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

Module No # 08
Lecture No # 40
Application of Adaptive Filter in CCS, Echo, Scrambling and Graphic Equalizer in MATLAB

Welcome back to real time digital signal processing lab, so we will see that how our adaptive filter is going to run yesterday one of the wav file was missing in the previous class. So today or it has been put into the system so we will see that how it is going to run with different people writing their own code and then testing their algorithm. So the adaptive filter what it is shown in this of MATLAB file so you will be seeing that this also GUI yesterday we were trying to run the LMS algorithm.

(Video Starts: 01:06)

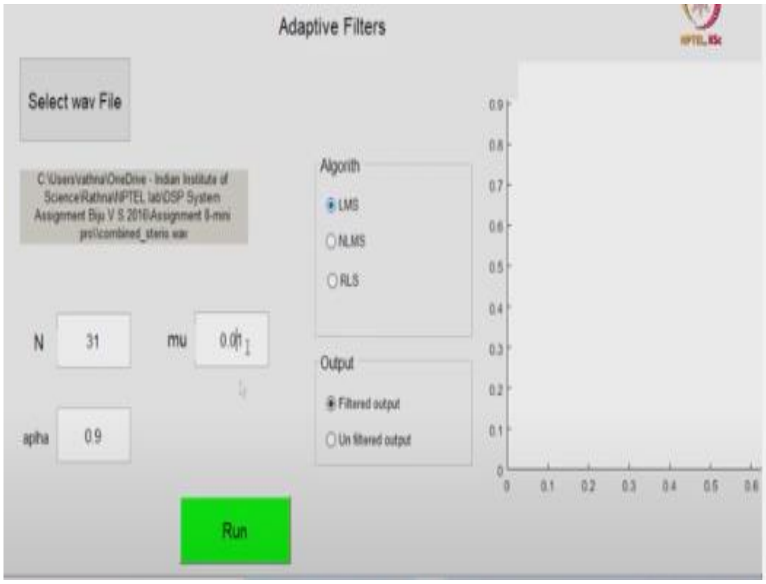
So the voice and then the tone dot wave together basically that is what we call it as here it is included in the combined stereo basically. So you will be seeing that there is a mistake in the stereo spelling so that that is what you have to do the thing and then we can run it. So when we are running it independently then we have to open this. Now we will see that how using the adaptive filter directly including we call this as the now you will be seeing that code is written in this way GUI code basically.

So that you can change your input file and then new and other parameters you need not have to go back to the system to change it. So you will be seeing that it is getting as yesterday I said in the last class sorry that is 2 tones what it is taking it one and then 2, 1 is the input the other one is tone. And then you will be combining it and then sending it that is we have left channel and then right channel for stereo, so both are taken together.

So then how our algorithm is going to work with mu and definition that is LMS algorithm alpha mu value what it is taken because, it contains both NLMS and LMS algorithm. So you will be selecting alpha for NLMS and only mu for our LMS algorithm. So these are the inputs because you can see their getting in handles made you can vary those values. So this is the complete code what it has it as it is given the thing, it is going to a code for adaptive filter dot figure creates a new adaptive filter and raises the existing single tone. So you will be seeing the property GUI what it is going to run it.

(Video Ends: 03:36)

(Refer Slide Time: 03:36)



So we will run and then see how it is going to work on this file. So you have seen that in the previous class that all the algorithms were together here you can select the wav file. So this is a combined stereo what I will be selecting it from this place and then the n is given 61 order, mu is selected as 0.1. So what I will do is I will change the thing to 31 order and then 0.1 for 0.01 for my adaptive mu what I will be delaying it.

(Video Starts: 04:20)

So we will run the LMS algorithm first on this run it, so if I want to see the unfiltered output will it is FFT and then output what you are seeing it. So filtered output, if I run the filter output so you will be hearing it clear speech which is coming out of it. So now what we will do is we will change the order to 61, and then our new value to 0.1 and then see what is the thing is going to happen, I will run the LMS algorithm it is a clear speech.

So you are seeing how your output is coming, so change of the order of the filter and then the mu value which is set to 0.1 step size, so you will be seeing how output is going to behave. So now we will see with respect to an LMS algorithm same thing what I will keep it and then alpha what selected is 0.9, so we will run the algorithm. So if it is a unfiltered output here also will check the thing to make sure that our input contains both the that is noise plus our speed signal can you guess what is the thing is happening.

It has stopped in between you would have heard the little bit of echo anyway we will be demonstrating echo also later. So this is the recursive least square algorithm, so we will run it and then see just like the previous example this also how it is going to behave. This is equivalent to using a recursive least square algorithm what it is being used the computation as we mentioned it is very high in this case.

So we can check the unfiltered here also so that we are sure same signal is getting passed. So we have seen the filtered output correct this is the way how you can design your adaptive filter.

(Video Ends: 08:01)

Now what we will do is we will take up as you can see there are different assignments so we have done the demo also the same way so this was a LMS algorithm was given as a mini project to all of them. So they designed their own programming concept what they have taken, because usually we do not allow for copying. So you have seen two different so we will have multiple students developing their own algorithm. Now we will see that one of the student think how our echo is going to work.

(Video Starts: 08:50)

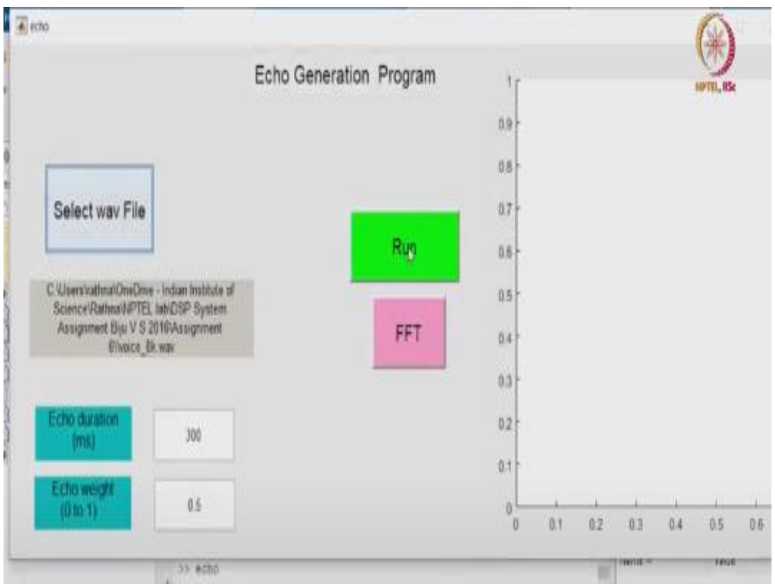
So we have seen the thing echo generation I will so what is the echo how we have given the equation, so I will show you the equation here. So you will be seeing that you are going to read your audio sampling frequency is set at 8000 and then echo duration of what you will be setting from the your handles. You can set the thing minim how you will not have the echo path and then if you set the duration little more then, how you will be seeing the echo path.

And then this is the delay, so that is given by echo duration multiplied by your sampling frequency divided by 1 kilohertz basically. So a 1000 what you have taken the thing so in this case delay 1 is this is the time duration what you will take it. So our out of I will be input of i what you will be taking it and you will be calculating the delay plus 1 to length of it. So you will be adding with input of i your echo wait into output of i – delay.

So this is how you will be adding to your input the as you have called out of i is input itself so this is the input delay what we are going to give it, whereas in the reverberation it is the output which is come whatever out of I you have calculated will be going as delay. So the will run this and see how it is going to respond to our data.

(Video Ends: 10:36)

(Refer Slide Time: 10:37)



So here also you will be seeing that you have been given echo duration millisecond default first we will run the thing and then we can modify and then see it. This is a 300 millisecond what the delay is there and then echo wait is between 0 and 1 what we have to give it, so here it is taken 50 that is 0.5 what it is. And then we have to select the wav file for running it. So we will see that this is the voice signal what we will take it. So then we will run and then please see that whether you are going to hear the echo.

(Video Starts: 11:17)

So you have heard the echo coming, so what is the FFT of it, so this is how the FFT of your echo path what you will be seeing both your input and then echo. So we will change the duration we will make it 30 millisecond, so in the theory we said that 40 millisecond our path what it is going to take for the, our telephone line to come back. So here we will make it and then we will make it as .1 also in this case, and we will see whether we are going to hear the delay or not.

So this shows that if the delay is lesser you will not be hearing the echo in your telephone lines or server IP or if you want to generate it, so this is how it works. So we will increase it to little more and see how the echo can be perceived, and I will give wait for it as 0.5 and see it. So you have heard that remember which is occurring so many times, so you can consider echo can be used for your reverberation. So it is not once it is multiple times what it is coming fine this is one of the examples.

(Video Ends: 12:57)

(Video Starts: 12:58)

So we will see from the next is our scrambler basically so you have to hold your breath for this example in the next example I will show you, I will go with this example later because they have not given a reset command to reset the thing sometimes it goes continuously fine. So we will go to the other this thing, this is the MATLAB code so I have to go to the particular thing. So you will be seeing that this is echo scrambler and equalizer all together in this case, in the previous case we saw only one running it, so here everything is combined.

(Video Ends: 13:47)

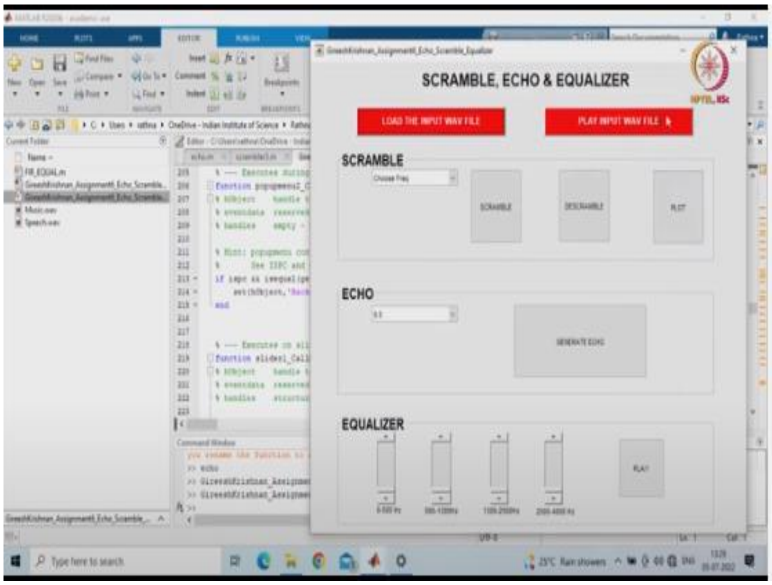
(Video Starts: 13:48)

So we will run the thing one more student thing how it is going to work here you will be seeing that all the 3 should be coming. So there will be a case statement to take care of it which one you are going to select based on it you will be running, you will be seeing that scrambler which is going to run or it can be your echo. The first one will be running echo, the next one will be scrambler, and then the last one will be equalizer we will be seeing that also how it is going to work.

So you are seeing the codes here this is scramble and then they scramble both is going to happen you can select one of it. And then you will be getting the plot also how it is going to look like and your this thing audio file what you can read the thing here, it is echo strength you are calling with alpha in this case. And then you will be computing your delay i - d is the delay. So which with the wait function will getting added with your input and this is what it is going to generate your echo so will run this code.

(Video Ends: 15:07)

(Refer Slide Time: 15:08)



As it is going out of the thing so we will just reduce the thing to 100 that is size basically scale the thing so that I will be able to show you the demo part of it. So we will run it so now you will be seeing the complete GUI otherwise some of it is getting suppressed. So what we will see first will check the echo basically, you can here it is given in seconds there in the previous example from the other student it was in millisecond.

Here you can define it as in seconds but it is as you can see 0.1 to 0.6 what you can go with it. So we will choose same as that one and then I have to load the input wav file so here will be putting the same speech file in this case, and then first we will play the input wav file.

(Video Starts: 16:13)

So one can select any wav file, so usually we give this and then even noise getting added corruptive signals are given to the students to run the thing. So now we will generate the echo from this case, so as you can see here you do not have the option to select the delay. So only single echo has been generated in the previous case even the echo duration you are had the provision to do it. Here only what wait what you are giving it, so if I give the wait a little lesser as .1, and then generator, so you will not be hearing any echo.

So as in when my wait is going to increase delay path is defined then I will be getting the delay now we will see the same thing how the scrambler is going to work it is the same input file what I will be giving it and then we can the frequency what I have to choose it. So as we discussed in the theory, so we had selected 3000 hertz as the thing we kept but here you have the variable thing.

So first we will check 1000 and then do the scrambling, so you will be hearing a scrambled voice; so you heard the scramble and do this de-scrambling link. So you can see the plot how does it look like, this is the scrambled signal unable to make out much difference in this case and then the de-scrambled output which is almost equivalent to your input. So we will change the duration that is filter frequency basically, so we will choose it as 2000 and then do the scrambling.

So you can hear that is there is some this thing speech you can make out and then de-scramble, we will choose 2500 hertz 3000 anyway, so then we will scramble it and then here how much you can hear or it is completely scrambled. Since you have heard the think earlier, so you would be able to make out what is that de-scrambling.

Next equalizer, so what is an equalizer if you have a nowadays anyway music system nobody is buying the thing so although headphones and other things what you buy for noise cancellation and you know boss is the best manufacturer of your all the, you we call it as audio equipment. So here the equalizer you can select which one you want to retain and which one you want to suppress it.

So as you will be seeing here it is a 4 channel equalizer so some of them go with 10 channel equalizer so that you are narrow band only those frequencies which you want to highlight it you can do the thing. When all of them are in the center you know that it is a balanced system what you will be getting it. So some of them design a 3 channel equalizer with the frequencies in this case what is it? It is 0 to 500 hertz is a low pass filter what it has been designed.

So that anything above 500 hertz by making it 0 whether I can cancel it will see the thing whether they have given the 0 option or not I am not sure of the thing, so we will verify it. So the next one is band pass filter what is the frequency after passing 500 hertz; next 500 hertz to 1300 hertz what we can use it to, or do are this thing frequencies that can be passed. So if I suppress the low pass filter and then only using this some of it is going to be highlighted and some of them will be suppressed; and then the next band pass filter is 1300 to 2500 hertz.

So this is what our or this thing 2 band pass filters in different two frequencies from here to here what you have the thing so they will be passing the signal between these 2 frequencies from them one can be 0 the other can be uploaded. The last one band pass filter what it has been used is 2500 to 4000 hertz the sampling rate here it is assumed is 8000 so till f is by 2 that is up to pi value what you are giving your frequencies to pass through.

So what we will do is here because we have to here the speech file does not have all the frequencies usually all your bands and other things you will be having in the higher frequencies. So we will see that how it is going to work for music in this case. So because usually the regular our august semester ends in December this student has chosen a merry Christmas as music signal to run the thing. So we will play and then see what is the output will be getting it, so I have loaded the thing i can play the input wav file.

It is a part of it not the complete because it will take little time nobody will have patience to listen to the thing, so will play the thing with respect to whatever the default settings play it. I have to have minim value for the let me check whether it is going to play there is, so some of the control this thing is giving the problem. So we will see that rerunning the thing because I think it might not have got cleared, so we will load the file again speech file will set minim value for all of them will make it at least little higher.

We will change the input file we will load the music, so now we will reduce these 2 if you had noticed some of the things, as you have seen the thing only it is getting passed with this; so I will include one more frequency. And then now we will play the thing still some of the things are missing so we can make it 0 here and then play only one frequency is getting passed. So you the clarity is gone.

So we will make it all of them this way you can as I mentioned in the introductory class that you can keep playing like this writing your own code what you want to generate it. So as this example shows that this is from the MATLAB file what we are running it, so you want to design your own hardware.

(Video Ends: 26:03)

So you can do that and then we will be seeing the demo in the next class using our DSP processor. So as you have seen it is a programmable device DSP processor we will be using it for dimming all these experiments in the hardware thank you.