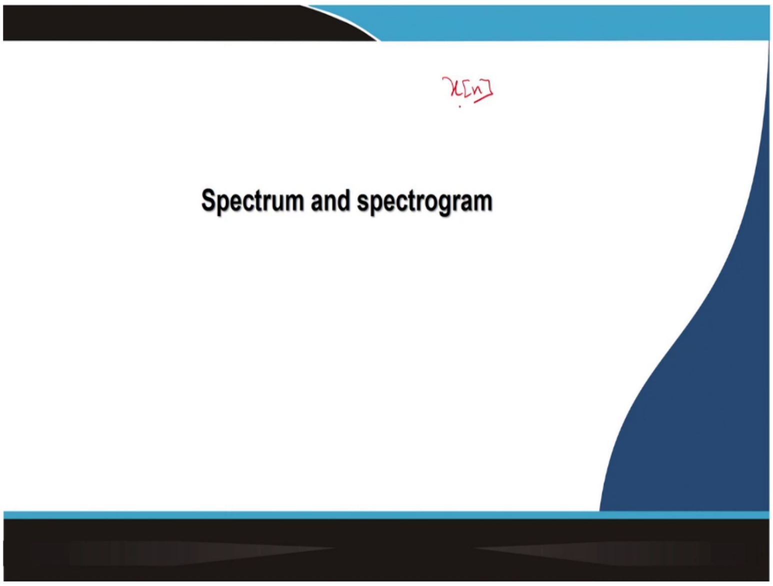
Signal Processing Techniques and Its Applications
Dr. Shyamal Kumar Das Mandal
Advanced Technology Development Centre
Indian Institute of Technology, Kharagpur

Lecture - 34
Spectrum and spectrogram

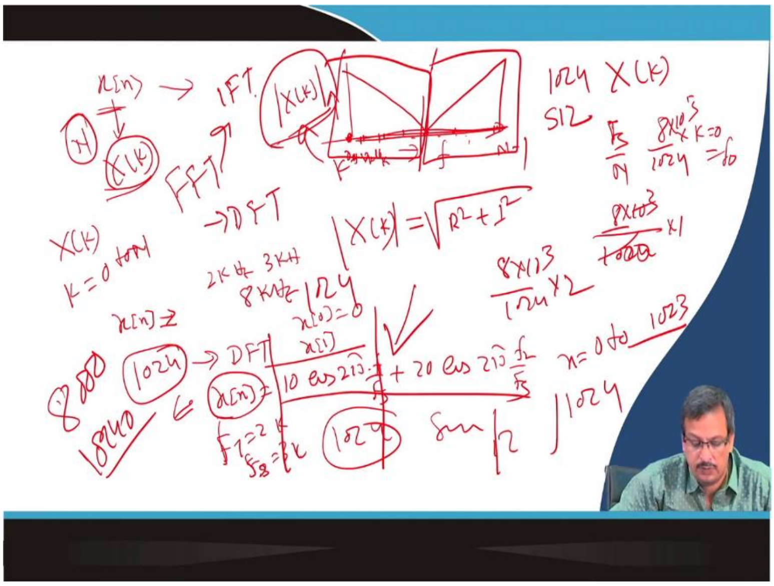
So, I have discussed the radix 2 algorithm and divide and conquer method for implementing discrete Fourier transform and using an FFT algorithm. Let us say you know how to implement the FFT algorithm.

(Refer Slide Time: 00:37)



So, I know how to implement the FFT algorithm, then what can I do? Let us see a simple application that supposes I have a signal x[n]. Let us say I take the next slide.

(Refer Slide Time: 00:51)



So, I have a signal x[n], and I want to draw the signal's spectrum. So, what is spectrum? The spectrum of the signal is nothing, but when I plot this axis, it will be the frequency, and this axis will be the amplitude of individual components. So, let us convert x[n] to X(k) using the FFT algorithm.

So, x[n] is a discrete signal. I used discrete Fourier transform implemented in using the FFT algorithm, which I converted to X(k) with N point DFT. So, once I do that, I get X(k), k varies from 0 to N. I want to draw the plot of the frequency. This x-axis will be the frequency, and the axis will be the mod of X(k).

You have to do this by hand. So, what I said? Let us say generate a signal x[n] which has a frequency component of the cosine frequency component; let us say 2 kilohertz and 3 kilohertz sampled with 8 kilohertz. Generate, let us say, 1024 samples, compute discrete Fourier transform using the FFT algorithm, then plot the mod of X(k) versus frequency or spectrum of the signal.

So, how do you do that? So, I will do what else I have to generate. I can write down

$$x[n] = 10\cos\left(2\pi\frac{f_1}{F_s}\right) + 20\cos\left(2\pi\frac{f_2}{F_s}\right)$$

So, f1 is equal to 2 kilohertz f2 is equal to 3 kilohertz I can get that x[n]. So, n varies from 0 to n varies from 0 to 1023. So, n is equal to 0, x[n] is equal to 0, n is equal to 1, I calculate program will calculate, n equal to 2, program will calculate.

So, I collect that signal; then, I compute n point DFT using the FFT algorithm. I get X(k). Now, I have to draw, so X(k) is a complex number. So, what do I get? I get the real part and the imaginary part. So, the mod of X(k) is nothing but a real part square plus an imaginary part square.

So, I can easily do that X(k) is a complex number. I can use a real square plus an imaginary square and get the mod of X(k). Now, how do I get the frequency? So, if I draw k versus a mod of X(k), I know k, k equal to 0 means this one, and k equal to 1 2 3 4 up to n minus 1 I will get.

But k is a discrete frequency, but I want to draw against analogue frequency. So, how do I convert k to f? It is nothing but a from the resolution. So, I know what the resolution of this conversion is. Fs by N. What is the Fs? 8 kilohertz.

So, 8 kilohertz by 1024 is my resolution. So, when I multiply with k equal to 0, then it gives me the f0 first frequency. The second frequency is 8 kilo 8 into 10^3 8 into 10^3 multiplied by 1024 into 1.

Third, 8 into 10^3 divided by 1024 into 2. Let us say it is considered approximately 10, so 8 hertz. So, the first component is 0 hertz, the second component is 8 hertz, and the third component is 16 hertz, so that way, I will get.

Now, as you know, the discrete Fourier transform is in symmetry. So, whatever I will get n by 2 looks like this. So, at n by 2, instead of 1024 points, if I compute X(k) up to 512 points, then I get this plot, and that is my spectrum, and this is nothing but a mirror image.

So, that is my spectrum. Now, see, my signal is a long signal. So, here I said to generate 1 024 point signal, and the signal is stationary because if I continue, the signal does not change its colour or its property.

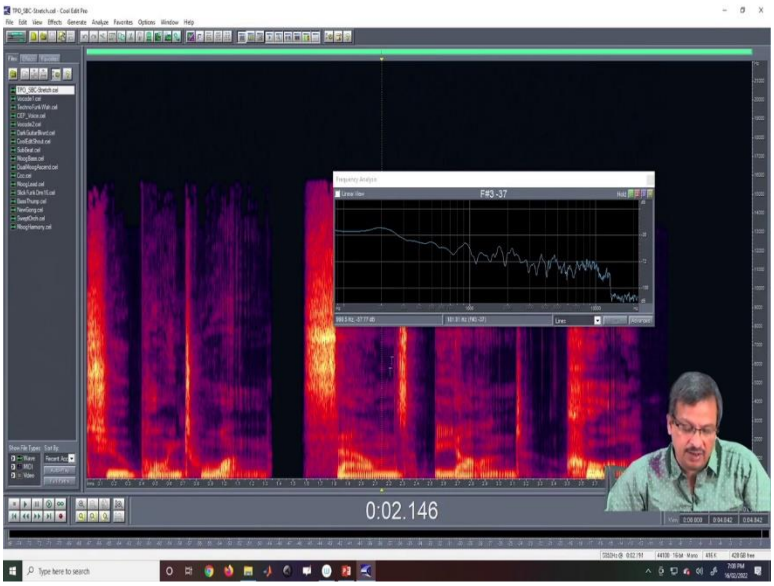
Now, suppose it is a long signal with a stationary signal long signal, and I want to process 102; let us say I generate 8000 samples or 8000 or less multiplied by 10, 10240 samples and my length of the DFT is 1024.

So, every time I cut the signal with 1024 points, if it is a stationary signal, then add an overlap. The same method can be applied when I convert to the inverse DFT ok. One important piece of information I forgot to give you is that the FFT algorithm is applied for discrete Fourier transform; this is also applied for inverse Fourier transform.

Because the algorithm is the same, it is applied not only to the Fourier transform for the discrete Fourier transform, but it can be applied to the inverse Fourier transform also. So, the algorithm will be the same, only the w n instead of minus, it will be plus or normalization factor 1 by n is added up ok.

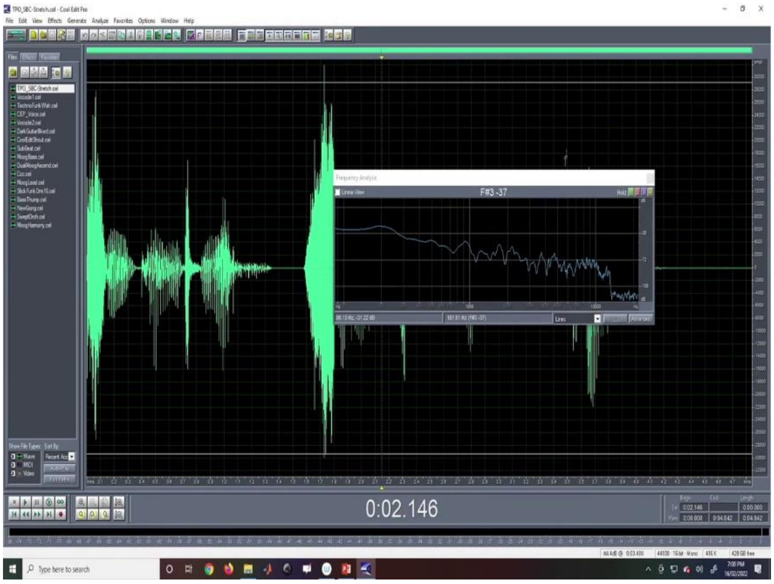
Similarly, I know it is such a long signal. Now, if it is a non-stationary signal, suppose I give you an example. Let us say I will show you a demo of a cool edit software.

(Refer Slide Time: 07:40)



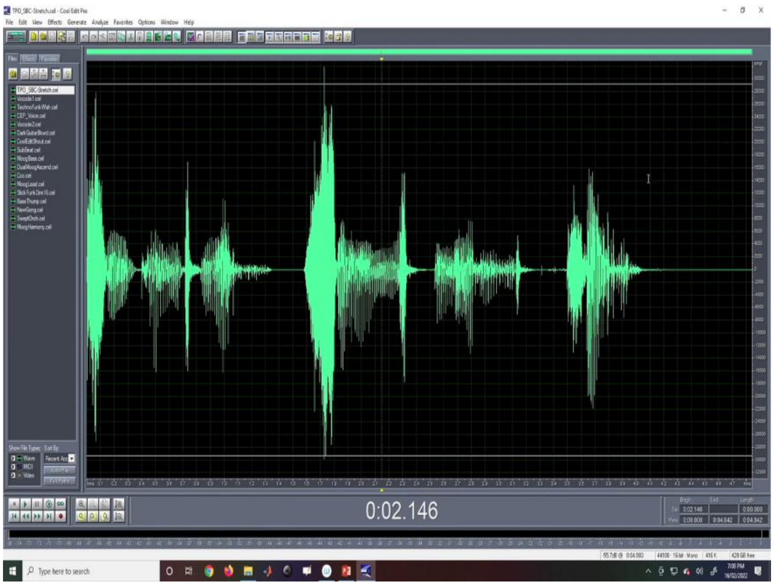
So, I have a signal.

(Refer Slide Time: 07:42)



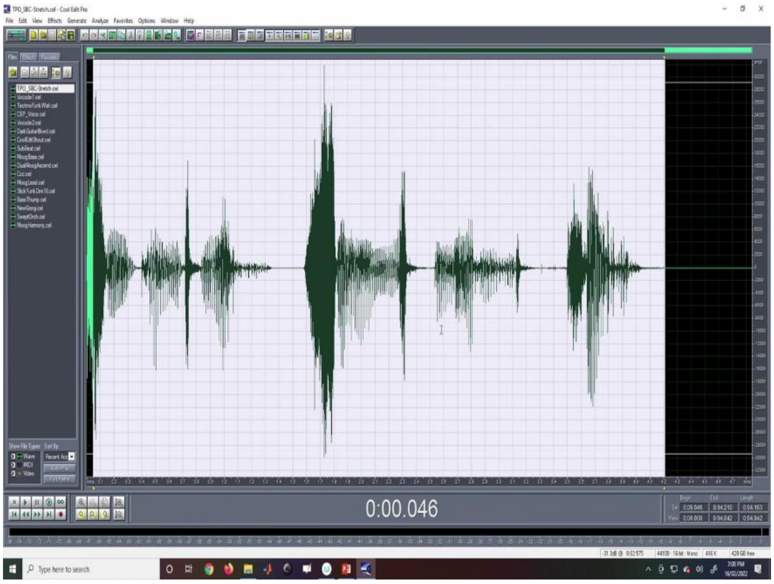
So, this is my speech signal.

(Refer Slide Time: 07:45)



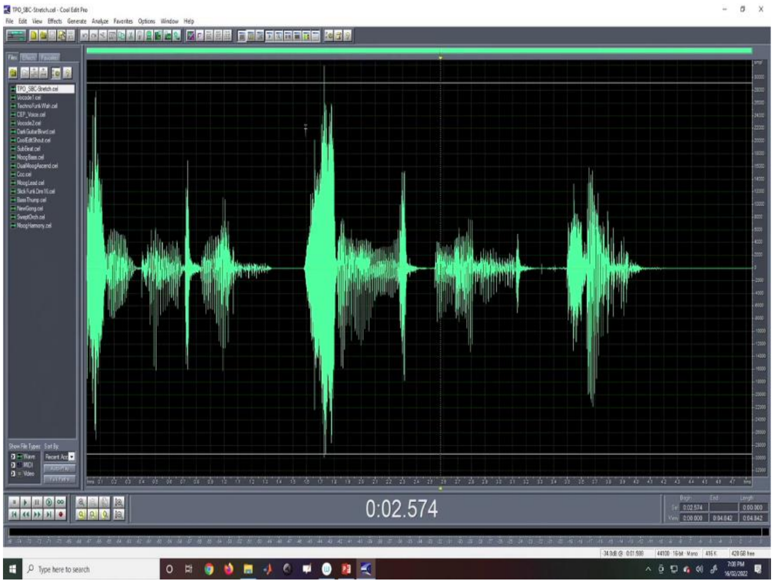
Let us say this is my speech signal.

(Refer Slide Time: 07:48)

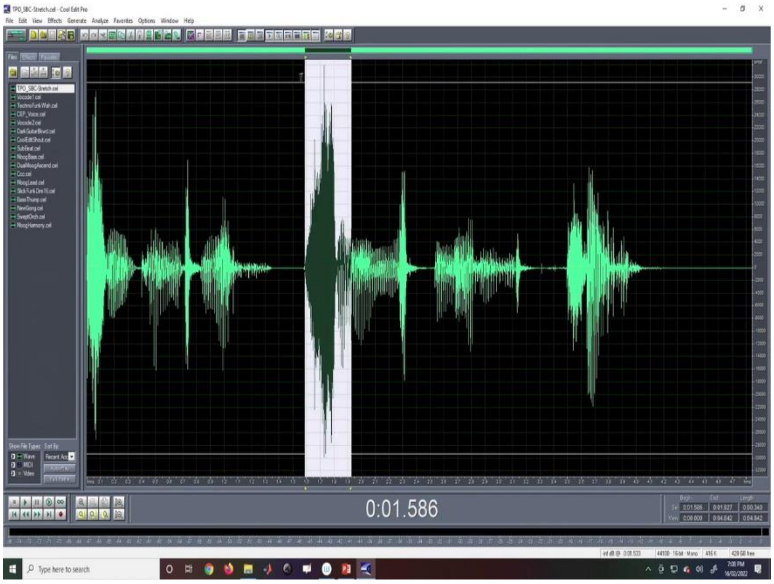


If you see, this is a speech signal I have recorded.

(Refer Slide Time: 07:50)

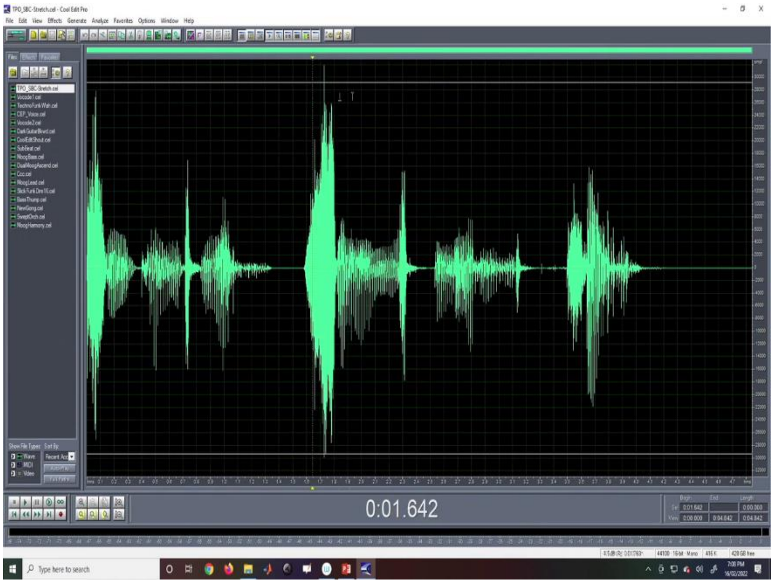


(Refer Slide Time: 07:51)



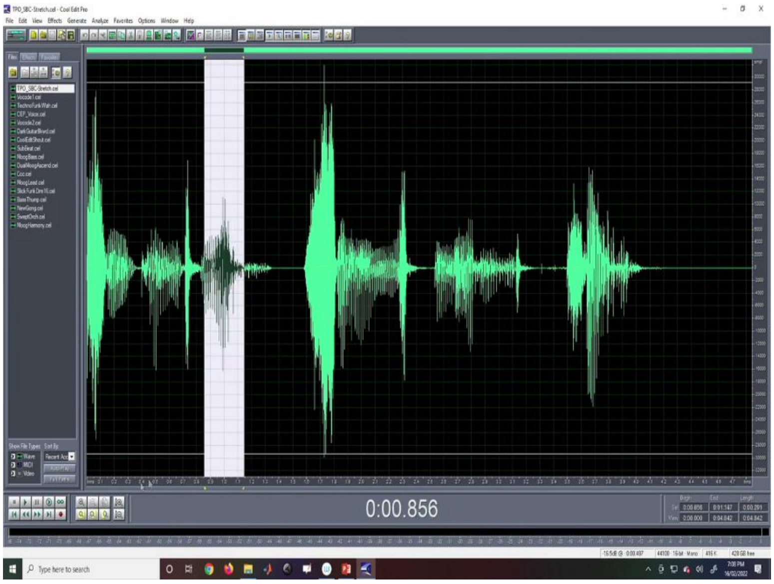
Although the signal is clipped.

(Refer Slide Time: 07:52)



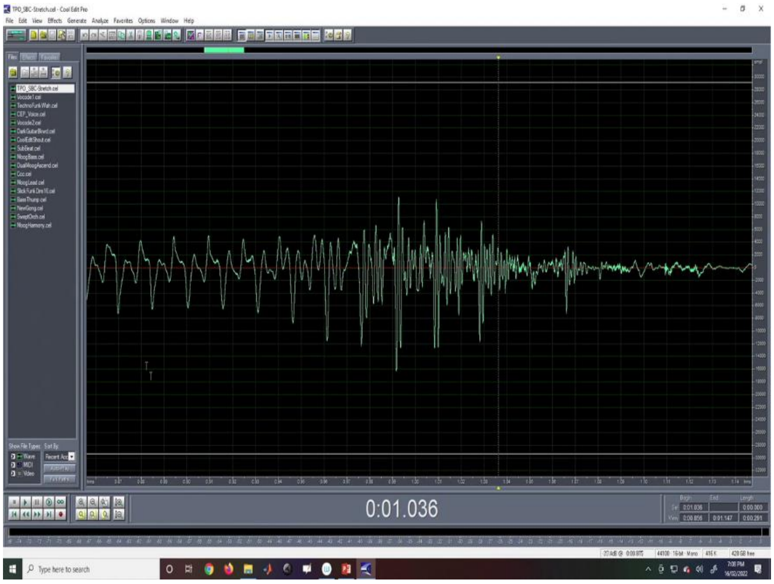
So, I am not going into detail on that.

(Refer Slide Time: 07:55)



Let us see this portion of the signal.

(Refer Slide Time: 07:57)



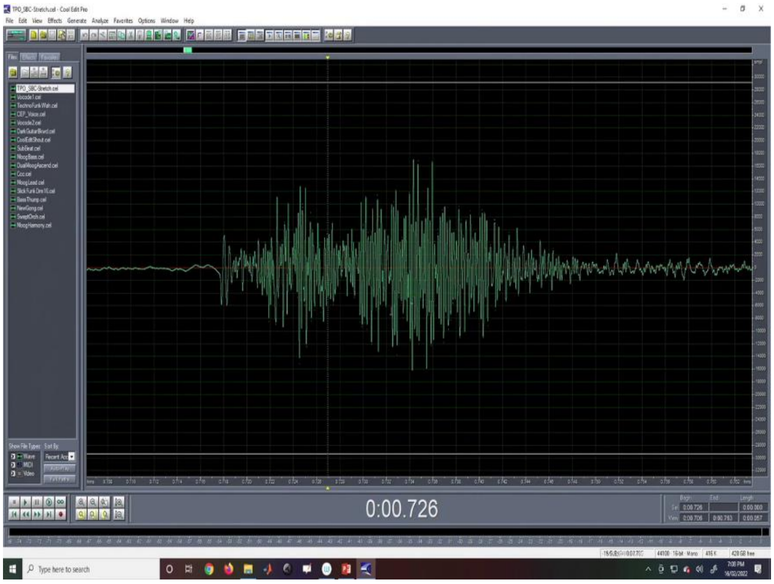
I can zoom it. So, this is my signal.

(Refer Slide Time: 08:03)

If you see the signal is non-stationary, the property of the signal is changing.

(Refer Slide Time: 08:11)

(Refer Slide Time: 08:15)

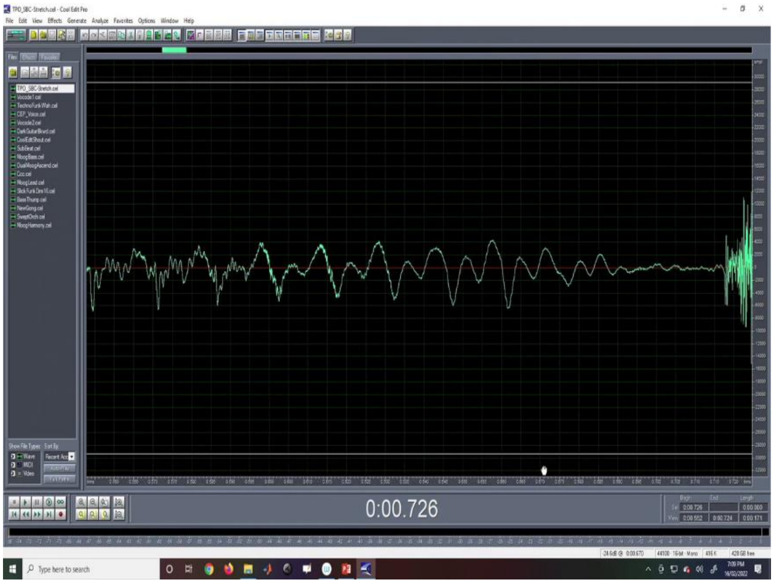


This time, it is noise, completely noisy kind of thing.

(Refer Slide Time: 08:16)

This time, if you see this time, it is periodic.

(Refer Slide Time: 08:18)

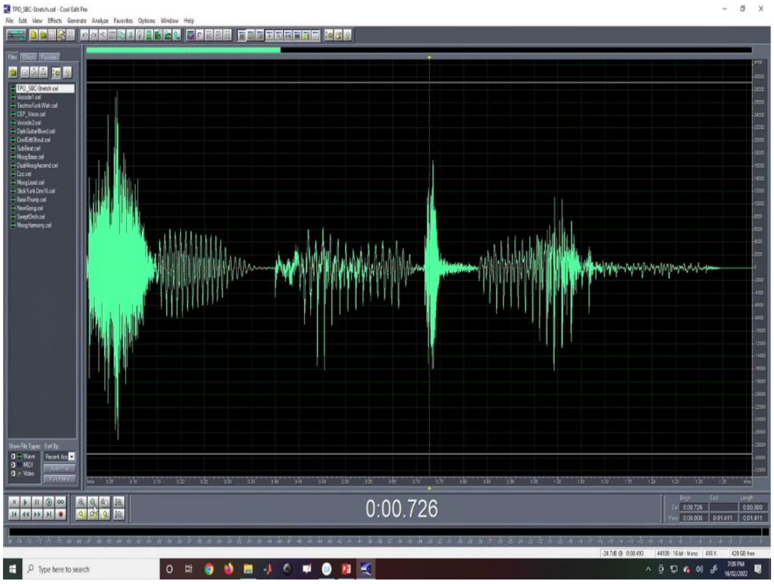


And this time periodicity also changes.

(Refer Slide Time: 08:25)

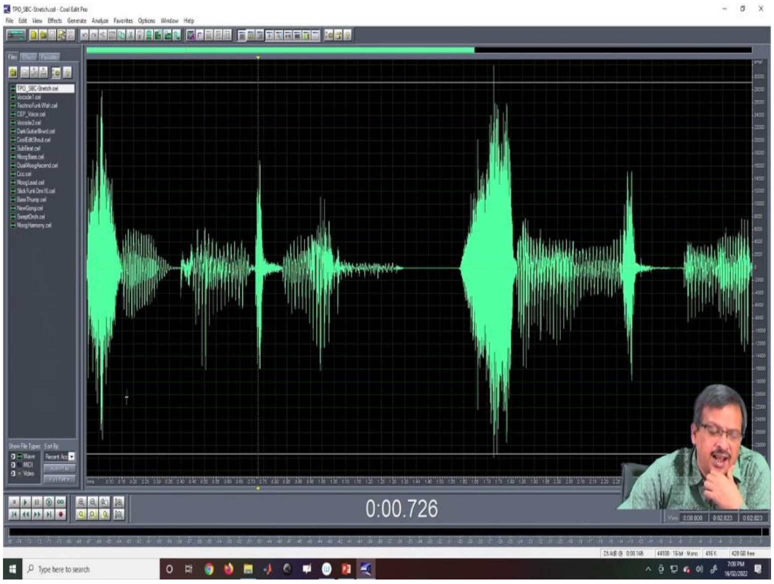
So, the signal is not time-independent.

(Refer Slide Time: 08:27)



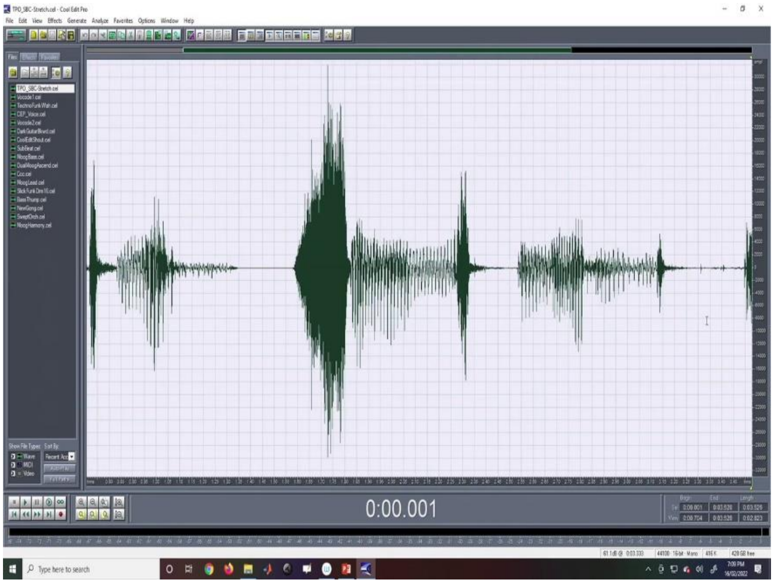
So, it is time-dependent.

(Refer Slide Time: 08:32)



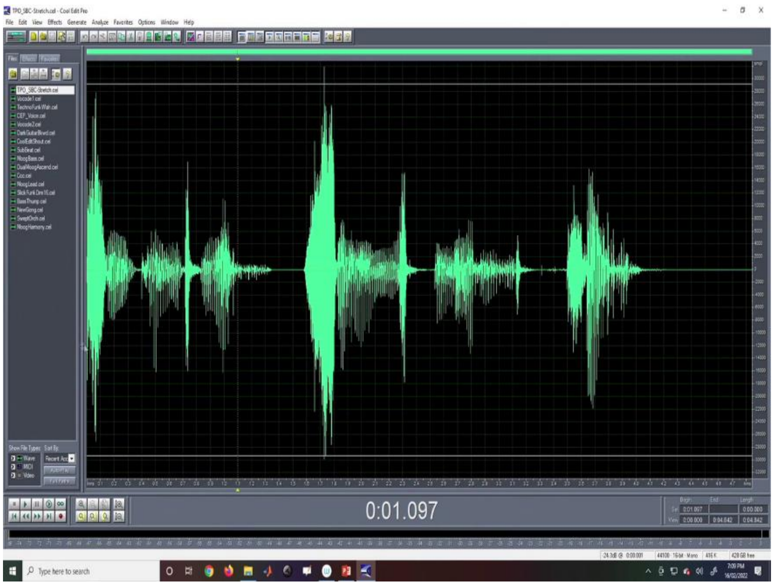
So, in that case, if I take the entire signal and take the spectrum.

(Refer Slide Time: 08:34)

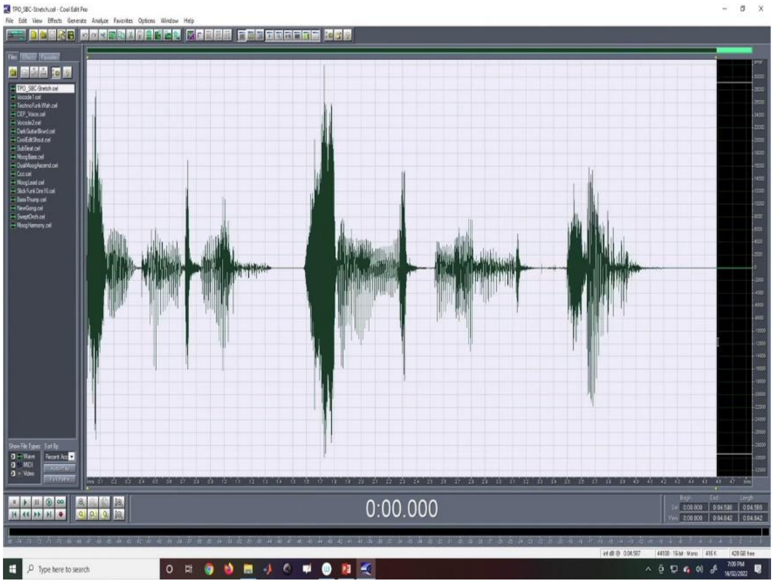


The entire signal takes the Fourier spectrum.

(Refer Slide Time: 08:37)

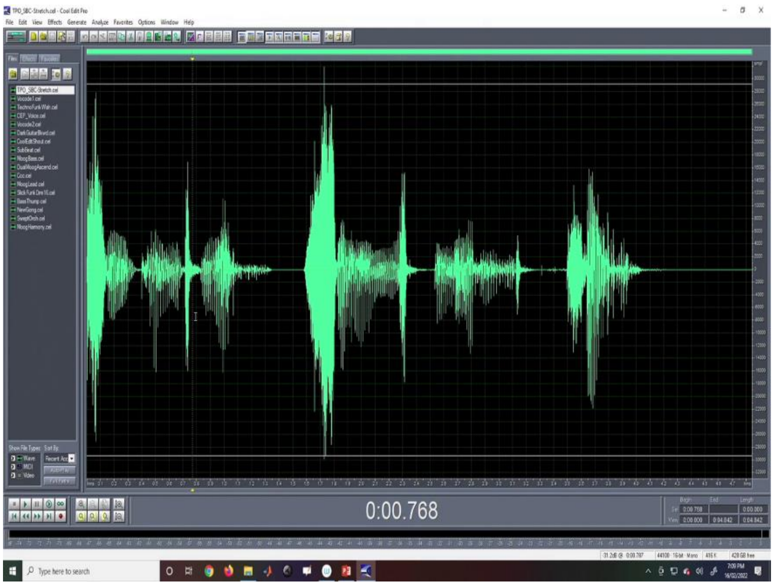


(Refer Slide Time: 08:42)

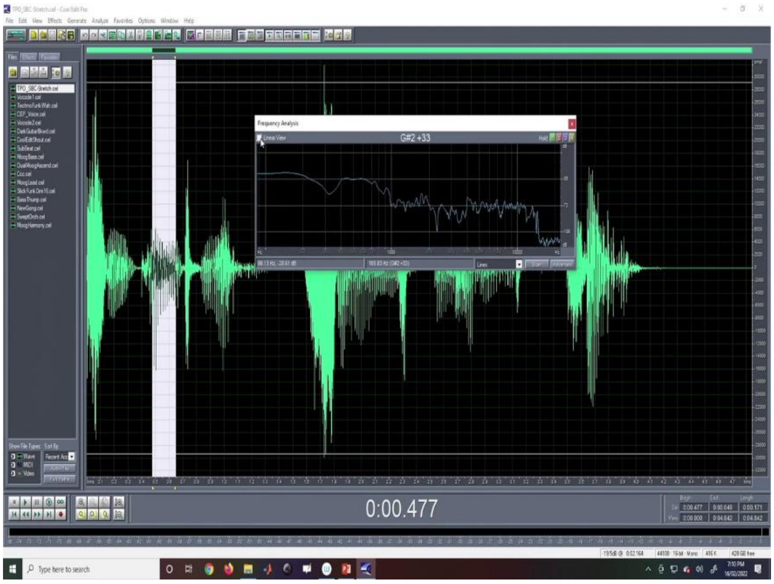


So, what will I get? I get an average representation of the signal, but that is not the correct analysis.

(Refer Slide Time: 08:50)



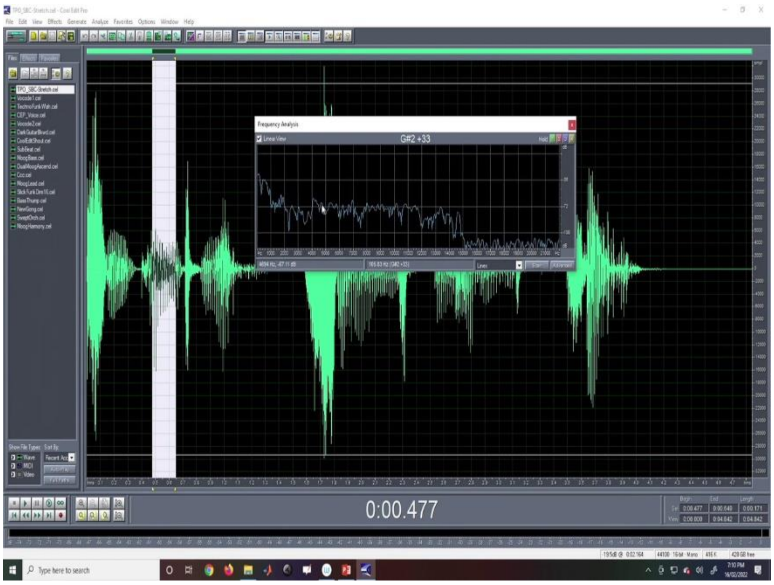
(Refer Slide Time: 08:52)



So, suppose I want to know what the spectrum representation of this portion of the signal is. So, once I select the portion, I will be doing a short Fourier transform.

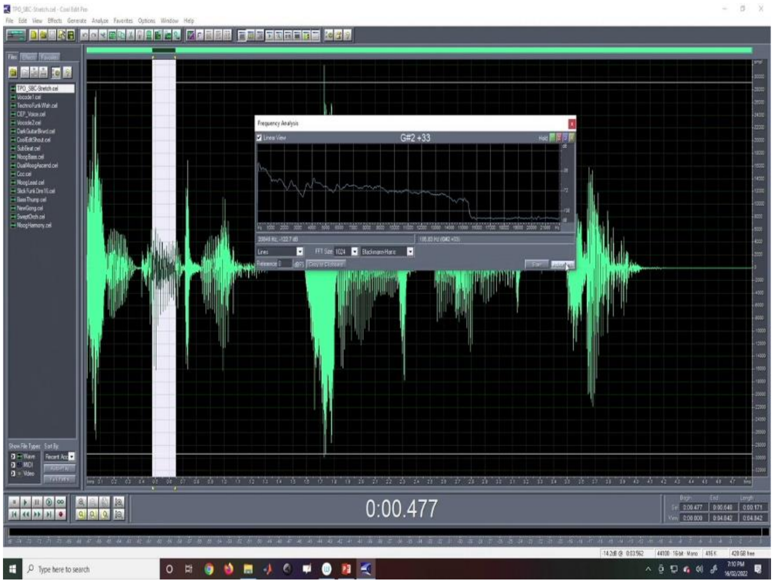
So, what comes into the picture? The window comes into the picture. So, if you look at the frequency analysis, I get it. Here, hertz, this is the linear view, and this is the non-linear log view. So, it is a linear view.

(Refer Slide Time: 09:17)



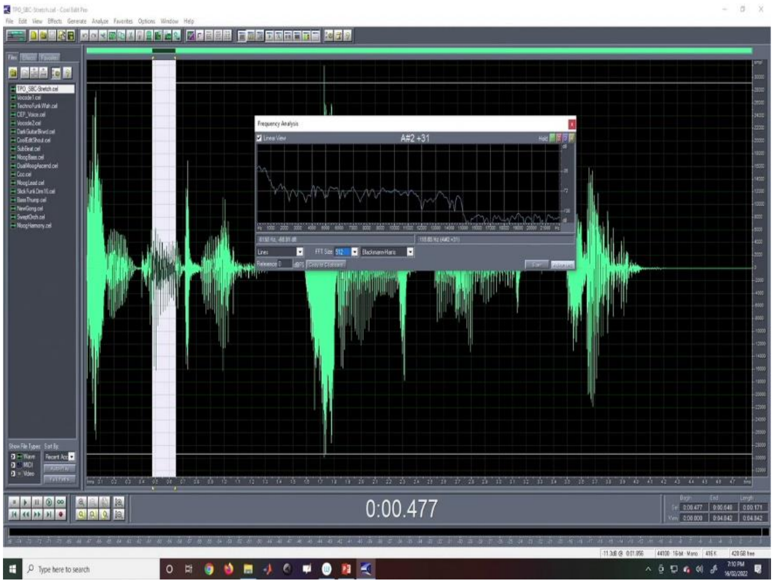
So, this is the axis of hertz, and this axis is the intensity in DB.

(Refer Slide Time: 09:32)

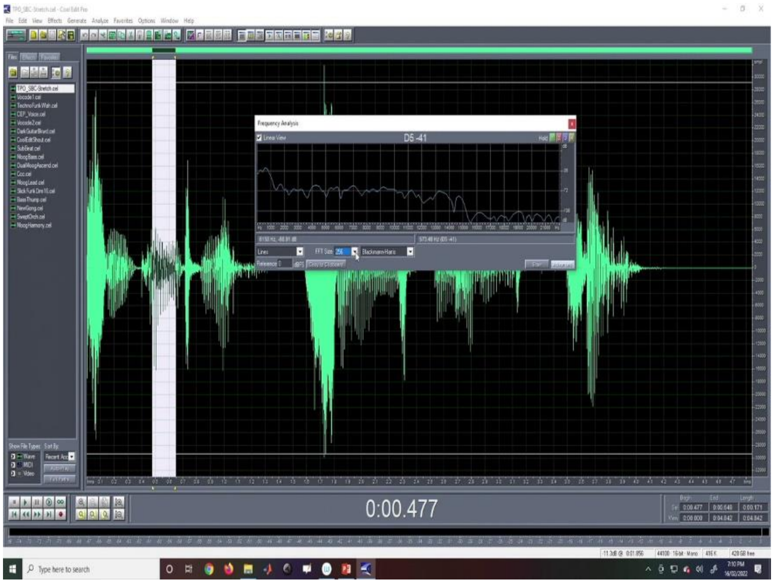


Now, when I say that, then if you see instead of a line, I can, although those things are different. Let us say what their advanced characteristics are.

(Refer Slide Time: 09:46)

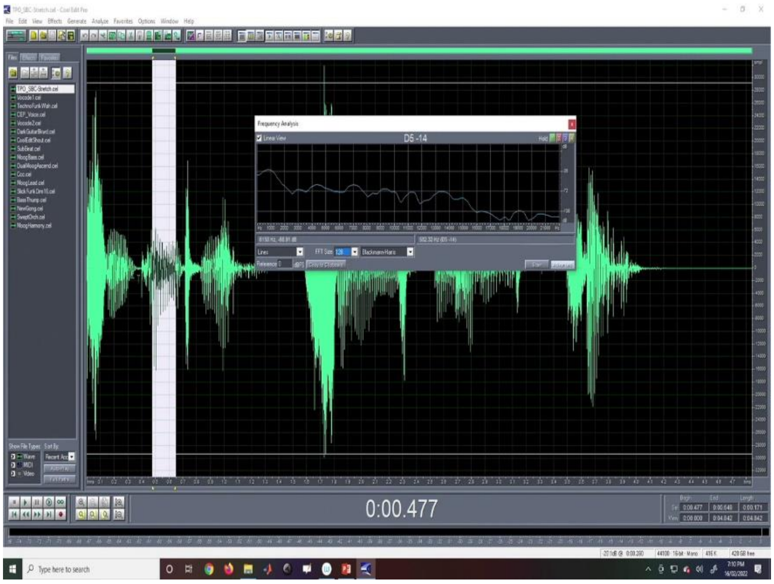


(Refer Slide Time: 09:48)



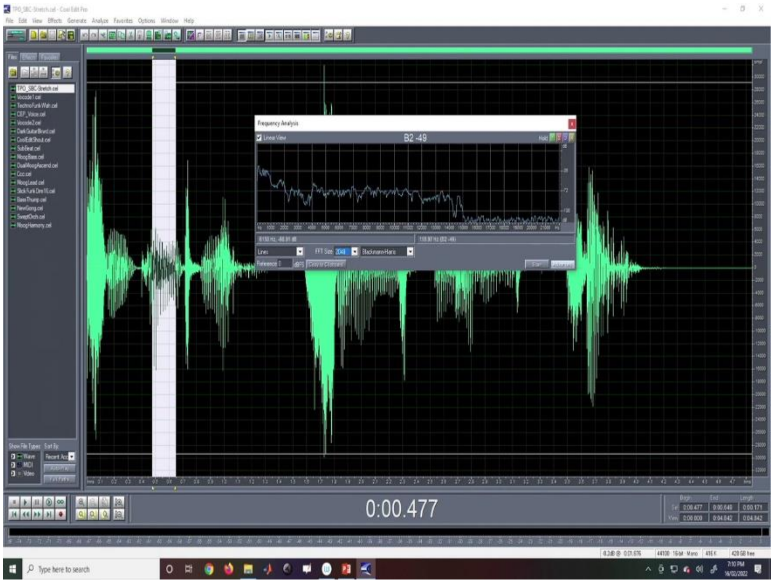
Here, if you see, I can change the FFT size.

(Refer Slide Time: 09:50)



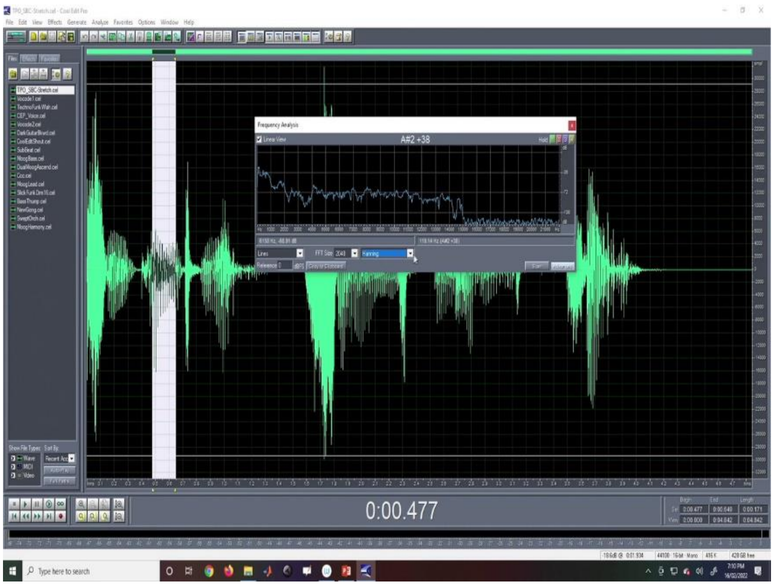
If you change the FFT size, n changes, but the signal remains the same.

(Refer Slide Time: 09:58)



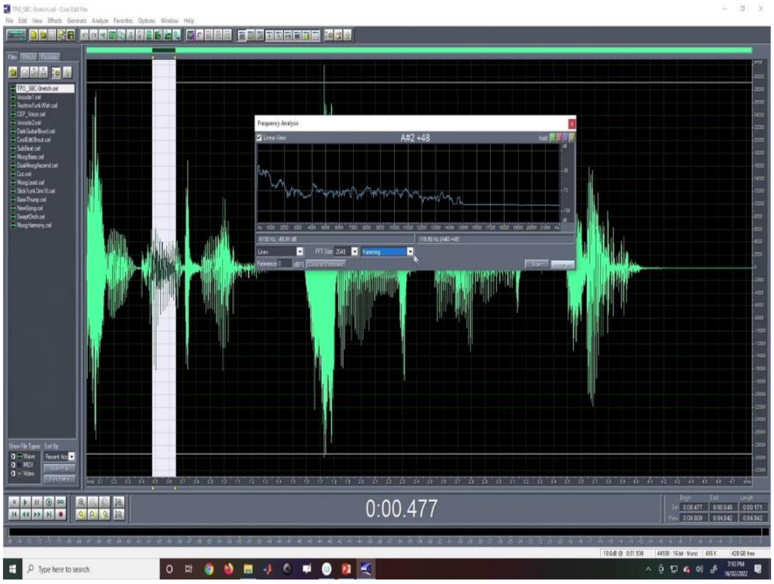
So, the small portion they have doing.

(Refer Slide Time: 10:02)



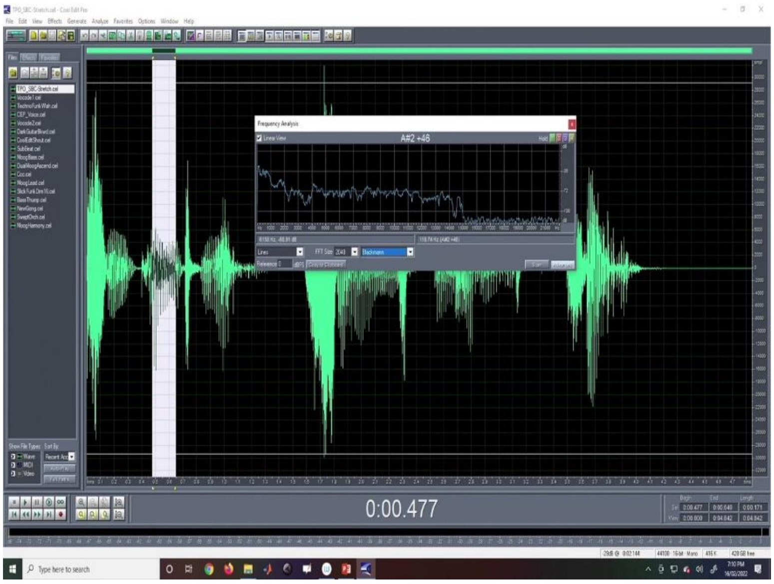
I can change the window.

(Refer Slide Time: 10:04)



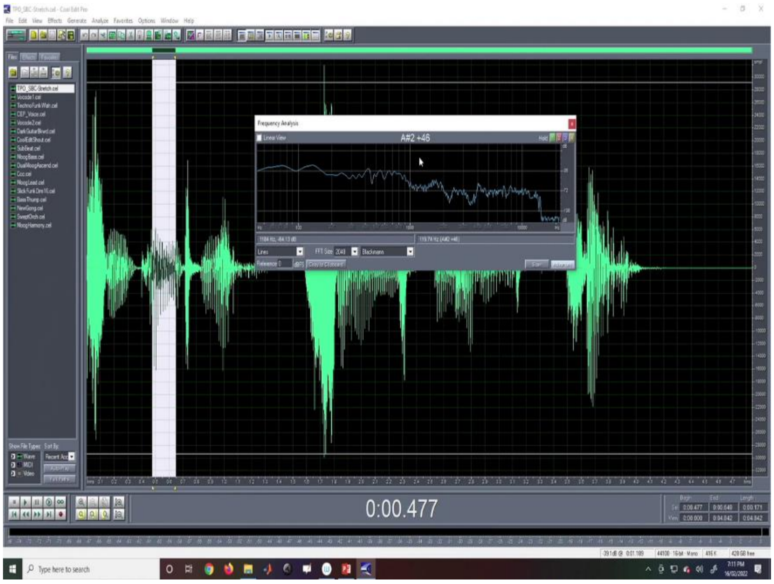
Hanning window, hamming window.

(Refer Slide Time: 10:07)



Blackman window window.

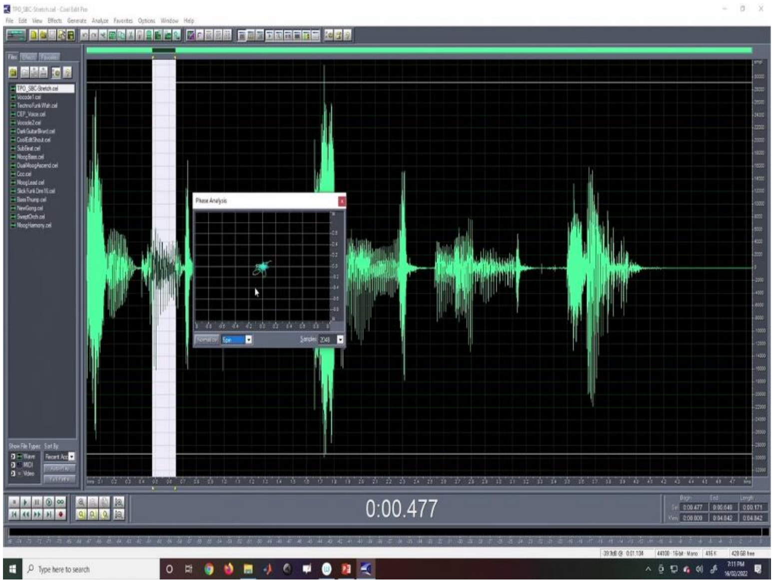
(Refer Slide Time: 10:12)



I can say that it is not a linear view; it is a log view. Log spectrum frequency axis in here. If you see this is in the log, log scale, or I can line graph.

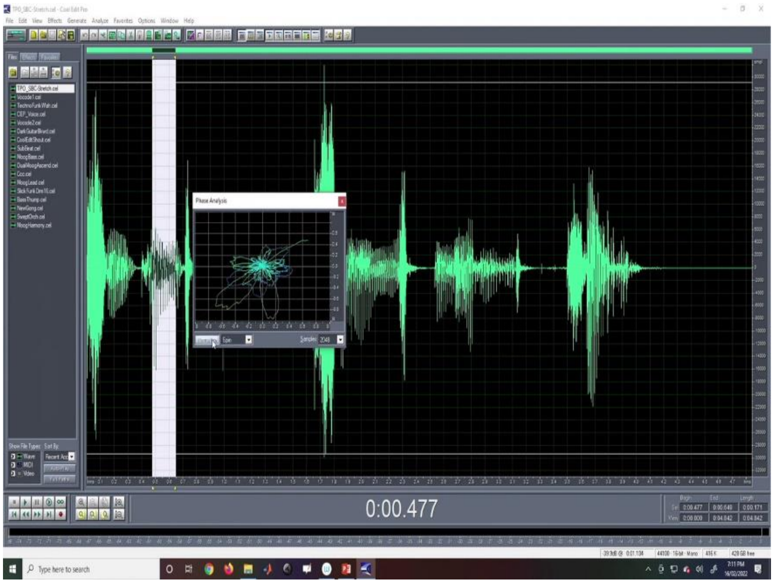
So, all those things I can do for frequency analysis of the signal. So, suppose I give you a target that supposes I told you to write a program to compute the spectrogram of a signal. There may be phase spectra. If you look here, there are two kinds of spectra is there one is called phase spectra, and another is called amplitudes, which is amplitude spectra.

(Refer Slide Time: 10:55)



If you see, go to analysis. So, in phase analysis, you can get the phase spectra. So, if I told you again, phase spectra are also the same thing from left to right mid to spin, you can say the spin.

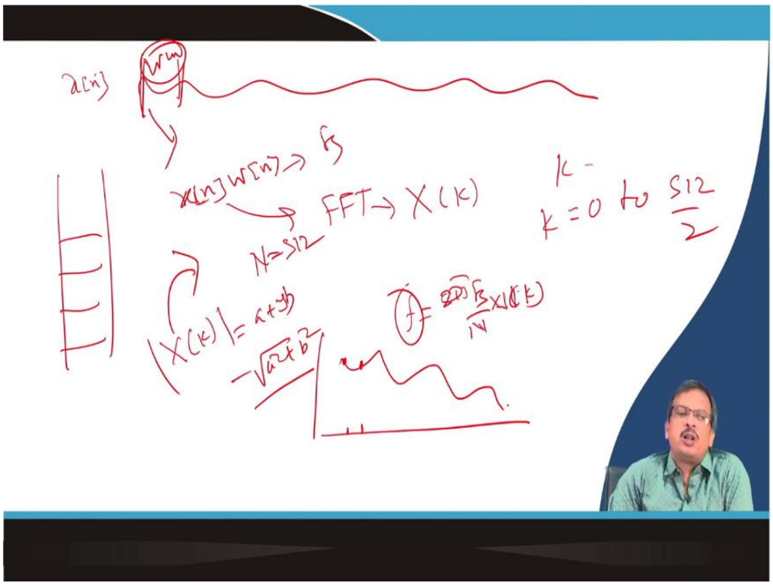
(Refer Slide Time: 11:11)



So, this is the phase spectra normalize like that. Suppose those things are the graphical representation.

So, suppose what is phase? What do you do with the amplitude spectra and phase spectra? So, what is given? The signal is given, which is also non-stationary.

(Refer Slide Time: 11:28)



x[n] is given, which is non-stationary. So, I have a long signal, and I select a portion I want to the spectra.

So, how do I select a portion? I multiply the signal with a window function ok. So, when I write different window functions are there, as you know, Harris Blackman window hamming window hanging window rectangular window, and I have described what the effect of each of the windows is. Some of the window's main lobe is very, very high, and the side lobe leakage is less.

Many times, the rectangular window's main lobe width is less, but the side lobe leakage is very high. So, you know that I will not get the original signal spectra. Only the original signal will be convolved with the frequency response of the window spectra. So, I implement this rectangular window.

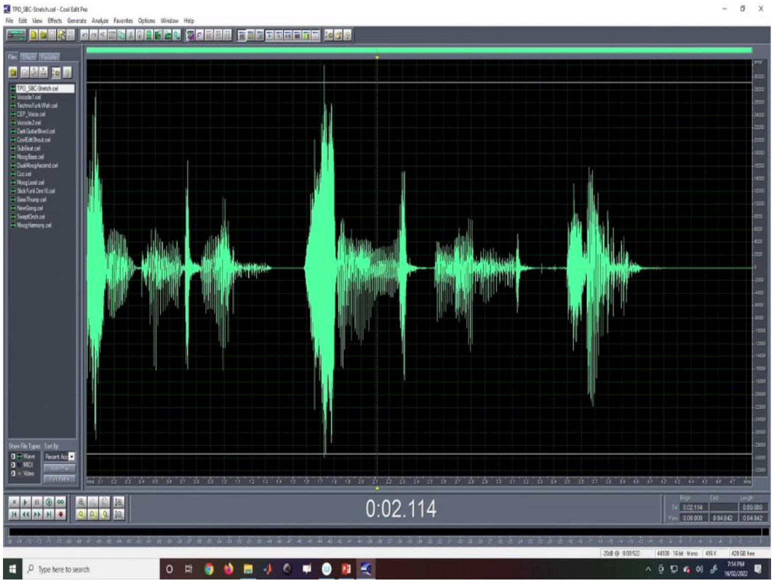
So, I first get X[n], then multiply with W n, then pass through the FFT algorithm, which implemented the discrete Fourier transform less N is equal to, let us say, 512. Then, once I get that, I get X(k) k varies from 0 to 512. I know I do not require the last portion. So, I can say the k equal to 0 to 512 divided by 2 is sufficient.

So, I get, and then I take mod of X k; that means X(k) is a complex number. I can take the real square plus the imaginary square. So, X(k) is represented by a by j b. So, I compute the root over of a square plus b square, and once I do that, I know the sampling frequency of the signal.

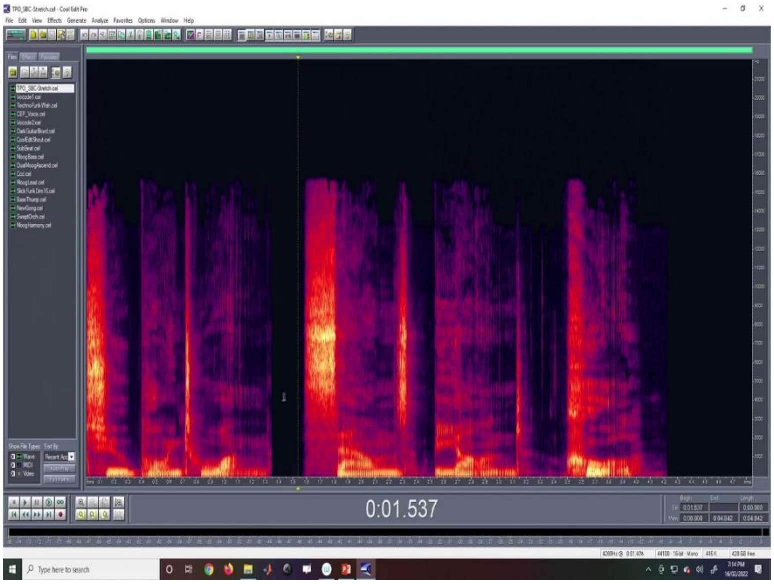
So, once I know the sampling frequency of the signal. I said f is the relation between the f and k 2 π is nothing, but Fs by N into k. So, I know the k index, and I calculate the corresponding analog frequency. So, corresponding to that frequency, I pointed out that amplitude, corresponding to the frequency, pointed out the amplitude you get the spectra.

Now, see spectrogram. What is a spectrogram? If I show you, so this is a time domain signal.

(Refer Slide Time: 13:56)

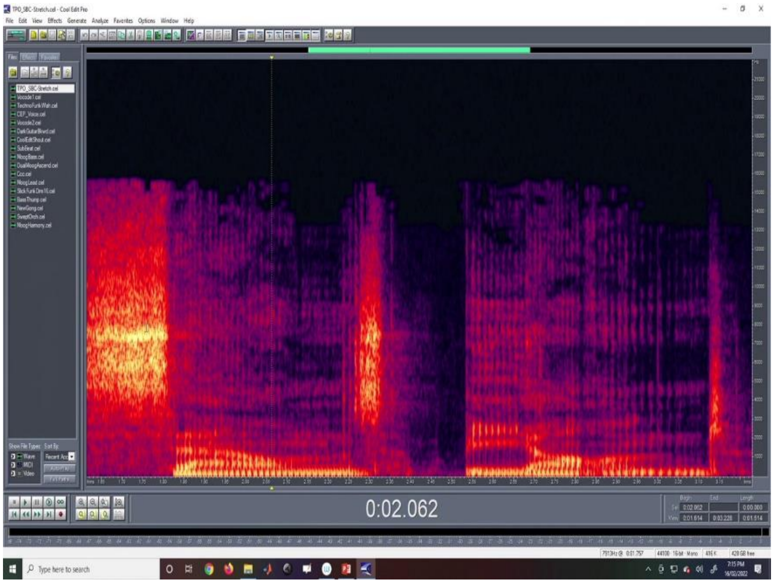


(Refer Slide Time: 14:00)



If I draw this one, it is called a spectrogram. Let us say I zoom this portion.

(Refer Slide Time: 14:07)

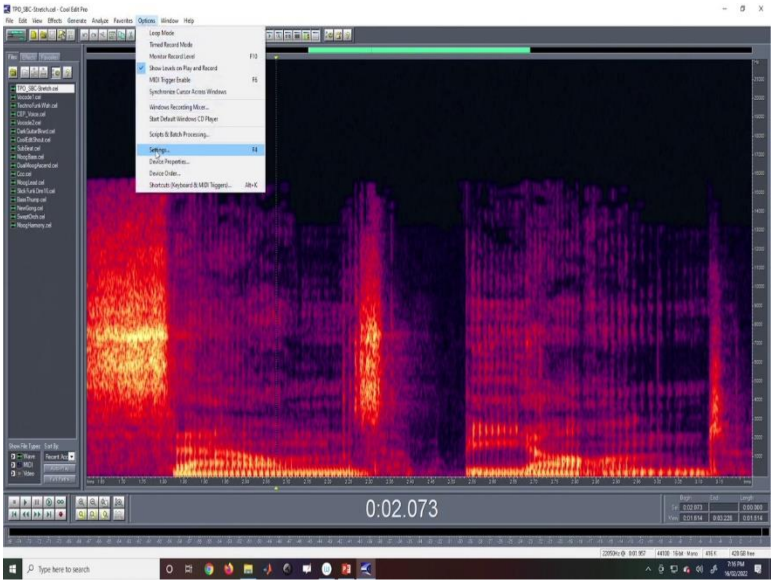


So, more yellow means more power, and more prominent in the frequency means that frequency has a higher power. If you look at, I think this axis in the spectrogram this axis represents this axis representing the frequency, this axis represents the frequency, and this axis represents the time.

So, with different time instants, what is the frequency composition of the signal as represented by a spectrogram? You can see this looks like an X-ray plate. So, how do I do this? This axis's x-axis is the time axis, and this y-axis is the frequency and intensity or color of the image. The intensity of the signal can change to represent the colour of the image.

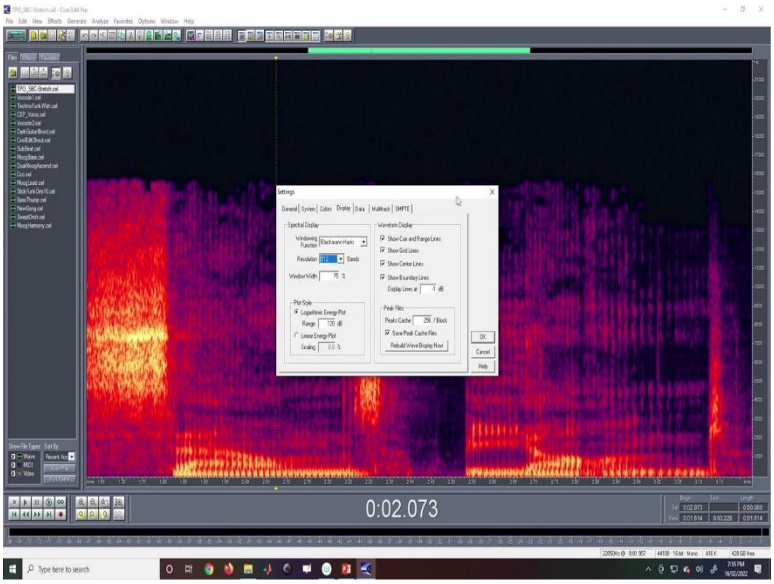
If I represent it in purely black and white, then if I say black, more black means more energy, and more white means no energy. So, this portion will be white, and this portion will only have a black line. So, that is called a spectrogram, but how do I do that?

(Refer Slide Time: 15:12)



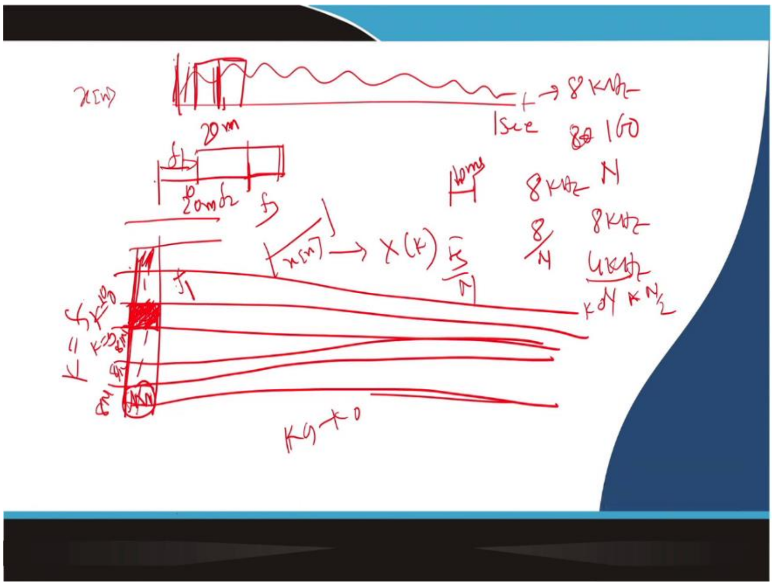
If you see here also, when I am doing the spectrogram, there is some setting.

(Refer Slide Time: 15:26)



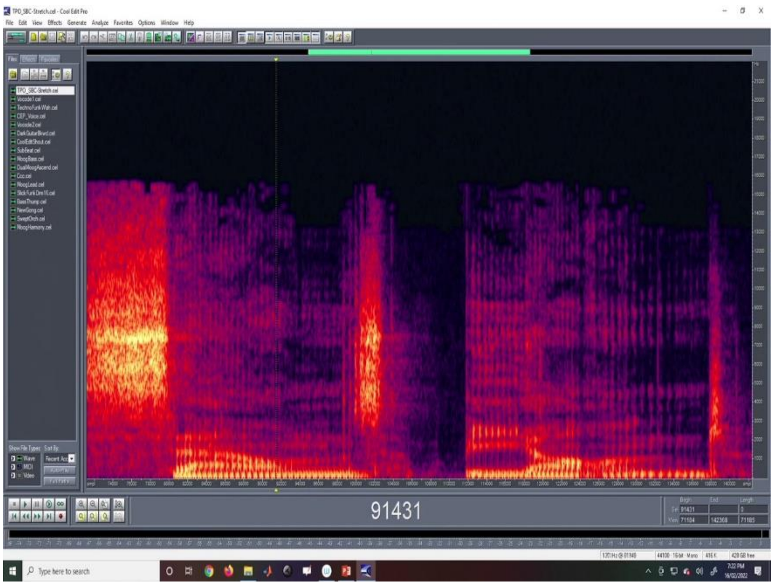
Window function, which window function, what resolution, how many bands you required, window width 70 plus overlap. So, all kinds of settings you can do. So, suppose I told you to write a program for computing a spectrogram or representing the spectrogram of a signal.

(Refer Slide Time: 15:54)



So, if you see I have a signal X[n], this axis is time I know, and the signals are there along the timeline. I want the timeline frequency component to be written down. But, if you see if I can compute the discrete Fourier transform for sample by sample, then if you see if it is equal.

(Refer Slide Time: 16:21)



Suppose these cases are in this demo case; how many samples are there? In the signal, if I click it here, it comes to the samples 142368 samples are there. So, if I compute a spectrograph and want to draw the spectrogram by sample by sample, it takes a huge amount of time. So, what will I do? So, instead of saying that I am taking the time resolution, let us say I take the time resolution to define a window.

So, how do I define in window? Let us say I told you that I would take a 20-millisecond window. The length of the window is 20 millisecond, and if you know that there is a complete recovery of the signal or proper representation, there is some overlap required. Let us say 50 per cent overlap.

So, windows I take 20 percent 20 millisecond signal with 50 percent overlap. So, if it is a 1-second signal, then 20, how many frames will there be? So these 10 milliseconds are called frame 1, then 0 to 10 is frame 1, 10 to 20 will be frame 20 to 30 frame 3, but the window is 20 milliseconds.

Why? If this signal is supposed to be 8 kilohertz, the sampling frequency is 8 kilohertz. Now, you know in 10 milliseconds, there will be only 80 samples. So, if I had so many 0, the spectrum would not be that correct, and the frequency resolution would be a problem.

So, what will I do? I will, instead of 80 samples I take 160 and shift the frame by 10 milliseconds. So, if I see the time axis, what will it look like this? For every 10 milliseconds, I get the signal.

So, I am assuming that every 10-millisecond signal is the same; the signal is stationary. So, for this 10 millisecond, I get the spectrum. So, for that 20-millisecond signal, $x[n]$ converted to $X(k)$ ok. So, this axis is my k, k can be converted to f. So, for this signal, I get a horizontal frequency response for different k.

So, each cell represents frame number frame 1. Let us say k is equal to 10 here. So, this is k equal to 9, and this is k equal to 10. So, this box basically represents this portion of the signal: the k equal to 9 to k equal to 10 frequency band.

So, this is called a band. So, how many cases are there? If my sampling frequency is 8 kilohertz and if the length of the DFT is n, I know the resolution is 8 by N. So, each length is 8 by N, 8 by N, and 8 by N.

So, I can say those are the bandpass filter which are 8 by N. So, how many bands will be there if it is, you know, if the sampling frequency is 8 kilohertz for the signal content of 4 kilohertz? I only have to analyze up to 4 kilohertz n by 2 ok.

So, k by 2, k equal to N by 2, ok 4 4 kilo up to 4 kilohertz. So, how do I set k equal to 0 to k equal to N by 2? So, up to 4 kilohertz, I will; I want to know what the resolution is. Fs by N, and how many bands will be there? If the Fs is 8 kilos of 4 kilohertz by N is the number of by N is the bandwidth ok.

So, N by 2 number of the bands will be there N by 2 number of such bands will be there, and each band bandwidth is Fs by N by Fs by N. So, if you see when you see the spectrogram, it is not, and you can see there is a vertical line and horizontal line.

Vertical lines correspond to the time resolution, and horizontal lines correspond to the frequency resolution, which is clear. So, that way, I can quickly draw the spectrogram of any signal spectrogram of any signal ok.

So, this is the application, and also, when we analyze the signal in frequency domain spectral estimation, I will cover that part of spectral estimation. So, whenever I have to analyze the signal in the frequency domain, whether it is single-dimensional or two-dimensional is nothing, I have to compute it 2 times: one is the axis, and the other is the axis.

First, I compute DFT 1 dimensional DFT along the x-axis, then for every x, I will compute the DFT for; so row-wise then column-wise, but the algorithm will be the same one-dimensional FFT algorithm will work in there also, but I have to compute in 2 times ok.

So, you can apply this kind of knowledge. So, whatever I covered in this FFT this week how do I efficiently compute discrete Fourier transform? If you see DCT, it can be represented by a discrete Fourier transform of 2 m instead of n.

So, if I implement the Fourier transform and if it is very efficient, then instead of directly computing DCT, I can also use this FFT to compute DCT. So, FFT, what I said, pure First Fourier Transform, is nothing but an algorithm to implement discrete Fourier transform or efficiently calculate discrete Fourier transform.

Thank you.