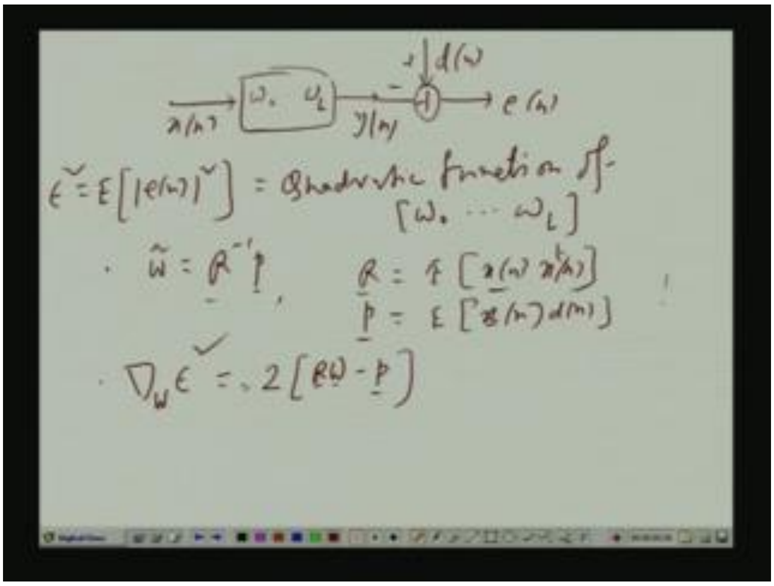
Probability & Random Variables
Prof. M. Chakraborty
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 40
Adaptive Filtering - LMS Algorithm

What we have seen in the previous class, in the nutshell was this, we are considering a filtering optimal filtering problem like this. There was a random process x n.

(Refer Slide Time: 00:55)

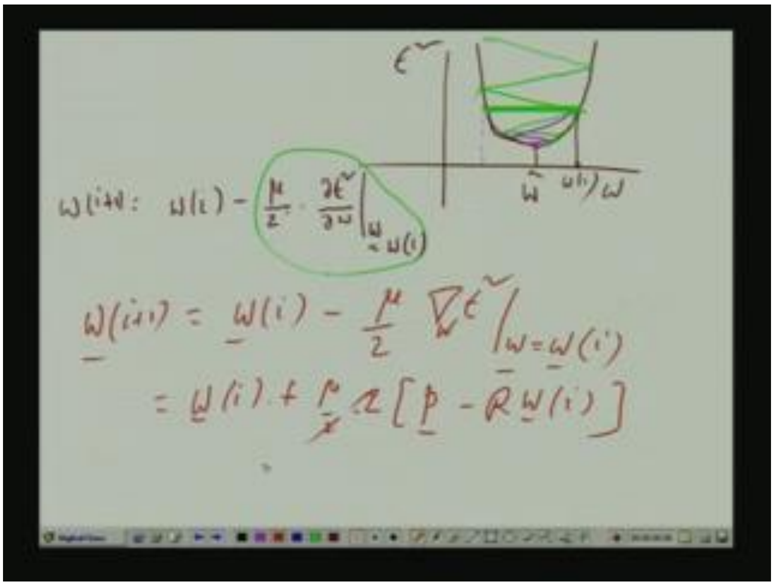


It has passed through an FIR filter w 0 dot dot dot wp. Instead of p as I told you, I will be changing it to some other thing though, because I am using p as a vector for some other what this was w 0 to say wl. So, n plus 1 coefficient, output is y n. There is a desired response d n the error is. So, you try to minimize the mod square of this error, with respect to the filter taps, because this E of. In fact, you can call it epsilon square is independent of m n because of its stationary. This quantity you have seen is a quadratic function of w 0 dot dot dot up to wl.

We have also found out 2 things. What is if we calculate the optimal filter directly, w tap is R inverse P where, R is E x n x n transpose P is E x n d n. Other thing you have seen, the gradient. This in the morning you have seen, for a particular w it was twice Rw minus p vector this we have seen. Now, one thing is that you can find out the optimal filter, directly by R inverse p. But suppose, you do not have to follow this, for suppose you know and how to compute the inverse of a matrix, we can propose an alternative method which is entire deep method. So, that as a iteration progresses finally, the filter weight W convert this on what is called the optimal weight.

We now propose add equivalent technique. That is called method of that is interactive technique it is called method of steepest descent. Now, to explain that what I do first we just take a very simple case of single coefficient.

(Refer Slide Time 03:23)



Say epsilon square say w only 1 coefficient, you can easily generalized this treatment to multiple coefficient case; there is nothing not much loss, interval generality. Now, we know epsilon square is a quadratic function of the filter taps; here you have got a new 1 tap that is w. So, epsilon square is a quadratic function any quadratic function has got only 1 minima or maxima. In our case minima, because we will be we know that there will be a minimum mean square error only.

And, then if from that minima, if you move towards your right or left, as move further away the epsilon square can only grow, it can never decay, because if it goes and they slightly decays, again there will be a local maxima, that is not possible. So, as we go further and further for that minima point, optimal point, minima point the epsilon square only has to grow that is a gradient is always positive. I mean as you are going out to what it’s only growing. For that matter it could be like this; these are optimal point w tap. If that be, so we propose that entirety of procedure like this, that suppose at i’th step of iteration you are here w i. What I do that, from w i, we I have to find out a w i plus 1. What I do, I look at the gradient here.

The gradient here is positive, because I am to the right of w tap. So, gradient here is positive. So, I do from w i, suppose a subtract a quantity proportional to the gradient, proportionality constant is a mu by 2, this a proportionality constant this 2 I am just introducing, because it will cancel some other 2 that will come up soon. So, the proportionality constant mu by 2 times the gradient. That is del epsilon square del w, at w equal to w tap and I calling w I plus 1.

Suppose, I do that here what is happening? Here the gradient is positive. So; that means, del epsilon square del w sign is positive. So; that means, from w i, I am subtracting something which is positive because mu is a positive number. That means from w i, I am not going further to the right, but I am coming back. So, in fact, I am approaching w tap Isn't it? On the other hand, if I am here say if it is still here I was here, then gradient pair is negative. So, negative and negative positive, so I add some positive quantity to w i here. And again that is for sake, because from w i here I then go to the right.

So, I begin approaching w tap, see if I am to the right of w tap, I approach it, so I go backward. If I am to the left of w tap, I go forward there is a again I i approach it. Which in 1 case gradient is positive and therefore, I subtract in other case gradient is negative and add. Further with the gradient magnitude is high, in that case del epsilon square del and del del the w its magnitude is high. So, I jump from w i, I jump to the name next point w i plus 1 by a bigger margin. By step where you can jump is a bigger, because there is a difference between w i plus 1 and w i it is magnitude is larger I jump by a bigger.

On other hand, if the gradient is small in magnitude, I just scroll, because this term which is called the update term or correction term that will have small in magnitude. So, from w i I just go to a very close point w i plus 1. So; that means, when I am further away and the gradient is high, I comeback come close to w tap very fast. When I am close to w trap, then I do not move much, I approach w tap slowly. So, maybe I can go like this, monitor 1 into as, I then I go like this, then this, then this, then this and finally, I converge. Here 1 thing is important that, your mu should be chosen carefully.

If mu is chosen suppose mu is chosen very small. So, naturally this term, this is very small in magnitude. So, you just scroll from here you go, here then here, then here, then here, then here, then here like this. It takes lot of time, many steps of iteration to converse to w tap, but at least we have bound to converge. On the other hand, if you take mu to be large, then what is happening your rate of converge is faster you jump, but so much, then so much then so much like that, but there is a danger.

If you take mu very a quite high it can. So, happen that you jump here, this point and that from here again here. So, you oscillate along this line, you never come down to w tap. So, it you never converge. Quartz could be if, you from here, you jump to this point and then to this point and then this point, so diverge out. So; that means, that is a there should be some upper bound of mu and mu should be chosen less than that. Those upper bound expression changes case or walked out, but I will not going to that, because I am not so bothered about, the steepest descent algorithm. I have derive and adaptive fitter, I will go in from this.

So, in the more general case what does this mean then? When you got not single taps, but multiple taps from w i vector you go to the w i plus 1 as vector, i is the iteration index by what subtracting from w i quantity proportional to the gradient vector. Evaluate the gradient, this was a w i. Actually, what I am doing here is nothing very different, instead of treating each word separate them, putting all the weights in a vector form and this del w epsilon square is what. It has got all the gradients, put in the vector form that is del epsilon square by del w 0 del epsilon square by del w 1 dot dot dot dot.

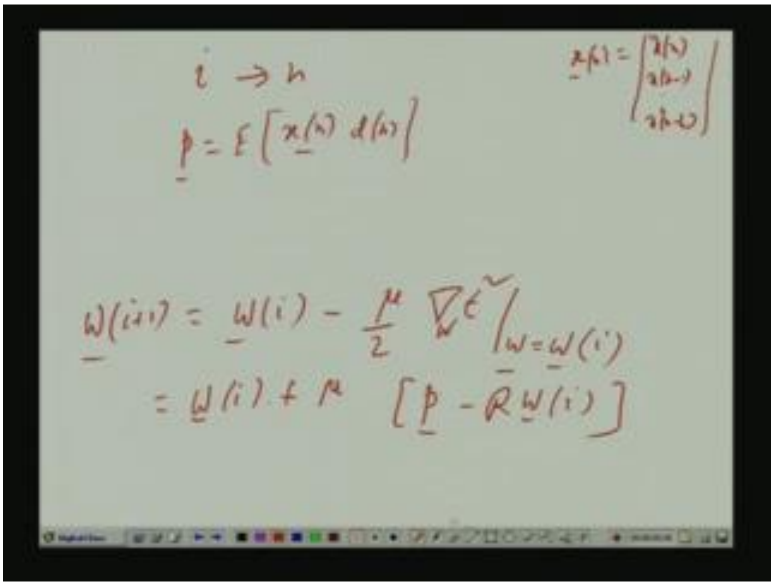
So, from each w i, I am subtracting the corresponding gradient by multiplying by force constant mu by 2. And now, I know and this method is called steepest descent, steepest, because wherever gradient is positive, I am just following that path going in the opposite direction. I am following a, I mean by which path I am following, if the gradient is positive I am following, because I am approaching w tap. If it is gradient is negative I am following, because I am approaching by which path I am following the path of gradient. That is called steepest descent, I am not going in other directions, I am following the steepest descent. It is called steepest descent I am descent steepest descent.

Now, we know what is this expression del w epsilon square in the previous slide only you have written it. So, you can put that back here, what you get, that mu by 2 and there was a 2 2 R W minus p 8, we can put a p plus here p minus R w and w equal to w i to this thing and 2 2 cancels. This is the steepest descent algorithm mu there is an upper bound for mu for convergence. Those theory I am not going into, because I am not so much bothered about this steepest descent excise, but from these I will derive an adaptive algorithm.

Now, suppose you do not know what is R, input is coming you do not know its autocorrelation function, you do not know the P vector and further means R and p they are not static. From time to time they change, because there is scenario, the context changes, the scenario changes. So, input statistics of all these and input statistics changes the joint statistics between desired response and input that is p vector that changes.

So, obliviously there is no fixed wiener filter. For 1 set of R and p, you have got 1 wiener filter. After sometime in that R and p changes you have got another wiener filter. So; that means, we have to adapt yourself, by here just looking to the data and you have to generate the corresponding wiener filter, by some you know by some adaptive method. Next time when I input statistic changes or p vector changes, you will get adaptive yourself and approach and then corresponding wiener filter so and then so forth. If you want to do such a thing I propose this. Suppose, I will to carry out this iteration in time.

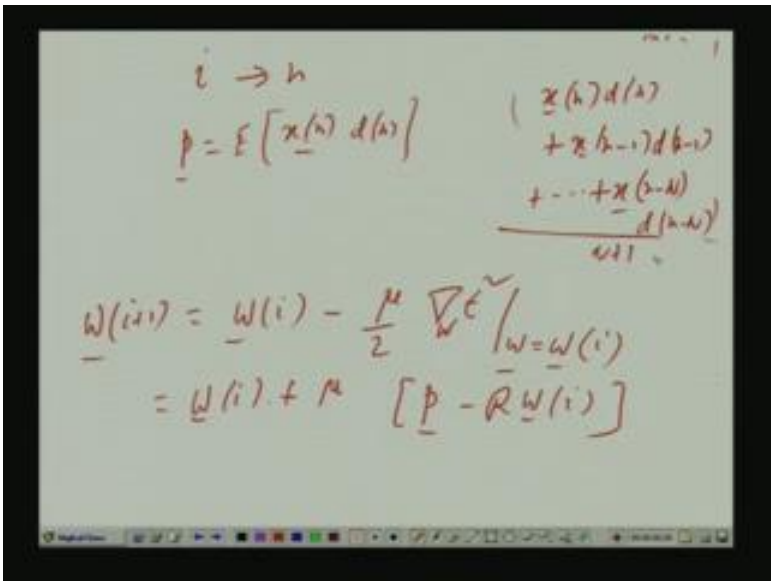
(Refer Slide Time: 12:49)



That is iteration index n b i b get it by n. So, at zeroth clock cycle, I carry out the 0 is step. But first clock, cycle I carry out the first iteration, second clock cycle a carryout the second, see response is suppose some fun. I am doing the iteration in time, same iteration.

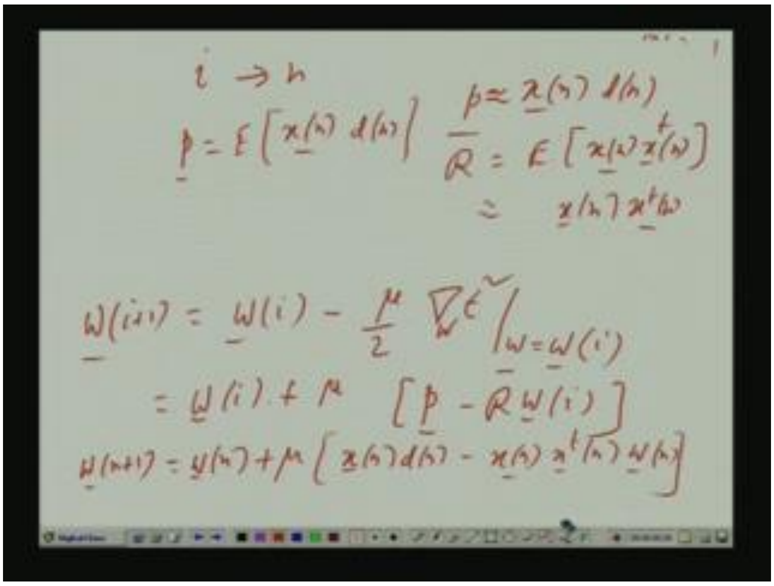
So, instead iteration index i, I can change it to n number 1. Number 2: we know p vector, what is p vector? E of x n d n, x n we know; it is a vector that has x n x n minus 1 dot dot dot up to x n minus l that times the n and expected value. Now, suppose I want to have a estimate of this, what will you do? I will be using this sample averaging technique, that is; I should have x n vector. I suppose you know what is x n vector. For your sake I am writing again x n, earlier it was p, now it is l; this x n vector.

(Refer Slide Time: 13:59)



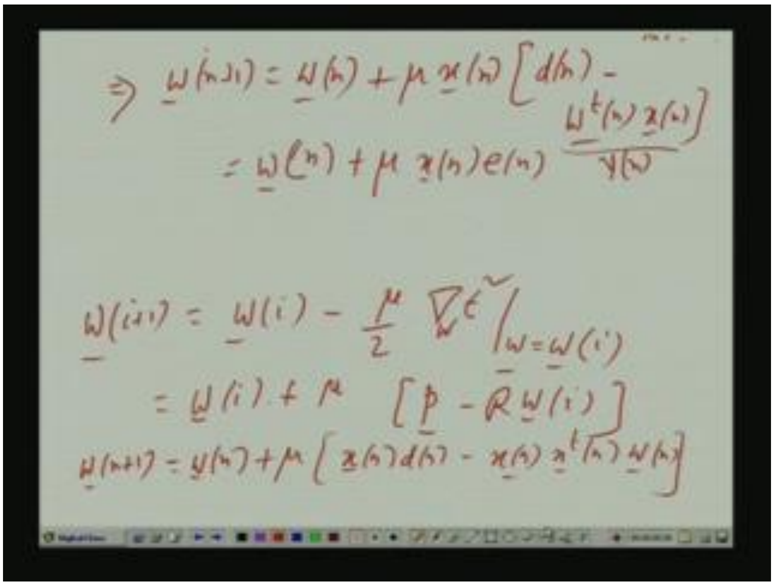
So, I really what you should do, you take 1 case; x n d n, take another vector x n minus 1 d n minus 1 dot dot dot dot, take many in minus say N d n minus N. So, this you add and then divide by N plus 1. This I have sample of this technique. That will be a some good estimate if N is large, that will be a good estimate of p, you can use that. But suppose, I should that look, I will be using not a good very estimate, just a very weird wild bad estimate, I do not mind, that is instead of, so many terms.

(Refer Slide Time: 14:43)



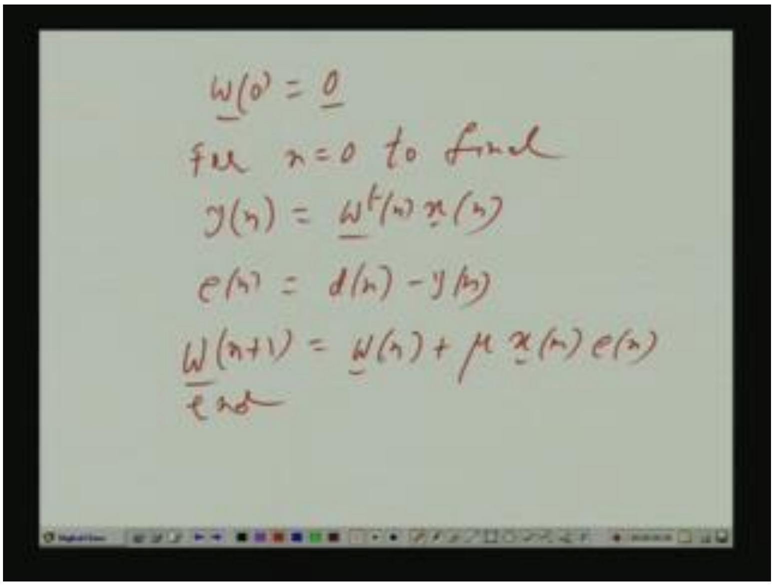
I have only that is p i approximators, simply x n d n. I agree the estimate will be not a good 1 and therefore, there will be many errors, but we will see that that errors will not affect the convergence much, that we will see. Similarly, x n vector R, so you should have x n into x n transpose plus x n minus 1 into minus 1 transpose, but dot dot dot many terms and then add and divide by the number of terms to have good estimate of power. But again, like above, I will be following a very weird path, where you I mean, I will not take a very rough estimate; just x n x transpose n. If I am put them here, what do I get? If I make this substitution w n plus 1, i has become n plus mu, E is x n d n and then x n x transpose n w n, take x n out.

(Refer Slide Time 16:23)



That means w n plus 1 is w n plus mu x bar n take common, d n minus x transpose n w n and x transpose n w n is same as w transpose n and x n, because a transpose b and b transpose w are same. You can also write this as w transpose n x n. And what is w transpose n x n? At any index n, w transpose n gives the filter coefficients w 0 w 1 up to w l. They multiply the elements of x n vector. So, what you get here is the corresponding filter output or index n and d n minus y n is error e n. So, that gives you this algorithm. This is a celebrated lms algorithm. There are many theories associated with this, I will not, I do not have time for it, I will just write down the equations.

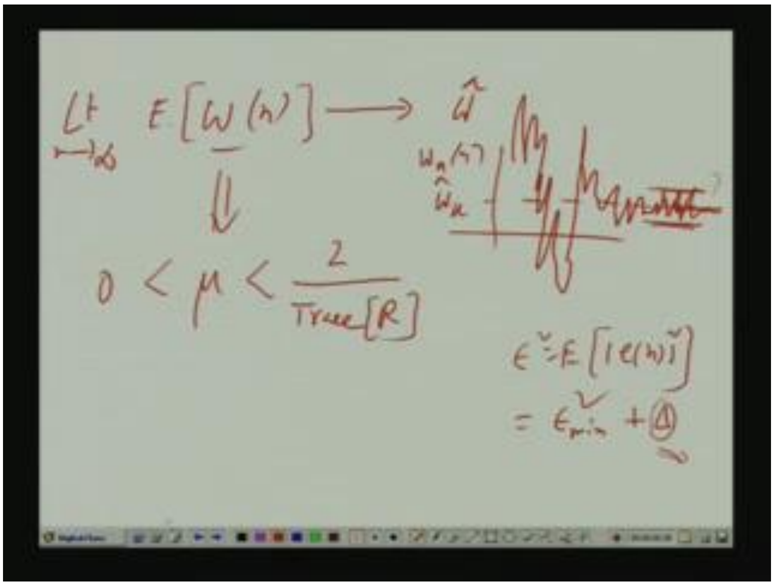
(Refer Slide Time: 17:27)



That is may be the starting value is all 0. Then for n equal to say 0 to final that is, final is the time index up to you will run this iteration, what happens. You find out first filter output y n s w transpose n x n. Then e n as desired response minus y n and then you go for the next set of weight coefficient, filter coefficients, end. So, this is a real time algorithm you see; the moment you replace i by n and used some estimates, real life estimates, real time estimates for R and P matrix R matrix and P vector, everything is data dependent. There is no e operator here, there is a direct data vector x n multiplied by w transfer given y n, then e n and x n vector is used along with n, to update the weight for next index, so on and so forth. This is an adaptive algorithm, what it does?

So, we do not need any longer any information about R and p. This algorithm you know. You have brought in some approximation, because R and p we are not replaced by their good estimates for bad estimates. For that is why we pay some price, this algorithm does not converge in the absolute sense.

(Refer Slide Time: 19:31)



It is not that, w n vector will converge to w tap as limit tends to infinity, no. It is called absolute convergence that, each tap approaches is a corresponding optimal value. This does not happen because; you know I told you, we have now replaced R and p by the exact values, but by some very weird estimates, but what does converge is the mean of this. So, the filter weights keep dancing, keep changing keep fluctuating, but as time progresses the DC for each filter tap, the DC value across which it is fluctuating that, DC value converge this to the corresponding optimal value.

So, it could be like this; 1 particular weight say wk n, it can be like this. Finally, it can be, it can go like this. This could be the DC value. Obviously, you should also see that this range of variation around the DC value, there is the optimal value is not much. Actually this is related to, what is called misadjustment, receive this you know spread to this range of fluctuation is mod. You say there is mod miss adjustment and vice versa. Before I go to that misadjustment limit will be 1 thing; this convergence will take place not always, it will take place provided; you take mu within the range 0 and less than 2 by what is called trace of input correlation. What is trace of a square matrix?

So, it is the sum of diagonal matrix, diagonal elements. Sum of all the diagonal elements, there is a trace 2 by that. If I do that, then it is guarantee that the mean of the filter weights only converge to this. Now, misadjustment or we sometimes call excess mean square error, because even with the DC weight; there is some value for epsilon. Even if the optimal weights, this quantity, it has got some value, it is minima I agree, but some some value.

But, if we do not put the optimal filter, naturally the error it is not minimized. So, that time, this epsilon square is more something more, not this epsilon mean square, but there is something a some extra term delta. What is epsilon mean square? That is if you really put the optimal filter weights, find out the e n and find out these gradients. That is the minimum possible error variants attainable, that we will call epsilon mean square.

But, the filter weights are not the optimal once. And as we have in elements algorithm, filter rate is not converging on the optimal value directly, only its DC value average value expected value converges. So; that means, here the error variants will not be the epsilon mean square, because we are not using optimal values of optimal, this wiener filter for all indices, as in test to infinity. It is only fluctuating around optimal filter.

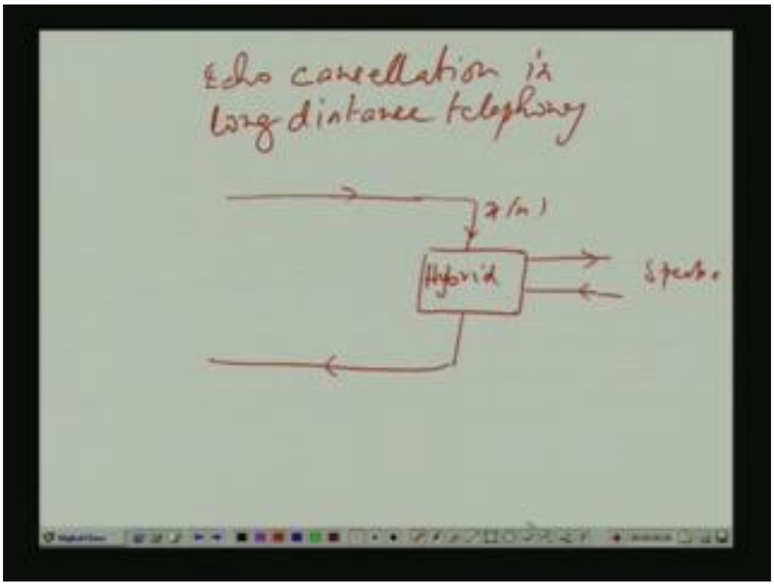
So, therefore, total epsilon square will be, not just epsilon mean square, but some extra positive quantity delta will be there, it is called excess mean square here. This excess mean square error can be minimized, if you take mu less unless, for less mu this will be less, but the price you pay is this. If mu is on the higher side in this range, the rate of convergence is faster. It converges faster, but the fluctuation will be mod around the optimal value.

If on the other, mu is on the smaller site, rate of converges will go down. It will take more time to converge, but this miss adjustment or this access mean square your delta; that will be less. Now, when you use an adaptive filter, what do you do, then if see we need en; that means, there is actually a training phase in an adaptive filter, during which the desired response d n actually is given to you, given to the filter as a input xn is given. If you find out output y n, then found out e n and use the e n in that elements algorithm to get the next filter with w n plus 1 so on and so forth, in the iterative algorithm.

So, during this phase, d n is actually available and you train your filter, so that from w n to w n plus 1 to w n plus 2 to w n till dot dot dot, you get better and better and better estimate filter value. And so that, at least in mean, the filter weight converts to optimal value. Once the training phase is over, you have no other option, but to trust the filter coefficients as to be the best possible 1 that you can have and use them, as a good substitute for w tap and do the filtering. So, that output may not be very close to d n, but will be some out good estimate of d n, you have to have be happy with that.

If after some time, inputs statistics changes, again you use new d n. So, that is why this filter is train from time to time. May be after every half an hour you dedicated some you know some particular some time. Time period is training period, give the d n, take input and train the filter. So, adaptive filters are trained periodically. These are convergence rule and all that. Let us take some example.

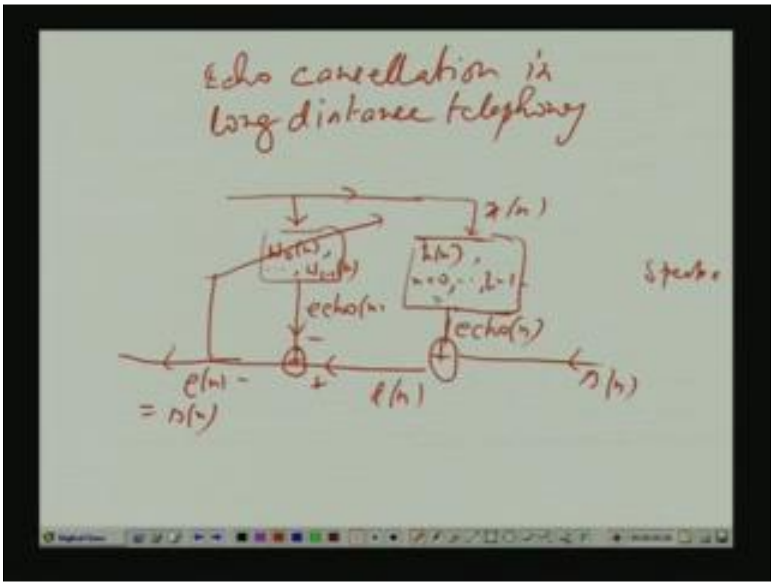
(Refer Slide Time: 24:58)



There are many practical applications of this adaptive filters, if not possibly half an hour time, I cannot cover them consider what is called Echo cancellation in, whenever you speak to somebody in overseas. Sometimes you might have heard that, an echo of your voice is coming your ear, which is very annoying and disturbing. It affects you; you know you are hearing the other end the speaker. What happens there is that; to this end, suppose this signal is going, then there is something called hybrid. What happens what this hybrid does is this; it takes the input signal and diverts it to this side to the speaker and takes the speakers signal and diverse it to this side.

So, no part of the incoming signal gets leaked to this side, but often what happens; the hybrids are not faulty. So, part of the signal, if the input this signal is x n, part of the x n, some function of the x n comes out as an echo and that goes travels to the other side. It reaches the ear of the speaker horizontally speaker of the left hand side, as an echo. To avoid that, what you have to do, we model the hybrid as the linear time invariant system, maybe I mean usually of very high order that is.

(Refer Slide Time: 27:10)



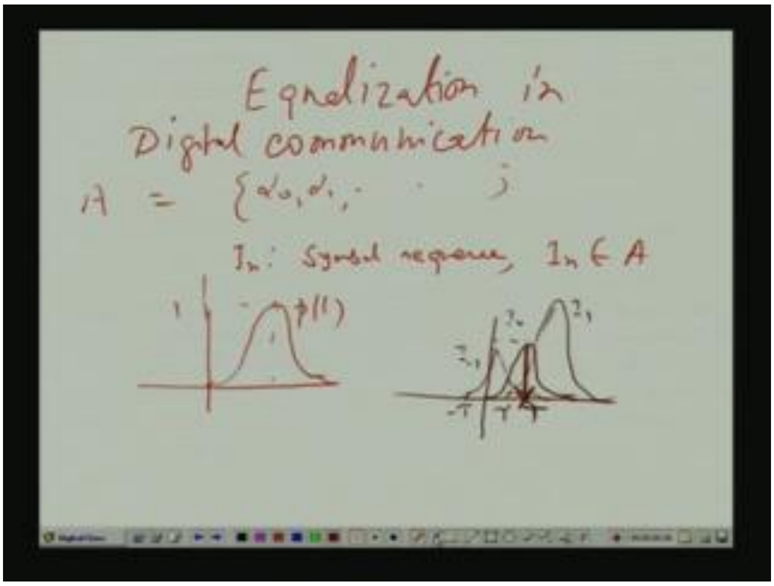
Here you forget this for the training, as though this is a filter h n, n from 0 dot dot dot say l minus 1 very large error filter and the thing that comes out here is echo. What comes out here; is echo plus this person's speech. If this person is speaking, so what comes out here? x n covert to this sequence as the echo plus this person's speech. So, may be instead of doing it here, these are model actually, the echo is united. So, e n and this is this person's speech say s n. So, s n plus d n, if you call it d n, it is d n which goes to the left hand side.

So, not only s n, but also e n that is, the echo comes back to the original speaker. To avoid this e n, if I can, if I put a filter here adaptive filter dot dot dot same order say w l minus 1. It was this, the signals as is desired response, take the error and this is error. So, this is not e n, let me change the symbol e n to something else. Let me call it echo; echo n and this is a e n. And if, I use the e n, this is the symbol to update, to adjust this filter weight by the elements algorithm, where they are adjusted by the elements algorithm, by this en. What will happen?

Then this filters, if everything is else is done properly, this filter weights will finally, converge to at least in the sense of mean, it will converge to it is h n. If it is actual, if the misadjustment is very small, what will be the happen? They will actually converge to h n, but for very small difference. In that case, the 2 outputs will be same. If, it generates echo and it also will it generate echo n, because input same for both. In that case the 2 outputs will be same if it generates echo n it was. So, will generate echo n because input is same for both.

So, if you take echo n subtract from d n, once again get back e n equal to s n. So, each s n is s n goes that has no echo or very little echo. This is called echo cancellation. This echo cancellation is particular for data modem, because human beings can still, using their intelligence can you know differentiate between echo and actual sound to be heard. But data, if it is data modem, a computer cannot. So, echo cancellation for data mode is very essential. Take one more example, echo equalization in communication.

(Refer Slide Time: 30:42)



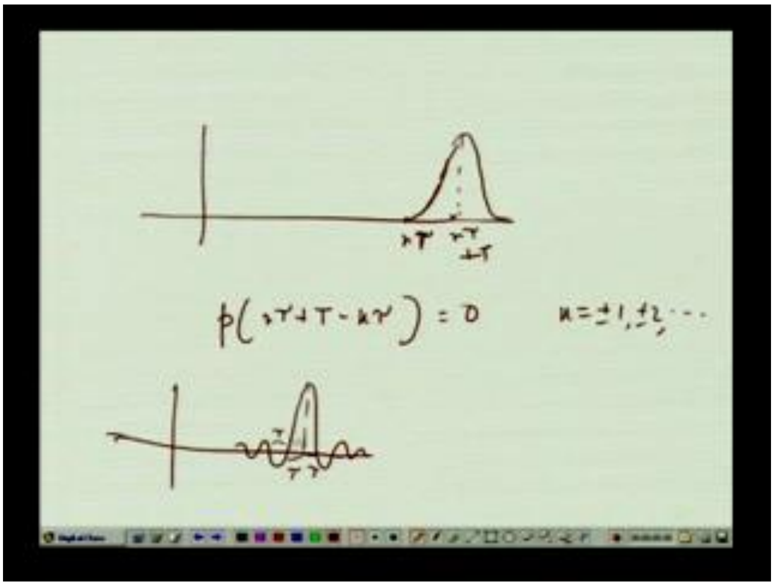
In digital communication what you do is, you are transmitting sequence of symbols. There is an alphabet A of various symbols; alpha 0 alpha 1 dot dot dot finite alphabet. So, there is sequence, what you transmit is a sequence of symbols? You know symbols can be complex like A plus jb, you know that i component and q component and all that. So, that see In is the symbol sequence; In element of A that is, I 0 is 1 symbol from here I 1 is a another symbol from here this is the symbol set, you transmit.

But essential problem them 1 of the main problem in communication is that, even if you order symbol sequence to a transmitted is discrete, you that media through is there will travel, that media is continuous or analog. So, for transmission of size discrete messages, you need an analog pulse to carry that information. Suppose, that pulse is designed like this that, there is the basic pulse. Assume value could be 1 p t, no trigger it here and observe it here.

Now suppose, we this pulse you have a thing like this. If multiply the pulse by I naught, so it will be I naught. You triggered it here and measure it the receiver at this point of time, so T. But since, you are transmitting fast, you will not away for time up to T. You might trigger the next pulse at tau and multiply the pulse by I 1, at minus tau you trigger another pulse;, so I minus 1 so and so forth.

So, at the receiver if you want to measure at T, if you sample, you see there is a I mean from this figure it appears there is a interference from the these pulse, from the right hand pulse and the left hand pulse and from many low right hand pulses, from many left hand pulses. That gives us to inter symbol interference. To avoid that we design the pulses so that at T, when a measure these guy, all other guys should go through, this 0 0 crossing.

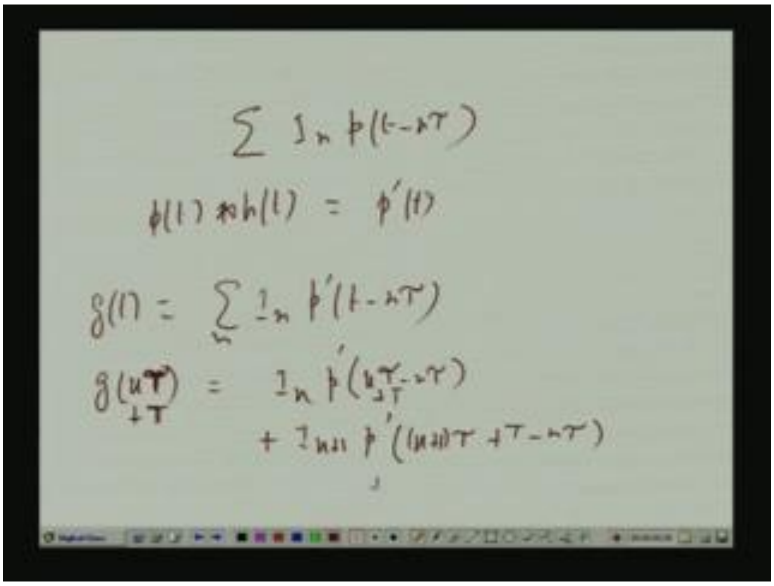
(Refer Slide Time: 33:41)



tTat means, in general you design the pulse so that, at n T a tau equal to 0 started, we added T, then tau you switch on the next pulse, so on so forth and measure in that T plus tau. At n tau, you suppose triggered 1 pulse and this is n tau plus T p. Then, the pulse to the right this, should pass through 0 here, pulse to the right of that also pass through 0 here. So, all other pulses to the right of this and left this should cross through 0. That means: p the center point minus k tau should be equal to 0; k can be for both positive and negative. This is called 0 forcing condition like a Nyquist pulse. You know this seen pulse for that matter. The pulse we have shown, it does not follow this 0 forcing condition, but look at this pulse.

Suppose, this is your center point and this much is tau, this much tau like that, this much is tau. So, as you go to the left by tau or 2 tau or 3 tau or to the right also by tau, you have 0 crossing. So, this will ensure that there is intersymbol interference. So, you design such pulses, a these pulses called Nyquist pulses. Problem is. So, that, so; that means, what you transmit is In p t minus n tau.

(Refer Slide Time 36:07)

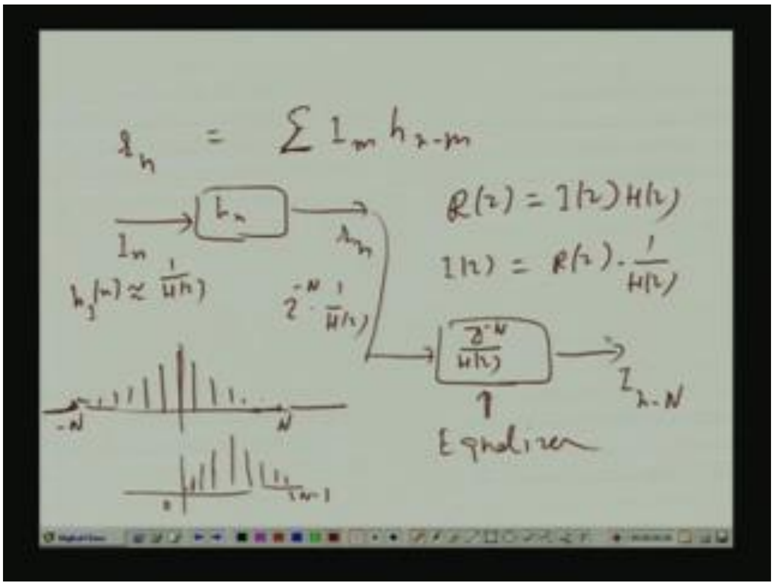


Problem is this pulse is pass through the channel and that channel it is not the ideals, it is not an ideal system. So, p t convolve with channel, will give rise to some p prime t and this p prime t will no longer satisfy the 0 forcing condition. So; obviously, I will get back this, what I will receive is say g t plus In p prime t minus n tau. So; that means, I will have intersymbol interference. If you want to find out, say if you want to sample this, at and a particular tau say, a summation is e n.

Suppose, you want to find out that k tau k T I sample, what will we get Ik p prime kT minus n tau plus say Ik plus 1 p prime, this should be k tau pulse T. It was triggered at k tau and I am sampling at T for the to the rise of k tau plus T. So, k tau plus T and then k plus 1 tau plus T minus n tau so on and so forth. These terms pulse will not be 0 because 0 forcing has been the demolished. As a result you will get a combination kind of stop there is Ik times some constant plus Ik plus 1 times some constant Ik minus 1 times of constant so on and so forth.

The constants are coming from p prime t and p prime t is what? Convolution between transmitted pulse and the channel impulse response. That means you get a discrete time model, that if you sample it what you get we receive sequence.

(Refer Slide Time: 38:28)



Say R n is nothing, but convulsion that is Im sum h n minus m. That means; that is the equivalent model is this, you transmitting a sequence In, there is channel responses h n and you are getting this R n. May be essentially I am putting everything in subscript, will be put this also a subscript. So, R n is to longer In. That means what is R z? That is I z times H z. So, what is I z? That is R z by into 1 by H z.

Now; that means, if I take this, pass it through a system 1 by H z. What will I get? I should get back In. Problem is H z is what it is as a z transform a sequence H n. Its higher sequence, because you know intersymbol interference they are only from adjacent neighbors and not from very far away pulses. So, those coefficients through the die out, this co efficiency which are related for away pulses, they are die out, so its higher sequence.

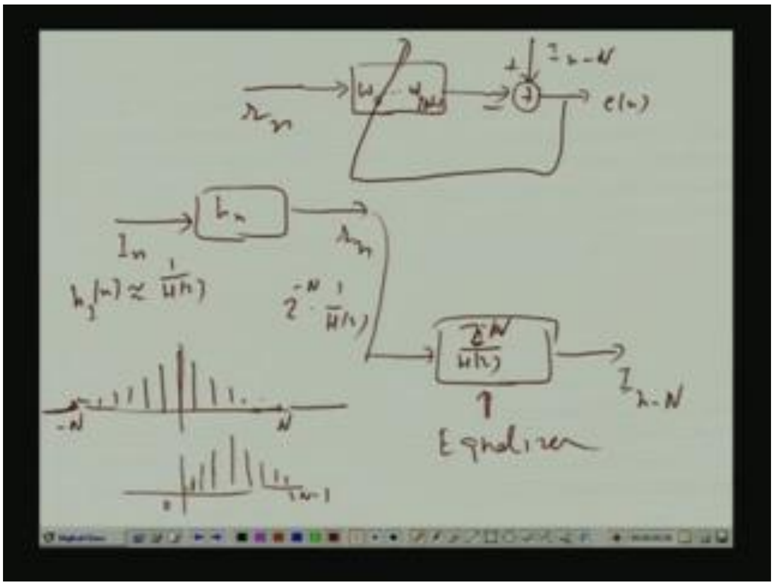
But higher sequence a z transform is to H z that I have no control on the roots. There is on the 0s obvious, if they are polynomials. So, in generally some of the 0s can be within unit circle and some of the 0s can be outside the unit circles. 1 by Hz is 1 is a filter, where for 0s of H z becomes pole here. So, some poles are within unit circle and some are outside unit circle. So, it is a stable, but non causal system, something like this got by H z.

So, if H prime and corresponds to or may be h I, I for inverse here whether the inverse is system corresponds to 1 by Hz. It will because stable, some sort of this hobble, but will be non causal. See this is stable and therefore, absolutely some hobble it will die out. You know any stable system impulse response, has to die out after some time, because if they have appreciable value all along, then it is not absolutely thermal, it is not stable.

So; that means, we can trunk it, may be up to N and minus N, beyond that we notice already 0 and 0 or very close to 0, so trunk at it. Now, this is your hIn and this is not clear eligible and practice. But if I have z inverse N 1 by Hz, the corresponding filter is response will be what, delayed version of this delayed by N and that will start here, 2 n minus 1, this is reliable, but this is causal.

So, if I what will do? I construct a filter z to the power minus N by H z and what will I get here, it is not I n, but because is z to the power minus N, you can club with R n. So, R n here is actually getting delayed by n cycles pass through 1 by Hz. So, what will get here is not In, but I n minus N. This is called equalizer linear equalizer, which equalizes the effect of the channel. Problem is, if I really know the channel, I know H z therefore, a can construct this. Unfortunately channel is of often not known, I mean channel characteristics changes for time to time. So, you have to do some adaptive filtering and then it becomes at adapt equalizer. So, what we do there is a following; it is a training phase, did you who is a transmitter transmit some known sequence In, which is also known to the receiver. This standard sequence is called pilot sequence. So, receiver knows what the equalizer output should be ideally, that is what are transmitted. So, what it does.

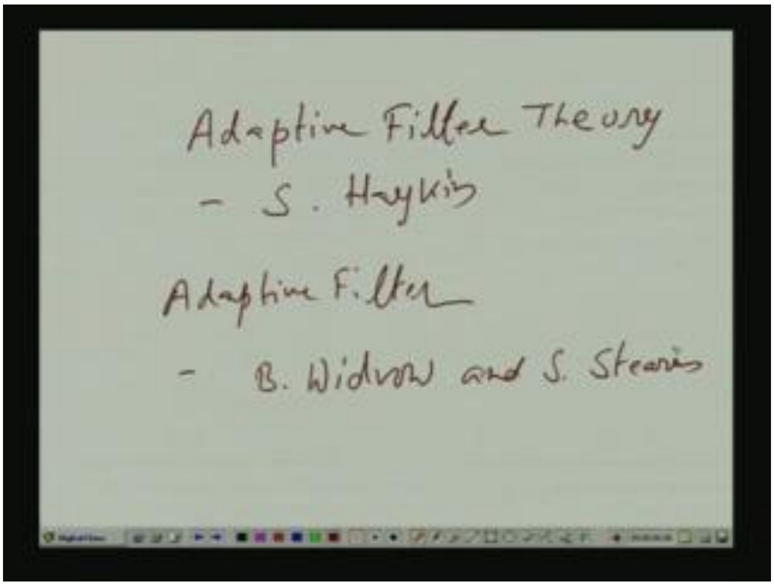
(Refer Slide Time: 43:27)



It take the receive signal R n, it constructs an adaptive filter, because this equalizer is a causal if I get system of length 2 n as I shown. So, you construct and equalizer here, may be from w 0 dot dot dot w 2 N minus 1. This output should be as close as possible to got not In, but In minus n. In is 1 that transmitted sequence, that pilot sequence, it has delayed by N because; you know that your channel could have been, I mean this equalizer actually is non causal, but you have merit causal.

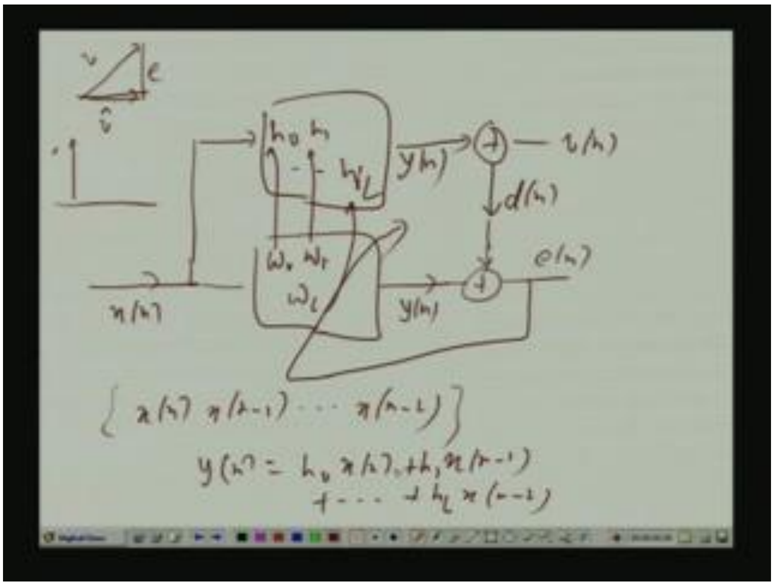
So, there is a delay that has got into the system so; that means, desired response also has to be delayed and use the corresponding e n and use it to adapt it. What is that equalizer, this is strain properly, this will approach, this equalizer Z to the power minus N by H z. Therefore, output will be a good estimate of In minus capital, this is called adaptive equalization. There are many other application on adaptive filter you know, I will not. You can consider this beautiful book, is wonderful book by Haykin.

(Refer Slide Time: 44:55)



Then there is a book on adaptive filter by Widrow and Stearns. Widrow was the inventor of LMS algorithm by Bernard Widrow and S Sterns. 1 more example I can consider and then I can wind up; system identification.

(Refer Slide Time: 45:50)



Suppose, there is a plant, this is model by some fir system; w 0 w 1 dot dot dot wa minus 1. You are observing it, you do not to the coefficient. There is sequence that random sequence x n that goes in, you put an adaptive filter here. Instead of using w, let me use some other symbol here. h 0 h 1 dot dot dot h l minus 1 and there is a measurement noise v n together is d n. This measurement noise is uncorrelated to it x n and therefore, uncorrelated with this output y n.

Now, what is the optimal filter here? d n that filter which corresponds to what the orthogonal projection of d n on what, on the space spend by x n x n minus 1 upto x n minus l plus 1. Maybe x 2 of l minus 1 y will l minus 1 make it l. That is, if you consider this space spend by these fellows; d n must be, what is a base filter with optimal filter here? The orthogonal the 1 that gives the orthogonal projection of d n on the space spend by them. What is d n? y n plus v n. So that means, what is the projection? Projection is a linear operation.

So, projection of d n on the space spend by them is same as projection of y n on the space spend by these fellows and projection of v n plan by these fellows. Now, you see if v n is uncorrelated with them. So, v n is orthogonal to each of them, because when is uncorrelated. And if where is orthogonal to each of them, then; that means, projection of v n on them on the space plan by them is 0 listen it, it is like this situation. If it is not orthogonal, you have got 1 projection of a vector v; you can call it v tap and is the error e. But if v itself is orthogonal, then v tap is 0.

So, v n, the projection of v n under space spend by them is 0. So, projection of d n on the space spend by these fellows is same as the projection of y n plan by the on these fellows. Now, what is the projection of y n what is y n after all? y n is h 0 x n h 1 x n minus 1. So, y n is already given as a linear combination of x n x n minus 1 up to x n minus l. So, y n already lies in the space plan by x n x n minus 1 up to x n minus l.

So, projection of y n on that space is y n itself. So; that means, this output here for the optimal filter case should be y n and therefore, w 0 should be equal to h 0 w 0 should be equal to h 0 w 1 should be equal to h 1 w l h l. So, since optimal filter is not compute, it is compute in adaptive ways. So, you take you put it in the adaptive filter frame work. This d n, this is y n, this is e n. Use this e n in the elements mode to adapt them.

So, finally, after convergence, this w 0 should have approached h 0, w 1 should approach h 1, w l should approach h l so on and so forth. It is called system identification. There are many other applications in that adapting; noise cancellation, interference separation and that entire erupting beam forming which is very used in, smarted in applications where you know I mean, suppose there at 2 persons, it is very usefully into this mobile communication.

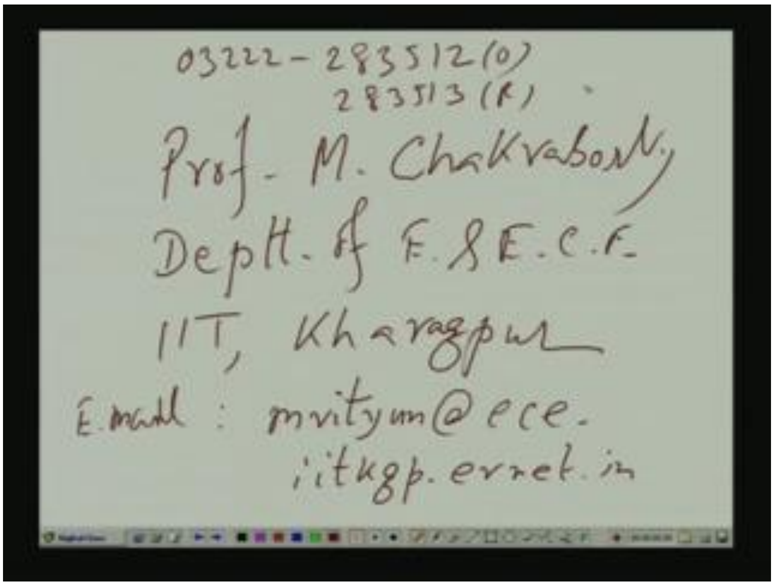
Suppose, there are several persons, they are using the same frequency brand, but sending signal from defined directions. I want to receive only 1 and I do not have to hear others. So, I will be you know I mean, adjusting my antenna weights so that, you know the BV's formed, there is antenna gain this maximizing only 1 direction. The direction which is another to here or to do and noise are created in other direction from which other signals are coming. So, thereby you do you some kind a special filtering.

Now, here the persons who are receiving the transferring signals, then in motion, so that direction changes. So, this business of giving, you know null in certain directions and applying they beam in certain direction, this need to be again adaptively adjusted. So, that gives us to erupting beam former. So, there are many applications which you will find in this book. So, I will not discuss any further. I hope enjoyed this course.

Now, you know a probability random process, there are too many things in this. So obviously; then you know in course like this, all cannot be covered. But, I suggest, I feel that whatever has been covered some has a good background and you can know go into you know, I mean more advanced topics. By yourselves you can going to topics like Martin Gales, you can going to topics like Q theory and things like that. These are all given in I mean in at a enter theory level in Papoulis book. And the background that is required for you know reading those, for the studying those, that is already been covered.

So, I hope you enjoy this course and for any feedback, you contact me, you can write to me or you can contact me. It will be nice to have a communication with your end. So, my you can contact me here.

(Refer Slide Time: 51:49)



My e-mail number e-mail address is mrityun at ece dot iitkgp dot ernet dot. My phone numbers 0 3 2 2 2 2 8 3 5 1 2; that is office, 2 8 3 5 1 3 that is residence. Then hope you enjoyed this course and in case you learn something from these that will be a good reward for me.

So, thank you very much for all your attention. Good luck and good bye.