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

Lecture - 19
Lab - FIR Filter in Generation of Music

Welcome back to real time digital signal processing the lab. So today we will be discussing about the filters in detail. So you will be seeing that first we will discuss the MATLAB and then we will go to the code composer studio how we will be verifying whatever result we are going to get from MATLAB which is going to match with hardware implementation. So, you are going to see that the same thing sa re ga ma pa what it we generated sine wave that is has been taken as the input or FIR filter.

(Video Starts: 01:02)

So, you will be seeing that all the 8 notes getting generated using the sine function here. And then you will be concatenating just like last time what we have taken the thing and then we will be sounding all notes and then if you are hearing the notes before adding noise and then you can check that after noise what is going to happen, what is the noise here, we are going to add in this case. So, you are adding for all notes, you are adding 0.3 times the random number generated with size of all notes.

So, we will play this after adding the noise what we are going to hear then what happens. So, we know that anything beyond in this case what is the maximum frequency we have taken is basically 5 times that of the you will be seeing that is 5 times star F Sa what it is been taken divided by 6. So, what is this? This is 440 F Sa. And then we will be going up to 880 hertz in this case and sampling frequency chosen is 8000.

So, after 880 hertz, so, you are going to eliminate the noise present in the thing. So, for that the coefficient has been designed in this way. So, you have to go to fda tool box which we saw in the last class and designed the FIR filter to remove the noise so, you can have 2 different voice of you having it so, we will see both the ways how it is going to look like then what you are going to call it as filter out initially you will be calling them as 0s and then concatenate and then you will be going from this up to 56,000.

So, the order of the filter is 121 in this case, so you will be going 1 to 121 so, you will be calculating that is count coefficient j all notes $i - j + 1$. So, you will be shifting them get the first sample and then rest of them you have assigned them 0s. So, you will be seeing that filter out of i is nothing but filter out of $i +$ the count initially count is 0. So you will be increasing it and then once all the samples have been filled, you will be getting the filter output.

So then you will be playing this note filtered out actually and then you can do the plotting also. So you will be doing that is next power of 2 from length of y what it will be taking it NFFT or normal FFT it will be padding 0s and then it is taking it. So the length of it l what you will be taking it Fs it is chosen as 8000 and then original what happens notes and then that is original FFT of the notes after adding the noise what you will be looking at and after filtering what is the note will be getting it.

So this it is going to be displayed using the plots basically we will be having subplots. So we will run this and then see that how it is going to look like. This is the original note what we have generated using sine generation. So that is what, what is this getting played here now. Now, what we are going to do is it says press any key to listen to the noisy notes. So, you will be hearing it, this is the noisy note that is random number 0.3 times the value of that is magnitude of it what you have added to all the notes.

So, still you are able to hear the notes because only a little bit of it, what it is added now, we will see how FIR filter is going to work. Still there is a little noise present, why because you have added in the complete region the noise in the domain of input. So, you can see that there is little noise left out in the even in the passband region. So, that is what, what it is seen and anything beyond the passband region so you are seeing that it is a plot response.

So I will be closing this, this is the output magnitude plot original this thing signal and then you will be seeing plot noisy plot. So, you are seeing the noise in the passband region and then here also you will be seeing the noise and then the filtered noise you are seeing that it is getting eliminated. And then little noise you will be seeing to the original which is being left out. So this is what we will say using FIR filter in MATLAB what we have done that.

So now what we can do is we can run our IIR filter here. And then we will go to demo of filters in board basically hardware. So, we will run this IIR dot m. So in this case, you will be seeing that your input and then F s so you are going to read file that is corruptive dot wav what you are going to read using audio read. So, this is how you are wave read has to be changed to audio read in the latest version earlier versions have wave read. So, you will be sounding this and then you can printf whether we are same thing what we are going to do.

So, now what is the thing you have to design your IIR filter using MATLAB and then you will be seeing that you will be calling numerator is a floating point values what you are going to have it and then denominator values for the f_1 represent the first section. So, you will be getting the denominator always a 0 will be 1 and then you will be having the rest of this thing values a_0 a_1, a_2, a_3, a_4 what you are seeing that thing.

So, the second section what it has is? It is f_2 and denominator we represent it as 2. So, the same thing here it is instead of a biquads, it is a fourth order what you have been chosen and then you are working on the thing. So, this is what, what it shows for, you will be seeing that can you think of this last one denominator f_2. So, how many you have a_0 a_1, a_2, a_3, and then a_4. So, you will be seeing the numerator is going to have one more section. So, for that reason, you will be having the scaling factor that is present in this.

So, now you will be going that is you will be initializing it and then you will be having 4 counts what you are going to have it so you will be going with numerator f_1 with 2 sizes of it what you are going to take it and then you will be counting them basically. So that is based on your numerator f_1 of i into x of i - j + 1 and then count 1 what you are keeping it for the first structure and then you will be checking the count 4 that is denominator f 1.

So, you will be adding it and then adding count 3 for that and then you will be making all the counts equal in this case, so that you will be able to move forward. So, you will be calling filter output 1 of i will be giving using this count 2 - count 4 divided by denominator f_1 1 and then your filter out initially you will be making them 0s and then you will be seeing that count 2 is numerated 2 into j, j is wherein from 1 to size of numerator f_1, 2.

So, you will be computing count 2 and the if it is $i > j$, so, you had to count in this way and then count 4 is denominator what you are counting it so, you will be getting again you are making it count 1 count 2 that is if the order of, because we are doing it continuously we have to come back to field filter that is circular buffering what we should have it so, that is how you will be computing the second section also.

Then you will be doing that you are passing your noisy signal through this IIR filter and then you will be checking that whether you are going to get the correct results or not. And then you can plot your FFT here. So, we will run this code and then see whether it is giving us the correct results. You can see that what is it? There is an error in the audio read that is corruptive voice dot wav. So, what we have to do is I had to give the path for it, which is not available in the current folder.

So, one of the way of doing it is I can push that corruptive voice dot wav here and then we can run it. So, what I will do is it is present in my sample files here, or I can give that complete path I can give it and then run from there. So, I can do the copy and then we will go to the MATLAB portion of it. And I will be putting this thing, I will be pasting it here. Now, I can do as all of you know clc will be clearing all, the thing.

So, I can go back and then read and the thing so, what we have added here is 1 single tone sine wave frequency what it is getting added to your speech signal that is what, what you are hearing it continuously. So, now what I will do is I will press the thing always. So, you can see that it got eliminated. So, what is the thing here it is there are 2 frequencies, in this it is 900 hertz 1 peak and then the other one is 2700 peak. So, which was added to the input speech to check that whether we; are going to get the output correctly.

So, you are seeing that initially there was this thing filtered output you will be seeing that only my speed signal is present in this so you will be seeing these 2 peaks have got eliminated. So, what I will be designing is a notch filter from for my application. So, how I can design the notch filter? So again, we may have to go back to fda tool. And then if you remember the portion of it, so we will see that how to design the notch filter or one stop filter what we called it.

So, notch is the single frequency what we will be eliminating it. So, this is my filter design. So, here as we; said that either we can design the frequency component which I want to eliminate. So, I will be taking IIR filter. So, either I can take Butterworth elliptic, 1 or 2, one of the thing.

So, in this case, what I want to design is notching what I am going to select in this case. So, then it says it is the comb filter what it is designing here.

So, if you do not want this single notch what I can eliminate 1 at a time or both together I can comb filter if I provide 2 frequencies what I can eliminate it that is why what you are seeing is there is the 4 this thing 0s and 4 poles what you will be seeing it so if I give you a single notch only one frequency what I can eliminate, so I can specify the order or I can do the minimum, this thing how it says here Fs is 8000 hertz. So, the notch what I want to eliminate is 900 hertz first time. So, we will see what is the thing is going to happen.

So, the bandwidth, what it is asking me bandwidth, I will keep it as 100 hertz here. So Q factor, what we want is 45 and units in dB passband what I want is a 1 dB ripple, so I can design the filter. So, you are seeing that this is the passband and it comes and then removes the 900 hertz here and then goes back. So, order what it has designed is the second order filter. So, you will be seeing that the phase response is going to be nonlinear, what you will be seeing it here see the thing it is not like FIR filter phase is not linear.

But if there is no necessity of the linearity in the phase, so you can design the filter. So, you will be seeing that the magnitude and phase together what it is showing here. So, if I want to see the group delay, so you will be saying that, although it is a second order filter, the maximum delay, what I am going to get is almost 50 samples. So, in the case of FIR filter, we can count how much delayed that is what we saw it last time, if it is the 50th order filter, the delay if I use the linear phase, which is going to be 25.

If I do not use the linear phase concept, then the delay of that filter is going to be 50. But whereas although this is a second order filter, so you will be seeing the group delay is almost approximately going up to 50 samples in this, so and then if you want to select the target code composer studio, what I will be selecting it here and then part one has to remember in this case, it is a 16th order filter basically and 16 bits are there, as we did the in theory that minimum of 2 bits required for my integer part 2 represent my filter coefficients.

So hence, what we will be using here is, if I want signed 32 bit integer, I can do it or I can say that signed 16 bit integer what I can generate from it. So, then we will see that will generate the filter and then it will be saved as always it is going to store it as fda coefficients so we will be storing it in this itself. And then we will see what we are going to get from that? Since we have not specified the debugger again the error will come. So, this is MATLAB code what we will see it so we have stored the fda coefficients here.

So, you will be seeing that it is discrete time IIR filter what it has designed and direct form to what it is doing designing. And then it said the length of it is 2 so, order of the filter will be a length plus 1. So, which is 3 basically and denominator length is also in this case 3. So, you will be seeing the values how it is generated. So, this is numerator length 3 what it has, so, you will be seeing 3 of them.

If you want you can do it in floating point you can run the code and the denominator all the 3 poles what you are seeing it here, this is how your fda coefficients will be coming out. So, this is what, what it was store in demo effect as you are seeing it here. So, it is the fourth order. So, there are one for your this thing one of the filter that is 900 hertz, the other one is designed for 2700 hertz. So, we will see that whether we can give it as a comb filter and then we can specify 2 frequencies here.

So, instead of single notch, I can select it as a comb filter. And then order I can specify it as basically fourth order let us see how it is going to look like. And then I have one the 8000 hertz and bandwidth still, I will be giving it as what I will give let us see 900 and 2700 whether it is going to accept it. So it should be single, so we will give it as 900 hertz. So, what is the thing happening here, so, you can see that so, even the approximately 900 into 3 is going to be your 2, 7.

So, I can design this and then this is my passband region what I will be selecting it and there is going to be notch here and stopband what is happening here. So, I can use this to take my filter coefficients and then run it also. So, or you can take individually and put both together. So, one after the other filter what you are will be running it. So, this gives the little bit inside of how to use the fda tools in our and one more thing as you seen in the thing.

I can go back and then store this filter descriptor to know that what filter I have designed how I taken and then I am using it in my, it is asking me I have to give that I need not have to store this. So, it came out of it, this is to remove 900 hertz sine wave what it has and the other one is to remove 2700, so, this is how we design and then cross verify it and then go back and then run this design filter coefficients whatever we have done the thing and then check in our hardware.

So, we have one more these thing filter dot m if you want to design the thing. So that is FIR filter, you can direct method if you want to run it. So, you can design the thing each one will have their own way of putting the thing. So, the books gives that is what, what it says is sinusoidal noise components from a corrupted speech signal here it is to bandpass filter with rejection frequency of 900 hertz and 2700 hertz are used.

So, there should be bandstop filter not bandpass or you will be having it to bandpass filter to eliminate this 900 hertz and then 2700 hertz. So, here also you can different ways of writing code what I am showing you. So, you will be directly what you are going to do is b1, a1 b1 is 0s basically. And a1 is my poles basically what we will be designing it? So, we will take Butterworth filter what it is being used. You are giving the day wp 1 and then wp 2 that is passband what you are giving 800 to 1000 hertz, what we want is the stopband frequency.

So, that is we are eliminating the frequency from 800 to 1000. And then I can do the quantization here that is basically fixed point implementation what I wanted. So, then what is the bits I am going to give, 16 bits in that coefficients are represented with 13 bits, that is Q13 format, as I have been mentioning, because coefficients need 2 integer bit so that is why we will be designing Q13 bits and you will be seeing b1 quantize Q, b1 and then a1 also you quantize that.

And you will be seeing that how the poles position is going to be then? We can 2 filter that is eliminate only one portion of it. And then we can say that, however, I am going to have here 1 frequency after removal and both together and second time I will be running it and then I can see that the last one how does it look like so we will start from the thing actually, the first one what we have is you will be hearing the original corruptive voice, so we will run this code, you have heard the corruptive voice.

So, how to find out, what are the frequency component present in it? So, this is FFT of the voice it is not the clean voice. So, it has gone to present the last one after removing the frequency component that is sine waves from that so, you will be seeing that 2 poles and then 2 0s what you have it here xy coordinates and then you will be seeing that these are the complex conjugate poles and 0s what you will be presenting it in the z plane and you will be seeing this is my audio over that what we have put in here.

The 2 frequencies are somewhere around 2, 7 and then 5, 3 or something like that. So, each one have a individual way of putting their own frequency and then try to eliminate that and then you will be seeing that real part how does it look like these are the 2 poles and these are the 2 0s of filter and then you will be seeing that the mirror image of the frequencies both together that is this is what you have is approximately 900 hertz basically what we have it.

The other one is 2700 hertz, you will be seeing that approximately you will be getting it as 2692 or whatever it is got generated, 9 6 2700 hertz. So, now what I will do is because all of them if they are run, so it will be causing an overlap. So, you will not understand the thing. So, you have heard the corruptive voice now. So to comment in MATLAB we use this percentage sign. So, coming to the next one I will remove one of the things and we can eliminate 1 frequency.

And then first is the 900 frequency what you are eliminating and you will be hearing the 2700 hertz here. So, we will run this again, same thing you will be looking at only you have seen that 900 hertz is eliminated and only this frequency is present after the first round of it. Initially it was these 2 frequencies present in the voice signal 1 got eliminated. Now what we will do is? We will comment on it so that we can uncomment the last one.

So, it goes through biquads section and then so, we will be hearing whether both got removed or not. So, you are hearing the clean voice that is what the clean voice what it is shown here. So, the original voice and this should have the same thing. And these are the you will call it as the second order section of the second filter and you will be seeing after the first one there was 1 component left out, which was removed by those 2 poles and 0s.

So, I am going in the reverse direction, these 2 poles and 0s have removed, whatever present our 900 hertz thing. So, this is how we run MATLAB code. So, you can have your own imagination, add your own sine wave or any noise to this and see how you are going to eliminate those noises. So, we will stop for this class with the MATLAB demo.

(Video Ends: 31:05)

So in the next class, we will take up completely DSP processor board demo so, that you will see that how little bit of noise whatever left out in the passband is going to emphasize in the board we will be looking at it allthough we have designed the same kind of filter in that. Thank you.