Real Time Digital Signal Processing

Prof. Rathna G N
Department of Electrical Engineering
Indian Institute of Science – Bengaluru

Lecture – 17
Lab Real Time Audio Output through Sine Generation

Come back to real time digital signal processing lab part of it. So, we will see. Last class we have seen FIR sine generation, how we can use them in resonator and how the sine table can be used to generate an audio wave.

(Video Starts: 00:40)

So, today we will see how the filter can be designed, and then how it can be stored in different ways. So, we will go back to our MATLAB and here again, I will be using fda toolbox to generate my filter coefficients from MATLAB. So, you will be seeing that as I have been mentioning it will be using the filter designer in the next versions and other things. So, still, it accepts fda tool at present.

So, we are going to design a low pass filter, I can make it full screen and then we can design the FIR filter. So, here we will use the window technique. And then I can specify the order here. So, we will specify it as 30. So, I will be getting $n + 1$, it is going to be 31 coefficients what it is going to generate. So, window I can give it as Hanning window and then the sampling frequency, what we can give is 800 hertz, and then we will see that cutoff frequency is 1000 hertz, what I will give it.

So, then design the filter so, you will be seeing the magnitude response in this way. So, you will be seeing that the cutoff frequency Hanning window is somewhere around between -40 and then 60. So, you will be seeing the attenuation after 1 kilohertz it is drop down at 1 kilohertz it will be -3 dB that is 0.7307 what it is going to come down and then it will be coming down fully. So, you will be seeing that the response that is this has a linear response. That is what we have been telling FIR filter in the passband region as you can see it here.

Now, as last time, we will be importing this coefficient to our file dot h file what we can use it. So, if you want what it suggests is double precision floating point but since we are using in our processor. So, I can specify that it can be 16 bit signed integer what I want to generate it. So, if you want to give the processor board you can give here. If you specify here, I am going to specify it as 6478. So, this is going to select the target, and then we will generate the file basically as you will be seeing that variable name in the header.

So, what I am generating is fda coefficient. So, we will go back and then put it in the document file. At present, I am storing it as you can see, it is already there. So, I can give it us fda coefficients 1 dot you are seeing the header file is generated is dot h file. So, it says the board is not available, but still it is fine with us. So, we will go and then check in our documents file. So, you will be seeing fda coefficients 1 here. So, one of the way of doing it I will come in a while.

So, you will be seeing that the filter length is 31 and we have the direct form FIR representation and linear phase what it has designed and then type 1 filter what it has used it and then you will be seeing that it is stored in tmwtypes dot h basically in this include file if you want to directly include it in your processor, since it has not taken, it gives the warning. So, we know the length is 31 and then it is int 16 what we have given the thing.

So, what one of the way of doing it is I can take these coefficients and then go and paste it in my filter basically here you are seeing that FIR interrupt driven is there, I will come to that in a while. So, here I have one more FIR 1 kilohertz what it shows the thing I will be there are so many main dot c. So, if you want you can call by this name so, that will not have any confusion. So, you will be seeing that here in this case it is declared as float here. So, what is the problem with our thing is I have declared it as int 16 underscore t in the thing.

So, whether we are going to go back and then declare it as float, we will see it now, so, we can go back and then I can call it as single precision which is going to be 32 bit float what I will be creating it and then I can generate it. So, I will be over writing on the same thing. So, I want to overwrite this. So, and then if I go and then open this 1 so, these are the values what you will be seeing in the floating point.

So, I can take this values control c what I can do and then go into my CCS studio and then I can paste it, there are a lot of it, which has got open. So, that is why it was hiding in the thing. Now, what I can here because I thought I will not change this N that is the reason why I gave by even 31st order. So, if I want I can go then paste here. So, what was the cutoff frequency used in this case is 1 kilohertz. So, that is a name what it gives us FIR 1 kilohertz. So, what we will do is, We will do is debug this code.

So, you can see that it is building the console, and then it is now written onto the board. And then we have got the code. So, what I will do is? I will select the frequency here, component what I have to run. So, we will select this as 800 hertz. So, which it should be able to play on our output. So, I will run the thing here my code. So, and then play the 800 hertz sine wave. So, you will be hearing a tone coming out of this. Hope you are able to hear the tone, otherwise, you can try it on your own and then you will be hearing that tone of 800 hertz to come out.

So, sorry what I want to show again is that when we are playing little more than the 1 kilohertz, so, you will not be hearing the tone so, we will change the tone here. So, that it becomes I can go and then search in the directory. So, we have tones generated lot of them. So, if I press 2000 hertz, it should have got attenuated so, you will be hearing blank in this case. So, I will play this, as you can see, that filter is attenuating anything above 1 kilohertz in this case. So, that is the one of the example.

Now, how to store our coefficients in a different way? We will see it. So, in this case, what we did was we cut and pasted within the code. The other way of doing it is, as you can see that in this way I call it as FIR interrupt driven, basically. So, you will see that your main FIR filter is here. And then you can include the file lp55 dot h. So, you are seeing that as a include file, whatever generated from MATLAB, as you can see that this is the 11th order filter what you have it here generated.

So, you will be seeing this one, completely as a dot h file, directly you can store it in the thing. So, in this case, how this FIR filter is working? We will just see the code once. So, what we say is, as I have been mentioning, this is an interrupt driven here interrupt 4 is going to be accessed. So, what happens to this, we are calling our i is the continuous counting length, what we are going to have it and then you are initializing your output to 0, and then you are getting the input x of 0 is the first input.

So, we know that it is a float and then we get from the right sample input from ADC, compute your filter output, then what you are going to do for $i = 0$ to $i < n$. So, you should be calculating your $yn +$, this is the length of the filter what I have it. So, from here, you will be computing your yn+ equal to your h of i multiplied with $x(i)$. So, how we are going to update your weights in the filter so for $i = N - 1$ $i > 0$, $i --$ so, your updation has to happen in the reverse.

So, the last what we call it as $x(N)$ has to be thrown out and then the rest of them has to be moved into here. So, that this x is your circular buffer basically and x of i is going to be $x(i-1)$ value what you will be storing it. So, that, and then $x(0)$ first one what you will be taking it later. Output channel so, which is multiplied by 32000 basically, we have that is the maximum value as we discussed in the last class also.

So, it goes to our DAC and then return and then as usual in the previous demo, we have seen that it is initially it is interrupt driven what we have taken it and then 8000 hertz what we are running so, both ADC and DAC game, we have made it as 0, and then we will be taking it from the line input of the port. And this will run for continuously. So, if I stop here also the code will be running unless I reset the processor from the board or you will be recompiling and then putting one more code onto the thing. So, that is what, what is going to happen.

So, this is how we will be what we will? I will call it as MATLAB and then code composer will go in hand in hand for designing your FIR filter. So, coming to IIR filter so, demo we will see in the next class, but we can do the MATLAB design here. So, I will be showing you the demo also. Although we have seen it in the last class, I do not know how much you have given attention to that. So, I will give that as IIR filter.

So, whether I want Butterworth filter or Chebyshev so, we will see their responses, as I have been mentioning it get. So, we will go for the minimum order design in this case. So, we have to what we are going to match exactly whether I want to match stopband criteria or the passband criteria.

So, if you are not sure it is better to select stopband criteria because we call it as aliasing is not going to happen. So, it is better to choose stopband criteria to meet the thing based on it the order of the filter is going to be designed.

So, what I will design is here 8000 hertz, and then my passband I can keep it as this thing 3 dB what I want to come for 1000 hertz. And then I want 1200 what I can choose it as my stopband. So, and I have to specify in IIR filter. So, what is my passband ripple? Usually we keep it at 1. So, if we want 0.5, we can select the thing, it depends on your application. And in the stopband, whether I want to go down 80 dB or 90 dB or 60 dB what we have to specify.

So, we will see that at present will give this and then design the filter. So, you will be seeing that your filter is getting designed, and you can see that the order has gone to 48th order. If I have not given any specification for the order, it has gone to 48 and number of sections as we know we will be using the second order sections it is 24. In the theory class, what I mentioned was not to go beyond 12th order filter what I said the thing otherwise, we may have the oscillation.

But, MATLAB has the capability whether it can design a stable filter, as you will be seeing that if it gives it is a stable filter, still you can go with this order filter. And then how we will you will be seeing the response as you can see the thing and we will see its phase response, as we are telling that it is not going to maintain its phase. So, this is together what I am plotting the thing. So, this is my frequency response. And then this is my phase.

So, if we want to see only the phase magnitude response is this way, and then in dB, what it is going to show me the magnitude response up to 1 kilohertz, it is almost 0 after that it is start coming down. So, this is just the phase response. So, you will be seeing that passband is 1 kilohertz what we have given the thing, like FIR filter, it does not have a linear phase. So, for every frequency, you will be seeing that there is a change in the thing. That is why we call it as nonlinear phase IIR filter.

So, what I want to do is? How I can? Order of the filter I can come down. So, I do not want to design so much better order, there are 2 ways of it. One is I can increase my transmission band and say that, then with the same Butterworth filter, I can go down in the order. The other one is either I can select the Chebyshev or elliptic, actually, so we will see Chebyshev type 2 what is the thing is going to happen. So, we will design the filter.

So, my you will be seeing that you are what is it? Phase response is not going to be linear even in this case. So, you are seeing that from 24th order it has dropped down to 16 using my Chebyshev design. So, then what we are going to do is? We will see what happens with the elliptic. So, you will be seeing the magnitude response. Sorry this is the magnitude response. So, you will be seeing in Chebyshev 2, you have the oscillation in the stopband, and it is going to give you a flat response in the passband.

So, if you do not can allow oscillations in the passband, and then you want a flat response, because I do not want any other frequency that is higher order frequencies aliased into my system, then what I can do is? I can go for Chebyshev type 1 and then I can design the filter. So, you will be seeing that your magnitude, it is hard to increase the thing. So, in the stopband you will be seeing no oscillations are formed. So, I want the lowest order then what happens I can go with the elliptic design.

So, you will be having match exactly both of it I will be designing it. So, you will be seeing how much order it is going come down. So, you are seeing that order has the come down to 9th order and a number of sections have to be 5. So, the last section will not have the, it will be a first order section instead of second order because it should be multiple of 2. So, the 5th section will have only the single order filter. So, we will see the coefficients in a while.

So, as you can see you have a ripple in the passband as well as in the stopband. So, if I want to save this, this is what the filter specification you have given. So, I can give the same target as my Code Composer Studio. So, you will be seeing now from the FIR filter, you have got both numerator and then denominator coming in. So, IIR filter as we know it is a feedback concept what it is going to use so, it will be having both numerator and denominator.

So, I will be still keeping it as single precision float, and then I can generate this and then we will call it as fda coefficients 2. So, I can save this and then go and then check the thing. So, as you can see, how much order which has come down from 31 order to 9th order using our elliptic filter using the IIR filter. So, you will be seeing that what it says is? It is a real and then number of sections is going to be 5. And then whether it is a stable filter or not what it will give you? So, it says yes.

And then how the coefficients are going to be represented in this? Because I need both what we call it as numerator and denominator so, it will be shown as this is the scaling factor what we discussed in the last class today also, we will be seeing it why we have to go for the scaling? Automatic scaling for each sections what MATLAB is going to give out. So, this is my this thing coefficients what I have it in the because these are 2 are 0s, this is my a 1 is 1 always, you will be seeing it.

This is my a 2 coefficient - 1.2315 and sorry this is a 0, this is a 1 and this is a 2. And then you will be seeing 0s here. So, that is how the scaling function so, and then the second section what you will be seeing it here. And then what it says is? It has 3 values, what it will be giving you the thing. So, we will discuss our scaling and then come back and then see the thing. So, how many sections I am going to have it you will be seeing it. So, there are 5 sections in this case. What output you will be getting the last one will have only 2 of them.

So, this is how you are if you want to see the pole positions, you will be seeing it in your MATLAB. So, you will be seeing second order section of overall filters what you will be seeing the thing. So, you have the pole 0 positions based on it. So, you are seeing that circles are represented with 0s. And then what you are seeing this cross are your poles, how they are located, what you are seeing that this is a complex conjugate, what you are going to have it.

And then the grouping it is going to do automatically using MATLAB. So, just if you want to see the values of them. So, what it is going to give. So, you will be seeing that numerator is 1 and then this is b naught is 1 and b 2 is 1 and this is b 1 value. And denominator value this is a 0, a 1 and then a 2 and gain of that section, what it is giving? The section first and then after that we will be going with section 2. This is how we will be getting the values as your output from the MATLAB.

So, we will see the demo and then come back you will be seeing how IIR filter code is going to be written. So, this is with dsk 6713. So, you will see signed integer. And then how this is the order of the filter is what it is shown? Filter coefficients, this is how band stop filter for 900 hertz, what it has been designed. And then this is for 2700 hertz. So, we will see the demo of this in the next class. So, this is how you will be writing the code in c the more detail of it what I will come back and then discuss.

So, we will see this thing MATLAB code how it has been written. So, we will have to change the folder. So, one of the issue with this is what we will be reading is the corruptive voice dot wav in this. So, what wavread has been a removed and then in the latest version, it is the audio read basically. So, these are the ones one has to change it to as you can see, it is telling its error in the line 8 if you go with the thing. So, it says that it has to be audio read instead of wavread. Sorry I have given let us see whether it is going to run or not otherwise whatever still I have an error.

So, this is read audio also, here, which has to be replaced with audio read basically. So, it needs these are the errors file name range and then data type what it has an error in the case which has to be fixed and then we have to correct it and then come back. So, this is how you will be from one version this was implemented in 2016 b basically version and now what I am using is 2020 b. So, we will see the corrected one in the next class. So, thank you for listening. So, we will come back in the next class for the demo.

(Video Ends: 27:43)