi in fact it is a matter of replacing 2 pi f T with omega in which expression, this expression right.

(Refer Slide Time: 1:01:00)



So, to determine the spectrum the discrete time spectrum of the sample signal what we need to do is the following; first is form 1 by T x (f) minus K by T, all right. Do you think this is periodic or not?

Student: Yes sir.

This is periodic with what is the period? 1 by T. So, once you form. So, let us so, this is the original x (f) this is x (f) minus 1 by T, this is x (f) plus 1 by T and so on, and this will give me, what is this now? This is nothing but 1 by T sigma x (f) minus K by T, you follow; once you form this what is the x axis here? f or omega it is f right, to get the discrete time spectrum all that we need to do is to replace f with omega by 2 pi into T. In other words; it is a scale factor, correct. So, in other words if since this is a scale factor let us you know make a small table of what to expect f is 0 means omega is what? 0, yes 0, very good f is 1 by T must correspond to an omega of 2 pi, all right, which therefore; means and please recall that this spectrum is periodic with.

Student: 1 by T.

1 by T, which means; that the discrete time spectrum will be periodic with 2 pi because 1 by T corresponds to 2 pi, and 1 by 2 T must correspond to 4 pi no it is not 4 pi, it is Pi correct ok, all right. So, since this is periodic with since the continuous time spectrum is periodic with 1 by T it follows that the discrete time spectrum is periodic with 2 pi which basically means; that you know any chunk of the x axis extending over a range 2 pi is all that you need. The rest of it is just copy and paste versions of what is there here. A convenient thing to choose is minus pi to pi. So, once you form this the discrete time spectrum is simply replaced 0 with 0, 1 by T with 2 pi, 1 by 2 T with pi, and so on, all right.

So, and this also clearly shows why you need a sampling rate greater than, I mean; twice the bandwidth because if the sampling rate is not high enough then, what will happen. These 2 will overlap with each other and, you know the signal get corrupt. In other words; when we what do we mean by we can uniquely reconstruct the signal from its samples, what it means is that given these samples you can use some kind of processing to generate the continuous time signal back, you understand. So, all the what we have said now is that if you have an x (t) you sample it at 1 by T hertz then, you will get a discrete time sequence x K T. And, the discrete time spectrum of x of K T is basically something which goes from is periodic with 2 pi. And, can be obtained by making taking x (f) and repeating it at multiples of 1 by T and then, replacing 1 by T with 2 pi is scaling the axis ok.

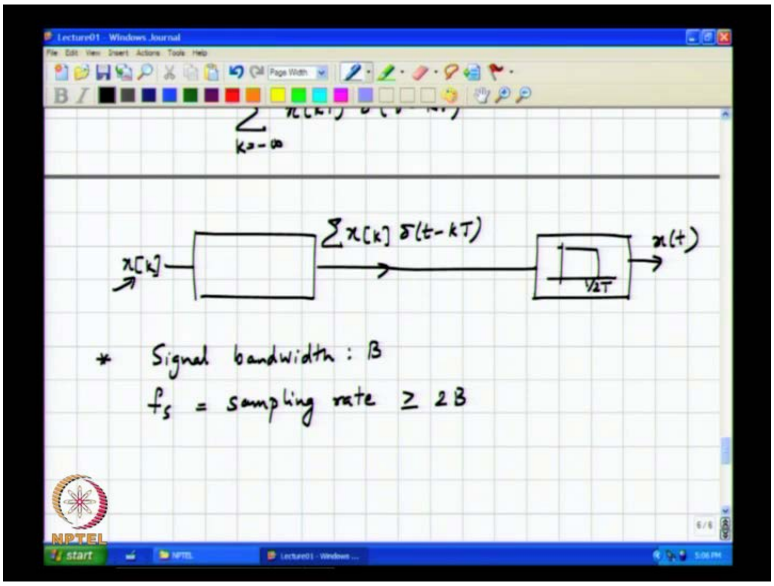
Now, the reverse also holds to, which means; that if I have a discrete time sequence x of T all right, if I made a continuous time impulse train I had in other words I have a discrete time sequence x of K T, right. Now, if I made a continuous time impulse train like this, what would be the spectrum of that train, I mean; what is same?

Student: ((Refer Time: 1:07:26)).

It is. So, now if I turn everything upside down and why I am interested in turning upside down, because eventually I am interested in also converting from digital to analog right, which is exactly the reverse operation. So, if I had a discrete time sequence right, and if I form a continuous time waveform like this then, I can relate the spectrum of the continuous time wave form to the spectrum of the discrete time signal in exactly the same fashion, right. So, now it is mentally you have to think of this is not being there only this axis is being there, the lower axis that is the discrete time signal by definition its spectrum is periodic with 2 pi.

Now, if I made a continuous time signal which is x of K T multiplied by Dirac impulse train then, the spectrum of the continuous time signal thus formed can be gotten from the spectrum of the discrete time signal by simply replacing 2 pi with 1 over T, right. And, then there is a scale factor, but that is not important right, you understand. So, now once I have this spectrum; how I mean if I want to generate x (t) back again what do you think I can do? I need a filter which only selects this part and rejects all the others. So, conceptually converting from discrete time to continuous time is x (n), right.

(Refer Slide Time: 1:09:24)



I will somehow push this into a box which generates or let me call this x (k), x (k) delta of t minus k T, right. Once it does this I need to pass this through an analog filter whose; bandwidth is what? I mean; where is, I mean this is 1 by T.

Student: 1 by T.

Where is 2 by T to the right or to the left?

Student: Right hand. ((Refer Time: 1:10:20))

You need a filter with the bandwidth of 1 by 2 T, correct. I mean you need to get rid of all these images. So, if I put in principle on ideal low pass filter of bandwidth 1 over 2 T, I will be able to get x (t) again where x (t) was the continuous time signal from which we got the samples, right. So, this you should be I mean you know familiar with both sides of the operation, you understand. And, this is and in this whole process we have therefore; reconstructed the, we have taken a signal we have sampled it right, at sufficiently high frequency and then, we have also reconstructed the original signal back from the samples that what is it means when you say perfect reconstruction, right.

Now, if the sampling rate was not too high, what do you think will happen to the reconstructed signal? Let us say our sampling rate was not high and we used a sampling rate which was, we did this. So, the sampling rate is not 1 by T right, but it is I deliberately reduce the sampling rate by a factor of 2. So, the new spectrum will look like this. So, when I try and reconstruct; what am I reconstructing right. So, when I do this what am I getting? I am getting a spectrum which kind of looks like this. So, when I add things up it roughly looks like this. And, when I put an ideal low pass filter to cut off images what am I getting? I am getting something, I am getting a signal I am selecting this part of the signal as you can see that is very different from the original signal.

So, from this it is apparent that as long as there is no overlap of the images you would be able to reconstruct perfectly right, which is simply reproving the Nyquist theorem, you understand. So, the bottom line is therefore; if the signal has a bandwidth B sampling rate must be at least 2 B, all right. So, let me ask you a question before I close for the day; we know that human voice for example, cannot exceed well, that is telephone I mean telephone voice quality the bandwidth is only 3 kilohertz. But let us say you are not interested in telephone, but you are interested in a music quality reproduction what do you think is the range of?

Student: (Refer Time: 1:14:16))

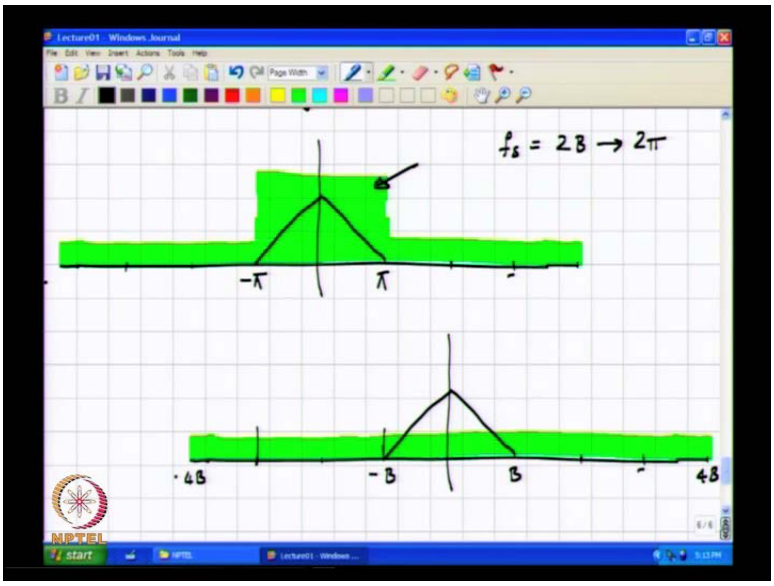
So, a human hearing is goes all the way from 20 hertz to 20 kilohertz right. So, in principle do you think I can just simply take the signal that is coming out of a mike and sample it at 40 kilohertz and I am ok, I mean say of 50 kilohertz.

Student: If you can represent it with sufficient number of bits then, it is if you have perfect low pass filter.

No the query of the let me rephrase my question right. So, we all agree that we cannot hear anything greater than 20 kilohertz anyways. So, 0 to 20 kilohertz is all that we are interested in. So, if I have a mike for example; I mean for example, whatever I am saying right now it is being recorded do you think it would be done it is if I simply record it directly put this simple input signal into an A to D converter, which is sampling at with sufficiently large number of bits to resolve. But if it is sampling at say 40 or 45 kilohertz do you think it is ok.

Student: Sir even though the audible range is 20 the input signal might have higher frequency in only very good.

(Refer Slide Time: 1:15:49)



So, the point I wish to make as he pointed out is that even though the desired signal has only a bandwidth B, in practice any sensor that you use to pick up the desired signal will also pick up noise. And, most often that noise is broadband and its bandwidth will exceed the bandwidth of the signal. So, if I represent the noise by this green stop here, for argument sake let me say the bandwidth exceed of the noise exceeds, which this is 4 B. So, bandwidth of the signal exceeds the range of the desired input signal by a large factor in this example chosen to be 4.

Now, if I directly sample this at 2 B what do you think will happen, how do we form the spectrum of the sample signal? Simply take the input spectrum shift and add and how much do we shift it back? Copy, paste I shifted this by 2 B, 2 B is the sampling rate, I shifted this by 2 B I must also shift it by 4 B minus 2 B minus 4 B and all multiples thereof, and add up all these things. So, when I add up all these things what do you think will happen?

Student: Noise.

So, within the final thing we are I mean if the sampling rate is at 2 B, 2 B corresponds to in the discrete time domain how much? Sampling rate is 2 B. So, in the discrete time domain this must correspond to 2 pi, correct. So, in the sample signal and B will correspond to therefore; 2 pi. So, in the sample signal eventually what will happen there is of course, the desired signal which ranges from minus pi to pi completely simply because we have sampled exactly at Nyquist, but you also have you know adding up all these components. So, what will happen noise will get piled up higher and higher right, because of I mean as far as the sensor is concerned its signal the sensor is seeing a signal whose bandwidth is actually you may be interested only in bandwidth B. But the sensor is picking up signal with much higher bandwidth, which includes also noise.

And, while your sampling rate is good enough for your signal that you want it is not good enough for I mean broadband disturbances which most often are noise. And, really this noise is very broadband which basically means that; just because your desired signal has got bandwidth B, it does not mean it that you can go ahead and directly sample it with at a rate 2 B, correct. Because noise components from outside the bandwidth will all alias or fold back into and basically as far as this is concerned its aliasing is not it. These are all components aliasing is a phenomenon where a frequency f I mean; which is much higher than the sampling rate. Masquerades as a tone which is within the sampling rate, a classic example of this is the old movies, right. Where you see you know somebody chasing somebody on a on a horse carriage right, and you it seems as the wheel is turning.

Student: Backward.

Backwards how is it possible that the cart is moving forward and the wheels are turning backward?

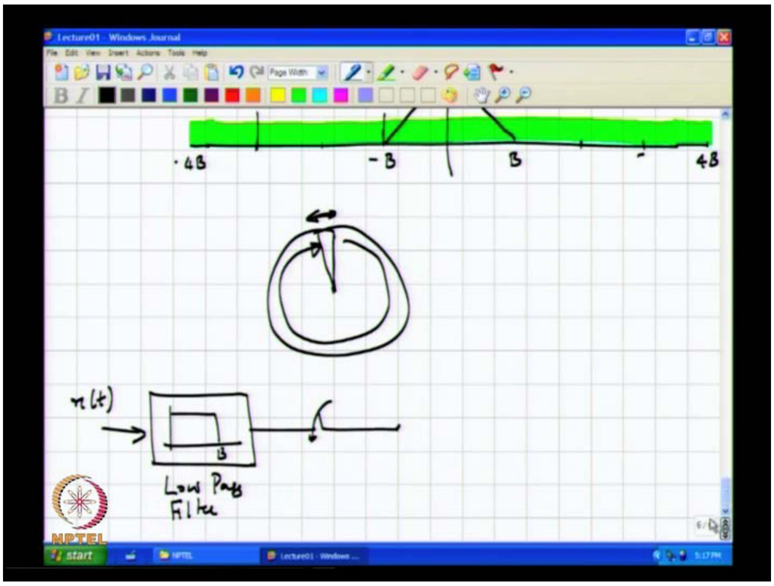
Student: aliasing.

I mean what is aliasing here?

Student: number of rims is less number of rims per second.

Correct. So, picture is basically. So, many frames per second in other words the image is being sampled at 24 hertz 24 frames per second movie, and if the wheel is turning very rapidly.

(Refer Slide Time: 1:21:16)



In reality; what might be happening is that between successive shots of the camera the wheel might have turned almost 1 full circle. So, a spoke which was here in the first frame, in the second frame when you sample it might look like that correct, you understand. But for us when you look at it looks as if this spoke has moved backwards. In other words; we think of this as a signal in the opposite direction at a very low frequency. You understand, but that is happening what is happening in reality is that the frequency the rated which were sampling is not adequate enough to catch the signal, right.

So, this signal is masquerading like a low frequency signal, you understand. So, it is pretending to be a low frequency signal, but in reality it is much higher than the sampling rate. And, that is basically what is happening to all these noise components also here, right. The Nyquist band is only minus B to B, but all these noises beyond B when you sample this at 2 times B, these noise components are all folding back into this minus B to

B range. And, what do we see here? In that discrete time spectrum; the signal noise is this, but in relation to the signal the noise is increased dramatically, right. Because if you look at the original continuous time spectrum, if you looked at only places where the original signal was the noise is some level relative to the signal. And, the discrete time spectrum; the noise is much more simply because of aliasing of noise from outside the signal band. So, what do you suggest we do, to fix this problem?

Student: bandwidth.

You can say that I know that my signal is restricted to bandwidth B, and to prevent out of band components from aliasing into my signal band after sampling what I will do is to put a low pass filter with a bandwidth B, you understand. So, in other words you can, you should never sample without low pass filtering out, signals outside your bandwidth of interest. So, the job of this filter is to remove potential components which can alias into your signal band. So, what do you think the logical name for this is?

Student: Anti-aliasing filter.

It is called an anti aliasing filter. So, no A to D converter should operate without an anti-aliasing filter, you understand. So, I will stop here.