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

Lecture – 15
FIR and IIR Filter in MATLAB Using GUI

(Video Starts: 00:31)

This lab class, we were doing the demo of musical notes. So, I wanted to show that how the musical note is going to look like in this spectrum. So, we will be doing the FFT of the musical notes. So, I will get the complete spectrum. So, have patience to listen to this music again. So, we will see the thing, as you have seen the thing the note has got increased. So, you will be seeing the spectrum of this. So, you will be seeing from the basic of your frequency here it is approximately 200 hertz what it is kept.

And then which goes up to 400 hertz, this will be my sa, you will be seeing here gah, ma, ba, the, ne, sa that is what what you have heard. And you can find out from the spectrogram what are the frequencies present in your music note or audio functions. So today, what we will do is as we are discussing about the filters in the theory, so we will have a look at how we can use the fda toolbox to represent what will the design what we can have with respect to, as I have been mentioning, fda tool box is going to be removed from the recent editions of it.

I can give filter design also what it will be taking it, so we will see that this is our fda tool box. So, earlier we have seen the thing that is I can have a response type, whether I want a lowpass filter or highpass, I can select the thing and then what kind of filter so FIR filter, either I can design the Equiripple filter. So, if you click on this, you will be getting the various modes of the filters. If I want to use the window technique, I can use it or any of direct implementation, what you can see that you can select them.

For the time being, we will select the window because we have comfortable with Hamming and then Hanning window in the theory. So, even the case of window what you can select it, so for the time being we will select first as the Blackman window, then what happens? So, whether either I can specify my filter order, or I will be going with the minimum order design, so I will click on the here, what it is shown? Shown is the sampling frequency is 48 kilohertz, so we will bring it down to 8000.

And this what we have seen in the last class is 1000 hertz, what will give it so, I can it is not allowing me to design the minimum order. So, only it is giving me specify the order. So, I can specify as 30 is my order, and then I can view the response of it. So, we had seen in the last class that is, this is my lowpass filter what I am designing it and samples what I have seen this is my normalized frequency what it will be showing.

So, that is my 8,000 kilohertz is 0 to 2π what you will 0 to π on the right hand side you will be seeing it on the left hand side it will be 0 to - π the response will be. So, here it is pointing to 0 to 30 is the thing order of the filter what it has taken this is the impulse response in the time domain now, it looks frequency domain in the theory we have seen it so, now we are seeing how in the MATLAB what it is going to show it. So, the next one is I can do the design of the filter. So, we will design the filter. So, you have to hold on for a while.

So, you will be seeing the response. So, what is it I have given the cutoff frequency is 1000 hertz. So, you will be seeing that so when I map this to frequencies this is going to go from 0 to 4000, so normalized will be 4 what you will be seeing it in this case. So, this is my approximately 1000 hertz, so which it should be giving me 3 dB cut off what we say. So, you will be seeing that approximately it comes down to this as my cutoff frequency comes down by - 3 dB that is equivalent to 0.707 of the magnitude.

So, you have to be careful when you are designing your filter, the output may be what I say is amplitude may be modified basically. So, scaled version what you will be getting it so, if you want if it is 0 dB, the frequency is what I can pass as you can see the thing is only up to 500 hertz. So, if you are doing anything beyond it, so you will be having the attenuation in the magnitude, so, you may have to compensate. Now we will see that how the bandstop frequencies are going to look like.

So, here it has come down to approximately minus 80 dB what we can say using the Blackman filter. So, it depends on how much down you want to come down. So, you can think of it and then use one of the filter the order if I increase it as you will be seeing it so, I will make it as 50 order filter in this case, so we will see the Blackman filter itself so, I can design the filter. So, now, you will be seeing that as in when I am increasing the order of the filter. So, you will be seeing that the flat response is much more wider here.

That is the passband will be for more frequency and this is my transition band what we call it from the 0 to whatever the delta is represented. So, it will be narrowing down as and when the order of the filter is going to increase. So, now I want to see that what is the group delay of this, so I can click on it. So, you will be seeing that hopefully, there is a trigger in your mind which is happening. So, direct form FIR filter is designed. So, the order is going to be 49 and then whether it is stable filter or not what you will be seeing it.

It says yes and then it is a designed. So, why am I getting the group delay it is a constant in case of linear filter. So here, the delay is going to be although I have for 0 to 49 samples, I should have had a delay of 50 samples, since I have used the linear phase property. So that delay because I will be combining it if you remember your structure. So, which comes down by half basically. So, the constant delay of this filter is going to be 25 units. So, after that, you will be getting it continuously your output if you are running for your real time applications.

So, coming to if you want to represent the phase of it. So, you are seeing that my phase has to be linear in the past band that is 0 to 1 kilohertz what we have given the thing this is cutoff frequency, so which you will see that it is constant. So, if you want to represent your response is this and then if you want to see the filter specification, you can go and then see that this is how my cutoff frequency lowpass filter what I have designed, so like this you can go design your highpass filter or bandpass or bandstop.

So, we will see one bandpass filter specification. So, you will be seeing that there will be 2 cutoff frequencies that is F c1 and then F c2. So, these are the 2 things what you have to provide. So, as usual, we can say that 8 kilohertz what I will take as my sampling frequency instead of 48. And then I will say my low frequency I want up to 400 hertz, which has to be eliminated. And then here I can give it as 3200 hertz. That means I want to allow frequencies between 400 and 3200 hertz and then rest of them have to be eliminated.

So, I will design the filter here, same Blackman window I am doing it. So, this will be the response what you can see it. So, what is it? So, 400 hertz is going to be this thing somewhere here what you will be getting it and then this is 3000 approximately 200 what you will get the thing. So, this is what the design so, if I change my filter to Hanning window, so, you will be seeing the response what is the thing is going to happen. So, you are seeing that, there are ripples, what you will be seeing it and then your response is going to vary.

So, when you view this, this is what you will be seeing in the time domain how the Hanning window looks like and then this is the frequency domain. So, you will be seeing that your magnitude your ripples will be somewhere here that is -20 dB or whatever you will be getting it. So, that is what it says mainly width is -3 dB in this case, you will be seeing it. So, these are you will be analyzing these things maximum relative sidelobe attenuation what it says is -31.5 dB.

So, somewhere here what you will be getting it below that. So, this is how you can play with your design thing and then you can use it in your filter design. So, you can use FIR 1 or FIR 2 to do the design or these are the coefficients one of the things what you have to do is if you are using it for code composer studio, so, you will be seeing there is a IDE or generation of coefficients any other platform if you want to use Xilinx platform or HDL code you want to generate it or C header file if you want to use the C code you can generate using that.

Or I will put it as code composer studio what I want to generate my coefficients with so I had to specify the header file C header file and then I will be getting here it is numerator and the length is defined with a BL and then export what is suggested what it says is signed 32 bit integer what your export it as that is what it is showing. So and then I can specify the DSP board also select the target for which I am generating it.

So or I can specify what I want it to be exported as I can export it as signed 16 bit integer or I can specify any other format I want unsigned or unsigned 32 bit or signed 8 bit integer depends on hardware, which one you are using it for us 16 signed 16 bit is enough. So, I have chosen signed 16 bit integer in this case, and then if I call it as a generate it, you will be seeing that it is going to generate and then it gives the name as FDA coefficients. So, we can save it. So, I will be saving it in the documents because I have lot of things in downloads so better to save it here.

So, you will be seeing that it is saved as star dot h file. So, that is FDA coefficients dot h what it will be storing it. So, we will go and then look at the values of it. So, in the documents file, what I have stored it, so, you will be seeing that FDA coefficients stored here dot h file. So, if you see the format here, so what is it? It is filter coefficients, see source file what it is generated, and then from filter design and analysis toolbox and MATLAB whatever version signal processing Toolbox what it has used is 8.5.

MATLAB registered what it says and generated when date and everything. So, you will be seeing that what is the filter we have generated it is FIR filter direct form FIR what it says and the filter length is 51 because we gave order as 50 so, it will be generating the 1 n + 1 what it will be taking it 51 is the order and linear phase it is using type 1. So, if you want to change and other things you can go and then look at the manual and then you can do that. So, it says that filter options were truncated to fit specified data type.

Because we gave signed 16 bit, so you will be seeing that BL is my length of the coefficients, and then you will be seeing the values of these coefficients here. So, it is from 0 - 3. So, you are already seeing that it has taken care of truncation of the values in this format in the fixed point format what I have given. So this, you will be using it in your code composer studio to run your FIR filter applications. So, this is one thing. So, we will see how one can run in MATLAB itself. So, we can close this.

And then what I will do is from the fda toolbox, I will be coming out of it, if anybody wants to see rest of the thing. So, you will be seeing FIR filtered bandstop or highpass filter, one can design depending on your application. So, coming back to this thing. Have you can one by one run and then check it in the MATLAB dot m file, what you can generate and then do the thing. So, here usually I asked my students to generate a GUI for it will be easy to show that what is the filter they are using it and then how I can modify.

So in this case, as a demo, I am showing you this one, one of which is developed. So, I will be going into this and then running it. So, you will be seeing that GUI is getting generated. So, now I have to specify here filter type what I want it so we will see that I want FIR filter we were trying it we can do that and then what is the response type what I want whether I want a lowpass response or a highpass, bandpass or bandstop. So, since we have done the lowpass and then bandstop design will take up here bandstop.

So now, what is the thing is usually why we use filter, in this case, we are going to eliminate the noise. So, we will play the original wave file and see what I will be getting the output as play. So, this is the original audio speech file what you heard it now will say when the noise is added how does it look like. So, there is a single turn noise what has been added to the speech and then which is fed as input to your filter. So, that is what it says which tone has been added is 2100 hertz as noises loaded into this now, we can play the filtered output.

So, you will be seeing that, so, this was a single tone, what in terms of frequency what you are seeing it so approximately 2000 although it says 2100 hertz, so, you are seeing as a peak at 2500 and rest of it, it is vanishing, it is not staying so your speech of magnitude filtered output what it showed and then went off. So, if you want to design an IIR filter, so I can give this I can select a lowpass filter IIR filter. So, it will be saying that some of the speech file with noise sine tone what it has added is 3750 hertz, which is going to be eliminated.

Since we have seen the original already so we will hear for our change. So, if you are keen in observing the notes, so you will be seeing the difference between 2100 and then 3750 hertz. Now, we will play the same filtered output. So, you have seen the peak frequency which got added has got changed to 3500 approximately in the thing scale has not been normalized I have expanded the thing and what was the output so this eliminates the higher frequency component present in the thing and then it is allowing the speech to come out as clean.

This is how what you will be experimenting with FIR and IIR filter together that is it is GUI. Otherwise, you can run the code in MATLAB create a dot m file and then you can run them also. So that is what one of the example what we have seen using the MATLAB filter. So, we will see in a while now. So, if you want to save or whatever it will be showing so we said that our sin generation using the dds we had seen in the MATLAB, so we will see in our DSK 6713 board ti board as I have been mentioning it, so how it is going to generate the sine wave.

So, this is the C code what we have written. So, we have to use the library function and then the math function and we are defining my length of the thing as 256 and we are keeping the rest of the code what we have used in the MATLAB function. So, if you correlate the equations, it remains the same thing. So, your desired frequency is 1000 and then the sampling frequency amplitude is 32,000 and then pi in this case, we have to define because there is no built in pi value in this case like in MATLAB in C as you know, you have to specify that.

And then we have given the phase increment here and sampling frequency chosen as 48000. So, I can generate both out 1 and then out 2. So, what is this out 1 and our 2 we will see in a while? So, one thing one has to keep it in mind when you are using the board. If you declare them as global variable, then you will be able to see their output. So, if it is a local variable, so you may not be able to plot them. So that is the reason why output is kept as a global variable, then we will be using the equation for i = 0 less than i less than n like even in MATLAB we did the thing.

So, we will be doing the phase increment and then what is with these desired divided by f s what we have it desired is 1000 f_s is 48 kilohertz what we have used it and then increment the phase. So, we are checking if phase is greater than or equal to 2 π. So, if it is so then we will be subtracting it with 2 π minus will be subtracting with 2 π. So, we are including as a modulus 2 star pi operation in this line, then what we are calculating first one is the output 1 of i this is magnitude multiplied by sine function what we are calling with the phase.

So, scaled what we call it as L in output and then we can feed it as a left hand sine one side and then the other one can be cos. So, you will be seeing that output 2 is given as amplitude into cos f

with respect to phase what you have calculated here. So, when I do these are the dot c files, what I am showing it as a main dot c. Now, one of the things is automatically when you open the new what we call it as give it as a new CCS project, if I select it, so it will ask me what is the file name I wanted.

So, I will call it as here in this case $\sin dds$ because I have given underscored there, then it will ask me whether I want the executable will say yes, and then what is the family I am going to choose is C 6000. And then we have given DSK 6713. If you are using the board, then I have to use spectrum digital DSK EVM eZdsp onboard USB emulator. So and then you can give finish, then you will be seeing that your main dot c along with it is going to be opened. So, here you can write your code and then run the thing.

So, for that and then it will show this as the active debug folder or project which is selected. So, for because we have already written the code, so you can go and then insert it so I will be selecting this as my active code to be run in this case. So, then I will select. So if I do this, it will be doing the compilation and even the build is going to be done in this. So, if there is any error in the thing, please go and then fix them. So, otherwise I can go directly for debugging then what happens it debugs and then it is loaded onto the processor.

So, you are seeing your cursor has entered into the main function. So, we will see that we will be going for the debug operation. So, you have this has to come when you are doing it, it says it has used spectrum digital DSK EVM dsp onboard 671X what is the thing taken and then now it has entered the main function and then interrupt what it is going to have is c underscore interrupt 00, it will be going into the board that is the entry point for the thing.

So to see that my output is coming out correctly so if I run completely, then I may not be able to see whether my output 1 and then 2 are sine and cos functions. What I put as a breakpoint here. So, if you double click on it, it clear set. And then if you double click again, it is going to put a breakpoint. So, we will run this code, and then you will see that it stops here. Now what I have to do is see whether I am getting the sine wave generated correctly or not. So, I will be going to the tools, and then I will be selecting a single time graph.

So, I have to specify what is my length of it. So, we said 250 and then my output data type what I have declared is float. So, I can use 32 bit floating point and then I will be giving this start address as output 1. So, then if I give the thing, so you will be seeing that your sine wave has got generated using that DDS method. So, if you want to, again plot one more graph, that is single time, I will call it, so this also uses going to be 250. And then this again, now, what I have is a floating point 32 bit floating point, so I want to see my output 2.

Weather I am getting the sine wave sorry, you will be seeing that this is the single time this is what you have it the other one is shown here. So, you will be seeing that this is starting from 0 what you are having it whereas your cos file you will be seeing that it starts from 1 which is in this case 32000 is the amplitude what we have taken so that is what it is showing, and then it is repeating. So, if you want to see their FFT magnitude.

Whether you are generating the, sorry, correct this thing frequency is getting displayed or not, I can select the FFT magnitude. So, I will be acquisition buffer again, I will keep taking it is 250. And then what I have is as 32 bit floating point, so I can give a start addresses for one of it, we will see that one you can see it yourself output 1, and then what is the frame size, we will be taking up spectrum analysis in little later classes. So, if you want, you can change the order of it.

So, I will give it as 10 2^{10} will be my order of the FFT. So, you will be seeing that this is how what it is represented, because 1024 so, if you want to represent in a lower thing, so to get 1 kilohertz so you can multiply what is the magnitude of it. So, the value what you are going to get it is going to show multiply with your sampling frequency into 1024. So, you will be getting what is the frequency what you are getting it.

This is our key point where it is shown. So, if you want to change it, you can change your order of the filter and then see what output you will be getting whether you will get the 1 kilohertz represented or not. So, this shows how we can use our DSK board using C function to generate sine wave. So, in the next class, we will in the next lab, we will be seeing how to do the filtering after generating a sine wave that is using IIR filter in the oscillatory mode.

And using it to do the filtering with the different lowpass and highpass what we have seen in the MATLAB same way we will see it in code composer studio. Thank you.

(Video Ends: 30:07)