Digital Signal Processing

Prof. C.S. Ramalingam
Department Electrical Engineering
Indian Institute of Technology, Madras

Lecture 03:
Signal Symmetry, Elementary Signals (1)
Recap of affine transform

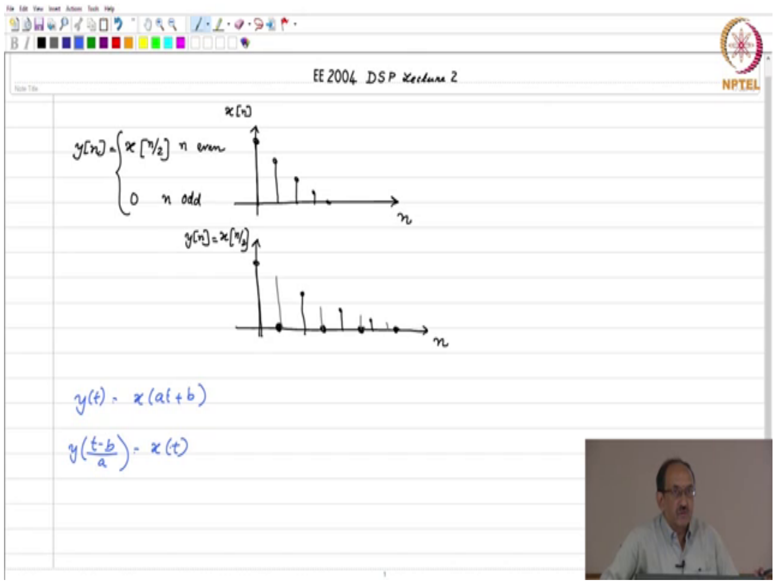
Keywords: elementary signals, affine transform

Let us get started for today's lecture. Yesterday, we got introduced to the basic signal definition and we saw that signal in our context means a function that takes on scalar values, the independent variable is also a scalar. In the continuous-time case, we call the independent variable as t to denote time and in the discrete-time case, the index was n, still we call it as time and n takes on values in the set of integers, $\mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$. Whereas in the continuous-time case, t takes on all possible values between $-\infty$ to ∞. Then we briefly saw classifications of signals. I have expanded more about this in my notes and then we started to look at operations on the independent variable.

And even though lot of manipulations are possible, the class that we are interested in is the affine transform. That is we saw in the continuous-time case, modifications of the form $at + b$ and in the case of discrete-time counterpart, the affine transform took on the equation $Mn + N$ and then this kind of led us to the first difference between continuous-time case and the discrete-time case. In the continuous-time case, you can undo the affine transform, whereas in the discrete-time case, you cannot undo the affine transform in general.

And the example that we looked at was we looked at $y[n] = x[2n]$ and we saw two examples in which two different sequences that differed only in the odd indices gave rise to the exactly the same result after picking every other sample, the even indexed samples. And expansion was, the example that we saw was $y[n] = x\left[\frac{n}{2}\right]$ and again this led to differences because if you look at $\frac{n}{2}$ when n is odd, the sequence at those values are not defined.

(Refer Slide Time: 02:43)



So, the example that we left with towards the end of last class was $y[n] = \begin{cases} x\left[\dfrac{n}{2}\right] & n \text{ even} \\ 0 & n \text{ odd} \end{cases}$ and the sequence $x[n]$ that we started off with was something like this. And then $y[n]$ which was $x\left[\dfrac{n}{2}\right]$, I am drawing a rough sketch here and at these indices, things are not defined. You can supplement this definition with it being 0 for n odd and this is for n even in which case you can go ahead and make this as 0 values.

But really we would want something like this and towards the end of last class I mentioned that more processing is needed to get something like this starting from this definition. Starting from this to get something that is similar to what happens in continuous-time, more processing is needed and why this difference comes about is easy to see. In the continuous-time case, when you have $y(t) = x(at + b)$ and $y\left(\dfrac{t-b}{a}\right) = x(t)$ i.e., absolutely no loss of information. Whereas in the discrete-time case, when you have say $y[n] = x[2n]$ you are dropping every odd indexed sample values, you are just throwing them away.

So, because of this you cannot get back the samples that you have discarded given the very nature of the transformation. In this context, it is worth mentioning that under certain conditions you can recover the original signal. You must have already studied about sampling from the previous course. Suppose you sample the signal that is much more than the required rate.

So, if you over sample the signal say by a factor of 2, then it stands to reason that you can down sample by a factor of 2 and yet not lose any information. So, it is not that down sampling always leads to loss of information under certain conditions it is possible to recover the original sequence, but in general this is not possible.