Introduction To Adaptive Signal Processing

Prof. Mrityunjoy Chakraborty

Department of Electronics and Electrical Communication Engineering

Indian Institute of Technology, Kharagpur

Lecture No # 38

Formulation of the RLS Algorithm

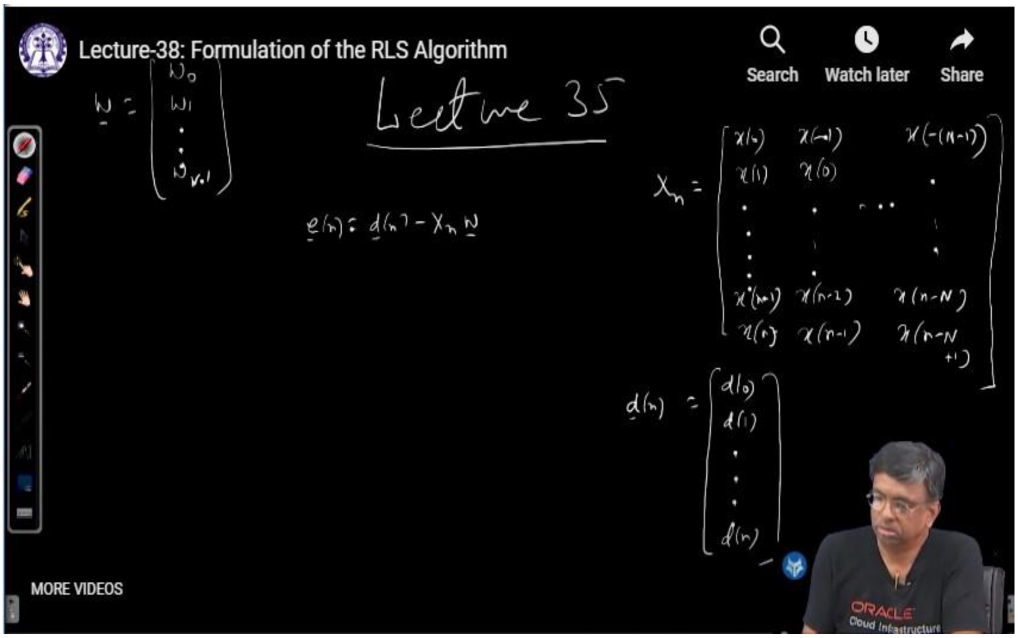
We are considering recursive least squares transversal filter. You remember we had a data matrix at nth clock built using data obtained from n equal to 0 to I mean time equal to 0 to time equal to n. I do us like this x0 x1 dot dot dot dot xn minus 1 n. So, in one cycle delayed version xn minus 2 xn minus 1 dot dot dot dot. Lastly x minus n minus 1 here it is n minus capital N plus 1 xn minus capital N plus 1 minus 1. So, this dot dot dot dot and there was dN vector.

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

So, these are all old stuff that is why I am not explaining them again. You can go back to previous lecture and see how they came and all that. Then the error vector at nth clock was dN minus xn filter weight vector. W was just a general filter weight vector w0 w1 dot dot dot wN minus 1.

$$\underline{e}(n) = \underline{d}(n) - X_n \underline{w}$$

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



You can see one thing xn w. So, if you multiply this by a w you start with the last row this times w that is a filter weight filter output vector at nth clock because xn into w0 xn minus 1 into w1 dot dot dot this into wN minus 1 and summation. So, that we know is the filter output at nth clock. Then next row times the same w. So, w0 times xn minus 1 w1 times xn minus 2 dot dot dot the summation there is a filter output at n minus 1th clock so on and so forth.

So, this is actually giving a vector of filter weight outputs filter outputs at nth clock n minus 1 at n minus 2 is up to 0th and when that is subtracted from this dN then dN minus filter output at nth clock so that will be output error at nth clock then output error at n minus 1th clock and all that there is overall error vector. Our purpose is to minimize the norm square that is if we take the norm square of this vector which I have forgetting factor of course, that is if you take it was like you know I mean current term is e square N next was some lambda times e square N minus 1 is the current term then lambda square e square N minus 2 and dot dot dot lambda N 0 where lambda real and lambda greater than 1 less than 1 greater than 0, but typically very close to 1 maybe 0.99 something like that.

$$||e(n)||^2 = e^2(n) + \lambda e^2(n-1) + \lambda^2 e^2(n-2) + \cdots + \lambda^n e^2(0)$$

So, what it does it just recap when lambda is lambda or lambda square lambda cube and all that its value is not that less. So, in this summation the current contribution e square N or previous cycle contribution e square N minus 1 and all they dominate, but contribution from remote past they diminish their effect is much less that helps us in concentrating on the norm square of the error vector on the current history not on past history because I mean system characteristics and all that you know can change from time to time.

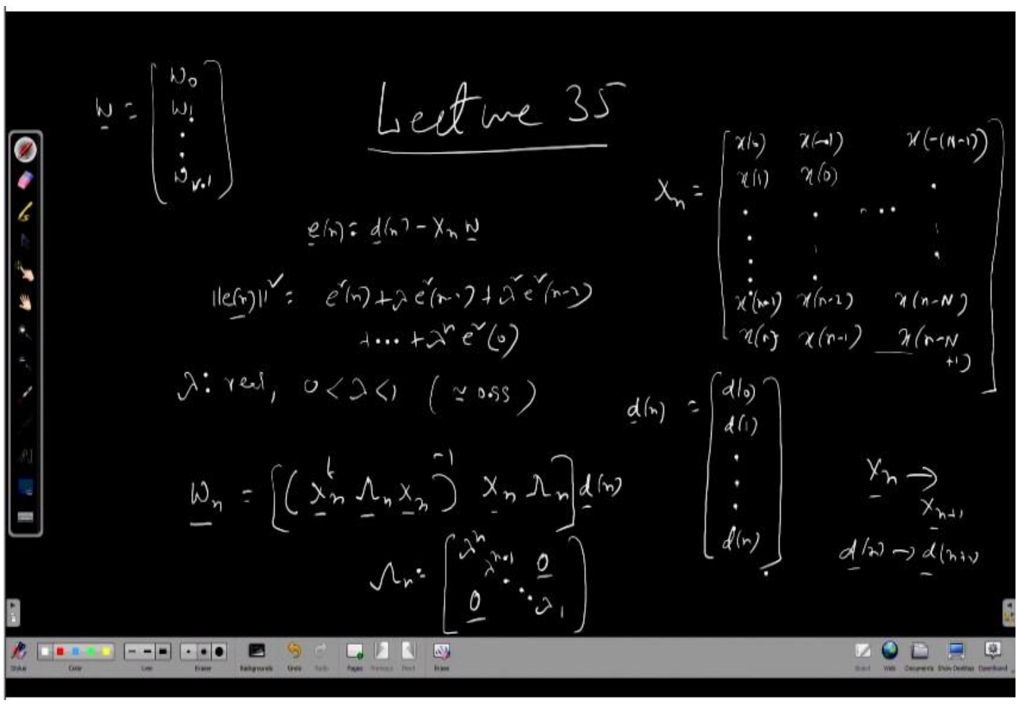
So, we should focus on this norm norm square from the current history point of view all those were explained last time ok. And by minimizing this we obtain the recursive least squares estimate of the filter weights WN at Nth clock as this, this is XN this is diagonal matrix I will write about it though you all know this XN transpose inverse ok, where the diagonal matrix these of course, assume that XN matrix there is this matrix the columns are linearly independent this is a full rank which means number of rows must be at least equal to number of column or more.

$$w_n = \left[\left(\underline{X}_n^t \underline{\Lambda}_n \underline{X}_n\right)^{-1} \underline{X}_n \Lambda_n\right] \underline{d}(n)$$

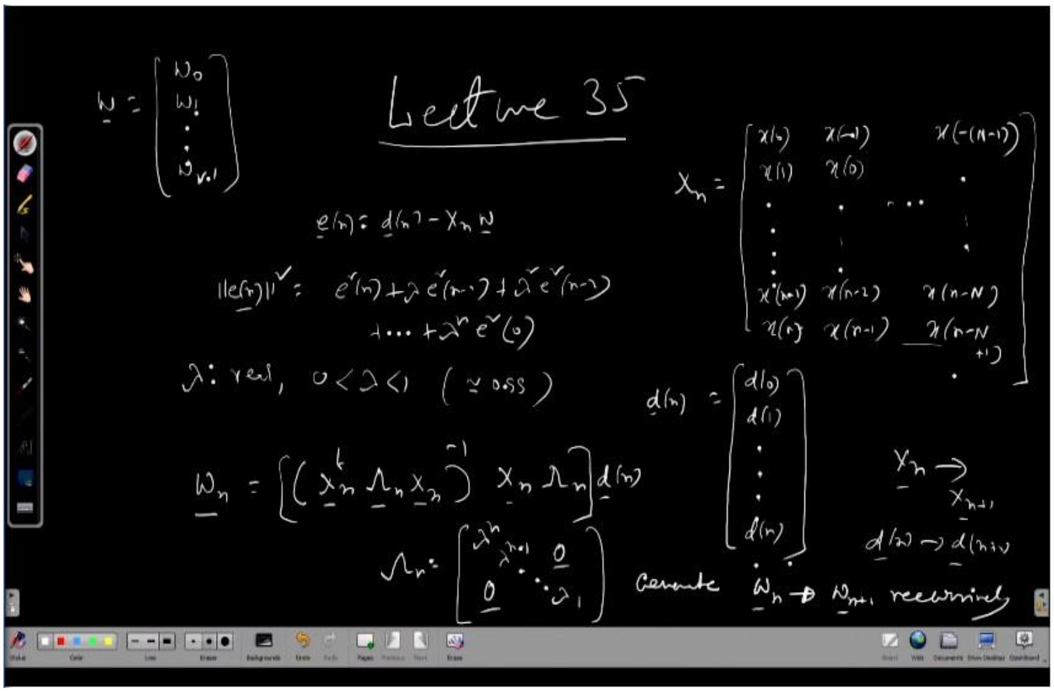
So, that means N how many columns we have N number of columns 0 1 like that ok. So, small n should be at least equal to N or more otherwise it will be rank deficient. So, that is what we are considering we are assuming that this XN is full rank which is not possible if small n is less than N ok.

So, even when small n is greater than N it may or may not be possible, but that is what we are assuming it is invariable we will talk about that more later this is what we have derived earlier. Now, point is that Nth clock you got all this data you carry out all this business and find out this filter coefficient vector ok, which is the best estimate recursive least squares estimate use it for filtering ok, the input data vector input data ok. Next time at N plus 1th clock from XN we go to XN plus 1 one more row comes in where a new data XN plus 1 that comes then of course, XN N minus 1 they are all already obtained past data, but at least one new data comes XN plus 1 using that I mean I construct a new row. So, the matrix size increases number of rows goes up by 1. Similarly, D N another new D N plus 1 comes D N also goes up ok D N also goes up by 1.

So, my previous estimate WN will not work I have to replace XN by XN plus 1 this should be delta lambda N plus 1 ok that is 1 lambda dot dot dot lambda to the power N minus 1 lambda N another term lambda to the power N plus 1 and again XN plus 1 and all that you know D N plus 1. So, shall we do like this that at every clock we carry out this whole computation find out WN and the filter next time XN matrix changes to XN plus 1 D N vector changes to D N plus 1.

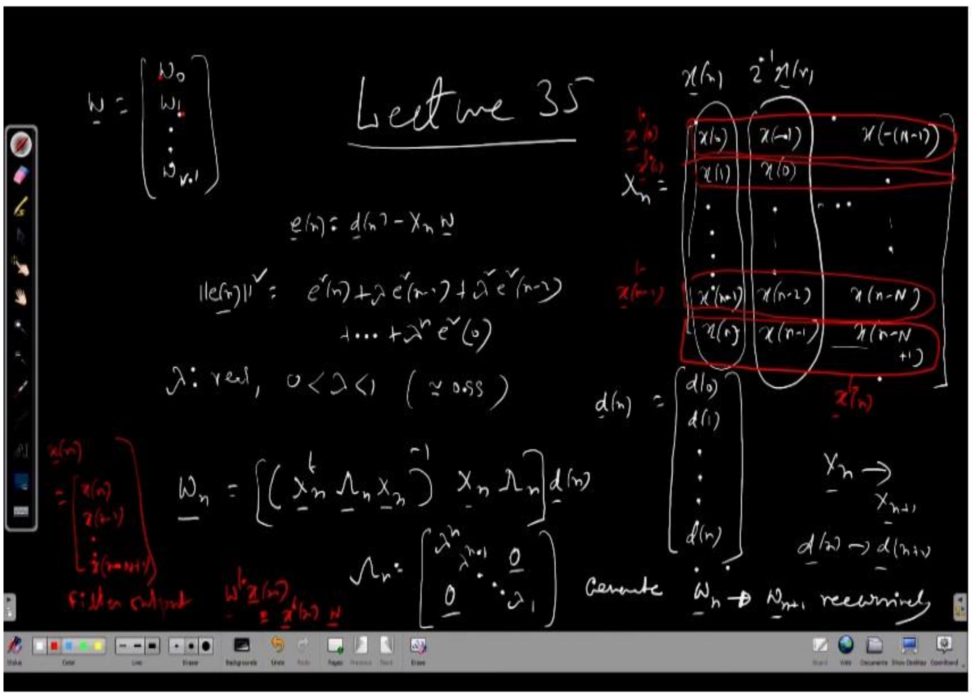


So, let us again compute the same thing replacing N by N plus 1 get WN plus 1 again do filtering that we cannot do that will be too much you know we should rather recursively obtain generate WN to WN plus 1 recursively that is instead of carrying out the whole computation this inverse formula pseudo inverse formula just using some simple update relation like we had in LMS where from current WN we go to next WN plus 1 by some extra update term can we generate from WN WN plus 1 similarly recursively without going through this base of this computation that is what we will be approaching because lot of things are common between WN and WN plus 1 because matrix XN getting changed to N plus 1 just one extra row is coming, but previous part remain same. So, that is common with the previous case and current case similarly D N vector one extra term D N plus 1 comes at the bottom, but rest of the terms are common ok. Therefore, there may be possible it may be possible to find the relation between WN and WN plus 1 because just one extra row has come up here in XN one extra element has come up in D N ok other things are common.



So, why not try and find out some relation between WN and WN plus 1 which will be of course, recursive relation from WN you should be able to generate WN plus 1 that is what the whole goal is ok to generate WN plus 1 from WN recursively all right. To do this I mean today I need to do certain things first then we can go into the derivation derivation is very interesting I will just introduce certain change in notation I mean earlier I called this vector this to be vector XN then this was Z inverse XN and so on and so forth. Now, that will be slightly changed now I will call this to be like in the LMS case this to be XN transpose that is now my XN definition is the data vector at the filter input at Nth clock that is current data previous data dot dot dot. So, filter output is W transpose XN as we know which is equivalent to XN transpose W all right this is the notation we are familiar with filter input vector XN XN minus 1 dot dot XN minus capital N plus 1. So, W transpose XN means W0 XN W1 XN minus 1 dot dot dot WN minus 1 XN minus capital N plus 1 their summation which is the filter output.

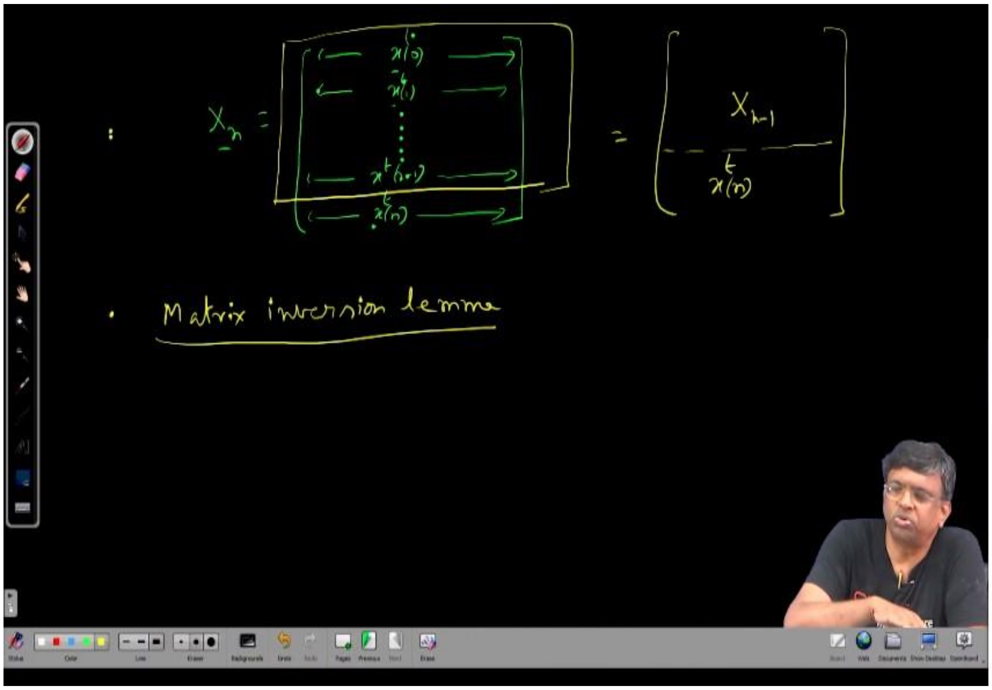
So, if it is XN you see it is XN transpose. So, this will be XN minus 1 transpose right so on and so forth this will be X0 vector transpose this will be X1 vector transpose all right.



So, then I rewrite XN like this this is just a change of notation conceptually there is nothing new this was just using that notation. So, if I write it down you can see one thing if I divide this this upper half is nothing but XN minus 1 instead of N I have N minus 1 I will go from X0 transpose X1 transpose dot dot up to XN minus 1 transpose. So, this is XN minus 1 and this bottom I have got 1 X transpose N that has come up.

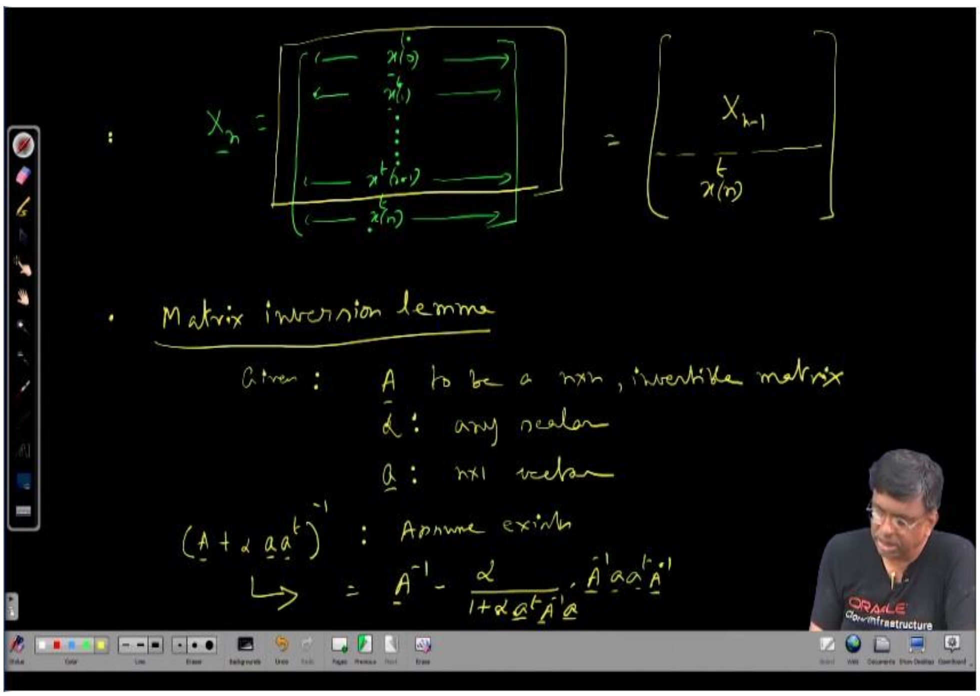
$$\underline{X}_n = \begin{bmatrix} X_{n-1} \\ x^t(n) \end{bmatrix}$$

So, this is a relation we will be using. So, we will be using this notation this is one thing. Another thing is very famous matrix inversion lemma. This is a very general layout which we will not discuss here a special case which is applicable to us that all will be discussed that will make things simpler. And this lemma is very useful in you know this adaptive signal processing statistical signal processing and all that.



So, this is general result I am quoting I will be using it in that recursively square derivation but this is a separate result. So, given A to be maybe N cross N the square matrix invertible alpha any scalar real or complex does not matter and A some N cross 1 vector A is invertible then if I have a form like this A matrix it was already invertible but an extra component another matrix is getting added with that then what happens to the inverse assuming that inverse exists that additional component is A column vector into its transpose row vector. So, column into row it is an additional matrix I introduce the alpha scalar here, alpha could be 1 also but we want to be more general so alpha is this. So, let us keep a provision of alpha assume invertible that is assume exists that is not only A is invertible if along with A we add this alpha A, A transpose this total matrix also if we assume invertible then what is this inverse. Then claim is this is equal to A inverse minus scalar alpha divided by 1 plus alpha I get scalar A transpose which is a row vector now capital A inverse A.

So, capital A is a matrix A inverse is a matrix small a is a column vector. So, A inverse small a is a column vector this is a row vector row into column is a scalar. So, this whole thing is a scalar this times a matrix of course, because it is a matrix this is a matrix this matrix is A inverse matrix A A transpose. So, A is a column vector A transpose row vector. So, this is the matrix at once again A inverse this is a result I will prove this result it is not very difficult.



To prove that right hand side is the inverse of A plus small alpha A A transpose what I have to do I will multiply this quantity within this bracket by the right hand side and so this is identity. Suppose you have to show that A inverse is B what will you do you then show A B is I of course, A B they are square. So, there are square matrices and A inverse is called B. You have to show that A inverse you have to show that certain A inverse is equal to certain B how to show if I can show A B equal to identity that means B is the B is A inverse

I because A square and A is supposed to be invertible. So, A inverse A into B is A inverse I A inverse I is A inverse A inverse A is I.

So, I B is B. So, B equal to A inverse. So, assuming A to B invertible if you have to show that A inverse equal to some matrix B how to show that just take A and B multiply A with B A B. So, that equal to I that means A is the inverse of B B is the inverse of B that we know this from basic matrix algebra which you all studied in fact in school. So, let me erase this part once again I just write the formula quickly for you that was A plus assuming this to be invertible then the claim these are claim this inverse is equal to A inverse minus alpha by times A inverse. These are claim proof this whole thing into right hand side that is if I do this A plus alpha A A transpose this quantity not inverse this quantity into right hand side right hand side we show that this quantity into right hand side is equal to identity means right hand side is the inverse of this as I told you A B A inverse is B means A B is I if you can show A into B is I.

$$\left(\underline{A} + \alpha\underline{a}\,\underline{a}^t\right)^{-1} = A^{-1} - \frac{\alpha}{1 + \alpha\underline{a}^t\underline{A}^{-1}\underline{a}}\,\underline{A}^{-1}\underline{a}\,\underline{a}^t\underline{A}^{-1}$$

So, that thing works here right this is what I have to show. So, we just work out this term little messy A plus alpha A A transpose right hand side is our claim is this part before I write this I can make it with some simplifications you see one thing A transpose A inverse A this is a scalar because A as I told already A is a matrix assuming it is invertible A inverse is a matrix matrix into column vector A is a column vector A transpose is a row vector row into column is a scalar. Let me call that scalar gamma in the denominator I have 1 plus gamma which is equal to say suppose I call beta say say then notationally it becomes simplified nothing else. So, it becomes alpha by 1 plus gamma sorry there is an alpha here.

$$\left(A + \alpha\underline{a}\,\underline{a}^t\right).RHS = \underline{I}$$

$$\underline{a}^t\,\underline{A}^{-1}\,\underline{a} = \beta$$

So, I cannot do like this. So, let me keep it just gamma just fine or maybe instead of gamma because gamma I will be using afterwards. So, we have to be cautious maybe call it beta say. So, this is 1 plus alpha by 1 plus alpha into beta right 1 plus alpha by 1 plus alpha into beta times as before. So, if I multiply A A inverse that is identity then alpha A A transpose A inverse alpha A A transpose A inverse then A times this quantity. So, A A inverse cancels identity.

$$\left(A + \alpha \underline{a}\, \underline{a}^t\right)\left(\underline{A}^{-1} - \frac{\alpha}{1+\alpha\beta}\, \underline{A}^{-1}\underline{a}\, \underline{a}^t \underline{A}^{-1}\right)$$

So, minus alpha by 1 plus alpha beta A n this is A A transpose A inverse alright and lastly this term with this term. So, alpha square by 1 plus alpha beta alpha square by 1 plus alpha A A transpose with B C, but actually not B C A A transpose A inverse A A transpose A A inverse looks messy, but actually not. This term looks very complicated A column vector then row vector then matrix then A column vector then row vector all that, but you see if I take out this part this is scalar and this is this beta is it A transpose A A. So, at beta is a scalar. So, beta is a scalar like 2 3 4 anything I can pull it out ok.

$$= \underline{I} + \alpha \underline{a}\, \underline{a}^t A^{-1} - \frac{\alpha}{1+\alpha\beta}\, \underline{a}\, \underline{a}^t \underline{A}^{-1} - \frac{\alpha^2}{1+\alpha\beta} \underline{a}\, \underline{a}^t \underline{A}^{-1} \underline{a}\, \underline{a}^t \underline{A}^{-1}$$

So, alpha square beta. So, it is identity I plus alpha A A transpose A inverse these are all here as it is minus alpha square by 1 plus alpha beta this beta comes up here I can pull it out with a scalar. So, A A transpose A inverse A is a column vector A transpose is a row vector. So, this is a matrix matrix into matrix matrix same matrix here here here. So, I can take that common I plus rest are scalar. So, you can write does not matter you can first write the matrix and then scalar alpha times this matrix alpha scalar minus alpha by 1 plus alpha beta and then minus alpha square beta by 1 plus alpha beta.

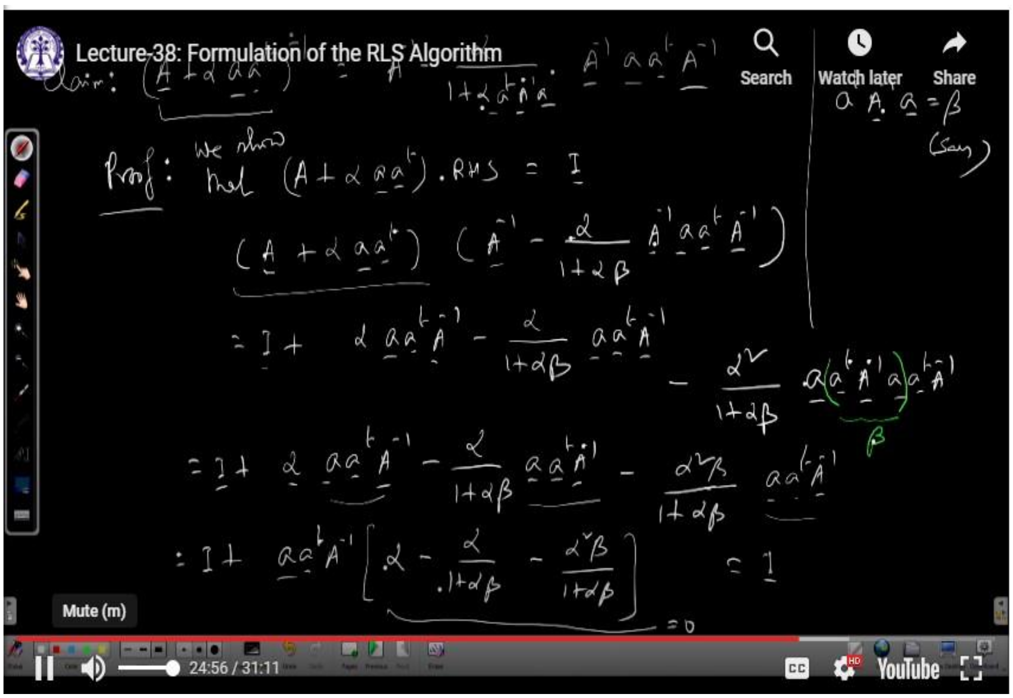
$$= \underline{I} + \alpha \underline{a}\, \underline{a}^t A^{-1} - \frac{\alpha}{1+\alpha\beta}\ \ \underline{a}^t \underline{A}^{-1} - \frac{\alpha^2\beta}{1+\alpha\beta} \underline{a}\, \underline{a}^t \underline{A}^{-1}$$

$$= \underline{I} + \underline{a}\,\underline{a}^t \underline{A}^{-1}\left[\alpha - \frac{\alpha}{1+\alpha\beta} - \frac{\alpha^2\beta}{1+\alpha\beta}\right]$$

And let us see what happens alpha into 1 you can see easily alpha into 1 alpha minus alpha cancels the alpha square beta minus alpha square beta cancels. So, this whole thing is equal to 0. So, this is identity which proves that one is the inverse of the other that is if I call this matrix something say theta and this is let this matrix be phi then theta into phi identity.

$$= \underline{I}$$

So, phi is the inverse of theta, theta is the inverse of phi with the square matrices which proves this result ok. This is the result I will be using in derivation this is very famous result for signal processing in particular ok.

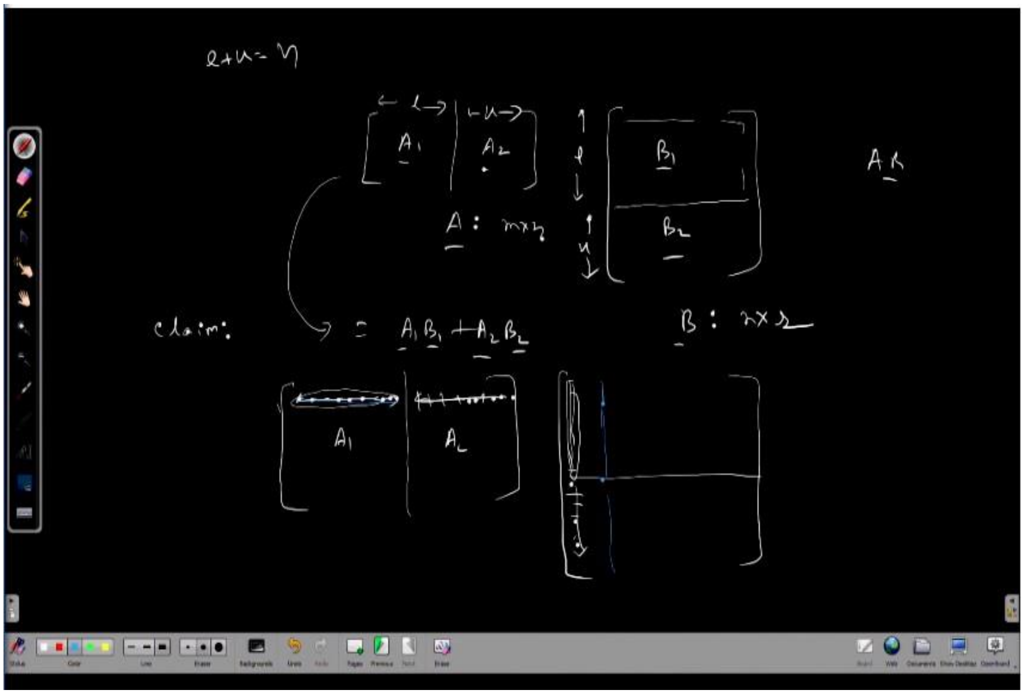


And this is the result I will be using in the derivation of the RLS algorithm. And another thing very simple, but I will be using it is called block partitioning of matrices and multiplication block multiplication. Again this is from elementary matrices theory if you easily appreciate. Suppose you got a matrix A which is m cross n and a matrix B which is n cross may be r you are multiplying obviously we will take this row into this column take this row next column and so on and so all of us know. But suppose I divide this matrix into sub matrices that is this may be L number of columns and this may be K number of columns that is L plus K equal to total number of column N.

So, this part I call A1 this part I call A2 and similarly here I take L so I take L here and this is K. So, this part of this matrix B I call B1 this I call B2 then by claim this will be same as claim this is same as equal to A1 B1 plus A2 B2 very simple. Consider this this is visually very clear A1. So, consider this A1 and A2 the first row first row of this total matrix A. So, this has one part this is one part and similarly if you take the first column of this total matrix B this is one part here one part here original A into B how will I how was I doing A into B this row times this column.

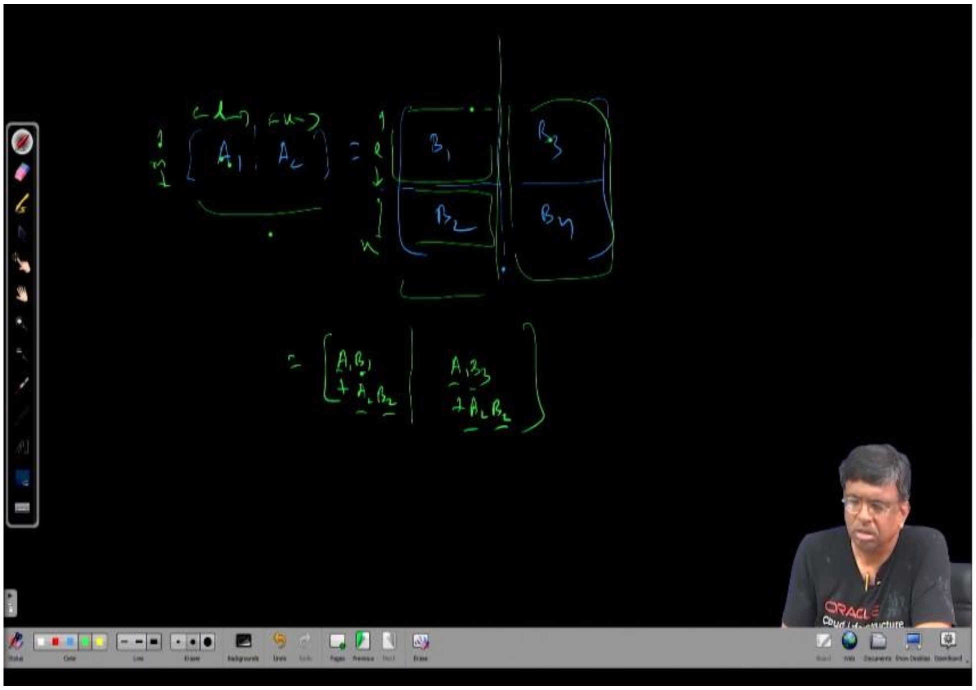
But that you can break as this times this plus this this times this that is what you have here you see A1 B1 means first row of A1 which is this into first column of B1 which is this. So, you are multiplying so that component is coming and additional component this into this is coming from here first row of this which is this much first column of this which is this much their product and you are adding. So, you are getting one element then you go to again this row times this means what this times this plus this times this, this times this means first row of A1 second column of B1 and again first row of A2 second column of B2 that is what you have here also A1 B1 first row into first column then first row into second column of B1. Similarly first row of A2 into first column of B2 then first row of A2 into second column of B2 that is what you are doing.

So, you get that result right. So, this way you can very simple you can take some examples of you know I mean maybe some 2 by 3 by 3 or 4 by 4 matrices and like that you know some simple matrix you can easily identify even otherwise also it is visually very clear. So, you can partition matrices like this blockwise ok, you can partition matrices blockwise and carry out the product like this. So, AB will be A1 B1 so A1 has L number of column so this has L number of rows A2 has K number of columns so B2 has K number of rows so A1 B1 you can carry out the product and A2 B2 ok.



This way you can generalize further generalize like this A1 A2 and B matrix you are not only breaking here you are breaking here it is B1 B2 B3 B4 this is m cross n.

So, this was 1 minute L K ok. So, this was L this was K this part is m so this is your L this is your K. So, this is B matrix I divide like this so this part as the I carry out product of this A with this first. So, I use the previous philosophy it will be A1 B1 plus A2 B2 ok that will be A times this sub matrix. So, it will be this part first A1 B1 plus A2 B2 this will come here and then again you do this product total A matrix into this part. So, A1 B3 plus A2 B2 this simple this just generalization this what I will be using extensively in my derivation of the recursively square algorithm.



So, this algorithm I will develop in the next class. Thank you very much.