Introduction To Adaptive Signal Processing

Prof. Mrityunjoy Chakraborty

Department of Electronics and Electrical Communication Engineering

Indian Institute of Technology, Kharagpur

Lecture No # 37

Introduction to RLS Algorithm

So, we continue from what we were discussing in the last class. dN was this desired response vector xN hat x0 this column is called xN column then one cycle delayed if it is N it is N minus 1 if it is 1 it is 0. So, our notation was this we go up to z to the power minus N minus 1 xN. So, N minus 1 delay should be 0 that means, minus N plus 1 and W EN was what is E0 E1 dot dot dot dot EN at Nth clock what is the filter output? Filter output will be if you take this row xN xN minus 1 dot dot multiplied by this vector W0 xN W1 xN minus 1 and dot dot that is the filter output when you subtract it from dN that will be EN and so on and so forth for all the other indices. So, dN minus we have to do minimize the norm square of EN which is actually E transpose EN and we found out that this gives us to W Ls this squares as there is we minimize this with respect to the filter weights the one that minimizes that gives us to this and this tends to the optimal filter or inverse P as N becomes large that we have seen yesterday.

$$\underline{d}(n) = \begin{bmatrix} d(0) \\ d(1) \\ \vdots \\ d(n) \end{bmatrix}$$

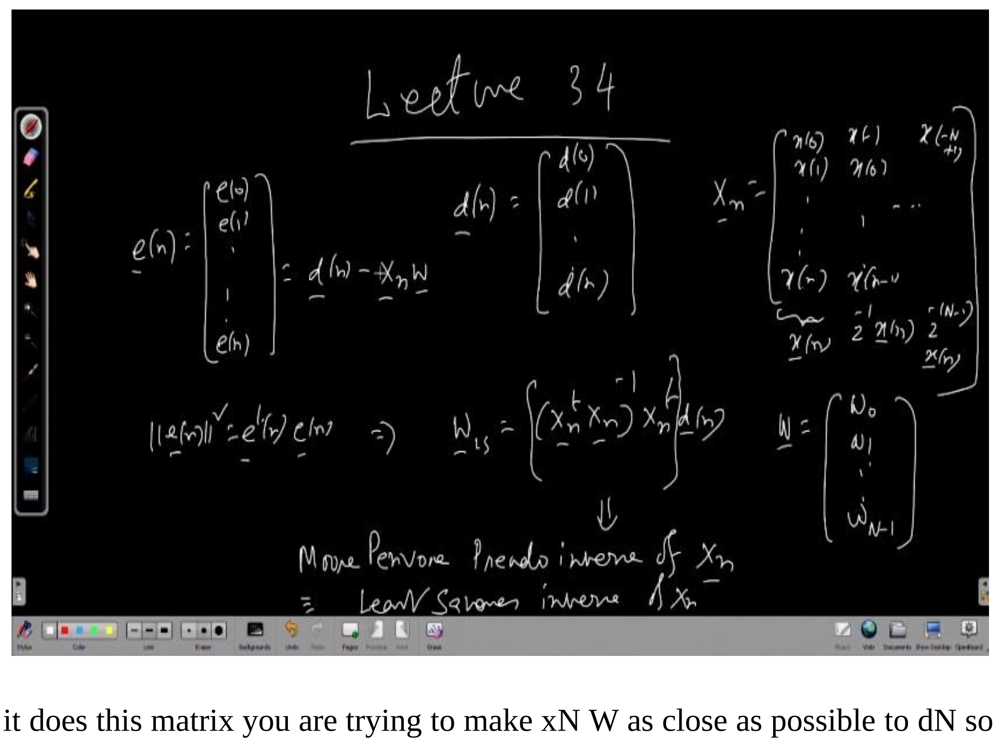
$$\underline{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{N-1} \end{bmatrix}$$

$$\underline{e}(n) = \begin{bmatrix} e(0) \\ e(1) \\ \vdots \\ e(n) \end{bmatrix} = \underline{d}(n) - \underline{X}_n\underline{w}$$

$$\left\|\underline{e}(n)\right\|^2 = \underline{e}^t(n)\underline{e}(n)$$

$$\underline{w}_{LS} = [\left(\underline{X}_n^t\underline{X}_n\right)^{-1}X_n^t]\underline{d}(n)$$

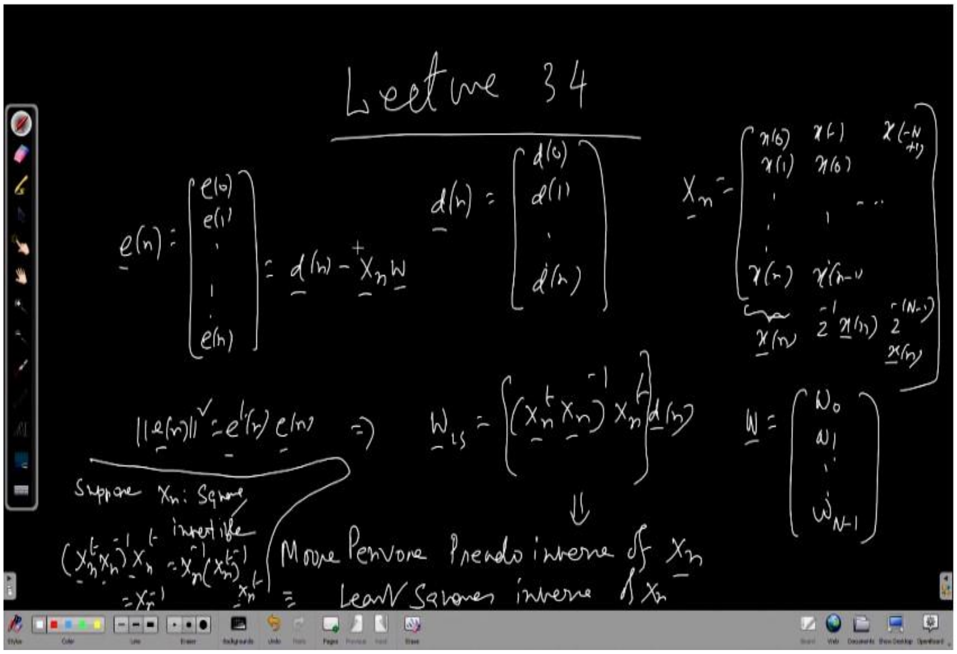
But there is no explicit use of R matrix there is input autocorrelation matrix or P vector there is a cross correlation vector between x of N vector and dN that is why you know we do not have to make any approximation like we did in the case of LMS where R and P were replaced by so called bad estimates and we paid the price that convergence was not absolute, but it was only in the mean. This is this thing this is called pseudo inverse of matrix xN actually called Moore-Penrose pseudo inverse also called least squares inverse of xN.



What it does this matrix you are trying to make xN W as close as possible to dN so that if you have the error vector error vector norm square is minimum then the solution is get is this. If an exact solution exists that is you find you really there is a W for which xN W equal to dN then when you minimize EN the norm square of P N then you can minimize up to 0 value because for that solution error is 0 if there is a solution and this one solution exists xN W equal to dN then the error is 0 which means the minimum norm square of EN achievable is 0 alright which means EN itself can be 0 and if N is 0 in that case dN you will get W which under normal circumstances you get by ordinary inverse that I am showing.

But suppose xN is square matrix also invertible that is full rank that is xN inverse exists if xN is invertible it is full rank. So, xN transpose also invertible we all know from basic matrix theory right. In that case this xN transpose and if it is invertible then xN transpose xN if we take inverse xN transpose now AB inverse is B inverse A inverse. So, it will be xN inverse xN transpose inverse xN transpose. So, these two cancels these two cancel you get xN inverse.

So, in that case if xN is square and invertible then obviously, dN minus xN W if we equate that to 0 there is a solution which gives exactly dN that is xN into W gives exactly dN because that solution is from this equation if xN is I am telling again if xN is square and full rank. So, xN W equal to dN there is a solution and that is xN inverse dN and for that the error is 0 right that is what this pseudo inverse you know reduces to from the formula you can see if xN is square and invertible xN is invertible xN transpose is invertible. So, xN transpose xN inverse is xN inverse xN transpose inverse and then xN transpose these two cancels you get identity. So, you get xN inverse. So, in that case this pseudo inverse is nothing, but ordinary inverse and solution will be just xN inverse dN whole error is 0 because xN is full rank.

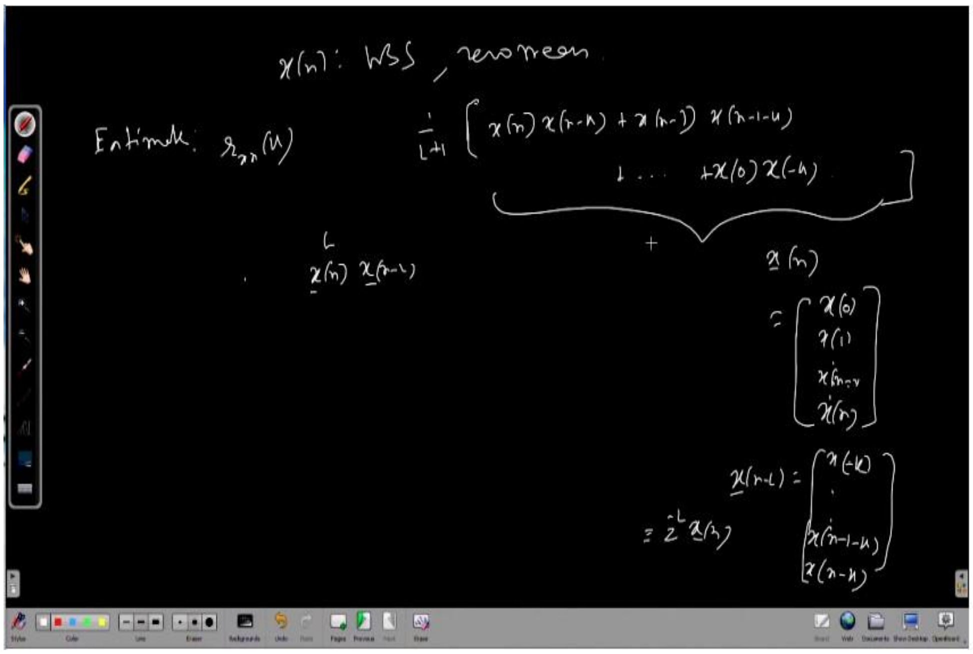


So, there is a unique solution xN W equal to dN that equation in which case error will be 0 minimum. So, norm square will be 0 there is a minimum norm square achievable and that is the optimal solution right. So, this boils down to ordinary inverse when this matrix is square and invertible, but not otherwise alright this is very famous thing alright. Next thing suppose xN is a random process you are measuring variance suppose you are measuring not even variance make it more general WSS 0 mean you have to estimate R x x k estimate. So, that means, you should have xN xN minus k plus xN minus 1 x N N minus k N minus 1 N minus 1 minus k and dot dot dot may be you take up to some L.

So, it is N minus L minus k. So, total L plus 1 N N minus 1 N minus top N minus L. So, total L plus 1 cases. So, you divide by this this what we have right. If I take this part this is actually you can see xN transpose because xN vector we all know it.

If I define xN like this here or maybe I go up to index 0 x0. So, this is the x0 x minus k alright. So, you have got x0 as before x1 dot dot dot xN xN minus L L cycle delayed. So, xN becomes xN minus L then N minus L minus 1 dot dot dot and this will be 0 minus L which is also denoted as z inverse L xN by your notation right. And this summation is nothing, but xN transpose x N minus L you just take a transpose of this x0 x minus L ok.

I have taken to be k. So, this is not just a means it is not L it is k just a change of notation it should be k xN becomes N minus k xN minus 1 becomes N minus 1 minus k like that alright. So, if you take this kind of thing xN transpose xN minus L then you get this x0 x minus k here x1 x minus k plus 1 index dot dot dot xN xN minus k xN xN minus k ok xN minus 1 here xN minus 1 N minus 1 minus k like that ok. So, this will be a good estimate of the correlation ok this will be a good estimate of the correlation. This product just you have to take the average by dividing by L plus 1 right.



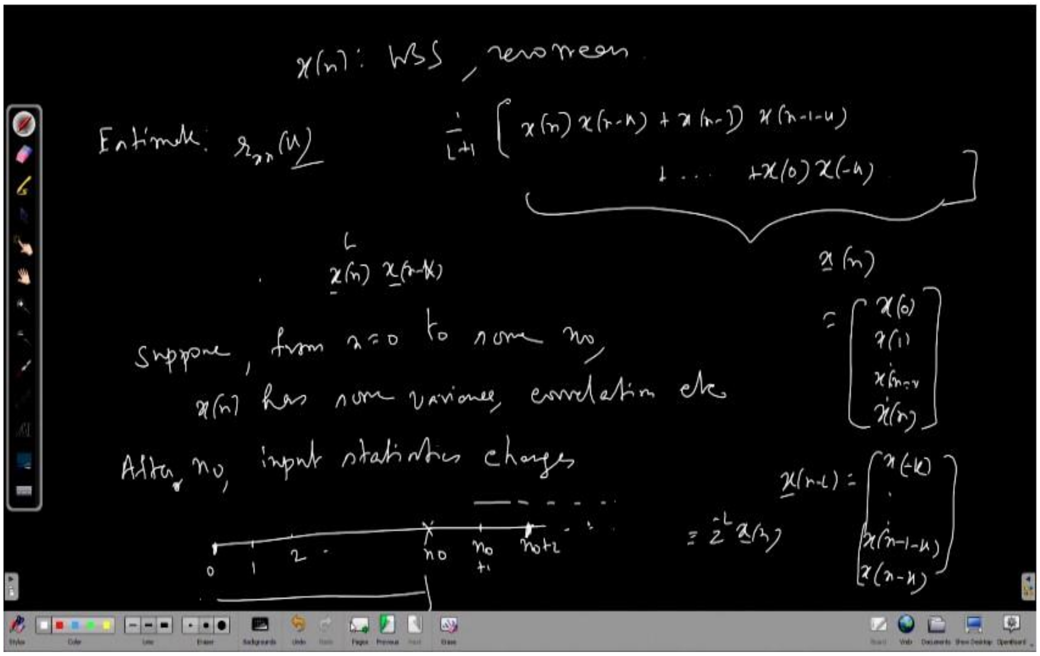
But now suppose so, correlation includes variance also if you just have k equal to 0 k equal to 0 then x transpose xN will be xN into xN xN square xN minus 1 into xN minus 1 that is xN minus 1 square and dot dot up to x0 square add all the square terms and then divide. So, that will be a good estimate of the variance. This is a special case of correlation right or xN is 0. Now suppose from N equal to 0 to some N0, xN has one xN has some statistics some variance correlation etcetera. But after this N0 the statistics changes.

So, now you are if you are calculating correlation ok you are going like this index 0 1 2 dot dot dot here is your N0 then N0 plus 1 N0 plus 2 dot dot dot dot. Suppose you are here up to this and suppose you are taking variance as a special case. So, we will be measuring what you will take x0 square x1 square x2 square up to xN0 square and then square xN0 plus 1 square xN0 plus 2 square. But remember during this part onwards statistics has changed is the variance here is no longer the same as the variance here. But you are mixing up you are mixing up the squared values from here and squared values from here adding and then averaging.

So, you are part of from this part from this, but as you move along you should have new variance is not it not contribution from old from past should come down. Similarly, more generally speaking from correlation suppose you are finding out correlation with the gap k. So, x0 x minus k x1 x minus k plus 1 dot dot dot up to x0 x minus N0 minus k that is fine, but now the moment you are crossing to N0 plus 1 and N0 minus N0 plus 1 minus k and so on and so forth. This part you are taking data from this part, but here the correlation has changed, but you are mixing up 2 domain this side when correlation Rx k k had some value. So, if you had just restricted yourself up to this up to N0 and take the average x x0 into x minus k x1 into x minus k plus 1 dot dot dot up to x N0 into x N0 minus k and then average you would have got very good estimate.

But you are now going further ahead you are bringing out data from the right of N0. So, from that that time after N0 correlation has changed. So, what you should then do you should forget this past and start from N0 plus 1 then N0 plus 2 you go further ahead take lot of samples just restrict your calculation of variance and correlation only there ok. Then there will be no mixing ok. Otherwise in that intermediate stage you know when part of this side and part of this side both are present both are taken into account then we will have wrong result of variance correlation etcetera.

To get rid of this we will introduce called a forgetting factor. The forgetting factor will be a scalar which will try to reduce the effects of past into the present computation past data into the present computation.



Let me explain by writing the expression suppose earlier you are minimizing this. There is a 1 by N plus 1 that I am not considering because that that do not depend on the filter weights this errors depend on filter weight that we have explained earlier. Earlier you are minimizing this sum of squared error ok, but what happened? So, what you have here e square 0 e square 1 may be e square some intermediate point N0 e square N0 plus 1 dot dot dot e square N.

Now, suppose up to N0 input statistics during change. So, if you are restricted to calculation only up to this e square 0 e square N up to e square N dot add them divide by the number of terms you will get a good estimate of the previous variance or if you start from e square N0 plus 1 e square N0 plus 2 and go up to some e square N and suppose there are many terms here just add them and divide by the number of terms you will get again a good estimate of the new variance, but you are not doing that you are actually going one at a time. So, after N0 you included e square N0 plus 1 ok. And this whole sum then we will have one contribution from the current statistics and all other contribution from the previous statistics and you are averaging. So, that will be neither the variance of previous case not the variance of the current case and this will continue for some time till you have got you know very long contribution from the current N0 plus 1 onwards you go to N which is very large.

$$\sum_{i=0}^{n} e^2(i) = e^2(0) + e^2(1) + \cdots + e^2(n_0) + e^2(n_0 + 1) + \cdots + e^2(n)$$

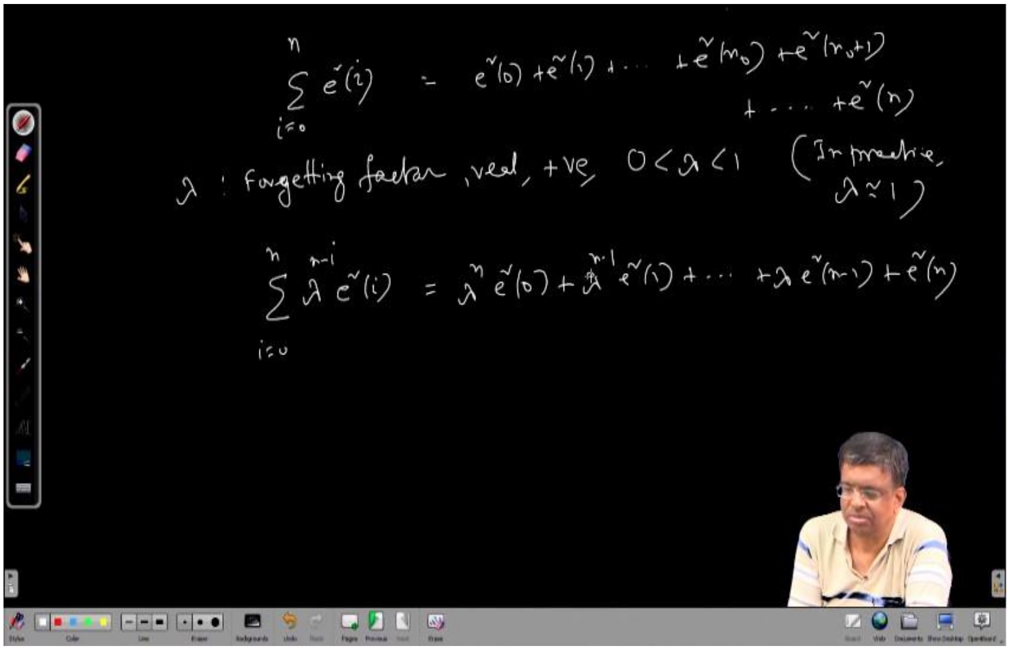
So, contribution from this side right hand side of N0 is much larger than the contribution here because here the length is up to N0 plus 1 0 1 up to N0. So, N0 plus 1 here N0 plus 1 to N ok. So, if this side is becoming very long because N is very large ok, then overall sum will be dominated by this side which is fine the past is almost forgotten, but till that happens there is a intermediate range transition range when there will be error when part of some contribution from past and some contribution from present they will get mixed up. To avoid that we introduce a forgetting factor real positive lambda this between 1 to 0 in practice lambda is close to 1 like 0.

$$0 < \lambda < 1$$

99 0.98 and what we do is this we modify the summation lambda to the power N minus i this summation we considered. What is the summation let us write down i equal to 0 means lambda to the power N square 0, then lambda to the power N minus 1 square 1 plus dot dot dot lambda to the power this is N0 for it N0 I am going to the last point it was 0 N it was 1 N minus 1 if it is previous term is N square N minus 1. So, it will be lambda times N square N minus 1 and then just N square N ok. What it is N minus 1 it is lambda to the power 1 lambda to the power N minus i i. So, if i is N minus 1 N minus within bracket N minus 1 is lambda lambda 2.

$$\sum_{i=0}^{n} \lambda^{n-i} e^2(i) = \lambda^n e^2(0) + \lambda^{n-1} e^2(1) + \cdots + \lambda e^2(n-1) + e^2(n)$$

So, lambda square N minus 1 and lastly when i equal to N lambda to the power 0 is 1. So, square N these are summation. Now in this summation because lambda is less than 1 even if it is 0.99 as we go for to this side to the left before lambda, I have got lambda square then lambda cube lambda square lambda to the power 4. As you go into the past power of lambda is increasing and lambda is less than 1 means progressively it is becoming less and less because lambda is less than 1.



So, contribution for past is becoming less and less it is getting diminished because they are increasingly multiplied by higher and higher power of lambda. So, their contribution is diminished whereas, current term has full contribution previous term has little less because of lambda then previous term will be lambda square so, slightly less. So, overall contribution of the errors in this summation will be coming mostly from the current E N and some of the past E N's E N minus 1, E N minus 2, maybe E N minus 3 like that. I mean terms which are very remote like E 0, E 1, E 2 their contribution will be very less because their corresponding power of lambda will be large and lambda is less than 1. So, how much lambda to the power 10 that is much less almost 0 that is why so, this will this arrangement will help us forget the contribution from the past remote past and concentrate on and allow us to concentrate on the recent ok.

This means earlier we are considering earlier x transpose y where x and y just two vectors I am taking x transpose y where considering that was called the dot product or inner product between x and y x 0 y 0 x 1 y 1 dot dot x n y n summation, but now we will have lambda x n y n ok. This was x 0 y 0. Now, it will be x n y n current term as it is then just one lambda dot dot dot this is lambda to the power n x 0 y 0. So, this is nothing, but this when x and y both are same and both are equal to E N vector. So, E N transpose E N that was the thing earlier, but now there was that summation, but now with lambda summation has changed.

So, this will be actually not x transpose y, but x transpose y is this now a lambda has come. So, this is equivalent we can see x transpose then one diagonal matrix I will tell you what the diagonal matrix is y where this lambda capital lambda lambda to the power 1 and lambda to the power 0 is 1 this side 0 this side 0. This lambda n y means lambda to the power n y 1 lambda to the power lambda to the power n y 0 lambda to the power n minus 1 y 1 dot dot dot ok. Then x transpose. So, you have x 0 lambda to the power n y 0 that is what you have here plus x 1 lambda to the power n minus 1 y 1 ok.

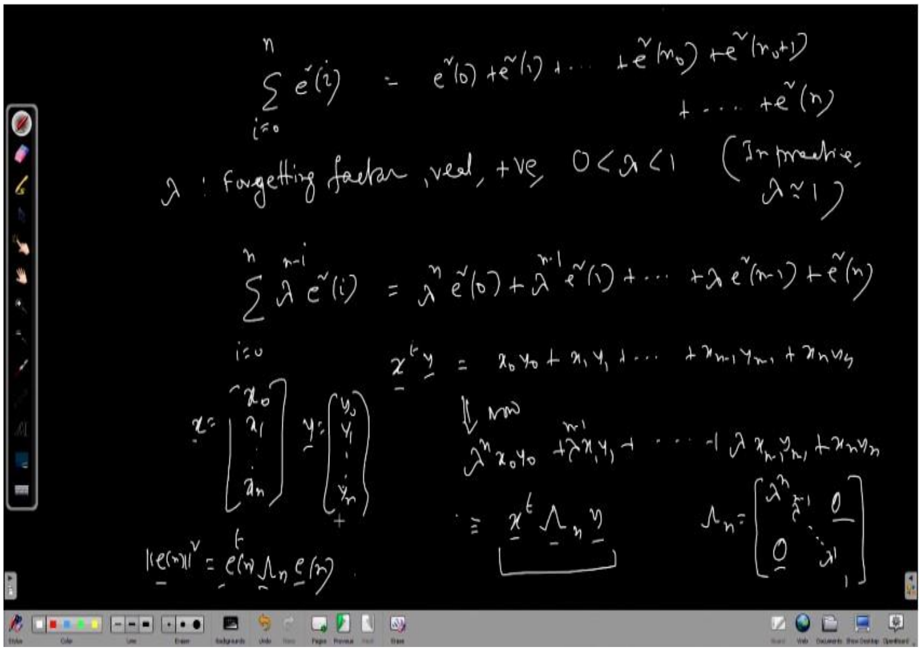
$$\underline{x} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \quad \underline{y} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$$\underline{x}^t \underline{y} = x_0 y_0 + x_1 y_1 + \cdots + x_{n-1} y_{n-1} + x_n y_n$$

$$\Downarrow$$

$$= \lambda^n x_0 y_0 + \lambda^{n-1} x_1 y_1 + \cdots + \lambda x_{n-1} y_{n-1} + x_n y_n$$

So, x 1 lambda to the power n minus 1 y 1 and so on so forth this is the expression. In the case of variance both x and y are E N vector. So, it will be so now, norm square will be no longer E N transpose E N like x transpose x or x transpose y no longer E N transpose E N, but it will be E N transpose then this lambda N factor alright this is what this is what we have to minimize.

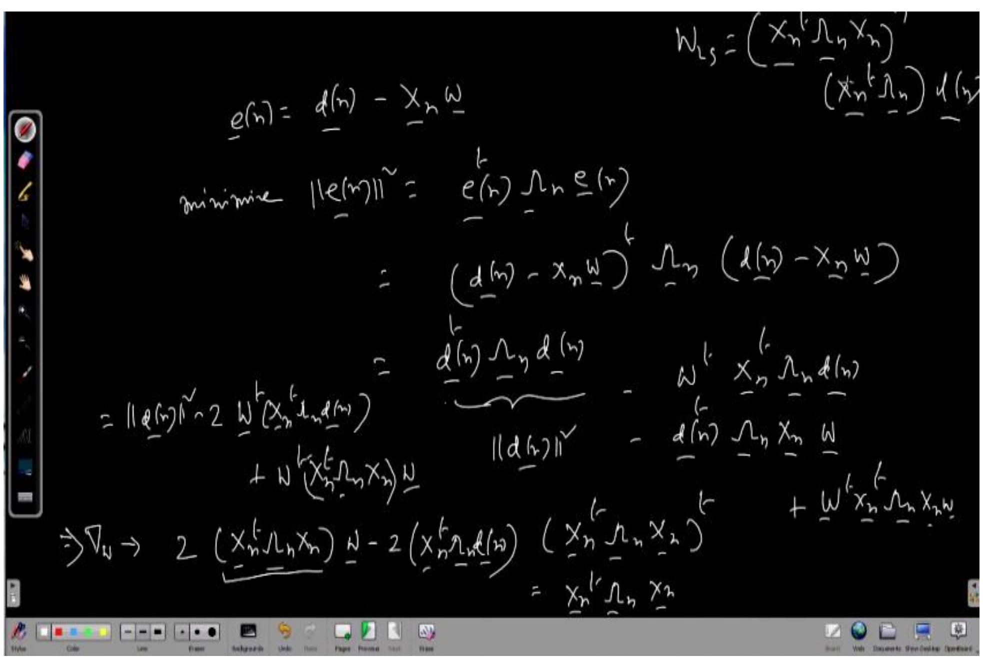


That means, if you find out as before E N as E N minus x N w. We have to minimize this norm square which is E N minus E N minus ok.

Then we break it E N transpose E N transpose which is the new norm square on in the presence of lambdas then x N w transpose this this. So, that means, w transpose x N transpose minus other cross term D N transpose a row vector this plus, but before that these two as I said earlier these two these two terms both are scalars these two terms are same. Because one is both are scalar and one is the transpose of the other if you take transpose of this w transpose comes first and transpose of this transpose of this means x N transpose then this transpose, but this is a diagonal matrix transpose is itself D N D N transpose transpose which is D N. So, both are same instead of writing two I can write one and then twice that that I will write x N w plus the last term x N w transpose means w transpose x N in these x N w which is now I am going to write twice w transpose this is a vector column vector x N transpose lambda N matrix into matrix matrix matrix into column vector D N is a column vector overall column vector u can call it u. So, w transpose u if you differentiate into with respect to all the components of w and put it in a stack you will get u that means, this only.

So, first writing and then I will differentiate it. Remember this matrix also Hermitian because this is a diagonal matrix x N transpose right and we use the transpose here x N w transpose. So, w transpose x N transpose these Hermitian because if you take its transpose you get x N transpose here then transpose of this means transpose of this diagonal matrix which is itself transpose of this transposed x N which is x N. So, same as this this is a Hermitian matrix. So, if you have a Hermitian matrix R w transpose R w which is a scalar when differentiate it with respect to w all the components of w and put it in a stack it will be twice R w we have seen that.

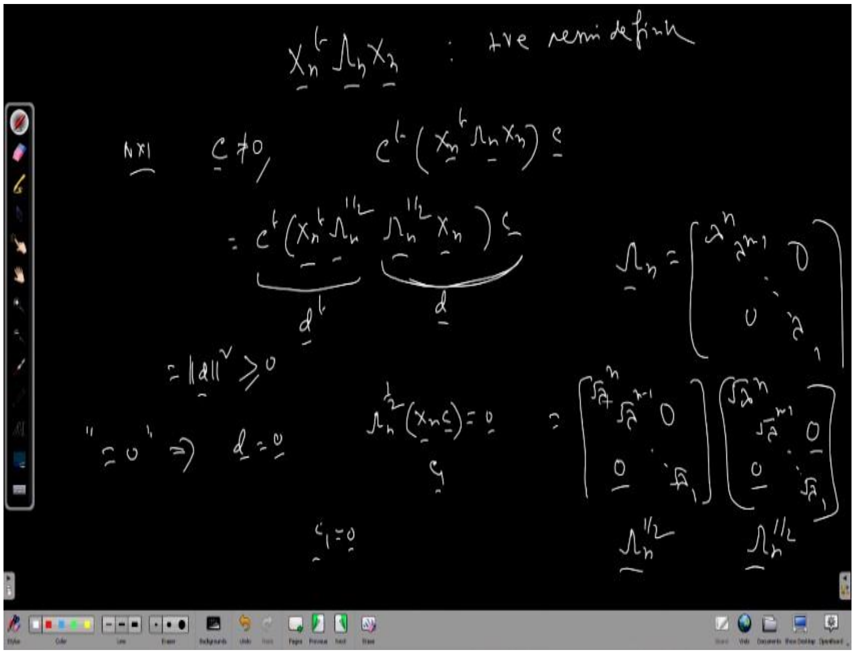
So, now that means, after differentiation if I apply del w on this I will get twice of and here minus twice this thing you equate this to 0 and if this is invertible assume this to be invertible as before. So, your w L S will be inverse 2 2 cancels. So, only this lambda N matrix has come if you take that out it is same as what we had earlier and I am assuming this matrix to be invertible because of this region that even if this lambda N has come this will be positive semi definite always we will assume it to be positive definite. Firstly, it is Hermitian we have already seen you can take transpose we have already seen it is positive semi definite because if you take a C vector non 0 C vector non 0.



N cross 1. So, this C transpose this matrix C what happens to this now this lambda N which is lambda to the power N lambda to the power N minus 1 lambda 1 and 0 you can takes positive square root of lambda. So, if I construct this two matrices I am just denoting by this and this is square root lambda to the power N square root lambda to the power N minus 1 square root lambda 1 0 0 and again same matrix. You can easily see if you multiply these two you get this all right. So, here C transpose X N transpose this square root and there again diagonal means they are Hermitian symmetric and.

So, I take this part call it D. So, if it is D you can easily see this is D transpose C transpose X N transpose and transpose of this is itself. So, this is norm D square. So, always greater than equal to 0 that is your positive semi definite, but when will be 0 I have to rule out that case. If it is 0 that means, D must be 0 norm square ordinary norm square D 0 square plus D 1 square plus D 2 square dot dot if it is equal to 0 everybody is non negative and adding and adding. So, every comp point has to be 0 then every contribution 0 overall 0 ok.

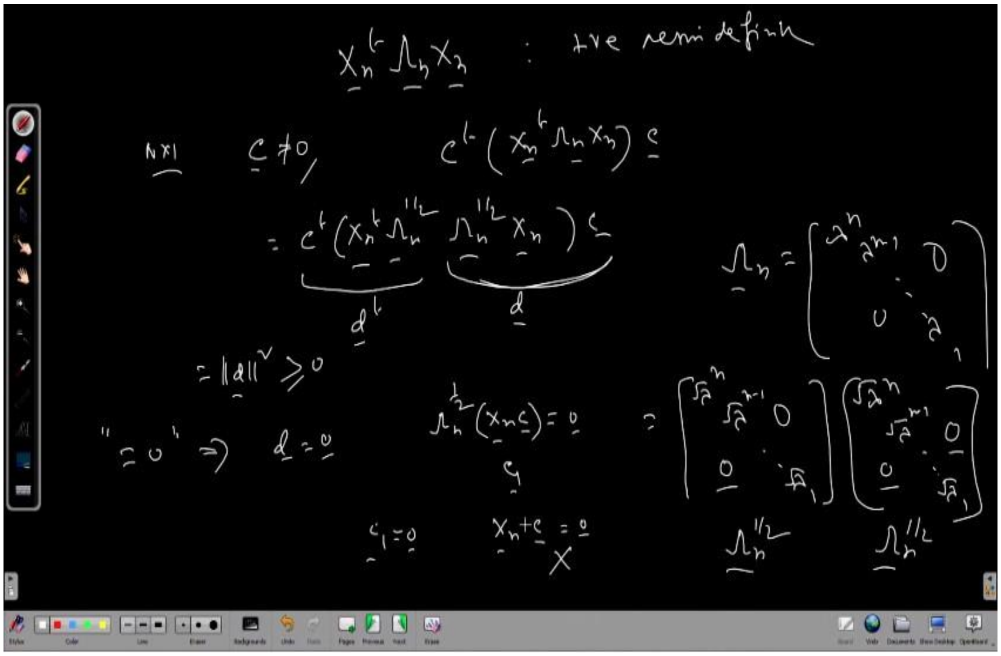
If D is 0 then that means, 0 you call it C 1 so that means, this matrix time C 1 0, but these are all positive elements. So, this matrix time C 1 means what this square root lambda to the power N into top first guy of C 1 then square root lambda to the power N minus 1 second guy of C 1 dot dot dot and if that is that is equal to 0 and since lambda is positive square root lambda is positive then every element of C 1 has to be 0 ok. That is C 1 must be 0 vector understood if you take this matrix lambda N to the power half and take a C 1 vector what you get this square root lambda into the to the power N times first guy square root lambda to the power N minus 1 second guy dot dot dot and you are equating that to 0.



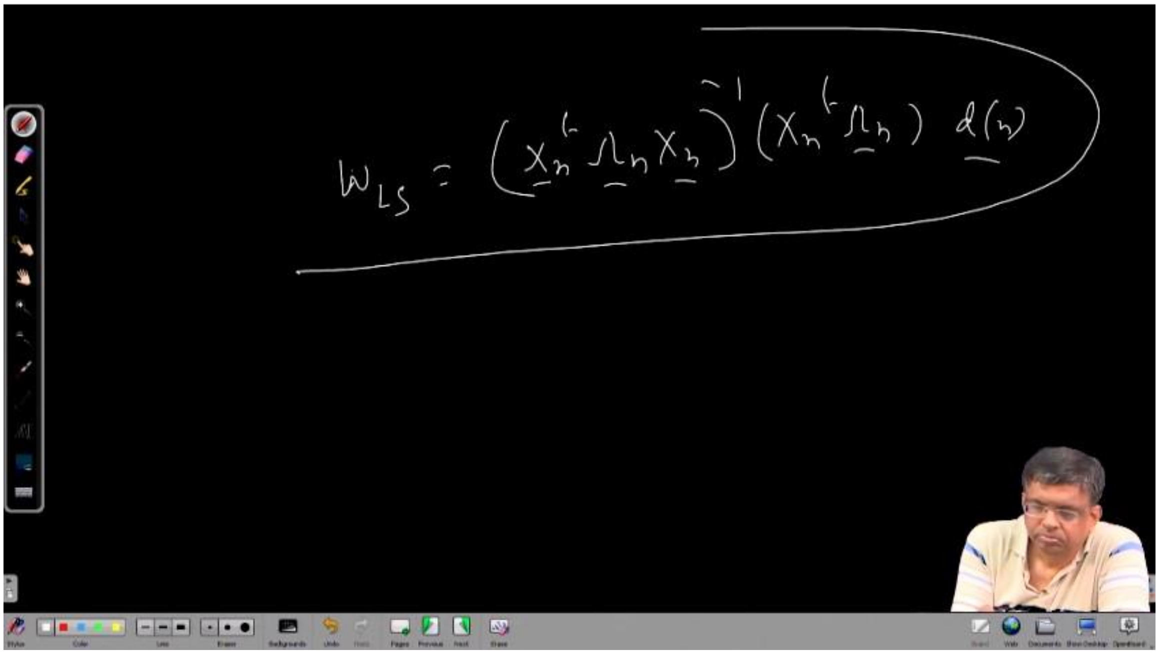
Therefore, elements of C 1 should be 0 because these are not 0s lambda is non 0 square root lambda is non 0 ok. So, C 1 0 vector and C 1 0 means x N C 0 which means columns of x N since C is a non 0 vector you can linearly combine the columns of x N by the elements of C and equate to 0.

So, one column you can write as a linear combination of the other columns, columns are linearly independent that is there is a linear relation involving the columns which we assume not to be not the case because data is coming purely randomly there is no linear relation in the background involving them or connecting them. So, that will not happen. So, this is not allowed ok because data in x N they are coming randomly there is no linear relation in those data vectors that does not happen in practice. Therefore, x N C you cannot find a non 0 vector C. So, that if you combine the columns of x N by the elements of C you get a 0 vector meaning at least one vector you can keep on the left hand side which has a non 0 coefficient from here others you take to the right hand side divide both side by that non 0 coefficient.

So, one column you can write as a linear combination of the other columns as though there is a linear relation between columns one column there is a formula by which you can get one column as a linear combination of other columns, but that does not happen in practice. In practice all the data come randomly there is no linear relation binding them. So, this will not happen. So, for non 0 C for non 0 C you cannot find a non 0 C. So, that x N C is 0 because columns of x N are linearly dependent ok.



Of course, this for the same rigid I have said earlier N plus 1 should be greater than equal to capital N number of rows should be greater than equal to capital N number of columns that I have already explained the same logic holds here all right. So, this is the more general formula of W L S with a forgetting factor. We will start from here in the next class. Thank you very much.


$w_{LS} = (\underline{X}_n^H \underline{\Lambda}_n \underline{X}_n)^{-1} (X_n^H \underline{\Lambda}_n) \underline{d}(n)$