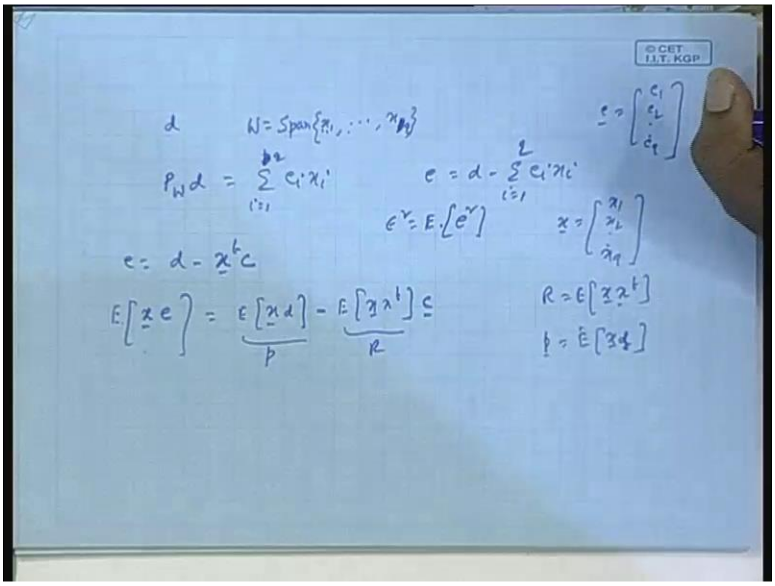
Adaptive Signal Processing
Prof. M. Chakraborty
Department of Electrical & Electronics Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 28
Introduction to Recursive Least Squares (RLS)

We considered, one category of adaptive filters, you know that is the one, which is based on minimizing mean square error, which leads to basically LMS algorithm. There is another category, a very powerful category also, in some sense very powerful not in all sense, that is called recursive least squares. So, we will be now, marching towards Recursive Least Square approach, it is also called RLS abbreviation, so most of time I call it RLS rather than recursive least square. To initiate this, to kind of you know, I mean to introduce you to this topic.

(Refer Slide Time: 01:40)



$d \qquad W = \mathrm{Span}\{x_1, \cdots, x_p\}$

$\underline{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_p \end{bmatrix}$

$P_W d = \sum_{i=1}^{p} c_i x_i \qquad e = d - \sum_{i=1}^{p} c_i x_i$

$\epsilon^2 = E[e^2] \qquad \underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$

$e = d - \underline{x}^t c$

$E[\underline{x} e] = \underbrace{E[\underline{x} d]}_{p} - \underbrace{E[\underline{x} \underline{x}^t]}_{R} \underline{c} \qquad R = E[\underline{x} \underline{x}^t] \qquad \underline{p} = E[\underline{x} d]$

Suppose we consider random variable d, and a set of random variable x 1, dot, dot, dot x p, they are spanning a space, all 0 being done random variable say w. And I want to estimate d, as a linear best estimate, as a linear combination of x 1 to x p, we have seen, that what will that be, that will be nothing but, projection of d, orthogonal projection of d, on the space w. We have the inner product there, inner producer correlation, so for the time being assume all, I mean for this, RLS means this assume everything is going to be real.

Because, complex will be little more difficult, I mean, you can generalize also, whatever result we get for real, we will extend it to, you can easily extend it to, complex case for the time being you assume, that we are dealing with, real value random variables. We are given already, the inner product, in the space as the correlation, between any 2 random variables, e of x y is the inner product between x and y.

That, keeping that inner product in mind, you know if you orthogonally project d, on the space w; what you get is that the best linear combination of x 1 to x p, because the error then is the, one having minimum norm square, this minimum variants, and that error is orthogonal to each of the element, that you all know. That is, you wanted to you want to find out Pw d, Pw is projection operator corresponding to this w, d projected on that, that has some kind of c i x i.

May be instead of p, you replace it to, instead of P, I mean use some other name q, say x 1 to x q, so this is what we do, in our mean square estimation. How is this P w d it is such that, d minus this quantity, the error should be having minimum norm square, and obviously, that will then be orthogonal to x 1 to x q, that some from the theory of orthogonal projection right.

So, there what we do there, we found out the error, error e was, d minus this c i x i, and we said that, due to the variants of this error, Epsilon square, all are real, so just e square is enough. This is the quadratic function of c i n has to have a one unique minima, which it is, which it is, because it is quadratic function also, so find out all the coefficients, those coefficients are obtained as what, in terms of the auto correlation values, of the correlation between x 1 to x q, and the cross correlation between d with each of x 1 to x q.

That is if you define, x vector as, x 1 x 2 dot, dot, dot x q, then you define the matrix R, as E x, x transpose, transpose is enough, additions is not required, because all are real and say p vector as, E x d. You understand, in the optimal filter case we had the variable x n minus 1 up to x n minus q or whatever, instead of that, I am questioning a little more general case, just q, independent random variables q 1 2, independent means, q define variable x 1 to x q, that is all.

But, treatment will be same, the optimal filter coefficient vector will be, R inverse p, is not, how did we obtain that, you knew that, this E is orthogonal to each of the vector, each of the elements x 1 to x p. If you have this, coefficient vector c as, c 1 c 2 dot, dot c q, then, this error e can be written as, d minus, this x vector transpose these are repetition, x vector transpose, c is not it, x vector transpose c vector, this is the summation d minus that is the error.

And how to obtain c, we will we knew, that is the orthogonal projections, so this error must be orthogonal to each of the components of x, each of the random variables. So that means, if you take this, E of, x vector into e, it is a scalar, x vector means, x 1 into e, x 2 into e dot, dot, dot x q into each of the term will be 0. And that give rise to E of, x we replace E by this expression, x into d, minus E of x, x transpose c, this is your R matrix, this is your p vector, and you get that c optimal as R inverse p.

This is slightly more general way of doing things, then that, filter case, in which case these variables are nothing but, x n, x n minus 1 dot, dot, dot x n minus q here. I am just, making it little general, but approach is same, the orthogonal projections, the error as to be orthogonal to each of the components. So, you put the components vector form, each correlation term here will be 0, from that you get this equation, that is c, c optimal is R inverse p, is not it, this was the 1, this was the thing, that we started from that time.

And then we said, that look, R n p, this correlation values, may not be available with me, or may be something, that changes from time to time. So, in that case, my algorithm should be adaptive, so that, I do not have to supply, the system with this information about R n p simply, because the I do not have, that also that that changes from time, to time, it should be made adaptive. So, that from the data only, it adjust itself, so that actual R n p is, we are learnt, in some indirect way, from the data, with that mission we started.

So, we instead of computing this directly by R inverse p, we said that, first we go for an iterative procedure called ((Refer Time: 07:33)). Because, what was the ((Refer Time: 07:36)) procedure, this epsilon square is quadratic function of all the weights, that quadratic function as 1 unique minima, corresponding to the orthogonal projections. We start, doing the arbitrary way vector c, and then going the opposite direction of gradient, and that has bound to, provide you choose the mu, step size that correctly, within a limit, it is bound to converge on, exactly, exactly on the optimal point there, in the ((Refer Time: 07:58)).

((Refer Time: 08:00)) Need, the exact value of R n p, in the gradient computation, remember, the exact value of R n p, but I say it again in the beginning the, R n p are not know to me, that is, why I want to make adaptive. And then, from that ((Refer Time: 08:11)) I took a 1 step ahead, and there I, brought approximations, that to R n p, you replace by wild approximations.

R is simply, see this case x vector times x, instead of taking many such products and averaging, just 1 x vector, which is transpose, p vector also, just 1 x vector into b, instead of taking many such products and averaging. So, I am introducing the approximation there, immediately I am paying a price, what price ,that I can have convergent, subjects to proper choice of mu, but convergence is not exact convergence, it is only convergence in mean.

So, it is not the filter weights will converge really on, the optimal 1, it will be dancing around that optimal 1, that means, that too in a steady state, steady state as n tends to infinity. But of course, we really do not have for, n equal to infinity after, sufficient run, and there is a steady state at that time, after 200, 300 meters usually, and you find steady state, and that time, with this filter coefficients each of the coefficient will fluctuate.

But fluctuation, will be around the optimal value, but actually, converges is taking place, but not absolute convergence, but not convergence mean, not convergence exactly, convergence is only mean, that is price I pay, because I injected those approximation in the beginning, R n p was replaced by very wild estimate. So, number 1 was that, we are not getting absolute convergence, we are only getting convergence in the mean.

Secondly, even if you had not, replaced R n p by those wild approximations, by still carried out ((Refer Time: 09:35)) as a fine procedure, using current value of R n p, supposing know that. If in, that is we say a search technique, that is a search technique we will be, going in the opposite direction of a gradient, and again move further, and then again move further, sometime in a gradient is positive. So, going in the, opposite direction, go backward, sometime when it is negative, so go forward.

So, backward, forward movement, it will take sufficient number of turns, then only you, will converges, you better converge. And then you are going in the approximation, so this, actually the element is nothing but, searching the ((Refer Time: 10:07))) sense, but it is a approximation. So obviously, search means as such it will take time, to find it is optimum value, and then, it is using approximation for R and p, which means very approximate search, that is why it will take lot more time, to reach that convergence stage, and convergence also will be mean, this are the 2 problems in the LMS.

And that, rate of convergence that depends on actually the data statistics, because R and p they related in, R and p they are required R and p. If I mean, I have not thought all this, and I will not teach, just, I can just give you this information, that suppose the input auto correlation matrix. Is such, that the Eigen values, of the auto correlation matrix, there is a huge disparity, suppose a auto correlation matrix is positive definite, so all are positive real, it is always real, because hermesian, but suppose it is a positive definite, so all are positive real and the highest to lowest Eigen value ratio is very high.

In that case, it will take time for more time to converge, on the other hand, if Eigen values are having similar values or same value, it takes less time. So, again in, convergence also depends a lot on, input statistics, this are the problems in LMS, from here, we go to, another class of adoptive algorithm, where will be looking for, exact convergence. That approach will be, free of, it will be basically, in the I mean, it will not be rejecting the statistical notion, that expected value mean variants, but it will not be explicitly using, E operator anywhere.

And as a result, it will give raised to, convergence exactly, no convergence mean, it was a exact convergence, number 1. And also, it will take for less iterations, to convergence, only thing, only price, that you of course pay and because of, which you know, it is not mean, so widely used, even today. Is that, it requires more computation, because we are trying to do for you see, I mean, better range in, that more accurately result faster result. So, naturally more computation, and if you want to look for well assigned implementation, not all forms of RLS is a really very good.

So, LMS still, I would say know, I mean, when there RLS came up, in the middle of 80’s, early 80’s, we all are thought, that RLS will kill LMS, but LMS as survive, but again RLS also is very good. So, let we go to this RLS thing now, recursive least squares, here this is the epsilon square, what is the epsilon square, there is a E operator, variants of the error, this directly the random variable, this is a linear combination of those random variable, this is the error, error is a random variable, you take the square of that, all real, so it is square of, that expectable this mean square.

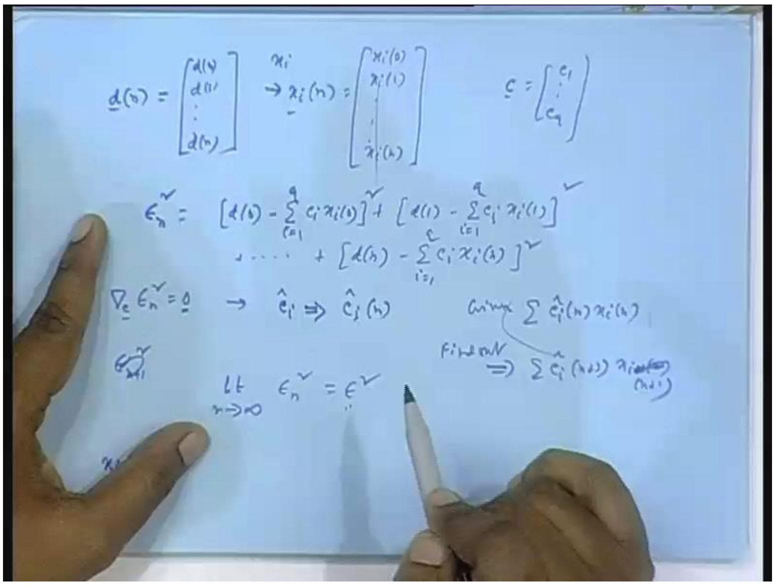
And this, will be this will be a quadratic function of ci and all that, that is how you are proceeding. Now, suppose what am doing is this, I want to find out, an estimate of this parts, what will I do, I will take 1 value of d, 1 experiment, 1 of the observation, 1 value of d, and similarly, 1 value for x 1, one value of x 2, and 1 value for x 3 is, I observed all of them once, find out the error. Again next time observe the another value of d, and again observe the corresponding values here, again find out the error, constant do not change, ci's a same.

So, every time, I find the error, I square the error, and go on adding, sufficient after, a sufficient number of times, 100 times, 150 times, if a average that, that will be a good estimate of epsilon square, but that, average thing will depend on, your choice of this coefficients, is not it. Suppose I minimize, that quantity, with respect to this coefficients, that also will be square function of, quadratic function of this coefficient, what are the after this squaring, this terms, it will still be a quadratic function of, see you minimize you get a set of coefficients.

Point is, if you have, if you take sufficiently large number of such, observation samples, this epsilon square, will be all most equal to, this, and it will have unique minima, because of the, nature of the, mathematical nature of the functions quadratic functions of the weights So, then the minima, point will give raised to, really very close, to the optimal 1, at least intuitively shown, we will expand on this, we really show, we show that is, it need in that case.

Let me now, make it more formal, suppose I am observing d, from time t equal to 0 on words, n equal to 0 onwards. So, it become a time series, it give raised to a, sequence of observation, because d n, d 0, d 1, d 2, d 3, d 4, dot, dot, dot up to the nth index, nth index is the current index, time of sequence, or a record of data, d 0 to d n. Similarly, I observe each of the random variables, x 1 0, x 1 1, x 1 2 dot, dot, dot x 1 n, it give rise to 1 sequence, sequence I put in a vector form.

(Refer Slide Time: 15:08)



$$\underline{d}(n) = \begin{bmatrix} d(0) \\ d(1) \\ \vdots \\ d(n) \end{bmatrix} \quad x_i \to \underline{x}_i(n) = \begin{bmatrix} x_i(0) \\ x_i(1) \\ \vdots \\ x_i(n) \end{bmatrix} \quad \underline{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_q \end{bmatrix}$$

$$\epsilon_n^2 = \Big[d(0) - \sum_{i=1}^{q} c_i x_i(0)\Big]^2 + \Big[d(1) - \sum_{i=1}^{q} c_i x_i(1)\Big]^2 + \cdots + \Big[d(n) - \sum_{i=1}^{q} c_i x_i(n)\Big]^2$$

$$\nabla_{\underline{c}} \epsilon_n^2 = \underline{0} \to \hat{c}_i \Rightarrow \hat{c}_i(n) \qquad \text{Current } \sum \hat{c}_i(n) x_i(n)$$

$$\text{Final} \Rightarrow \sum \hat{c}_i(n+1)\, x_i(n+1)$$

$$\epsilon_{n+1}^2$$

$$\lim_{n \to \infty} \epsilon_n^2 = \epsilon^2$$

So, d n suppose is a vector, it is nothing but, it is started d 0, d 1 dot, dot, dot d n, x in vector, and this vectors are purely column vectors, n indicates the, current index, you start at, I am starting point could have been anything, I am starting at 0. So, then I form a summation, epsilon n square, this epsilon n square is not this, there is no n here, I am defining this is a new, new notation. This will be what, first take the error at 0th index, ci xi 0 this error square.

Then, d 1 minus c i x i 1 square, plus dot, dot, dot plus d n, then you should divide by the total number of terms, but here I am not taking that, because if I minimize this, this is a quadratic function of the weights. This is quadratic, see we are squaring up, coefficients are not changing, my point is, I am line using a set of coefficients, c 0 c 1 c 2 dot, dot, dot and just combining those observation, x 1 to x q. So, at 0th index I got some data for x 1 to x q I combined them, took the error, numerical value of the error square up.

Take another case, at n equal to 1, I got some data for d, some data for the xi, x 1, x 2 dot, dot, dot x q, again combine them with this, took the error, squaring up and going on adding. So, if I have sufficient large number of terms, with that, averaging in mind then, this will be a good approximation of epsilon square, is not it, but now this, suppose I have sufficient terms here, but you see, each term has got square of this, filter coefficients. So, it is a quadratic function of the weights, that will see later also, it is a quadratic function of filter weights.

So, that means, this will be unique minima, and this will give rise, I mean, so if I differentiate this ((Refer Time: 17:48)) each coefficient and all, equate to 0 you will get unique minima. That minima will gives rise to a set of coefficients, those optimal coefficients, that is, if you define c, as I already define it C q, and if you do this, del c epsilon n square equal to 0. Differentiate you can equate to 0, you will get the optimal coefficients, my point is you see, the optimal coefficients c i, should be replace by c i cap, cap is for the estimable, instead of writing opt, here I am putting cap.

Because, already there is a subscript I, so no point in putting comma, and another subscript opt, these are optimal 1, by how is obtain, by solving this equation. You are differentiating, with respect to, c i's equated to 0, you get a sit of set of linear equation, like we did, in that optimal filter case, you know inner filter case, you get a set of. But, there I should now replace, instead of just this as a function of n, you see, I have used here data up to n, and I have form this, sum of square errors, square error, square error, square error, some of square error.

So, many terms, the entire thing is, differentiated with respect to c i's , equated to 0, you get a set of solution, you get solution. Suppose, I make it down epsilon, n plus 1 square, get on 1 more term to this, again the entire thing is differentiated uses to c i's, equated to 0, you will not get this. Simply mathematics say you will not get this, this is a new expression has come, in an overall minimization has to be done, that is why, that will be, that call c i cap, n plus 1.

In the recursively least squares ((Refer Time: 19:40)), you will be giving ci cap n, and the new data, for d n, and each of the variables.

Student : ((Refer time: 19:46))

Data up to n yes so.

Student: ((Refer time: 19:56))

No, no, I am not, I will do that later, I will do let me proceed it, there is a plan by, which I am proceeding. This, taking data up to n, taking data up to n, ci cap n, in the recursively squares, please listen to me, you will be given previous, there is up to I mean the optimal filter coefficients, up to nth index. There is using data up to nth index, whatever best you could get, this will be given to you, and the n plus 1th index data, for d, that is d n plus 1, and for those variables x 1 n plus 1, x 2 n plus 1 dot, dot, dot x q n plus 1, will be given to you.

The new setup data will be given, for the index n plus 1, and the previous estimate will be given, you have to find out, by simple equations, in a recursive way, that you set a weights ci cap n plus 1, that is the entire story, no approximation anywhere. You see, I have removed E operator, no business of E and all these things, that is playing a role in, because after all it is sum of square of errors, it will give rise to, this epsilon square after all, we will approximately epsilon square and all that.

But, when I give this mathematical problem, there is no E operator anywhere, it is a numerical linear algebra problem. It is very interesting and more trick, far more trickier, for than far more challenging than, you gradient directly lattice on that you say this.

Student: ((Refer time: 21:33))

What

Student: ((Refer time: 21:36))

No, no, but before I converge, I have to have the equation, I have to take this ci cap n to ci cap n plus 1 into ci cap n plus dot, dot, dot finally, it will converge. I have to have this recursively, I have to have that, if you are go on doing it every time, by the first principle calculation, you will be finished.

Student: ((Refer time: 22:01))

But, summation was, is not like lattice, that 1 term will give a separate conclusion and let do that.

Student: ((Refer time: 22:12))

Not so easy, not so easy, is not what he is trying to say, is that think that, if you add 1 extra term, the it is contribution can be computed separately and add to, the previous estimate. It is not so easy, not so easy, overall minimization, will lead to some ci cap n plus 1, that will not be shown easily related to ci cap n, it is not so easy, I am looking for exact relations now, no question of any approximation anywhere.

So, we have understood our purpose, the basic purpose of recursively least square is this, or if I revise ci cap n, instead of ci cap n, I may be interesting in projection itself, like the filter output. So, instead of the ci cap n, I may be interested in, combining these terms, using the ci ci cap n and get the projections, so get the projection using, get the like this quantity. Suppose this I know, because ci cap n I know, find out, this quantity find out, given this find out, this also I may be interested, instead of coefficients, I am more bothered about these.

Recursive filter output, or combiner output, you got the philosophy, but 1 thing you see, you can tell me, so far, that you are getting 1 estimate, next time you have 1 data, again do over all minimization, you will get ci cap n plus 1. Then ci cap n plus 2, but what, where is it going, to it will always give new, and new estimate my question is no, because obviously, you understand, that at least theoretically, if I take to infinity you do not have to go up to infinity, take sufficiently large number of n 100, 150 that is fine.

But otherwise theoretically, this is equal to the correct epsilon square, no doubt about it, is not it. So if you take, in this epsilon square, infinite such terms theoretically speaking, again that will be a quadratic function of the weights, and if you minimize that, whatever you get, that will minimize this also, this also and minima that is unique, minima of this is unique, so that means, those coefficients will indeed exactly converge on, the optimal width.

Student: ((Refer time: 24:49))

I am not talking about I plus 1, what is that I.

Student: ((Refer time: 24:57))

Oh sorry, it is n plus 1, thank you, you understood this, the philosophy, theoretically if I take infinite number of such terms, then that is equal to these. With averaging in mind, I will be basically taking, up to finite number of terms, divide by the, number of terms, and laid the number of terms take to infinity, that issue will turn out to be, very large in epsilon square. If I minimize that, with respect to the coefficients, then finally, the coefficients will converge on, those ones, which minimize epsilon square, and that is optimal ones.

But there is no expectation operator, I will take this square and minimize this quantity, so that means, if I now find out a recursive relation, for each I. If I recursively can find out ci cap n, then ci cap n plus 1, then ci cap n plus 2 dot, dot, dot, then that will finally, converge on, the optimal value for ci, exactly not in mean and all that. And also intuitively you can see, it will converge very fast, because no approximation no search, I am not doing any search blindly and all that, is not it.

It should take much less time and also no approximation anywhere, that is why convergence is found to be faster. But computational burden for this is much more than, elimination and also structurally, it will be not so nice, as you have for LMS, for well assigned implementation, those are the things, that you pay for. So, to get into this, let us now again, come back to, the corresponding vector space notions quickly.

If there are some silly mistakes here, there you correct it yourself, tell me please, where is the sign.

Student: ((Refer time: 26:43))

Here this one.

Student: ((Refer time: 26:46))

Each xi goes from xi 0, xi is a variable here, random variable and these are the data samples, these are the data samples for each xi, this is the summation, where is the confusion here. At n equal to 0, I found some really data for x 1, x 2 I call, that x 1 0, that is xi variable giving rise to, that is I have told you, this is giving rise to sequence, a vector. The variable xi, give rise to, when I observe the numerical values, by after trial, after trial, after trial, after trial, after experiment, after experiment, I got values I put them in a vector form, which also a sequence.

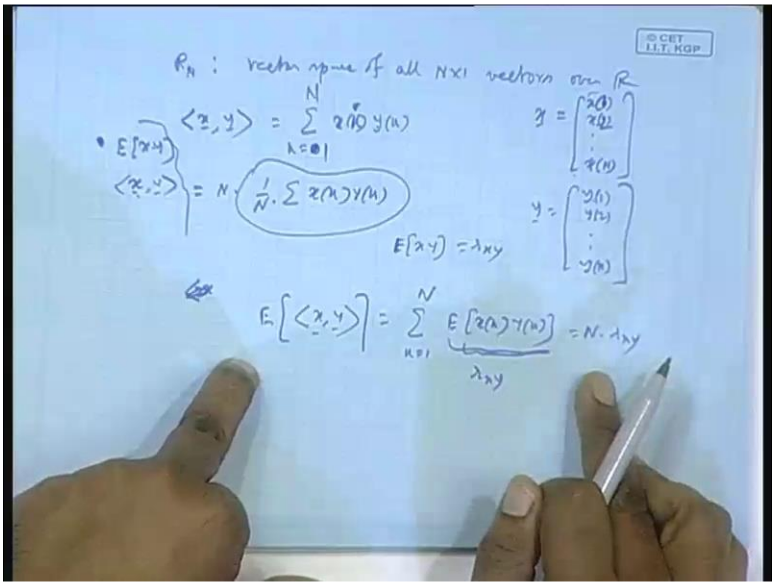
My God, I varies from 1 to q, x 1, x 2, x q these are fixed x 1, x 2, x q, I took any arbitrary xi, I could be 1, I could be 2, I could be q. Any of them is taking the changing from, at n equal to 0 I got some numerical value, at n equal to 1 I got some numerical value, I put into vector form. That n equal to 0, I found find out the numerical value of the error, what is the error, take the value I obtain from d, take the values I obtain from variants xi’s, combine them, that is, this quantity I evaluate at, n equal to 0, I get this.

This quantity, I evaluate at n equal to 1, I get this, I was squaring and adding, that is a good estimate of the, which I am calling epsilon square, of large n, that is a good estimate of these. Is he understood or ((Refer Time: 28:38)) is he understood, please understand this is the basic, if you do not understand that, then entire analyze thing will crash. I is independent, I have got I, how many i's, x 1, x 2, x q, I have taken general xi here.

I have taken general xi here, I could be 1, I could be 2, I could be q, each of them I am observing at n equal to 0 finding out the value of the error, at n equal to 1, finding out the value of the error, squaring of the error adding, that is the good estimate of this quantity, that I minimize with respect to this coefficients. And if I have got, sufficiently large number of data for, each of these variables, so that, the averaging for this is done for, where large number of samples, then that epsilon n square, will indeed turn out be, very close to epsilon square.

If I minimize these, whatever optima values I get, that will be the once, which optima, which minimize these optimal once, is not it, you understood these, this is basic, this is the basic thing in analyze, is this clear to all, ((Refer Time: 29:50)). So, now you in order do everything in a vector space frame work, let us now define the corresponding vector space, or whether inner product space, I have dealing with real valued, data here, absolutely.

(Refer Slide Time: 30:05)



© CET I.I.T. KGP

R_N : vector space of all $N \times 1$ vectors over R

$$\langle \underline{x}, \underline{y} \rangle = \sum_{n=1}^{N} x(n)\, y(n)$$

$$\underline{x} = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N) \end{bmatrix} \qquad \underline{y} = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}$$

• $E[xy]$

$$\langle \underline{x}, \underline{y} \rangle = N \cdot \frac{1}{N} \cdot \sum x(n) y(n)$$

$$E[xy] = \lambda_{xy}$$

$$E\left[\langle \underline{x}, \underline{y} \rangle\right] = \sum_{n=1}^{N} \underbrace{E\left[x(n) y(n)\right]}_{\lambda_{xy}} = N \cdot \lambda_{xy}$$

So, suppose I consider R, actually it is called script R, it is so real, R n is a vector space of, all N cross 1 vectors, and these vectors are actually column vectors, N cross 1 vectors means, really column vectors, over R. That means, each entry of the vectors, is a real number, you understand, that this is a vector space, how to add two column vectors, sampalize, that you get still a column vector of dimension same, it remains in R n, that is a quickly.

R n is a vector space, that we may know very trivially it is obvious, how, consider any 2 vectors of length n, you add them, you still get a vector of length n, closed under addition. Then a vector, I mean, R n has got a vector all 0, so 0 vector is here, because if you had any other vector with that 0 vector, you get that original vector, with every vector there is negative of that also. Because, to every LMS you change the sign, you get the negative of that, you add that 2, you get the 0 vector, all that accessibility compatibility everything is valid.

Take any column vector, multiply by scalar, real scalar, you get still a vector of length n, and all other properties are valid there, it is a vector space, it is a vector space of dimension N. How I showed that, is an example, you can consider a trivial basis, for this 1 is 10000 then 01 0000 then 001 000, how many you get n such vectors. They are linearly independent, and this span by entire thing, because any n cross vector can be optimized by the linear combination of those trivial basis.

That means, this is span by the trivial basis, but total number of vectors in trivial basis is capital N, so dimension is N, once I know the dimension finished, I do not care for the trivial basis, all this is repetition, this I this I took example, of vector space R n, it is dimension is N. Here I define, inner product, between any 2 vectors x at y, as what, that is if x is, say x 0 sorry, x 1, x 2, x n, that the inner product is simply sample wise multiplication and addition.

Xi or may be x k not i, y k, k from 0 to sorry, k from 1 to starting this 1, one to N, just sample wise multiply and add, this will satisfy all the properties of inner product first you can see. X 1 plus x 2 comma y, here also, x 1 k plus x 2 k into y k, you can separate out linearity valid, x n y and y n x in this case they are real, so they are same here also, x k, y k, y k, x k no change, c times x into y, c x, k c can go out. And x with itself means, summation mod x square, all are real, so x square means, mod x square actually, because x is real.

So, that is always real non negative, and equal to 0 only, if each x square is equal to 0, because nth 1 is x square, each x square is 0 means, x k 0, that means, all the components are 0, so it is a 0 vector, fine it is enough. This is a most standard inner product, but remember one thing, if this x, corresponds to a random variable does so, there is a random variable x, and I am observing, capital N number of some data, N, capital N num number of times, I am observing it.

And I am, putting all the data, in a vector form, are you observe this, random variable x1's get x 1, next time x 2, next time dot, dot, dot, next time x n, put them in a vector form, same for y. Then, this inner product x y, it is, N times if I write 1 by N, this is a good estimate, of the correlation between x and y, suppose you want to find out, that you have given me 2 random variables x and y, E x y you want to find out, E x y you want to find out, x is a variable here, not vector, x is random variable, I am not putting it in a underscore.

X is a random variable, y is a random variable, E x y you want to find out, a good estimate of that will be what, you observe x and y, for various in indices, at N equal to 1, N equal to dot, dot, dot, N equal to capital N, multiply samples wise and add an average, this quantity. So, just this inner product is nothing but, capital N times, for large number of cases of course, is a good estimate of capital N times, the correlation, so In fact, if you take this, and divide by 1 by capital N this will go, and this is a good estimate of the correlation.

If it is x comma x, it will become capital N times, good estimate of variance, please see this, at in any case, this E x y, if I now take this quantity, as it is, and I am saying, as though x is not just numerical data it is a sequence. So, every time, that it is a sequence actually, every time you observe, you get 1 value for this, 1 value for this dot, dot, dot, 1 value for this, we viewed that way, it is another view point.

Similarly, 1 value for this, 1 value for this, 1 value for this, next time you observe, this sequence changes, this sequence changes, in that case, inner product value will change from observation to observation. One observation means, one set of value here, one set of value here, if you see that way, in that case, expected value of, where inner product is this inner product is changing, data changes from case to case.

One observation, 1 set of value, 1 set of value, 1 value of the inner product, next set of val[ue]-, next observation, 1 set of value, 1 set of value, another value of inner product, see its random, now expected value of, that will be what, E x k y k. But, E x y if I call E x y as R x y, then this quantity, where there is k equal to 0, k equal to 1, k equal to 2, this quantity will always be, R x y. So, this is N times R x y, so this inner product is such, so if I define by capital N, if this 1 by N, this gives rise to R x y.

So, this inner product is such, that mean value of that inner product, indeed gives, that is if you really you are bring in expectation operator take the mean of that and mean value of that divide 1 by N, indeed gives the actual correlation. This kind of estimates are called, this form of estimates theory unbiased estimate, for a while for the time, being little me tell you something about unbiased estimate.

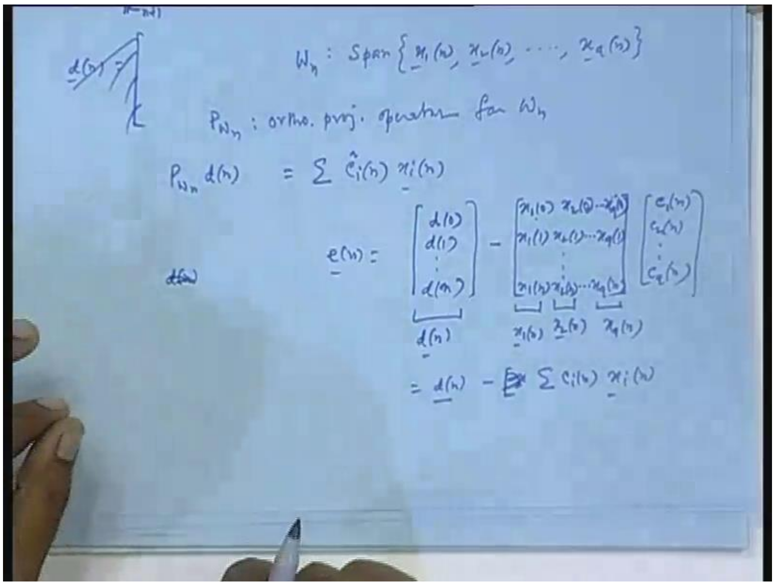
Suppose there is parameter theta, you are observing, you want to estimate theta, you are observing lot of data, what theta, using that you are finding out some theta cap, by some algorithm. Again next set of observation comes, your theta cap value changes, again another set of observation comes, observing some data you know other set, again theta caps comes, your algorithm working on data set give you something. So, your estimate itself is random, because that is dependent on the data set.

But the theta cap is varying, but if the expected value, mean value of that theta cap turns out to the same as the original theta, which is being estimated, then it is called unbiased estimate. It is very, very, very crucially in estimation theory, whether estimate is unbiased or not, sometimes theory by theoretically you can find out that, estimate is not unbiased, but there is a dc component added. Which is also for us, dc can be subtracted dc fixed, you know, non deterministic, I mean deterministic quantity, non random quantity added.

That is expected value of theta cap is not just theta, theta plus some c, where c is a non random known constant, you just subtract out just see, you can get your, at least from expected value of theta cap, you can get yours theta. But sometimes, that quantity is either unknown or even random, not random unknown yes, after expectation, it cannot be random, so it is unknown, then there is a problem.

These are something I told you from ((Refer Time: 39:01)) theory, so these is the inner product thing, I have. So, using this notion, and I already told you, what I do in recursively least squares, the sum of square thing, this I will be minimizing right, this I will be minimizing, now you see, how I will ((Refer Time: 39:15)) how, I will write it.

(Refer Slide Time: 39:24)



W_n : Span$\{\underline{x}_1(n), \underline{x}_2(n), \ldots, \underline{x}_q(n)\}$

P_{W_n} : ortho. proj. operator for W_n

$$P_{W_n}\, d(n) = \sum \hat{c}_i(n)\, \underline{x}_i(n)$$

$$\underline{e}(n) = \underbrace{\begin{bmatrix} d(0) \\ d(1) \\ \vdots \\ d(n) \end{bmatrix}}_{\underline{d}(n)} - \begin{bmatrix} x_1(0) & x_2(0) & \cdots & x_q(0) \\ x_1(1) & x_2(1) & \cdots & x_q(1) \\ & \vdots & & \\ x_1(n) & x_2(n) & \cdots & x_q(n) \end{bmatrix} \begin{bmatrix} c_1(n) \\ c_2(n) \\ \vdots \\ c_q(n) \end{bmatrix}$$

$\underline{x}_1(n)$ $\underline{x}_2(n)$ $x_q(n)$

$$= \underline{d}(n) - \sum c_i(n)\, \underline{x}_i(n)$$

Suppose I have got a vector d n, and now I will be, in these vector space R n plus 1, that is vectors of length, n plus 1, starting from index 0, to current index n. So, length will be n plus 1, I mean this vector space over real ((Refer Time: 39:40)), so I define d n vector as, In fact, I have already defined, I have already defined them. So, no point in rewriting, I take these definitions, this is vector belong to R n plus 1.

So, suppose, I now consider, a space w, may be w n I proved, span by, this column vectors, find you column vectors x 1 n, x 2 n no longer random variables, no random is no expectation, no random nothing now, from now onwards, this up to the pure and simple linear ((Refer Time: 40:27)). But relation of, of course, into the statistical relation was there, otherwise, you can say that, why, why, where will be this exercise, the factor usually converge to the optimal 1 they are that, stochastic thing comes on, that we have already elaborated, to elaborated.

X q n, I am assuming this, vectors to be linearly independent for the time being, that also, has to be seen and verified, you will see that this vectors are not always linearly independent. The all those things will come up, but for the time being, assume they are linearly independent, conditions required to the, for they to linearly independent, that is satisfied. Suppose and w it is a space, and p w n is the orthogonal projection operator, ortho projection operator for w n.

Then my claim is, whatever we are doing here, is nothing but, the what we did, we found out this error and minimize this, with respect to the coefficient and what this. What the optimal coefficient, which have function of n, my claims is nothing but, computing this as, linear combination of this elements. That is computing this, that is I am doing orthogonal projection of d n, on the space span by this, I will very I will having verify.

Now only listen, d n vector, suppose I project it orthogonally on this space, span by this, so that projection I define like this, notation only, which will be a linear combination of this. My claim is, there you will get the same coefficient as you get by, solving this equations, that is minimizing this, with respect to this coefficient, that is easily seen also, that is, min this my claim, you will get the same coefficients here. Obviously, because suppose I want to compute the projection, suppose I want to compute the projection, a projection of what, d n on the space, span by them.

So, that means, I want to find out, what may be I, do it in a other way, you are minimizing this sum, that time we are minimizing this sum. But, can I not write it like this, that, this visible, d 0, I am using the subscript n, because coefficients after all will depend on the index n, because I am using data up to index n, if it is call n, if it is n plus 1, and n plus 2 new coefficient will come, only for that. I am using this notion.

Suppose I consider this first, d 0 minus summation of this terms c 1 n times x 1 0 c 2 n, that is what you have here. This quantity is nothing but, first this is a reality is a vector, column vector minus matrix into column vector, this is totally column vector, Column minus column, column vector, where is the first element on the column vector, this quantity only, is not it. d 0 minus, you are combining them, instead of ci, I am putting ci n here, because there is more correct, to bring the depend dependence on the n explicitly.

So, ci a, c 1 n times, the data for x 1, c 1 n times the data for x 2 dot, dot, dot, cq n times data for x q, this n only indicates that, final coefficients, which will obtain, which will be obtain by minimizing, I should, make it dependent on n, because I am taking data up to n. If instead of n, I have 1 more data, I will get new set of coefficient, so that is why, I should, show the dependence on n explicitly, then d 1, that is this quantity, is not it, and dot, dot, dot d n current index last quantity.

So, you get a vector, first element is this, second element is this dot, dot, dot third element is last element is this, if I take the norm square of the vector, square of first element this, square of second element dot, dot, dot square of last element, add it as for my inner product definition. So, minimization of this means, minimization of the norms square, and what is this quantity, I told you, how to carry out matrix vector multiplication.

C1 n time this column ,this column is x 1 vector n, this column is x 2 vector n, this column is x q n, and this column is d n. So, this error vector, if you call it, e n vector, is nothing but, d n vector minus, we can write this way, ci n times the corresponding vectors. That is d n minus a linear combination of, those x 1 to x q, we found the error, that error vector norms square is this, squares some of square errors, that you want to minimize means, we are finding out the orthogonal projection, because we are minimizing norms square.

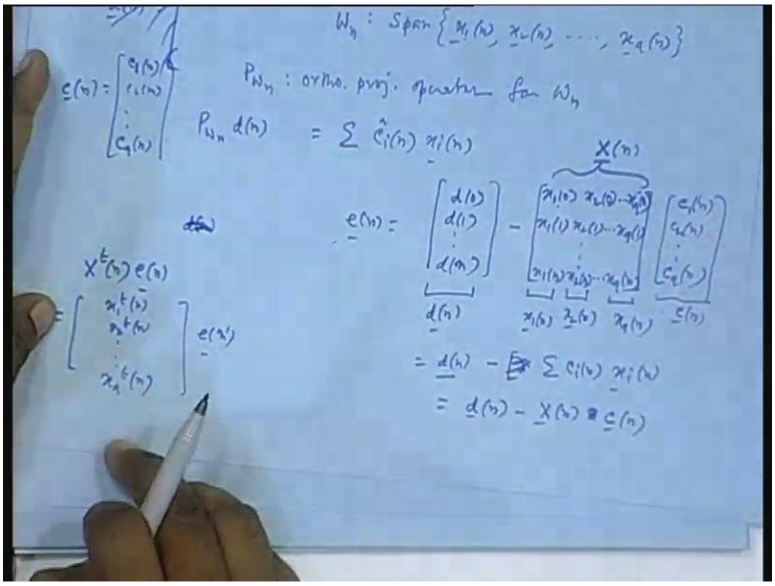
Norm square in the new vector space R n plus 1, but you are ((Refer Time: 46:57)) minimizing the norm square. Are you getting me, this minimization of this, this nothing but, this is very simple actually, this is nothing but, this quantity, which is nothing but, that is, this error, take each term of the error square up. First term, second term, last terms square up and add, there is a take the error vector take the norm square of that, your are minimizing that norm square error, with respect to the coefficients.

So, we are minimizing the norm square of this, with respect of linear combination of coefficients, from orthogonal projection theory, this norm square. Where that orthogonal projection theory, independent of vector space, particular type of vector space, that is generally. So, again in the that projection will be exists will be unique, norm that minimum norm square solution will come up, which will corresponds to the orthogonal projection, the error will, this error will be orthogonal to each of this terms, this error will be orthogonal to each of this terms.

And will be that error b then, see you are understood, this is the orthogonal projection problem, projecting d n on the space orthogonally ((Refer Time: 48:13)) projecting orthogonally d n, on the space w n. So, there projection will be a best linear combination of this column vectors, those are the coefficients, this real thing error vector norm square will be minimize, because of this guys.

So, what are these coefficients we had already find out, we know in the case of optimal filter R inverse p and all, but that entire business is out of the pavilion, no e operated anywhere and all that. We can carry out a similar treatment, you are telling me, e n is orthogonal to ((Refer Time: 48:41)) and this vector, this matrix you permit me to call, X n capital X n fine. So, this equivalent I can also write as, d n minus c n.

(Refer Slide Time: 49:02)


W_n : Span$\{\underline{x}_1(n), \underline{x}_2(n), \ldots, \underline{x}_q(n)\}$
$\underline{c}(n) = \begin{bmatrix} c_1(n) \\ c_2(n) \\ \vdots \\ c_q(n) \end{bmatrix}$
P_{W_n} : ortho. proj. operator for W_n
$P_{W_n} d(n) = \sum \hat{c}_i(n) \underline{x}_i(n)$
$\underline{X}(n)$
$\underline{e}(n) = \begin{bmatrix} d(0) \\ d(1) \\ \vdots \\ d(n) \end{bmatrix} - \begin{bmatrix} x_1(0) & x_2(0) & \cdots & x_q(0) \\ x_1(1) & x_2(1) & \cdots & x_q(1) \\ & \vdots & & \\ x_1(n) & x_2(n) & \cdots & x_q(n) \end{bmatrix} \begin{bmatrix} c_1(n) \\ c_2(n) \\ \vdots \\ c_q(n) \end{bmatrix}$
$\underline{d}(n)$ $\underline{x}_1(n)$ $\underline{x}_2(n)$ $\underline{x}_q(n)$ $\underline{c}(n)$
$= \underline{d}(n) - \sum c_i(n) \underline{x}_i(n)$
$= \underline{d}(n) - \underline{X}(n) \underline{c}(n)$
$X^t(n) \underline{e}(n)$
$= \begin{bmatrix} x_1^t(n) \\ x_2^t(n) \\ \vdots \\ x_q^t(n) \end{bmatrix} \underline{e}(n)$

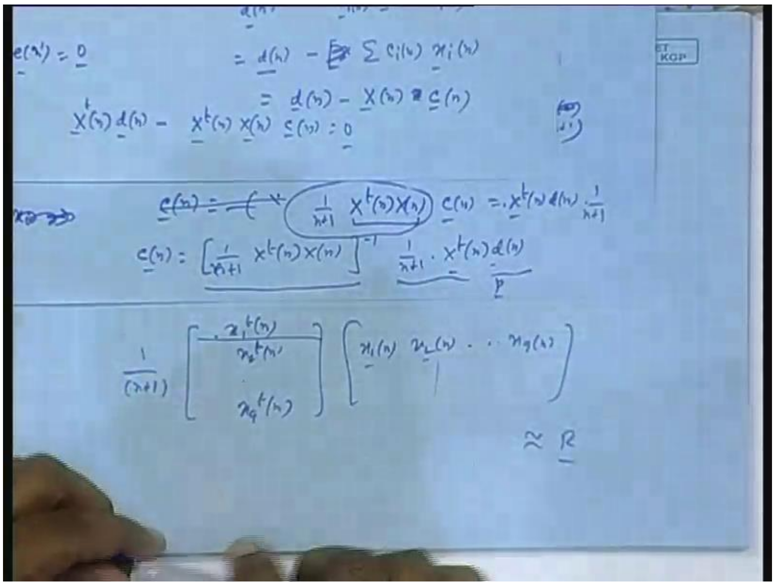
c n is, I have already defined as earlier, c n con consists of those c1 n, c2 n dot, dot, dot c q n, this coefficients I have put in a common vector form. So, x n into c n ,this is c n, this is a c n this entire minus fine, this error is orthogonal to, each of the fellows, fine, that means, X transpose n en that should be 0, what is X transpose n, x n first column is this vector. When we make it transpose, this comes of here on top, as row vector, what is X transpose n, first column and next class please come with this preparation, otherwise very soon you will be, totally lost actually with this topic.

I am telling you this, far more difficult, than whatever you studied so far, mathematically far more challenging exiting and so difficult, and this things are used in not only in single positional error filter, most importantly you are learning something, which is useful in many context today. So, generally of the thing, I pick up things in this course, which are used elsewhere also, this recursive squares taking it is and all those. It is not that you know, that quadrature active filter, so you do not take much into not that.

So, X transpose n, X transpose n, n means first row, first column, very simple this x1 n, so this it becomes the row. x 2 transpose n dot, dot, dot times e n vector, and what is this x 1 transpose e n, what was our inner product, this inner product, is what, this I can write, with x here y, here I can write x transpose y or y transpose x, sample wise multiplication addition, x transpose y, x 1 into y 1, x into y 2 dot, dot, dot, the two vectors just one column, that is what is happening here.

This fellow after transposition has become row, row times en, that inner product has to be 0, because en is orthogonal to, orthogonal directly, not in random session all that, en, because they are spanning the space, en is orthogonal to this guy. So, either en transpose this column or this column transpose en both is 0, then same for x 2, same for x q, so this is 0, 0 vector, directly exactly without any expectation of at anywhere, fine. But, now en is, this quantity you substitute, en is this quantity, so now substitute, X n d n vector sorry, X transpose n, X transpose n d n vector, X transpose n, en means this quantity.

(Refer Slide Time: 52:24)



$\underline{e}(n) = \underline{0}$

$= \underline{d}(n) - \sum c_i(n)\, \underline{x}_i(n)$

$= \underline{d}(n) - \underline{X}(n)\, \underline{c}(n)$

$\underline{X}^t(n)\, \underline{d}(n) - \underline{X}^t(n)\, \underline{X}(n)\, \underline{c}(n) = \underline{0}$

$\left(\frac{1}{n+1} X^t(n) X(n)\right) \underline{c}(n) = \underline{X}^t(n)\, d(n) \frac{1}{n+1}$

$\underline{c}(n) = \left[\frac{1}{n+1} X^t(n) X(n)\right]^{-1} \frac{1}{n+1} \cdot \underline{X}^t(n)\, \underline{d}(n)$

$\underline{P}$

$$\frac{1}{(n+1)} \begin{bmatrix} \underline{x}_1^t(n) \\ \underline{x}_2^t(n) \\ \vdots \\ \underline{x}_q^t(n) \end{bmatrix} \begin{bmatrix} \underline{x}_1(n) & \underline{x}_2(n) & \cdots & \underline{x}_q(n) \end{bmatrix}$$

$\approx \underline{R}$

So, d n vector minus c n is zero, which means c n is, so that means, I can write like this, X transpose n, X n into c n, as x transpose n d n right, this quantity, if I, divide this by 1 by n plus 1, I divide this also by, 1 by n plus 1. What is this quantity, and this quantity, is a matrix after all, X transpose n is a matrix, X n is a matrix, where everything is a matrix, just a scalar time, every element divided by a scalar, say entire thing, so what is c n, you can ignore this also, but I am putting it for a specific reason.

This 1 by n plus 1 and this will cancel, after all you see, 1 by n plus 1 is not required here, so c n will be just, inverse of this matrix, times x transpose d n. Actually, when I use it later, I will not be bring the 1 by n plus 1 they cancel, two things I important, next class I will show this is, not always invertible, but I am assuming it to be invertible will see, when it to is invertible, it is not always invertible, those exam conditions has to be same, but I am assuming, that we are in a condition, where this is a invertible.

Secondly, what is this quantity, X transpose X means what, very quickly, first row is x 1 transpose n, x 2 transpose n dot, dot, dot, x q transpose n, and here x 1 n, x 2 n dot, dot, dot x q n. First one will be, x 1 transposes x 1, norm square of the vector, divided by n plus 1, for large n, it will be a good estimate of, the auto correlation variances of that random variable x 1. Then, x 1 transpose x 2,it will be a correlation, between x 1 and x 2, for large n, so this entire quantity, for large n will approximately equal to R.

This quantity will approximately equal to X transpose into d, so they have got this, not projected only d n, so x 1 transpose d n, correlation between x 1 and d, x 2 transpose d n correlation between x 2 and d. So, this will become p approximately, and you will get an inverse p, so this solution approaches, that optimal solution for large n, that is all for today.

Thank you, please come prepare with all this next class.