Introduction To Adaptive Signal Processing

Prof. Mrityunjoy Chakraborty

Department of Electronics and Electrical Communication Engineering

Indian Institute of Technology, Kharagpur

Lecture No # 17

Convergence Proof of LMS Algorithm

Ok, in the previous class we did a very important thing. We derived the one of the most popular forms of additive filter called Least mean square LMS ok. It was like this at any nth clock there are three operations, first filtering you have got filter coefficients at nth clock as Wn vector which is W0 coefficient, but a function of n because they are changing from index to index because of the adaptation. So, filtering part is this Wn filtering part it gives you Yn by virtue of this, Xn is the input data vector. This you are familiar with I have done it many times. So, filtering then output error computation En which is difference between Dn.

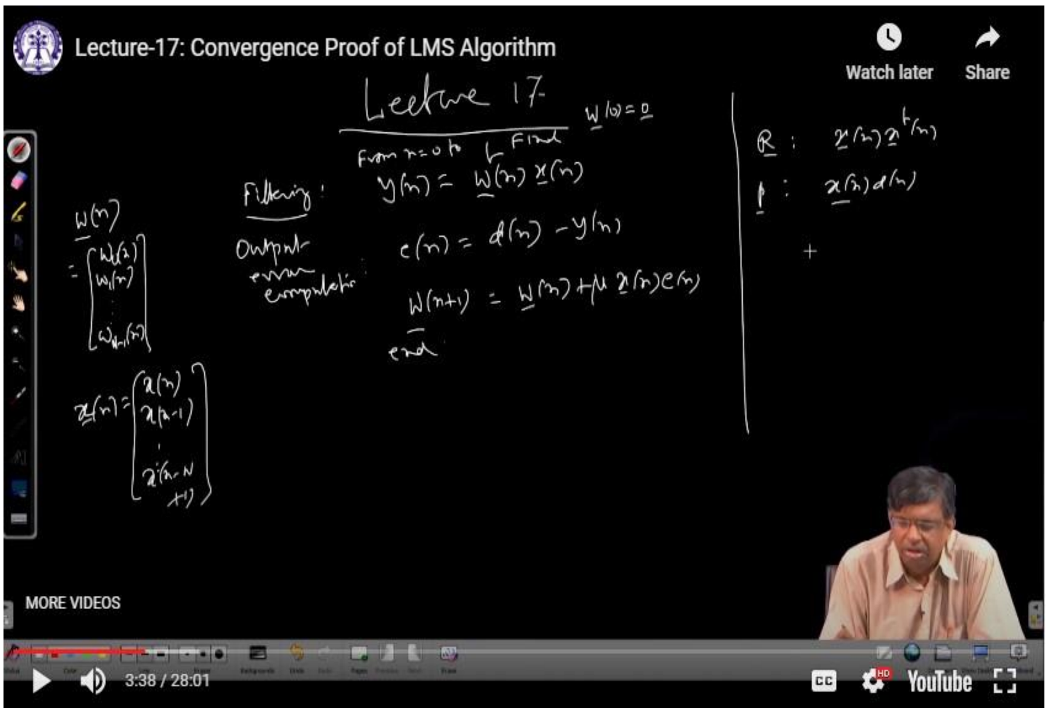
$$y(n) = \underline{w}(n)\,\underline{x}(n)$$

$$e(n) = d(n) - y(n)$$

So, Dn is given to you during what is called training phase during the adaptation phase ok. So, sample Dn is given and clock after clock you do this computation and using this En and Xn you move from Wn to Wn plus 1. There is filter coefficient vector to be used at the next n plus 1th cycle. There is that LMS formula from Wn plus mu is a step size very important parameter Xn vector En scalar end.

$$\underline{w}(n+1) = \underline{w}(n) + \mu \underline{x}(n) e(n)$$

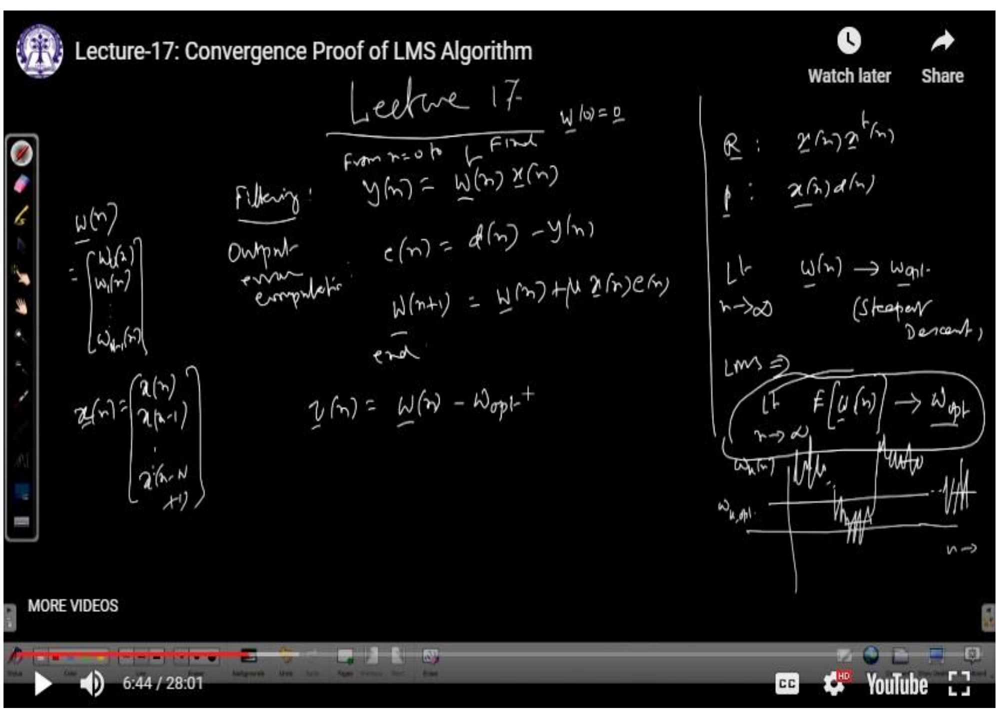
You can start from n equal to 0 to some point final of your choice and you can start with that initial condition, W0 required, because n is 0 means W0 must be known then only you can find W1. So, W0 any initial condition is fine, but normally we prefer 0 vector. This was LMS. And remember this was obtained from the steepest descent procedure by approximating R by a so-called quote unquote bad approximate bad approximation. Bad approximation in the sense we just took it to be Xn into X transpose n not averaging over many such you know matrices. This is a matrix Xn into X transpose there should could have been Xn minus 1 vector into X transpose and dot dot dot and they added and averaged that we are not doing we replace this by this and p by just Xn Dn.



Because of these errors and this part is called the gradient update I mean weight update part weight and the next cycle update part. Because of these approximations we pay a price the weights do not converge absolutely to the optimal one they converge in the mean that is limit n tending to infinity Wn was converging earlier to W opt in the case of steepest descent. We did not prove it, but from the graph we plot it from the figure it was clear it can be shown it converges directly, but in the case of LMS it does not. But in the case of LMS data vectors are random because you see with Wn I am adding mu into Xn vector into En. So, En into Xn En into Xn minus 1 En into Xn minus 2 and En again Dn minus Yn.

So, all those are random. So, every clock I am adding a random vector component to current to it and that is how Wn is generated by this process clock after clock, but there is also random. So, if it is random if you take the expected value at any clock. So, every point has some expected value that will go to sorry this is not X I am very sorry this W that is as I told if you take a particular weight Wk m this is a corresponding optimal value Wk its optimal value. So, Wk is random.

So, it might fluctuate like this like this like this its average is here, after sometime it may fluctuate here, then it may fluctuate here like this, but as it goes to very large index n there is n tends to infinity it will be fluctuating around the optimal one. So, mean of the fluctuation will be Wk opt then we try to minimize the range of fluctuation. So, it fluctuates in a small range around that optimal one which will serve our purpose ok that is controlled by mu actually, but that we will see later this was LMS algorithm. So, this I have to now show how this happen how this happens I have to now show this is my first goal all right. So, at any clock I define Vm as the weight error vector this is the optimal one this is the actual one at nth clock this much is the error.



So, this is the weight or coefficient filter weight or filter coefficient they mean the same thing weight error vector. I want to make a substitution I want to substitute Wn by this everywhere. So, that I get a corresponding equation in terms of Vn, because I have to show that this error vector in some sense goes down. So, I do one thing I subtract Vn from left hand side I subtract Vn from right hand side that will give me, sorry, I subtract W opt I subtract W opt from left hand side W opt from right hand side. So, Wn plus 1 minus W opt will be Vn plus 1, Wn plus 1 minus W opt, Wn plus 1 minus W opt that will be Vn plus 1 right hand side Wn minus W opt plus this.

$$\underline{v}\,(n) = \underline{w}\,(n) - w_{opt}$$

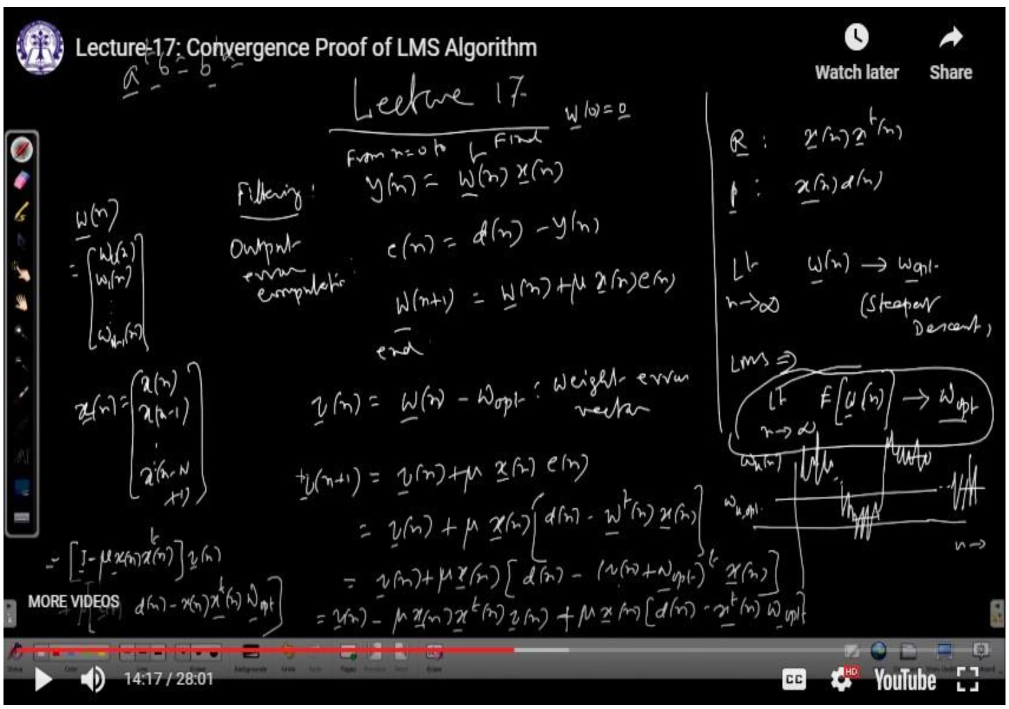
So, Wn minus W opt is Vn mu into xn En, but this is not enough because you know En itself contains Wn this is how, what is En? Dn minus Yn and Yn is W transpose n xn. So, here also I have to make substitution Wn in terms of Vn. So, Wn is equal to Vn plus W opt. So, that is what I will bring here. Dn minus Vn plus W opt transpose xn all right.

So, here what I will do mu xn then there is one term Vn transpose xn. Vn transpose xn you know it may give you two vectors a and b, a transpose b sorry, you know a transpose b is b transpose a and we have two column vectors this is very simple. So, Vn transpose xn or maybe the whole thing transpose xn I can write as Vn or ok Vn transpose xn that itself Vn transpose xn is same as xn transpose Vn and that I take out. So, I have Vn plus mu xn minus I am taking that particular term mu xn into minus of x transpose n Vn. So, it will be mu xn this one and then Vn transpose xn is same as x transpose v, a transpose b is b transpose a.

So, mu xn minus Dn minus W opt transpose xn ok. This also I write as x transpose W opt, a transpose b is b transpose a using that, this also write as x transpose n ok W transpose xn is same as x transpose n W opt alright. This is actually output of the filter if I use W opt as a filter coefficient vector. So, this is the best filters output error it will have the minimum variance x transpose n W opt and W opt transpose xn they are same, there is a filter output if I put W opt as the filter coefficients W optimal, corresponding error which will have the minimum variance because I am using W opt ok that error is given by this Dn minus filter output alright. So, I keep it as it is and I can further write it this way. I take identity matrix Vn is identity matrix into Vn identity matrix into Vn is Vn itself.

So, this is identity matrix into Vn minus mu xn x transpose n Vn. So, this Vn again I take common. And this much is as it is. Let me do one thing. Let me erase this bracket xn Dn minus again mu xn Dn minus mu, mu is common.

So xn is xn x transpose n W opt, this is what I have and we all know W opt is R inverse P right, W opt is R inverse P. So, we consider this second term now if I cannot proceed any further you know this is what I have done and you can try very well, but you cannot simplify any further you cannot proceed any further you have to do something very new now and then again doors will open you can proceed further.

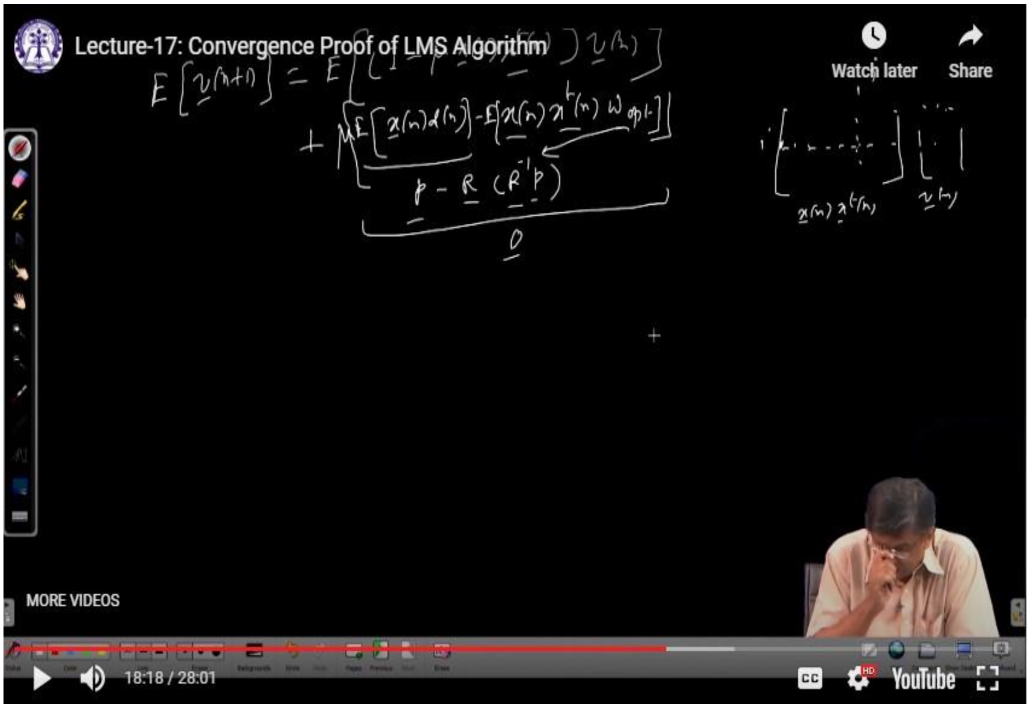


That is if I now take the expectation expected value of the left E of this and then E of right-hand side then again new door will open and let us see that expected value of Vn plus 1 will be expected value of these i-minus mu xn x transpose n outside Vn. Expected value of i minus mu xn plus the other term mu xn mu xn Dn minus xn x transpose n W opt expected value. Now expected value of this xn into Dn is my P vector right.

So, you can even work like this E on this minus E on this E on this this much is P by definition cross correlation vector between xn and Dn xn vector and Dn. So, this is P and here xn x transpose n there is a matrix with random elements times W opt W opt is constant. So, you can as well apply E on this matrix first and then multiply by W opt you will get the same thing ok that I have done earlier. So, E opt this is R, xn x transpose n that is autocorrelation matrix R and then W opt. W opt we know is R inverse P that is your W opt. So, R and R inverse cancels I get identity to P is P, P minus P this is 0.

So, this is how the door opens if I apply E on it. So, at least one part gets 0 becomes 0 and it becomes simplified all right. So, I carry on with this next I got x this term is a most complicated term xn x transpose n. So, I got a matrix xn x transpose n. So, every every element will be a product of one term may be xn minus k, xn minus i and may be another term xn minus j that will give you ijth element times Vn.

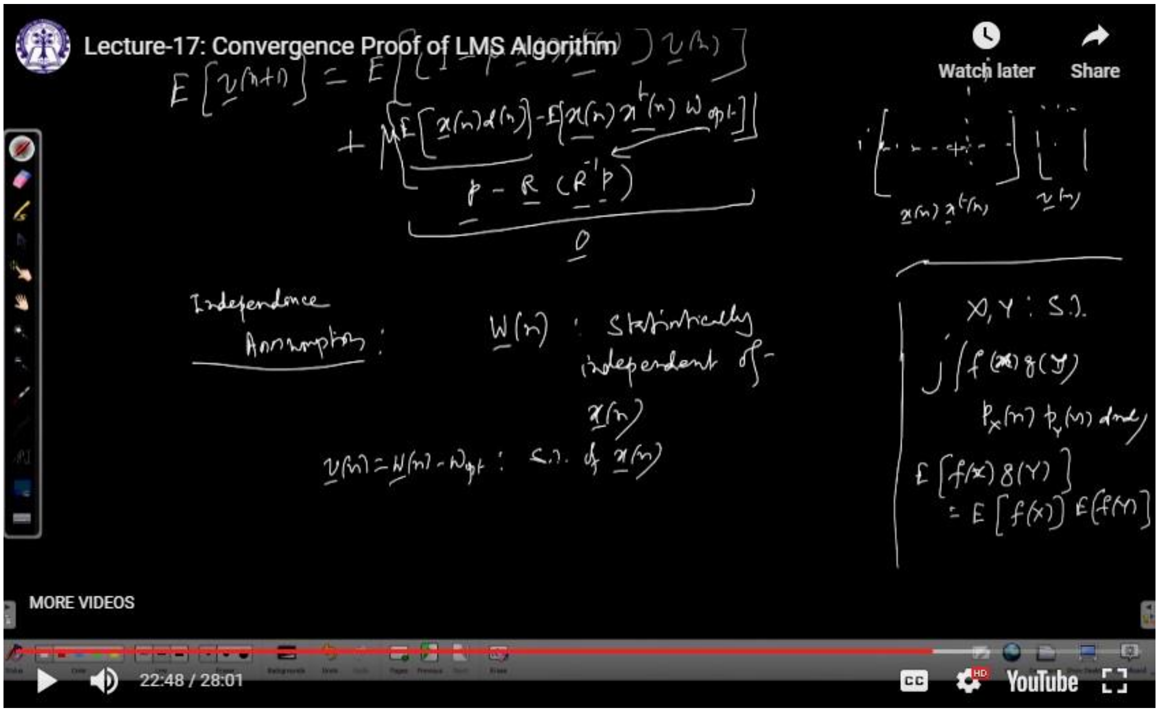
So, this is this this matrix times Vn. So, what will happen any row times this. That will be 2 terms of xn times one term of Vn multiplied again added with again 2 terms of xn multiplied by one term of Vn like that any row here, ijth and jth suppose this is xn minus i xn minus j and I am taking expectation of course, xn minus i xn minus j and when we multiply this by Vn we will multiply this by Vn minus I mean j this will be the multiplying top guy this the next guy. So, Vn minus j and this you keep doing for all the elements these times this, these times like that this times this, this times this, this times this and added. So, every term will have 3 components 2 multiplied from here and then further multiplied from one component from here.



So, 3 and then expected value. Here we make an assumption called independence assumption I will tell you later what is the basis of this.

So, we assume Wn is statistically independent this vector is random after all I have told earlier, but this is statistically independent of xn vector ok. That means, also Vn which is nothing, but Wn minus a constant W opt that is also statistically independent Si of xn. Now, why statistically independent because if I now apply E, you remember one thing from our previous discussion on probability random variables and all that. Suppose x y are Si statistically independent ok. Then if you have got some function like you know fx gy, fx gy then its expected value will be this into joint density, but joint density will be this because they are statistically independent and then double integral dx dy and you can take fx pxx under one integral that will give you expected value of fx gy py y under other integral that will give you expected value of gy. So, not only you have E of x y is E of x E y it will be for any function fx gy if you have anything that can be separated into one function of x another of y and take the expectation because probability density also joint density also is separable now product of marginal density for x marginal density for y the whole integral you can separate out ok outer and inner integral and that will give you the two expectations E of fx, E of fy.

But if they are only uncorrelated this will not happen because probability density if I cannot break down into product of individual marginal densities this cannot happen. Uncorrelated only means that expected value of x y is Ex into Ey that follows from here also if you have only fx equal to Ax gy equal to y x y and this means x times gy y times. So, you get E of x E of y. So, that I told that time that in the beginning only if they are statistically independent they are always uncorrelated, but this does not follow if they are uncorrelated because you must be able to you know break the probability density write it as a product of individual marginal densities, one of x another of y and then only this happens. I want that here because here I have got product of three terms one term of x maybe x n minus I, n minus j and one of vn expected value of that it is not simply x and v then I could have one x one v and E of that then I could have applied uncorrelated assumption and that would have been E of this x term into E of this v term, but here I have got more than that, here I have got one x minus i n minus j and then one term from v this product I have to apply E and I want it to be separable. For that this is required statistically independent. Then I can have the overall expectation is an expectation of this part ok function of this x and function of y.

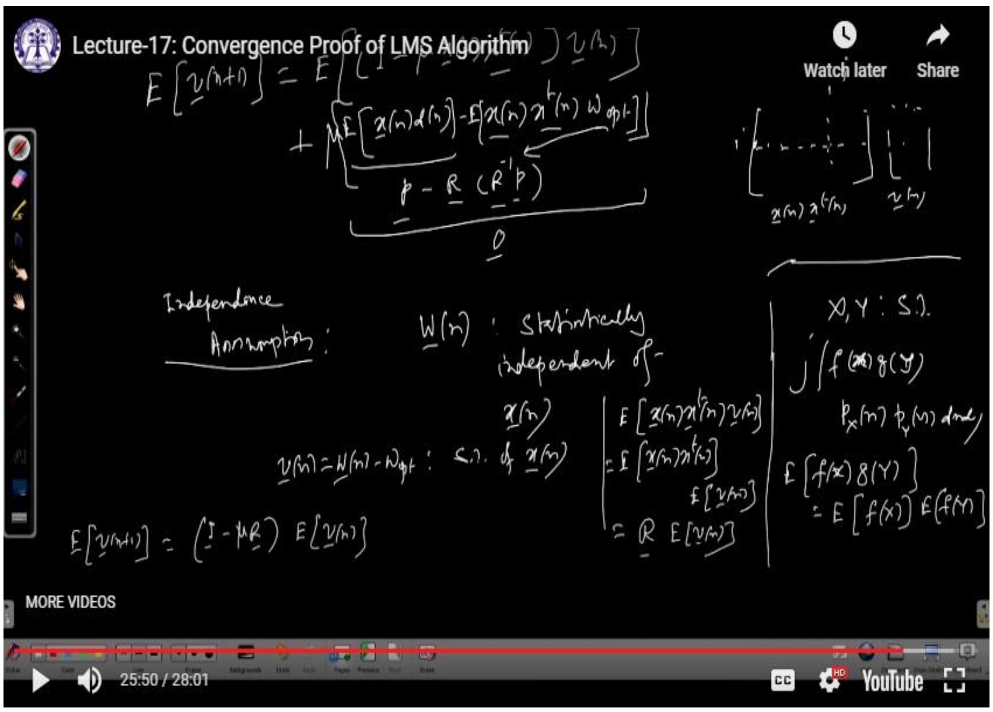


So, v so expectation of x n minus i x n minus j this part and multiplied by expectation of the corresponding term here, vn minus j, vn minus yeah vn minus j ok and so on and so forth for all the elements. So, it will be like E of the whole matrix because I am applying E on every term ok every term is a product of one x and one, one data of x and another data of x xn into x transpose n ok. So, if it is ith row xn minus i and then xn minus j. So, xn minus i into xn minus j that product and ith row times this. So, this will multiply vn minus j.

So, even it will be xn minus i xn minus j into vn minus j expected value of that under statistical independence if x parts all the x terms and all the v terms are statistically independent. Then it will be E on the components involving x only, like xn minus i xn minus j E of that multiplied by E of whatever data from, whatever components I have from vn in this case only one term ok. So, like I am putting E here I am putting E here and this have to do everywhere. So, this will become then E of so, under this assumption E of this matrix and vn it will be separable and this is my good old autocorrelation matrix R this is E vn. So, E of i into vn is vn E of vn minus E of this which is R into E of vn.

$$E\left[\underline{v}(n+1)\right] = \left(\underline{I} - \mu\underline{R}\ \right)E[\ \underline{v}(n)]$$

So, this will become this will give rise to this i as it is minus mu this part will give you R this R and E of vn. i into E of vn is E of vn that is what I have here E of i vn that is vn. So, E of vn i into E of vn, E of vn minus mu E on this part because of statistically independent. So, E on this part is R as I told you and E on this part E vn. So, take E vn common that is this ok.



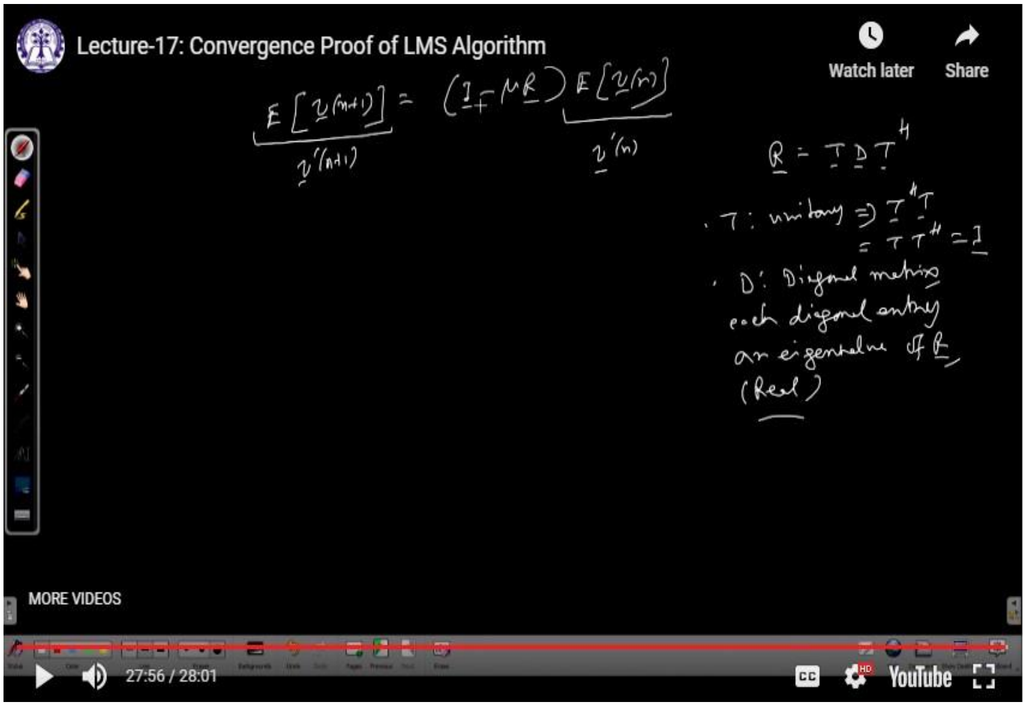
Very simple equation which is I will move to the next page I am rewriting what I obtain there ok.

E vn the moment we apply E it then the result is no longer random ok it is constant may be function of n. So, let me call it v prime n, this vector it is not random no function of n and then it is v prime n plus 1 all right this I do. Another thing we have seen R is a Hermitian matrix autocorrelation matrix that we can write it like T d T h I did it extensively very elaborately in my one of my lectures you know where T is unitary means T h is the inverse of T, T h T and if T h is the inverse, you can also have T T h both is identity. So, T h is the inverse of T, T is the inverse of T h. So, T is unitary basically all the columns are the orthonormal eigenvectors of R and d diagonal matrix each diagonal entry of an eigenvalue of R, but R is Hermitian.

$$\underline{R} = \underline{T}\,\underline{D}\,\underline{T}^H$$

$$= \underline{T}^H\underline{T} = \underline{T}\,\underline{T}^H = \underline{I}$$

So, they are real. So, I now have a much simpler equation v prime n plus 1 is i minus mu R, v prime n and R I will replace like this ok.



So, in the next section next up I will complete the proof. Thank you very much.