Real-Time Digital Signal Processing
Prof. Rathna G N
Department of Electrical Engineering
Indian Institute of Science - Bengaluru

Lecture - 24
Lab FFT in CCS – 1

Welcome back to real time digital signal processing lab. So, today we will discuss about DFT and FFT, whatever we have covered in theory. So, we will see first in MATLAB then we will go to the DSP processor board and then see how we are going to run our code today. So, first we will consider the MATLAB thing.

(Video Starts: 00:49)

So, this example as I was mentioning about the filters, so, this is from the book Welch, Wright and then Morrow so, you will be seeing that real time digital signal processing book published in 2005 there was an m file which shows that how the windows can be generated, and then the filter responses for different kinds of windows what we can see it so, we will be applying; one of the window for our DFT or FFT to remove the noise and then see that how they will be represented in the frequency domain.

So, here are a number of inputs what it is chosen is 128 and then alpha for Kaiser window is chosen as 3 and the number of FFT points are chosen as 1024 into 8 and then the sampling frequency in this case chosen as 48,000 and then we will be representing that is line type control for the plot and then you will be setting the font size in the plot of MATLAB. So, first one will be calculating Bartlett window with the end points and then hamming window, rectangular window and then the Kaiser window.

So, we have to pass the alpha parameters which can be variable So, you can select them as we have seen in filter design tool box. So, you can generate this, then you will be computing their frequency response, we call it as frequency z basically. So, you will be calculating a for that is $h(3)$ what you have seen the thing $w3$ by some of $w3$ what you will be putting it and then the sampling frequency what you will generate.

And then the frequency response for $H2$ is shown that with this parameters, and you will be outputting the figures actually, so, we will be plotting $w3$ that is sub plot what we will be having it and then font size line width what it is shown and all the 4 figures what it will be

plotted so, with different colours, and then later on you will be seeing that sub plot of what you are going to calculate that is P3 what it is chosen.

And then you will be finding these are the plot parameters one can go through you will be getting help file in MATLAB and then you will be, all these are the sub plot. So, you will be seeing that you will be calculating hamming window what it is going to be used and then its frequency after passing your data through that what you will be seeing it so, we will run and then see how we are going to get the output.

So, you can see the thing you will be seeing the first the figure we will look at it. So, you will be seeing that this is a rectangular window. And you will be seeing with the pink as Kaiser of window with $\alpha = 3$ what it has been selected. And then hamming window and this is the Bartlett window what you are seeing it is just like a triangular, and then this is the rectangular and then the other 2 represent your smooth response.

So the second figure, we will see what has happened. So you will be seeing that in this you have passed through the rectangular window. And then you are seeing the frequency response of after passing through the rectangular window. And then you will be seeing using the hamming window so how the output is going to be represented. So, you will be seeing that this is our main lobe, this is a low pass filter basically, and then these are the side lobes.

So, you will be seeing that rectangular window you will be seeing it has a main lobe is very narrow compared to your hamming window or any other window what you can take the thing and you will be seeing the ripple somewhere in minus 14 DB or something like that, but they have come down and then finally, it will be settling down to 21 DB or so, whereas, in the case of your hamming window, this is a smooth response what you are having with your input signal, so, the main lobe is little bit increased.

And then you will be seeing your side lobes have come down approximately what we can take it as minus 40 or 41 DB. So this shows that how our filter helps in having the main lobe and then side lobe required for your application, what you can select one of the windows basically designing with windows. So we will close this and then we will go to the DFT, basically.

So, here, what we have is usually we ask the students to run a DFT, FFT and then overlap add and save method. So we will demonstrate today up to this thing. We will go with the DFT, FFT editor, because I can split the window are editor can be in one of the thing. So here will maximize on the editor so that you can see the codes what it has been written. So initially, we you use the clear all, so you should be calling back the theory.

So we did the circular convolution using DFT properties. So we will be using, in this case, FFT to run our circular convolution, one of the way of doing it is I can put a breakpoint because the continuous code is running. So I will be setting the breakpoint here. So that we will be seeing one after the other. So first one is our circular convolution using FFT as you can see that x has the value 1, 2, 3, 4 and then H is 1, 0, 1, 1.

So this example we manually worked it out. Now we will see MATLAB how it is going to compute. So we will be first what we are doing convolution usually direct convolution as we have done it earlier. Now, using FFT, how we are going to do the circular convolution, we will look at it. So first, what we do is Xk is our FFT of x. And then Hk is FFT of H, what we will be taking it and in the frequency domain.

We know that Yk is going to be represented by direct multiplication of $Hk \times Xk$. So, that is our as you can see $Yk = Xk \times HK$ or $HK \times Xk$, then what we will be doing is I can take the inverse FFT of the Yk and compute and display a circular convolution result. So then, accordingly, we can do the linear convolution. So, how it is going to what are the values we are going to get it through a linear convolution is given by this equation.

Now, what is it ly linear Y output what we will be getting it so that convolve is the command prompt in MATLAB. So you can one can use it x comma h. So it will be computing and display linear convolution results. So we will run this code since I put the breakpoint till here what it will be running. So you can check the think results it will be displayed on the command window so, how to bring it down.

So you can see that as the circular convolution of 1, 2, 3, 4 and 1, 0, 1, 1, what we got the result was 6, 9, 8, 7 in the theory, so you are seeing using the MATLAB it was asked for you to verify with MATLAB, whether we are going to get the same thing. And then the linear convolution

what you can see is it is 1, 2, 4, 7, 5, 7 4 what we got it. So that we will be seeing that is direct convolution what it has been done.

So using the convolution whether we can do the linear convolution that is what the example we worked it out. So here you will be, so you are seeing that linear convolution by 0 padding in the MATLAB code what it has been written, so you will be 1, 2, 3, 4. And you will be we have added 1, 2, 4, 0 so this is 0s 1, 2, 4. So it will be generating 1, 2, 3, 4, 4 0s followed by and then the same way with hz also, 4 0s what it has been added.

Now you will be doing the DFT of our FFT computation basically, of our xz and then hz, then we multiply in the frequency domain, both of it, so why we can take the ifft of i of t, and then we will be getting the results. So just to show that, I can put the breakpoint and then see that what our output is going to be so we will continue the running. So we will see that the results are displayed here.

So, you can see that what was the thing the previous one, we had 1, 2, 4, 7, 5, 7, 4. And then y is the what we have got the output with linear convolution, and using FFT, so you will be seeing that it is 1.0, 2.0. And then so on, the last one is minus 0, because we added 1 extra 0 in the thing, because $L + M - 1$, as we know about it, so it has to be 7. So till there, what you will have it rest of them are 0s by doing the 0 padding.

So we will go back to the next, this thing, so I can maximize, and I want to show you the results. So we will reduce it and then go back. Now what is z? So we will calculate the amplitude spectrum of sine wave using FFT and then display it. So here are the thing is a number of samples that is 256, we call it a sampling rate. And then sine wave frequency what chosen is 50, basically hertz, and number of points chosen is 120.

So this up to 128 points 0 to n - 1. So you will be calculating sine of $2\pi \frac{f}{f_s} N$. So that is how we will be getting our xn samples in input sample, then we can calculate $x(k)$, that is FFT of $x(n)$, n number of points, and then we will be plotting the absolute value of $x(k)$ because we know that it is a complex conjugate what we will be getting it only the amplitude what I want to have the thing.

So we will be taking the absolute of $x(k)$, which gives us the plot the amplitude spectrum alone, and then this is going to give us the magnitude spectrum. So for the axis is going to represent 1 to 64 show only up 2 points, $\frac{f_s}{2}$. So in this case, what we have is 128 points, which represents up to 1 to 2π or f_s values so you will be labelling. So, what we will do is we will put a breakpoint again. So I had to go down and then the selector, a breakpoint here.

So, we will set in a build the breakpoint here. So, you will be seeing where next computation is going to happen we will have the breakpoint. So, we can continue this is to avoid and multiple files, all the codes have been incorporated in one. So, you will be seeing that this is the for up to 64 samples what you have it so, what we have is $f = 50$. So you will be seeing the magnitude whatever you are going to get the thing and then the frequency index you will be seeing that it is at approximately 25.

So, what you are having the frequency at $k = 25$. So, can you compute and then look at it whether we are getting the frequency as 50 hertz however it is going to be so, you have to multiply it by 2 which you will be getting it as 50 Hertz. So now, the next one is compute and display amplitude spectra at 2 sine waves. So, how we are going to do this, we hold on for a while so, we will set a breakpoint here also.

So, you know that next breakpoint is set. So, what we are going to do here, so, again our sampling frequency we have chosen as 256 sampling rate and sine wave frequency is 50 hertz and number of points, what we have chosen is 128. And we will be calculating sine of the thing 50 hertz sine wave. So, we will be doing the FFT of this one, and then calculate our this thing magnitude. So, this is what, what we did the thing.

So, the next one is we will see that amplitude spectra 2 sine waves, what is going to happen, so, we will put up the next breakpoint it is already selected so, it is the same thing. So, what you have is the first frequency or is 50, f_1 is 61 frequency of first sine wave and then second sine wave what it has been chosen and we will be calculating $x(n) = \sin\frac{2\pi f}{f_s}N$ and then $x_1(n)$ will be $2 * \pi\frac{f}{f_s}N$.

So, generate 61 hertz generate 50 hertz, then take Xk is equal to FFT of the first input signal and $X1k$ is of FFT of x_1, n basically second sine wave and then calculate their magnitude spectrum first sine wave and the second sine wave magnitude spectrum and then plot them with n values which is varying between 0 to 127. So, you will be only representing 1 to 64. So, when we run this, you will be seeing that how the 2 of them look like.

So, you are seeing this is multiple as you can see this is you will be seeing at 50 hertz and this should be at 31. So, you can see that approximately X is there 30 A little bit movement will give select 31. So, that is of what your Y magnitude is 40.7109 what you will be seeing it and you will be seeing this if I put the thing X is 25 and then Y is 64. So, you will be seeing that approximately you will be getting between 60 to 61.

What you are seeing the thing which is 61 hertz sine wave generated whereas, your peak is at 25 for your 50 hertz signal as because we are representing with 64 samples in the thing $\frac{\pi}{2}$ what we have done the thing. So, this is how your frequency spectrum and other things work. So, the next one is we are going to see that overlap of 2 spectral lines due to frequency separation is less than frequency resolution.

So, how we have defined our frequency resolution it is fs by N what we have taken the thing. So, you have 256 here again N is 128 and $f1$ in this case sampling frequency is sorry the frequency that has to be passed first sine wave is 60 the next one is 61. So, compared to the previous one, so, we have 1 hertz difference. So, we will see the resolution, how it is going to be represented then we will be calculating xn and then $x2n$ generate 60 hertz and then 61 hertz.

As usual previous then add them up and take the FFT of xn that is mixed these 2 sine waves and take the DFT computation find the magnitude response and then plot, so we will be checking the break point again so we will set the enable the breakpoint in the next stage and then we will run this so I think we will expand this you will be seeing the combined spectra sorry almost they are overlapping you are unable to see even the expansion of it.

So you can reduce the thing so, approximately it shows at 30 here combined frequency response so your resolution what you are going to see is very small in this case, so we may have to increase number of samples. So, what we can do is I can sample at this thing what we will call

it as 1024 we can modify this sorry 1024 and whether number of points whether I can increase it to 256.

We will see the thing and then we will rerun the code again, actually, it has gone off to further, we will clear all the breakpoints. And now we will put the breakpoint here and the next breakpoint will be in the next place. So 1 second clear all, I will break the thing. So that we can restart it because it has gone in to the end. So we will come back and then run that portion.

So, I had to put the break point now where we are calculating, here f_s we had given it as enable the breakpoint here. And then again, we will enable the breakpoint here, it has come to this. So, you can see that in the previous one what we are seeing as the output. So, now, I had to run it again because it is pointing to here, we will continue the thing. So, now you will be seeing that all the sorry it has been mapped to only 64 points.

So it will be little shifted you may have to multiply still you will be seeing that the peak has gone up to 200 and odd so whether we can expand it. So you will be seeing that 60 and 61 almost has got merged in this that is what, what we call it as frequency resolution which we are unable to look into the thing. So now what is it? How we are going to compare our rectangular and Kaiser windows for spectral analysis.

So the next example, so here the sampling frequency is 256 and we have N = 128 points. That is what it says sampling rate and then signal length and then the frequency selected is 61. That is sine wave what we are going to generate it, and then beta value in this case is selected as 8.96. And then we are going to use instead of alpha, we will be selecting beta in this case, wn Kaiser N comma beta, we will be applying it.

And then we are going to do the normalize the gain against rectangular window and then rectangular windowed spectrum what we will be calculating absolute of that. And the other one is we will be calculating Kaiser windowed spectrum from that, that is normal G into absolute of Xlk. And then we will try to plot the thing. So here we will put again the break breakpoint.

So that will not spill over so we will run and then see what will be the frequency and then magnitude what we will be getting it. So you can see that what is the other first one what we have it here is absolute of Xk. So that means to say which is the window you are going to have

it rectangular window here and then this is my rectangular window. And this is the Kaiser window what it is representing.

So with the beta selection, so we will be having the response of the Kaiser window as this way. So, you will be seeing that the almost equivalent to rectangular window the main lobe what you will be selecting it so you will be seeing your frequency will be approximately 61 hertz both of them are passing with both rectangular and then Kaiser. So we will see how to find the power spectral density of 2 sine waves embedded in our random noise.

So you can generate a random noise that is initialized random signal generator. So and then your sampling frequency what you have chosen as 1 kilohertz and then you will be having generate f s by 10 is you will be seeing that 100 samples of the every 1 point what you will be selecting it and you are being given what is it? Amplitude of 2 sine waves that is 1 and then 2 which is 150 and 140 frequencies of sine waves and then amplitude of the first sine wave is 1 second sine wave is 2.

So generate xn with A, $\sin 2 * \pi f *$+0.1 times the random value what you will be adding with your signal then you will be calculating your spectrum and periodic drum. So that is what the next assignment or lab portion of it will again enable the breakpoints here. So we will run it and then you will be seeing the spectral density here. So that is what was your power spectral density.

So what we had was 2 frequencies 150 and then 140. So you will be seeing that this is approximately 140 and this is approximately 150 or 4 or something approximate what you have the thing and you will be seeing that have your side lobes are buried in your nice what you are seeing it. So the next one is whether we can play and compute a spectrogram of speech file in this case.

So, next, we will put this thing in this case, you have a speech file sampled at 8 kilohertz you can record it using MATLAB and number of bits chosen is 16 bits in this case and soundsc played the speech signal, and then you can find it spectrogram using the function spectrogram, so we will see that first and then we will look at it.

Hopefully you have heard the thing this is the speech spectrogram what you will be seeing that so you will be seeing that your power frequency 0 db per hertz what it is shown so these are the 50 hertz and this is frequency in kilo hertz what it is marked so most of the speech signal as we say that it is going to be up to 3.1 kilo hertz so that is that reason why or maximum 4 kilo hertz that is why the sampling frequency is chosen as 8 kilo hertz and time duration in seconds that is the 3 second.

So this things this is how you can generate your sine this thing speech signal and see its spectrograph so now what will do is we will overlap our techniques for fast convolution what we are going to use it this theory we will be covering it in that how to do this fast convolution will be covered in theory and then how to implement FIR filter that attenuate this whatever 1 kilo hertz sine wave tonal noise speech file.

So, what is it? So you will be using the same speech file and that sampling rate is chosen as 8 kilo hertz. So first we will be playing the original speech signal then pure that is what speech is played then what you are going to do is we are going to generate a 1 kilo hertz sine wave which is called $\omega 2 * \pi \times \frac{f}{f_s}$ frequency of sine wave so you have done the thing and you will be merging with your speech signal that is what.

What we call it is correct speech by 1 kilohertz sine wave and then you can hear that corrupt voice and then we can display the spectrogram of noisy speech and then later on we will be applying window technique that is pass band filter 900 to 1100 fs by 2 is 4000 in this case what we have given and apply FIR 1 filter with stop band that design and FIR filter to stop the frequency this is a band stop filters what it is not band pass it is a band stop 900 to 1100.

So we want to eliminate 1000 hertz so that is why it is design and then you will be passing the coefficient b coefficient through your fft filter that is FIR filtering using overlap add method is being used here so that it is a continuous input signal what you have it so you would not be able to take the complete fft so part by part what it will be done actually in the overlap method add method is being used so we have the overlap same method also one of them can be used here overlap add method is going to be demonstrated.

So you will be displaying the spectrogram of the filter speech and then you will be hearing your output so we will run to the end. So that was the original so will. So you are seeing 1 kilo hertz sine wave is overlap with the speech signal clearly you can see that this is a noise signal so we will go back and then press the thing.

Although you faintly see the 1 kilohertz but still in the speech you are unable to hear the sine waves so this is a filtered output what you have got it. So this shows that how our discrete Fourier transform is computed using fft method implement for different applications it can be circular convolution, linear convolution or you want to eliminate your noise from the speech using the filtering technique so we can do it in frequency domain, thank you.

(Video Ends: 35:22)