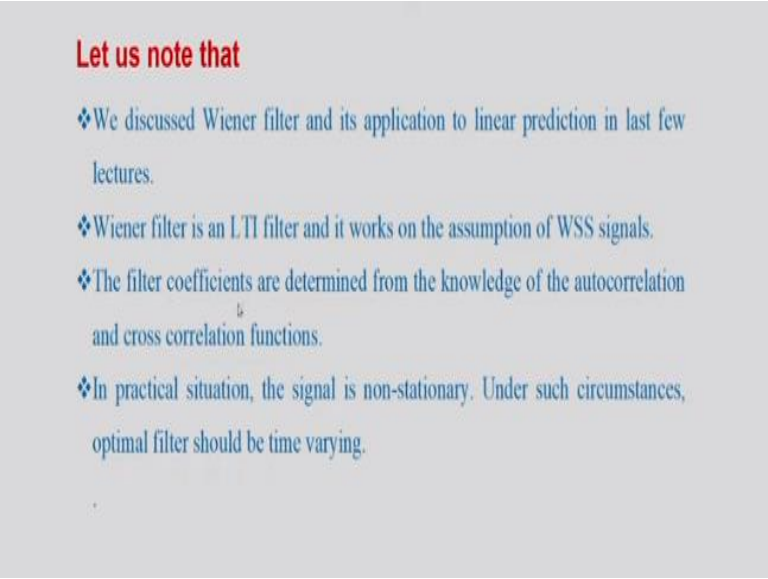
Statistical Signal Processing
Prof. Prabin Kumar Bora
Department of Electronics and Electrical Engineering
Indian Institute of Technology – Guwahati

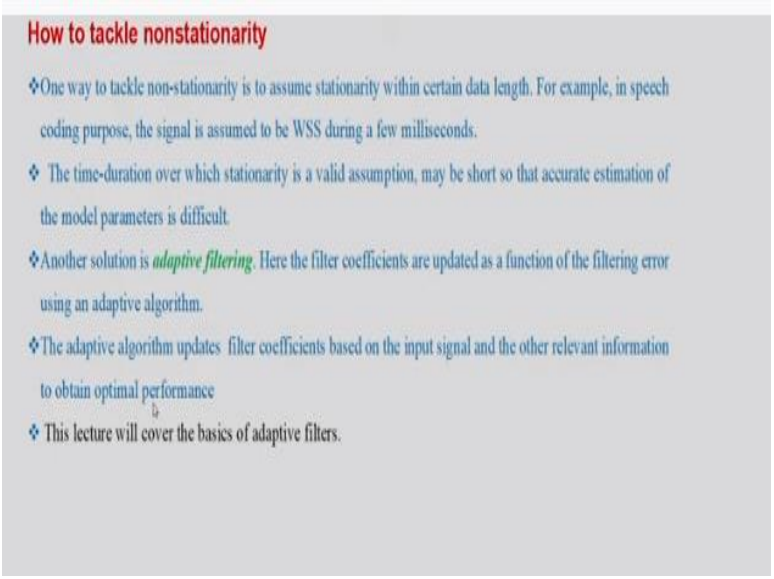
Lecture – 25
Adaptive Filters 1

(Refer Slide Time: 00:42)



Hello students welcome to the lecture on adaptive filters. Let us note that we discussed wiener filter and its application to linear prediction in last few lectures. Wiener filter is an LTI filter linear time-invariant filter and it works on the assumption of WSS signal wide sense stationary signals. The filter coefficients are determined from the knowledge of the autocorrelation and cross correlation functions of the signals used. In practical situation this signal is non-stationary under such circumstances optimal filter should be time varying.

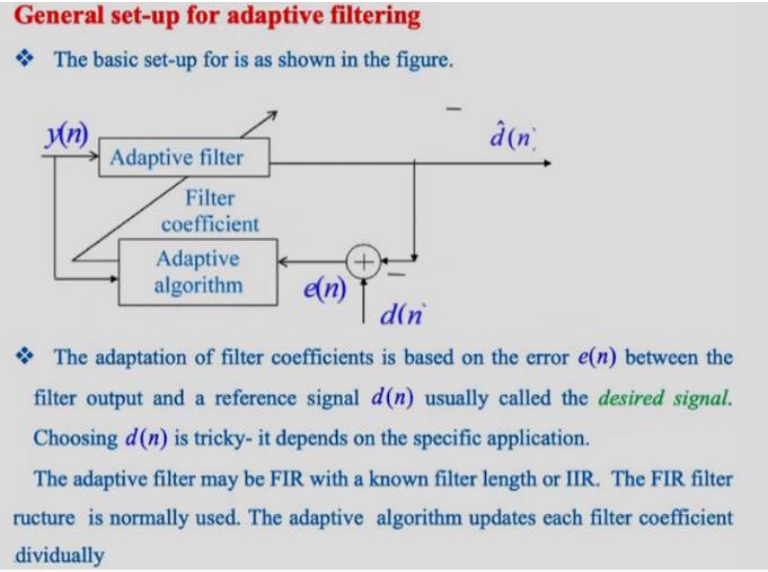
(Refer Slide Time: 01:05)



Now how to tackle this non-stationarity one way to tackle non-stationarity is to assume stationarity within a certain data length. For example in speech coding purpose the signal is assumed to be WSS during a few milliseconds. This approach has a drawback the time duration over which stationarity is valid may be short. So that accurate estimation of the model parameters is difficult.

Next solution is the adaptive filtering here the filter coefficients are updated as a function of deep filtering error using an adaptive algorithm. So in adaptive filtering we have an adaptive algorithm which will update the filter coefficients as a function of the filtering error. The adaptive algorithm updates the filter coefficients based on the input signal and other relevant information to obtain optimal performance. This lecture will cover the basics of adaptive filters.

(Refer Slide Time: 02:45)

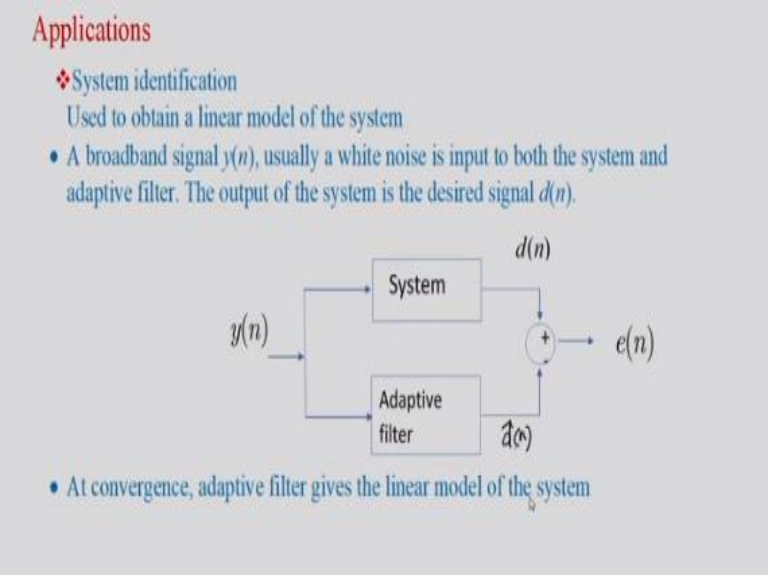


This general setup for adaptive filter is as shown in this so here yn this noisy signal is the input to the adaptive filter and there is the adaptive algorithm and d hat n is the output of this adaptive filter. The adaptation of the filter coefficients is based on the error en between the filter output d hat n and the reference signal dn usually called the desired signal. Now in the adaptive filter terminology we have a desired signal dn and the difference between dn and d hat n.

And that is the error this error is sent to the adaptive algorithm for updating the filter coefficients and these filter coefficients will be used in this adaptive filter. Choosing dn how to choose this desired signal dn is tricky it depends on this specific application. This adaptive filter may be a FIR with known filter length, or it may be IIR. The FIR filter structure is normally used the adaptive algorithm updates is a FIR filter coefficient individually.

We saw the general setup for adaptive filtering here yn that is the input it passes through the adaptive filter to estimate the desired signal and this estimated signal is compared with dn and the error is passed to the adaptive algorithm to update the filter coefficient. And this updated filter coefficients are used here. So this is the basic setup for adaptive filtering.

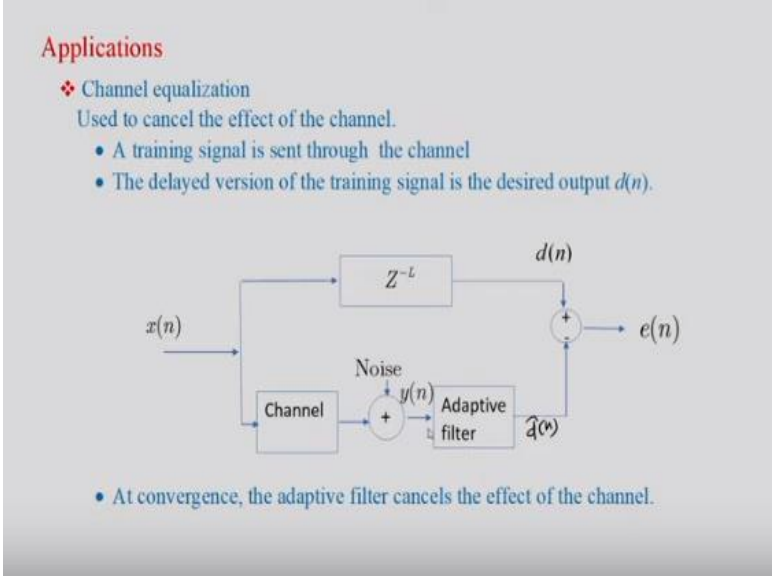
(Refer Slide Time: 04:44)



We will see how the desired signal dn can be obtained in different applications first one is system identification application here this is the system with unknown model parameters. So we have to identify the model parameter that is this system identification problem and here a broadband signal yn is usually white noise is input to both the system and the adaptive filter. Now the output of the system is the desired signal dn so this dn is passed through the for error calculation.

So the adaptive filter I will estimate d hat n and the error will be computed this error en will be now used to update the adaptive filter coefficients that block we have not shown but this error will be sent through to the adaptive filter algorithm to compute the filter coefficient iteratively. So that way this is the adaptive filter used for system identification when the updation the adaptive filter algorithm converges, the corresponding filter parameters will give the linear model of this system. So therefore at convergent adaptive filter gives the linear model of the system.

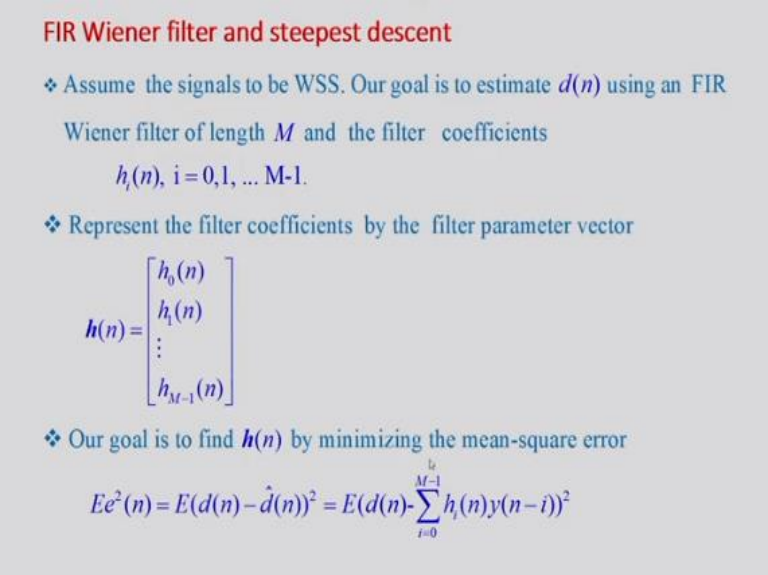
(Refer Slide Time: 06:16)



We will consider another application channel equalization it is used to cancel the effect of this channel it training signal is sent through this channel this is xn is the training signal suppose it is again it is a white noise usually and through some delay operation we will get the desired signal in this case the receiver knows this signal and xn the training sequence is passed through this channel noise is added here.

So we will get yn and this yn is passed through the adaptive filter to estimate d hat n. So we will get d hat n here and now this error will pass will be passed to the adaptive algorithm to compute the filter coefficients iteratively. So that way we saw that this adaptive filter now can cancel the effect of the channel. At convergence the adaptive filter, cancel the effect of the channel when this noise is not there this adaptive filter will be the inverse filter corresponding to this channel. So that this channel equalization basically we can say that it finds an inverse filter to cancel out the effect of channel.

(Refer Slide Time: 07:51)



Now we will consider one important algorithm what is known as this steepest descent algorithm FIR Wiener filter and steepest descent algorithm assume the signals to be WSS here in the adaptive filter structure we assume that yn and suppose dn they are WSS. Our goal is to estimate dn using an FIR Wiener filter of length m and the filter coefficient is hin i going from 0 to m - 1 i is the index for filter coefficients and this n may be the time or it may be representing the number of iterations in the case of iterative algorithm.

Represent the filter coefficients by the filter parameter vector hn vector = h0n, h1n up to hM - 1 and so this is m length filter coefficient vector. Our goal is to find hn by minimizing the mean square error and mean square error is expectation of e square n that is E of dn - d hat whole square = E of dn - summation hi n yn - i i going from 0 to m - 1 whole square.

(Refer Slide Time: 09:23)

FIR Wiener filter and steepest descent ...

❖ Representing the observed signals as a vector

$$\mathbf{y}(n) = \begin{bmatrix} y(n) \\ y(n-1) \\ \vdots \\ y(n-M+1) \end{bmatrix}, \text{ we get}$$

$$Ee^2(n) = E(d(n) - \mathbf{h}'(n)\mathbf{y}(n))^2 = E\left(d^2(n) - 2d(n)\mathbf{h}'(n)\mathbf{y}(n) + \mathbf{h}'(n)\mathbf{y}(n)\mathbf{y}'(n)\mathbf{h}(n)\right)$$

$$= R_d(0) - 2\mathbf{h}'(n)\mathbf{r}_{dY} + \mathbf{h}'(n)\mathbf{R}_Y\mathbf{h}(n)$$

Where

$$\mathbf{r}_{dY} = \begin{bmatrix} R_{dY}(0) \\ R_{dY}(1) \\ \vdots \\ R_{dY}(M-1) \end{bmatrix} \text{ and } \mathbf{R}_Y = \begin{bmatrix} R_Y(0) & R_Y(1) & \ldots & R_Y(M-1) \\ R_Y(1) & R_Y(0) & \ldots & R_Y(M-2) \\ \ldots & & & \\ R_Y(M-1) & R_Y(M-2) & \ldots & R_Y(0) \end{bmatrix}$$

Now we represent this optimization problem in the matrix form for that we represent yn this specter yn vector = a vector comprising of yn, yn - 1 up to yn - m + 1 therefore E of e square n = E of dn - h transpose n into yn whole square and = E of d square n - twice dn into h transpose n into yn + h transpose n yn, yn transpose into hn and after taking the expectation we will get our term will be Rd 0. Similarly second term will be 2 h transpose n into R dy + hn transpose into Ry into hn. So this is the mean square prediction error.

(Refer Slide Time: 10:21)

FIR Wiener filter and steepest descent ...

❖ The Wiener filtering problem can be written as

Minimize $Ee^2(n)$ (1)

With respect to the filter coefficient vector $\boldsymbol{h}(n)$

❖ The cost function represented $Ee^2(n)$ is a quadratic in $\boldsymbol{h}(n)$ and a unique global minimum exists

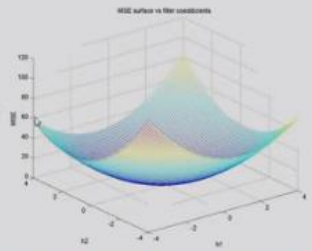
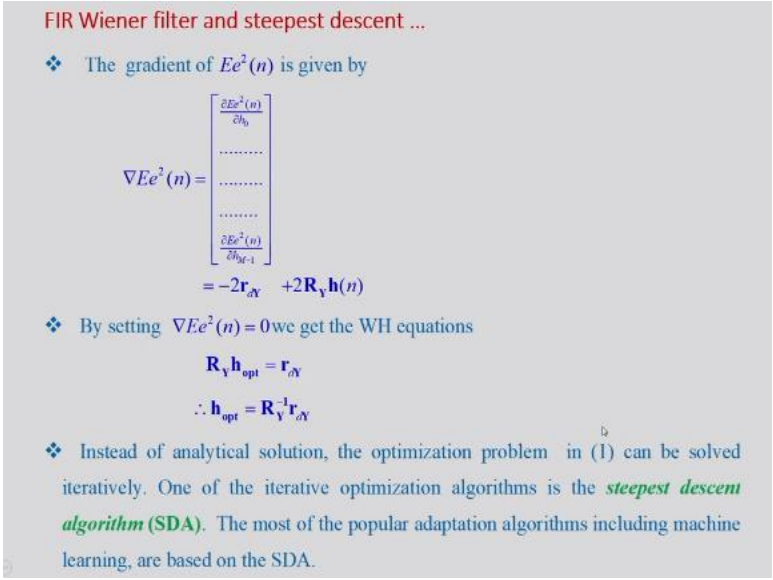


Figure - Cost Function $Ee^2(n)$ for a length 2 FIR Wiener filter

The wiener filtering problem can be written as minimize E of e square n with respect to the filter coefficient vector hn. The cost function represented as E of e square n is a Quadratic function in terms of hn and therefore a unique global minimum exist. Here we are illustrating with the help

of a 2 parameter filter, so this is h 1, h2 and this is the MSE. If we plot MSE versus h1, h2 we will get a quadratic surface like this for the surface 1 optimum will exist.

(Refer Slide Time: 11:07)

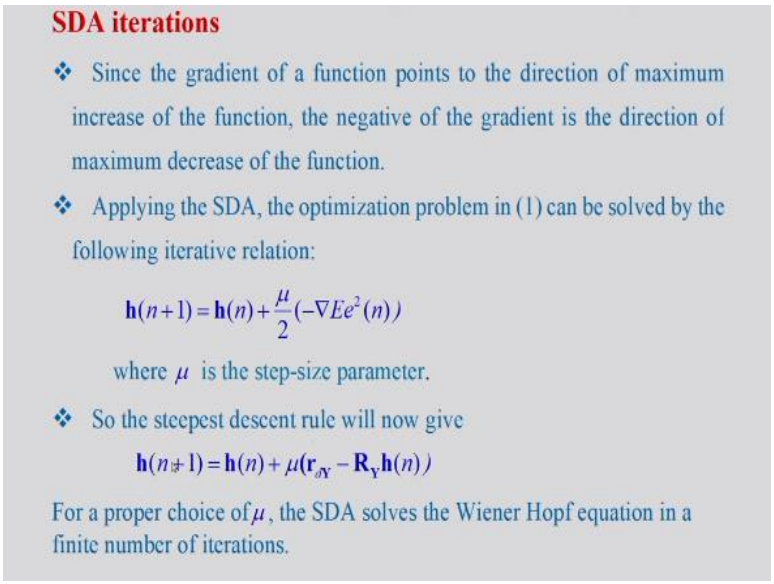


The gradient of E of e square n is given by a gradient of E of e square your V E square n is a scalar gradient is a vector so this = del del h 0 of E of e square n even next element will be del del h1 of E of e square n like that the last element will be del E of e square n del h M - 1 and if I compute this vector from this expression if I now take the partial derivative this term will go down to 0 so that way we will get this expression – twice rdY + twice Ry into hn for the minimum gradient of E of e square n = 0.

So that we get the Wiener Hopf equation Ry into h optimum = rdY. So autocorrelation matrix into coefficient vector is equal to cross correlation vector and we can get h optimum by this relationship Ry inverse into rdY. Instead of this analytical solution the optimization problem in 1 this optimization problem can be solved iteratively. One of the iterative optimization algorithm is this the steepest descent algorithm SDA.

The most of the popular adaptation algorithms including machine learning are based on the SDA so that way as this steepest descent algorithms are very popular from machine learning to deep learning this algorithm is used.

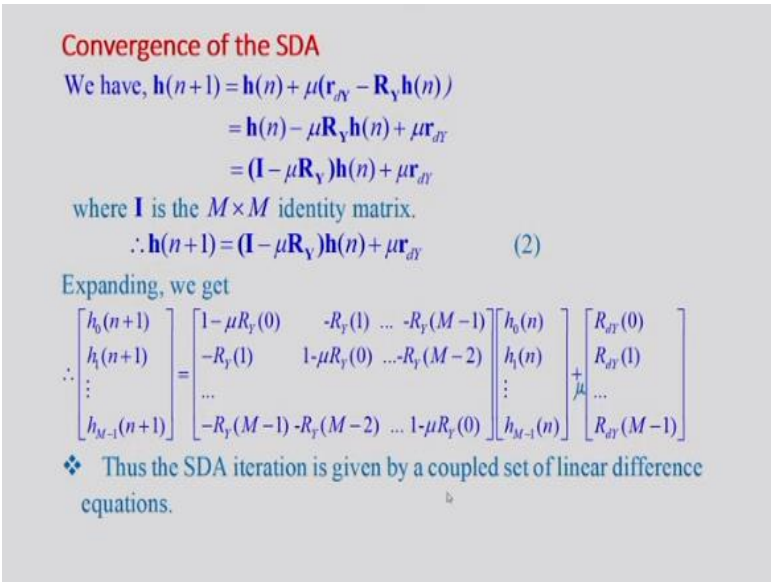
(Refer Slide Time: 13:00)



Now let us see how SDA iterations are done since the gradient of a function points to the direction of maximum increase of the function the negative of the gradient is the direction of maximum decrease of the function this is the logic. So negative of the gradient gives the direction of maximum decrease of the function. Applying SDA the optimization problem can be now solved by the following iterative relation h of n + 1 = earlier hn + mu / 2 multiplied by negative of the gradient.

Now this 2 is just to scale this part generally this part will give a 2 terms you can tell that we have used 2 here but mu is the step size parameter or learning parameter. So this steepest descent rule will now give h of n + 1 = hn + mu times rdY - Ry into hn by completing gradient of E of e square n we get this expression. For a proper choice of mu this SDA solves the Wiener Hopf equation in a finite number of iterations.

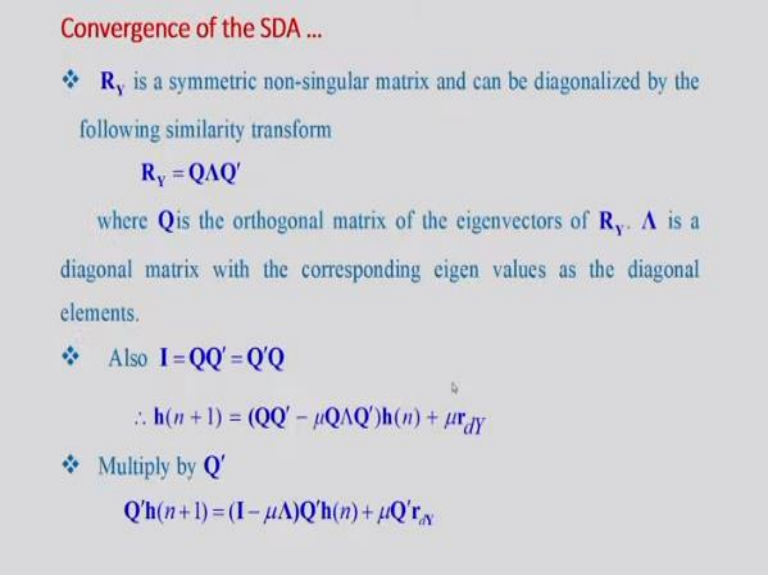
(Refer Slide Time: 14:27)



Now let us discuss about the convergence of the SDA we have h of n + 1 = hn + mu times rdY - Ry into hn and this we can rewrite as hn - mu times Ry into hn then + mu times rdY and if I take hn as common this will be I - mu Ry into hn + mu into rdY where I is the M / M identity matrix. Therefore, the SDA iteration step can be written as h of n + 1 = i - mu Ry into hn + mu rdY. Expanding this expression, we get suppose this one is the hn + 1 vector.

And this is the i - mu ry matrix and this is the hn vector is n + 1 = i - mu ry matrix into hn this is i - mu Ry matrix + mu times and this rdY vector. From this we see that the SDA direction is given by a couple set of linear difference equation for example the party prince equation will be h0n + 1 is equal to this row multiplied by this column. So it will involve all these coefficients + mu times rdY0 so that will be the first equation which involves all the filter coefficients in the previous days. So that way it is a couple set of linear difference equations. For convergence analysis we want to represent this set of difference equation into a uncouple set of difference equations.

(Refer Slide Time: 16:52)

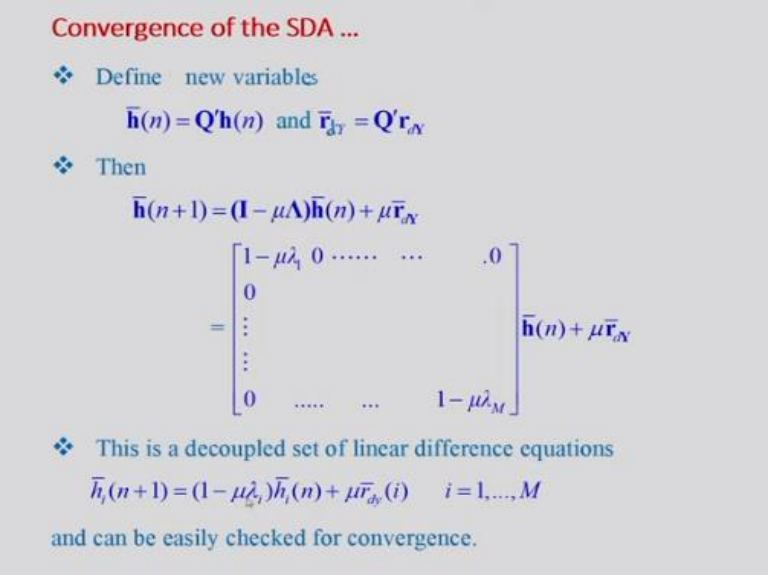


So for that we do this similarity transformation Ry is a symmetric non-singular matrix and can be diagonalized by the following similarity transform Ry = Q lambda into Q transpose where Q is the orthogonal matrix of the eigenvectors of Ry. Lambda is a diagonal matrix with the corresponding eigen values as the diagonal elements. So Q is the matrix of the eigen vectors and Q is orthogonal it means Q transpose Q = Q, Q transpose = identity matrix that is the orthogonal matrix.

So that way in this expression we can write Ry as Q lambda into Q transpose also this I matrix we can write as Q into Q transpose identity matrix can be expressed as the product of Q and Q transpose. So that way because Q transpose is Q inverse and that = Q transpose into Q therefore the SDA iteration is h of n + 1 = I - mu mu Ry into hn + mu rdY. This iteration can be written as h of n + 1 = Q, Q transpose - mu Q lambda Q transpose into hn + mu rdY.

If we multiply by Q transpose both sides this Q transpose into hn + 1 now if we multiply by Q transpose here Q transpose into Q will be equal to i and here Q transpose into Q will be equal to again i because of that we can take this Q dash as common and what we will have is here right-hand side will have I - mu times lambda into Q transpose hn + mu times Q transpose rdY. So these relationships we get by multiplying Q transpose both sides.

(Refer Slide Time: 19:33)

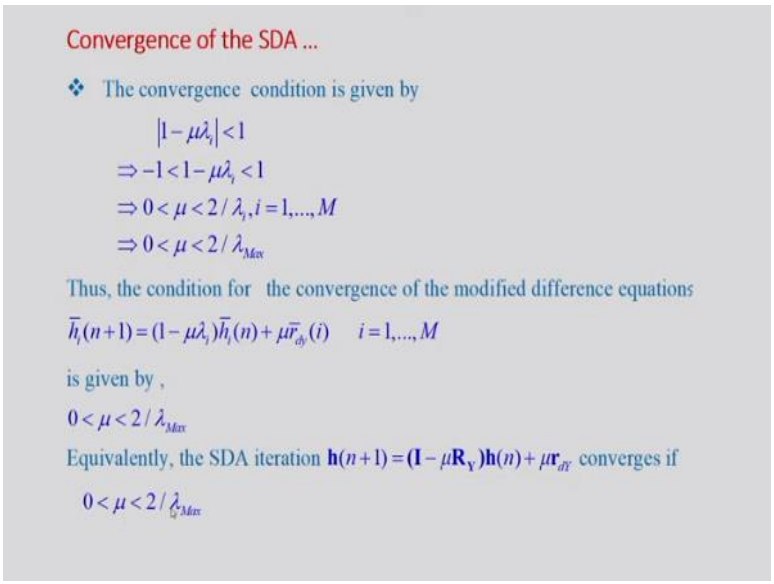


Let us define new variables that is h bar n = Q transpose into hn + rdY bar = Q transpose into rdY. So in terms of these new variables we will now have h bar n + 1. So from this equation this is h bar of n + 1 and this = h bar of n and this = rdY bar. So that way we will have h bar of n + 1 = I - mu lambda into h bar of n + mu times rdY bar. So expanding this we can write this as first term will be 1 - mu lambda 1 and rest of the terms will be = 0.

Similarly here this term will be 0 and next term will be a 1 - mu lambda 2 like that we can complete this matrix multiplied by a square of n + mu rdY bar. Now this set of difference equations is a decouple set of linear difference equations and it will be given by hi bar n + 1 = 1 - mu lambda into hi bar n + mu rdY bar i for i = 1 to m and this can be easily checked for convergence.

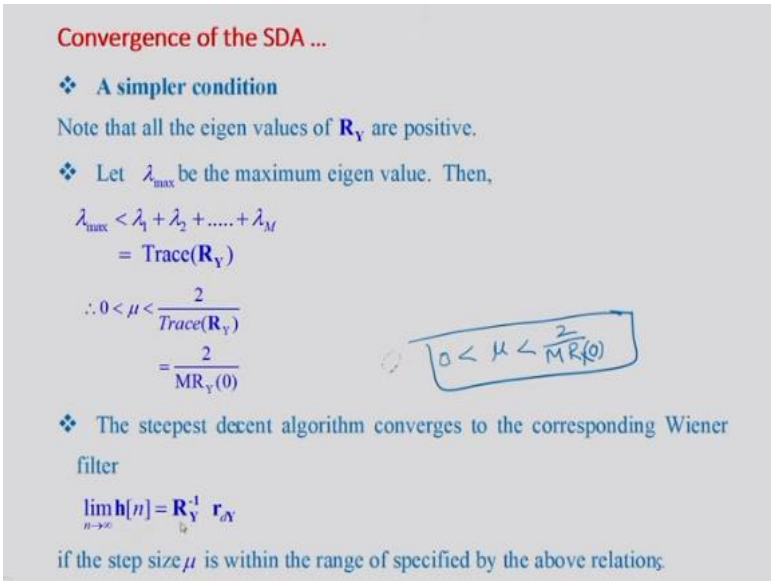
Because now we have a difference equation like this, and this coefficient will determine the convergence. For example the first equation will be h1 n + 1 will be = 1 - mu lambda 1 into h1 n + mu rdY.

(Refer Slide Time: 21:36)



The convergence condition is given by this mod of 1 - mu lambda i is < = 1 for all i = 1 to m so this implies that 1 - mu lambda lies between - 1 and 1. So from this relationship we will get 0 < mu < 2 / lambda i for i = 1 up to m. Thus the condition for the convergence of the modified difference equations hi bar n + 1 = 1 - mu lambda i times hi bar n + mu times rdY bar i so i = 1 up to m is given by mu lies between 0 and 2 / lambda max this is the condition for convergence of this set of difference equations. Equivalently the SDA iteration hn + 1 = i - mu Ry into hn + mu rdY converges if 0 < mu < 2 / lambda max. So this will be the condition for convergence for SDA iteration

(Refer Slide Time: 23:03)

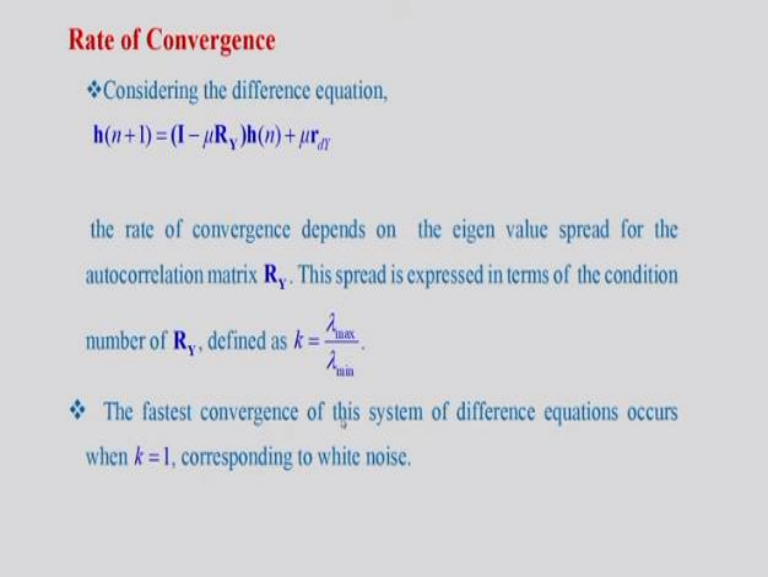


We can get a simpler condition note that all the eigenvalues of Ry are positive. Let lambda max be the maximum eigenvalue then lambda max will be less than equal to sum of all these eigenvalues. Because all are positive therefore lambda max is one of them and definitely it will be less than the sum of the eigen values and this sum of the eigenvalues equal to Trace of Ry. Therefore this condition can be written as 0 < mu < 2 / Trace of Ry.

Also all the diagonal elements of Ry are Ry0 therefore m diagonal elements are there therefore this expression 2 / Trace of Ry will be = 2 / m times Ry0. Therefore the simpler condition for convergence is 0 < mu < 2 / m times Ry0. So this is the condition for convergence, so we have to select mu in such a way that it is less than 2 / m times Ry of 0. Thus this steepest descent algorithm converges to the corresponding Wiener filter limit hn as n tends to infinity = Ry inverse into rdY if the step size mu is within the range specified by the above relations.

So this relation or this relation if we consider these 2 relation for limiting mu then and these steepest decent algorithm will convert to the Wiener solution given by this.

(Refer Slide Time: 25:11)

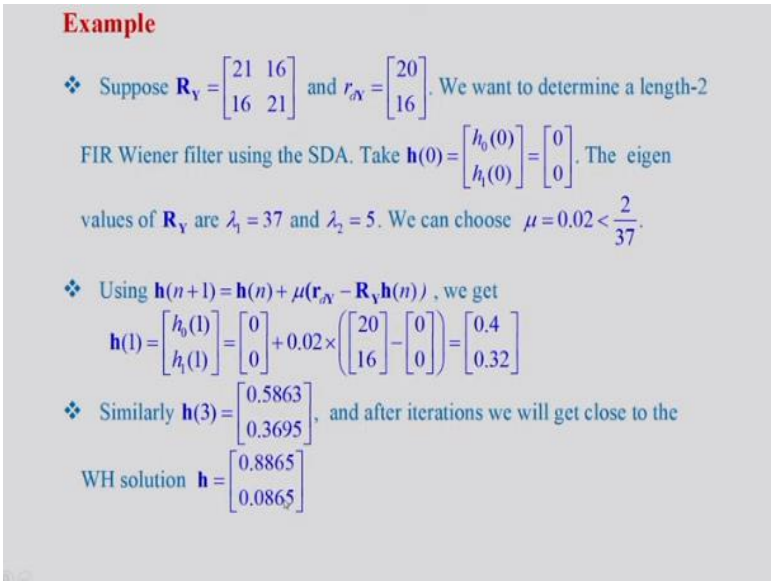


Now let us make a comment about the rate of convergence of the steepest descent algorithm considering the difference equation h of n + 1 = i - mu Ry into hn + mu rdY. The rate of convergence depends on the eigenvalue spread of the autocorrelation matrix Ry. So the spread of the eigenvalues of this matrix will determine the speed of convergence. This spread of the

eigenvalue is expressed in terms of the condition number Ry which is defined as k = lambda max / lambda min.

So if this case large then convergence will be slow. The fastest convergence of the system of difference equations occur when k = 1 and this corresponds to white noise and this also explain why in our system identification problem and channel equalization problem with 2d input sequence as white noise.

(Refer Slide Time: 26:25)

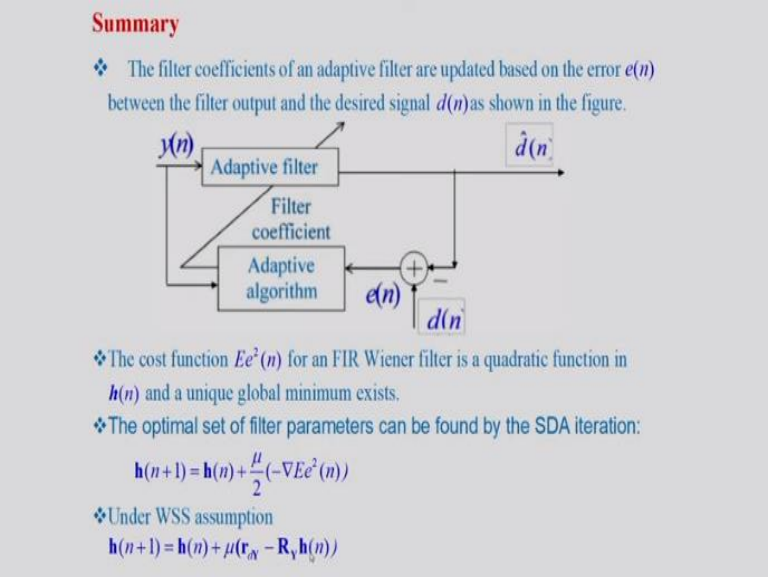


We will consider one example suppose Ry = this matrix diagonal elements are 21 of diagonal elements are 16 and rdY vector that is the autocorrelation vector is 20 16. We want to determine in length to FIR Wiener filter using the SDA take h0 = h0 of 0, h1 of 0 that = 0 0 that is the initial value of h0 the eigenvalues of Ry we can determine lambda 1 = 37 lambda 2 = 5 and therefore you can choose mu = 0.02 which is less than 2 / 37.

So we can consider suppose some of the diagonal elements is 42 therefore we can have a condition that this mu should be less than 2 / 42. So that way also we can select this mu = 0.02 using h of n + 1 = hn + mu times rdY - Ry into hn we get h1 that = h01 h11 that will be = initial value is 0 0 + this is the multiplier 0.02 times our d y vector that is 20 16 and Ry intuition that we will get 0 0 only so that way I will get if we carry out this computation we will get this as 0.4 0.32.

So h1 vector is a vector comprising of 0.4 and 0.32 similarly after the iteration we will get this value h3 vector will be = 0.5863 and 0.3695 meaning that h0 of 3 will be 0.5863 and h1 of 3 will be 0.3695. In this way we can continue and after a number of iterations we will be getting close to the Wiener Hopf solution which is given by h = 0.8865, 0.0865 this is the relationship which we can get by direct matrix inversion.

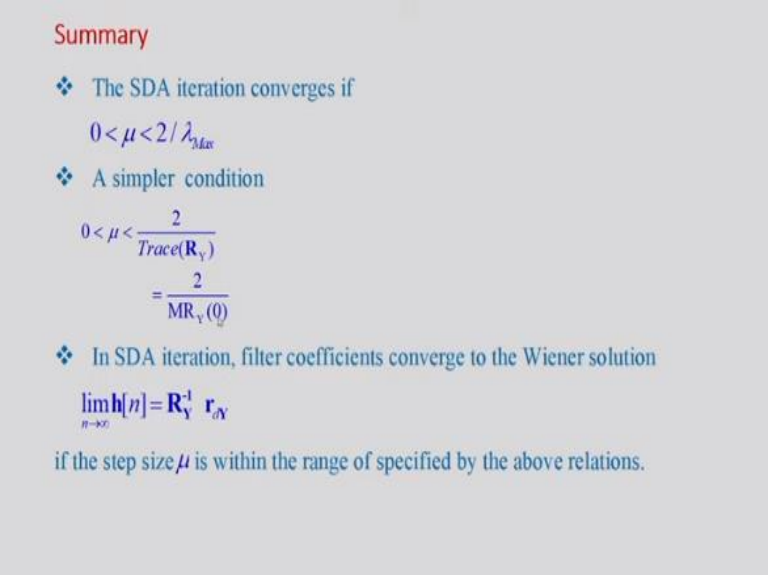
(Refer Slide Time: 29:08)



Let us summarize the lecture the filter coefficients of an adaptive filter are updated based on the error en between the filter output and the desired signal dn as shown in the figure. With this figure we discussed so here this is the adaptive filter and filter coefficients are determined according to the adaptive algorithm and input to this adaptive algorithm is the difference between this desired signal dn and the adaptive filter output d hat n.

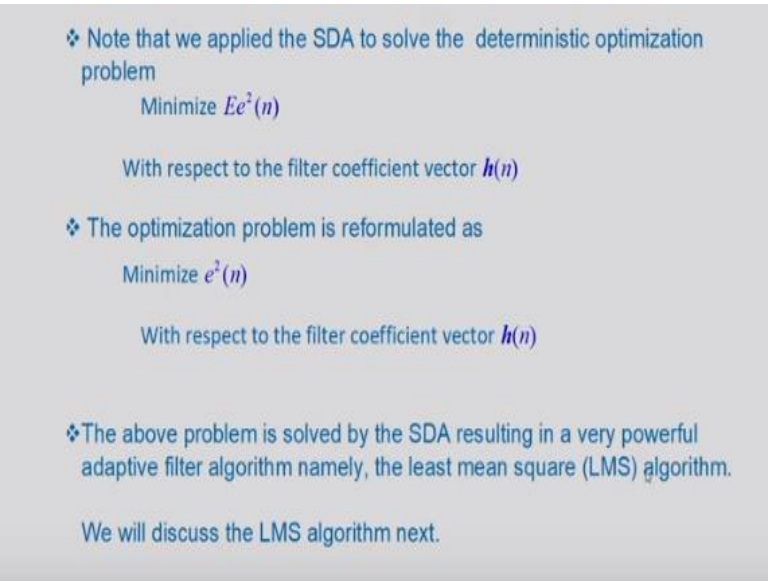
So this error is the input to the adaptive filter algorithm and accordingly the filter coefficients will be updated and fed to this adaptive filter. The cost function E of e square n for an FIR wiener filter is a Quadratic function in a sense and a unique global minimum exists. Now the optimal set of filter parameters can be found by the SDA iteration that we have established h of n + 1 vector = hn vector + mu / 2 times - gradient of E of e square n and under WSS assumption of these signals we get a h of n + 1 = hn + mu times rdY vector - Ry matrix into hn vector.

(Refer Slide Time: 30:54)



We also establish depth the SDA iteration converges if we select mu to be less than 2 / lambda max where lambda max is the maximum eigen value of the autocorrelation matrix. A simpler condition is obtained that is 0 < mu < 2y Trace of Ry matrix and this Trace = m times Ry of 0 therefore the SDA iteration will be convergent if we select mu < 2 / MRy0. In SDA iterations the filter coefficients converge to the Wiener solution that is limit of hn as n tends to infinity = Ry inverse into small rdY if this step size mu is within the range specified by the above relations that is this relation and this relation.

(Refer Slide Time: 31:53)



Note that we applied the SDA to solve a deterministic optimization problem that is minimized E of e square n with respect to the filter coefficient vector hn. The optimization problem is

reformulated as minimize instead of E of e square n we will write e square n with respect to the filter coefficient vector hn. The above problem is solved by the SDA resulting in a powerful adaptive filter algorithm namely the least mean square LMS algorithm. We will discuss the LMS algorithm next. Thank you.