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

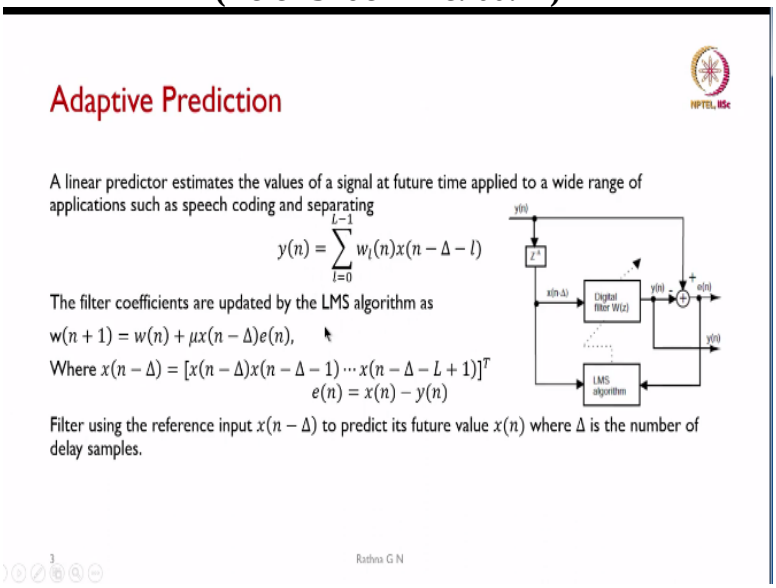
Lecture - 37
Adaptive Filter Applications

Namaste welcome back, to real time digital signal processing course. So, today we will look at adaptive filter applications.

(Refer Slide Time: 00:33)

So in the previous class, we discussed about the LMS algorithm, leaky LMS and then NLMS algorithm how with little modification, we can implement the adaptive filter.

(Refer Slide Time: 00:47)



So, today we will look at a little bit on the application of adaptive filter. The first one what we will take is adaptive prediction. So, how we are going to predict the system which is connected

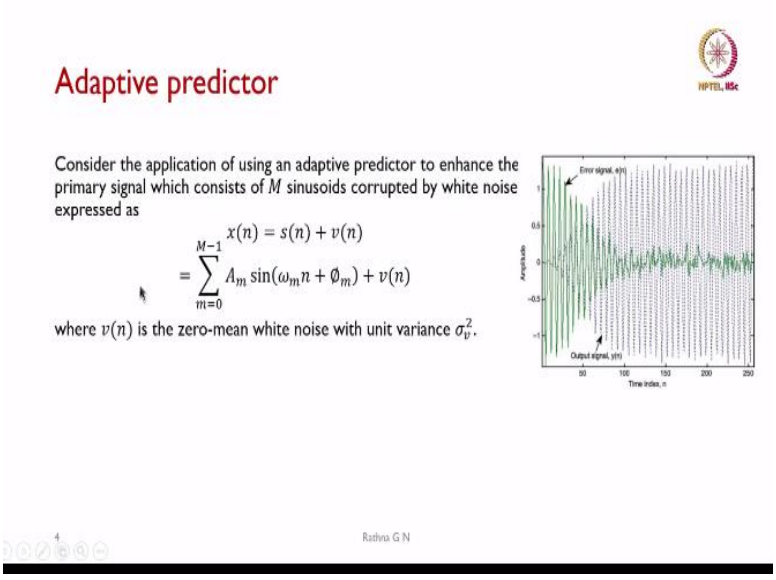
to the system? So, what it says is the linear predictor, it is going to estimate the values of signal at future time applied to your wide range of applications such as speech coding, and then separating. The equation for this is given as $y(n) = \sum_{l=0}^{L-1} w_l(n)x(n - \Delta - l)$.

So, you will be seeing according to the figure, there is a delay of the signal, which is going to be fed into the system to find out the error of the function so y of n is taken out here. And then we have the LMS algorithm here, which is going to decide weights on the system, basically, and this is the delayed signal, that is $x(n - \Delta)$, which is going to come into the system. And then $y(n)$ is the thing, this is what $y(n)$ is the predicted one.

So we will be subtracting and then taking it out minimizing the error, then what happens? So, we will be predicting what is y of n basically in this case. So, the filter coefficients are going to be updated by the LMS algorithm as you will be seeing that $w(n + 1) = w(n) + \mu x(n - \Delta)e(n)$ that is error which is fed back into LMS algorithm.

So, we say where $x(n - \Delta)$ is nothing but $x(n - \Delta)$ is the first sample that is delayed by Δ units. So, the other ones are delayed as earlier, we take it $[x(n - \Delta)x(n - \Delta - 1) \cdots x(n - \Delta - L + 1)]^T$ what we will be giving as an input? So, L is the length of the filter that is what, what we are doing l? l = 0 to L - 1 we are calculating $y(n)$. So, then as usual like LMS algorithm what we have is $e(n) = x(n) - y(n)$. And filter using the reference input $x(n - \Delta)$ to predict its future value $x(n)$ where Δ is the number of delay samples.

(Refer Slide Time: 03:49)

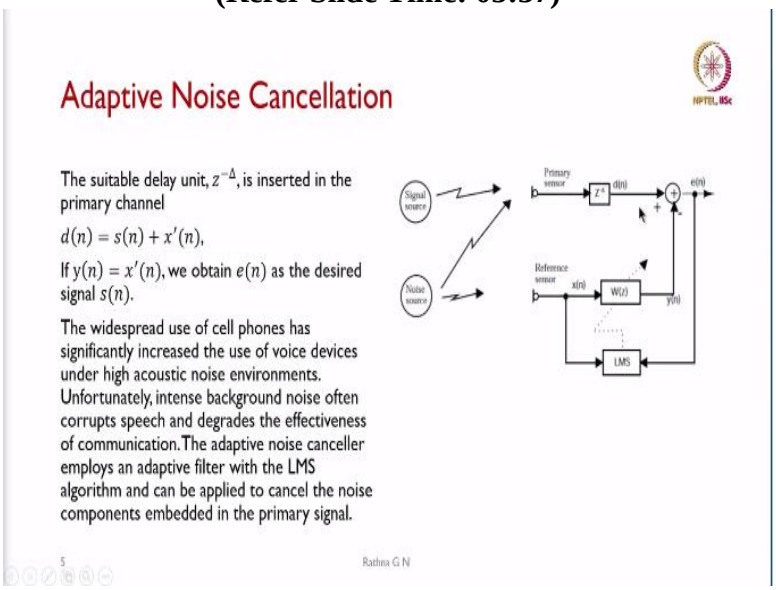


So, continuing with the predictor, so, we are going to consider an application of using an adaptive predictor to enhance the primary signal which consists of M sinusoids corrupted by white noise. So, which is expressed as $x(n) = s(n) + v(n)$ is white noise basically what it is being considered, then what happens to output $= \sum_{m=0}^{M-1} A_m \sin(\omega_m n + \emptyset_m) + v(n)$ because we have to considering the sine signal here.

So, $\sin(\omega_m n + \emptyset_m)$ is the phase of the signal $+v(n)$ where we have n is this 0 mean white noise with unit variance assumed as σ_v^2. So, then what is the thing is going to happen this is the error signal. So, you will be seeing that and then it starts coming down and then settles with whatever the value of it. So, now, you will be seeing that this is output $y(n)$ and $s(n)$ is our input.

So, you will be seeing that initially it was small after that they are going to merge and then go one another. So we will be able to generate whatever sine function what we have given the thing, so, we are able to predict this is output.

(Refer Slide Time: 05:37)



So, coming with next application is the adaptive noise cancellation. So why do we have to cancel will see the noise. So, here we are going to give a suitable delay that is unit delay of $z^{-\Delta}$ is inserted in the primary channel. So, here we are going to have this is the signal source and this is noise source and this is primary reference signal and here is the reference sensor, which is going to collect noise source. So, when we apply this, this we call it as the reference sources $x(n)$.

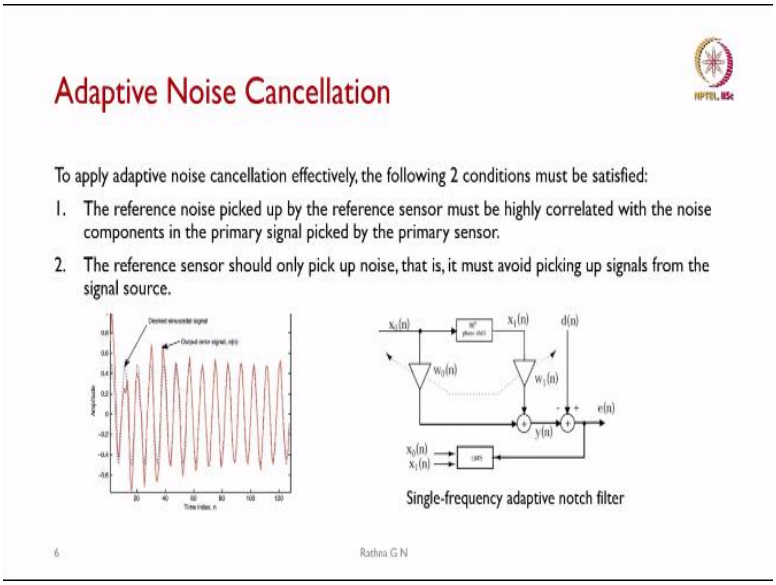
And then primary signal source whatever we are collecting through the primary source, which is going to be Δ delay, what we are going to get the thing and we say this is my desired signal what I want it and this will be adapting according to the LMS algorithm, which is $y(n)$ and then we will be subtracting desired signal with the output and then this error what we try to minimize and then which is fed back and then the weights are going to be updated based on error function.

So, how we are going to define this will be desired signal is nothing but $d(n) = s(n) + x'(n)$ basically. So, what is that, where if $y(n) = x'(n)$ we obtain $e(n)$ as the desired signal $s(n)$. So, what is it? The widespread use of cell phones has significantly increased the use of voice devices under high acoustic noise environments, most of you would have observed that when you are using your mobile in the completely noisy environment.

So, sometimes what you would try to do is close one of the ear and then try to hear from the other ear. So, that is what, what we are going to do the thing. So, how you can do adaptively noise cancellation within the mobile based on the surrounding noise. So, what happens? That is intense background noise often corrupts speech and then degrades the effectiveness of communication.

So, you will be using the adaptive noise canceller basically employs an adaptive filter with LMS algorithm and can be applied to cancel the noise components embedded in the primary signal. So, if we know the noise source and other things, so, we can capture and then try to subtract it so that we will be cancelling the noise and then output can be a clean speech signal.

(Refer Slide Time: 08:44)



So, as an example, so to apply adaptive noise cancellation effectively, so, these are the 2 conditions which has to be satisfied, what is it? That the reference noise picked up by the reference sensor must be highly correlated with the noise components in the primary signal picked by the primary sensor. So, because we know that primary data is getting corrupted with the noise, so, the noise sensor should pick up this noise from this thing and then it should be fed.

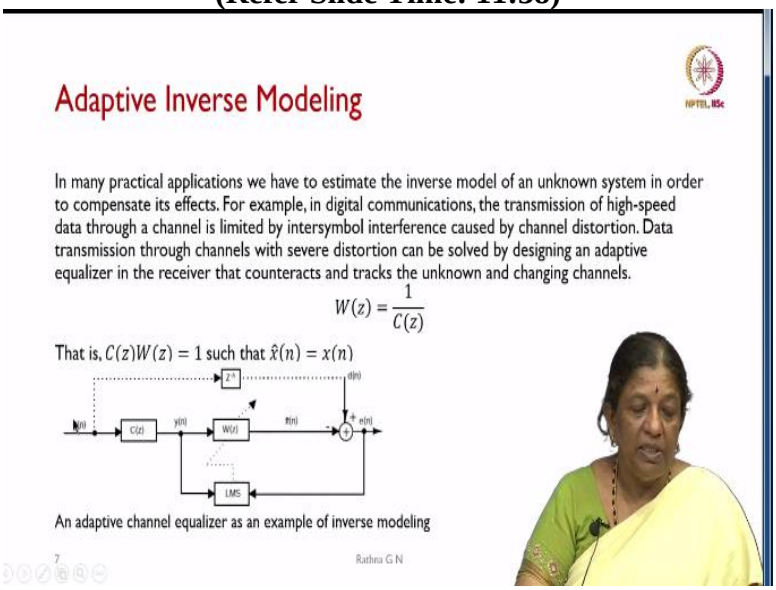
So, the reference sensor should only pick up noise that is it must avoid picking up signals from the signal source. So, you are going to have a contradiction in this case. So, one has to look at this. So, how it is going to be so, you will be seeing that $x_0(n)$ is the input this thing, phase shift that is we are going to give 90 degrees phase shift for the thing, which we call it as $x_1(n)$ and then you are seeing the FIR filter what we have used in this case to fine tune and then do the noise cancellation.

So $\omega_0(n)$ to $\omega_1(n)$ filter thing what you are looking at it that is second order filter what you have seen it. So, this is going to be fine tuned, based on what is the thing like LMS filter. So this is the desired signal what I have given as an input, and then based on this, we will be calculating y of n and then as usual, we will be subtracting it. So, now, what is this? Error signal is fed into LMS algorithm. So, this is going to take both $x_0(n)$ and then $x_1(n)$ as input.

And then adjust the weights based on the error only it is written little below so that we are not smudging or crossover of the line is going to happen from this input and this input and it is shown separately this is what we call it a single frequency adaptive notch filter what we have

designed in this case, so, you will be seeing that what is the desired signal is given here and then you will be seeing that output error signal e of n. So which is going to correspond to output basically, what we are going to get out of the thing? So noise cancellation is happened in this case, and then we will get the signal correctly.

(Refer Slide Time: 11:38)



The other one is how we are going to inverse model the channel in this case example is communication channel is one of the application what it will be considered. That is what, what it says in practical applications, we have to estimate the inverse model of an unknown system in order to compensate its effect as an example. So, in digital communications, the transmission of high speed data through a channel is going to be limited by intersymbol interference.

So, those who do not know the thing you can refer to what is intersymbol interference, which is caused by channel distraction. So, we know the frequencies of the channel which it is passing. So, if other frequencies overlap will have the intersymbol interference, so, how to distortion which is going to cause? And data transmission through channels with severe distortion can be solved by designing an adaptive equalizer in the receiver that counteracts the tracks the unknown and changing channels, how we are going to design this?

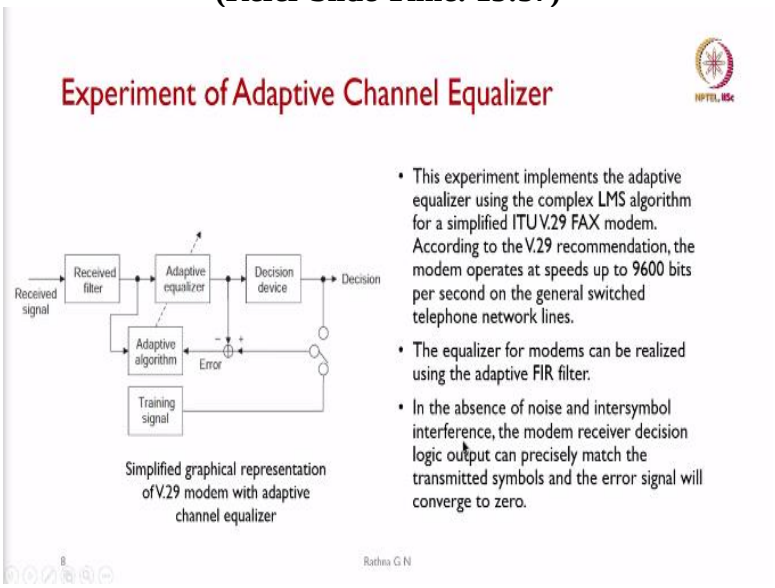
So, you will be seeing $W(z) = \frac{1}{C(z)}$. So, we are taking C we assume that $C(z)$ is the filter at the transmission end at the receiver end, you will be designing the inverse filter. So, one must be thinking how the inverse filter is going to be designed. So you should have what we call it as a minimum order filter basically. So, all the poles and zeros of the input filters should be inside our unit circle.

So that when we design the filter $\frac{1}{C(z)}$, so which is both zeros and then poles will be getting interchange. So, they should not be going out of the unit circle. So, this is what one has to design and then what we call that when we do the cross multiplication $C(z)W(z) = 1$ such that $\hat{x}(n) = x(n)$. So this is what shown in equalizer as an example of inverse modelling. So what is it?

So, we have the input $x(n)$ and then this is a $C(z)$ what we have the thing and then output is $y(n)$ and then which is a fed LMS algorithm as well as weight function for calculation. It passes through the filter here, and then the $\hat{x}(n)$ is the output what we are calculating and then this $x(n)$ is going to be delayed by Δ and then which is fed as desired signal. So when the difference is minimal then will be knowing that $\hat{x}(n) = x(n)$.

So, this is how we will be trying to get whatever we have passed through the communication channel will be received in this fashion. So, almost it is equivalent whatever we receive $\hat{x}(n)$ is equivalent to almost $x(n)$. And as usual error will be through the LMS algorithm and then taking the input from $y(n)$. So, we will be designing weight function.

(Refer Slide Time: 15:37)



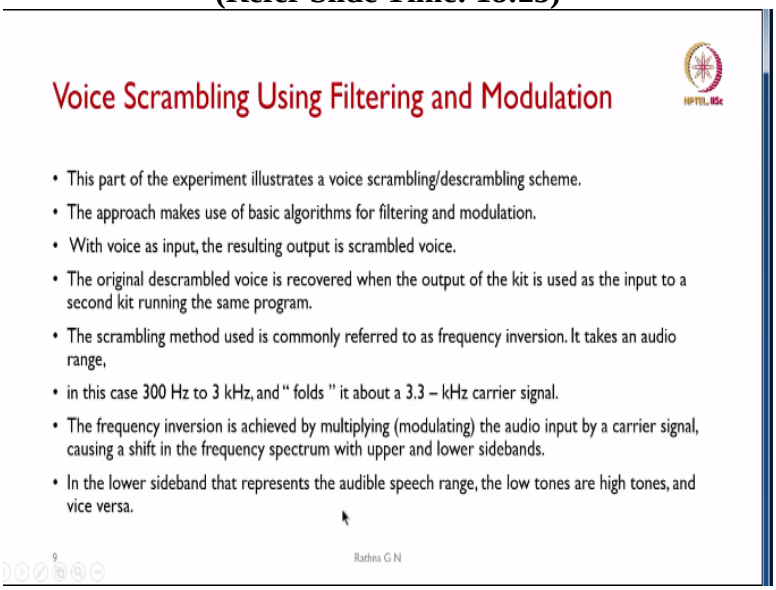
So, how we are going to do the channel equalization is shown in this because this is an experiment where it is going to be used adaptive equalizer using the complex LMS algorithm for a simplified ITU standard, communication standard what you will be taking it V.29 is a fax modem. So, for various other; this things we call it as audio or speech we have different standards. So, in this case for fax modem, this is the standard, one uses.

So, according to this V.29 recommendation, the modem operates at speed up to 9600 bits per second on the general switched telephone network lines and the equalizer for modems can be realized using our adaptive FIR filter what it is shown here. So, in the absence of noise and intersymbol interference, the modem receiver decision logic output can precisely match the transmitted symbols and the error signal will converge to 0 this is the ideal case one is considering then error can reach to 0.

So, what is this thing graphical representation as it is shown, so, we have the received signal and we have the received filter constructed here. So, the output of which is going to adaptive algorithm. And then the same thing will be going for the adaptive equalizer weights calculation and then we will be putting a decision device here depending on the thing, so, either the switch will be turned on to find out the error if it is this thing adapting to the channel noise.

And if there is no noise you can put the decision to this one, change over to here the output will be coming. So, here you will be passing it to the training of the signal if the decision is on the other side and you will be using this training the signal to adapt itself, so algorithm uses the thing. So we will be here also we will be minute trying to minimize output of the error. So this will be equalization output what we will be getting it?

(Refer Slide Time: 18:25)



So the next application what we call it as voice scrambling? So we will be using filtering and then modulation techniques here to do the scrambling of the voice, why one has to scramble the voice will see it. So, the approach makes use of basic algorithms for filtering and then

modulation with voices and input the resulting output is scrambled voice will get it. So, the original descrambled voice is recovered when the output of this from the kit if we are using the hardware.

Or if we are using the MATLAB the output of this is given to input to your second kid running the same program or in the same place we can run this program and then imitate that we are depicting that 2 places that is wherever we have recorded the voice we have scrambled it because I want to have it as a private one. So that I know how I have scrambled my speech or audio then at the receiving end, the person knows how the scrambling has happened. So we will use the same concept to record this.

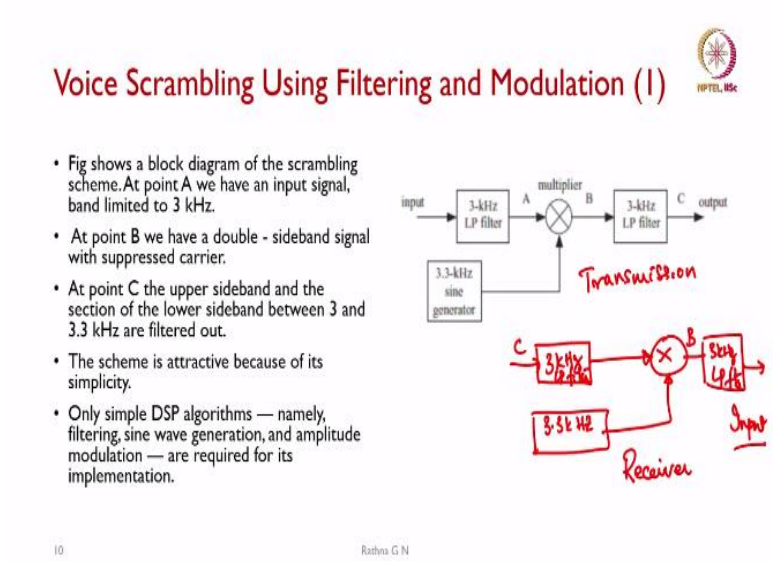
So that way, you will be calling it as a watermarking on speech and then audio signals. So how we can do it using scrambling we will see in the experiments both using MATLAB and then code composer studio. So even in the code composer studio, what you can use is if you have 2 kits, you can connect 1 kit to do the scrambling, the other kit will be rescrambling and then you can get the correct speech depicting both transmission and then receiving concept.

So, commonly referred to as frequency inversion in this case is going to happen, it takes an audio range in, that is 300 hertz to 3 kilohertz. So we call this as narrowband frequency what we will be using it and which will be folding, because we are going to scramble with whatever the input frequency, in this case we will be using 3.3 kilohertz as the carrier signal in this case, and will be merging with the thing will be folded and then it will be sent.

So the frequency inversion is achieved by multiplying or modulating the audio input by carrier signal, causing a shift in the frequency spectrum with the upper and lower sidebands. So we will be getting both the higher one with the 3 and then 3.3 it will be 9.9 kilohertz and then the difference will be 0.3 kilohertz. So, one will be upper and the other one will be lower one. So this, lower sideband that represent this audible speech range, the low tones or high tones, and vice versa what happens?

So, these low tones become high tones when we have received it, whereas the high tones will be converted into low tones. So this is what, what we will be getting it

(Refer Slide Time: 21:58)



As an example, how it is going to be done is shown in the figure. So I have a taken input as this thing audio input, which is sample with 8 kilohertz. So, or a speech signal, which is going to be sampled at 8 kilohertz, then will be representing between 300 to 3 kilohertz is speech null, which is going to come so then this has to be this thing, what we want is the frequencies about 3 kilohertz has to be cutoff. So we will be putting the low pass filter with the cutoff frequency of 3 kilohertz and then we call this as point A.

Then we take 3.3 kilohertz as carrier signal that is using the sine generator, which you will be multiplying with this thing input signal which is cutoff at 3 kilohertz, and we will be generating B. So this can be fed through the 3 kilohertz low pass filters, eliminating high frequencies, and we will be getting C as the scrambled signal as output. So, what it shows us scrambling at point A we have an input signal band limited to 3 kilohertz and at point B we have a double sideband signal with suppressed carrier basically.

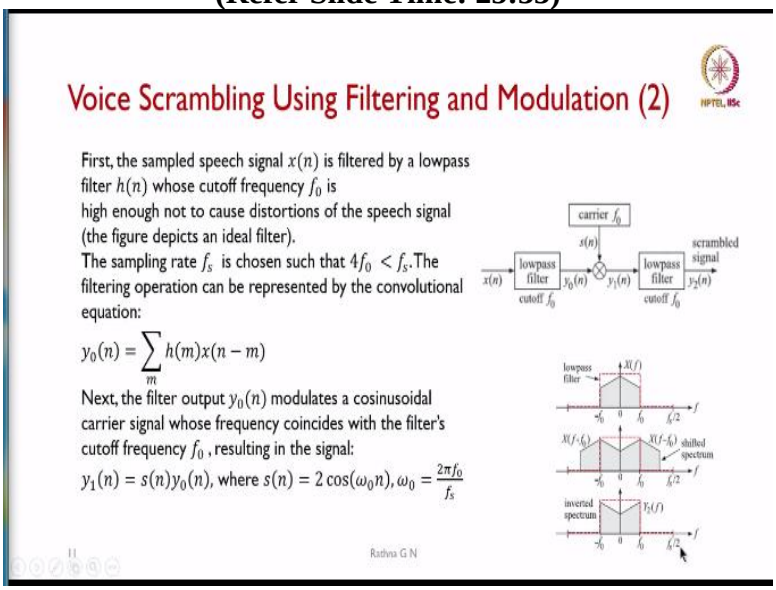
And then at point C the upper sideband and the section of the lower sideband between 3 and 3.3 kilohertz are filtered out. So the scheme is attractive because of its simplicity, only simple DSP algorithms namely filtering sine wave generation and then amplitude modulation are required for this implementations. So when we come to the reverse of it, we can rewrite the C can be given to as an input signal. I will write instead of my input C is the one and same thing C, 3 kilohertz a low pass filter, we can design the thing.

And then this is given to multiplier and then same reference signal what I am going to keep it as 3.3 kilohertz. So, this goes to this thing, make a multiplier here. And then the output B is

going to be passed through 3 kilohertz low pass filter. So output will be equivalent to input whatever we have given at the transmission end this is the thing is going to happen that I will put it as this is transmission and here at the receiver when you want to recover your voice or your audio if you want to protect your thing, this is the way one can do it.

And then you can add whoever wants to receive it you can tell them how you have done your modulation. So, they can depict this in hardware and then at the receiver they will be getting whatever input you are intended for that particular person. So that is how the scrambler is going to work.

(Refer Slide Time: 25:33)



So continuing with the thing, how it is going to we will be seeing how the carrier frequency and then low pass filter will be mixed and then how our filter response is going to be in the frequency domain is shown here. So, what is going to happen? First, the sample speech signal is filtered by a low pass filter we said $h(n)$ is the order of the filter whose cutoff frequency what we will be calling it as f_0 is high enough not to cause distortions of the speech signal.

So, that is the ideal filter what we are going to call it here as you can see in the figure, this is the low pass filter and then what is it? Its band is $-f_0$ to f_0. And then what happens to the thing the sampling rate f_s is the sampling frequencies chosen such that $4f_0 < f_s$. So, that means to say that your sampling frequency has to be greater than 4 times of the frequency component what you want to pass.

So, the filtering operation can be represented by the convolutional normal expansion. So, we have $y_0(n) = \sum_m$ to order of the filter you can take it as $l - 1$ if it is the lth order or mth order $m - 1$. So, we will have $h(m)x(n-l)$. So, if you have taken L as the thing, so, it will be coming $h(l)x(n-l)$ is convolution thing. So, the notation can be any one of it what was you.

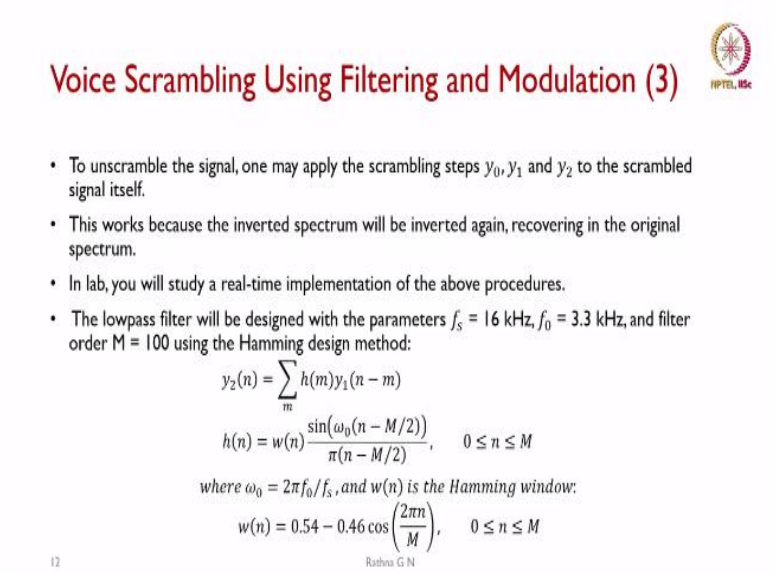
So, the filter output y naught n gets modulated as this thing cosinusoidal carrier signal whose frequency coincides when the filter is cutoff frequency f_0 resulting in the signal as shown here, what is it? $y_1(n)$ as you can see that $x(n)$ which is, uses a low pass filter with cutoff frequency f_0 which will be equal to $y_0(n)$ and then this is the carrier frequency what you have chosen as f_0 . So, which is that is desired signal s of n what you will be putting it.

So, you will be multiplying them and $y_1(n)$ is the output of it what are the things it contains. So, $s(n)y_0(n)$ where your $s(n) = 2\cos(\omega_0 n)$, where $\omega_0 = \frac{2\pi f_0}{f_s}$. So, then you will be passing it through your low pass filter with cutoff frequency as same as this one f_0 to eliminate the higher frequency component present in it and then output is $y_2(n)$ is the scrambled signal.

So, when you can see that in the frequency domain how the thing is going to be getting shifted, what you can see it? So, that is $f - f_0$ is the shift what you have given the thing. So, the signal you will be seeing that it is shifted spectrum what you are seeing in this case and then when you take the inversion at the receiving signal, so, this is how the inverted signal looks like this is $y_2(n)$ here in the with respect to frequency domain. So, this is $-f_0$ to f_0 and we have $f_s/2$ what it is marked in this case?

So, this is how what you have to say that your $f_s > 4$ times the frequency component of your basically filter cutoff frequency f_0 of the filter what one has to assume.

(Refer Slide Time: 29:47)



So, continuing with the thing how it is going to represent it? That is unscramble the signal we already have put the thing one may apply the scrambling steps y_0, y_1 and y_2 whatever it was shown to the scrambled signal itself. So, this works because the inverted spectrum will be inverted again recovering in the original spectrum. So in the lab, so, you will be studying a real time implementation of the above procedures.

So we will be demonstrating so, those who are interested can look into the code and other things they can run your own algorithm to implement this scrambling and then descrambling. So, the low pass filter will be designed with parameters that is f_s = 16 kHz, f_0 = 3.3 kHz. So, approximately as you can take it as 4 times the thing. So, if you assume 4 kilohertz as the thing, f_0 so, 4 into 4 will be 16 kilohertz. So, you are choosing your sampling frequency as 16 kilohertz.

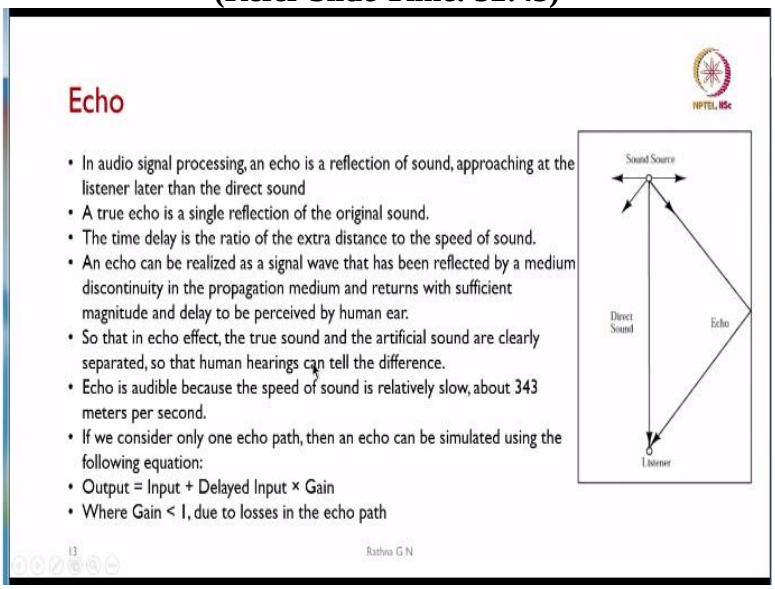
And the filter order if you use M = 100 using the Hamming window, which is given down here that is w of n is the Hamming window you will be using it this is the equation for designing the Hamming window. So, which is $w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{M}\right)$, n is going to be your input sample. So, when you pass through that, this is passing through your Hamming window. So, convolution equation is given as $y_2(n)$ is output.

So, this will be depending on your order of m in this case 0 to 99 what it is going to be $h(m)y_1(n-m)$ and then $h(n)$ is going to be impulse response has to be restricted with respect to window because this is a continuous minus infinity to infinity what we assume so, this is

$w(n)\frac{\sin(\omega_0(n-M/2))}{\pi(n-M/2)}$. So, here $0 \leq n \leq M$ basically and $\omega_0 = 2\pi f_0/f_s$ in this case it is 3.3 divided by 16 kilohertz.

And $w(n)$ is the Hamming window which is given by this equation. So, $0 \leq n \leq M$ whatever we have designed in this case we have assumed it as 100.

(Refer Slide Time: 32:43)



So, the next application what we will be doing it is echo. So, here we will be doing the generation in the next class will take up the cancellation. So, how the echoes are going to be generated? First synthetically we can do the generation. So, most of you would have visited where you have rocks and other things, when you speak so, you will be hearing the echo coming back to you your own voice goes and reflects that is what, what we say reflection of sound. So, which is going to come back to you a little bit delayed.

How we can synthetically generate? So, using code what will be looking at it, so, approaching at the listener later than that direct sound. So, as you can see is this is the sound source. So, you will be seeing that all the places it will be reflected your sound when you are speaking and there is a hard surface that is what, what you will be seeing it what happens it goes and hits and then it come back to you.

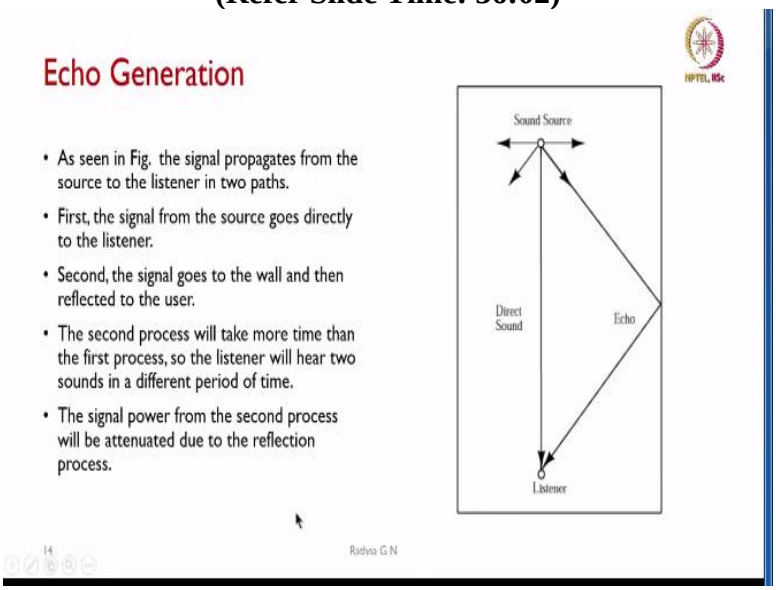
The first one is the direct sound what your it is coming to you the later on you will be getting the reflected sound to your listener this we call it as a echo basically, what it says a true echo is single reflection of the original sound. So, that is what, what we call it as echo we will see if

multiple reflection comes what we call in the next slide. The time delay is the ratio of the extra distance to the speed of sound.

An echo can be realized as a signal wave that has been reflected by medium which has the discontinuity in the propagation medium and returns with sufficient magnitude and delay to be perceived by human ear, some places it gets absorbed then you might not be able to hear it as echo. So where you have sufficient magnitude and then delay, then we will be hearing it as an echo. So that, in echo effect, the true sound and the artificial sound are clearly separated. So that human hearings can tell the difference.

So, Echo is audible because the speed of sound is relatively slow, about 343 meters per second. And if we consider only one echo path, then an echo can be simulated using the following equation. So that is output is the echoed output. So which is going to have the input plus delayed input with gain what we can get the echo signal. So, we assumed gain should be less than 1 in this case, because we have the magnitude of input is 1. So, we assumed this is less than 1 due to losses in the echo path that is what we are going to imitate and then look at it.

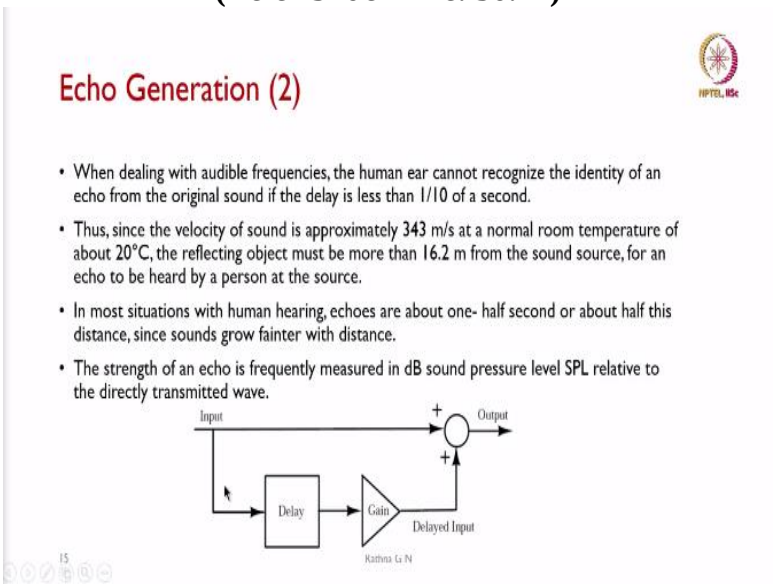
(Refer Slide Time: 36:02)



So, this is again continuing with the generation. So, as it is seen that signal propagates from the source to the listener in 2 paths. In this case, we have assumed that this is going to reflect and these are going to be absorbed only one path get reflected as I said, signal from the source goes directly to the listener, second signal goes to the wall and then reflected to the user. So, the second process will take more time than the first process. So, the listener will hear 2 sounds in

a different period of time. And the signal power from the second process will be attenuated due to the reflection process, this is what we discussed.

(Refer Slide Time: 36:44)



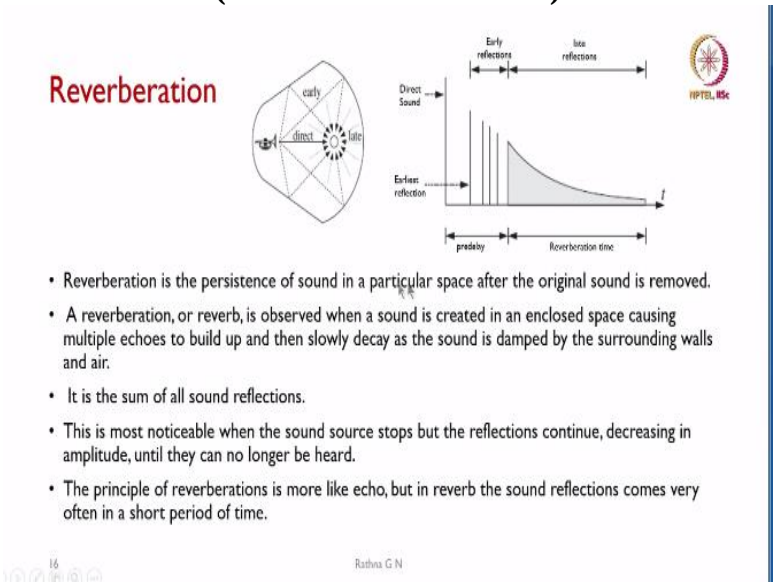
So, how we can digitally generate an echo by showing in this diagram, so how we are going to do the thing? So, when dealing with audible frequencies, human ear cannot recognize the identity of an echo from the original sound, if the delay is less than one tenth of a second. So as you can see that if it is within this then you may not hear it as an echo. Thus, since the velocity of sound is approximately what we call it as 343 meter per second at a normal room temperature of about 20 degrees centigrade.

The reflecting object must be more than 16.2 meters from the sound source for an echo to be heard by a person at the source. So these are the things then only you will hear you would be wondering wherever I speak I should be able to hear my voice back. So most of the you will be seeing sound recording rooms you will be having what we call it as a echo cancellers. So, padding with not to reflect the sound so it is going to absorb so that only the pure voice is going to be heard.

So you as an example, even I am sitting in a room where recording room where my voice is going to be only heard and then outside noise and everything is cancelled, even whenever my voice hits the wall, so you will not be getting any reflection in this place. So in most situations, with human hearing echoes are about one half second or about half this distance, since sounds grow fainter with distance.

So the strength of an echo is frequently measured in dB sound pressure level, what we call it as SPL sound pressure level relative to the directly transmitted wave. So we will be generating in the lab that is input is given. So, the same input will delay we can see that by varying the delay, how we will be perceiving it? And then we are going to multiply with the gain and then this is the delayed input which gets added with the normal input and output what we will be hearing it as the echoed signal in the output.

(Refer Slide Time: 39:17)



So, the next one is application is the reverberation. As we said you can see that there is a sound signal here. So you will be seeing that one can be direct. And then there are multiple reflections in this case. That is what I said it is a single reflection what we call it as echo when it happens with the multiple reflections. So, what you will be seeing that this may be coming early, and then the other after hitting here it may be coming these are the late once what you will be getting it this may come early.

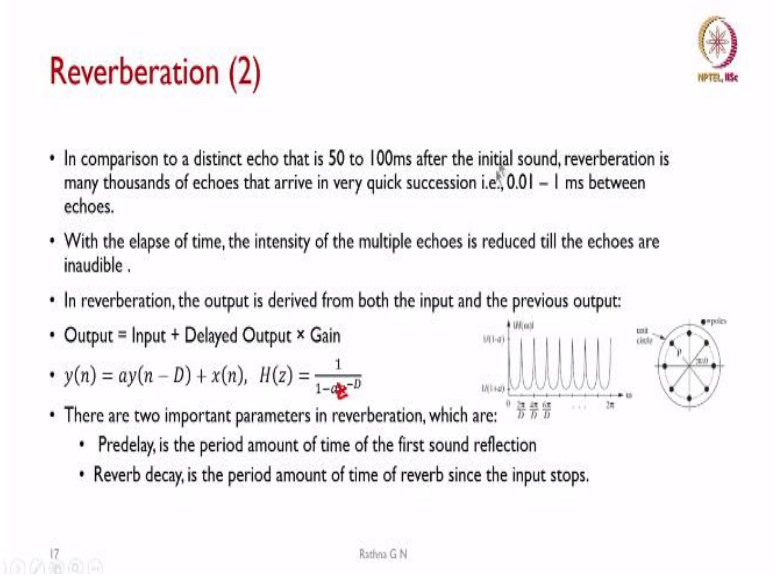
So you will be hearing multiple this thing reflections that is what, what it says reverberation is the persistence of sound in a particular space after the original sound is removed. And in the reverberation or reverb what we call it is observed when a sound is created in an enclosed space causing multiple echoes to build up and then slowly decays the sound is damp by surrounding walls and then air it is the sum of all sound reflections what will put it.

Most noticeable when the sound sort of stops, but the reflections are going to continue decreasing in amplitude until can no longer be heard. So, the principle of reverberations is more like echo but in reverb the sound reflections comes very often in a short period of time. So, you

will be seeing that this is the direct sound what is happening and then these are the earlier reflected what you will be seeing it this we call it as during the predelay what you will get it and then these are the early reflection time from here to here, what will be perceiving it.

After that these are the late reflections, you will be seeing that it is slowly dying it out. So this is the reverberation time what we call it when it is going to down slowly.

(Refer Slide Time: 41:25)



So considering the thing comparison, to echo that is 50 to 100 millisecond after the initial sound reverberation is many 1000s of echoes that arrive in very quick succession that is 0.01 to 1 millisecond between echoes. So with the elapse of time, the intensity of the multiple echoes is reduced till the echoes are inaudible. So in reverberation how we are going to generate the thing output will be input plus delayed output into gain.

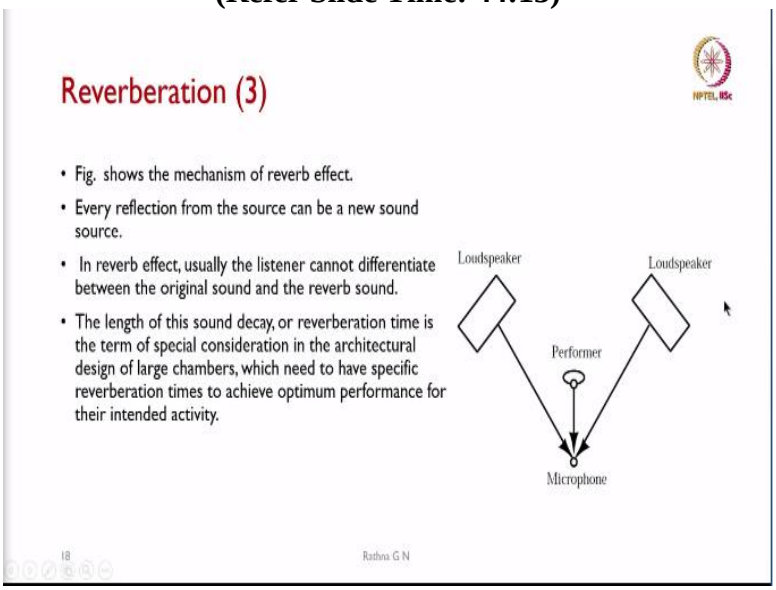
So in this case, we will be considering the feedback signal that is $ay(n - D)$ output is getting delayed by D $+x(n)$. So then the response of the system is given by $H(z) = \frac{1}{1-az^{-D}}$. This is the delay in the zee domain. There are 2 important parameters reverberation, which are the ones one is the predelay as we have seen in the previous case, we have the predelay and then we have the amount of time of the first sound reflection.

Then the reverb decay, is the period amount of time that reverberation since the input stops. So, we have seen that this the reverberation, this is the predelay, which is the early reflection which is going to come then how does it look like because we have said $(n - D)$ delays what

we are going to have it so, this each reflection we will be saying that they are spaced at $2\pi/D$. So, I will be getting a peek here, next peak is going to be $4\pi/D$ and $6\pi/D$ and so on till 2π.

So, and then we will be seeing that this is unit circle and then we have this thing $2\pi/D$ spacing what we are going to have it and rho will be the radius of the where the poles have been located. So, you are seeing that poles are all around the circle. And that is what, what it says is these are the poles what you will be putting across the thing at distance of $2\pi/D$.

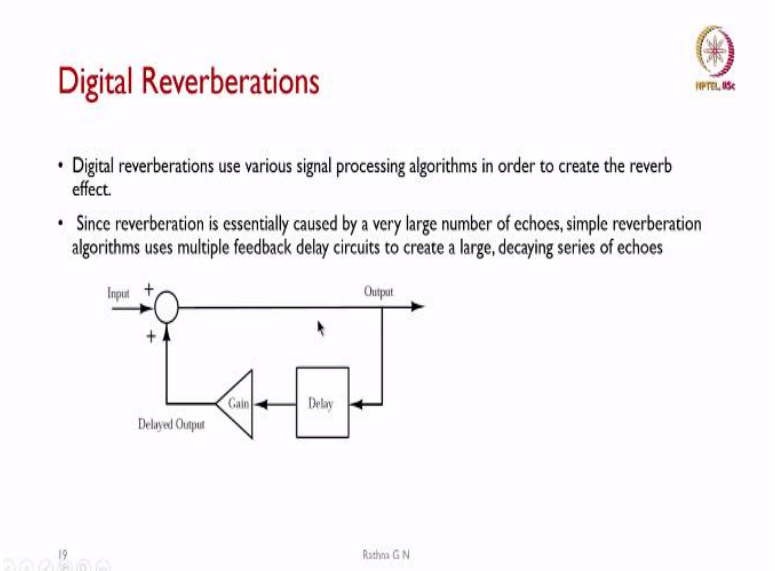
(Refer Slide Time: 44:13)



So, coming with the reverberation this shows the thing that is this is microphone what we have it and the performer is here from here to microphone, comes direct and we know that loudspeakers are in the sideways. So, sometimes as you will be seeing that even in the mobile or when we are switching on for conference call or something will be telling if there is one more loudspeaker we have it.

Or this thing input will ask them to switch it off and keep only one performer receiving and also both Mic and speaker should be switched on. If both the speakers are on even now, you would be hearing that the echo will be coming. So, whenever because I think pandemic has taught us very nicely that online classes and other things. So, when 2 devices are on side by side will say that please mute the other device and only one devices of the 2 of them are sitting together be on so that we will not have any echoes or we call it as reverberation coming into picture.

(Refer Slide Time: 45:26)



So, how to generate just like our echo digital reverberation is shown in this figure, this is our input and then the output is going to be delayed and then fed back there in the echo case input itself is delayed and then fed into system here the output is going to be delayed and then fed with into the system with gain actually. So this is delayed output, what we will be hearing that reverberation so you have to differentiate between an echo and then reverberation when we run the example.

So thank you very much in the next class we will be considering echo cancellation and then equalizer. Why do we need it? So what we will be taking it up in the next class. Thank you for listening.