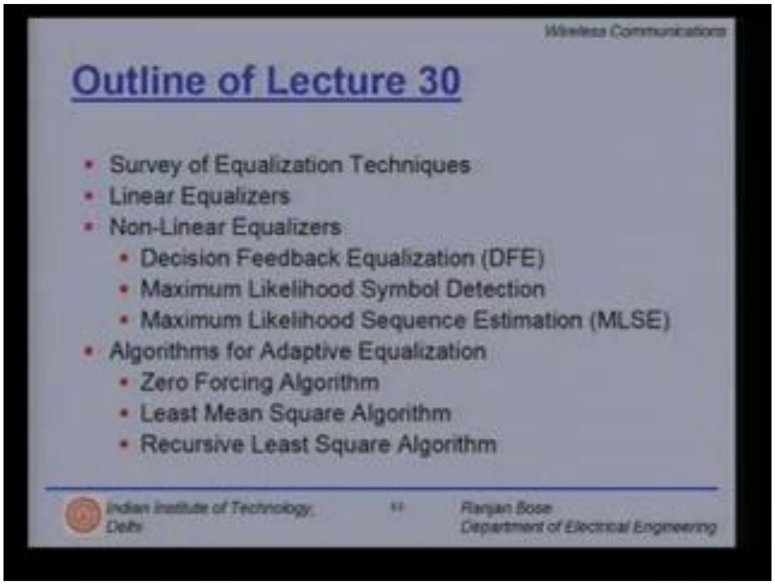
Wireless Communications

Dr. Ranjan Bose
Department of Electrical Engineering
Indian Institute of Technology, Delhi
Lecture No. # 30

Equalization and Diversity Techniques for Wireless Communications (Continued)

Welcome to the next lecture on wireless communications. Today we will look at equalization techniques, first a brief outline for today's talk.

(Refer Slide Time: 00:01:27 min)



Wireless Communications

Outline of Lecture 30

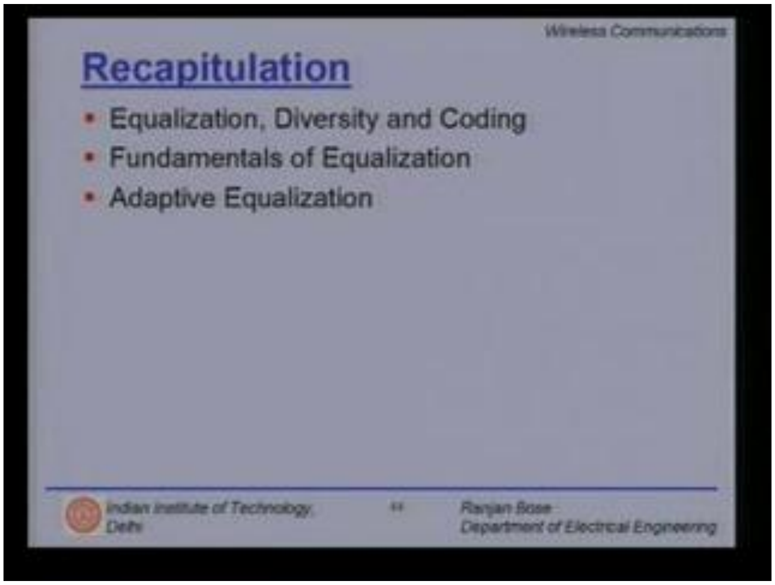
- Survey of Equalization Techniques
- Linear Equalizers
- Non-Linear Equalizers
  - Decision Feedback Equalization (DFE)
  - Maximum Likelihood Symbol Detection
  - Maximum Likelihood Sequence Estimation (MLSE)
- Algorithms for Adaptive Equalization
  - Zero Forcing Algorithm
  - Least Mean Square Algorithm
  - Recursive Least Square Algorithm

Indian Institute of Technology, Delhi
Ranjan Bose
Department of Electrical Engineering

We will start with a brief survey of equalization techniques followed by a study of linear equalizers then we will move into the domain of non-linear equalizers. We will look at the decision feedback equalizer or DFE. Then the maximum likelihood symbol detection and finally the maximum likelihood sequence estimation. Finally we will talk about algorithms for adaptive equalizations, the three important algorithms that we will study are the zero forcing, the least mean square or LMS and finally the recursive least square algorithm. So this is the brief outline for today's talk. Let's recap what we have learnt so far.

In the previous lecture we saw that the channel impediments because of multipath and hostile mobile environments can be overcome using one of the three techniques equalization, diversity and error control coding. However we can use them individually or in tangle. Then we looked at the basic fundamentals of equalization followed by a brief structural investigation of the adaptive equalization technique.

(Refer Slide Time: 00:02:17 min)



Wireless Communications

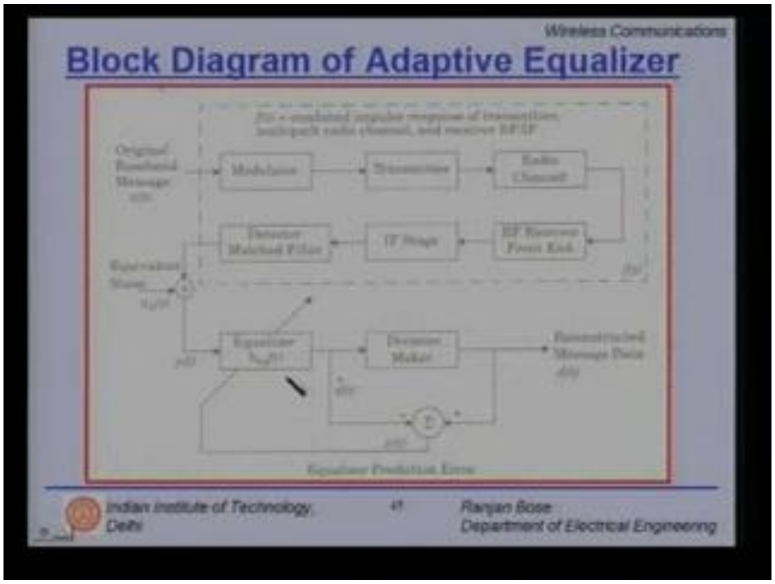
Recapitulation

- Equalization, Diversity and Coding
- Fundamentals of Equalization
- Adaptive Equalization

Indian Institute of Technology, Delhi 44 Ranjan Bose Department of Electrical Engineering

Now what is the block diagram for a generic adaptive equalizer? We saw last time that the original baseband signal x(t) goes through a modulator then through a transmitter finally through the radio channel wherein there is a distortion and noise and also fading effectswhich is received back at the receiver, RF receiver front end down converted to IF stage followed by the detector matched filter. Finally there is an equivalent noise added and it goes through the equalizer, the job of the equalizer is to undo the effect of intersymbol interference. Now please note that this reconstructed message data dt that you saw last time is an estimate and should match x (t). If we use dt and use it in a feedback loop so adjust the weights of the equalizer we get into the domain of non-linear equalizer.

(Refer Slide Time: 00:02:54 min)



Wireless Communications

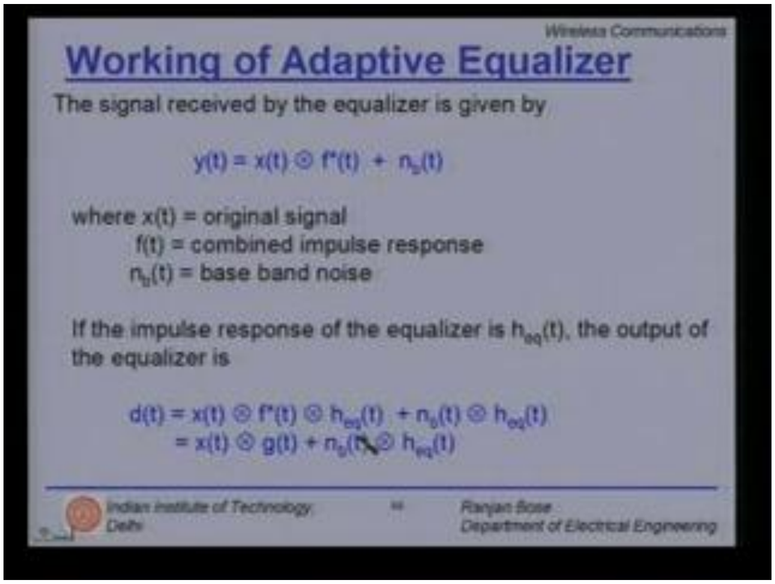
Block Diagram of Adaptive Equalizer

f(t) = combined impulse response of transmitter, multipath radio channel, and receiver RF/IF
Original Baseband Message x(t)
Modulator
Transmitter
Radio Channel
RF Receiver Front End
IF Stage
Detector Matched Filter
f(t)
Equivalent Noise $n_b(t)$
y(t)
Equalizer $h_{eq}(t)$
Decision Maker
Reconstructed Message Data d(t)
$\hat{d}(t)$
Σ
e(t)
Equalizer Prediction Error

Indian Institute of Technology, Delhi 45 Ranjan Bose Department of Electrical Engineering

However if this feedback is not there we just have a feed forward loop then it is a linear equalizer, we will talk about these two in greater detail today.

(Refer Slide Time: 00:04:15 min)



Wireless Communications

Working of Adaptive Equalizer

The signal received by the equalizer is given by

$$y(t) = x(t) \otimes f^*(t) + n_b(t)$$

where $x(t)$ = original signal
$f(t)$ = combined impulse response
$n_b(t)$ = base band noise

If the impulse response of the equalizer is $h_{eq}(t)$, the output of the equalizer is

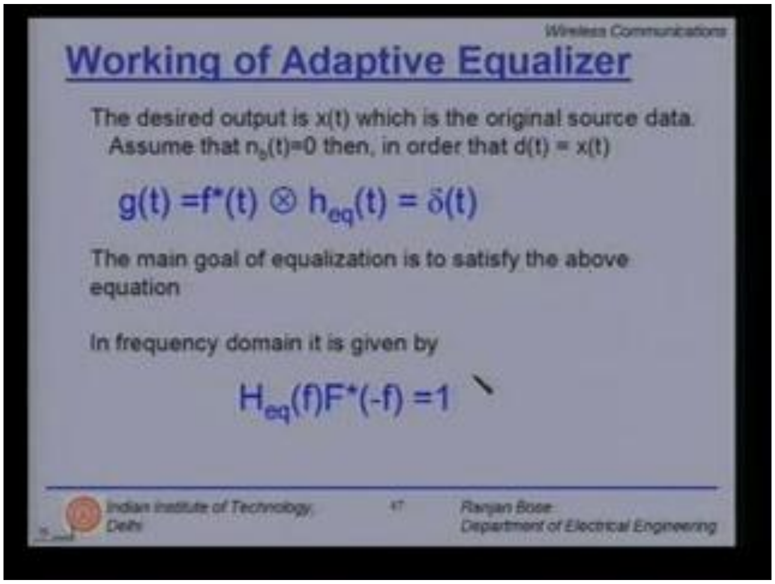
$$\begin{aligned} d(t) &= x(t) \otimes f^*(t) \otimes h_{eq}(t) + n_b(t) \otimes h_{eq}(t) \\ &= x(t) \otimes g(t) + n_b(t) \otimes h_{eq}(t) \end{aligned}$$

Indian Institute of Technology Delhi

Ranjan Bose
Department of Electrical Engineering

Now last time we also saw the simple working of an adaptive equalizer why, adaptive because the channel is changing with time and so the weights must be updated periodically. So the received signal y(t) is the convolution of x(t) and the composite impulse response $f^*(t)$ plus added with the baseband noise n_b, everything is analog in time domainand we saw that if the impulse response of the equalizer is h equalizer t, the output of the equalizer is simply given by this equation x(t) convolved with an effective g(t) plus n_b (t) convolved with h equivalent t.

(Refer Slide Time: 00:05:10 min)



Wireless Communications

Working of Adaptive Equalizer

The desired output is x(t) which is the original source data. Assume that $n_b(t)=0$ then, in order that $d(t) = x(t)$

$$g(t) = f^*(t) \otimes h_{eq}(t) = \delta(t)$$

The main goal of equalization is to satisfy the above equation

In frequency domain it is given by

$$H_{eq}(f)F^*(-f) = 1$$

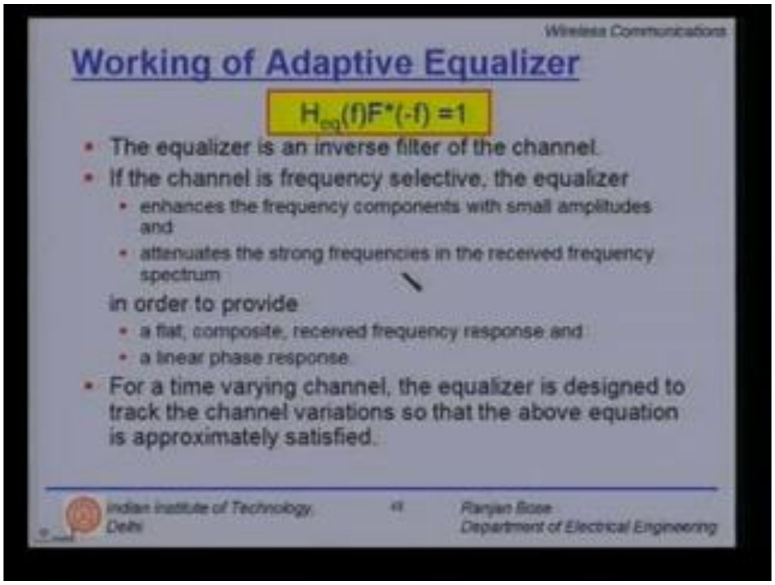
Indian Institute of Technology Delhi

47

Ranjan Bose
Department of Electrical Engineering

If you do the basic mathematics, we saw that in time domain in order that the equalizer must work in the absence of noise n_b (t) is equal to zero, in that ideal scenario only, when only the distortions are because of channel effects, you have g (t) should be equal to the impulse function delta t and in the frequency domain it should be H equivalent F^*(-f) =1. So if you have this condition satisfied in the frequency domain, then you are in business you can equalize the channel. In a Layman's language equalization undo's what the channel does, it is an inverse function it is an inverse filter which has a problem also because if you have deep nulls in the frequency spectrum of the channel then you tend to amplify the noise in those areas as well and therefore they need for a non-linear equalizer.

(Refer Slide Time: 00:06:15 min)



Wireless Communications

Working of Adaptive Equalizer

$$\boxed{H_{eq}(f)F^*(-f) = 1}$$

- The equalizer is an inverse filter of the channel.
- If the channel is frequency selective, the equalizer
  - enhances the frequency components with small amplitudes and
  - attenuates the strong frequencies in the received frequency spectrum

  in order to provide
  - a flat, composite, received frequency response and
  - a linear phase response.
- For a time varying channel, the equalizer is designed to track the channel variations so that the above equation is approximately satisfied.

Indian Institute of Technology Delhi

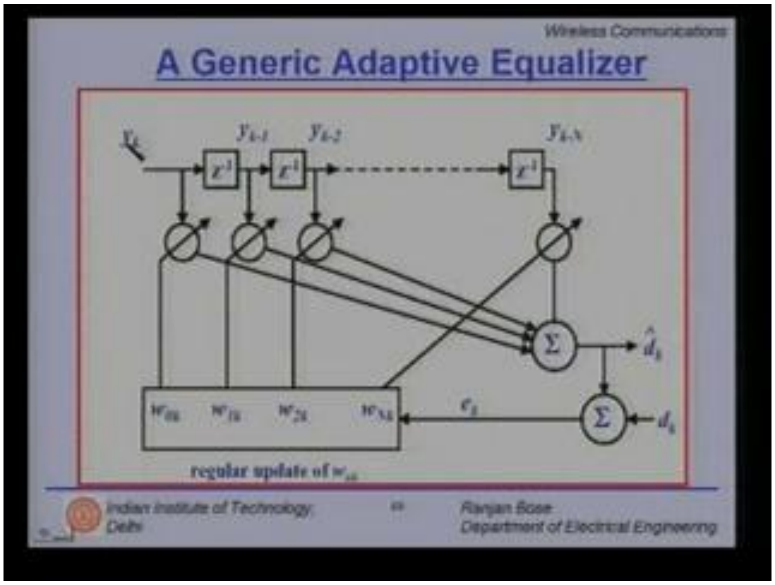
Ranjan Bose
Department of Electrical Engineering

So we realize that equalizer is in fact an inverse filter of the channel but in the absence of noise. So clearly if noise is present there will be some predictive error which you cannot remove completely. If a channel is frequency selective what does the equalizer do? It tries to undo the effects of channel that is it enhances the frequency components with small amplitudes and at the same time attenuates the strong frequencies in the received frequencies spectrum.

So it reverses whatever is done by the channel and this is done in order to provide a flat composite received frequency response as well as a linear phase response. Now mobile channels are time varying, so for a time varying channel the equalizer is designed to track the channel variations so that the above equation given here in frequency domain is approximately satisfied.

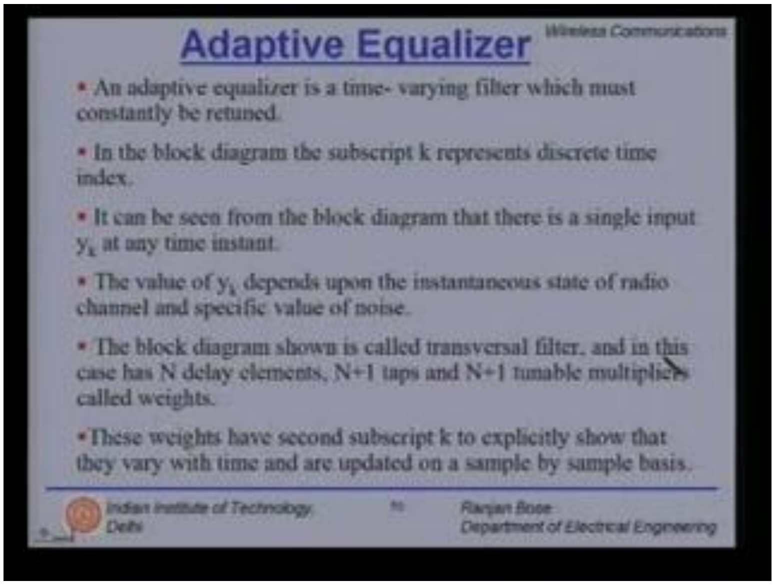
In this slide given below we have a generic block diagram for an adaptive equalizer. Please note that the weights here have two indices 0,1,2,3,4 for the different weights and the K represents the time index, Z inverse implies the delay y_k is in the input and this is multiplied by the weights added to get an estimate d hat k.

(Refer Slide Time: 00:07:29 min)



So an adaptive equalizer is a time varying filter which must be constantly retuned. How do we retune it? We will look at some popular algorithms today. In the block diagram the subscript K represents discrete time index. It can be seen from the block diagram that there is a single input y_k at any time instance and rest are the delayed versions. Exactly how many delay delays do we need, how many tap points do we need? It depends on what is the root mean squared delay spread of a channel.

(Refer Slide Time: 00:08:03 min)



Adaptive Equalizer

Wireless Communications

- An adaptive equalizer is a time- varying filter which must constantly be retuned.
- In the block diagram the subscript k represents discrete time index.
- It can be seen from the block diagram that there is a single input y_k at any time instant.
- The value of y_k depends upon the instantaneous state of radio channel and specific value of noise.
- The block diagram shown is called transversal filter, and in this case has N delay elements, N+1 taps and N+1 tunable multiplies called weights.
- These weights have second subscript k to explicitly show that they vary with time and are updated on a sample by sample basis.

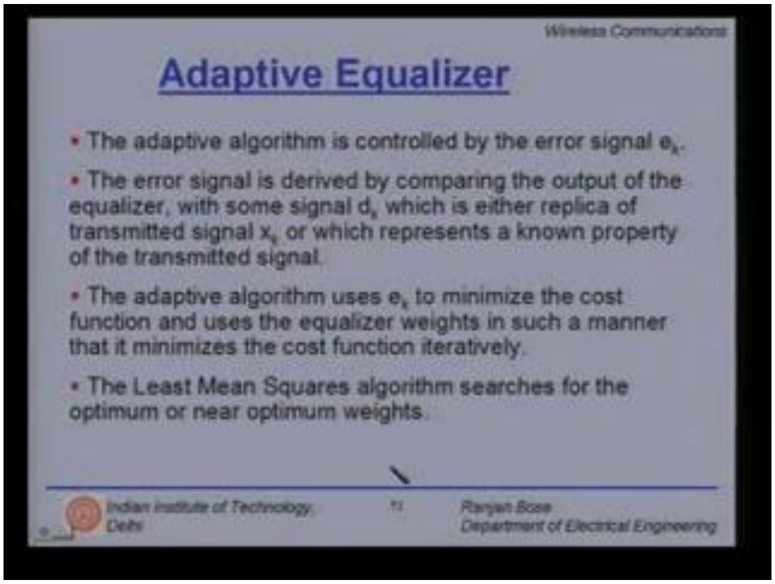
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

If you can have a very large root mean square delay spread of the channel then you need to have more number of delays and more number of tap points. It can only equalize to the delay built into the equalization. So that is a direct correlation between the delay spread of the channel and how many weights in the equalizer and what are the delays here should there be. In fact this necessitates the need to have some kind of an adaptive equalizer where the channel if it has a very large delay spread I should be able to use more number of delay elements and more number of tap points and then when my mobile moves to a region where the delay spread is much shorter, I should be able to undo some of the stages and retain only the necessary number of tap points.

We will see today that there is one kind of an implementation that precisely does that but in general the block diagram shown just before which is called transversal filter, it has N delay elements and hence N +1 taps and consequently N +1 tunable multipliers which we will call weights. Now these weights can be complex weights. As pointed out earlier the weights have a second subscript K to explicitly show that they vary with time and must be updated on a sample by sample bases or block by block bases that is for the whole weight vector. Now how do we retune the weights?

(Refer Slide Time: 00:10:39 min)



Wireless Communications

Adaptive Equalizer

- The adaptive algorithm is controlled by the error signal e_k.
- The error signal is derived by comparing the output of the equalizer, with some signal d_k which is either replica of transmitted signal x_k or which represents a known property of the transmitted signal.
- The adaptive algorithm uses e_k to minimize the cost function and uses the equalizer weights in such a manner that it minimizes the cost function iteratively.
- The Least Mean Squares algorithm searches for the optimum or near optimum weights.

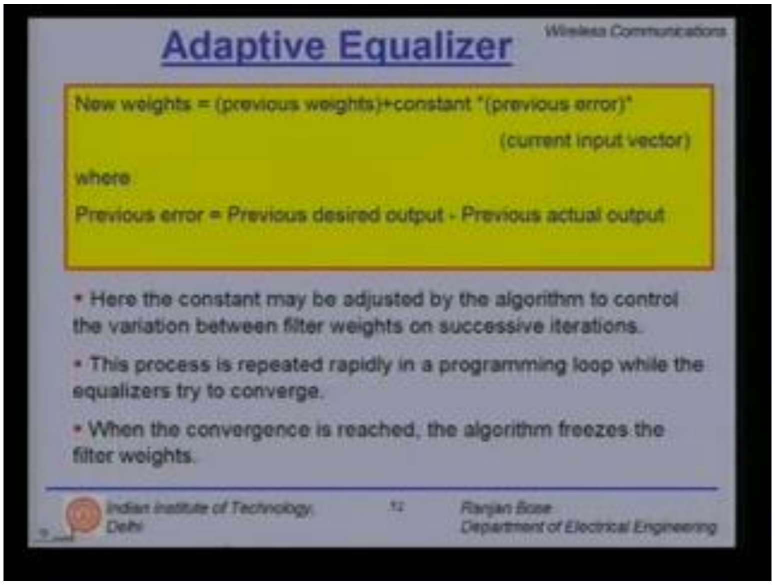
Indian Institute of Technology, Delhi

11

Ranjan Bose
Department of Electrical Engineering

The adaptive algorithm is controlled by the error signal e_k. The error signal is derived by comparing the output of the equalizer with some signal d_k which is either a replica of the transmitted signal x_k or which represents a known property of the transmitted signal, for example constant amplitude. The adaptive algorithm uses this e_k to minimize the cost function and uses the equalizer weights in such a manner that it minimizes the cost function iteratively. So it is upon you to define a cost function and today we will look at a couple of definitions of different kinds of cost functions. One of them the least mean square algorithm searches for the optimum or near optimum weights. The cost function clearly is the least mean square error.

(Refer Slide Time: 00:11:42 min)



Wireless Communications

Adaptive Equalizer

New weights = (previous weights)+constant *(previous error)* (current input vector)

where

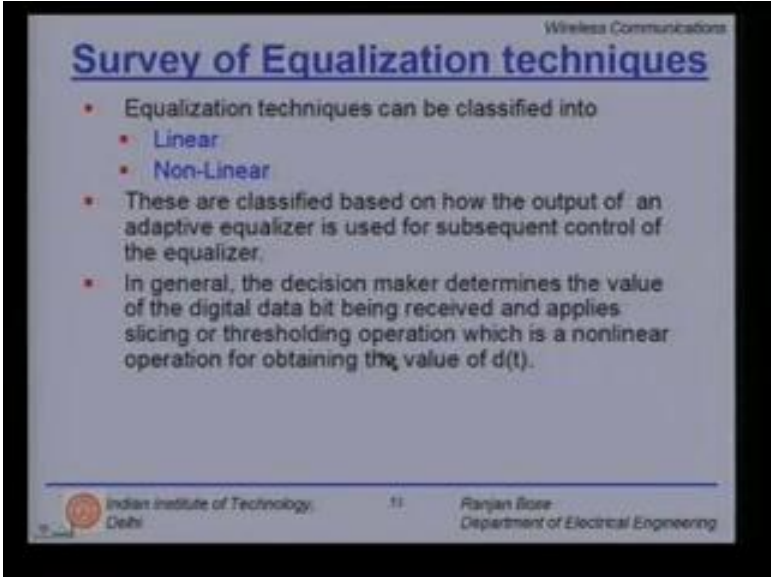
Previous error = Previous desired output - Previous actual output

- Here the constant may be adjusted by the algorithm to control the variation between filter weights on successive iterations.
- This process is repeated rapidly in a programming loop while the equalizers try to converge.
- When the convergence is reached, the algorithm freezes the filter weights.

Indian Institute of Technology, Delhi 12 Ranjan Bose Department of Electrical Engineering

So in an adaptive equalizer the new weight is updated as follows, new weights is previous weights plus some constant gain times previous error times current input vector is one possible implementation where previous error is previous desired output minus previous actual output. The constant shown here may be adjusted by the algorithm to control the variation between the filter weights on successive iterations. The process is repeated rapidly in a programming loop while the equalizers try to converge. Now please note how soon does it converge because only then I can have an good estimate of what was sent, if it takes too long to converge then I may not be able to use it in real time. This is one of the factors which will help in deciding which kind of equalizer to pick up, so rate of convergence. When the convergence is reached the algorithm freezes the filter weights.

(Refer Slide Time: 00:12:58 min)



Wireless Communications

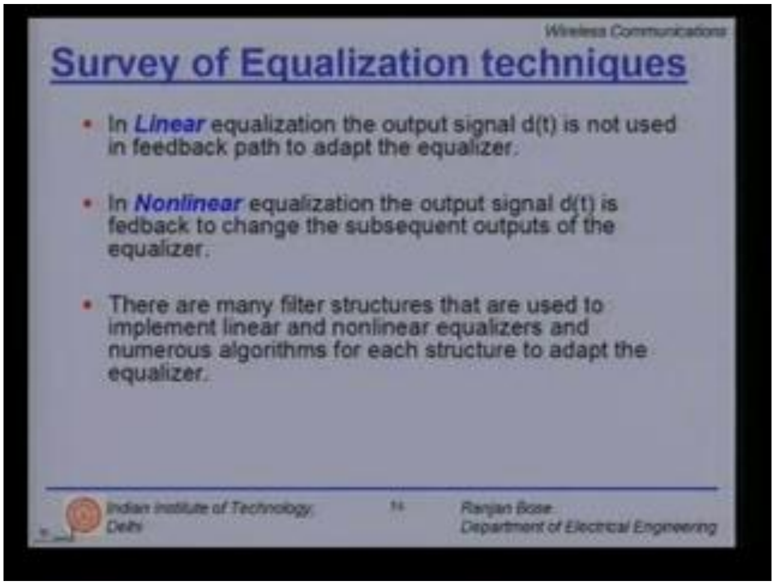
Survey of Equalization techniques

- Equalization techniques can be classified into
  - Linear
  - Non-Linear
- These are classified based on how the output of an adaptive equalizer is used for subsequent control of the equalizer.
- In general, the decision maker determines the value of the digital data bit being received and applies slicing or thresholding operation which is a nonlinear operation for obtaining the value of d(t).

Indian Institute of Technology, Delhi 13 Ranjan Bose Department of Electrical Engineering

Now let us have a quick survey of equalization techniques. Equalization techniques can be broadly classified into the following linear equalizers, non-linear equalizers. These are classified based on how the output of an adaptive equalizer is used for subsequent control of the equalizer. So if we have a feedback loop, if we use the estimate of d_k that is d_k hat and feed it back to update the weights then we get into the domain non-linear equalizer whereas if you only have the forward loop and there is no feedback then it is a linear equalizer. We will soon see that the linear equalizer can be easily implemented as a finite impulse response FIR filter. In general the decision maker determines the value of the digital data bit being received and applies slicing or thresholding operation which is a non-linear operation for obtaining the value of d (t). So there is a thresholding operation that will happen at the decision making level and only then you get an estimate of d (t).

(Refer Slide Time: 00:14:25 min)



Wireless Communications

Survey of Equalization techniques

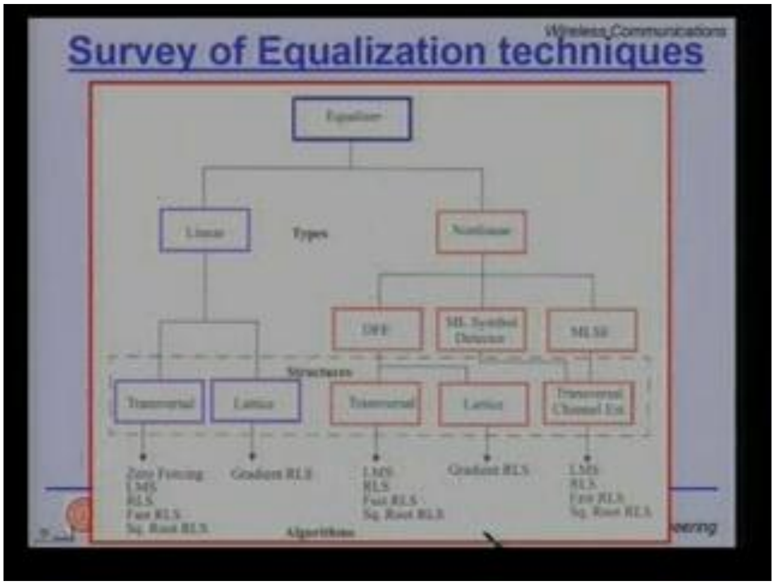
- In *Linear* equalization the output signal d(t) is not used in feedback path to adapt the equalizer.
- In *Nonlinear* equalization the output signal d(t) is fedback to change the subsequent outputs of the equalizer.
- There are many filter structures that are used to implement linear and nonlinear equalizers and numerous algorithms for each structure to adapt the equalizer.

Indian Institute of Technology, Delhi 14 Ranjan Bose Department of Electrical Engineering

In linear equalization the output signal d (t) is not used in the feedback path to adapt the equalizer. However in non-linear equalization the output signal d (t) is fedback to change the subsequent outputs of the equalizer is a basic difference. Now there are many filter structures that are used to implement linear as well as nonlinear equalizers and decoupled to that there are numerous algorithms for each structure to adapt the equalizer weights. So there are two components to an equalizer design, the structure and the algorithms to update the weights and there is no link between the two; you can choose one structure and use another algorithm to update the weights. Now let us classify our equalizers.

So the equalizer can be broadly divided into linear and non-linear types. Linear no feedback from the d_k, non-linear yes there is a feedback. Now let us talk about the structures, we have already seen the transversal structure. Again we will talk about it but linear equalizers have a transversals structure possible or also lattice structure possible.

(Refer Slide Time: 00:15:36 min)



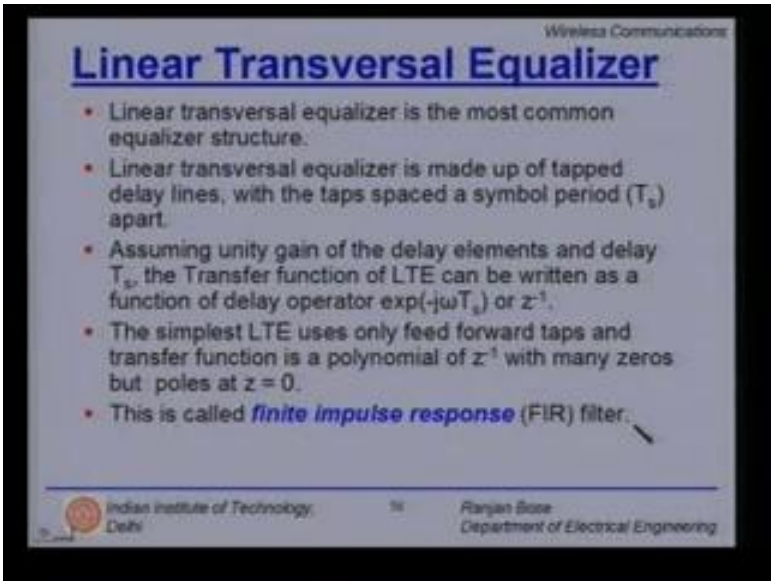
Wireless Communications

Survey of Equalization techniques

We will talk about these again in greater detail. At the other end the non-linear equalizers can either be a decision feedback equalizer or a maximum likelihood symbol detector or an MLSE sequences estimation, maximum likelihood sequence estimation. Again these three can be either implemented using a transversal filter or a lattice filter or a transversal channel estimator. Now for each of these structures you have the flexibility to use various kinds of algorithms to quickly update the weights. Most of them are recursive algorithms, you could either use the zero forcing or LMS least means square or RLS recursive least square or the variations there of. Again you have a possibility to use different kinds of algorithms depending upon how fast they are, what is the minimum error they give, how much power they can show.

So we will talk about how to pick and choose an algorithm for a particular type of structure later today but this gives you a brief over view of how equalizers can be classified in terms of linear, non-linear structure and the type of algorithm used to update the weights. Now first let us talk about the linear transversal equalizer, one of the simplest implementation and a popular one. The linear transversal equalizer is the most common equalizer structure simply because of its simple structure. The linear transversal equalizer is made up of tapped delay lines with the taps spaced at a symbol duration T_s apart. So the delay element gives a delay of T (S). Assuming unity gain of the delay elements and the delay equal to T (S), the transfer function of the LTE which is the linear transversal equalizer can be written as a function of delay operator e raise to power minus j omega T_s denoted by Z raise to power minus 1.

(Refer Slide Time: 00:17:41 min)



Wireless Communications

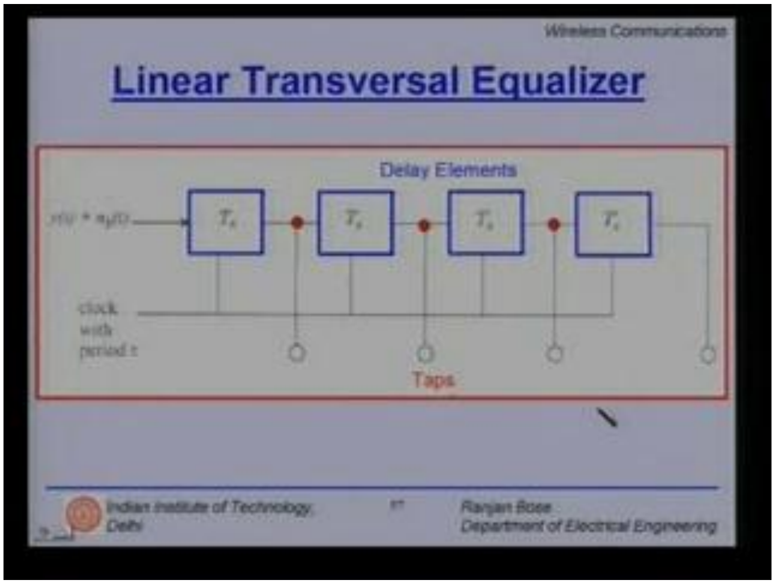
Linear Transversal Equalizer

- Linear transversal equalizer is the most common equalizer structure.
- Linear transversal equalizer is made up of tapped delay lines, with the taps spaced a symbol period (T_s) apart.
- Assuming unity gain of the delay elements and delay T_s, the Transfer function of LTE can be written as a function of delay operator $\exp(-j\omega T_s)$ or z^{-1}.
- The simplest LTE uses only feed forward taps and transfer function is a polynomial of z^{-1} with many zeros but poles at $z = 0$.
- This is called ***finite impulse response*** (FIR) filter.

Indian Institute of Technology, Delhi 16 Ranjan Bose Department of Electrical Engineering

The simplest LTE uses only feed forward taps and transfer function is a polynomial of z inverse with many zeros but poles only at Z is equal to zero. This is called a finite impulse response or FIR filter. So linear transversal equalizer can be simply represented as a FIR filter, there is no feedback. In this slide we have a simple structure for the LTE or the linear transversal equalizer as you can see we have the delay elements, T_s represents the symbol duration. Here you have y (t) plus n_b (t), n_b (t) represents the baseband noise. Please remember the equalizer is designed to work in the absence of noise, in the presence of noise there will be some predictive error. Clock with period tau, here we have the tap points and based on that you can calculate and obtain your d_k hat the estimate of d_k. So this is the basic structure of linear transversal equalizer, you have delay elements and you have the tap points.

(Refer Slide Time: 00:19:12 min)

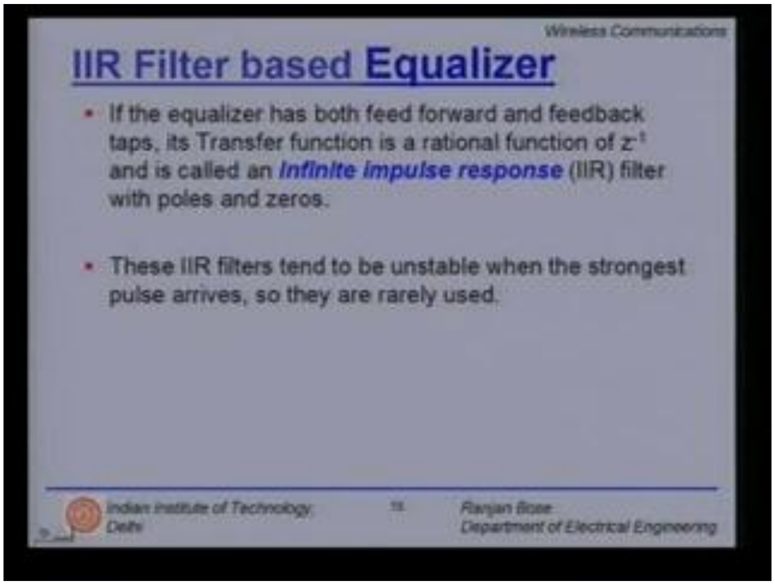


Wireless Communications

Linear Transversal Equalizer

Indian Institute of Technology, Delhi 17 Ranjan Bose Department of Electrical Engineering

(Refer Slide Time: 00:20:11 min)



Wireless Communications

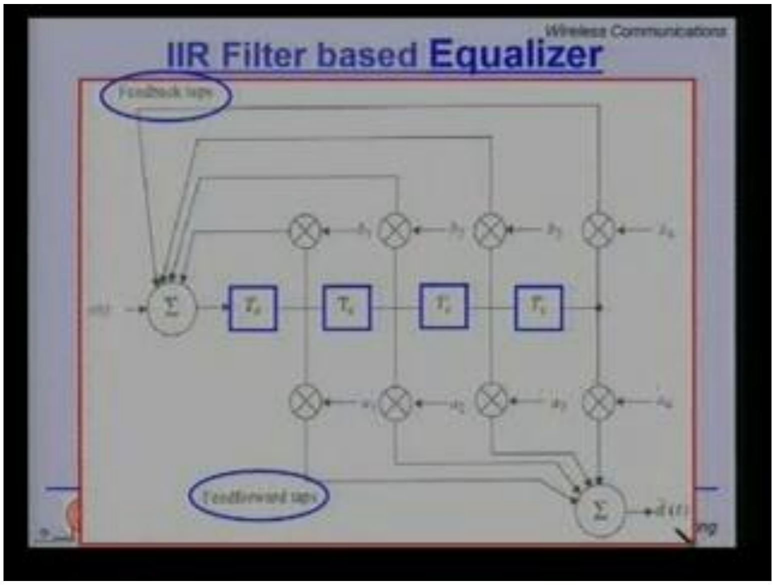
IIR Filter based Equalizer

- If the equalizer has both feed forward and feedback taps, its Transfer function is a rational function of z^{-1} and is called an ***Infinite impulse response*** (IIR) filter with poles and zeros.
- These IIR filters tend to be unstable when the strongest pulse arrives, so they are rarely used.

Indian Institute of Technology Delhi 15 Ranjan Bose Department of Electrical Engineering

Now let us talk about the IIR filter based equalizer; if the equalizer has both feed forward and feedback taps, if transfer function is a rational function of Z inverse and is called an infinite impulse response or IIR filter with both poles and zero’s. These IIR filters tend to be unstable when the strongest pulse arrives after the other pulses, so IIR filters tend to be unstable sometimes for mobile scenarios.

(Refer Slide Time: 00:20:56 min)



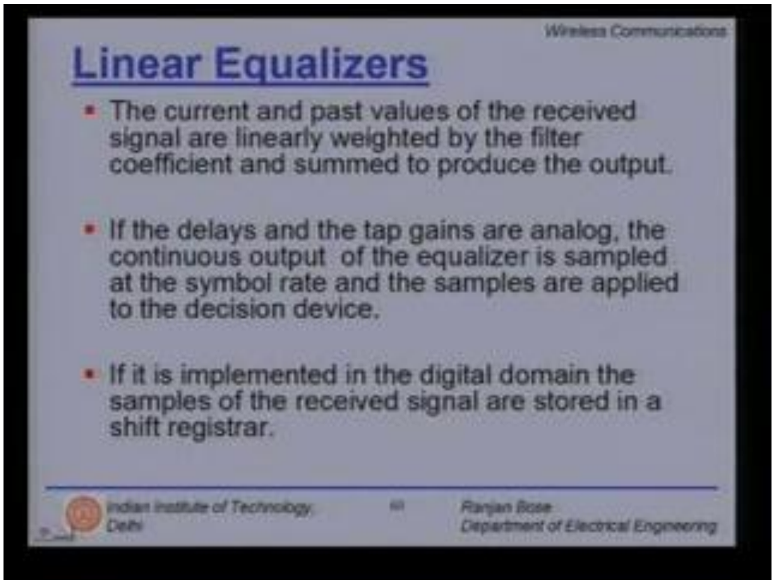
Wireless Communications

IIR Filter based Equalizer

Here is a block diagram of a simple IIR filter based equalizer. Please note y (t) is the input waveform; we have removed the noise part right here. Now here you have again the delays T_s then you have tap points.

So the a_1, a_2's and so and so forth represents the forward tap points and this is the feed forward taps. Here b_1, b_2, b_3 and so and so forth represents the feedback taps going here. Even though there is a feedback please note there is absolutely no feedback from the d hat t, the d hat t is not being used to update the taps at all, therefore it is still a linear equalizer. So there is a feedback path but feedback for the taps alone. So we have two loops the feedback taps adding up here and the feedback taps adding up here and the feed forward taps adding up here to give you the estimate d hat t.

(Refer Slide Time: 00:22:22 min)



Wireless Communications

Linear Equalizers

- The current and past values of the received signal are linearly weighted by the filter coefficient and summed to produce the output.
- If the delays and the tap gains are analog, the continuous output of the equalizer is sampled at the symbol rate and the samples are applied to the decision device.
- If it is implemented in the digital domain the samples of the received signal are stored in a shift registrar.

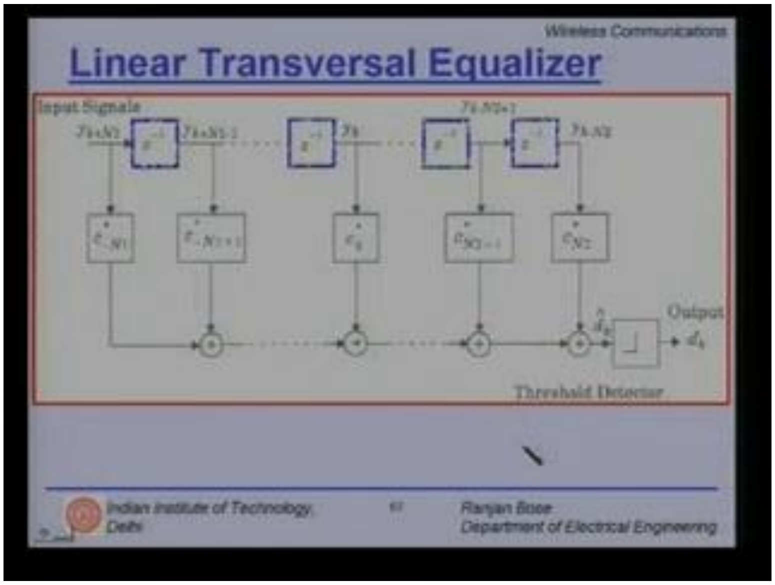
Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

Continuing with our linear equalizer; the current and the past values of the received signal are linearly weighted by the filter coefficients and summed to produce the outputs. So it is a very simple implementation and hardware, multiply with the weights and finally add them up.The shifting is usually done using shift registrar so it's a very hardware friendly operation. If the delays and the tap gains are analog, the continuous output of the equalizer is sampled at the symbol rate and the samples are applied to the decision device where thresholding or slicing is done. If it is implemented in the digital domain, the samples of the received signal are stored in a shift register.

Let us take another look at the linear transversal equalizer or LTE. Again we have a an input signal y coming in here and then there are certain number of tap points multiplied with the complex weights, add them up and this is the decision maker which is a thresholding device and output is a d_k. This is simple structure of the linear transversal equalizer. [Conversation between Student and Professor – Not audible ((00:23:59 min))] Question being asked is can d_k hat be a complex number? The answer is no, at this time you have taken a decision about the bit.

(Refer Slide Time: 00:23:19 min)



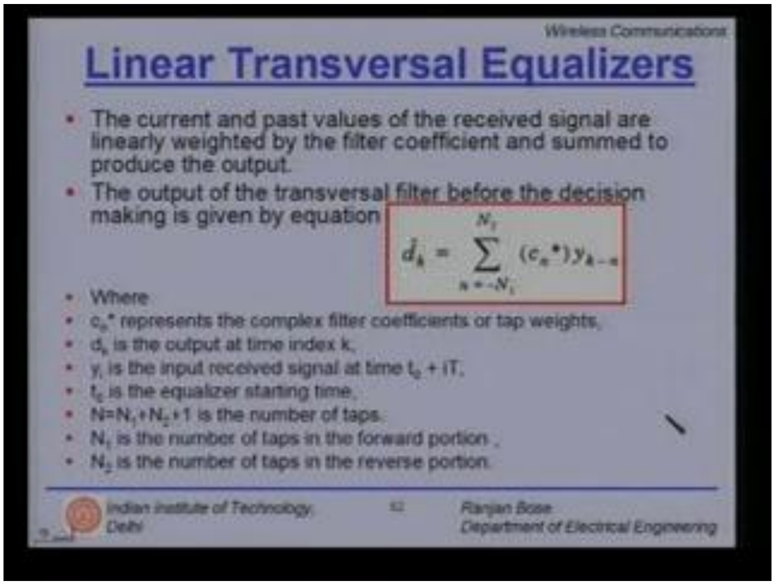
Wireless Communications

Linear Transversal Equalizer

Indian Institute of Technology, Delhi
Ranjan Bose
Department of Electrical Engineering

So it’s already decided so this will not be complex, prior to that it can be complex. So all these can be complex clearly they are complex you are multiplying with a complex number but whatever you do at the end you have to take a decision here and for a very simplistic binary case you are coming up with the 1 and a 0. Here please note these are the tap points delayed by T of S the symbol duration and the number of delay elements would be a function of what is the delay spread of the channel. The weights are complex as shown here, these weights must be updated by some method.

(Refer Slide Time: 00:25:06 min)



Wireless Communications

Linear Transversal Equalizers

- The current and past values of the received signal are linearly weighted by the filter coefficient and summed to produce the output.
- The output of the transversal filter before the decision making is given by equation

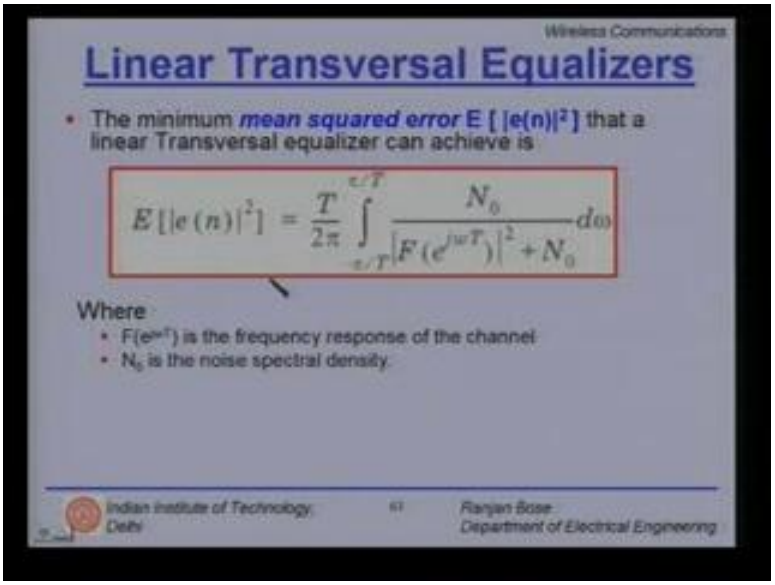
$$\hat{d}_k = \sum_{n=-N_1}^{N_2} (c_n^*) y_{k-n}$$

- Where
- c_n^* represents the complex filter coefficients or tap weights,
- d_k is the output at time index k,
- y_i is the input received signal at time $t_0 + iT$,
- t_0 is the equalizer starting time,
- $N=N_1+N_2+1$ is the number of taps.
- N_1 is the number of taps in the forward portion ,
- N_2 is the number of taps in the reverse portion.

Indian Institute of Technology, Delhi
Ranjan Bose
Department of Electrical Engineering

The current and the past values of the received signals are linearly weighted by the filter coefficients and summed to produce the output. The output of the transversal filter before the decision making is given by this equation d hat k is summation of the complex weights multiplied by the taped delay values of y_k. In this equation C_n^* represents the complex filter coefficients also called as the tap weights, d_k is the output at the time index k, y_i is the input received signal at time t_0 + iT, t_0 is the equalizer starting time, $N = N_1 + N_2 +1$ is a number of taps. What are N_1's and N_2's? N_1 is the number of tap in the forward portion, N_2 is a number of taps in the reverse portion. So this sums up the general linear transversal equalizer. How you adjust the weights is a different story but this should give you, if the weights are correct and optimal a good estimate or what was transmitted.

(Refer Slide Time: 00:26:31 min)



Wireless Communications

Linear Transversal Equalizers

- The minimum *mean squared error* $E[|e(n)|^2]$ that a linear Transversal equalizer can achieve is

$$\boxed{E[|e(n)|^2] = \frac{T}{2\pi}\int_{-\pi/T}^{\pi/T}\frac{N_0}{|F(e^{j\omega T})|^2+N_0}d\omega}$$

Where

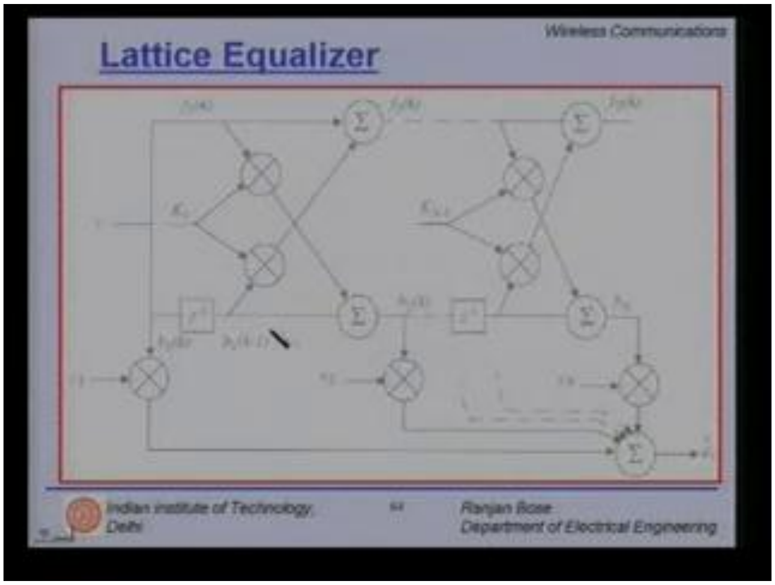
- $F(e^{j\omega T})$ is the frequency response of the channel
- N_0 is the noise spectral density

Indian Institute of Technology Delhi 61 Ranjan Bose Department of Electrical Engineering

Now let us talk about how good is it? The minimum mean squared error which is the expected value of error squared that a linear transversal equalizer can achieve can be worked out as follows. Expected value error squared is T over 2 pi integration minus pi by T to pi by T N_0 over this F e raise to power j omega t absolute value squared plus N_0 d omega. The derivation is not being given here, it is available in most of the text book for example (Not understandable) (Refer Slide Time: 00:27:14 min)) gives a very good derivation for this one. Here the F e raise to power j omega T is a frequency response of the channel. N_0 is the noise spectral density, so please note that it depends on the noise, I can do nothing about it. If I increase the noise power I helplessly have to see that the minimum mean squared error goes up I can do nothing about it and if my channel is not so hostile if I have a flat channel then this becomes pretty handelable, you can work around it and you can lower your error.

Now let us look at another implementation in terms of structure for a linear equalizer, we will call it the lattice equalizer. There is a lot of information on this slide let us try to understand what is going on. Here you can see y the input, now this input has been judicially divided into two branches. Here is the forward loop, here is the backward loop. The forward loop is been characterized with f_1 (k), f_2 (k) so on and so forth till f_n (k).

(Refer Slide Time: 00:28:11 min)



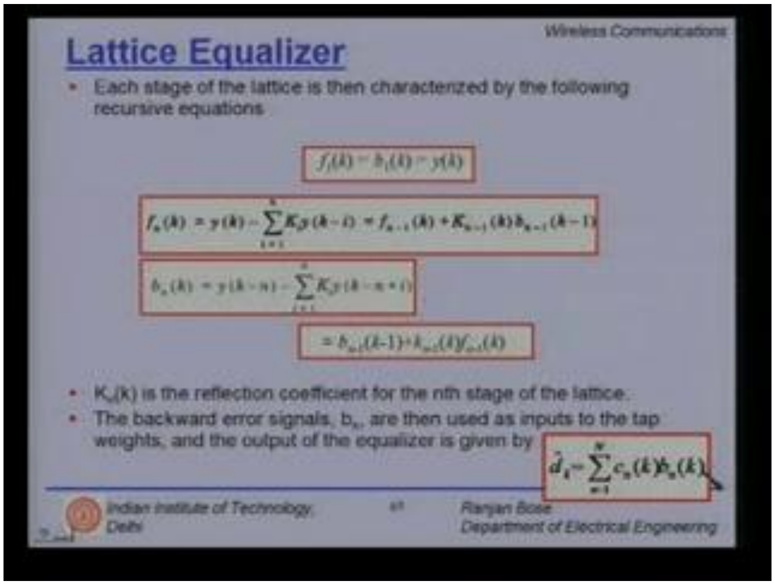
Wireless Communications

Lattice Equalizer

Indian Institute of Technology, Delhi 64 Ranjan Bose Department of Electrical Engineering

The backward loop as its coefficients b$_1$ (k-1), b$_2$ (k) and so and so forth till b$_n$k. The delays are here C$_1$, C$_2$ up to C$_n$ and this particular structure is called the lattice structure where this reflection coefficients K$_1$, K$_2$ up to K$_{n-1}$, you add them up and you get d$_1$ (t) you'll just now show the despite the structure the d$_1$ hat will come out to be nothing but the delayed version of y$_k$'s multiplied by the appropriate complex weights but there are certain very good advantages of the lattice equalizer which will talk about, so each stage of the lattice is characterized by the following recursive equations.

(Refer Slide Time: 00:29:38 min)



Wireless Communications

Lattice Equalizer

- Each stage of the lattice is then characterized by the following recursive equations

$$f_1(k) = b_1(k) = y(k)$$

$$f_n(k) = y(k) - \sum_{i=1}^{n} K_i y(k-i) = f_{n-1}(k) + K_{n-1}(k) b_{n-1}(k-1)$$

$$b_n(k) = y(k-n) - \sum_{i=1}^{n} K_i y(k-n+i)$$

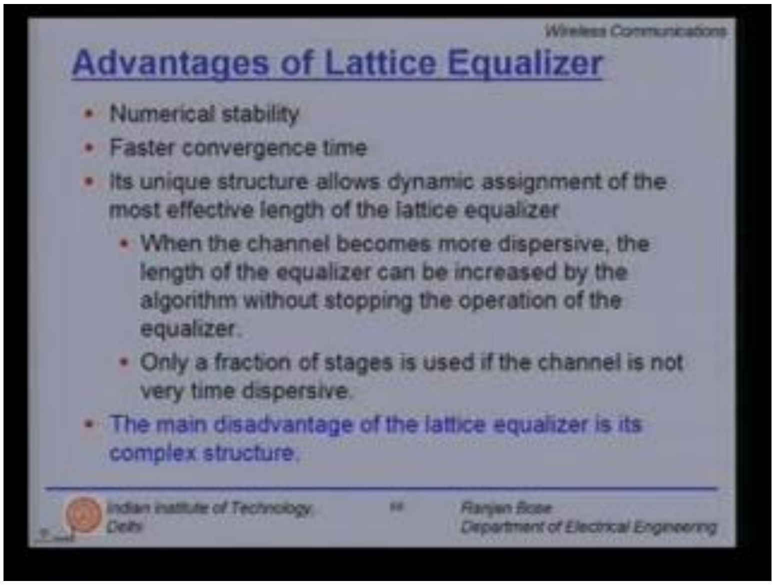
$$= b_{n-1}(k-1) + K_{n-1}(k) f_{n-1}(k)$$

- $K_n(k)$ is the reflection coefficient for the nth stage of the lattice.
- The backward error signals, b_n, are then used as inputs to the tap weights, and the output of the equalizer is given by $\hat{d}_k = \sum_{n=1}^{N} c_n(k) b_n(k)$

Indian Institute of Technology, Delhi 65 Ranjan Bose Department of Electrical Engineering

So if you go back and forth between the last slide and this slide you can easily correlate that f_1 (k) is b_1 (k) is y (k) that is the initialization factor and then you have a recursive equation f_n (k) is y_n k minus summation i is equal to one through n is reflection coefficient k_i y (k- i) is equal to this expression. So if you go back here and try to correlate, you will be able to figure out how you can write y (k) in terms of the previous values. Similarly b_n (k) is written by y (k- n) minus summation i is equal to one through n k_i y (k-n+1). Again comes from the lattice structure this can be written as b_n (k) is equal to b_{n-1} (k-1) + k_{n-1} (k) f_{n-1} (k), very simple relationship. Here k subscript nk is the reflection coefficient for the nth stage of the lattice. The backward error signals b_n are then used as inputs to the tap weights and the output of the equalizer is finally given by d_k hat is summation n =1 through n c_n (k) b_n (k) exactly what we are looking.

(Refer Slide Time: 00:31:31 min)



Wireless Communications

Advantages of Lattice Equalizer

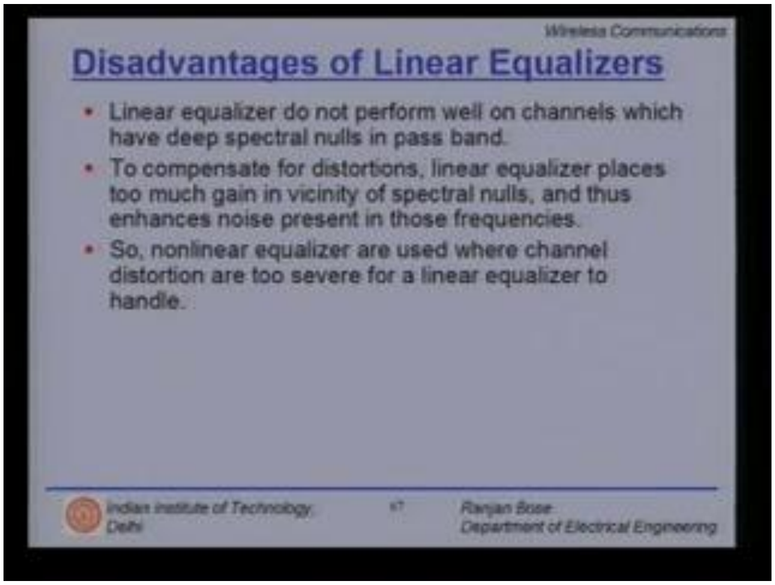
- Numerical stability
- Faster convergence time
- Its unique structure allows dynamic assignment of the most effective length of the lattice equalizer
  - When the channel becomes more dispersive, the length of the equalizer can be increased by the algorithm without stopping the operation of the equalizer.
  - Only a fraction of stages is used if the channel is not very time dispersive.
- The main disadvantage of the lattice equalizer is its complex structure.

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

So why do we have to go through that complex structure? We finally getting what we expect but why do we have to go for that structure? The first thing is there is a lot of numerical stability, we saw the IIR filters tend to get unstable mathematically, here the lattice equalizer has numerical stability then it has a faster convergence time very important. It’s unique structure allows dynamic assignment of the most effective length of the lattice equalizer. This is for the first time we are talking about the effective length required for the equalizer structure which should be matched to the delay spread of the channel. When the channel becomes more dispersive the mobile enters in a densely packed urban environment, the length of the equalizer can be increased by the algorithm without stopping the operation of the equalizer. On the other hand only a fraction of stages is used if the channel is not very time dispersive. What is the main disadvantage? It is obvious the main disadvantage of the lattice equalizer is its complex structure but this is again a good linear equalizer that is used.

(Refer Slide Time: 00:33:06 min)



Wireless Communications

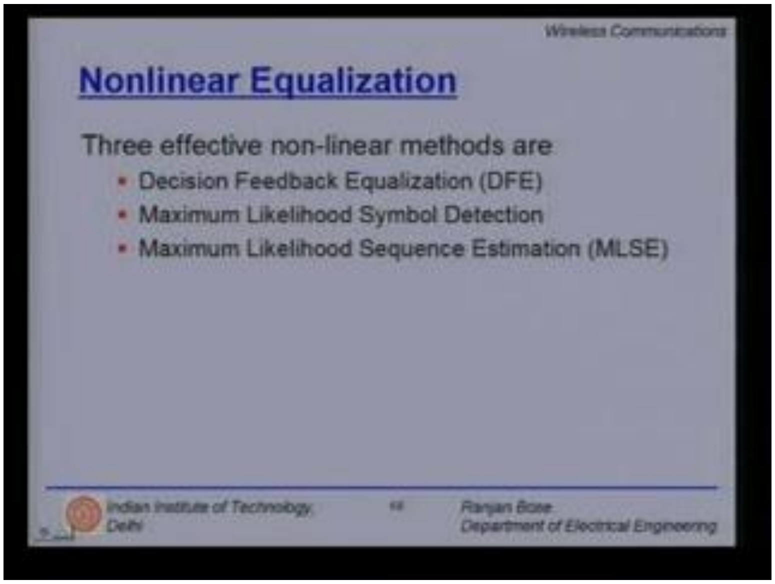
Disadvantages of Linear Equalizers

- Linear equalizer do not perform well on channels which have deep spectral nulls in pass band.
- To compensate for distortions, linear equalizer places too much gain in vicinity of spectral nulls, and thus enhances noise present in those frequencies.
- So, nonlinear equalizer are used where channel distortion are too severe for a linear equalizer to handle.

Indian Institute of Technology, Delhi 67 Ranjan Bose Department of Electrical Engineering

But in general linear equalizers have certain disadvantages. What are these? The linear equalizers do not perform well on channels which have deep spectral nulls in the pass band. After all what is an equalizer doing? It is trying to make an inverse filter undo the effects so in the frequency domain wherever there are deep nulls which there are in most hostile wireless channels, it will try to have a very high value in the frequency domain there by giving a lot of importance to the noise present at those frequencies, it'll amplify the noise immensely. To compensate for distortions linear equalizer places too much gain in the vicinity of these spectral nulls and thus enhances the noise present in these frequencies, not good. So non-linear equalizers are used where channel distortions are too severe for linear equalizers to handle that's where this step in.

(Refer Slide Time: 00:34:24 min)



Wireless Communications

Nonlinear Equalization

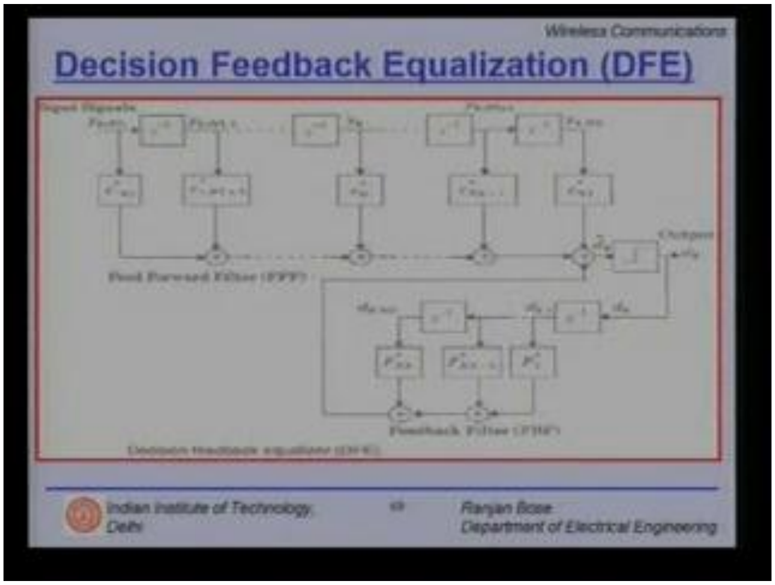
Three effective non-linear methods are

- Decision Feedback Equalization (DFE)
- Maximum Likelihood Symbol Detection
- Maximum Likelihood Sequence Estimation (MLSE)

Indian Institute of Technology, Delhi 68 Ranjan Bose Department of Electrical Engineering

So let us now move into the domain of non-linear equalizers, the three effective non-linear equalizer methods are as follows. The first which is also a very common one is the decision feedback equalizer DFE. We will talk about DFE in detail followed by the maximum likelihood symbol detection technique and then the maximum likelihood sequence estimation MLSE will talk about MLSE also.

(Refer Slide Time: 00:35:00 min)



Wireless Communications

Decision Feedback Equalization (DFE)

Indian Institute of Technology, Delhi

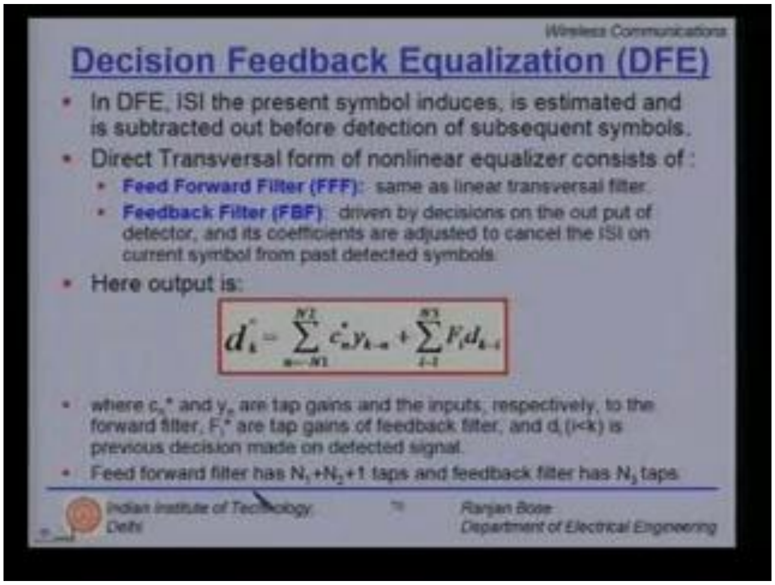
Ranjan Bose
Department of Electrical Engineering

First lest us look at the decision feedback equalizer the DFE. This is the general structure of DFE, the input signal comes in here and then it is delayed as always then you have the complex weights C_n (1), C_{n+1} and so and so forth. This is your feed forward filter also called as FFF or triple F. At this juncture you have d_k and then you have a thresholding device and then you get an estimate. Now what is interesting is for the first time we take this d_k after the decision device and feed it back to the feedback filter FBF shown here.

Again this has several delay elements and then the feedback filter coefficients F_1^* and so and so forth adding up and going in. Somewhere the philosophy is to estimate the intersymbol interference because you have already detected the past symbols so you can predict what are the intersymbol interference it will cause to the subsequent symbols and subtract it out. So if you can estimate the intersymbol interference that will be induced by the current decoded symbol and if your decision is correct then it is possible for you to subtract out the ISI. So that is the smart move that DFE does.

So in DFE the intersymbol interference the present symbol induces is estimated and subtracted out before the detection of subsequent symbols, fairly simple philosophy. If you already have the information why not use it to cancel out the intersymbol interference it will induce. Now the direct transversal form of the non-linear equalizer consists of as we have seen the feed forward filter FFF, it is same as a linear transversal filter we have learnt so far and then there is this new element called the feedback filter FBF driven by the decisions on the output of the detector and its coefficients are adjusted to cancel the ISI on the current symbol from the past detected symbols.

(Refer Slide Time: 00:36:59 min)



Wireless Communications

Decision Feedback Equalization (DFE)

- In DFE, ISI the present symbol induces, is estimated and is subtracted out before detection of subsequent symbols.
- Direct Transversal form of nonlinear equalizer consists of :
  - Feed Forward Filter (FFF): same as linear transversal filter.
  - Feedback Filter (FBF): driven by decisions on the out put of detector, and its coefficients are adjusted to cancel the ISI on current symbol from past detected symbols.
- Here output is:

$$\hat{d}_k = \sum_{n=-N1}^{N2} c_n^* y_{k-n} + \sum_{i=1}^{N3} F_i d_{k-i}$$

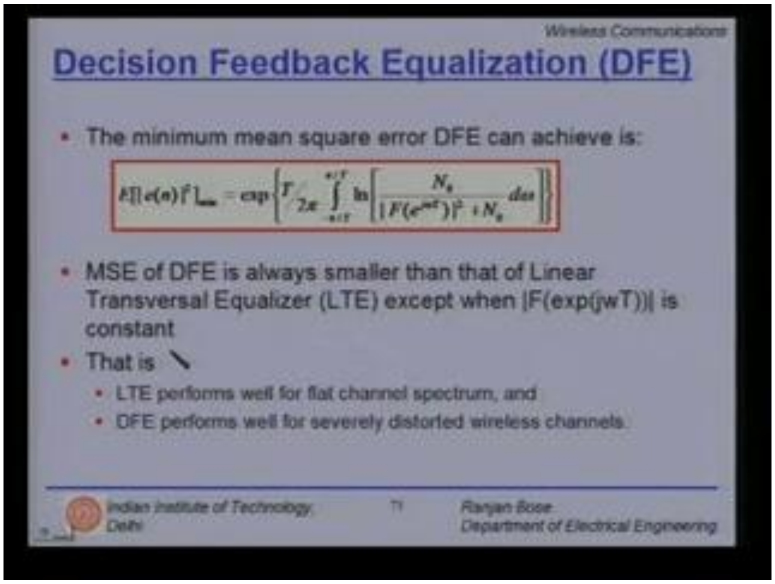
- where c_n^* and y_n are tap gains and the inputs, respectively, to the forward filter, F_i^* are tap gains of feedback filter, and d_i $(i<k)$ is previous decision made on detected signal.
- Feed forward filter has N_1+N_2+1 taps and feedback filter has N_3 taps.

Indian Institute of Technology Delhi

Ranjan Bose
Department of Electrical Engineering

For DFE the output is given by d hat k equal to summation n is equal to minus N_1 through N_2 complex weights multiplied by the appropriately delayed input plus this feedback filter coefficients F_i multiplied by d_{ki}. Please note here you are working on the delayed version of the decisions. What is your dk? The d_k is a decision on a symbol being transmitted and as you can take d_{k-i} you delayed the previous. So based on the previous symbols you adjust the weights and subtract the ISI. Here C_n^* and Y_n are tap gains and the inputs respectively to the forward filter and F_i^* are the tap gains of the feedback filter and d_i where i is less than k is the previous decision made on a detected signal. Note, feed forward filter has N_1+ N_2+1 taps whereas the feedback filter has N_3 taps.

(Refer Slide Time: 00:39:26 min)



Wireless Communications

Decision Feedback Equalization (DFE)

- The minimum mean square error DFE can achieve is:

$$E[|e(n)|^2]_{\min} = \exp\left\{\frac{T}{2\pi}\int_{-\pi/T}^{\pi/T} \ln\left[\frac{N_o}{|F(e^{j\omega T})|^2 + N_o}\, d\omega\right]\right\}$$

- MSE of DFE is always smaller than that of Linear Transversal Equalizer (LTE) except when |F(exp(jwT))| is constant
- That is
  - LTE performs well for flat channel spectrum, and
  - DFE performs well for severely distorted wireless channels.

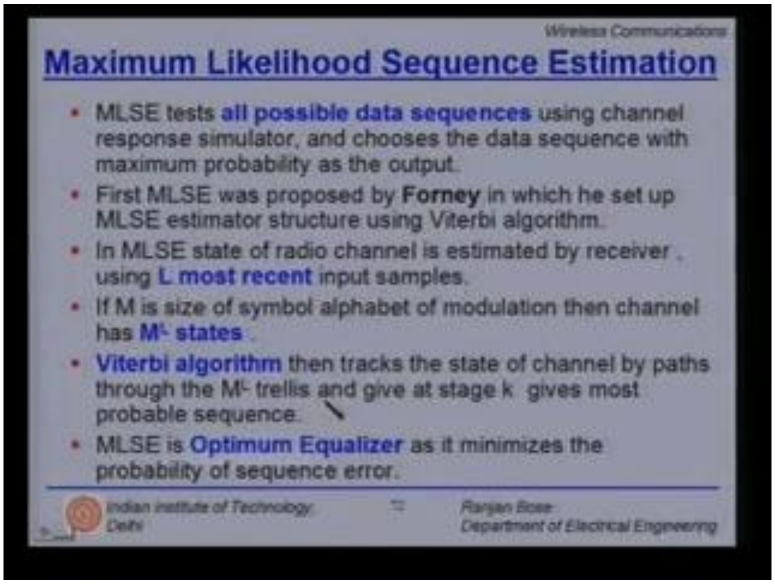
Indian Institute of Technology Delhi

Ranjan Bose
Department of Electrical Engineering

Now again we have to talk about how good or bad is your DFE. So we have to resort to the minimum mean squared error. So the minimum mean squared error for DFE can be given by the following equation, expected value of error squared minimum is e raise to power T over 2 pi integration minus pi by T to pi by T log of this expression. Please note we had a previous expression where both the exponential and logarithmic were missing. It can be shown that the minimum square error of DFE is always smaller than that of the linear transversal equalizer LTE except when the absolute value F e raise to power j omega T is constant, this value; there is a transfer function of the channel, N_0 is the noise.

Again we'll helpless as noise goes up, noise power goes up then your minimum mean square error will go up and we can do nothing about it. Now LTE performs well for flat channel spectrum, here F e raise to power j omega T is the channel spectrum that we are talking about. For a flat channels spectrum are general linear transversal equalizer works well but the movement you severely distorted wireless channels with deep nulls in the spectrum, you have to have the decision feedback equalizer or an non-linear equalizer to come to the rescue. What is important to note is the minimum mean squared error of DFE is always smaller than that of the LTE.

(Refer Slide Time: 00:41:34 min)



Wireless Communications

Maximum Likelihood Sequence Estimation

- MLSE tests **all possible data sequences** using channel response simulator, and chooses the data sequence with maximum probability as the output.
- First MLSE was proposed by **Forney** in which he set up MLSE estimator structure using Viterbi algorithm.
- In MLSE state of radio channel is estimated by receiver using **L most recent** input samples.
- If M is size of symbol alphabet of modulation then channel has **M^L states**.
- **Viterbi algorithm** then tracks the state of channel by paths through the M^L trellis and give at stage k gives most probable sequence.
- MLSE is **Optimum Equalizer** as it minimizes the probability of sequence error.

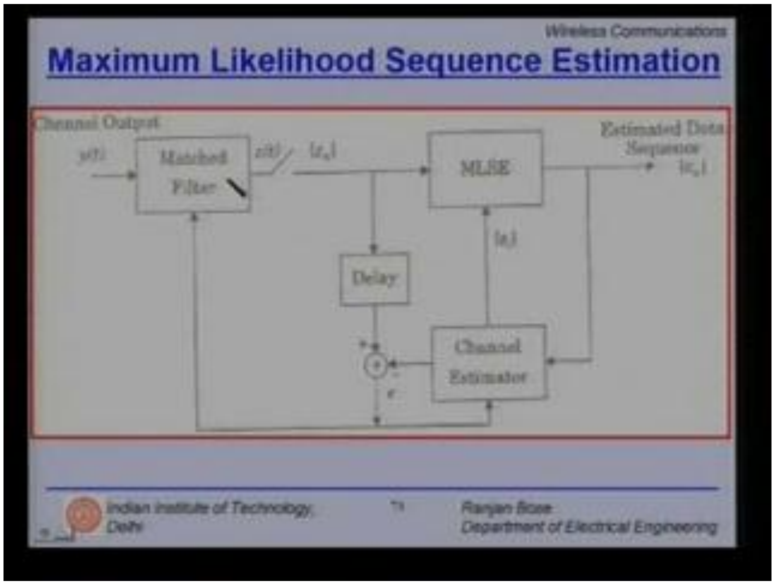
Indian Institute of Technology Delhi

Ranjan Bose
Department of Electrical Engineering

Now let us talk about the next one maximum likelihood sequence estimation. The MLSE test all possible data sequence using channel response simulator and chooses the data sequence with the maximum probability as the output. So it was first proposed by Forney in way back in 78 in which he set up the MLSE estimator structure using the Viterbi algorithm. The MLSE state of the radio channel is estimated by the receiver using L most recent input samples. If M is a size of the input alphabet of modulation then the channel has M raise to power L states. The Viterbi algorithm then tracks the state of the channel by paths through the M raise to power L trellis and gives at stage K the most probable sequence. So it is some kind of a sequence estimation, needless to say it will be computationally very intensive but you will get a much better minimum mean squared error.

MLSE is an optimum equalizer as it minimizes the probability of sequence error. Please note that two things, one is the probability of sequence error and one is the minimum mean squared error. The bottom line is the probability of sequence error should be minimized, it so happened that in earlier cases we have said okay we will minimize the minimum mean squared error and then it'll automatically lead me to have a low probability of error. Here for the first time MLSE works on minimizing the probability of sequence error, so in that sense it is an optimum equalizer.

(Refer Slide Time: 00:43:48 min)



Wireless Communications

Maximum Likelihood Sequence Estimation

Indian Institute of Technology, Delhi

Ranjan Bose
Department of Electrical Engineering

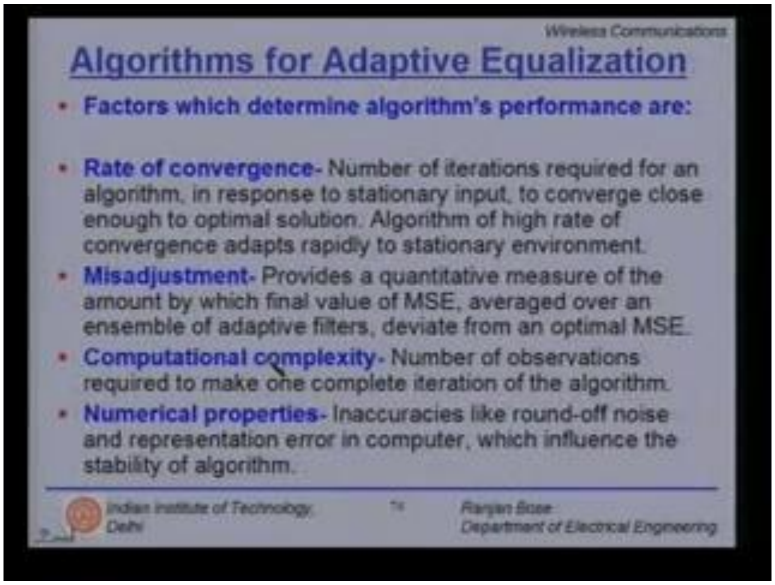
So let us look at the block diagram of the maximum likelihood sequence estimation. Here for a change this is a different structure, there is the input y (t) goes through a matched filter and immediately you have a Z (t), you sample it to get Zl ((00:44:13 min)) and you work on the samples. So you have this maximum likelihood sequence estimation which is being implemented using a Viterbi algorithm and then here you have a delay channel estimator and goes in a loop and adjust the matched filter. Finally after K stages you get the estimated data sequence ((00:44:43 min)).

Now let us focus our attention on the various kinds of algorithms for adaptive equalizer. These algorithms can be used both for linear as well as non-linear equalization and we will discuss them here in general for linear equalizers though the analysis that we discus today can be easily extended for the case of non-linear equalization. First what are the factors which determine an algorithms performance? First is the rate of convergence, we have discussed this before all of the algorithms are recursive they take their own sweet time to converge.

So the rate of convergence is important. What is it? The number of iterations required for an algorithm in response to a stationary input to converge close enough to an optimal solution. Algorithm of high rate of convergence adapts rapidly to a stationary environment and it's important for us to have a fast rate of convergence.

The second point is mis-adjustment. What is that? It provides a quantitative measure of the amount by which the final value of the mean squared error averaged over an ensemble of adaptive filters deviates from the optimal mean square error.

(Refer Slide Time: 00:44:46 min)



Wireless Communications

Algorithms for Adaptive Equalization

- **Factors which determine algorithm's performance are:**
- **Rate of convergence-** Number of iterations required for an algorithm, in response to stationary input, to converge close enough to optimal solution. Algorithm of high rate of convergence adapts rapidly to stationary environment.
- **Misadjustment-** Provides a quantitative measure of the amount by which final value of MSE, averaged over an ensemble of adaptive filters, deviate from an optimal MSE.
- **Computational complexity-** Number of observations required to make one complete iteration of the algorithm.
- **Numerical properties-** Inaccuracies like round-off noise and representation error in computer, which influence the stability of algorithm.

Indian Institute of Technology Delhi

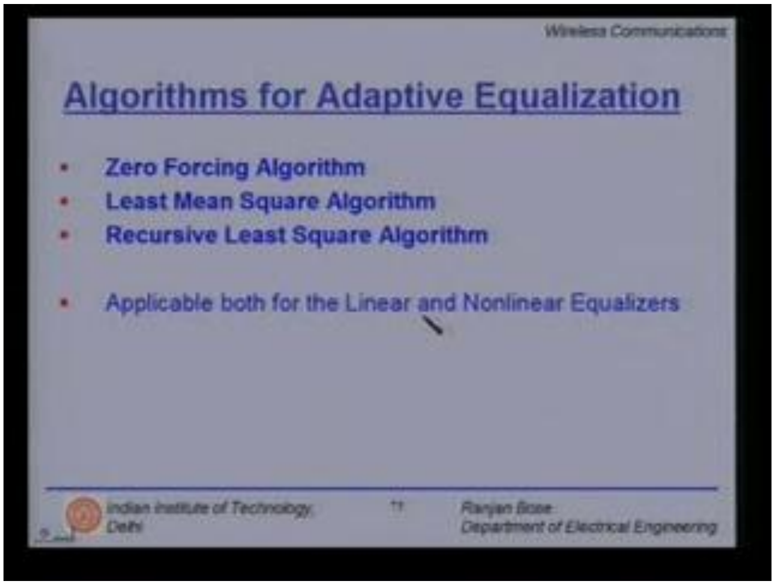
74

Ranjan Bose
Department of Electrical Engineering

Computational complexity important, number of observations required to make one complete iteration of the algorithm. It will clearly be correlated to the energy your equalizer consumes. Numerical properties; certain algorithm are prone to inaccuracies like round off noise and representation error in the computer and your fixed point operators. So many times you don't have floating point operations possible in your digital signal processers or things like that or round off errors or the finite length quantization etc. So those may lead to unstable algorithms at a slower rate of convergence. These are the four important aspects we must keep in mind while taking up a certain algorithm.

Let us briefly talk about the important algorithms. The three most popular one are zero forcing algorithm, the least mean square algorithm or the LMS and the recursive least square or RLS algorithm. Please note these are applicable both for the linear and the non-linear equalizers.

(Refer Slide Time: 00:47:20 min)



Wireless Communications

Algorithms for Adaptive Equalization

- Zero Forcing Algorithm
- Least Mean Square Algorithm
- Recursive Least Square Algorithm

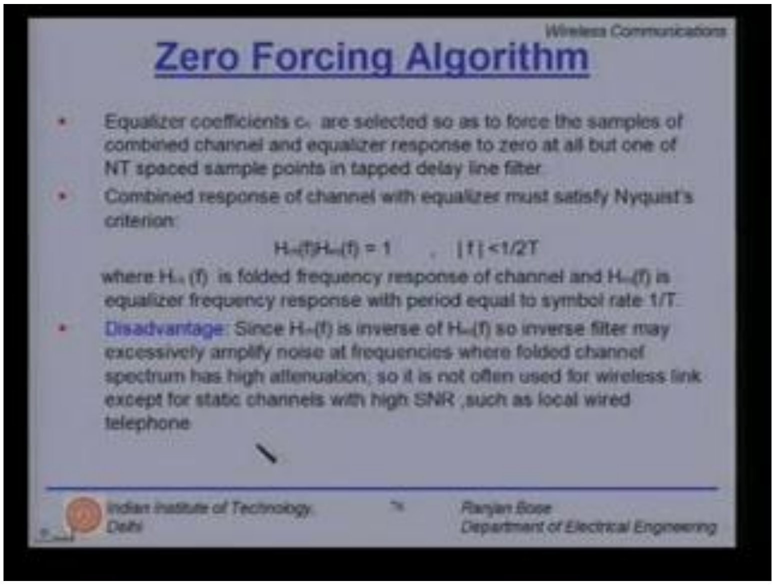
- Applicable both for the Linear and Nonlinear Equalizers

Indian Institute of Technology Delhi

Ranjan Bose
Department of Electrical Engineering

First let us talk about the zero forcing algorithm. Here the equalizer coefficients C_n are selected so as to force the samples of the combined channel and the equalizer response to zero at all but one of NT spaced sample points in a tapped delay line. What does it mean? It means it is trying to stimulate the Nyquist's condition where in other than its own sampling instance for all other instances it is forcing the value of the symbols to be zero.

(Refer Slide Time: 00:47:45 min)



Wireless Communications

Zero Forcing Algorithm

- Equalizer coefficients c_n are selected so as to force the samples of combined channel and equalizer response to zero at all but one of NT spaced sample points in tapped delay line filter
- Combined response of channel with equalizer must satisfy Nyquist's criterion:

$$H_{ch}(f)H_{eq}(f) = 1 \quad , \quad |f| < 1/2T$$

where $H_{ch}(f)$ is folded frequency response of channel and $H_{eq}(f)$ is equalizer frequency response with period equal to symbol rate 1/T

- Disadvantage: Since $H_{eq}(f)$ is inverse of $H_{ch}(f)$ so inverse filter may excessively amplify noise at frequencies where folded channel spectrum has high attenuation; so it is not often used for wireless link except for static channels with high SNR ,such as local wired telephone

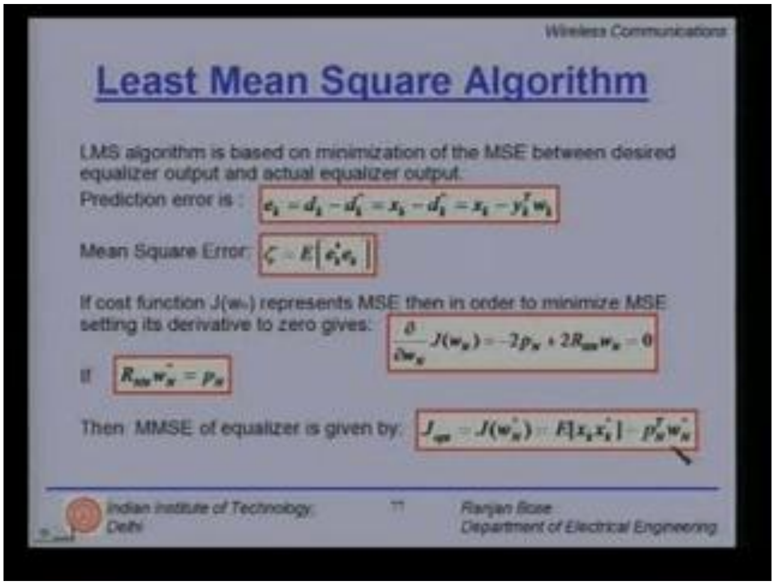
Indian Institute of Technology Delhi

Ranjan Bose
Department of Electrical Engineering

So here is the combined impulse response as expected for any adaptive equalizer H channel (f) H equalizer f is equal to one.

The disadvantage is as follows. Since the H channel is inverse of H equalizer, the inverse filter may excessively amplify noise at frequencies where folded channel spectrum has high attenuation. This zero forcing algorithm is not used for wireless links except for static cases. It was originally invented for wire line applications so it's normally not used.

(Refer Slide Time: 00:49:01 min)



Wireless Communications

Least Mean Square Algorithm

LMS algorithm is based on minimization of the MSE between desired equalizer output and actual equalizer output.

Prediction error is : $e_k = d_k - \hat{d}_k = x_k - \hat{d}_k = x_k - y_k^T w_k$

Mean Square Error: $\zeta = E\left[e_k^* e_k\right]$

If cost function J(wₙ) represents MSE then in order to minimize MSE setting its derivative to zero gives: $\frac{\partial}{\partial w_N} J(w_N) = -2p_N + 2R_{NN} w_N = 0$

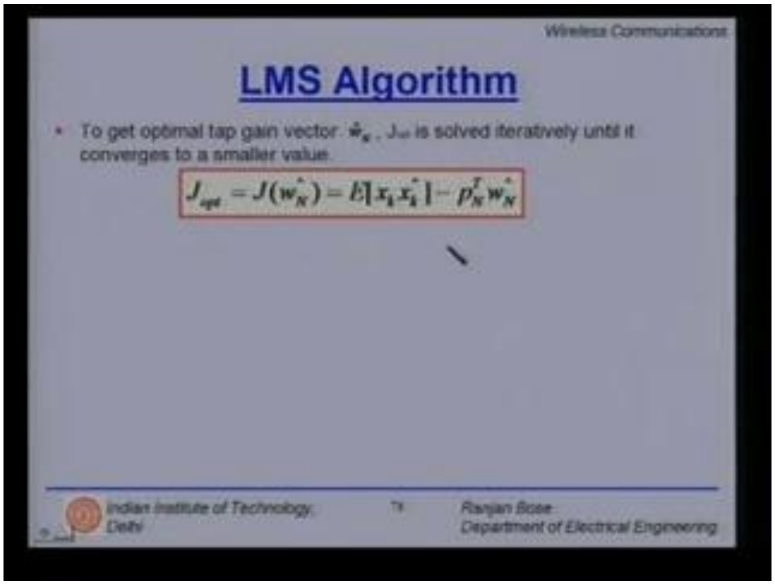
If $R_{NN} \hat{w}_N = p_N$

Then MMSE of equalizer is given by: $J_{opt} = J(\hat{w}_N) = E[x_k x_k^*] - p_N^T \hat{w}_N$

Indian Institute of Technology Delhi 77 Ranjan Bose Department of Electrical Engineering

A more popular one is the least mean square algorithm also called the LMS algorithm. The LMS algorithm is based on minimizing the mean square error between the desired equalizer output and the actual equalizer output. So it works purely on minimizing the MSE. The prediction error is given by e_k given by d_k minus d_k hat which can be expressed as x_k minus y_k transpose, these are the vector transpose of the delayed inputs and this is the weight vector W_k. The mean squared error is nothing but expected value of e_k complex conjugate e_k. Now we would define for the first time of cost function J is the function of the weight N represents the mean square error then in order to minimize the mean squared error its derivative of J should be zero. So there J omega N this is W_n by delta W_n should be equal to zero and you get this famous normal equation where you have to put R_{NN} the input covariance matrix times the weight vector equal to the $P_{N.}$ This is the correlation between the expected x_k and the y_k received vector. If this is set the minimum mean squared error of the equalizer is given by the following equation. So if we use LMS algorithm this is what you expect to get in terms of the minimum mean squared error.

(Refer Slide Time: 00:51:18 min)



Wireless Communications

LMS Algorithm

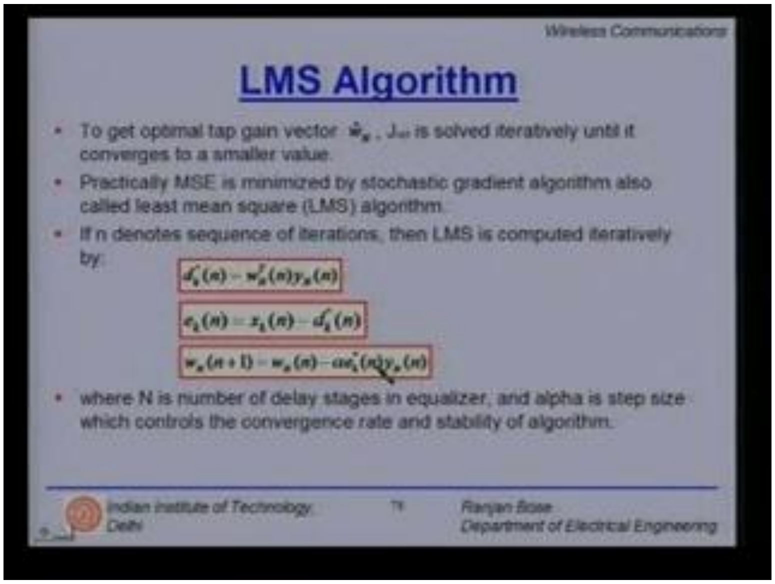
- To get optimal tap gain vector $\hat{w}_N$, J_{opt} is solved iteratively until it converges to a smaller value.

$$\boxed{J_{opt} = J(w_N^{*}) = E[x_k x_k^{*}] - p_N^T w_N^{*}}$$

Indian Institute of Technology Delhi 79 Ranjan Bose Department of Electrical Engineering

So to get an optimal tap gain vector W$_N$ hat J$_{opt}$ solved iteratively that's where the recursive nature will come until it converges to a small value, this is what we are working with. Practically MSE is minimized by stochastic gradient algorithm also called the least mean square LMS algorithm, if N denotes the sequence of iteration then LMS is computed iteratively by these following equations. Today these are fairly standard and you can get your weight vector by following. Here N is a number of delay stages in an equalizer and alpha here is a step size which controls the convergence rate and the stability. If you increase the alpha, the rate of convergence increases but your stability will not finally settle down close to the optimal value. So alpha is usually adaptive it can be changed on the fly.

(Refer Slide Time: 00:51:36 min)



Wireless Communications

LMS Algorithm

- To get optimal tap gain vector $\hat{w}_N$, J_{opt} is solved iteratively until it converges to a smaller value.
- Practically MSE is minimized by stochastic gradient algorithm also called least mean square (LMS) algorithm.
- If n denotes sequence of iterations, then LMS is computed iteratively by:

$$\boxed{\hat{d}_k(n) = w_N^T(n) y_N(n)}$$

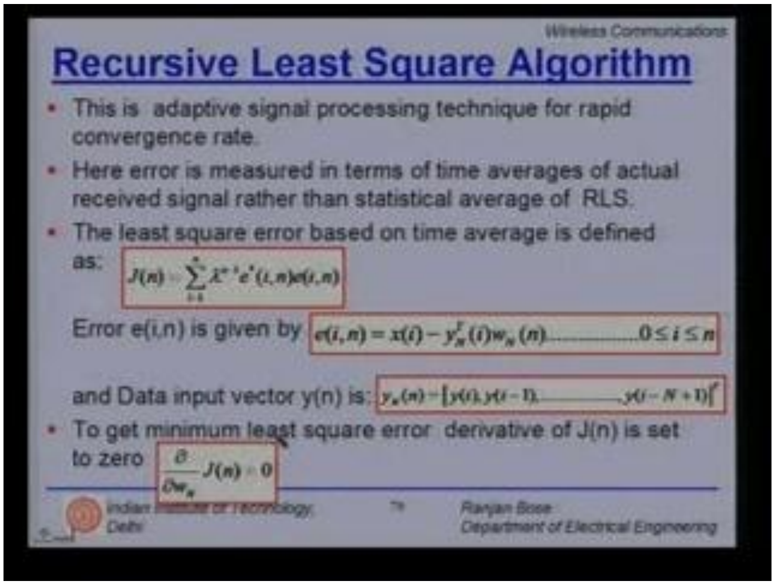
$$\boxed{e_k(n) = x_k(n) - \hat{d}_k(n)}$$

$$\boxed{w_N(n+1) = w_N(n) - \alpha e_k^{*}(n) y_N(n)}$$

- where N is number of delay stages in equalizer, and alpha is step size which controls the convergence rate and stability of algorithm.

Indian Institute of Technology Delhi 79 Ranjan Bose Department of Electrical Engineering

(Refer Slide Time: 00:52:27 min)



Wireless Communications

Recursive Least Square Algorithm

- This is adaptive signal processing technique for rapid convergence rate.
- Here error is measured in terms of time averages of actual received signal rather than statistical average of RLS.
- The least square error based on time average is defined as: $J(n) = \sum_{i=1}^{n} \lambda^{n-1} e^{*}(i,n)e(i,n)$

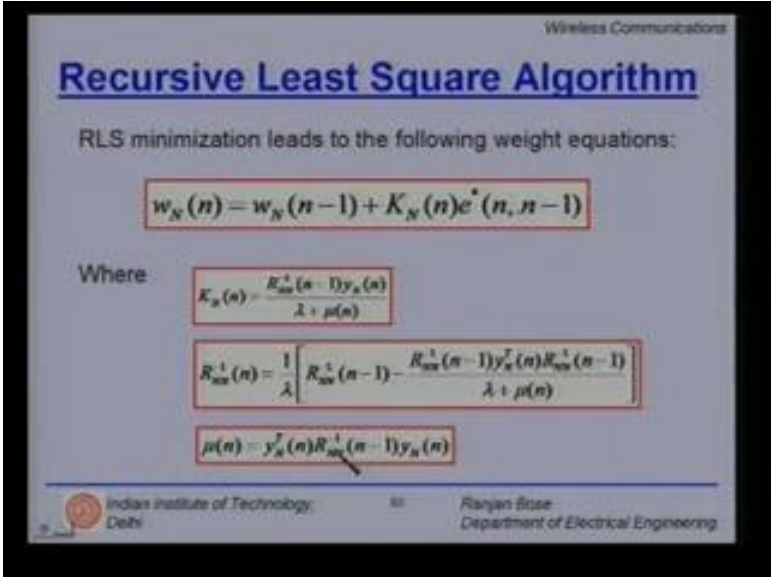
  Error e(i,n) is given by $e(i,n) = x(i) - y_N^T(i)w_N(n) \ldots\ldots\ldots 0 \le i \le n$

  and Data input vector y(n) is: $y_N(n) = [y(i), y(i-1), \ldots\ldots\ldots, y(i-N+1)]^T$
- To get minimum least square error derivative of J(n) is set to zero $\frac{\partial}{\partial w_N} J(n) = 0$

Indian Institute of Technology Delhi 79 Ranjan Bose Department of Electrical Engineering

Let us now come to the final RLS or recursive least square algorithm. This is an adaptive signal processing technique for rapid convergence rate. LMS is usually known to be slow, RLS on the other side is a much faster technique. Here error is measured in terms of time averages of actual received signal rather than the statistical averages of RLS. Here the least square error based on time average is defined as follows. So this is what we have to minimize. See in LMS algorithm, if the Eigen values of the input covariance matrix lambda vary too much that is lambda max over lambda min is much greater than one, it will take a long time to converge. So we have to put that thing into effect which is done by the RLS algorithm. The error is given by the following equation x (i) minus delayed version vector and the weight vector and the data input vector is defined as follows. To get the minimum least square error the derivative of J_N is set to zero.

(Refer Slide Time: 00:54:02 min)



Wireless Communications

Recursive Least Square Algorithm

RLS minimization leads to the following weight equations:

$$w_N(n) = w_N(n-1) + K_N(n)e^{*}(n, n-1)$$

Where

$$K_N(n) = \frac{R_{NN}^{-1}(n-1)y_N(n)}{\lambda + \mu(n)}$$

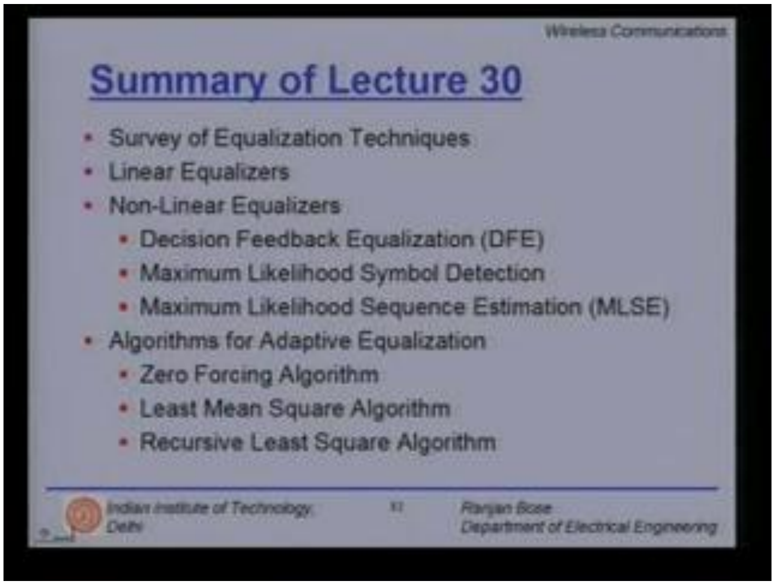
$$R_{NN}^{-1}(n) = \frac{1}{\lambda}\left[R_{NN}^{-1}(n-1) - \frac{R_{NN}^{-1}(n-1)y_N^T(n)R_{NN}^{-1}(n-1)}{\lambda + \mu(n)}\right]$$

$$\mu(n) = y_N^T(n)R_{NN}^{-1}(n-1)y_N(n)$$

Indian Institute of Technology Delhi 80 Ranjan Bose Department of Electrical Engineering

Here in this slide RLS minimization leads to the following weight equation, so we are not going through the whole derivation again you can consult ((00:54:18 min)) digital communications to go through derivations but this is the final crux of the matter. This is the weight vector definition, here K_N is given by this following equation and R_{NN} inverse can be expressed as follows. Thus the input covariance matrix and mu (n) which is used here can be found by this method.

(Refer Slide Time: 00:54:51 min)



Wireless Communications

Summary of Lecture 30

- Survey of Equalization Techniques
- Linear Equalizers
- Non-Linear Equalizers
  - Decision Feedback Equalization (DFE)
  - Maximum Likelihood Symbol Detection
  - Maximum Likelihood Sequence Estimation (MLSE)
- Algorithms for Adaptive Equalization
  - Zero Forcing Algorithm
  - Least Mean Square Algorithm
  - Recursive Least Square Algorithm

Indian Institute of Technology, Delhi   33   Ranjan Bose
Department of Electrical Engineering

So let us now summarize today's lecture. We started off with a brief survey of equalization techniques, then we moved on to linear equalizers. We then looked at nonlinear equalizer structures and talked about the decision feedback equalizer, the maximum likelihood symbol detection and finally the maximum likelihood sequence estimation. We also looked at three important algorithms for adaptive equalization, the zero forcing algorithm, the least means square algorithm and the recursive least square algorithm. We will conclude our talk here.