Introduction To Adaptive Signal Processing

Prof. Mrityunjoy Chakraborty
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture No # 29

NLMS Algorithm

So, in a previous class we discussed this normalized NLMS algorithm NLMS. I will not go through that procedure again, but this is the update equation basic update equation we obtained. En xn vector into en divided by a scalar which is a norm square of this. These are we had obtained. Now, some modifications of this firstly so, some modifications.

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

$$\left|\left|\underline{x}(n)\right|\right|^2$$

This norm square if it is suppose very small that is xn has very small valued data and therefore, this norm square is very small that time we may result in something like a division by 0 situation because after all it will be you know in computed there will be finite precision system. So, it might just be truncated to 0 and then we might have a division by 0 like situation. So, that is why instead of this we replace this by this plus one small cos constant small positive constant. Since it is small it is not going to change the algorithm much hardly has any effect, but it will ensure one thing the denominator will never be 0.

So, that division by 0 thing will not work that is one modification. Another is we have seen we have done everything without assuming I mean assuming there is no noise. But suppose the system is like this again I am taking just two coefficients for our explanation. So, this is your xn and some noise comes which is zn and this is dn. Let this be yn, but yn is not observable to us what is observable is dn what is whatever is coming out of the system which is mixed with noise.

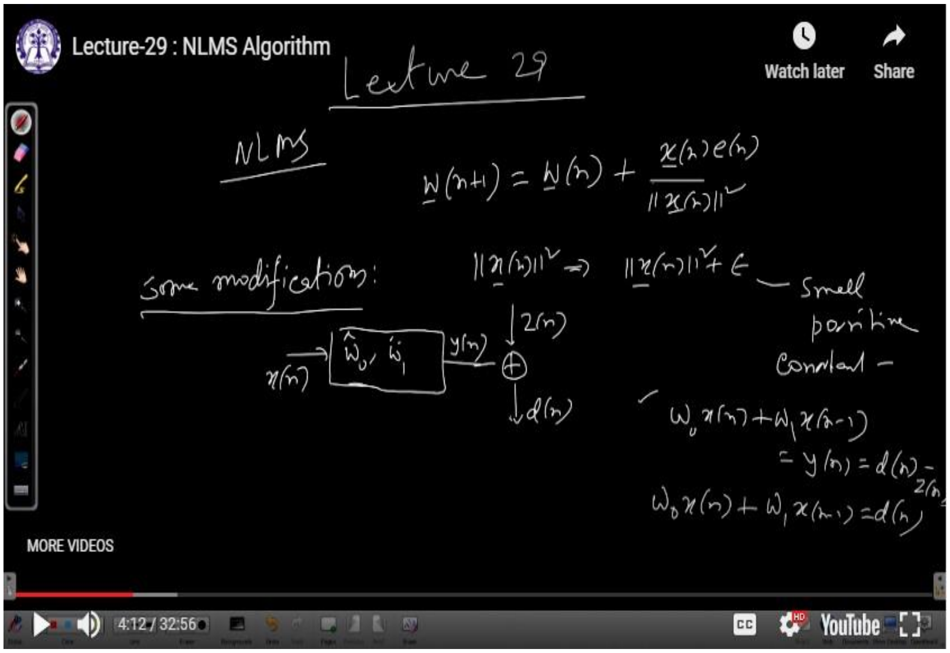
So, actual equation will be I mean since we do not know these coefficients let us assume general weight variables w0 xn plus w1 xn minus 1 that should be yn then you know it will be a straight line and all the previous treatment will work, but yn is not observable to us.

$$w_0x(n) + w_1x(n-1) = y(n)$$

So, if we still carry out the previous method that is we consider this hyper this straight line and in a more general sense we have got not just two coefficients maybe n number of coefficients it will be hyper plane not just a straight-line hyper plane. Like if you have three coefficients it will be just a plane in a three-dimensional world and like that more than higher the dimension, we cannot call it three-dimensional plane and all that is called hyper plane. So, in the case of two dimension that plane turns out to be a straight line as we have seen earlier. So, if I still continue with the previous method that is equal to dn then there is a problem this is not the actual equation actual equation is this.

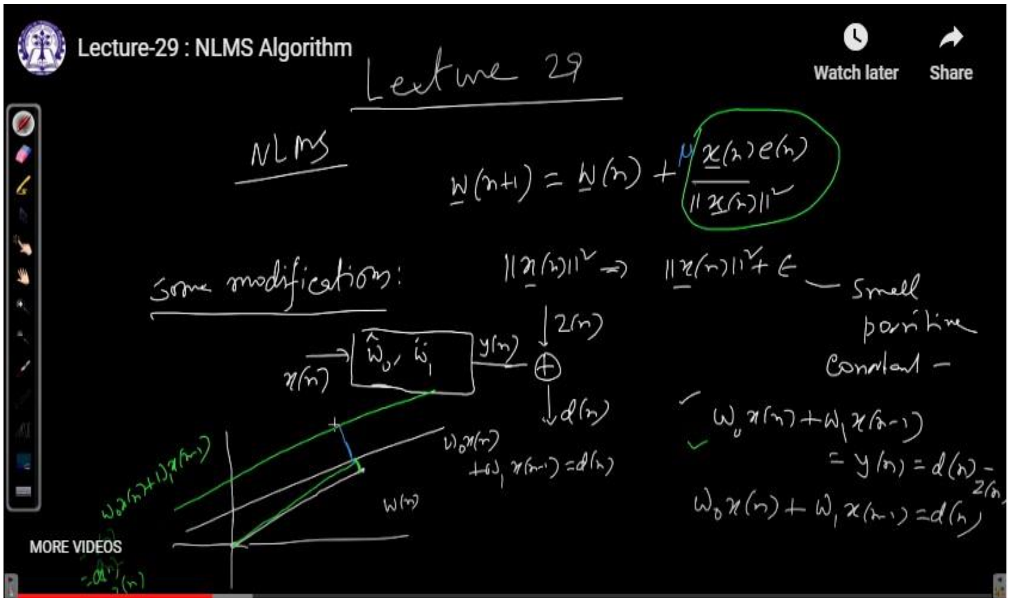
$$w_0x(n) + w_1x(n-1) = d(n)$$

So, that is, but this yn is dn minus zn, zn also is not known to us that means what we are doing so long this was we took this to be our equation the dn what is observable and this was my wn.



So, I was just projecting it on this, but this is not the correct straight line. So, we actually found out this much, this is what is this, that is wn vector plus this will give you this vector which we call wn plus 1. But remember this is not the correct straight-line equation correct straight line is this one what is the difference between the two see gradient remains same w0 xn w1 xn minus 1. So, gradient will be given in terms of xn and xn minus 1 here also xn and xn minus 1.

So, they are parallel lines, but one line I mean they cut this y axis the intercept on the y axis is different alright. So, they will be parallel. So, another line will be this parallel line this is the correct line that is w0 equal to yn which is dn minus zn. Therefore, I should extend this to this, if it is cutting on the higher side which means, in general you will not take this we will have a constant mu brought here which will multiply this try to make for this much. Now you can ask a question that how do you know that this parallel line will be above this.



So, I have to expand this much is not it. So, mu is positive it could have gone the parallel line could have been below because it just depends on zn zn is noise sometimes positive sometimes negative. So, in that case we have to project on this side opposite of this. So, that actually should be negative that way. So, therefore, the mu that comes actually is just a mu which statistically minimizes the you know error between the actual update and this update this much and this much this was the error.

So, it is just for one case. So, statistically mu will try to minimize this that is why this is not that it will exactly hit always this because you have chosen mu can be something else, but on an average, it will be close to the correct update this much is the correct update on an average. So, with this this equation then becomes this is the NLMS update equation.

$$\underline{w}(n+1) = \underline{w}(n) + \frac{\mu}{\left\|\underline{x}(n)\right\|^2 + \epsilon} \underline{x}(n)e(n)$$

Now, since we have brought in mu because there is noise and mu cannot be you know we cannot give a I mean if we have to be correct at every case, we have to give the corresponding mu like here the ratio between this whole thing and whole thing you know that should be mu. So, that when you multiply this by mu I get so much, but that is for one case, but here I am choosing a constant mu.

So, since I am choosing a constant mu obviously, there will be some error there will be some error I will not get the exact update I will get in the same direction, but a fractional update and that is why it will no longer converge absolutely when there was no noise in the previous example you have shown those diagrams you go on projecting Wn get Wn plus 1 then Wn plus 2 and all that eventually you will be hitting the optimal weight exactly. But the moment there is noise present in the system and you bring in a mu, but mu cannot be I mean ideally mu cannot be a constant mu depending on case-to-case mu should vary, but we are not doing that therefore, it will not converge absolutely, but it will again converge in mean average expectation that is expected value of Wn will go to W opt not Wn will go to W opt as n tends to infinity. Those proofs I am not showing that is here again sorry will go to W opt this can be proved.

$$\lim_{n\to\infty} E[\,\underline{w}(n)] = \underline{w}_{opt}$$

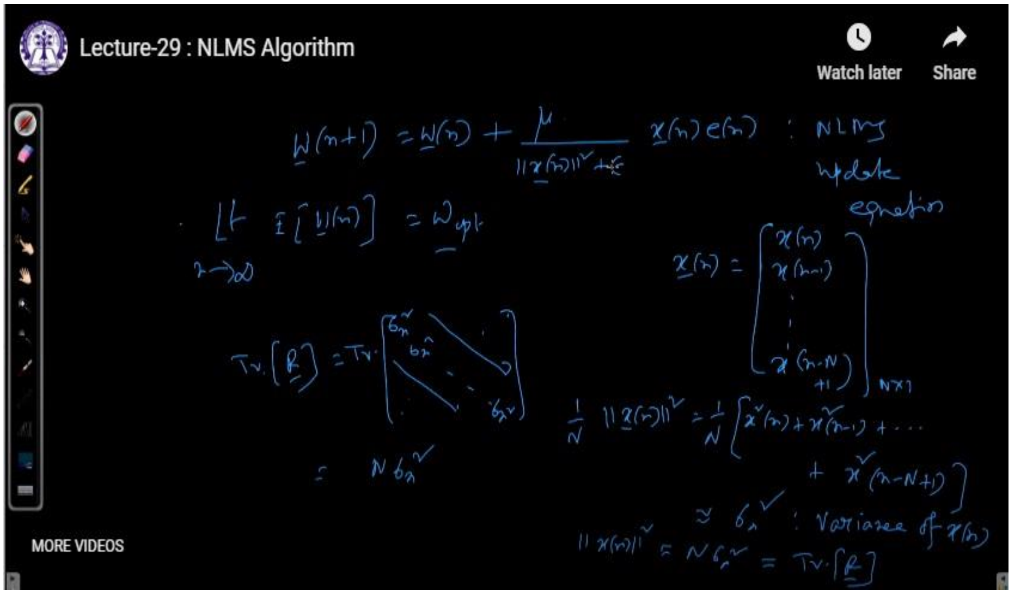
But we can intuitively argue this way what is norm xn square xn is a vector x of n x of n minus 1 dot dot dot I do not remember which order I took if I say n plus 1. So, it is capital N cross 1 vector. So, norm xn square is x square n plus x n minus square varying is term and adding and now if I suppose divided by 1 by n right hand side you see what is that it is a estimated variance of xn because I am squaring up every sample adding over a large number of sample and averaging.

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

$$\frac{1}{N}\left\|\underline{x}(n)\right\|^2 = \frac{1}{N}[x^2(n) + x^2(n-1) + \cdots + x^2(n-N+1)]$$

So, it is approximately sigma x square variance and xn is Wss wide sense stationary. Now, what is trace R? R that is trace of R matrix R matrix is what it is a top needs matrix whatever element you have one diagonal that will continue and that is the variance here Rxx0 correlation with gap 0 which is a variance and same will continue. Similarly, other terms dot dot dot dot right trace of this will be summation of the diagonal elements. So, it will be there are capital N of them. So, it will again be N sigma x square right.

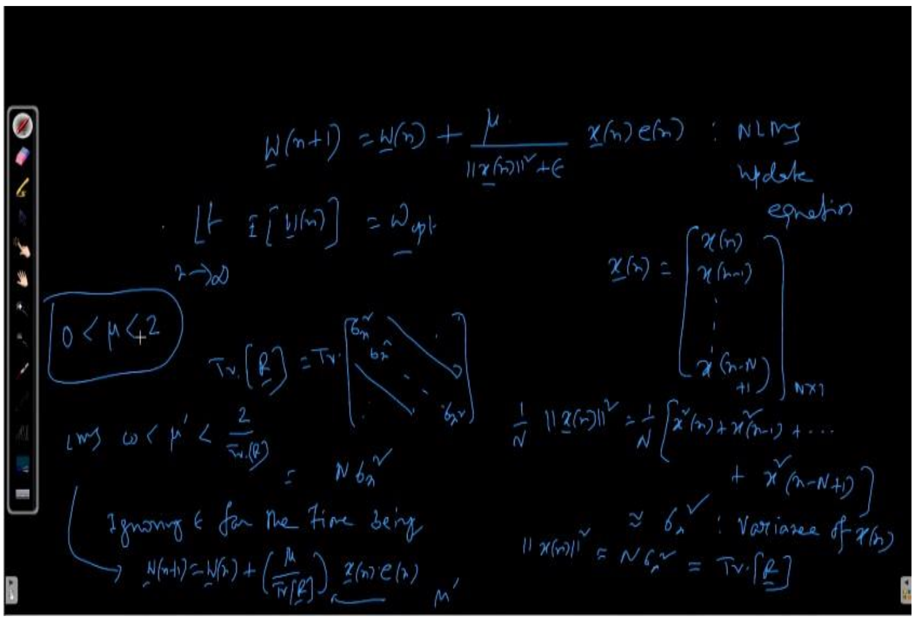
From here I find norm xn square is capital N sigma x square which is same as trace R. Therefore, if we ignore this epsilon for the time being because it is so small ok. Then for the time being because it is very small we will have Wn plus 1 as Wn plus mu by norm xn square which is trace R this much into xn En. If I call this bracketed quantity as mu prime then it is simply LMS algorithm is it not just Wn plus mu prime this is a constant mu constant trace R constant. So, this is constant some mu prime times are all positive.



So, positive constant times xn En. Therefore, its convergence will require mu prime greater than 0 as you have seen less than 2 by tracer. But mu prime is mu by tracer if you replace mu by tracer tracer and tracer cancels. So, you get this condition there is a mu which we use here ok. So, this shows that if mu is chosen between 0 to 2 then this algorithm will converge in bin because it turns out to be a LMS algorithm ok.

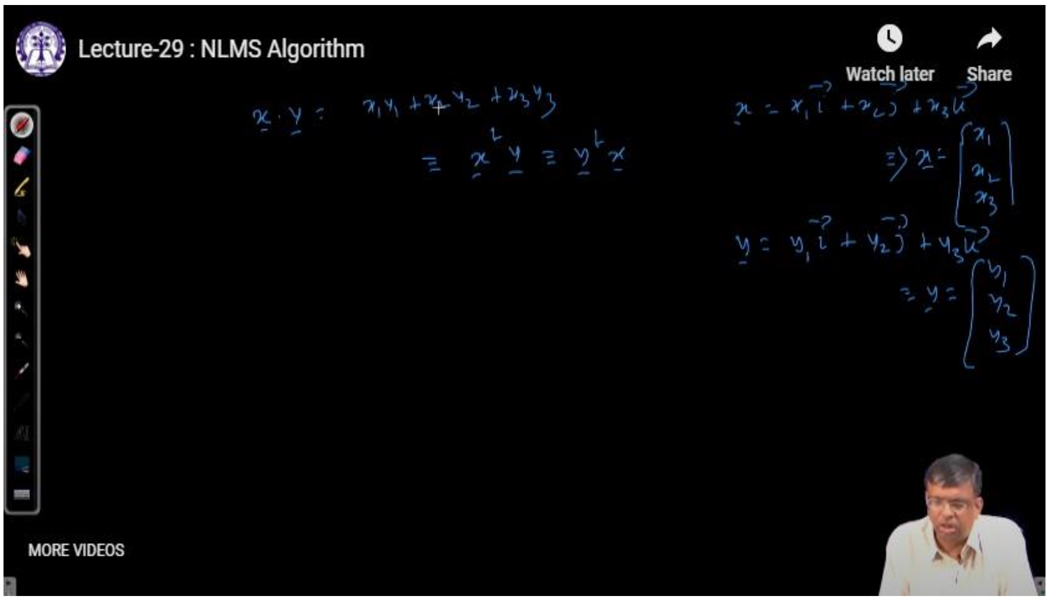
Especially when order and the length of this vector is large there is capital LMS large is a very good average. So, it is a good estimate of the variance ok. So, this relation holds good norm square of xn is same as trace R and this turns out to be LMS ok. And from LMS convergence we can say this, but here the advantage is upper limit of mu is not dependent on input statistics. It is a constant number 2 that is what we wanted from the beginning because if the input statistics changes in the case of LMS there is a problem because upper limit is 2 by trace R, R is the input autocorrelation matrix ok.

So, that matrix will change, but here even if input autocorrelation matrix changes, I am safe because upper bound of mu if I as long as I keep mu less than 2 I do not care I will always have convergence ok. This is the beauty of NLMS algorithm.



Another thing actually whatever I am going to say now it comes from a more general theory called abstract vector space, but then you do not know and neither I have any time or scope to take you to that in a you know course that I teach here I covered this and there is a vector space theory of linear algebra. Nevertheless, you have seen one thing that suppose I give you a vector in a 3-dimensional world x as x1 i x2 j x3 k equivalently x is a coordinate that is it points to a coordinate. So, you can be you can say it is a coordinate vector also like this x1 times i there is x direction x1 times y direction x2 times z direction x3 times.

So, you can equivalently write like this. Similarly, suppose y is y1 i plus y2 j plus y3 k that is equivalently y is the dot product between there will be x1 y1 plus i dot j0 i dot k0 like that. So, it will be x1 y1 x2 y2 plus x3 y3 which is same as x1 y1 x2 y2 x3 y3 that is equivalent to this x transpose y it becomes row vector x1 y1 x2 y2 x3 y3 or equivalent to y transpose x. So, if I have taken a vector of length 3 it could have been any general length say N. So, if you have 2 vectors of length N their dot product will be x1 y1 x2 y2 x3 y3 dot dot dot up to xny N.



If equal to 0 we say x y are orthogonal that is angle between them is 90 degrees in a geometrical sense. In the 3-dimensional world you can plot the vectors. So, there will be having an angle between them 90 degrees. In a general case of course, there is no motion of angle in dimensional space. So, we just say they are orthogonal if the dot product is 0 all right.

Now, suppose I consider that hyper plane, any hyper plane say you can call it W0 this is W1 and this equation is W1 xN, xN is a data W1 xN minus 1 is DN this you can equivalently write as W transpose xN is equal to DN where W is just this W0 W1 and xN we have already seen xN xN minus 1. So, obviously W transpose xN means W0 xN W1 xN minus 1 and the summation that is equal to DN we can alternatively write it like this.

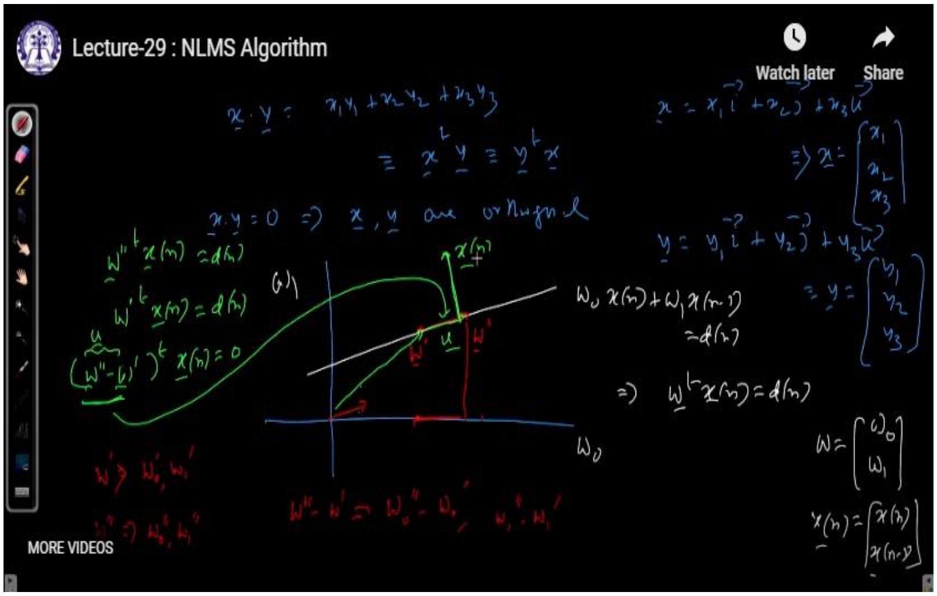
$$w_0 x(n) + w_1 x(n-1) = d(n)$$

$$\underline{w}^t \underline{x}(n) = d(n)$$

Now, you see one thing if I take a vector if I suppose take a vector here lying on this line. So, between any two points let it be W prime vector that is W0 prime W1 prime and let it be W this that is W prime is W0 prime W1 prime and W double prime means W0 double prime W1 double prime then this vector will be what it will be W double prime minus W prime ok. So, it will have coordinate I mean this W.

So, on this axis this much will be W0 double prime minus W0 prime and on this axis if you draw a line on this axis it will be W1 double prime minus W1 prime all right this is a vector. So, actually you can translate that vector and bring it here as though this vector has been added to this point has been added to this point this is another vector if you have to do addition of the two vector you pull it bring it here ok. And then you get this point. So, this vector lies on this plane because I took two points on the plane to the difference this vector lies on the plane. Now, you see one thing this point lies on the plane.

So, W prime transpose XN that is DN again W double prime transpose this point XN again DN. So, if I subtract what I get is subtract left hand side from left hand side right hand side from right hand side I get this W double prime minus W prime transpose you can subtract first and then the transpose XN equal to 0 which means this vector which is this which is this vector and XN they are dot product this is a vector you can call it a vector maybe u. So, this is my u and u lie on this plane this vector lies on this plane. So, u transpose XN is 0 that means, dot product between u and XN is 0 that means, my XN will be at 90 degrees to this plane because then only they will be at 90 degrees. So, this is my XN alright this is a property.



Now, see one very interesting thing I come back to the derivation of NLMS algorithm first at nth clock you have a WN ok. So, this equation is W transpose XN is equal to DN and this is my optimal point this vector is W opt. So, the other equation next equation for N plus 1th clock that will be another straight line as you have seen by it should still cut through this go through this point and this is my W transpose XN plus 1 as DN plus 1 maybe it is not visible here. So, let me use this we write separately again. So, this equation is this is a situation and what you just recall what we do we project on this we call this WN plus 1 again project it on the next line corresponding vector we call WN plus 2 and so on and so forth this is repetition of the old story.

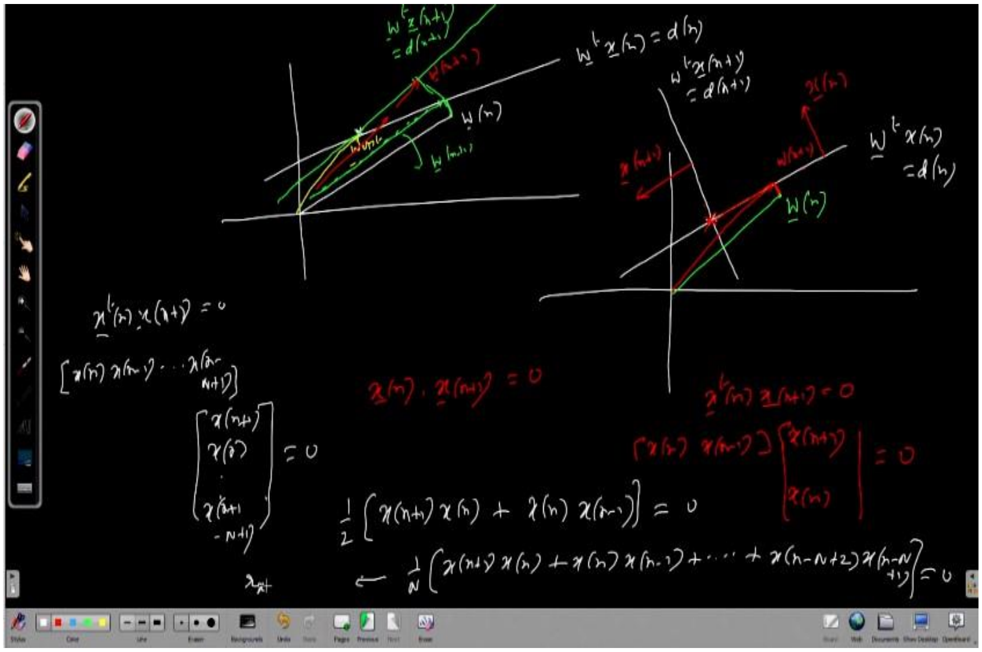
Now, suppose these two lines this white and green one for nth clock another for N plus 1th clock they are at 90 degrees with each other they are orthogonal to each other like I have got this straight line which is W transpose XN is equal to DN this is my optimal point the other straight line is like this W transpose XN plus 1 is equal to DN plus 1 suppose this is the case and here is your WN. Now, we have seen what is your just this is this line only any vector here lying on this line we have just seen previously will be orthogonal to that is at 90 degrees with XN vector. So, XN vector this is the direction of XN vector that is why W transpose XN we have seen earlier any vector on this line will be orthogonal to this so

XN vector fine. Now, let us follow the NLMS style I project it here. So, this gives me W this is my WN plus 1 this point next, I project this on this plane, but then since these two planes these two lines and in a general set of hyper planes are at 90 degrees with each other this is my perpendicular.

So, I directly heat here absolutely converging just in two steps when does it happen when the two planes are at 90 degree or equivalently speaking if it is XN for this plane XN plus 1 vector will be in this direction of the I mean 90 degree with each other it could be opposite direction or this direction does not matter angles will be 90 degree with this which means XN dot product XN plus 1 that will be 0 because there are 90 degree because this is 90 degree I mean from the diagram only you can see. What does it mean? It means X transpose N XN plus 1 0 X transpose N means XN XN minus 1 when I have just two terms and here that means, XN into XN plus 1 plus or maybe let me write this way XN plus 1 into XN plus XN into XN minus 1 that is equal to 0. So, if I divide by 2 that is also 0, but what is this I am multiplying two adjacent samples N plus 1th Nth Nth N minus 1th and then averaging. So, it is a rough estimate of correlation with a gap of 1. Now, I have just two elements in general I will have X transpose N XN plus 1 equal to 0 means it will be general you know XN XN minus 1 dot dot dot dot XN minus capital N plus 1 and this one is and this is 0 sorry I wrote wrongly this N plus 1.

So, N should be replaced by N plus 1. So, it is N plus 1 then N plus 1 minus 1. So, N dot dot dot N plus 1 minus N plus 1 like that. Now, if you multiply and this is equal to 0 if you multiply you will have more terms XN that is XN plus 1 XN next is XN XN minus 1 and so on and so forth. Last is XN plus 1 minus N plus 1 and XN. So, N plus 1 minus N plus 1 means N minus capital N plus 2 and it is N minus capital N plus 1 equal to 0.

Capital N number of terms if you divide by 1 by N that can say still is 0 there is a very good estimate of correlation with gap 1 N plus 1 N N N minus 1 N minus 1 N minus 1 like that.

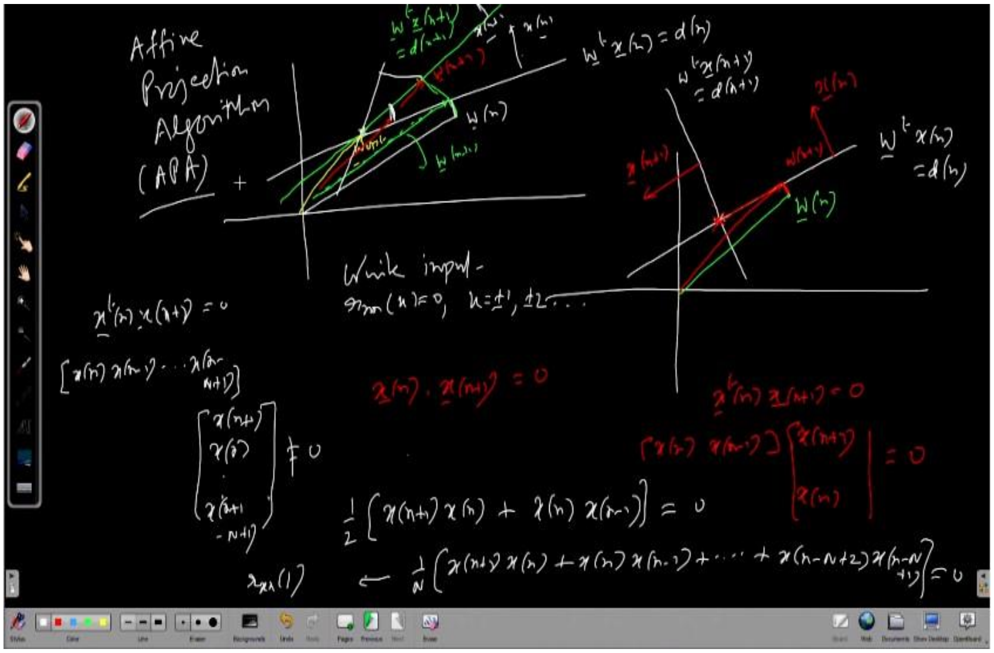


So, this is rxx1. So, when rxx1 is 0 in this case I am getting convergence very fast just in 2 iterations and in the absence of noise of course, I am just hitting the correct line point all right. Now, rxx1 0 means I mean as this is a good example of white input. White input you have rxxk equal to 0 for any non 0 k equal to plus minus 1 plus minus 2 dot dot dot dot.

So, rxx1 0 rxx2 0 here I need only rxx1 to be 0 in this case, but you know if there is any random process we expect maximum correlation at the minimum gap this their adjacent samples are highly correlated. If that is 0 further you know correlations at larger gaps will also be 0 close to 0 which means for white input in the absence of noise in the limits will converge very fast just 2 iterations. But in the on the other hand suppose they are not white that is this is not 0 suppose this is not 0 ok. So, not 0 means they are not at 90 degrees. So, they will be like you know this there will be an angle between them.

So, I hit here then hit here then hit here like and it will continue it will take more and more iterations I mean close smaller the angle between the lines and in the in the general case hyper planes I will have more and more iterations to converge. This is a problem of NLMS it is convergence you know it depends on the input autocorrelation if the input is white or correlation between samples are less it will converge faster, but if it is highly correlated there is a angle between the lines ok. It is small because of the angle between the lines is small means my xn is here and my xn plus 1 is here they are no longer at 90 degree they will have the same angle as the angle between these two lines ok. So, if the angle is less xn xn plus 1 if you take the dot product it will have larger value it will not be 0 which means correlation will be high. So, if the correlation is high and not 0 that means, angle between these two line perpendiculars or that is xn and x1 plus 1 or equivalently angle between the two lines which is same as that if you take this perpendicular and this angle between them and this they are same.

So, if the angle is less and less that is there highly correlated the two lines also will be closer to each other and take more and more iterations. So, for correlated input as a correlation increases within the samples we will take more and more iterations to converge. This is a problem of NLMS and this is overcome this is tackled in a further generalization which is called affine projection algorithm which is a more generalized version of NLMS APA this we will consider in the next class.



Thank you very much.