Introduction To Adaptive Signal Processing

Prof. Mrityunjoy Chakraborty

Department of Electronics and Electrical Communication Engineering

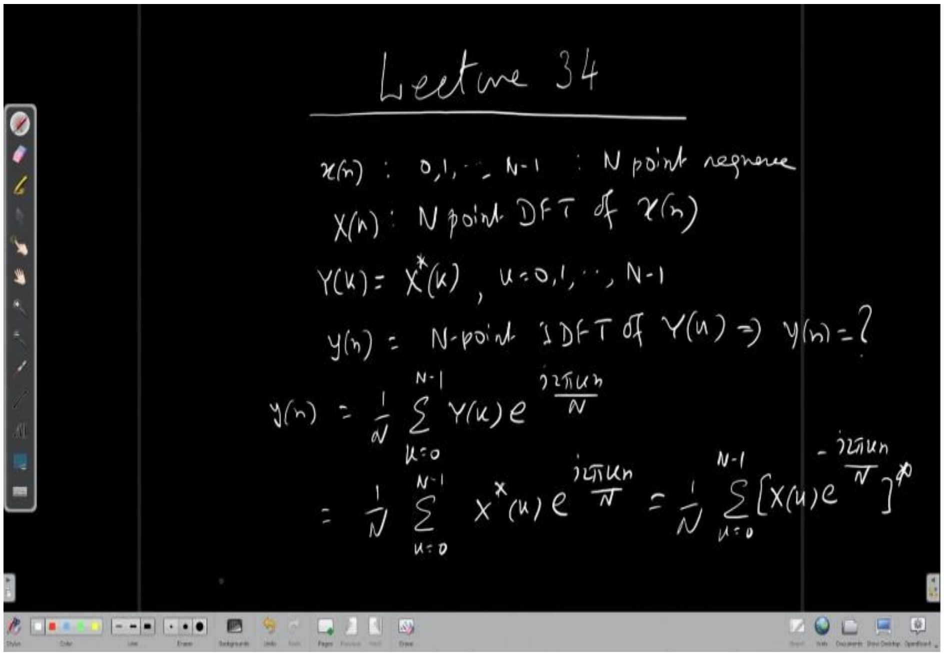
Indian Institute of Technology, Kharagpur

Lecture No # 34

Introduction to RLS Algorithm (Contd.)

Okay, in the last class, I gave you one exercise. Actually, I was working out something and then in the middle I left I said that I mean you take it home as an exercise and that I will do again because that is a basic thing basic result that I will be using today to present the FFT based implementation of the block LMS algorithm. So let us come back to that result. Suppose xn 0 1 dot dot dot n minus 1 n point sequence, x k point DFT of xn then suppose I take conjugate of each xn call it x star k and which I say yk again k equal to 0 1 dot dot dot dot n minus 1. And if I take the inverse DFT yn which is n point i DFT of yk then what is yn that was the thing. So, yn again it will be n point i DFT means this is a sequence of length n only.

So yn so let us work out yn as we know is 1 by capital N yk this is by the formula of i DFT e to the power j 2 pi k n by capital N summation is over k n is fixed from outside then you replace yk by x star k. This product you can write as xk e to the power then conjugate of this e to the power minus j 2 pi k n by capital N put everything in a bracket conjugate outside. So, it is like this xk e to the power minus j 2 pi k n star which you can write as let me go to the next page. I am running short of space so next page.



So, yn I write this first I first rewrite what I wrote there this what I had with a star outside. Then you can write it as I have to somehow bring i DFT kind of stuff and now you know this is a complex number xk e to the power minus j 2 pi k n by capital N call it z1 into z2 which is the complex number call it z so product is z. So, z star if I give you a set of complex numbers and their summation seems like this z1 star 1 complex number is a star plus z2 star there are capital N terms I am just taking 2 this is same as I have discussed it already z1 plus z2 star in the previous class earlier. So, I can even put it like this xk e to the power minus there is first sum and then star all right. Now this inner summation I want to write in the i DFT form there I should not have minus sign minus sign comes in the DFT.

$$y(n) = \left[\frac{1}{N}\sum_{k=0}^{N-1} X(k) e^{\frac{j2\pi kN}{N}}\right]^*$$

So, therefore, what I will do is simply do this xk I bring this factor e to the power j 2 pi k capital N by N.

$$= \left[\frac{1}{N}\sum_{k=0}^{N-1} X(k) e^{\frac{j2\pi kN}{N}} \cdot e^{-\frac{j2\pi kn}{N}}\right]^*$$

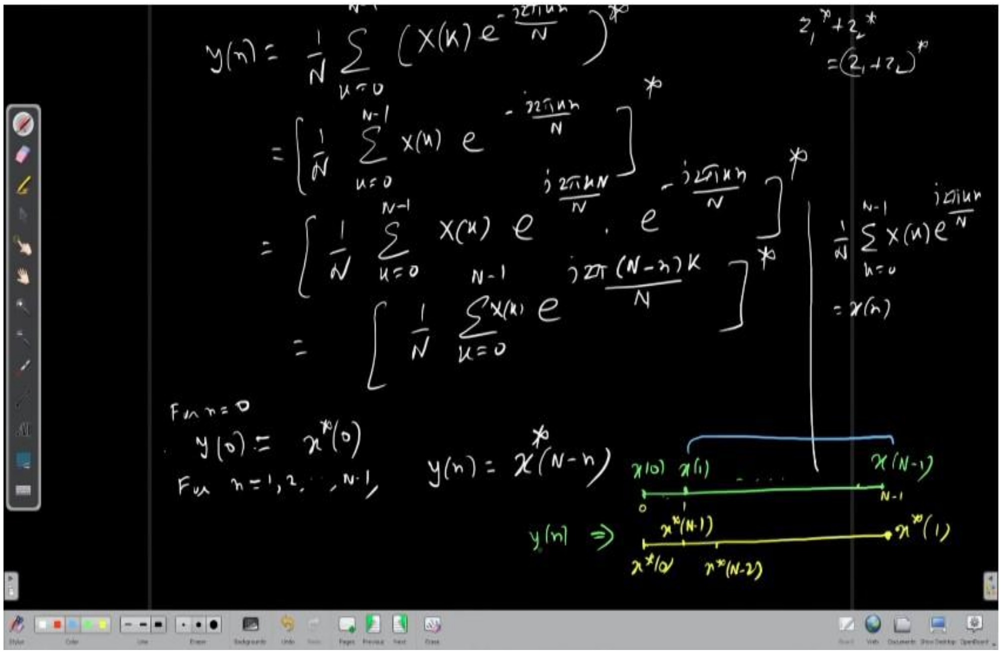
So, you see N and N cancels and k is an integer e to the power j 2 pi k that is always 1. So, this is nothing but 1 and this other stuff which is basically e to the power j 2 pi capital N minus N k is common by N star all right.

$$= \left[\frac{1}{N}\sum_{k=0}^{N-1} X(k) e^{\frac{j2\pi(N-n)k}{N}}\right]^*$$

So, first put k N equal to 0 take N equal to 0 case y0 if you put N equal to 0 it is e to the power j 2 pi N by N N into k by N N cancels e to the power j 2 pi k is 1 and small n is 0. So, this entire factor is 1 sorry this is xk this is here this is xk.

So, this summation of this you know xk into 1 by N star which is basically you know this that 1 by N xk e to the power j 2 pi k N by N if you sum over k you get x N this is the inverse GFT formula ordinary inverse GFT formula. So, at N equal to 0 if you put 0 here e to the power 0 is 1. So, it is just summation of xk s divided by N that is what we have here this is turning out to be 1 if small n is 0 because e to the power j 2 pi capital N into k by capital N is 1. So, you get nothing but x of 0. So, this is x of 0 with a star outside.

So, x star 0 for N equal to 0 and for N, 1, 2 up to the last point N minus 1 it is nothing but y N is equal to capital N minus small n if you call it an index small n is when small n is 1 it is capital N minus 1 when small n is 2 capital N minus 2 dot dot when small n is capital N minus 1 it is 1. So, you get various values of x N because this is nothing but IDFT at capital N minus N this was small n it is capital N minus N. So, it is that means, there is a star of course, that means, if originally you had thing like this x 0 x 1 dot dot dot x N minus 1 now y N will be y 0 is nothing but x 0 with a star. So, y 0 will be x star 0 this y 0, but then at point 1 when small n is 1 this small n 0 1 dot dot dot N minus 1. So, when small n is 1 it is x star N minus 1.

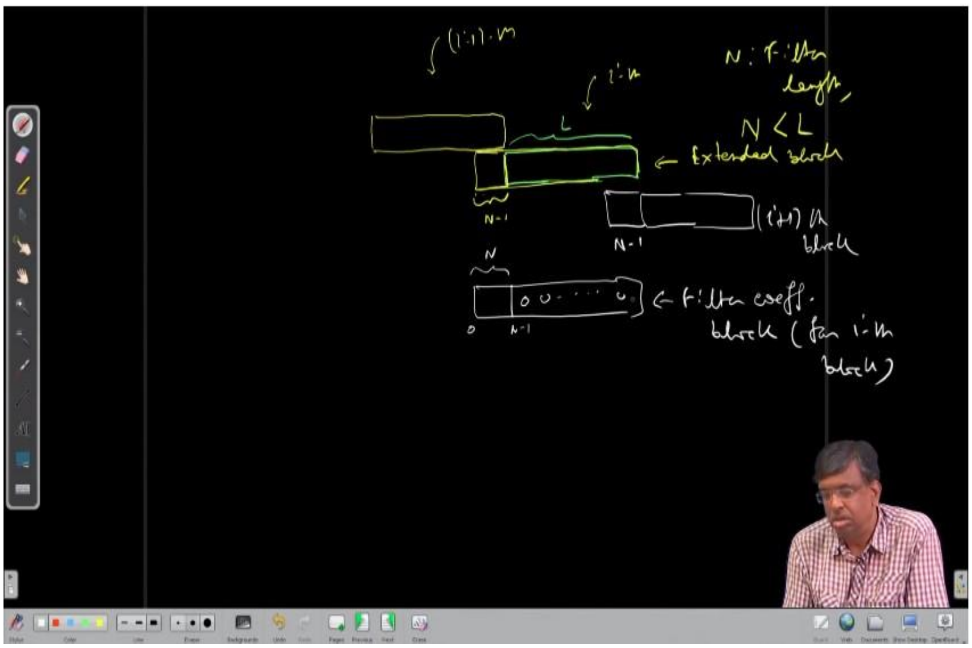


So, this guy will come x star N minus 1 then when a small n is 2 x star N minus 2 this guy will come and dot dot dot when small n is last point capital N minus 1 it is x star 1. That means, if you take this part first part x 0 it just remains there only with a star, but this part gets flipped x N minus 1 comes here with a star x 1 goes to right hand side it is just flipped here you started from x 1 then x 2 up to x N minus 1 that is N minus 1 N minus 2 dot dot dot x 2 x 1 with a star it gets flipped. So, leave the first guy as it is with a star and the remaining part you flip of course, with a star this is what y N is we will be using this remember this. Now, we come to our block LMS algorithm in the block LMS algorithm we know that we divide the data into non overlapping blocks for every block the filter weight is fixed it is not updated we update it only from only when we move from one block to the next block then we use the block LMS formula all right. And within the particular block we use the same filter carry out convolution and find out the output error and all that ok.

But remember we told you how to carry out this block partitioning and then convolution that is by overlap and safe method that is suppose your original block is this one particular block of length L all right length L. But in the previous block previous block is here actually there in the same line I am showing you separately what we do we make an extended block of this. So, if it is ith block if it is i minus 1th ith block we start with an extra part here length N minus 1 N filter length all these were discussed earlier N is less than L. So, this is my extended block here I take this block separately convolve the data here with the particular filter coefficient vector for the ith block which is fixed for the entire this ith block or ith extended block. But then I discard the output here and take output from here find out the output error that is desired response minus filter output from this point onwards keep them for the update for updating the weight vector from ith block to i plus 1th block like i plus 1th block will be here originally here what we extended again by N minus 1 this is i plus 1th block.

So, we take ith block separate ith extended block separately carry out convolution with the corresponding filter impulse response vector or filter coefficient vector which is fixed for the particular block does not change does not update carry out the convolution by overlap and same. That is we do not we discard the convolution output at this point we take convolution output here and this can be done efficiently using FFT we told that if you now take out I mean this filter this data block and take the filter block suppose this is the filter block for that is only up to here 0 to N minus 1 so total length is N here and we fill up this with zeros then first N minus 1 output if we carry out circular convolution that is you take the FFT of this block FFT of the filter coefficient block for i th block. So, FFT here or DFT here DFT here we multiply take i DFT which is a circular convolution between this block of data and this extended filter block extended with these zeros then the circular convolution first N minus 1 results are incorrect which is fine with me because I am otherwise I was going to discard them from this point onwards till this point results are correct. So, what I do I will take a particular block for explanation so let us take this block to be like this so this is 0 1 maybe up to N minus 1 for N minus let me do it in a better way. So, this is and we expand this goes from this is length N minus 1 so that means if I start from 0 it should go up to N minus 2 N point 0 to N minus 2 length is N minus 1 N minus 1 and then this is N minus 1 on the time scale this is a time scale N minus 1 to what this length is capital N this length is N minus 1 so total length is L plus N minus 1 L plus N minus 1 so this will be what 0 to L plus N minus 2 then only total length is total length this much is L is it not this much is N minus 1 total length is L plus N minus 1.

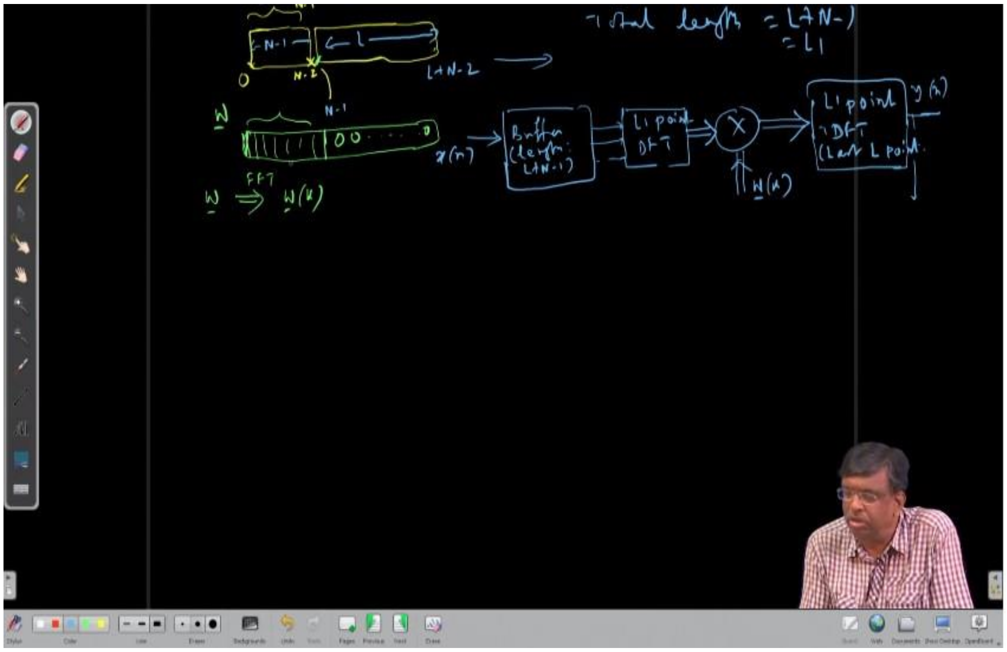
So, if you start at 0 and go to the right extreme right point will be L plus N minus 2 then only total length is L plus N minus 1 because you are counting from 0. So, this is the block we are considering.



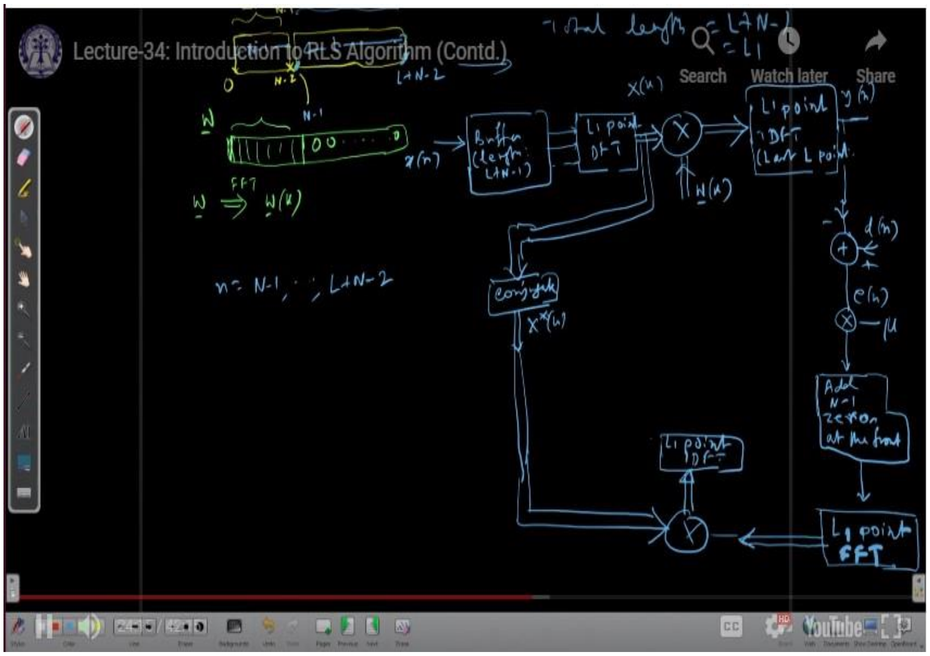
So, what we do we start with XN we make a we put them in a buffer this input data in a buffer length L plus N minus 1. So, these parallel lines come because all this data this data I mean this for the block of length L which is a block of our actually interest and this extra part which we added for doing the you know overlap shape. So, total length is L plus N minus 1.

So, this fellows parallelly come out I take FFT over it total length L plus N minus 1 let me give it again L1. So, L1 point DFT except for you showing so many lines I just do like this and suppose I also have because I am currently dealing with a particular block extended block. So, I know what the filter coefficient vector is for this and that will not be changed as long as I am in this block that will change only when I move from this block to the next block by the block elements update formula. So, I am having this 0 to N minus 1 here is my filter coefficient vector W and here we have 0. So, remember that is what we discussed for overlap shape method that is if I carry out circular convolution between this there is filter coefficients as it is 0 to N minus 1 not N minus 2 N minus 1 and remaining points L minus 1 points actually you know they are 0s and then we carry out circular convolution between these two blocks that is DFT here and DFT here multiply inverse DFT which in time domain is circular convolution.

Then first N minus 2 results are incorrect from here onwards so you also correct that is what we want where my point of interest from here to here. So, suppose this is my W and W if I take FFT with these 0s it gives me Wk then if I multiply DFT into DFT so here Wk somehow it is available suppose available then I take this out DFT into DFT then inverse DFT so I take L1 point i DFT which is actually by the FFT algorithm. So, then I get the circular convolution YN, if you take this output they are coming parallel you see if you put them side by side in a block from as if it is a sequence then from that YN you must discard the first N minus 1 points that is overlap shape method that is we discard the first N minus 1 output. So, in fact here only we can say if you take discard the first N minus 1 points and total is L plus N minus 1 that means I take the last take the last L points then equivalent if you discard the total length is L plus N minus 1 you are discarding the first N minus 1 points output which means from the circular convolution you are taking the last N points. So, from here to here from here these points they give me the correct convolution.

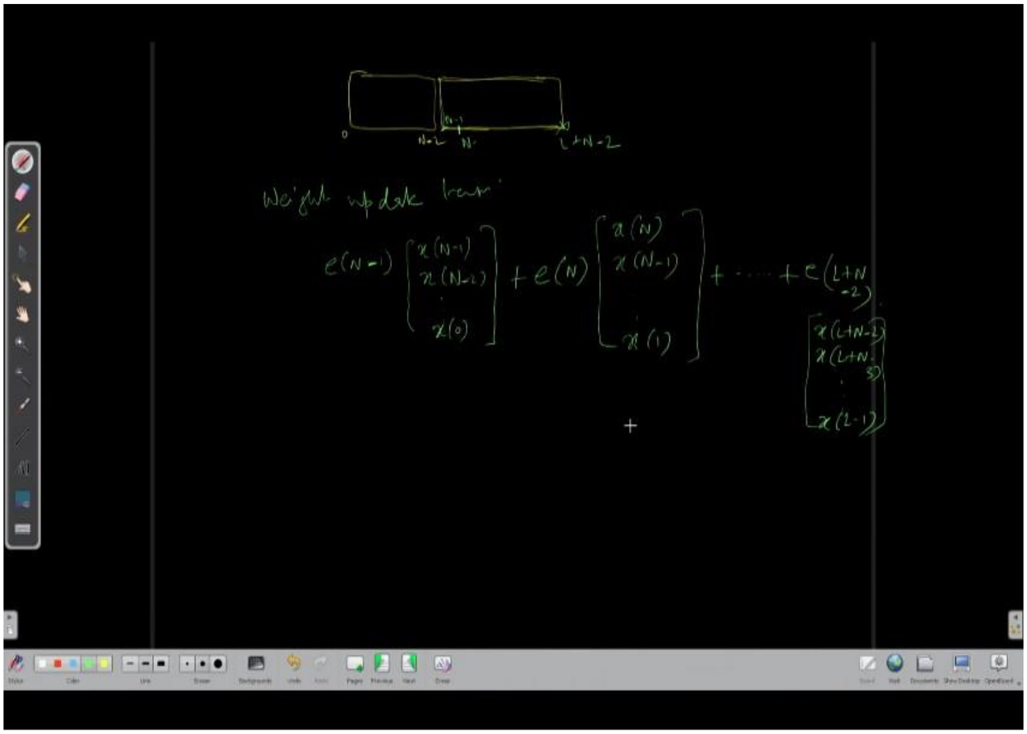


Then so suppose this is now serial they are coming parallely so I put them like this block you know one after another serially and then I subtract from BN plus minus so I get EN N from this point onwards N minus 1 dot dot dot up to L plus N minus 2 all right. Then I multiply this by mu these are very standard routine step all right. Now I do this and then I will explain why I am doing this I form a block in this block I add N minus 1 zero at the front at the front and then I take so it was L length L and N minus 1 zeros second length L 1 because I have added N minus 1 zeros in the front. So, then I take this in time domain so L 1 and why I am doing this I will explain in two diagram FFT all right. And this I come here and here I was getting x k right after the FFT x k so this I conjugate so basically; I get x star k here again let me write this because FFT output is parallel I multiply and then take L point so that L 1 point I DFT.



So, I go back to time domain to explain that weight of dead part now I will relate whatever I was doing to the weight of dead part of the block elevation algorithm earlier I took care of the convolution there is a filtering part by that overlap set method and all that. Now as you have done in the previous page this is what I have 0 to N minus 2 this is coming from the previous block overlap and safe part I will discard the output from here and then from N minus 1 to L plus N minus 2 this we have already done this is my block weight of dead part weight of dead part means previous weight or the current weight for this block current weight for this plus that extra term update that weight of dead term we have seen in the block elevation algorithm we will first take the error at this point N minus N 1 E N minus 1 into the data vector here data vector here is starting from N minus 1 X N minus 1 and another N minus 2 elements these are here X N minus 2 dot dot dot dot up to X 0 then plus then I go to this point N minus 2 N N. So, E N again data vector there. So, earlier it was N minus 1 now it is X N earlier it was N minus 2 it is now one step forward dot dot dot it is X 1 plus dot dot dot lastly I take at this point E L plus N minus 2 at this point the error. how is error calculated I have done the filtering at this point using the weight vector fixed for this block and then subtracting the output from the desired response at this point of time that is my error E L plus N minus 2 at that point that into the data vector here we starts at this index X L plus N minus 2 next is 1 less. So, N plus N minus 3 dot dot dot dot if you count total should be capital N because there is a filter length.

So, this one should be X L minus 1 this is the computation also I want to do by this DFT using circular composition and all that is what I was doing there ok.



Suppose I have got this data X N this is my X N this is my X N. So, suppose it is X 0 X 1 dot dot dot dot dot I go up to L plus N minus 2 all right. In fact, you can say X N minus 1 X N up to X L plus N minus 2 this is that block.

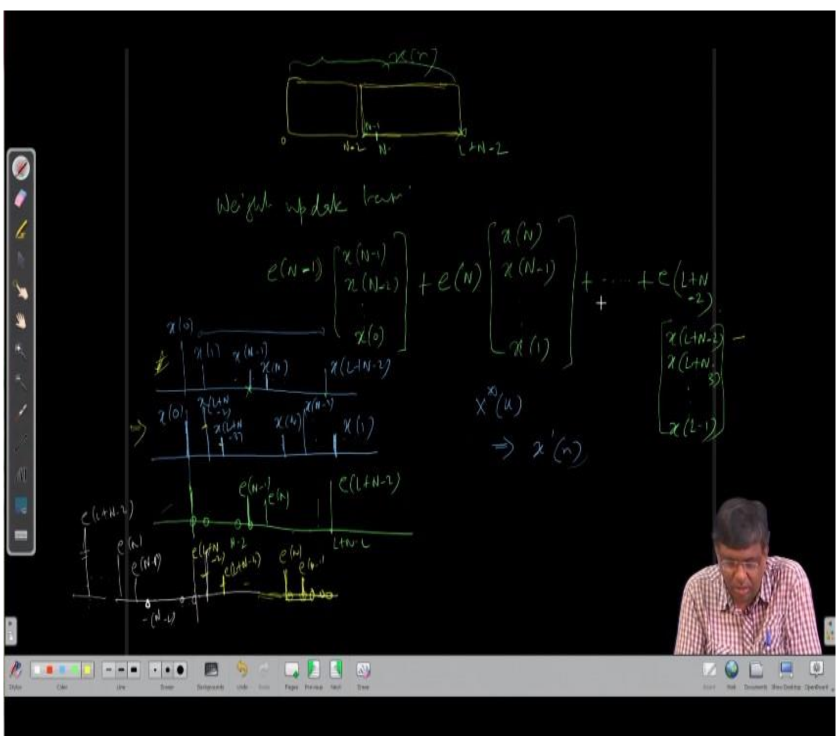
So, this is the data ok. So, if I take X star k in time domain it corresponds to maybe say X prime k X prime N. So, I know what is X prime N that is this remains as it is just get conjugated now for simplicity I am considering real data for your sake which means conjugation has no meaning that is why I will not conjugate which means X 0 will remain X 0 no need to conjugate because I am considering that all the data to me all X 0 X 1 all data to be real and for our purpose. So, conjugate has no meaning. So, X 0 will remain as X 0, but the remaining part this part will get flipped. So, next will come X this fellow L plus N minus 2 then will come X L plus N before that minus 3 dot dot dot dot.

So, this will get flipped this X of N will come before X of N minus 1 because it is flipped. So, somewhere here we will get this X of N there is your X N minus 1 and dot dot dot dot finally, X 1 at this point X 1 ok. This is what will happen because I have taken the conjugate of that DFT and then inverse DFT that is in time domain it corresponds to that is a conjugated DFT in time domain corresponds to this. And you remember I am multiplying after conjugation I am multiplying this with what in time domain, I form a sequence where the error started at N minus 1 then N minus 2 up to here, but I padded with N minus 1 0s there is N minus 1 0s in the beginning. So, 0 0 0 0 0 and then E capital N minus 1 capital N minus N capital N plus 1 up to this we have larger sequence and then I have taken a 50 of that a 50 into this a 50 and inverse DFT means circular convolution in time domain between the corresponding time sequence here and here.

Here it is that flipped version we have already discussed all right flipped version we have already discussed and then this thing I mean and here we already know what the time sequence is ok. So, now, we have to do a circular convolution of them to understand what is coming here ok. So, we know how to do this circular convolution first we plot that error part initially I have got 0 0 0 0 up to N minus 2 up to N minus 2 0 and then I have got E N minus 1 this N minus 1 is point E N dot dot dot this point and this point are same E L plus N minus 2 and 0s. To circular convolve circularly convolve what will I do I will make a periodic version of this and flip it and then keep moving to the right by one shift and then within this period 0 to L L plus N minus 2 see the I mean samples here samples here and term by term multiplication addition this we have done no need to explain again. So, if you do that what will happen is this just if you make it periodic and flip it this 0 will go to this side this 0s 0 0 0 0 up to here, that is minus it has plus N minus 2.

So, it will be minus N minus 2 and then all these fellows E N minus 1 E N this fellow will just go to this side dot dot dot dot E L plus N minus 2 and this thing will repeat. So, again here the same fellow will come here E then next will be E L before this whatever was you know L plus N minus 3 dot dot dot dot and after some time remaining 0s because you see this is one block right this is one block I started period from here. So, whatever I have up to this that entire thing will repeat here all right that is why from here to here here to somewhere here I am getting and then these 0s. So, 0s will come here all right. Now if you multiply samples from here and here sorry not here because I have taken the conjugate of DFT and then inverse DFT.

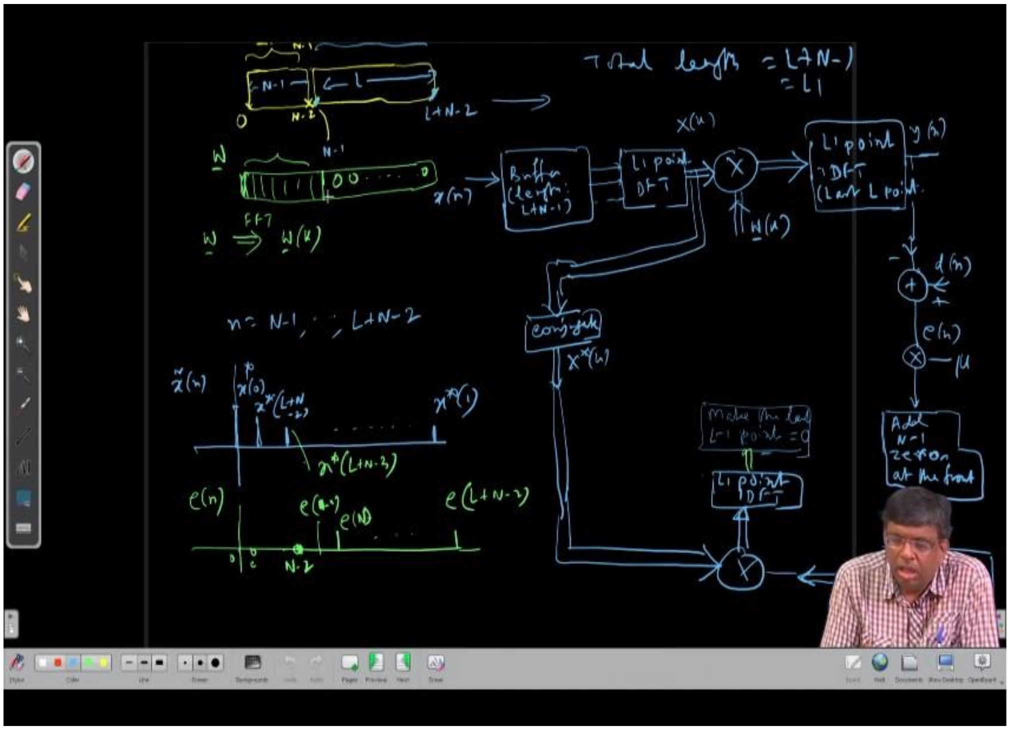
So, it is this sequence this sequence now you see this 0 multiplies this guy nothing. So, I start with E L plus N minus 2 into X L plus N minus 2 that is this term E L plus N minus 2 into this, plus E L plus N minus 3 into next term which is not shown here and dot dot dot dot then I have got E. So, this E N will multiply X N and E N minus 1 will multiply this is click very clear from here this is Nth here. So, this into this, this into this dot dot dot finally, it will be up to E N minus 1 E N minus 1 will multiply here actually since it is X N I should plot it here below X N is E N and below X N minus 1 is E N minus 1 then the 0s you can see E N into X N E N minus 1 into X N. So, you get the first row then you shift it to the right.



So, one 0 comes up one 0 comes in. So, E L minus L plus N minus 2 now multiplies next sample this sample that means, I am now here E L plus N minus 2 into the second guy now and similarly this into second guy this into second guy. So, this second row is evaluated and this you go on doing how many shifts because I need 1 2 3 up to N because filter length is N. So, I do it for 0 shift 1 shift up to N minus 1 shift that is first N shift get those results all remaining results are not required. So, there I will put them take them 0s all 0s. So, I will take the circular convolution I will take 0 shift output first shift output second shift output up to N minus 1 capital N minus 1 the total is capital N because I have got 0th row first row second row up to N minus 1th row total length is capital N because this is coming with respect to weight W if it is called W i ith block plus this.

So, this is length N. So, this has to be length N. So, I will take update of length N. So, 0 shift 1 update 1 term first shift another term I like that all right, but circular convolution will give me a longer result. So, remaining subsequent terms I will take to be 0 that means, what I will do here is this these I got the circular convolution.

So, I this is my time domain sequence. So, I take it as it is just making the last terms how many terms because convolution is L plus N minus 1 this is a total length out of which N is gone here. So, remaining is L minus 1. So, being the last L minus 1 point equal to 0.



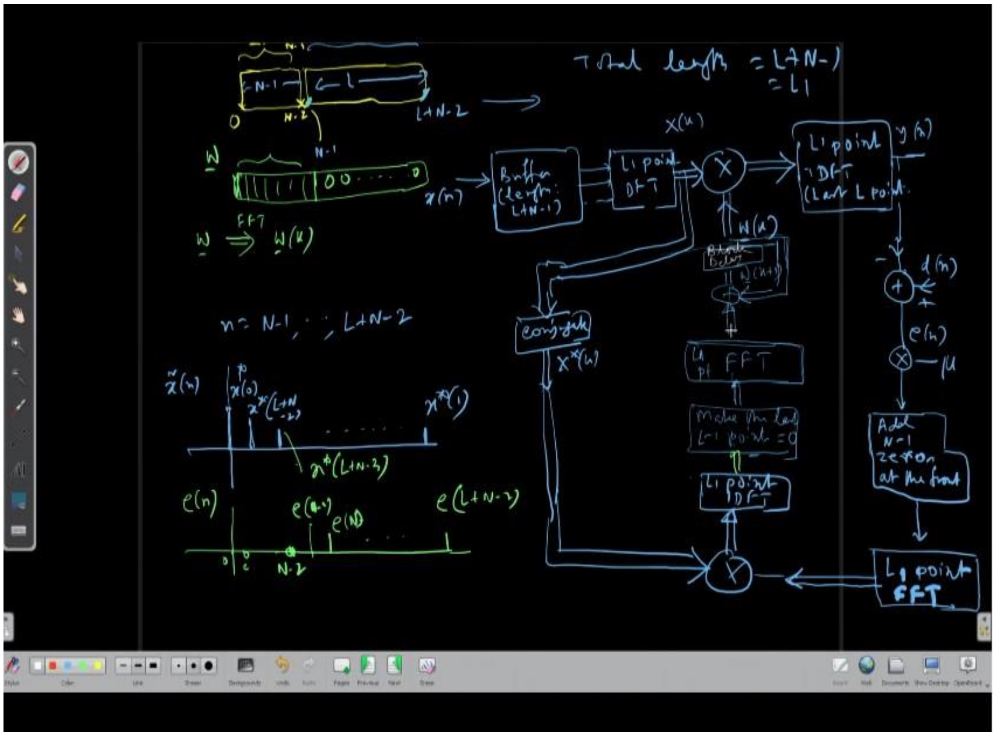
So, update term will be such that if it will be a vector like this of which the top part will have that update term as we calculated there is a as we calculated the update term and lower part this part will be here 0. So, that 0s plus 0s they remain 0 because even when W is updated for the next lock I want this part to remain 0s this is the this is what is required as you have seen for the overlap and same business right.

So, this part should not get disturbed into some non-zero elements right that is why this L minus 1 0s are here if it is the weight update term first N minus 1 first N points they give me weight update 0 to N minus 1. So, they get added with this. So, it gets updated, but last L minus 1 points they I am forcing them to be 0. So, that 0 plus 0 is 0. So, 0s are maintained that is what is obtained.

So, here what I get is in the time domain I get in the time domain new weight, but I am working here in frequency domain. So, now, new that is updated weight where I have got the data here we have weight values here and as before 0s here. So, this extended updated weight block on that again I apply a 50 because I have to go to frequency domain WK. So, I apply a 50 all right all a 50s are I am not writing them here all a 50s are L1 points L1 point even if it is not mentioned here there is only one point if it your I have 50 that is L1 point. So, this is my new one new weight new update all right because I am going into this frequency domain.

So, new update. So, existing weight with 0s plus updated update block with 0s in frequency domain DFT of this plus DFT of this will be new DFT this DFT of the updated weight I am repeating again this is my block which means the first part is the current weight vector and then extra 0s here that added with update term which we calculated here. Update term also has L minus 1 0s at this lower part and upper part has the update values when you add them in time domain you get the new weight update for use in the next block, we are in the frequency domain. So, if you take DFT of this updated weight it is nothing, but DFT of this previous weight plus DFT of the update term. So, DFT of the update term is coming here I am taking this 50 and this is the DFT of the current weight. So, I have to add them to get the DFT of the weight of the next block.

So, here I add. So, this gives me if you call it Wk this I call Wk plus 1 and then symbolically I say delay, delay by a block delay that is when I move to a next block then this fellow comes here again it starts doing the filtering operation and all that. This is the effective best implementation of the block elements algorithm.



I hope you understood you please go through DSP book by Oppenheim, Seifer or in your project you should want to read little bit more about circular convolution and think about what I said so far it should not be difficult.