Introduction To Adaptive Signal Processing

Prof. Mrityunjoy Chakraborty

Department of Electronics and Electrical Communication Engineering

Indian Institute of Technology, Kharagpur

Lecture No # 40

Derivation of the RLS transversal adaptive filter

So, we continue from where we left last time. So, some of the definitions I will rewrite again for our convenience. Xn is the data vector at the filter input at nth clock which is X of n. Then the data matrix the last row was transpose version of this. So, Xn X transpose n minus 1 dot dot dot X transpose 0 we are assuming all data to be 0 before n equal to 0. So, it is basically if you take out this part it is Xn minus 1 capital Xn minus 1 all these were discussed last time.

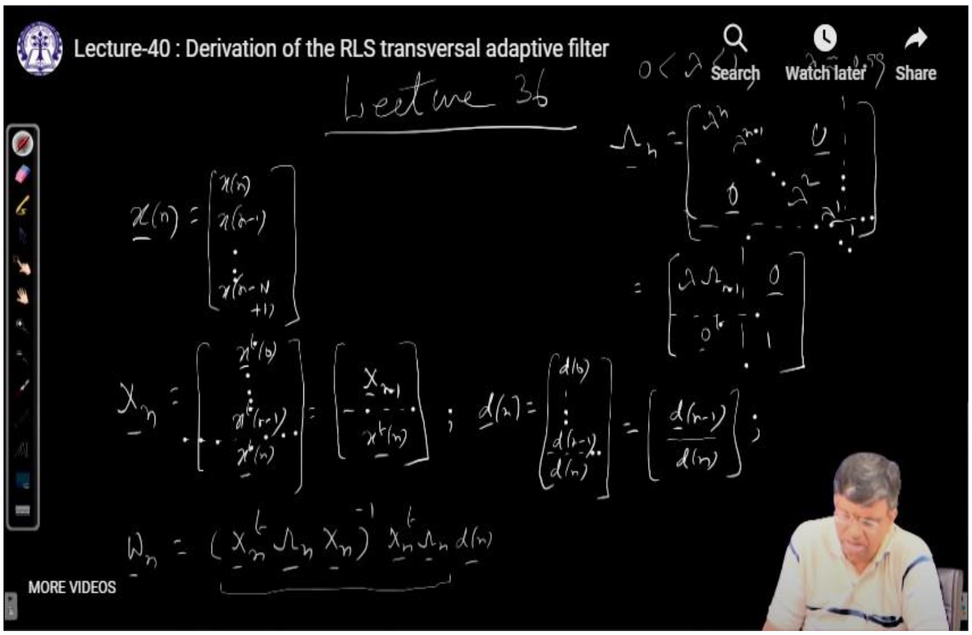
$$\underline{x}(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-N+1) \end{bmatrix}$$

$$\underline{X}(n) = \begin{bmatrix} \underline{x}^t(0) \\ \vdots \\ \underline{x}^t(n-1) \\ \underline{x}^t(n) \end{bmatrix} = \begin{bmatrix} \underline{X}_{n-1} \\ \underline{x}^t(n) \end{bmatrix}$$

So, I will not go to explanation again. This was one thing. Then we had a lambda forgetting factor which is and lambda was typically 0.99 or something.

So, we had this matrix lambda to the power n lambda to the power n minus 1 dot dot dot lambda square 1 0 0. So, if you take out this part. So, these remains all of them have lambda take lambda common. So, it was lambda to the power n minus 1 then it is n minus 2 and dot dot dot. So, it is lambda to the power the sub matrix is this if you take out lambda it is that lambda.

So, you had lambda to the power n minus 1 n minus 2 dot dot dot up to 1. So, that is your this matrix lambda n minus 1 and then multiplied by lambda which we took out remaining part 0 0 there is 0 transpose because a row vector is 0 column 0 vector 1 this is the 1. This we did then that optimal weight was this xn transpose lambda n all invertibility of this was discussed at length when invertible. Sometimes it may not be invertible specially when we are in the early stage there is small n early stage of the iterations that is small n time index is less than capital N something that all these were discussed, but we are assuming that for the time being small n is large greater than capital N and this is invertible under that we have dN, dN is the desired response vector. So, if you take out this part it is dN minus 1 vector and then scalar these are the things this is actually generalized pseudo inverse of WN normally in pseudo inverse this lambda is 1.



So, lambda n matrix is identity it is just xn transpose xn inverse assuming it is invertible xn transpose this part is the pseudo inverse. Now, I have brought in that additional factor weighting matrix is called weighting matrix weight factor is lambda. So, it is more generalized version of the pseudo inverse. So, that times dN is WN all right these are the definitions and then we worked out further this this matrix this part ok. Let me go to the next page.

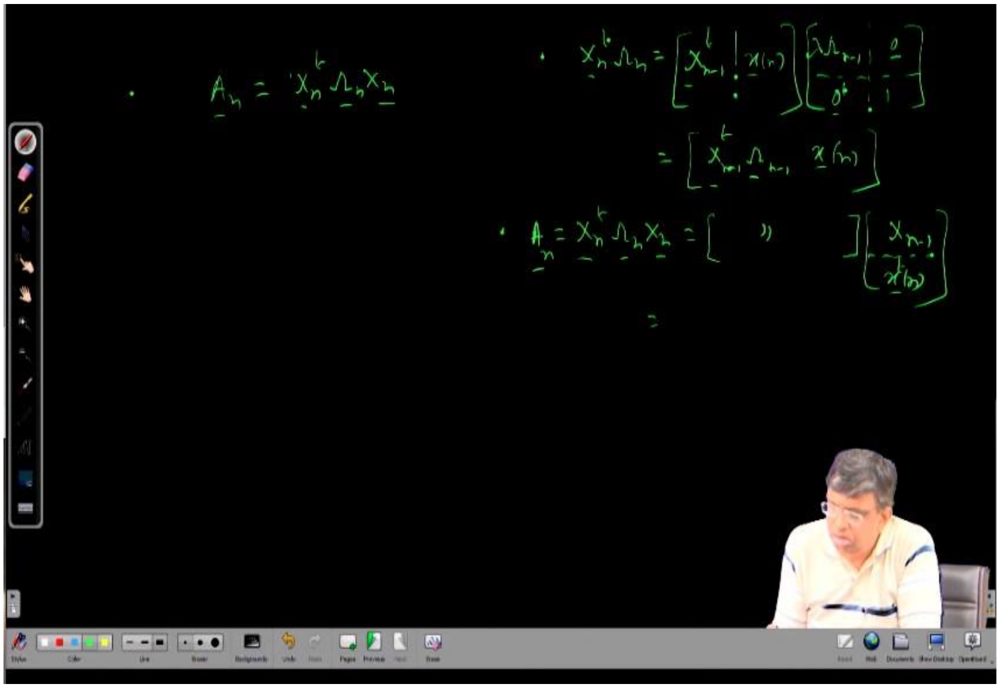
So, AN was ok this AN then we have seen xn transpose lambda N previously we have seen this is a repetition, but still does not matter we should we can do sometimes some repetition if you take xn transpose this first row becomes first column second row becomes second column like that. So, first comes xn minus 1 transpose then it gets transpose and comes first and this last column last row becomes last column. So, x transpose N is a row. So, column will be xn. So, it was discussed.

$$\underline{A}_n = \underline{X}_n^t \underline{\Lambda}_n \underline{X}_n$$

$$\underline{X}_n^t \Lambda_n = \begin{bmatrix} \underline{X}_{n-1}^t & \underline{x}(n) \end{bmatrix} \begin{bmatrix} \lambda \Lambda_{n-1} & \underline{0} \\ \underline{0}^t & 1 \end{bmatrix}$$

So, this is xn and as we have done in the previous page it was lambda times sorry lambda times. So, if you multiply this into this plus xn vector into this 0 xn column vector 0 transpose is a row vector, but that will give you all 0 matrix. So, it is xn minus 1 transpose matrix times this matrix. So, it was and then this matrix times 0 column vector plus xn vector times scalar 1 you get xn. All these were done last time in detail it is just recap this is one thing then after this we have to multiply by xn.

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



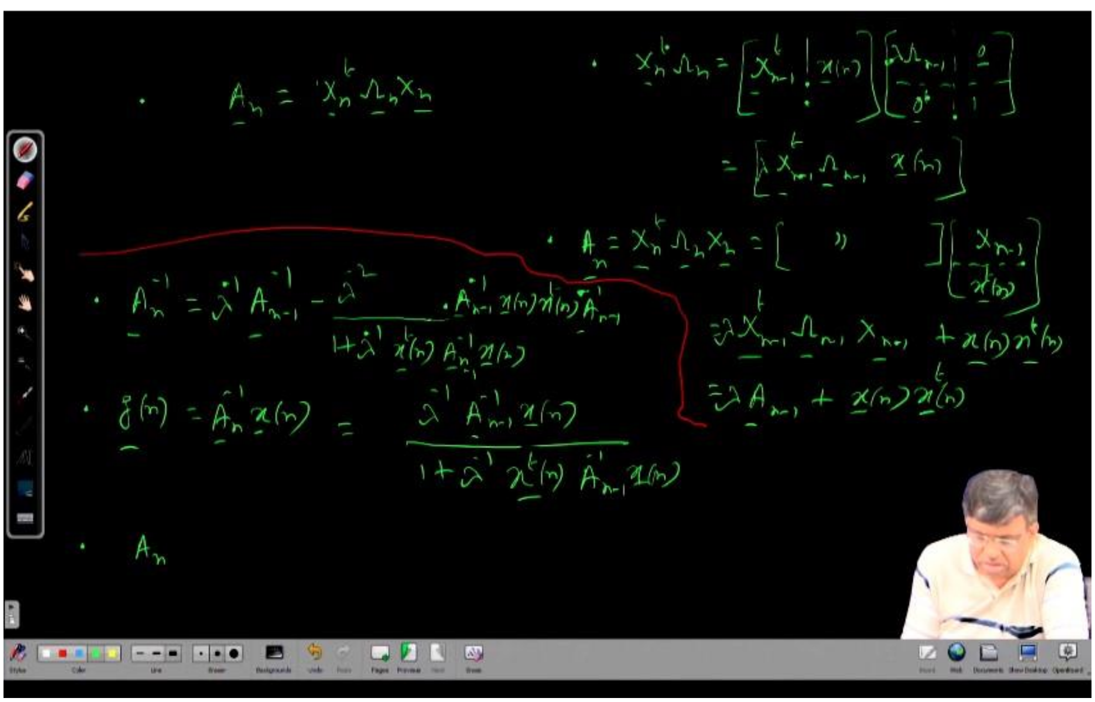
So, xn transpose so actually an is means this times I am writing this means this above thing into xn and xn we have seen xn minus 1 and so that by virtue of that block matrix multiplication rule this block will multiply xn minus 1. So, it will be transpose xn minus 1 plus xn x transpose n this is a very important result this is equal to an minus 1 same n if you replace n by n minus 1 you get this. So, an minus 1 plus this there is a lambda factor I missed out here xn yes lambda into xn minus 1 transpose this is lambda. So, this lambda is here this is very important result this shows how from an minus 1 an is generated it is multiplied by lambda, lambda is less than 1. So, every element here is suppressed by a small factor by a factor which is almost 1 and an additional matrix column vector into row vector additional matrix is added that is how from n minus 1 x transpose will move to nth x transpose in terms of for the a matrix is this.

$$\underline{A}_n^{-1} = \lambda^{-1}\underline{A}_{n-1}^{-1} - \frac{\lambda^{-2}}{1 + \lambda^{-1}\underline{x}^t(n)\underline{A}_n^{-1}\underline{x}(n)}\,\underline{A}_{n-1}^{-1}\underline{x}(n)x^t(n)\underline{A}_{n-1}^{-1}$$

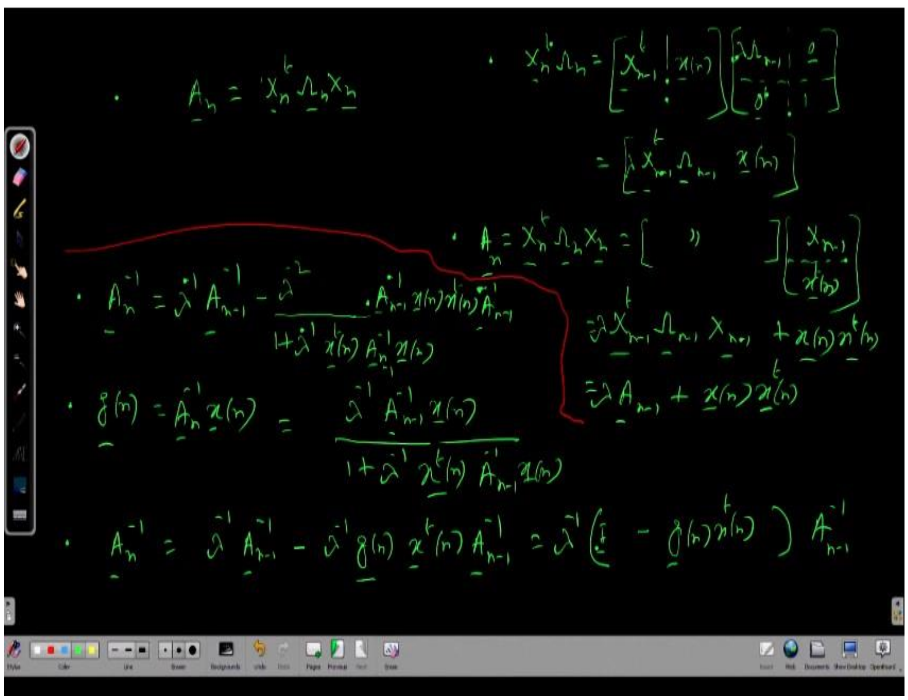
So, this is an important result then using matrix inversion lemma and always assuming invertibility we obtain this very important result lambda inverse. So, from the inverse also from corresponding to n minus 1th index how the inverse of a matrix and the nth index is generated that also you can work out using that matrix inversion lemma and that was actually this by there is a scalar this scalar lambda inverse xn transpose an inverse xn row vector matrix column vector matrix into column vector is column row into column is scalar. So, this whole thing is a scalar this times this matrix part is a scalar. So, let me draw a line here. So, that this is not mixed up with that all right this we obtained then I define the vector gain vector gn is an inverse xn and this turned out to be this we worked out I am not going to re derive things I am just quoting the result it was lambda inverse this first two sorry denominator as it is.

$$\underline{g}(n) = \underline{A}_n^{-1}\underline{x}(n) = \frac{\lambda^{-1}\underline{A}_{n-1}^{-1}\underline{x}(n)}{1 + \lambda^{-1}\underline{x}^t(n)A_{n-1}^{-1}\underline{x}(n)}$$

This is scalar that does not change this is n minus 1 lambda inverse by this denominator and then first two terms that was gn is a gain vector see one thing computing the inverse of a matrix is very cumbersome and all, but here we do not compute the inverse you know explicitly if the inverse is given for the previous clock. So, this is given. So, just by this given matrix in all these places you work out the sum this and you get new inverse. So, you do not have to calculate the inverse of an explicitly requiring your computation from an minus 1 inverse you generate an inverse just by an equation like this, but this an minus 1 inverse is known. this is known this is known this is known this is known ok.



So, this was gn using this gn you also had obtained this relation an inverse can be written like this first term as it is minus take 1 lambda inverse the remaining part is gn this part is gn and then last two all right and take lambda inverse common an minus 1 inverse common for both side. So, that means, I have to have an identity matrix matrix identity into this is this minus just gn times x transpose n that is all all right this common. So, these are the relations we derived last time and now I have to consider this wn I have to work out wn using them an inverse and this relation. So, now, wn is an inverse remaining part was xn transpose lambda n dn right. So, now, this term an inverse this we have seen xn transpose lambda n x transpose lambda n is this lambda xn minus 1 transpose lambda n minus 1 and then xn and dn.



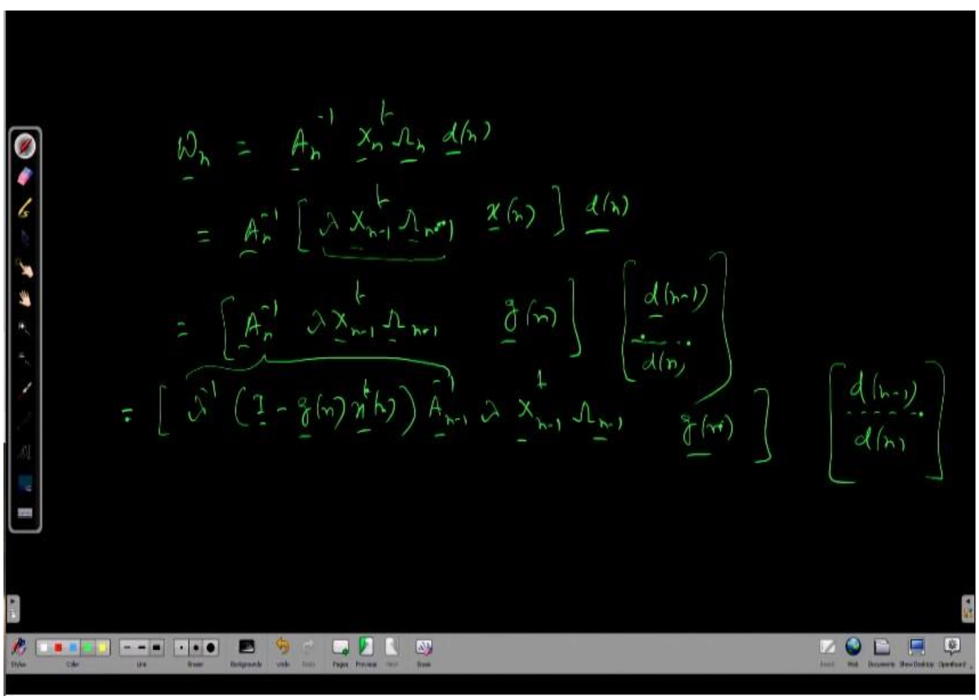
Now, an inverse xn that was defined to be gn an inverse xn is gn straight away second term is gn second term is gn. So, first is this an inverse this quantity lambda and straight second is gn and also dn vector can be written as dn minus 1 vector and scalar dn remember this this is what. So, far so good now an inverse an inverse we worked out lambda inverse then this into an inverse ok. So, we put that here lambda inverse in within bracket identity minus gn x transpose n and this. So, an inverse sorry this is lambda inverse and let us see again lambda inverse I minus gn x transpose n.

$$\underline{w}(n) = \underline{A}_n^{-1}\underline{X}_n^t\underline{\Lambda}_n\underline{d}(n)$$

$$= A_n^{-1}\left[\lambda X_{n-1}^t\Lambda_{n-1}\ \underline{x}(n)\right]\underline{d}(n)$$

$$= \left[\underline{A}_n^{-1}\ \lambda X_{n-1}^t\Lambda_{n-1}\underline{g}(n)\right]\begin{bmatrix}\underline{d}(n-1)\\ d(n)\end{bmatrix}$$

So, I minus gn x transpose n times an minus 1 inverse this much is this an inverse is this much then you have got this lambda this goes there and this gn fellow is here multiplied by this. So, this is the matrix it looks bit complicated, but actually it will be simplified very soon first you see lambda is a scalar. So, scalar number can be brought in the front and lambda lambda inverse cancels lambda goes all right. So, you are left with I minus gn x transpose n this is matrix identity matrix minus column vector into row vector is a matrix times this much this much and gn into this ok. Now, this is this is a matrix this is a matrix matrix into matrix is a matrix.



So, this matrix and this is a column vector. So, this matrix times dn minus 1 plus gn into dn ok. That means, I minus into this this matrix times dn minus 1 plus gn this much plus gn dn gn into there is a column vector that into scalar dn. We have already seen how to multiply by block matrices. So, this is one block this is one block.

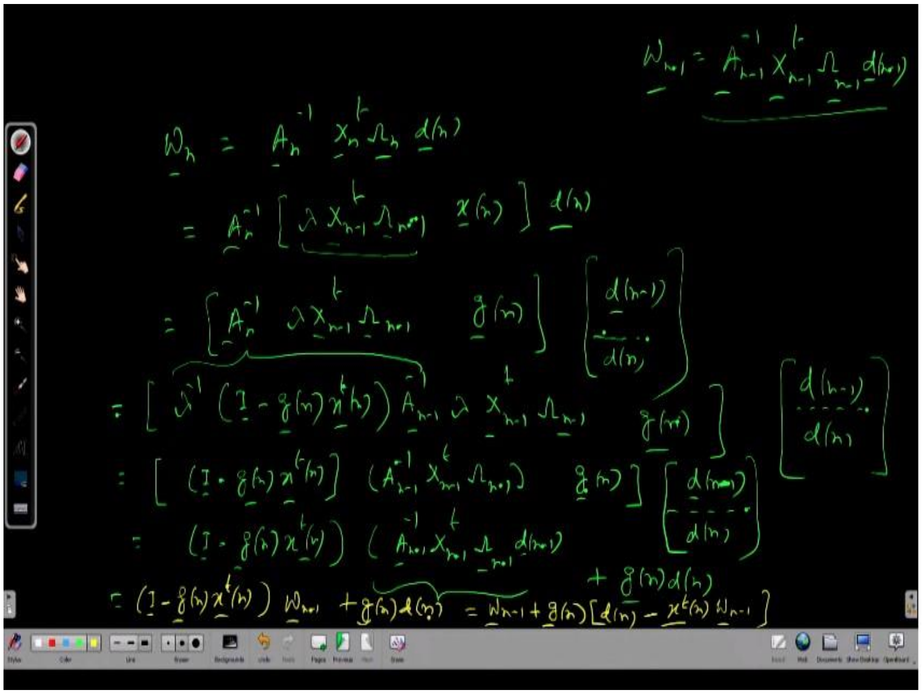
$$= (\underline{I} - \underline{g}(n)\underline{x}^t(n))(\underline{A}_{n-1}^{-1}\, X_{n-1}^t \underline{\Lambda}_{n-1}\underline{d}(n-1) + \underline{g}(n)d(n))$$

So, this column vector which is also matrix is divided into two block one block another this is scalar. So, gn times dn and this times this ok. Now remember one thing what is this? This is if this is wn what is wn minus 1 if you are at the n minus 1th block everything will become just function of n minus 1. And now see this is what I have here n minus 1 inverse x n minus 1 transpose lambda n minus 1 dn minus 1. So, this is nothing, but wn minus 1.

So, this is let me write using different things I minus gn times wn minus 1 plus gn dn dn is a scalar, gn is a vector. This is a matrix matrix into column vector. Now this gn so, I into wn minus 1 I is a 90 matrix that into wn minus 1 is wn minus 1. And then gn you take common. So, you have dn here scalar dn, dn vector the scalar dn minus gn if you take common x transpose n this is a column gn was a column vector x transpose is a row vector wn minus 1 is a column vector.

$$\underline{w}_{n-1} = \underline{A}_{n-1}^{-1}\underline{X}_{n-1}^t\underline{\Lambda}_{n-1}\underline{d}(n-1)$$

So, this row vector into column vector is a scalar the gn into that scalar. So, if you take gn out it will be this all right. I move to the next page deliberately did it here. So, that you can see things and then understand how this is coming all right. So, if I move to the next page I will rewrite this expression again.



So, wn is now obtained from wn minus 1. So, this is an update relation adaptation now from n minus 1th clock with the you have got the filter weight vector how to go to filter weight vector for the n minus clock it is an update relation. So, wn is wn minus 1 plus gn times I am simply rewriting from there dn minus x transpose n wn minus 1. Now see one thing suppose I use the filter I am having xn vector I am at the nth clock nth clock, but still at the nth clock if I do not use wn still use wn minus 1. So, filter output will be or equivalently x transpose n that means, though I am standing at nth clock suppose I am not using wn I am using the previous cycle I mean the w I obtained till the previous cycle that one I am using in the current cycle suppose and filtering the current input vector xn.

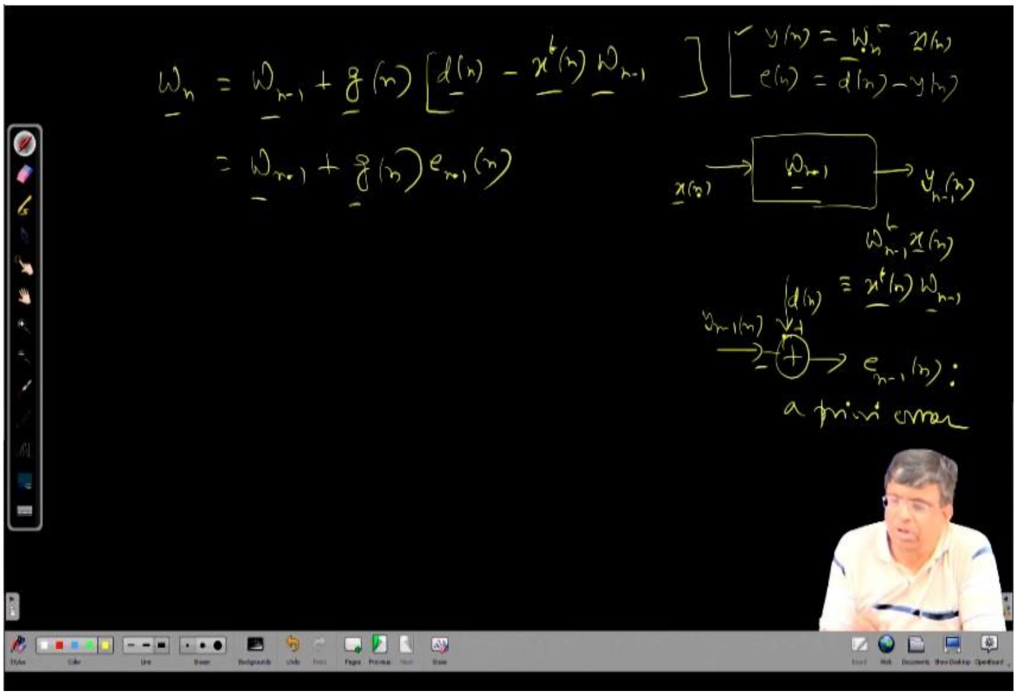
$$\underline{w}_n = \underline{w}_{n-1} + \underline{g}(n)[\underline{d}(n) - \underline{x}^t(n)\underline{w}_{n-1}]$$

So, output will be this all right. This output is not strictly the yn yn was yn when we normally see yn is w transpose n xn all right that is actually if you have wn use it to filter the current that the nth clock you have wn. So, fine you use it to filter the input vector. So, this is the output that you can do after you calculate the wn I am changing the notation let me call it wn transpose because n is in the subscript here. So, if I find out wn then I can happily do this filtering using a current cycle filter out filter coefficient vector wn transpose times xn this is what I will be interested.

But for this calculation only this calculation that is how to obtain wn from wn minus 1 this is the update relation I have to do this also that is I have to put the previous cycle the weight vector filter coefficient vector wn minus 1 that is the one I obtained by doing least squares minimization up to n minus 1th clock taking data up to n minus 1th clock that was wn minus 1 that if I use at the nth clock and filter the current input vector that is obviously not yn, but just for calculation I need it. So, that is again wn minus 1 transpose xn or equal to xn transpose wn minus 1 because if you have two column vectors say a and b then a transpose b and b transpose a they are same. So, it is like this. So, I cannot call it yn. So, let me call it yn minus sorry yn minus 1n that is taking data up to n minus 1 and that corresponding filter weight vector if I filter the corresponding the input at the nth clock.

So, that is why the n comes here. So, I mean the nth clock, but using filter weight vector that was obtained by using data up to n minus 1th clock that is by least squares minimization taking data up to n minus 1th clock. This is not actual yn I am interested, but it is just this calculation and dn minus that. So, this is your this y and then if I subtract it is not en, en actually is you know dn minus actual yn, but here it is dn dn minus dn minus this quantity where this n, but index n minus 1 ok. So, I call it it is called a priori error. So, this is my weight update relation alright this is the weight update relation where I know how to find out en minus 1 by this gn how to calculate gn we know gn carries the information about lambda and all this apparently is free of lambda, but gn carries the information about lambda alright.

$$= \underline{w}_{n-1} + \underline{g}(n)e_{n-1}(n)$$



So, using all this we will write this RLS procedure as an algorithm. So, that we will work out in the next class. Firstly, we have to take care of the invertibility I am assuming that the inverses exist I am assuming these inverses exist, but that is not only at nth clock I am assuming a n is invertible a n inverse exists, but I am also assuming a n minus 1 inverse also exists like this. But this way if I go back when small n becomes less than capital N we have seen earlier then it is not invertible the matrix capital X N for those lower values of n will be rank deficient.

So, inverse will not exist. So, how to take care of all these when small n is that small less than capital N that is in the very early stages of the iteration n equal to 0 n equal to small n equal to 0 small n equal to 1 2 3 dot dot dot. So, that time it will not be invertible as such. So, how to take care of that there is one crucial issue that you have to take care then just some manipulations and all we will write all these steps in a tabular form and that will be our RLS algorithm which we will do in the next class. Thank you very much.