Introduction To Adaptive Signal Processing

Prof. Mrityunjoy Chakraborty

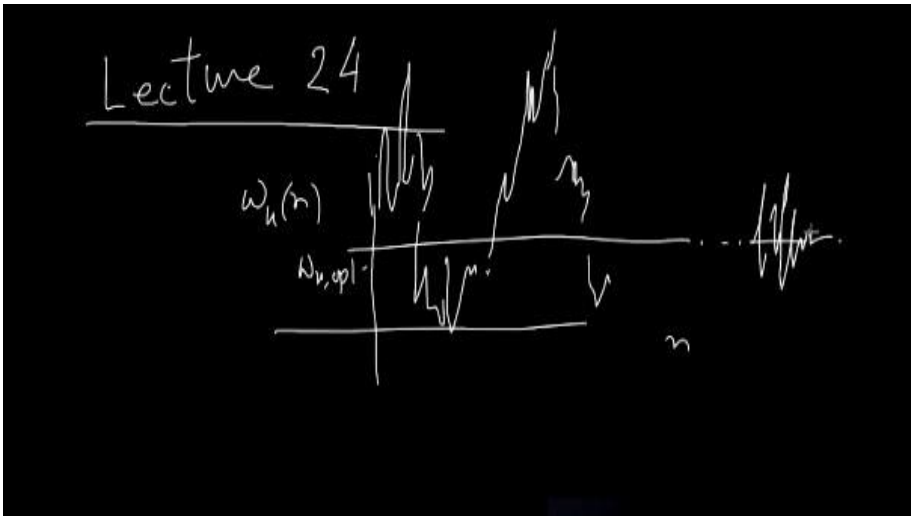
Department of Electronics and Electrical Communication Engineering

Indian Institute of Technology, Kharagpur

Lecture No # 24

Second Order Analysis of LMS Algorithm

So, we had seen earlier that the LMS algorithm the filter weights converge not absolutely, but only in mean to their respective optimal values as it has an index small n goes to infinity that is we need something called steady state that is for large value of n. Typically, we what we have there if you take say any kth weight we had already discussed this I am just repeating. So, wk n and you are going through iteration you are updating it via LMS algorithm ok, particular weight I am considering its optimal value may be here wk opt. So, initially it will fluctuate in any region you know if you go like this, but in the end, it will fluctuate in around the optimal one. So, if you measure the mean of this wk n at n very large n or as n goes to infinity like here the mean will be optimal weight.



So, we get convergence only in that sense then one can ask the question that how good it is because if it is still fluctuating over a wide range. So, at any iteration I get a value much

away quite different from wk op,t because it is fluctuating over a wide range how to control the range how to make sure that the range is small ok. So, for that we require a second order analysis that is, we will take the error between wk n and wk opt that is called filter weight error this is for only kth component, all the components put together will be a weight error vector vn we will take that and we will take the variance of each other ok. And sum all the variances that variance will give you the power of the fluctuation right because otherwise, it can I mean if you do not take variance if you take the just the value it can be highly positive highly negative when you add them is 0.

So, you can indicate that are on the average value is small. So, we are happy, but actually not because plus and minus both are there that is why they cancel each other, but the moment you take variance it is the difference between wk n and wk opt. So, that increment you are taking and you are taking the square of that. So, that positive increment you can get you get positive value positive power and their average. So, if that is less then the range of fluctuation will be less.

So, that is why the second order analysis that is variance analysis of the filter weight error ok that is what we have to we have to carry out and that is a very tedious exercise, but I will not go through all derivations because that is beyond the scope, but I will give some important results in between and then carry on with the remaining part of the derivation with that. So, that you can develop an insight all right. So, here I take the filter weight error vector not with respect to a particular component say kth component, but all together minus the corresponding opt all right vn wn as you know we have got.

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

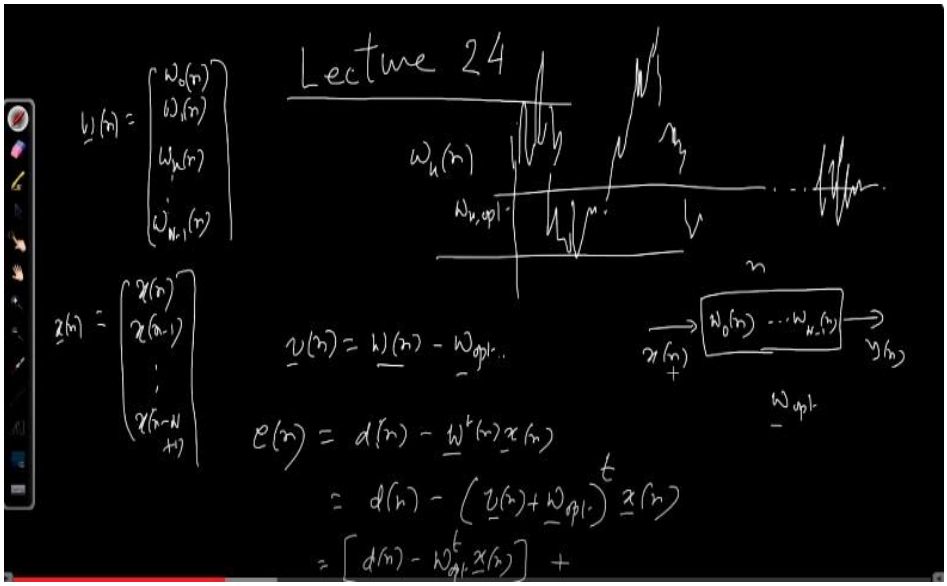
$$\underline{w}(n) = \begin{pmatrix} w_0(n) \\ w_1(n) \\ \vdots \\ w_{n-1}(n) \end{pmatrix}$$

So, all the weights you have wk n here and w opt is a corresponding optimal vector. So, this will give you the filter weight error wk n minus wk opt filter error for k equal to 0, k equal to 1, k equal to k like that all right this component ok.

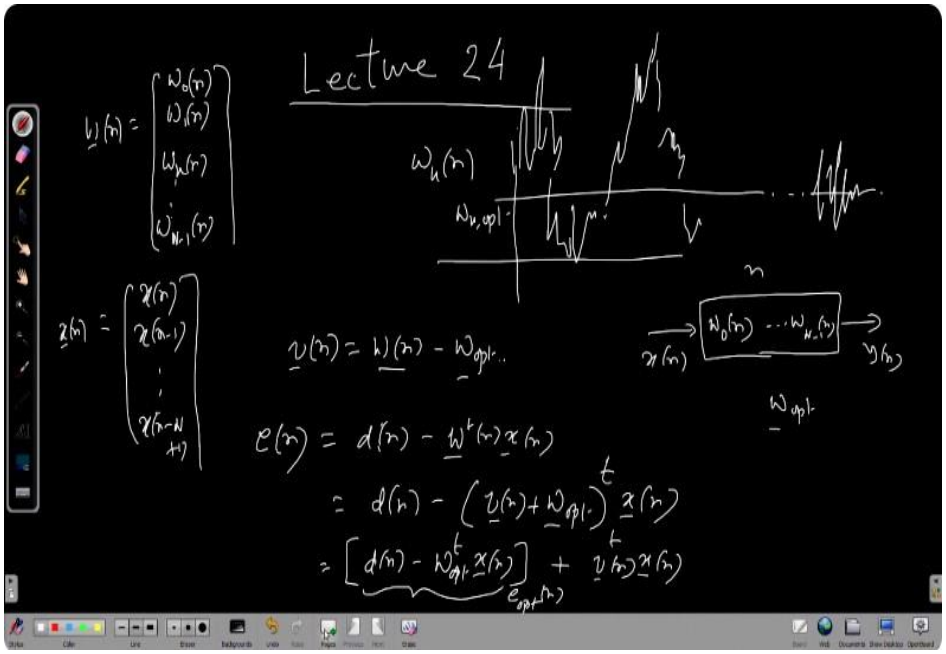
And in general, we are considering complex case. So, LMS algorithm in LMS or for that matter any adaptive filter says w n plus 1 or let me make it simpler. Suppose you have got an adaptive filter output error is en ok, not necessarily LMS what I am going to present is more general just an adaptive filter, filter coefficient vector at any nth clock is this you are updating them by some algorithm adaptive algorithm. So, at the nth clock the output error will be desired response minus as you have seen say w ok. Let us consider real case because I remember I have not done possible in the complex LMS algorithm.

So, let us carry out with the real case. So, in the real case it will be w transpose n xn right, w transpose n xn this is a filter output, because this is a filter xn goes in this you have seen many times filter output is yn at any nth clock, these are the filter these are the filter coefficients. So, this w transpose n xn means and xn is the data vector no change in definition I am just repeating them. So, w transpose xn is w0 xn plus w1 xn minus 1 dot dot dot which is the filter output and wn is updated by some adaptive algorithm fine, but at any nth clock wn is what if you take this w opt to the left-hand side it is a w opt plus the incremental part vn. So, if you replace that here it will be dn minus vn plus w opt transpose xn alright, then you take this in one place what is w opt transpose xn that is if I have w opt here then the corresponding filter output is w opt transpose xn ok.

Instead of this w0 n wn minus 1 n if I have w opt coefficients like w0 opt w1 opt wk opt and all that corresponding filter output will be that is w opt transpose xn like it is w transpose wn transpose in general case xn if wn is w opt it will be w opt transpose xn.



So, when you are putting the optimal filter, you are going to the best case because yn and dn they are difference en has a minimum variance that is how w opt was derived that you know optimal filter R inverse P that was derived by minimizing the variance of the output error between dn and yn ok. So, that is the best case that time yn becomes a very good estimate of dn because the error between them has been the minimum power minimum variance. So, that is this that is why I call it this also error output error filter output subtracted from dn, but this is an optimal error e opt n let me call it e opt n and another term I have coming from the weight error vector term vn transpose xn alright. Therefore, what is the variance of this error variance.



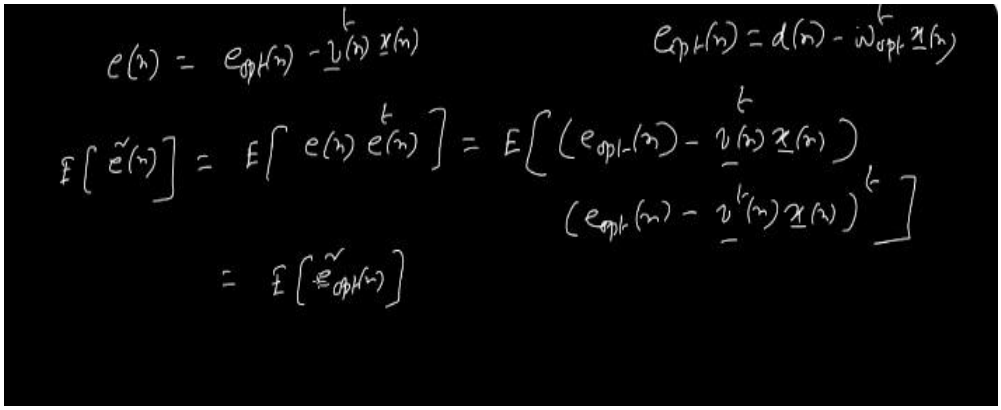
So, that is what we will work on in the next page. So, what we have is en is this. So, let me rewrite again what we have obtained is en is e opt n minus vn transpose xn where e opt n we have seen is dn minus the filter output with optimal filter as the filter coefficient vector ok. So, therefore, if I take the variance expected value of this is real. So, no point input in mod and square because square itself will make it positive.

So, e square n is this is a scalar alright this is a scalar the square of it I can always write like e opt n as it is or to make it even simpler for you I go through one more step e square n en is a scalar right. So, I can always say this is en en transpose because en is a scalar 1 by 1 matrix. So, its transpose is itself then if you put en as this it is e opt n minus vn transpose xn into its transpose minus or maybe I put the transpose outside minus the same guy transpose. And if you take down the product of these two e opt n e opt n transpose e opt n transpose is e opt n only. So, it will be expected value of e square e opt n square that is variance of e opt n because it is 0 mean and therefore, e opt n itself is a increment and squaring up and expecting variance.

$$e(n) = e_{opt}(n) - \underline{v}^t(n)\underline{x}(n)$$

$$E[e(n)] = E[e(n)e^t(n)] = E[\left(e_{opt}(n) - \underline{v}^t(n)\underline{x}(n)\right)\left(e_{opt}(n) - \underline{v}^t(n)\underline{x}(n)\right)^t]$$

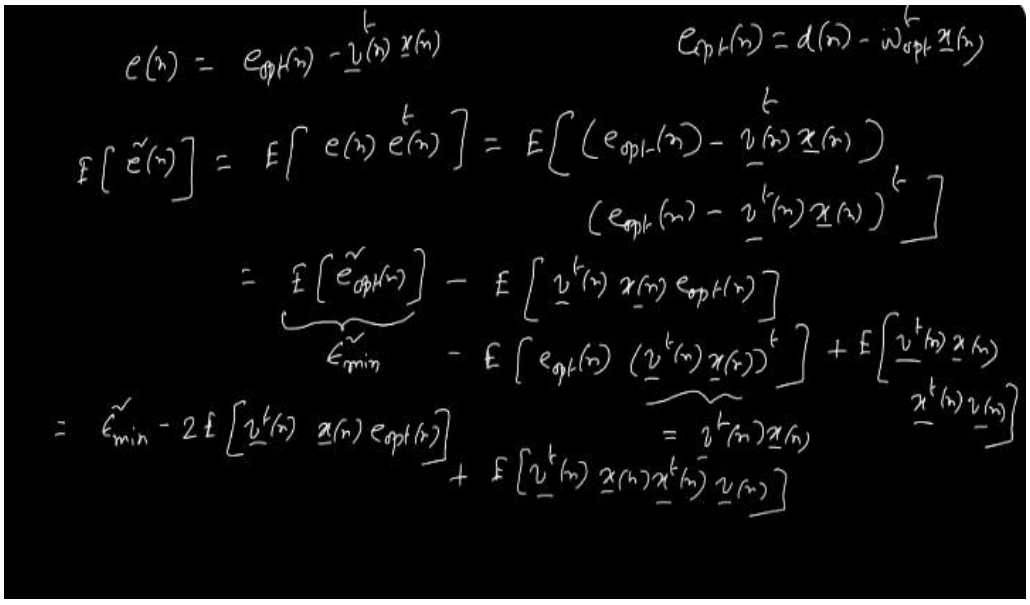
So, this is the minimum variance because e opt n means output error variance is minimized then only you put w opt n and the corresponding error has minimum variance. So, if you calculate its variance, it will be a minimum thing you know I mean if you square up and take expectation dn square if you take expectation n will disappear because of w sn. Similarly, cross terms also dn and xn elements of xn they are you know jointly stationary. So, again the cross correlations will have no n xn itself all the terms of xn they are w s s xn is w s s process. So, any correlation between them and again independent of n that is why variance is just a constant independent of n minimum attainable.



So, that is here what we will show is this except I mean in addition to the minimum attainable there will be additional components together will be this. This is happening because we are not using w opt when you are calculating en we are using some arbitrary wn which is updated by some adaptive algorithm. So, there en is not just e opt n and extra component coming due to the filter weight error that will contribute to some additional variance terms here that is what we evaluate this is fine. Then e opt this cross term v transpose n xn e opt n and minus actually e opt v transpose n xn e opt n and minus e opt n v transpose n xn transpose v transpose n xn transpose. Now, v transpose n xn is a row vector this is a column vector.

So, you got a scalar and scalar transpose itself. So, this is actually equal to itself because this is a row vector this is a column vector row vector into column vector is a scalar any scalar and its transpose, they are same. So, this scalar transpose of the scalar is the scalar itself. So, you can replace this by v transpose n xn v transpose n xn is a scalar multiplied by another scalar e opt n like here. So, these two terms are same and there is last term is expected value v transpose n xn if I take transpose of this x transpose v x transpose n vn all right.

So, this is equal to minus these two terms are same. So, twice e let me write v transpose n xn e opt n plus all right. Now, to analyze further we make an assumption we make a more generalized we earlier obtained I had what is called independence assumption right. In the elimination analysis we assumed that the current filter weight vector wn is statistically independent of the current data vector xn and the reason we made the assumption that was also the motivation was clear that time I made it clear that time. We will just extend it here we will assume.



We assume wn is statistically independent of xn of course, and also dn all right. Logic is same if you really write wn in terms of wn minus 1 wn minus 1 in terms of wn minus 2 and

all that and go up to w0 which is an initial condition. I mean all the other terms they did not involve xn and corresponding error en, but en will have again dn and you know xn I mean xn all right and so on and so forth ok. So, this summation as I explained that time will be largely you know dominated by terms from the past because there is that is majority in number and past terms are usually uncorrelated with or even statistically independent with current terms that is why you can say wn is statistically independent of this. If you do not understand you just go back to my lecture and that time go back to my lecture on LMS convergence analysis that time I made I explained what is meant by independence assumption.

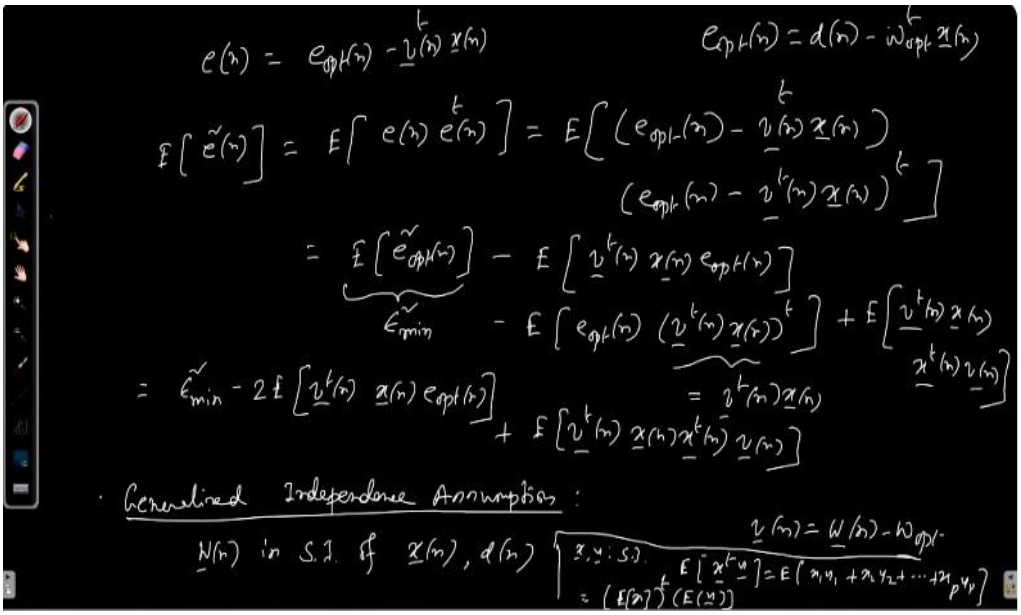
So, it is just this ok just that time it was wn was assumed to be independent of xn ok. It was assumed to be independent of xn and now dn has been brought into it all right dn has been brought into it. So, if that be now you see E of n it depends on dn it depends on xn w is constant here. That means, E of n is a function of dn and xn. So, here E of 10 is a function of xn and dn right consider vn, vn is nothing, but what is vn? vn is nothing, but wn minus w opt.

So, basically you are subtracting a dc a constant from wn ok. So, if wn is statistically independent of xn and dn so is vn, because vn and wn are equivalent it is just a dc shift of wn by a constant w opt. So, if wn is statistically independent of xn vector and dn scalar so is vn ok. So, here vn it is statistically independent of xn and E of n depends on dn and xn. So, it is a function of dn and xn this is the xn itself.

So, this part is a function of xn and dn this is a vector xn and this is a scalar E of n both are statistically independent with wn and therefore, vn E of n because E of n depends on dn and xn w is constant here and dn and xn they are statistically independent with wn and therefore, vn ok. So, E of n is statistically independent of vn and therefore, v transpose n so is xn by this assumption. So, xn vector multiplied by the scalar E of n this resulting vector is statistically independent of this. So, expected value of so what does it mean? It means if I have suppose I need some space if I had suppose 2 vectors x and y they are Si

statistically independent then E of x transpose y will be what E of x transpose y mean first component of x which is x1 first component of y which is y1 x1 y1 plus x2 y2 plus dot dot dot maybe x p y p suppose length is 1 to p then E working on x1 y1 will be E x1 E y1 then E x2 E y2 E xp E yp it is same as then becomes equivalent to E of x if you take E of x first that is E of x1, E of x2, E of x3 this vector that transpose E of y, E of x1, E of y1. So, here you have got this vector E of x1 here top guy is E of y1 they are multiplied then next guy this is a row vector because of transposition E of x2 here E of y2, E of x2, E of y2 because statistically independent overall expectation of the product is expectation of x component into expectation of y component alright.

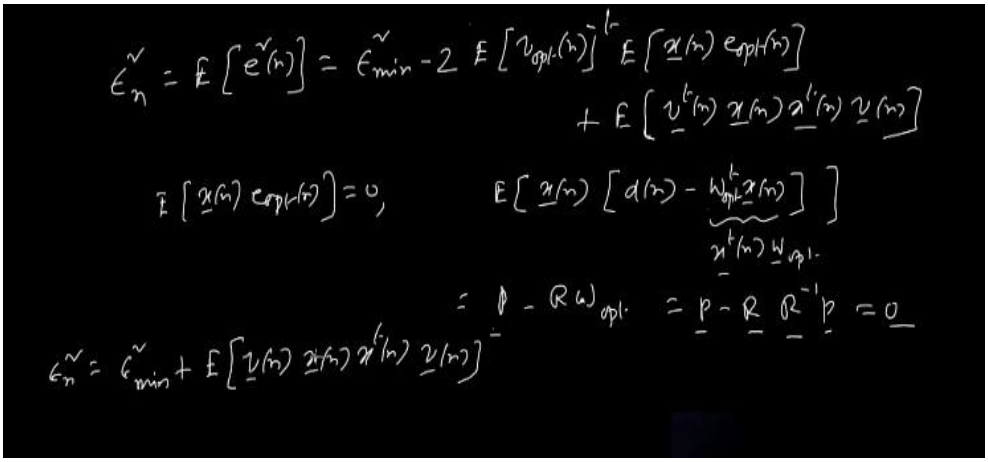
That means, this will be expectation of v transpose n into expectation of this ok, expectation of xn into E of n this we can do. That means, it will be let me call it give a name this the variance this will be having this component minus twice have a look E of this and then E of this. You can do this transpose I think there was transpose. So, you can either apply E on vn first then you take the transpose or first you do the transposition make it row vector then apply E on that either way it will be same into E of xn E of n plus this sorry E of v transpose xx transpose v. Now, what is this term xn E of n we know this should be 0 the vector into scalar.



So, scalar times every component and this is 0 because under optimality condition we have seen the current optimal error it is orthogonal to there is uncorrelated to it each component of xn ok. This again you can see you can write like you know E xn and E of n is dn minus W transpose xn and W transpose xn is same as x transpose n W and then x you write E of xn into dn that is p vector minus E of xn x transpose n W is constant. So, it goes outside E of xn x transpose n there is R and W. So, it is W opt under W opt because it is E of W opt and W opt is R inverse p.

So, p minus R R inverse p. So, R R inverse cancels p. So, it is 0 that this is exactly what we did last time all right. So, these two terms go. So, I am left with this much plus an extra component ok. So, because I did not use optimal weight optimal filter, I use just any arbitrary Wn.

So, overall output error variances gone up from the minimum attainable by a factor this extra factor. Remember one thing by your independence assumption Wn and therefore, Vn that is statistically independent of xn right.



Now, you can see one thing that suppose as an example a side story suppose I give you two vectors ok. A B is one vector may be alpha and I give another vector beta as x y and it is

said alpha beta there is a sign suppose given this is V transpose sorry V transpose and it is given something like this E of like alpha transpose like V transpose alpha transpose then beta say beta transpose and then again alpha. You can do one thing beta beta I mean what will happen is this you have got A B alpha transpose beta beta transpose and alpha.

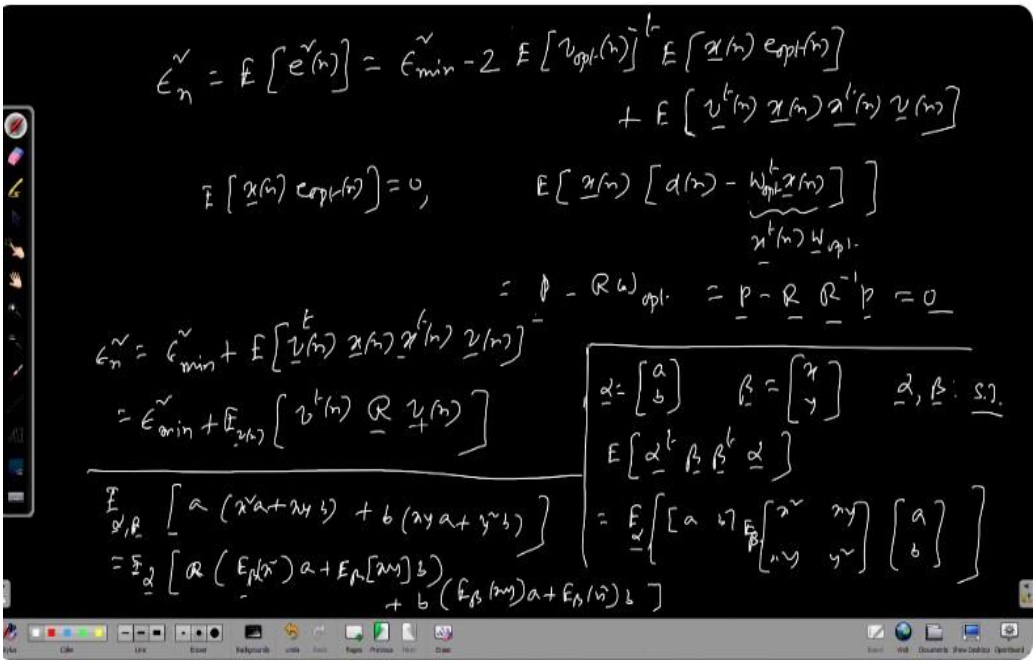
So, you get A B and beta beta transpose will be x y column vector x y row vector it will be x square x y y x there is x y y square right. And now x square into A plus x y into B multiplied by A on that if you apply expectation. So, that will be you know explain because alpha and beta are statistically independent E will get separated it will like you know I mean it will like this. This term E of x square A plus x y B into A plus again B into x y A plus y square B this E will acts separately on x square x y again x y and y square another E will be work on A into A B and all that B into A it will be like that. You will see same thing if I keep one E like you know this is E with respect to both alpha beta.

Suppose I keep E with respect to this alpha out keep it as it is here you apply E beta x square into A ok. So, because E beta is only for x y components. So, it will work out and I could separate out because they are statistically independent E alpha beta is E alpha E beta. So, E alpha beta if I apply on this, I take out E beta overall E over this will be what one expectation with respect to x y that is beta another expectation with respect to alpha A B they will separate out. So, the one with respect to beta I am directly applied here E beta x square A again E beta x y B and similarly E beta x y A plus E beta y square B alright this is what I am doing.

But this is same as though I am applying E beta here E beta E beta x square into A A into E beta x square into A again E beta x y into B into A this is A. So, A E beta x square E beta x square into A plus E beta x y into B E beta x y into B together times A plus B times B times within bracket E beta on x y E beta on x y into A plus E beta on y square into B. I am able to separate out E alpha beta as E alpha into E beta because they are statistically independent ok. So, then I can write like this I mean I can keep E alpha out and push like

this ok. So, on the inner thing this matrix I apply E beta which is a which is you know restricted to the elements of beta and then carry out the product then apply E alpha.

The same thing I can do here because here instead of alpha I have got VN. So, alpha transpose VN transpose and instead of beta I have got xN, but elements of VN and xN they are statistically independent because of the statistical independence assumption. So, same thing here so, which means I can write as E with respect to V if you want V transpose N E working on this and E working those will be input autocorrelation ok. E which is should be x working on this will be what nothing, but input autocorrelation. So, V transpose N or VN this product I have to now carry out which will be a scalar that have to take expectation with respect to VN elements of VN ok.



That will give me this extra component this I will analyze in the next class. Thank you very much.