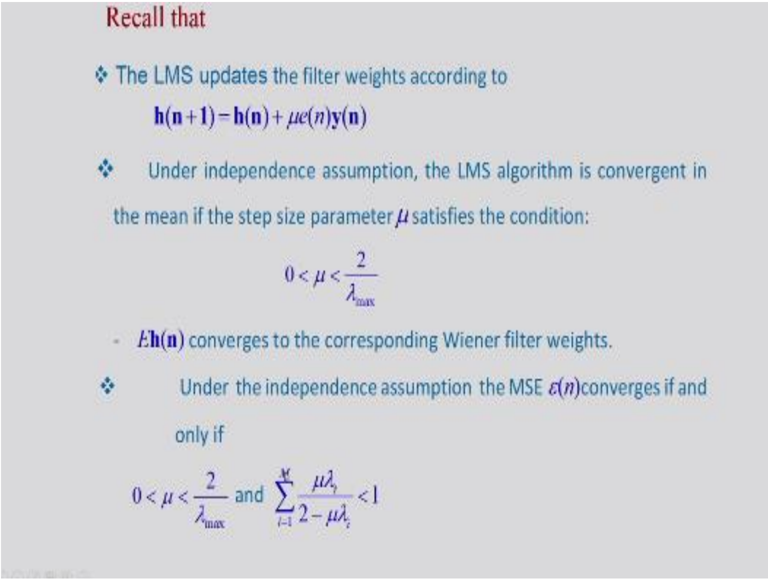
Statistical Signal Processing

Prof. Prabin Kumar Bora
Department of Electronics and Electrical Engineering
Indian Institute of Science – Guwahati

Lecture - 29
Adaptive Filters 4

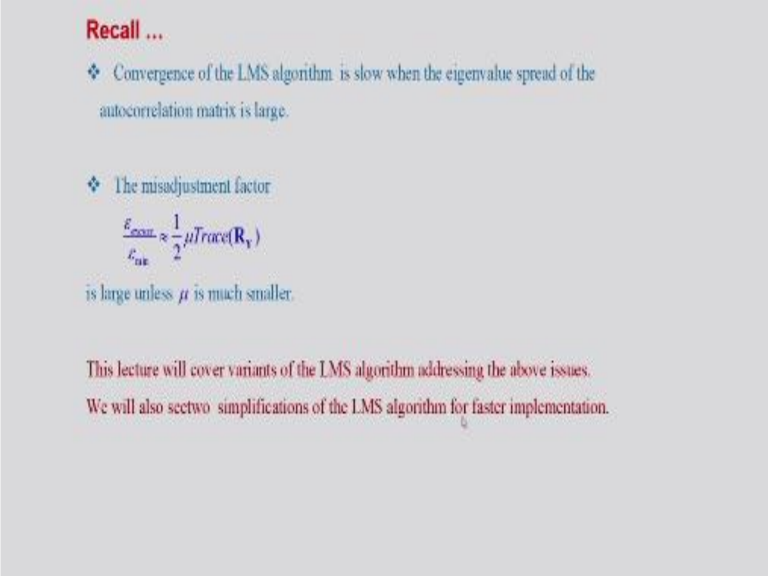
(Refer Slide Time: 00:38)



Hello students. Welcome to this lecture on Adaptive Filters. Recall that the LMS algorithm updates the filter with according to this relation h of n + 1 = hn + mu times en into yn yn is the data vector, en is the error and hn is the filter coefficient at instant n. Under the independent assumption the LMS algorithm is convergent in the mean if the step size parameter mu satisfy this conditions we know that mu < 2/lambda max.

And E of hn converges to the corresponding Wiener filter weights as hn tends to infinity E of hn converges to the corresponding Wiener filter weights. Under the independent assumption the MSE epsilon n converges if and only if again first condition is like this mu < 2/lambda max and the second condition summation mu lambda i/2 – mu lambda i, I going from 1 to m < 1.

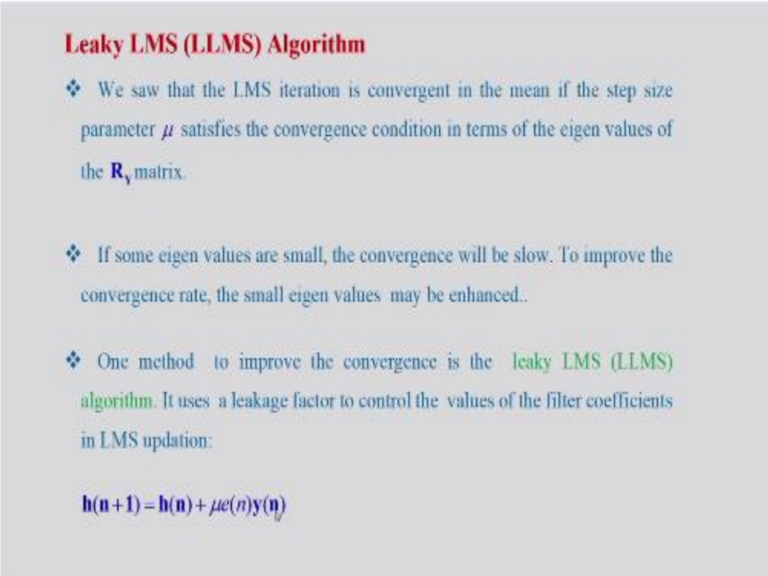
(Refer Slide Time: 02:11)



We also noted that the convergence of the LMS algorithm is slow when the eigenvalue spread of the autocorrelation matrix is large and there is a misadjustment factor that is the ratio of the excess mean square error divided by minimum mean square error corresponding to Wiener filter and that is approximately = half of mu times Trace of RY. Therefore, this misadjustment factor will be large if mu is large therefore mu should be small.

This lecture will cover the variance of LMS algorithm addressing the above issues convergence issues. We also see two simplifications of the LMS algorithm for faster implementation.

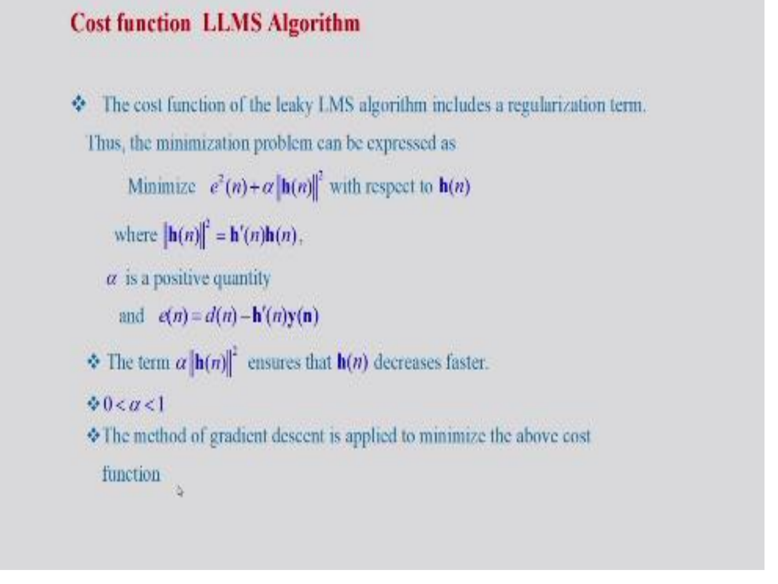
(Refer Slide Time: 03:10)



First we will start with the leaky LMS LLMS algorithm. We saw that the LMS iteration is convergent in the mean if the step size parameter mu satisfies the convergence condition in terms of the eigen values of RY matrix. So the convergence is determined by the eigen values of RY matrix. If some eigen values are small the convergence will be slow. Thus if anyone of the eigen values is small then the rate of convergence will be affected.

Therefore, to improve the convergence rate the small eigen values maybe enhanced. One method to improve the convergent is the leaky LMS LLMS algorithm. It uses a leakage factors to control the values of the filter coefficient in LMS updation this is the LMS updation h of n + 1 = hn + mu en into yn vector.

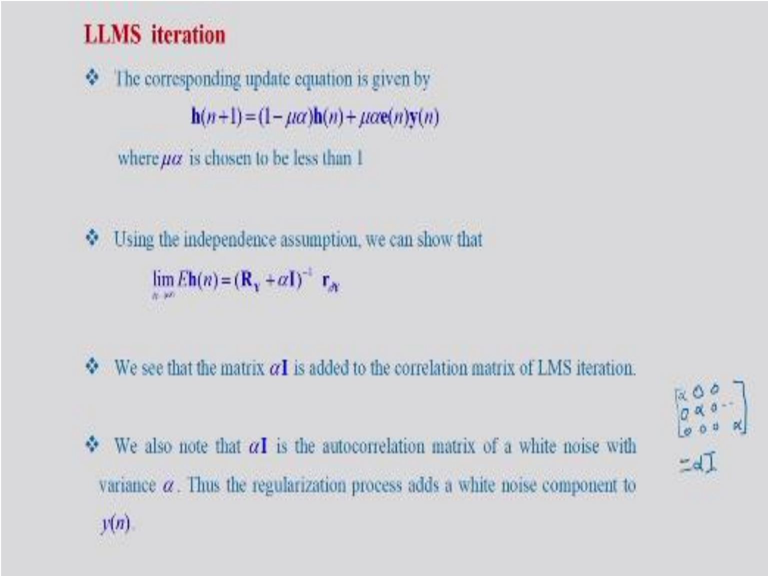
(Refer Slide Time: 04:35)



Let us examine the cost function for LLMS algorithm the cost function of the leaky LMS algorithm includes a regularization term thus the minimization problems can be expressed as minimize e square n + alpha times norm of hn square with respect to hn where norm of hn square = h transpose n times hn this is the (()) (05:08) norm alpha is a positive quantity and en is dn minus that is the desired signal - h dash n into yn this is the filter output.

So here e square n is the cost function corresponding to the LMS algorithm and this term alpha times norm of hn square is the regularization term. The term alpha times norm of hn square ensures that hn decreases faster because if it tries to increase because of this positive value the cost function will increase so our aim to minimize the cost function. Alpha is taken between 0 and 1 and the method of gradient descent is applied to minimize the above cost function just like in the case of LMS algorithm.

(Refer Slide Time: 06:18)



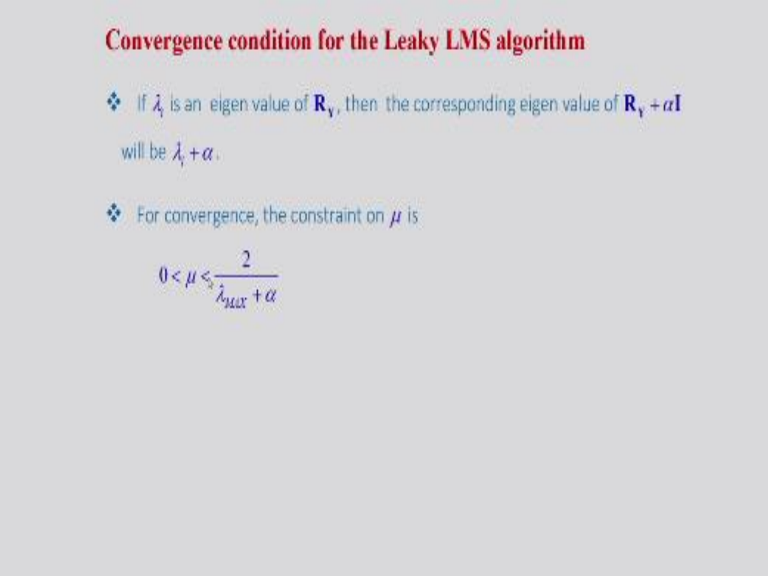
The corresponding update equation is given by h n + 1 = 1 – mu alpha times hn + mu alpha into en into yn vector. So the leakage term is introduced here now where mu alpha is chosen to be less than 1 because of this factor is to be positive so mu alpha must be less than 1. Using the independence assumption we can show that limit of E of hn as n tends to infinity will be = RY + alpha I inverse into rdY.

Limit E of hn as n tends to infinity = RY + alpha I whole inverse into small rdY this is the cross correlation between d and Y. Note that this is different from the corresponding Wiener filter solution. We see that the matrix alpha I is added to the correlation matrix of LMS iteration. LMS iteration correlation matrix is RY in leaky LMS this matrix is added alpha times I matrix is added.

We also note that alpha I, is the autocorrelation matrix of a white noise with variance alpha. So if we have a white noise with variance alpha then it is autocorrelation function will be alpha then rest of the elements in the first row will be 0 like that. Similarly second row this will be 0 then alpha 0 like that and finally the last row will be 0, 0, 0 and then last element will be alpha.

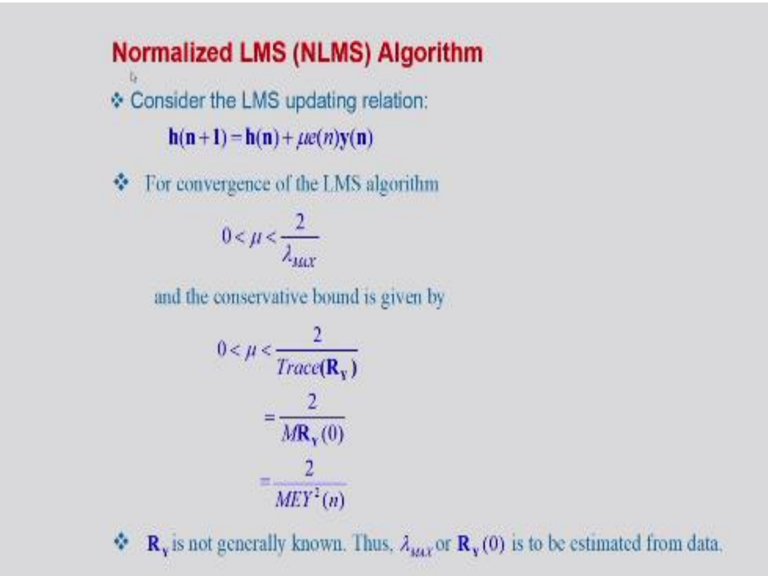
So that way this will be the autocorrelation function of the white noise which is same as alpha times I. So, therefore the regularization process a white noise component is added to yn to the data a white noise component is added that way is pre whitening of data.

(Refer Slide Time: 09:03)



Now let us examine the convergence condition for the leaky LMS algorithm. If lambda i is an eigen value of RY then the corresponding eigen value of RY + alpha I will be lambda i + alpha. So if eigenvalue will be increase by alpha. Therefore, for convergence the constraint on mu is mu should lie between 0 and 2/lambda max + alpha.

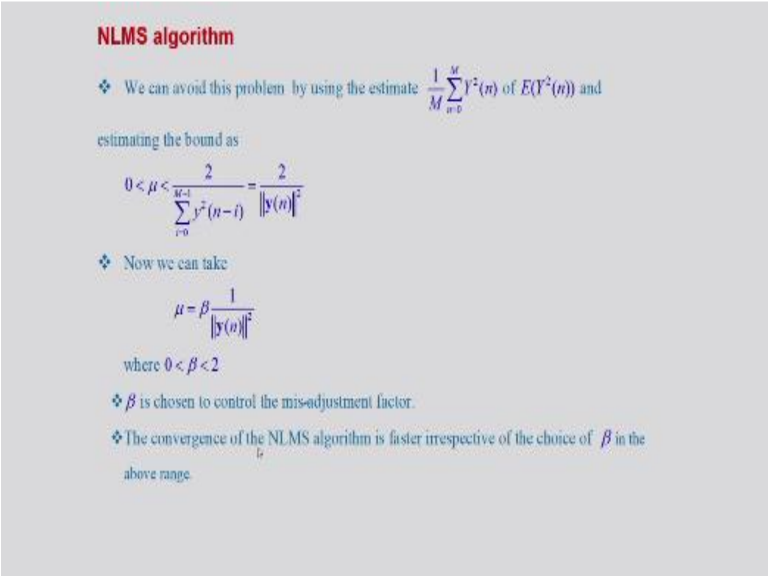
(Refer Slide Time: 09:39)



Next algorithm we will consider the normalize LMS NLMS algorithm. Consider the LMS update equation given by h of n + 1 = hn + mu times e of n into yn vector. For converge this mu should be < 2 / lambda max and conservative bound for convergence is given by mu < 2 / Trace of RY and that = 2/M times M is the number of filter coefficients M times RY 0 and RY 0 we can write it as expected value of E of Y square n.

Therefore for convergent mu should be < 2/M times E of Y square n. Note that RY is not generally known. Thus, lambda max or RY of 0 is to be estimated from the data. This expression E of Y square n is to be estimated from data.

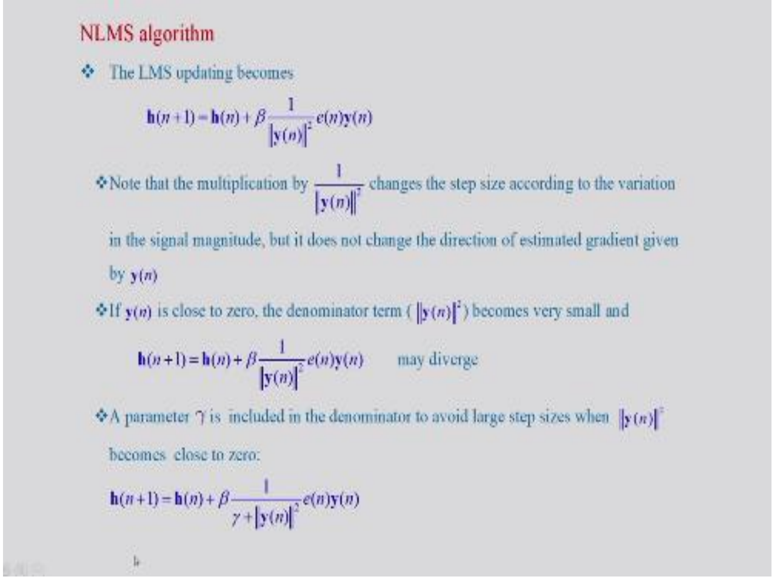
(Refer Slide Time: 11:03)



We can avoid this problem by using the estimate 1/M times summation Y square n n going from 0 to M in place of E of Y square n and estimating the bound as 0 is < mu < 2 by this expression because M and 1/M will get cancelled therefore 0 is < mu < 2 by summation y square n – i, i going from 0 to M - 1 and this is = 2/norm of yn square. Now we can take mu because mu < 2 by norm of yn square.

So we can take mu = beta times 1 norm of yn square where beta is a number between 0 and 2. We can take any value of beta between 0 to 2, but it is so then to control the mis adjustment factors small r value of beta will give less misadjustment factor. The convergence of the NLMS algorithm is faster irrespective of the choice of beta in the above range. If we choose beta in this range the convergence of the NLMS algorithm will be faster than the corresponding speed of the LMS algorithm.

(Refer Slide Time: 12:47)

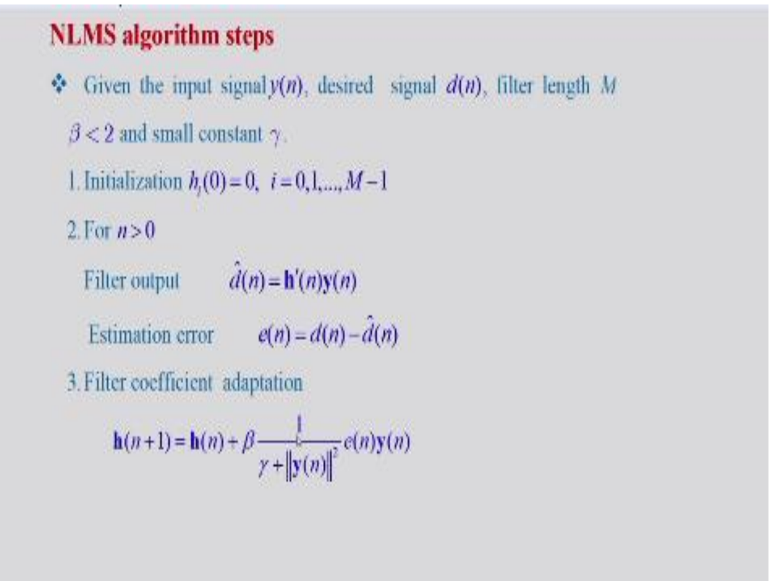


Thus the LMS updating becomes h of n + 1 = hn + beta times 1 / norm of yn square times en into yn. Note that multiplication by 1 / norm of yn square changes the step size according to the variance of the signal magnitude, but it does not change the direction of the gradient given by yn because yn will specify the direction of the gradient this yn is a vector and en is a scalar.

Therefore yn species the direction of gradient and multiplying by this factor will not change the direction of the gradient. If yn is closed to zero then the denominator term let say norm of yn square here becomes very small and in that case this expression hn + beta times 1 / norm of yn square into en into yn may diverge because this term may blow up. A parameter gamma is included in the denominator to avoid large step size when norm of yn square becomes close to zero.

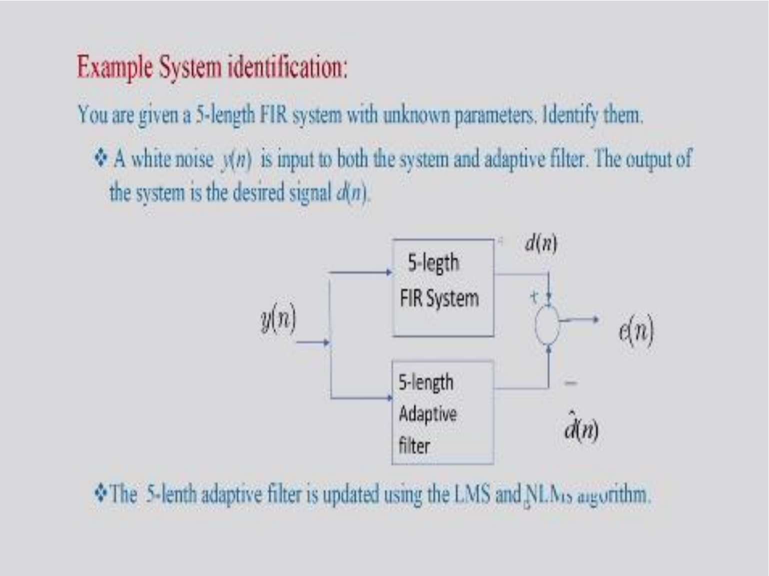
So therefore ultimately h of n + 1 will be = hn + beta times 1/gamma + norm of yn square into en into yn vector. So this is the NLMS update rule.

(Refer Slide Time: 14:47)



We can briefly state the NLMS algorithm so here given the input signal yn desired signal dn, filter length M, beta < 2 and a small constant gamma. Initialization hi0 = 0 for i = 0 to M – 1 all filter coefficients are initialized to 0. For n > 0 filter output will be d hat n = h transpose n times yn. This is the filter output. Estimation error en = dn – d hat n and filter coefficient is updated according to this relationship NLMS update rule. H of n + 1 = hn + beta times 1/gamma + norm of yn square into en into yn vector.

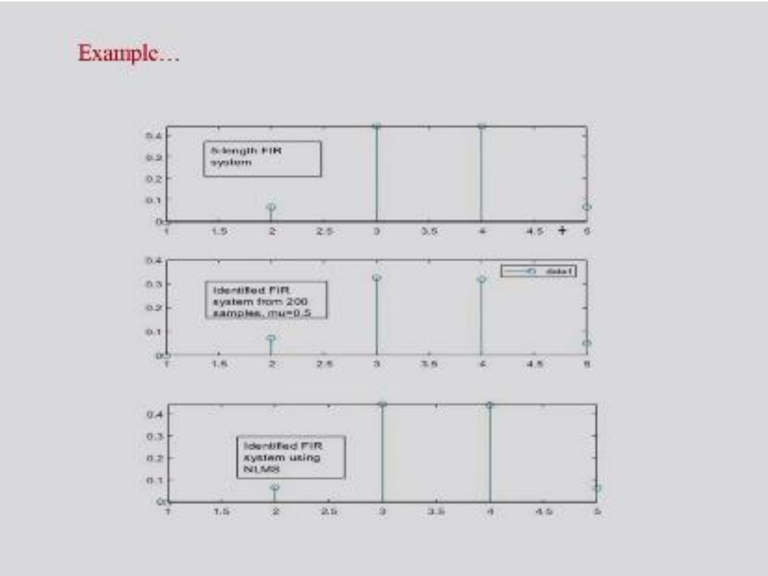
(Refer Slide Time: 16:00)



We will consider one example of system identification with the LMS and NLMS algorithm. System identification problem we introduced in the first class of adaptive filters. The problem is like this you are given a 5-length FIR system with unknown parameters identified them using adaptive filters. So here it is a 5-length FIR system we know that there are 5 coefficients in the FIR system model we have to identify those 5 parameters using a 5-length FIR adaptive filter.

So here yn is the output to both the system and the adaptive filter and usually yn should be a white noise and the desired signal dn is the output of this unknown system and the adaptive filter will estimate this desired signal and this error signal is feedback to the LMS or NLMS update algorithm. So that way the 5-length adaptive filter is updated using the LMS and NLMS algorithm.

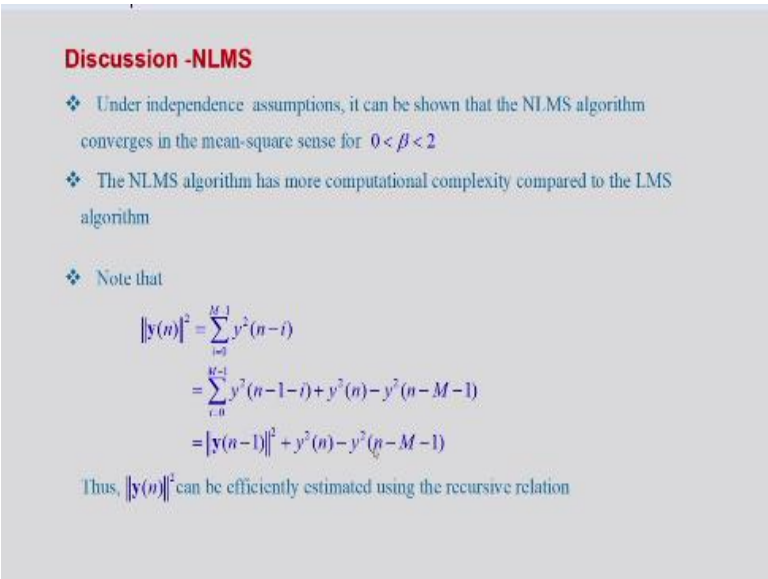
(Refer Slide Time: 17:30)



We show the simulation result so this is the original system 5-length FIR system these are the filter coefficient one coefficient is 0 here 0, this is the another coefficient third coefficient is like this, fourth coefficient is like this and fifth coefficient is like this and after identification by LMS algorithm with mu = 0.5 and data length is 200 we get this result. We see that this coefficient originally it was 0.44 like that here it is near 0.3.

So it is not exactly estimative. Similarly, this coefficient also it is not exactly estimating if we identify the FIR system using the NLMS algorithm we get the estimate results like this. Now this value and this value similarly this value and this value are almost matching. There is a small difference, but because it is 200 data only with that itself it is matching very well. Thus, we see that the NLMS algorithm converges faster than the LMS algorithm.

(Refer Slide Time: 19:00)

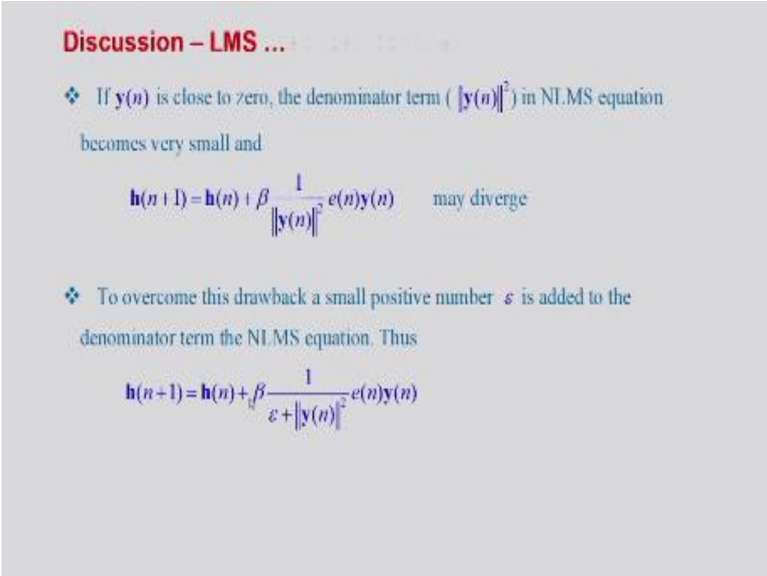


Let us quickly discuss the merits of the NLMS algorithm under the independent assumption it can be shown that the NLMS algorithm converges in the mean square sense provided beta lies between 0 and 2. The NLMS algorithm has more computational complexity compared to the LMS algorithm because of the computation of norm at every instance. Note that norm of yn square = summation y square n – i i going from 0 to M -1.

This summation we can rewrite as summation i going from 0 to M – 1 y square n – 1 – i. We are considering the earlier instances starting from n – 1 then we have to add y square n because here n – 1, i going from 0 so first term will be y square n – 1 so y square n we have to include. Similarly that last term here will be y square n – M -1 and therefore this term should be subtracted.

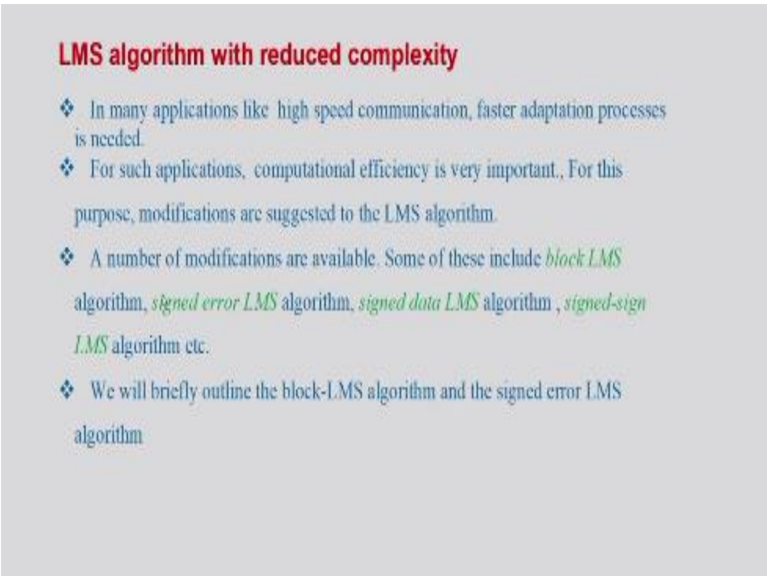
Therefore, this norm of yn square can be written as norm of yn – 1 vector square + y square + y square and – y square n – M – 1. So that way norm of yn square can be estimated recursively only this part we have to calculate this part and this part and then add to earlier estimate of norm square.

(Refer Slide Time: 20:58)



If yn is close to zero the denominator term norm of yn square in NLMS equation becomes very small and this term hn + 1 = hn + beta times 1/norm of yn square into en into yn vector may diverge. To overcome this drawback a small positive number epsilon is added in the denominator term of the NLMS update equation. Thus h of n + 1 = hn + beta times 1/epsilon + norm yn square into en into yn vector. This is the modified NLMS update rule.

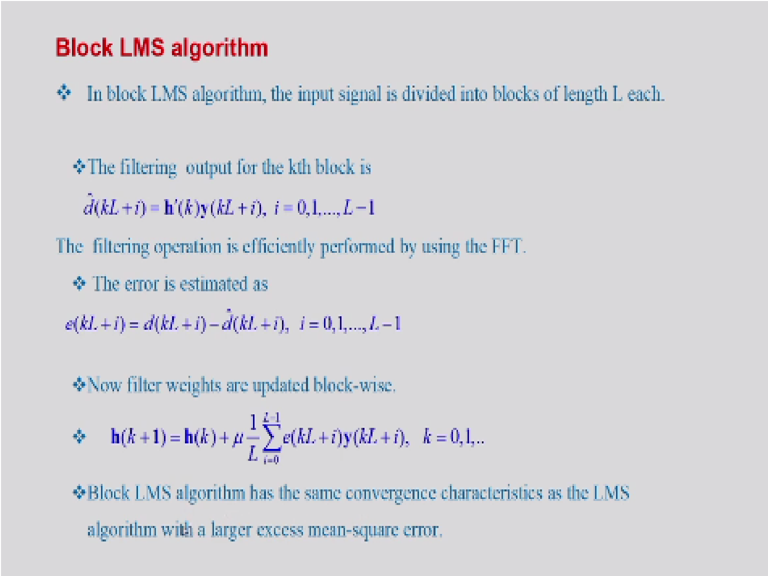
(Refer Slide Time: 21:54)



Now we will discuss some LMS based algorithms with reduced complexity. In many applications like high speed digital communication, faster adaptation is needed. For such applications computational efficiency is very important for this purpose modifications are suggested to the LMS algorithm. A number of modifications are available. Some of these include block LMS algorithm sign LMS algorithm, signed data LMS algorithm, sign-sign

LMS algorithm etcetera. We will briefly outline the block LMS algorithm and this signed error LMS algorithm.

(Refer Slide Time: 22:53)



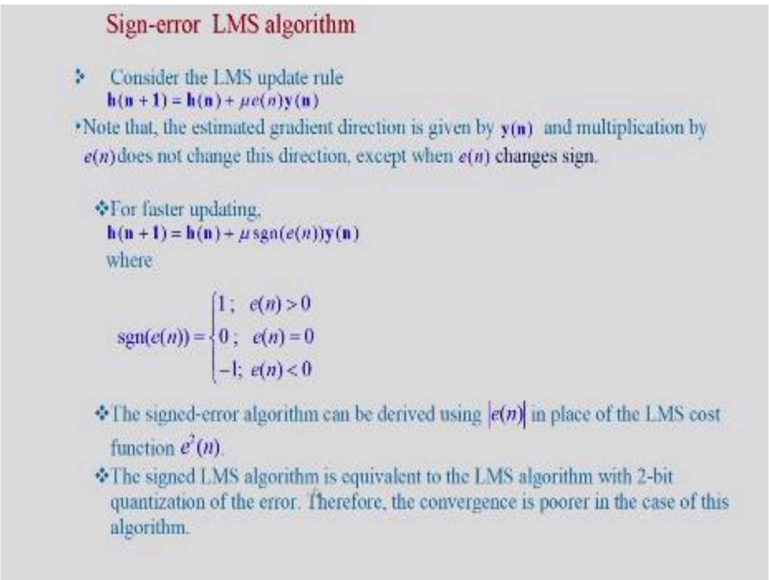
First we will see the block LMS algorithm. In block LMS algorithm the input signal is divided into blocks of length L each. So processing will be done in a block of L inputs. The filtering output for the kth block is given by this d hat kL ++ i = h transpose k into y kL + i this is the vector data vector i going from 0, 1 upto L – 1. So that way we will be considering suppose this is fixed now.

But for different data in this block that is kL + 0 kL + 1 etcetera those data vector we will be considering and filtering. Now this is the filtering operation since hk is fixed now because in one block only one filter coefficient vector will be used. Therefore the filtering operation is efficiently performed by using FFT. So you can use FFT to perform this. The error is estimated as e of kL + i that = dkL + i – d hat kL + i i going from 0, 1 upto L – 1 this is the filtering error.

Now the filter weights are updated block-wise according to this updating rule h of k + 1 vector that = hk vector + mu times 1/L into summation e of kL + i into ykL + i vector i going from 0 to L – 1 this is done for each data point in the block. So that way here we see that the error into the data vector these are summed up and then divided by L to get the estimate for the error into the data vector.

So this is the block LMS update in rule. Block LMS algorithm has the same convergence characteristics as the LMS algorithm with a larger excess mean square error It can be shown that each excess mean square error is larger.

(Refer Slide Time: 25:49)

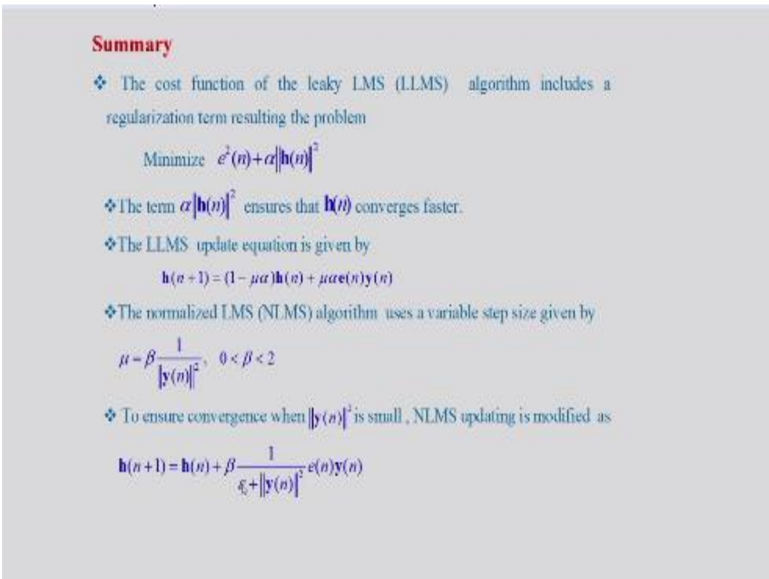


Next algorithm is sign-error LMS algorithm. Consider the LMS update rule h of n + 1 = hn + mu into en into yn vector so this is the LMS update rule h of n + 1 = hn + mu into en into yn. Note that the estimated gradient direction is given by yn this is the vector this vector gives the estimated gradient direction and multiplication by en does not change this direction except when en changes sign.

So if en is negative this direction will become negative so that way only that rule is performed by en. Therefore, for faster updating h of n + 1 = hn + mu into sign of en into yn. So where sign of en = 1 if en error is > 0 = 0 if en = 0 and – 1 if en < 0. This signed-error algorithm can be derived using mod of en in place of this LMS cost function e square n. If we use mod of en in place of e square n then by applying the (()) (27:28) we can derive this updating rule.

So that way it necessarily follows from this cost function mod of en cost function absolute error cost function. The signed LMS algorithm is equivalent to the LMS algorithm with 2-bit quantization. So we have 3 values of error that is 1, 0 and -1 so that way it is 2-bit quantization. Therefore, the convergence is poorer in the case of this algorithm because instead of using the entire error we are using it in 3 values only.

(Refer Slide Time: 28:16)

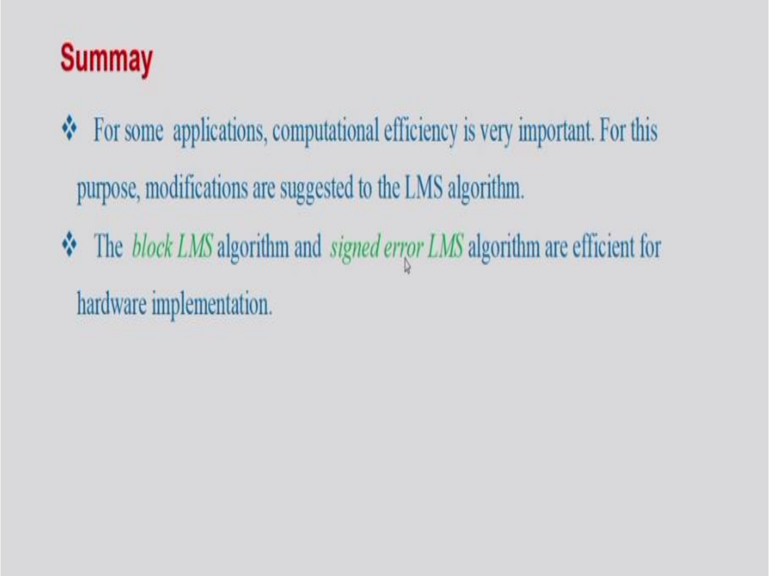


Let us summarize the lecture. The cost function of the leaky LMS, LLMS algorithm includes a regularization term resulting in the problem that is minimize e square n + alpha times norm of h n square where alpha is a constant. The term alpha times norm of hn square ensures that hn converges faster. So, because of this regularizing term. So this is the regularizing term it ensures that there is no mass deviation between two iterations.

The LLMS update equation is given by h of n + 1 = 1 – mu alpha times hn + mu into alpha into en into yn. So this is the update equation for LLMS algorithm. The normalized LMS NLMS algorithm uses a variable step size given this relationship mu = beta into 1/norm of yn square where beta lies between 0 and 2. To ensure convergence when norm of yn square is small because if yn become very small then this quantity will become small and this will diverge.

Therefore, to ensure convergence when norm of yn square is small NLMS updating is modified as h of n + 1 = hn + beta times 1/epsilon + norm of yn square into en into yn. So this epsilon is a small positive constant which ensures that there is no divergence.

(Refer Slide Time: 30:35)



For some applications computational efficiency is very important. For this purpose modifications are suggested to LMS algorithm (()) (30:45) modifications are block LMS algorithm and signed error LMS algorithm are efficient for hardware implementation. We knew that block LMS the filter coefficient are updated after a block of data. In signed-error LMS algorithm instead of en sign of en is used in the LMS updating equation.

So we have discussed about LMS algorithm. Next we will discuss another approach to adaptive filtering that is Recursive Least Square Algorithm. Thank you.