Digital Signal Processing

Prof: S. C. Dutta Roy

Department of Electrical Engineering

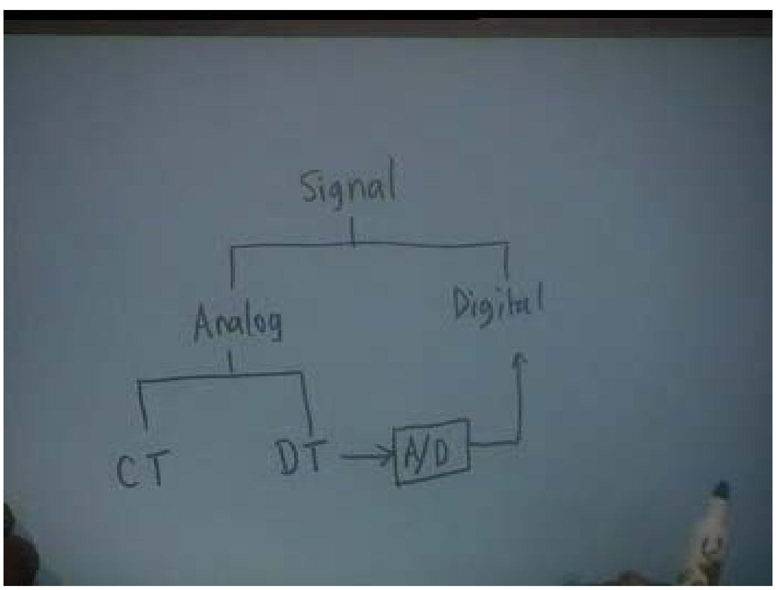
Indian Institute of Technology, Delhi

Lecture - 2

Digital Signal Processing Introduction (Contd…)

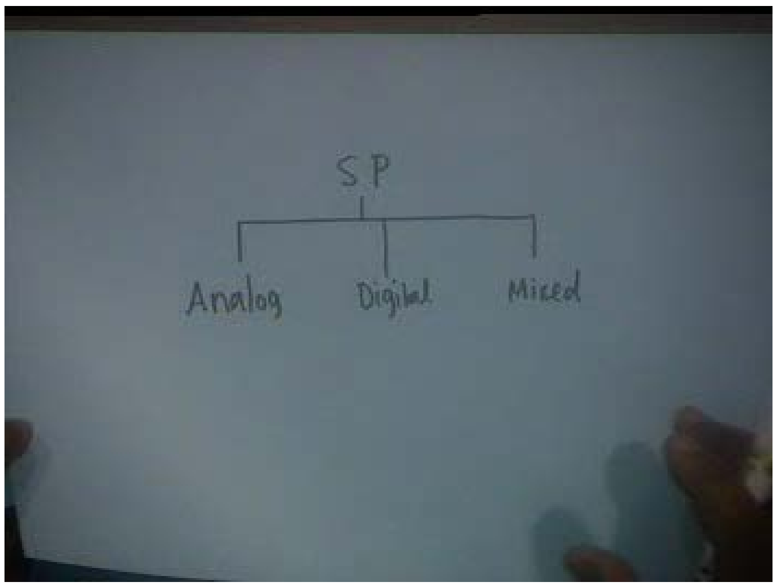
This is the second lecture on DSP and the topic of today is Introduction to Digital Signal Processing, continued from the last lecture, and we shall also discuss digital signals. In the last lecture we talked about the course contents, the books that are recommended, viz Mitra for DSP and Oppenheim and Schafer for DTSP. We talked about definition of a signal. A signal is a function of one or more variables. And accordingly it can be a one dimensional signal, two dimensional signal or multi dimensional.

(Refer Slide Time: 02:19 - 03:19)



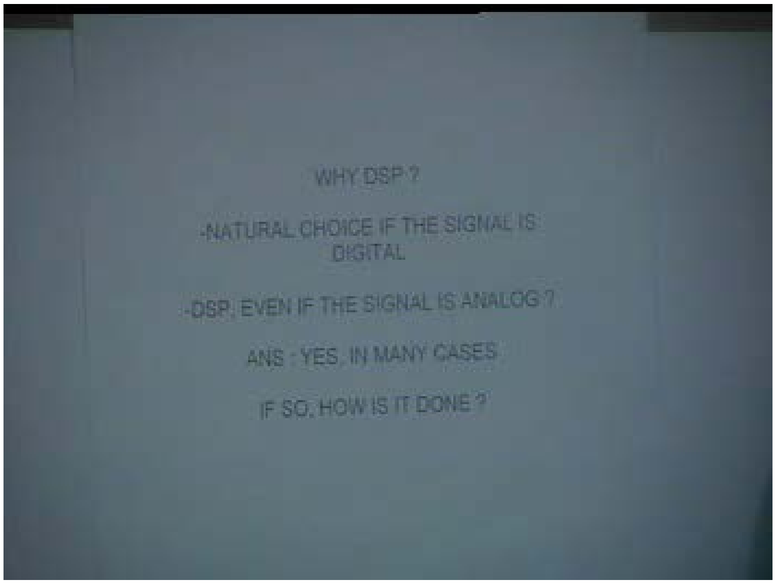
We also said that basically there are two kinds of signals: analog and digital. Analog can be continuous time (CT) or discrete time (DT). A discrete time signal is not a digital signal; a discrete time signal is one in which the amplitude is a continuum, and not discretized. On the other hand, if the time is discretized, this signal is still analog. Only when a discrete time signal is passed through an A to D converter, it becomes a digital signal. To repeat, in analog domain, we can have two kinds of signals-continuous time and discrete time. And then we also emphasized the need for processing: why should one process a signal? This is to obtain a better quality signal than the original one. It could be for example, noise filtering or improving the quality of a picture in terms of brightness or contrast or whatever it is.

(Refer Slide Time: 03:57 – 04:48)



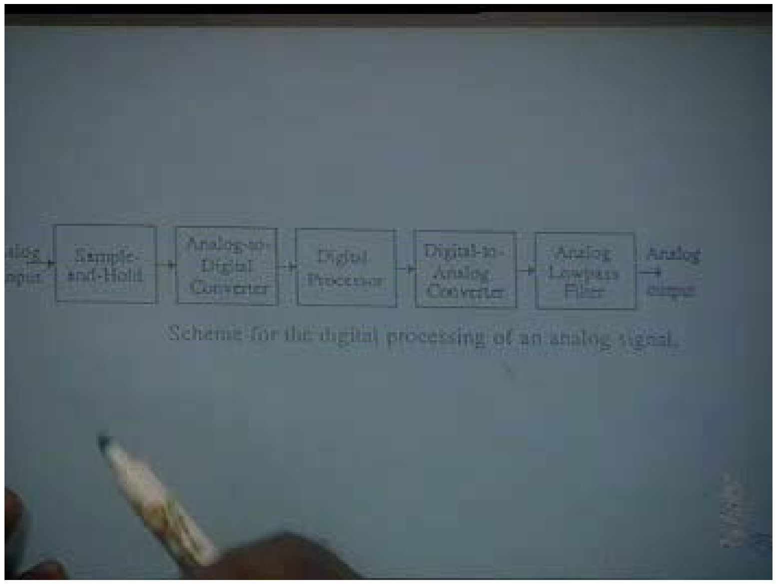
We also said that signal processing that one can make can be of three types. You can have analog signal processing or Digital Signal Processing or mixed signal processing. We also emphasized that analog signal processing dominated prior to 70s. Between 1970 and 2000 it is basically Digital Signal Processing which got dominance over analog. The current trend is to have mixed signal processing, namely a combination of digital and analog. The next question that we elaborate on is, why Digital Signal Processing? We can do analog signal processing only, because most of the signals in practice are analog. But DSP should be the natural choice if the signal is digital, for example, the rainfall data at Delhi over the last 20 years on which is based the prediction for the next year. Predictions sometimes fail miserably, as it has this year, but the data that you collect is a digital data: 1980 so many inches, 1981 so many inches and so on. Now the amplitude can be a continuum so it is a discrete time signal, but the amplitude or the number of inches of rainfall can also be coded, to make it a digital signal.

(Refer Slide Time: 05:56 - 07:14)



If you so desire, if you want to do DSP then the amplitude or the height of rainfall can also be coded. But you see the time is discretized: 1980, 1981, 1982 etc and there is nothing between 80 and 81, there is nothing like 1980.5 or 1980.41. So, the natural choice would be DSP and this is what is done. This data is fed with many other parameters, controlling monsoon or rainfall to a super computer which basically does DSP. What is DSP? Addition, multiplication and delay or recalling of past signals are the basic operations in DSP. Even if the signal is analog, one prefers to use DSP. The reasons are many. DSP has many advantages over ASP or Analog Signal Processing. And many a times it happens to be a less costly proposition. Digital IC’s are available off the shelf at very low cost and therefore in many cases, analog signal processing gives way to Digital Signal Processing even if the original signal is analog.

(Refer Slide Time: 07:49 - 11:08)

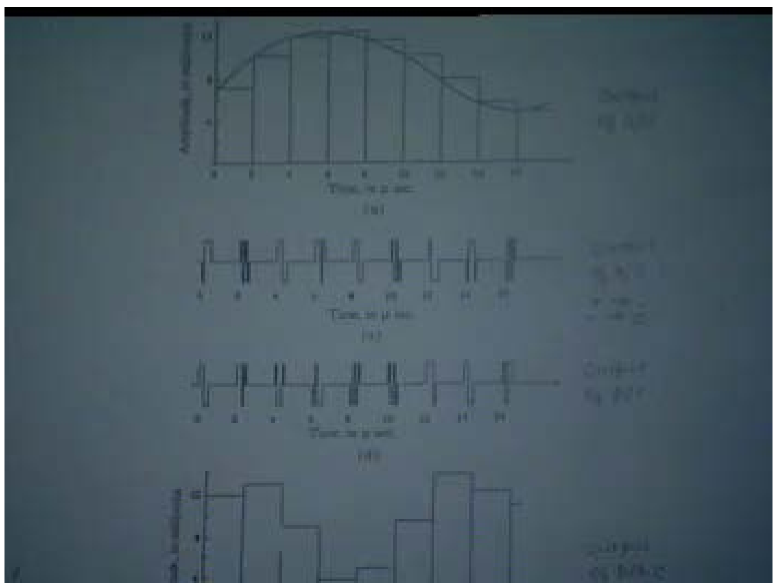


How it is done is represented by this block diagram where the input is analog CT; then we have a sample and hold device which basically consists of a switch which samples the analog signal and holds the samples for some time so that it can be converted to a digital form, because conversion from analog to digital requires time. During that time the signal must be held. After the A to D, you get a coded signal, whose usual form is binary. This is then processed by a digital processor. What basically the DSP does is to combine signals by multiplication, addition and by recalling past signals, through delays.

There are basically three operations: delay, multiplication, and addition. So this is what the digital processor does. It is a very simple device, it simply performs 3 operations. Multiplication is repeated addition, with only two signals at a time; subtraction is also a form of addition with one of these signals negated or the sign bit changed. So, there is no integration, there is no differentiation; there are no complicated operations which limits the usefulness of analog processing. The output of the digital processor is also digital, a sequence of numbers, and in order that output will be useful, you must convert it back to analog form. So you have a D to A or digital to analog converter. The output of the digital to analog converter is in the form of stair cases.

The signal changes from one level to the next and then stays. The abrupt change gives rise to high frequencies in the signal. These high frequencies present in the signal have to be gotten rid of by an analog low pass filter, and finally this analog output is the output that is to be used. It could be, for example, a piece of music which has to be filtered from surrounding noise and perhaps mixed with other signals, and finally it has to be played. It has to be heard on the loud speaker and therefore you do require an analog output. Pictorially this slide represents the basic operations.

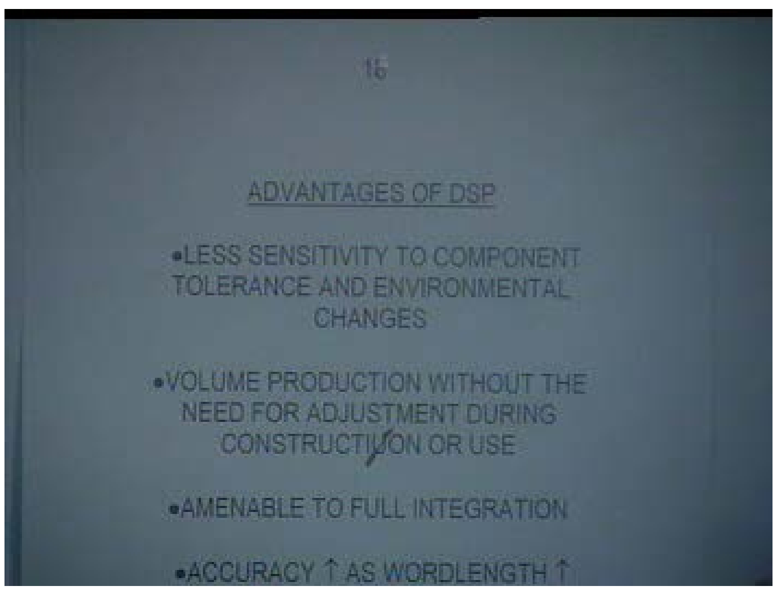
(Refer Slide Time: 11:23 - 13:19)



The first picture is that of an analog input which you wish to process by a digital processor and this is what happens when you sample it. The signal is still sampled at 0, 2 microseconds, 4 microseconds, 6 microsecond, 8 microseconds. This is uniform sampling and at each point, 6 microsecond for example, you pick up the value of the signal and then hold it till the next sampling comes. This holding is required for A to D conversion, which requires time. After the A to D conversion, the signal may look like a sequence like this, where 1 stands for positive and 0 stands for negative.

There are many other choices of coding but this is a typical coding where any pulse which is positive is taken as 1; for example this part of the signal is 1, 0. Then after processing by the DSP which carries out the operations of multiplication, addition and delay, may be the output of the DSP is something like this, which now has to be converted to analog form. So you have a D to A converter which again produces stair case waveforms like this. That is, these codes are converted to amplitudes, volts or amperes and a typical D to A output looks like this, which, as I said, contains high frequencies which have to be got rid of. Therefore you pass this through an analog low pass filter to get the desired output which may typically look like this. This picture shows exactly the operations which are involved in processing an analog signal by digital means. The advantages that accrue out of the use of DSP, compared to ASP, are many.

(Refer Slide Time: 14:07 - 17:17)



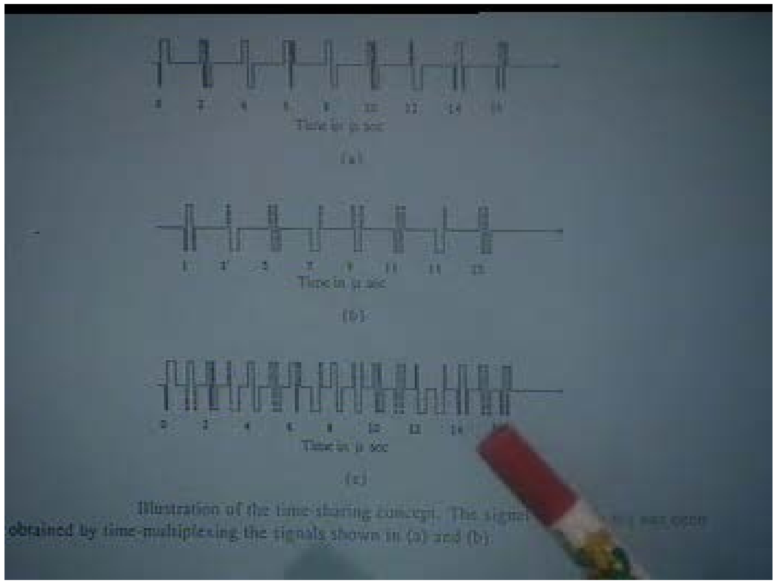
First, the digital signal processor is less sensitive to component tolerances and environmental changes. In analog signal processing change in any element, an inductance or a capacitance, or a small shift in the power supply causes a change in the performance. This also happens in digital circuits but since you are coding high amplitude by 1 and low amplitude by 0, even if the amplitude changes by 10% it does not matter, because 1 remains 1 and 0 remains 0. Unless the amplitude drops down to the noise margin level you have nothing to worry. Therefore digital circuits are less sensitive to component tolerances and environmental changes. Digital circuits allow volume production without the need for adjustment during construction or use. Once the process is set, the timings and the steps, nothing has to be changed. Digital circuits are of course amenable to full integration. This is not possible for analog circuits because of difficulty of integrating inductances and transformers.

A transformer cannot be made into an integrated chip; same is the case with an inductance of a respectable value. Inductances of very small values (Pico-Henry) can be made by IC's. But an inductance of a respectable value like micro-Henry or milli-Henry is very difficult to make because inductance basically requires a large volume. Inductance is flux per unit current. In order to generate flux you require space. You cannot generate a large amount of flux in a small space unless the magnetic permeability is astronomically high. Very high permeability materials are not available and therefore analog circuits are not amenable to full integration; Inductances and transformers have to be outside the chip.

In Digital Signal Processing, accuracy can be increased almost indefinitely by increasing the word length. Of course the cost also rises; as you put more number of bits you have to pay more but the accuracy increases. In an analog signal processor the dynamic range is limited by the power supply. In an operational amplifier (op amp) for example, the output cannot exceed + V_{cc} or – V_{cc}; it gets saturated. In Digital Signal Processing since we are handling numbers there is no question of saturation. We can handle as large numbers as possible and if fixed point does not do the job, use floating point with characteristic and mantissa stored in separate registers. You can then achieve very large dynamic range. This is not possible in analog signal processing.

The same digital signal processor can do multiple jobs. For example, use the same processor to time share two different operations and the following picture shows how this is done.

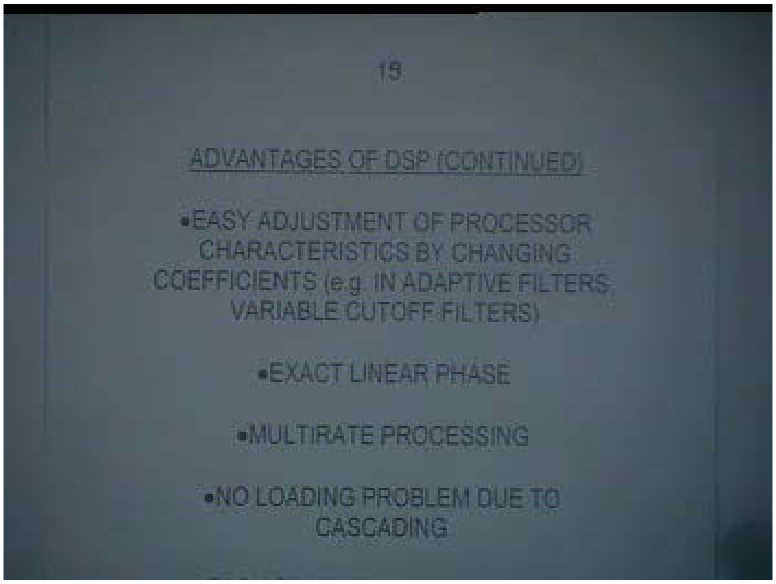
(Refer Slide Time: 18:23 - 19:26)



The upper signal is one which exists only at 0 seconds, 2 microseconds, 4 microseconds, 6 microseconds and so on. In the space in between you can put in another signal (this is time multiplexing) at 1 microsecond, 3 microseconds, 5 microseconds and so on. These two signals are there mixed and the composite signal looks like this. So if there is a processing occurring at odd values of time then it handles only the first signal. The same processor can also perform another operation taking signals at the even values of time. So the same processor can perform operations on two different signals which have been time multiplexed and then at the output you can separate the two. Use of the same processor simultaneously for different operations is a great advantage of using a digital signal processor. Time multiplexing is one of the great advantages of DSP; you cannot do it with ASP. If you mix two continuous time or discrete time signals, there is no way you can separate them.

Continuing the advantages of DSP, a digital signal processor can be very easily adjusted for different characteristics by simply changing some numbers, which form the coefficients. This can also be done in analog signal processing by varying a resistor, or a capacitor or an inductor but changing a number is much simpler then changing R, L or C. Varying an inductance for example is a very elaborate procedure. Varying a capacitance is not a problem but then you have problems of contact. You also have problems of environmental changes and so on and so forth. Similarly for resistors, you know that the potentiometer contact gradually wears off due to mechanical friction. There is nothing like this in DSP.

(Refer Slide Time: 22:14 - 22:50)



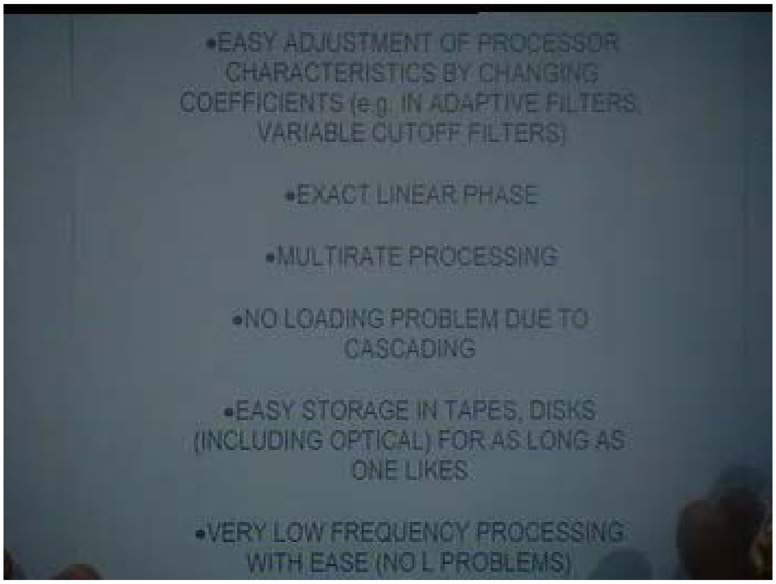
You simply change a number which is one of the coefficients and then you get a different characteristic. This is required in adaptive filters or variable cutoff filters.

One of the major and dominant advantages of DSP is that exact linear phase systems can be designed. This is not possible in analog signal processing and has been a dream for analog communication engineers. Exact linear phase means there is no delay distortion. We shall come back to the importance of linear phase in a later lecture. But this is one of the major advantages of DSP that it can attain exact linear phase and therefore it permits processing without any delay distortion.

One of the advantages of DSP is also that different parts of the signal processor can work at different rates. That is, the sampling rate can be different in different parts of the processor.

There are well established techniques for changing one sampling rate to another. You can sample up or sample down without any problem. This facility is not possible in analog signal processing.

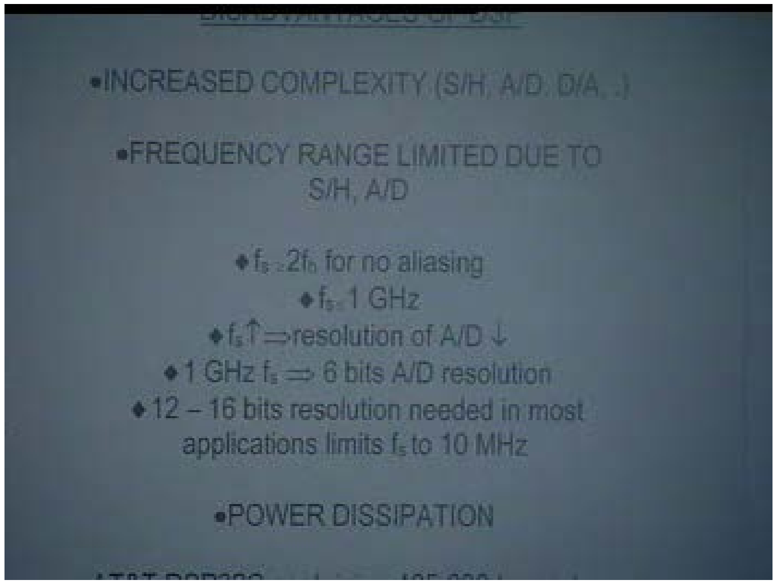
(Refer Slide Time: 23:50 - 24:58)



When two amplifiers are cascaded, ideally the overall gain is the product of the two gains. However, this is not invariably the case in practice because one amplifier loads the other. As soon as you put a loud speaker at the output of your power amplifier, unless it is perfectly matched, the power amplifier output goes down. There is no such problem in cascading two or more digital circuits provided you do not exceed the fan out capability of the circuits. In short there are no loading problems due to cascading in digital circuits. Digital signals can easily be stored in magnetic tapes and disks, including optical disks, as long as one likes. There will be no problem in storage.

In DSP, very low frequency signal processing is possible with ease. But in analog signal processing, inductance poses a problem. There is no such problem in Digital Signal Processing at very low frequencies. However, all are not roses with Digital Signal Processing, there are disadvantages also. The major disadvantage, of course, is increased complexity. You have to use more devices: sample and hold, A to D, D to A, low pass filters etc. Another major disadvantage is that the frequency range for operation of a DSP is limited due to the limited rate at which you can sample and hold. Also in A to D conversion, the speed is limited.

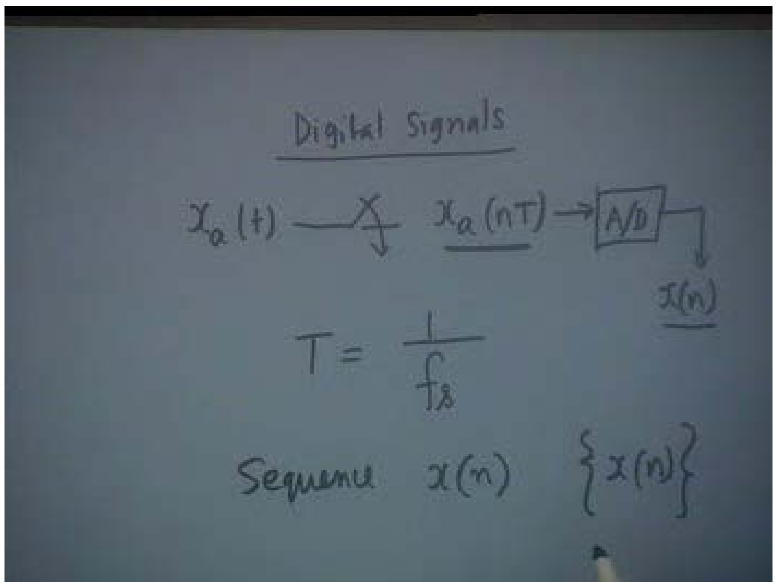
(Refer Slide Time: 26:02 - 27:24)



In practice you cannot use a sampling frequency, with today's state of the art, which is more than 10 mega hertz. And therefore if the sampling frequency is 10 mega hertz, naturally signal processing cannot be done for more than 5 mega hertz. The other disadvantage is that, if you have a purely passive circuit consisting of inductance, resistance and capacitance to do the analog signal processing, there is power dissipation only in the resistance. But it is much less compared to the power drawn by digital circuits, main dissipation being in the active devices, the transistors. A typical AT and T DSP32C processor which contains 405,000 transistors dissipates 1 watt. There are no such problems with purely LC circuits. But even then one does use DSP because of its dominant advantages. This picture is very rapidly changing, however, power dissipation being one of the major challenges for DSP engineers.

One of the aims of TI Texas Instruments is to make digital circuits which will require no power supply, particularly for devices which are to be implanted in live bodies like the human body. The heart transplant or the hearing aids would not, if DSP engineers are successful in future, require a battery. Battery is a major problem; to make a small battery is a severe problem. If you want to put it inside the human body, space as well as life of the battery are the problems. So they are aiming at devices which will work from body temperature. As long as the person is alive and his body temperature is between two limits, the device will work. We now come back to digital signals.

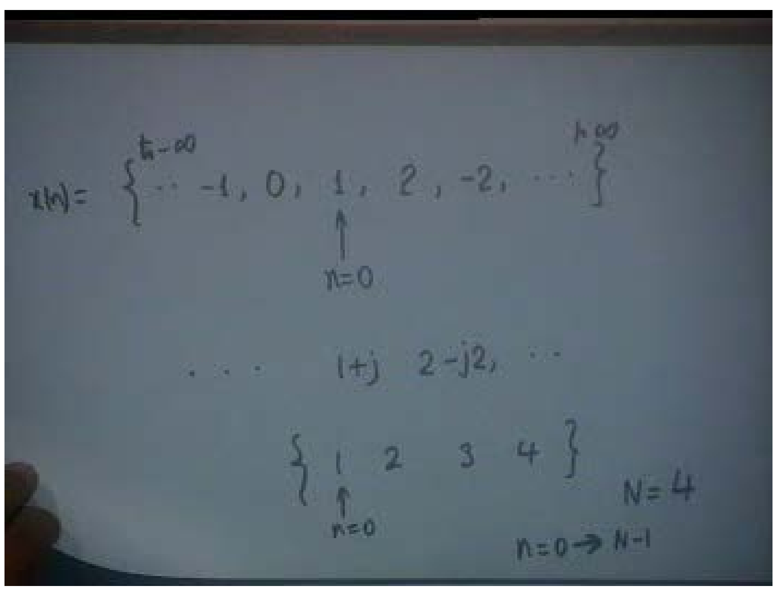
(Refer Slide Time: 29:12 - 33:09)



A digital signal is obtained from an analog signal by sampling. What you get here after sampling $x_a(t)$ at regular intervals of T is $x_a(nT)$; this is not a digital signal but an analog discrete time signal. It exists only at capital T, 2T and so on. But its amplitude can be a continuum. Then you feed it to an A to D and what you get at the output is a truly digital signal for which we give the symbol x(n); the subscript a is lost because it becomes digital and capital T in the process is also lost, in the sense that when small n becomes a variable, it can take integer values only- positive or negative. And each integer value of n in the corresponding discrete time signal corresponds to the integer multiplied by the sampling interval. So T is lost in the digital signal but if this signal was derived from an analog signal, then T has to be kept in mind. One way of interpreting x(n) is that capital T has been normalized to 1.

However, it is a beauty of DSP and Discrete Time Signal Processing (DTSP) that the mathematical techniques for handling discrete time signals and digital signals are the same. They are: difference equations, z transforms, Fourier transforms and discrete Fourier transforms. So whether the dependent variable is a continuum, like $x_a(nT)$, or of discrete amplitude, like x(n), the techniques are the same. And therefore we shall use $x_a(nT)$ and x(n) interchangeably. We shall always take that the signal has been discretized in time as well as amplitude. This sampling interval T is obviously the reciprocal of the sampling frequency f_s. We shall see later that our signal processing must be limited to half of the sampling frequency, we cannot go beyond that. Obviously then, digital signal is a sequence of numbers and the sequence sometimes is simply represented as x(n) where n takes values theoretically from – infinity to + infinity but at intervals of 1. To show that it is a sequence we sometimes give the symbol of x(n) within curly brackets.

(Refer Slide Time: 33:19 - 36:16)

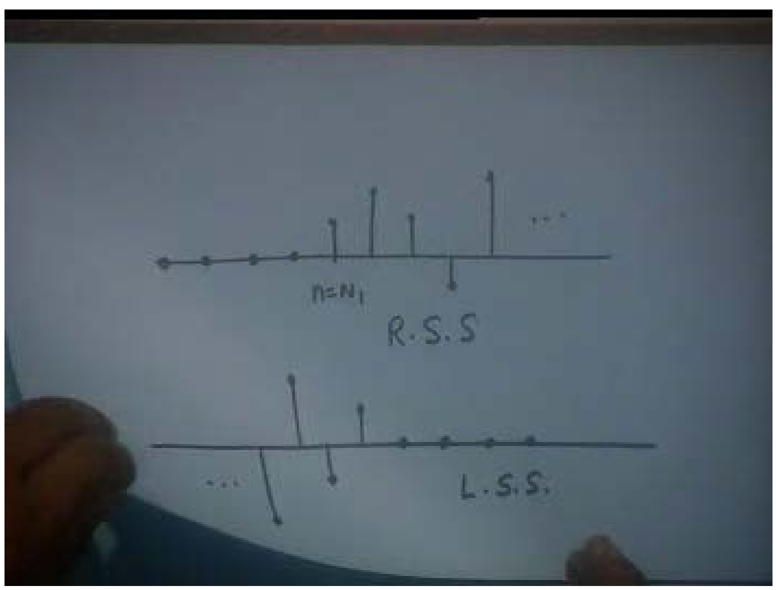


For example, we can have a signal like 0, –1, 1, 2, – 2 and so on. I am writing it in ordinary arithmetic but I have taken them to be coded, that is, quantized into integer numbers. In this sequence how is n indicated? n = 0 is indicated through an arrow and one writes n = 0 below it; in the first x(n), signal x(n) is equal to 2 at n = + 1, at n = + 2 the value is – 2, for n = – 1 it is 0, for n = – 2 it is – 1 and so on. So this is the complete description and representation of the signal.

We shall not write binary numbers here, we shall write the integer equivalent of the number. This sequence is not necessarily real; it can be complex. What you see is a real sequence, but we can have a sequence like 1 + j, 2 – j 2 and so on. For example, if you consider an analytic signal, it has a real part and an imaginary part; it simply means that there are two real signals which are orthogonal to each other.

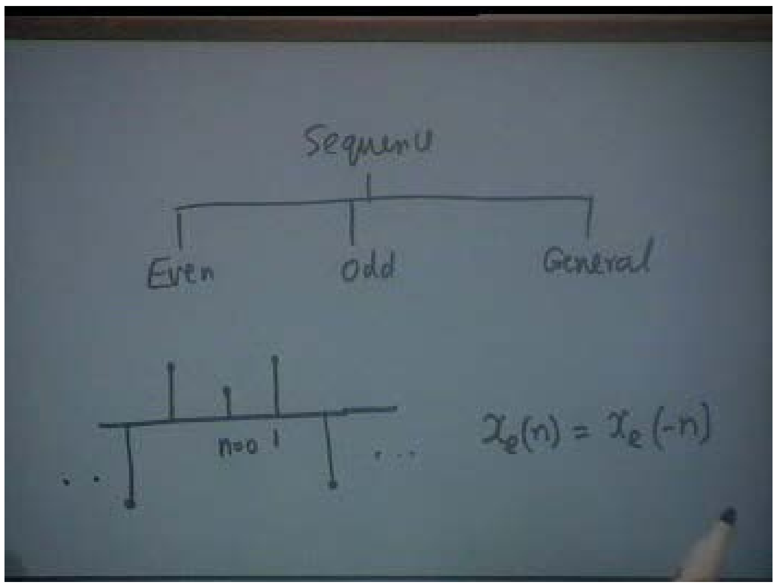
Therefore a digital signal can be real or complex. The length of the sequence is the total number of samples in the sequence. If n goes from – infinity to + infinity, then obviously it is an infinite signal. But suppose we have a sequence 1, 2, 3, 4 with 1 at n = 0; then obviously it is a finite length sequence .The length of this sequence is equal to the number of samples in the sequence. That is, the length of this signal is 4 although n takes values 0, 1, 2, 3. If the signal extends from n = 0 to N – 1, the length is N because there is a sample at n = 0.

(Refer Slide Time: 36:25 - 37:58)



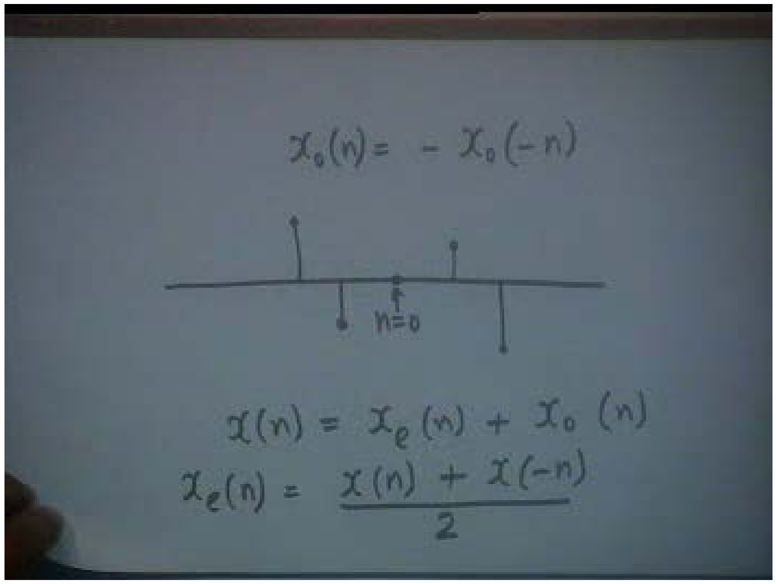
Consider a sequence that exists from some value of n, say n = N_1, and then goes to the right up to any value of n, but on the left this sequence is all 0, where N_1 can be anything like 0, –1, or +10. This signal exists to the right only and therefore it is called a right sided signal. On the other hand, if this signal exists only on the left side and is 0 on the right side, then this is called a left sided signal. A signal in general can be combination of left sided and right sided signals. If it is of finite length, then we can always view it as right sided or left sided. If it is of infinite length, then obviously it is a combination of right sided and left sided signals. A signal or a sequence can be even or odd or a mixture of even and odd.

(Refer Slide Time: 38:09 - 39:12)



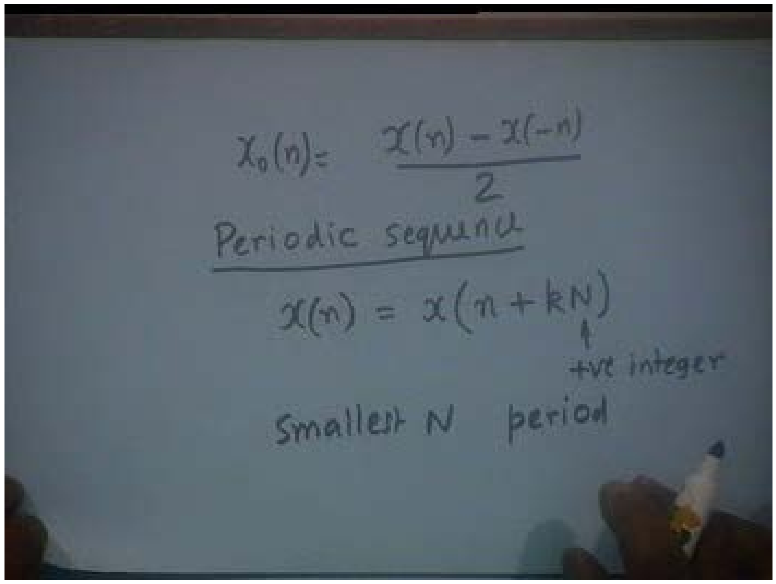
An even sequence is one whose left side is the mirror image of the right side. If centering at n = 0, you have the same amplitude at n = 1 and n = – 1, same amplitude at n = 2 and n = – 2, and so on, then it is an even signal. In other words, an even signal $x_e(n)$ is the same as $x_e(-n)$. On the other hand, an odd signal is one in which the signal $x_o(n)$ is the negative of $x_o(-n)$.

(Refer Slide Time: 39:23 - 41:08)



For an old signal, what should be the value at n = 0? Because $x_o(0)$ should be negative of $x_0(-0)$, it means that $x_o(0)$ must be 0. An even signal can have any value at n = 0, while an odd signal must have a 0 value at n = 0. In general, a signal is neither even nor odd but it can be decomposed into two parts an even part $x_e(n)$ and an odd part $x_o(n)$. And if you want to find out the even part, it is very easy. $x_e(n)$ is just half of x(n) + x(– n). When you add the two, the odd parts will cancel. In a similar manner, the odd part is given by $x_0(n)$ = half of [x(n) – x(– n)]. When you subtract x(– n) from x(n), the even parts cancel.

(Refer Slide Time: 41:13 - 43:33)

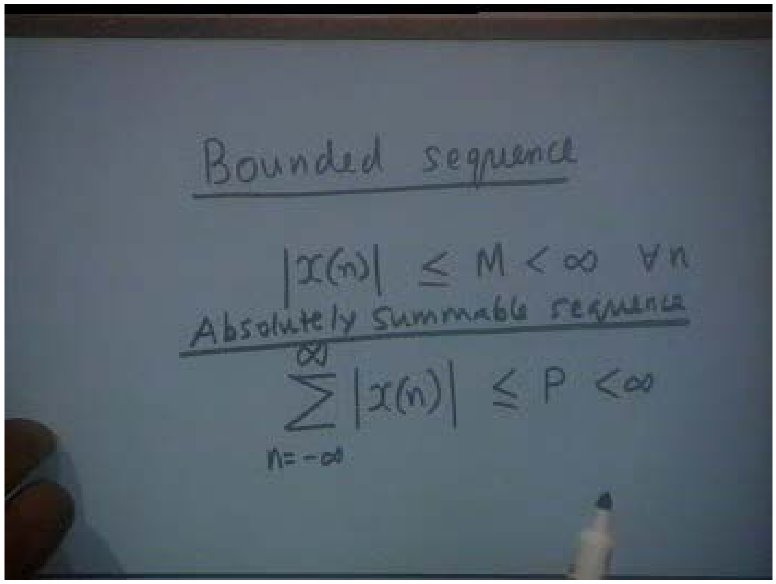


There are quite a few problems in the book on decomposition of signals into even and odd parts; try to work out as many as you can.

We introduce now a periodic sequence or periodic digital signal. A signal x(n) is said to be periodic if x(n) = x(n + k N) where N is the smallest positive integer; it means that x(n) repeats after every N samples. And the smallest N for the validity of this relation is called the period. Therefore the period has no dimension, it is not in seconds or micro seconds, it is a pure number. That is, it indicates the number of samples after which the sequence repeats.

Why is it the smallest N? Suppose 20 is the period then repetition will occur after 40 samples, 60 samples and so on, but the repetition will not occur for 10. So the smallest value of N for which this relationship is valid is called the period of the periodic signal or sequence. We will show a little later that while cos(ωt) or sin(ωt) is always periodic, this is not true about $\cos(\omega_0 n + \varphi)$ or $\sin(\omega_1 n + \varphi)$.

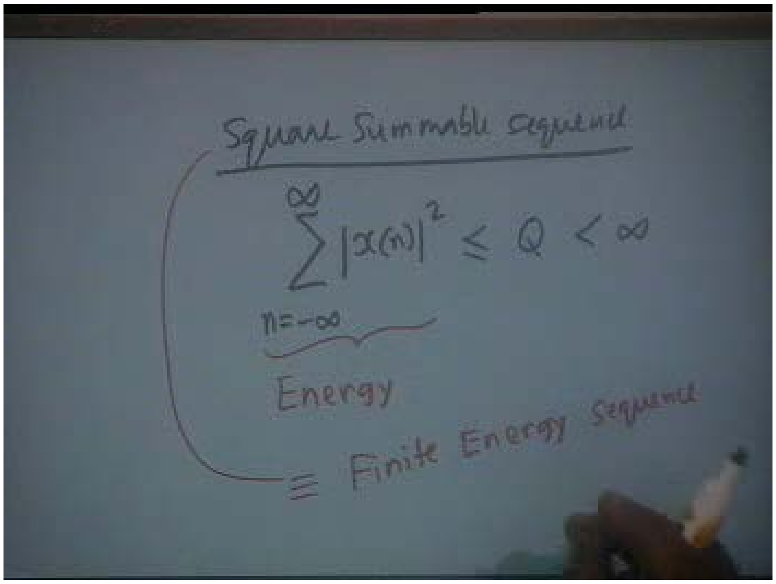
(Refer Slide Time: 44:01 - 45:26)



We now define what a bounded sequence is. A sequence x(n), which may be real (positive or negative) or complex, whose magnitude is bounded, i.e. less than or equal to some number M, which is less than infinity, is said to be a bounded sequence.

If the summation of magnitude of x(n) over its total length, in general from – infinity to + infinity, is bounded, i.e. less than equal to some number P, which is less than infinity, then x(n) is said to be an absolutely summable sequence. You can similarly define a square summable sequence.

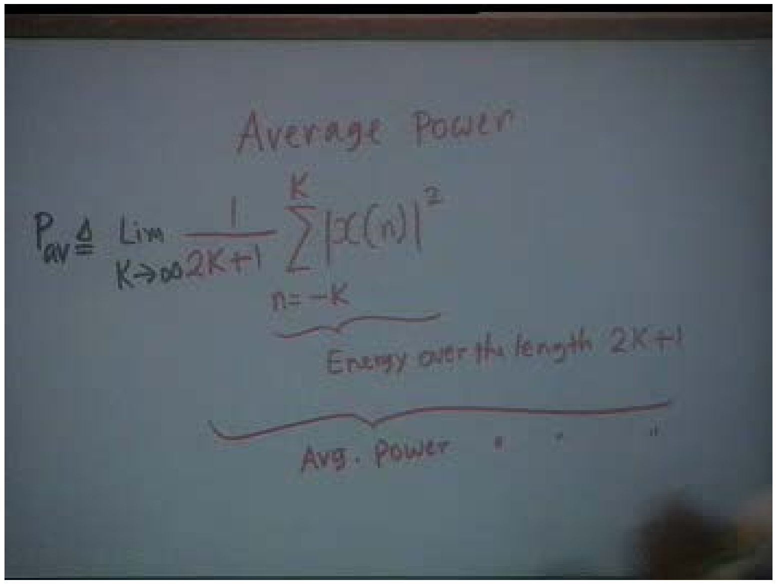
(Refer Slide Time: 45:35 - 47:23)



A square summable sequence is one in which the magnitude squared summed up from n = – infinity to + infinity, is less than or equal to some number Q, which is less than infinity. Suppose you have a sequence 1/n, n = 1 to ∞. Is it absolutely summable? No, because the summation n = 1 to ∞ |x(n)| does not converge. Is it square summable? Yes, it is. You notice that this represents the energy of the sequence, just as integral of the square of voltage across a 1 ohm resistance is energy, or integral of the square of current through a 1 ohm resistance is also energy.

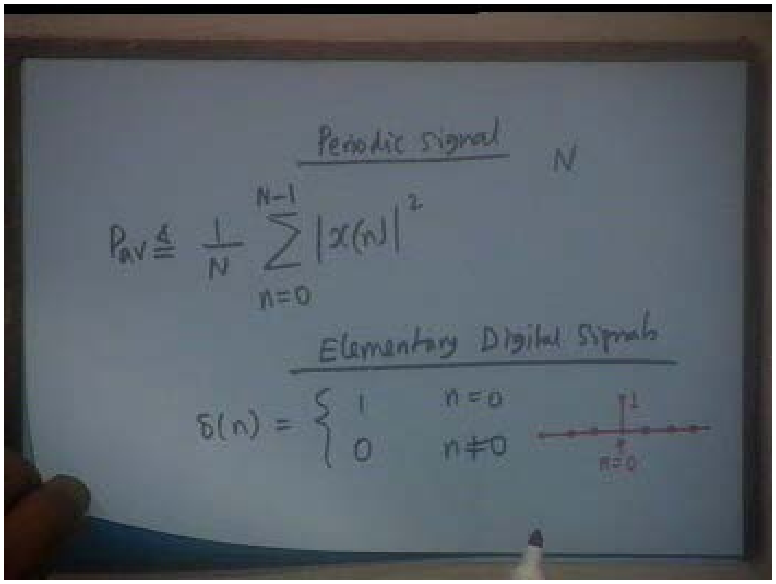
The energy of a sequence is defined as |x(n)| the whole square summed up over all the samples. So a square summable sequence can also be called, alternatively, a finite energy sequence. A sequence may not be square summable, just as an analog signal may not be an energy signal.

(Refer Slide Time: 47:38 - 49:28)



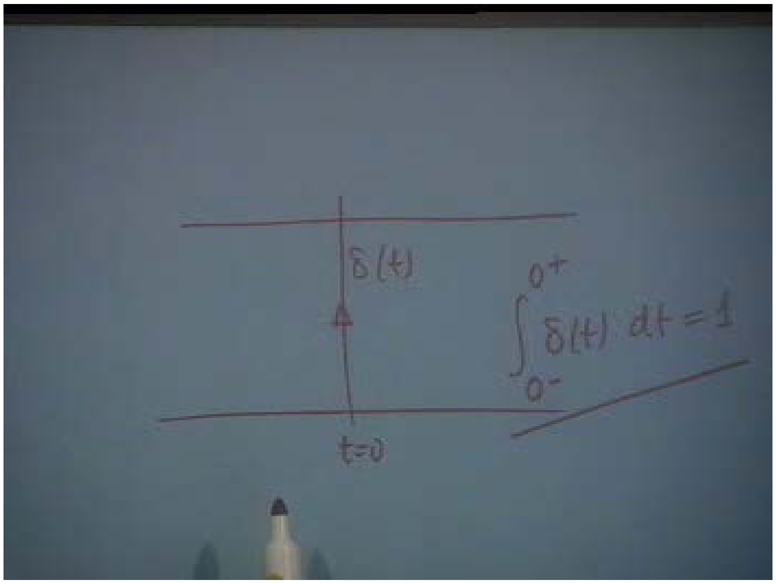
We now define average power. We have defined what energy is. In analog domain, power is energy per unit time. Similarity, if we find out the energy of a sequence over a certain length, say from n = – K to + K, then this represents energy over the length, 2K + 1. If we divide this by the length 2K + 1, then this will represent average power over the same length. Now we allow K to go to infinity so that we cover the total length of the signal, so then we would get the average power of the signal over its total length because it is a limiting process, it may not be possible to go on adding for infinite length sequences. So what you do is to take sufficiently large value of K, then increment it by 1, 2, 3 and so on and see whether the average power changes. Usually the sequences trail off at very large values of n. Therefore average power is a meaningful definition. If you have a periodic signal, then your job is very simple.

(Refer Slide Time: 49:36 - 51:29)



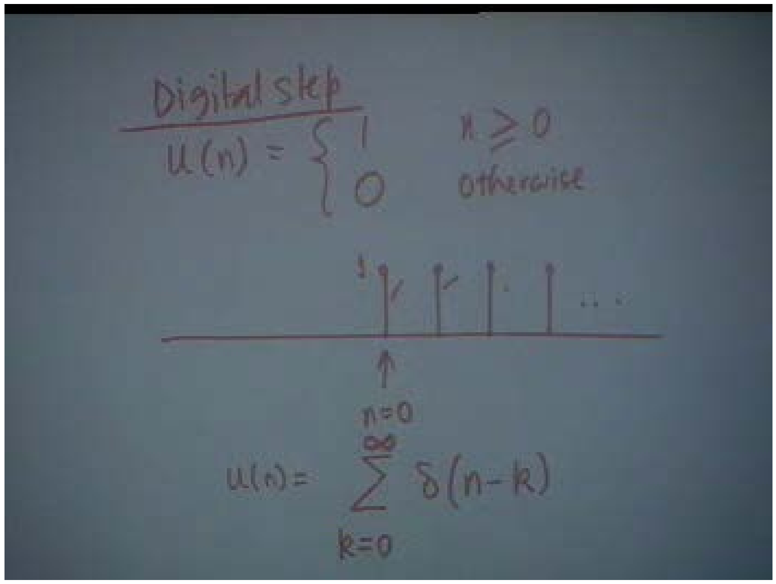
For a periodic signal of period N, all you have to do is to take the average over one period, because every period is identical replica of any other period. What you do is to sum up |x(n)| the whole square from n = 0 to N – 1 (because at n = N it starts repeating) and then divide by N. So this is the average power of a periodic signal. We have defined absolutely summable sequence, square summable sequence which is the same as finite energy sequence, and then average power in a sequence. Now look at some typical elementary digital signals. The most elementary signal is the δ(n) and it is defined as one for n = 0, and 0 for n not equal to 0. In other words its diagram shall be just one sample of amplitude 1 occurring at n = 0 and it is 0 on either side; this is the most elementary signal that one can think of and is called a digital impulse. In comparison, the analog impulse δ(t) is quite complicated. How is δ (t) defined?

(Refer Slide Time: 51:43 - 52:22)



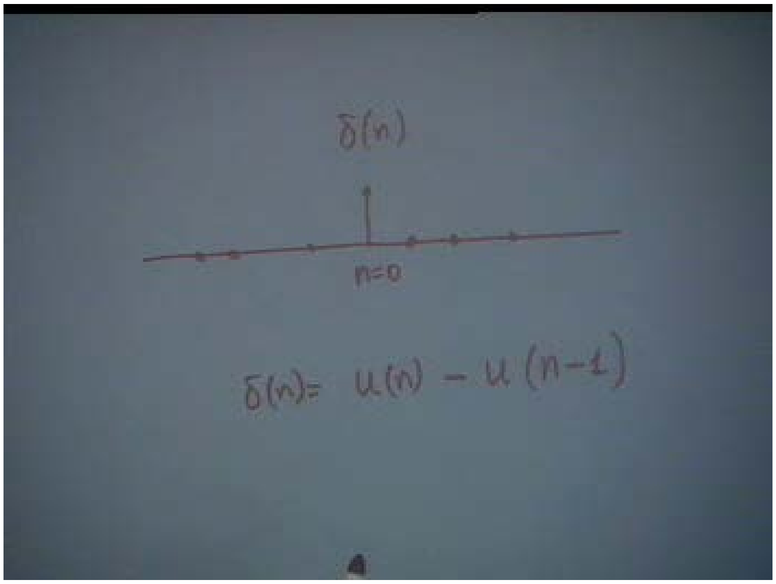
It exists at t = 0 only and the amplitude there is infinitely large which does not mean anything. You must define infinity otherwise the function will be useless for us; so δ(t) is defined by ∫ δ(t) dt from 0 – to 0 +, that is the area under the curve, is 1. δ(t) is not a function in the ordinary sense, it is an unlimited function whose amplitude cannot be defined. δ(t) can be defined only in terms of the area. However there is no such problem in a digital signal; δ(n) is 1 at n = 0 and 0 everywhere else.

(Refer Slide Time: 52:44 - 53:46)



A combination of such impulse functions give rise to a digital unit step u(n); u(n) is a signal which is 1 for n greater than or equal to 0 and 0 otherwise. Obviously u(n) can be written in terms of δ(n); it is summation of δ(n-k), k going from 0 to infinity. δ(n-1) describes an impulse with one sample delay, δ(n-2) is an impulse with two samples delay, and so on. So u(n) is very simply related to δ(n – k). δ(n) can also be related to u(n).

(Refer Slide Time: 53:58 - 54:40)



δ(n) is simply one signal of amplitude 1 at n = 0 and the rest all are 0. Obviously, for δ(n), you subtract from u(n) all that occurs to the right of n = 0, which is simply u(n – 1). Therefore δ(n) can be replaced by u(n) – u(n – 1). I think our time has come to an end and therefore we will stop here today.