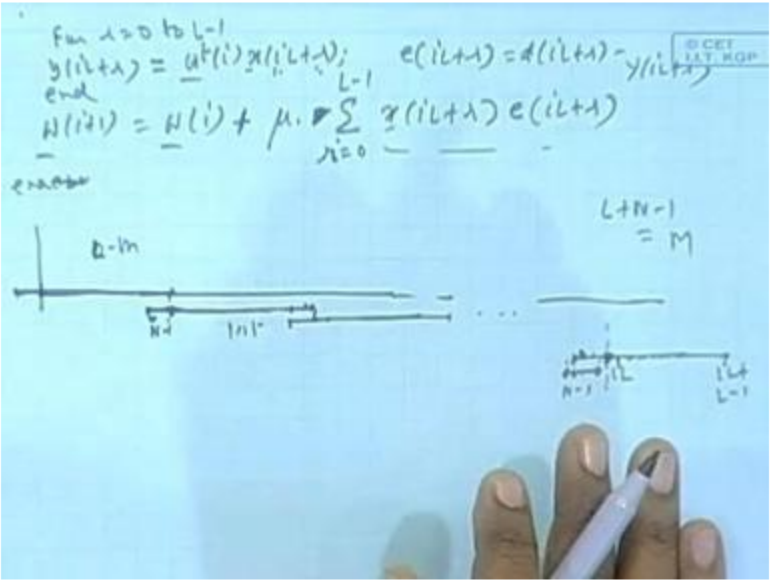
Adaptive Signal Processing
Prof. M. Chakraborty
Department of Electrical and Electronics Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 16
Fast Implementation of Block LMS Algorithm (Contd.)

So, I start at the point where I left yesterday. That is if you see this weight update part of the block elimination algorithm.

(Refer Slide Time: 01:13)



Sorry, this is r right r 0 to, this for the real case. You know what change will be there in the complex case. In the complex case this data vector will be as it is there will be a star across I mean one. In the filter part it will not be w transpose x n w transpose n, but w I times x n. That was the only difference we will extend this to complex case as well. I think that extension you can do easily from real to complex version of block elimination algorithm does not difficult, that the know LMS of see now it is we do be...

Any way let us see how this can be done, before that remember one thing. Yesterday the there was a filtering part also. If I write the entire algorithm y is L plus r was what, w transpose I x I L plus r and then you calculate e I L plus r as this look pretty simple minus y I L plus r.

I am not writing the full algorithm that for r equal to 0, 0 to l minus 1 if you want, for r equal to 0 to l minus 1. This is one loop, this will be outer loop this 1 I, this part of it.

Remember one thing what we discuss yesterday, in the filtering part we will follow what will happens safe, if the what will have happen safe what we do. How will the blocks then be formed that is we have to discuss that first.

Suppose, this is the block of length L then and what is the filter length filter length is capital N. So, I will take this is a zeroth block length L I will start the next block here somewhere here. So, that there is an overlap of N minus 1 samples. This will remain next block first next block will be here dot span. Suppose, it was earlier 0 to L minus 1 and then next could have started at L now it is starting at N minus 1 to the left of L.

Then it is going up to L. So, L plus N minus 1 total new length is L plus N minus 1. For the first block you can assume some 0's here because is length is L nothing no overlap portions 0's here that is ok, but for all the intermediate blocks the length is L plus N minus 1 you can call that capital M .

So, earlier it was length L, now M minus 1 0's L plus I mean length M. Then again it was starting here going up to L. I am appending N minus 1 data from the last N minus 1 data here. Then here also it was to start at what 2 L automatically I am taking last N minus 1 data from here it goes up to these, so on and so forth.

I will be using say that way dot, dot, do,t dot for the I th case this is I L and I am having extra N minus 1 points and it will go up to I L plus L minus 1. This is my I th block for this block I will be using w i as the filter. I will be using convolution at this point, this point, this point when you convolve with this point or this point or this point or even here. I get this data from the past when I am here it is assuming some 0 values, but in practice in real life there is data from the previous block.

We have discussed yesterday. So, at all these points I get wrong result. Only from N minus 1th point. This is 0 to N minus 2 total number is N minus 1. So, this is N minus 1th point from counting from here. From that point onwards only I get correct result. So, I will correct, hold that result that result showing this result, but before that I have to do the filtering for the filtering part.

For the filtering part, what I have to do I am interested in filtering here, all these points that is this equation, but if I want do by circular convolution and all that you know you want to use FFT. I use this part at all these parts and I will be using circular convolution by DFT at all these points whatever comes, I will throw that away. That is wrong that is not even linear convolution if just this block and the filter block was given, the linear convolution result will not come at this points.

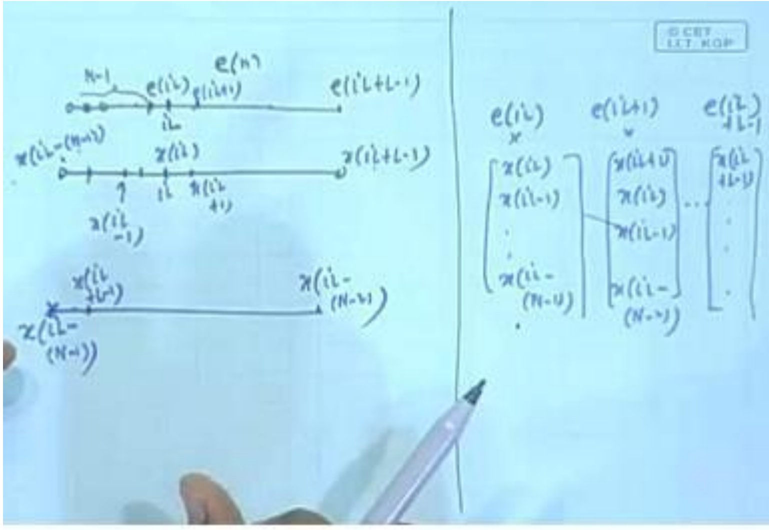
Even otherwise if they are correct either context of the entire data stream these are not correct because at it is assuming 0 to the left. Whereas this is just part of the continuous data stream. So, the real convolution doted to the left convolution point would have come in. So, that are thrown away. So, I would get convolution output here and that is what mathematically is this.

Convolution starts at I L plus 0 I L plus 1 and start at I L plus 0 then I L plus 1 I L plus 2 all these points convolution output. So, my current filter length is dotted length is not L, but L plus N minus 1 which is M. So, filtering part is all right, if it is complex value filter coefficients instead of w transform you will become w harmesian. Note that change also is that understood job there. Now, come to the weight update part. In the weight update part see the data structure x. So, put r equal to 0.

I will write separately on a page, but just here you see the data structure I L, I L means I Lth point and I L minus 1 I L minus 2 up to these fellows also the same points. There is a dotted vector at this point will consist of data up to this then I L plus 1 means you have to go to the next guy. So, data up to 1 less from here so on so forth. So, remember in this weight update part what are the data samples involved, the data samples present in this block.

First how because in x I L vector I take this entire vector and then this point and progressively you can go to the right. So, that means, both for weight update and all this 1 chunk of that 1 block of data is enough for me. In the filtering part I will do a circular convolution by FFT, but I will throw away the first N minus 1 M N minus 1 samples. Then from I Lth point onwards circular convolution wise also I will get the result as given by this equation. This is the convolution, but I L plus 0 I L plus 1 at those points. In the weight update part I will now show you how to do. So, please see the thing.

(Refer Slide Time: 08:59)



You have got one sequence say e n, this sequence suppose this point is I L. So, at this point you have got e I L dot up to say e I L plus L minus 1. Suppose, I have 0 0 0 dot 0. What is the how many 0’s N minus 1 0’s I put I put N minus 1 0’s. So, what is the length of this vector L plus N minus 1 that is M capital M same as that data vector. Remember in the weight update part if you see the weight update equation put r equal to 0. For your convenience I have to draw this way e I L, this is a separate story e i L is multiply what x I L vector.

What is x I l vector, x I L x I l minus 1 dot, these are this L part this part. I L minus N minus 1. Then e I L plus 1, this is multiplying, this is multiplying this vector. Is going 1 further, this is coming down like this you can see this this kind of diagonal moment dot, dot here it will be again 1 less dot. Finally, e I L plus L minus 1 last fellow will be multiplying whom, is this portion visible in the screen.

Student: No.

Is this visible.

Student: ((Refer Time: 11:13))

What no either yes or no okay, dot, dot. So, I will be multiplying these vectors by those e values and then add that the theory. Then I will get a real thing vector that is what I have to do in the weight update computation. That is my main job first I am going considering only the first entry of the real thing vector, how will that come e I L into x I L e I L plus

1 into x I L plus 1 dot added. The next 1 will be e I L into x I L minus 1 e I L plus 1 into x I L dot, dot added. That means if I graphically see e I L here. Suppose, here again I have my x I L those in the I Lth point, then I have got and those data x minus here. Here it is x I L minus 1 are you finding, it is okay. Can you understand what I draw.

Student: Yes

Now, suppose this way this I just multiply sample wise this 0's will multiply this past value. Only e I L into x I L from then onwards I will get e I L x I L e L e I L plus 1 into x I L plus 1. This is e I L plus 1 this is x I L plus 1 dot, dot up to this. So, I will get the first entry, then if I shift these to the right by 1 and let any garbage come from behind, but if I shift by 1 and then again you do sample wise multiplication.

These 0's will take care of this same portion of n minus 1 length data. E I L will multiply x I L minus 1 and this guy will go out next the next fellow will come. So, I will get the second entry. Secondary entry then I shift it further again let any garbage come. So, I will go on shifting how many times just n times 0 shift 1 shift up to N minus 1 shift. After N minus 1 shift this fellow this this guy will come below this e I L.

So, e I L into these total N 0 shift 1 shift N minus 1 shift 0 shift, these guys are at the 0th position. First shift at the first position N minus 1th shift. At the N minus 1th position N minus 1 this is N minus 2 because length is N minus 1. So, this is N minus 1th position fine. After that I would not shift or even if I go and shifting and do this business I ignore those results, I do not need them.

So, you remember that this is something what we do in circular convolution, what we do in circular convolution 1 sequence I hold as it is, other sequence I make a periodic version reverse. I shift I will chop up the right and left and sample wise multiplication. So, that means, what I see with 0 shift this is the 0 shift case, but after reversal. After reversal then that is why if you that is why if you multiply sample wise you get the first entry of the vector.

So, that means, after reversal 0 shift is this vector this sequence. That means already 0 shift. So, what are the original sequence I have to reverse it this direction. I will get a portion on the left make a periodic version bring it here. Then see it only in this window. This simple engineering, nothing else those who do not understand, they do not deserve to understand this is very simple.

I will reverse it because this is a reversed version to go to that I will reverse it. This portion will go to that side. Now, using periodicity these are periodicity things. This whole thing will be a periodically repeating. So, I will get 1 chunk in this. I will just separate out forgot about right and left that is my original sequence. If you proceed for circular convolution you will get these that is make a periodic version reverse it.

Then without shifting what we were seeing in the window multiply we get the first entry then shift it once. In the circular convolution it go on shifting even beyond the N minus 1 shifts because we have to compute the entire thing, but I ignore those result and I do not need them all right. That means originally what is the sequence, x here this as it is can you tell me if you reverse it what will be before this. This guy will go this is.

Student: ((Refer Time: 16:24))

Yes, I L plus L minus 1, see I am not showing that where if I have to really do like a, you know school level teaching I would reverse it. I will take it to the left hand side identify it is sample then pick a periodic version of that. See what comes here then only take that out. Those intermediate step I am not going into. I am asking you to imagine, are you understanding. If you flip it through this side this fellow will be the starting guy. Then progressively, if you flip it I L minus within bracket N minus 1. Next is within bracket N minus 2 N minus 3.

So, this will be I L minus N minus 2. This remains as it is next guy from this guy onwards it is flipped. So, this was the right most ever I mean the adjacent guy to the right this will be the adjacent guy to the left. Next adjacent guy here next to the left like that that way and the entire thing will be pushed back here. Then my claim is if I use this if you have a circular convolution between this sequence and this sequence.

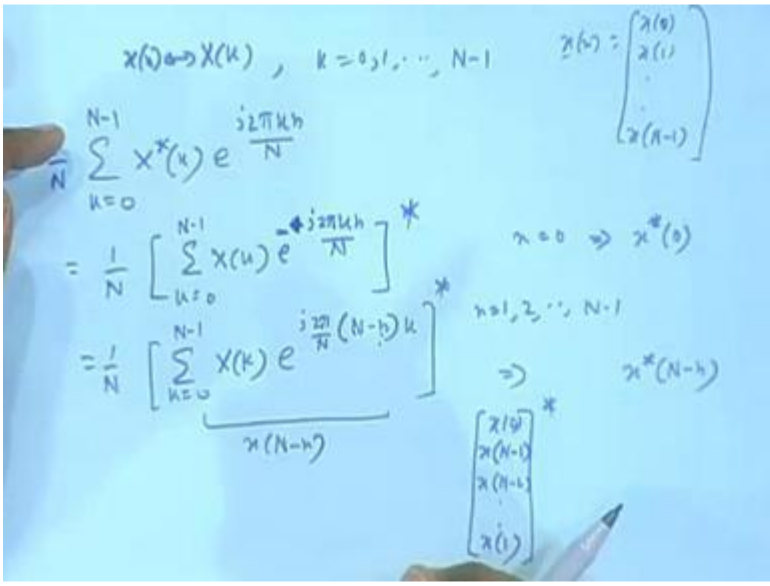
Whatever be the output, output is over this length, but there you take only the first N minus 1 first N 0 to N minus 1. After that you whatever result you get forget that then you will get that weight update thing. Is that understood or you want me to repeat I do not mind repeating this part do you need some clarification here. You understood or not.

You can easily check know you flip it. Then I make periodic version of it reverse whatever you see on the left hand side that will come here. This guy will remain as it is, this will come to the left hand side. So, that is again by periodicity coming here I L minus within bracket N minus 2. So, here it was N minus 1 N minus 2 that way. This was the right most ever I mean the right ever. Immediate right ever, this will be the left ever and that will come here, all right.

So, that means, I have to do a circular convolution between the 2. What is the DFT of this, I can take the DFT you give me the e sequence, error sequence I added some extra 0 may be make it a bigger vector of length capital M. I can take a M point DFT on that, but how about this guy. It has got some extra data.

Earlier I was leaving happily with L point data vector, now some extra garbage has come in that kind of got flipped and what are has been changed and all that. So, what do we do to that, that is the question. Now I show you some beautiful property of DFT. Suppose, now this is a separate story. Even if I use capital N symbol here that is not the N there. I show some property of DFT, since I was using capital N there I am using capital N. That is not the N used in that filter length of adaptive block adaptive filter. I am showing a separate property of DFT.

(Refer Slide Time: 19:52)



Suppose, you are giving x k for any length capital N sequence x k, DFT k equal to 0 1 dot N minus 1. Original x n was if you put in the vector form x 0 x 1 dot, dot. You have to give DFT. Now, it will be looked I want to play around with this coefficients. I will take a conjugate of them and then take inverse DFT. I will get another sequence, how is that sequence related to x n not exactly it is the result.

Suppose, I take x star k and I take I DFT of that then what will I get. So, fine I take I DFT formula I know this class you are know. I DFT at a particular index time index of your choice small n within the range small n equal to 0 to small n equal to capital N minus 1. The time index fixed, this is a quantity if there are no star I know this would give me the x of n small x within bracket small n, but there is a star here is a problem. So, let me do some mathematical you know manipulations here.

I will bring back the star here, no problem and e to the power minus j 2 pi, but in the DFT there is e to the power plus term not minus. So, what I will do, I can write like this, it was minus firstly, for you sake let me do it in 2 steps. Before I proceed further this is summation over this, your N minus 1. For N equal to 0 what is this sum e to the power 0 1.

So, independent whether it is plus sign or minus sign does not matter. So, what you get is x of n only, x of n because there also n 1 was 0 added to the power 0 1 x of n that is x of 0 with n is 0 conjugate of that. So, n equal to 0 these leads to x star 0 and for n not equal to 0, there is a problem because I can I really have to I cannot this is present with minus sign. Otherwise it become 1 independent of whether it is minus or plus.

Now, that fellow may not goes. So, that I apply the simple trick, I add j 2 pi into capital N by capital N into k, e to the power j 2 pi into k that is always 1 and that k I am writing capital N by capital N into k here n equal to 1 2 up to N minus 1. Then the star. So, n equal to 1 means capital N minus 1 n equal to 2 means N minus 2 dot, dot n equal to capital N minus 1 means 1. This is what x of. So, this give you x to x of. So, what is the sequence is very simple inverse DFT, I put the vector form original vector was this.

Now, it will be starting guy will be x 0 there will be a star, star I am putting outside. For the time being I will be really going with real values things. So, x 0 that data star for our case will be x 0 only to start with. Then it will be, that means, if have this sequence if I take this DFT, it will be x k star.

I went this way, but relation is 1 to 1 and unique. So, that means, if I take this guy and take this DFT I will get back x star k. Now, come to this. If we look at this thing 0th guys remains as it is, but the last is just flipped this is just flipped. N minus 1 goes to 1 and 1 goes to these. You understand flipping know, if you just flip it and then push it back. So, that it fits here, no circular shift by the way, this is a simple flipping, x 1 to 2, 3.

So, 1 goes here, it goes to the opposite direction 1, 2, 3; it was coming down like this coming 1 2 3 4 like that. Now, it will go up 1 2 3 4 like that, but only up to this last, but 1 position or first, but 1 position. Now, come to this guy these are sequence which I am going to do circular convolution. Then only periodic version repetition and then you see this part and you get that sample wise multiplication, immediately you get the weight update formula.

So, this is the sequence which I have to carry out circular convolution between I mean, with this guy on whom I have to carry out this. There we see DFT domain, DFT of this into DFT of this. DFT of this is either samples I put I did up to 0 vector take a DFT. This have to compute, but here if you ask me again to compute the DFT again, for this vector it will more computation, but I am saying no.

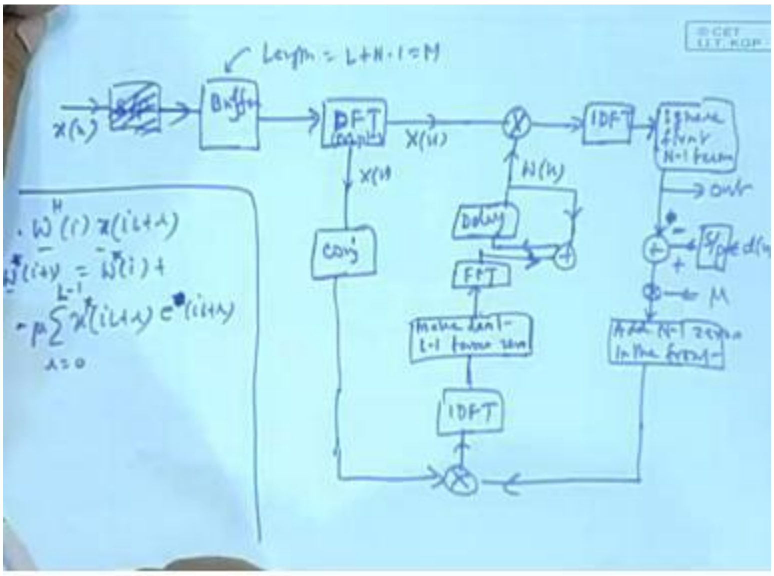
Look at this thing, if you consider this to be the starting guy here also here also same, but otherwise this is a flipping, this fellow was the last it came here. This fellow was this I L minus within bracket N minus 2 here this has gone here. Since, I am dealing with real case there is no star if I take star and say star is useless, but still you can put star. Then if I take a DFT I will get x star k, where x k is the DFT of what this basic vector simple block, this blocks.

Take a block and why where do I need this DFT in the filtering part overlap and same method. Same DFT I do not have to re-compute the DFT of this I know reordered vector. I computed DFT only once here that is how the computation will be otherwise 1 more DFT would have been required. This I need you know because I am doing the overlap segments of filtering. So, the circular convolution thing was there.

Throw away the first N minus 1 sample output and all that. Add 0's this filter with filter vector is length N and this is L M. So, add those difference of 0's and then circular convolution that I did it yesterday know. Overlap and same method filter length was less capital N here and this total length is what capital M. So, you have to have extra 0's. Then if you do circular convolution and throw away the first N minus 1 samples you get the correct output.

So, that time I need circular convolution means, what I will be doing in frequency domain DFT and then finally, inverse DFT. DFT into DFT then inverse DFT of that. So, I have already compute the DFT this DFT once now claim is here to compute the DFT of this vector I do not need to re compute the DFT of this guy. Earlier I got x k capital x k now it is simply x star k. So, then you know it amounts to a very simple thing. Let me draw the block how will you, do this business.

(Refer Slide Time: 29:13)



The input x n is coming you do this initial part serial to parallel serial to parallel. Then buffer, buffer length here is equal to L plus N minus 1 equal to M. The meaning of this buffer is, you see you are taking this data making it parallel actually it is 1 should 1 not use this serial to parallel you know because serial to parallel we use in the context of binary means this serial to parallel. I do not know why here it, it should not be there according to me because you just simply store the data in a buffer. So, you can ignore this straight because into buffer how many data I am taking L also.

I am storing the last N minus 1 data of the previous block. So, I am forming a bigger vector here. Then first thing is I will be taking FFT of that how many point M point M point for the filtering part for the filtering part. What will you do for this block, I took the DFT of this I took the DFT of this are you and filtered weight.

Suppose, I know the filtered weight has been updated I know the filtered weight for this block I th block or w i is known what how we do the convolution w i here, and then you have 0 0 0 0 0. Then the 2 lengths are equal then do circular convolution ignore the first N minus 1 remaining 1s are correct. So, that means, what have to do this FFT I found out. If I know this FFT suppose, by hook or crook I know this vector. It is FFT is known somewhere. Then the 2 FFT's of the DFT's will be multiplied and I have to take the inverse DFT ignore the first N minus 1 sample there is the filtering part.

So, suppose this is I do this way x k is here x k here also. Suppose, I know the wk and this wk means after appending 0's. You take the vector then take it is DFT that DFT is w k suppose I knew that. So, I will do that and then I have to take I DFT of that. I have did it on M point DFT only once. I hope you understand that everywhere when I write DFT or I DFT it will be M point.

In fact, you instead of the FFT you can call it DFT also because I am writing instead of IFF I am writing I DFT. So, in case you are confused DFT, but it basically will be you using fast algorithms only you can implement them not just primitive composition. After this I DFT you ignore, I am writing this way look like something else. Ignore first N minus 1 term. So, filter output is available here filter output is available from this point I L to this point.

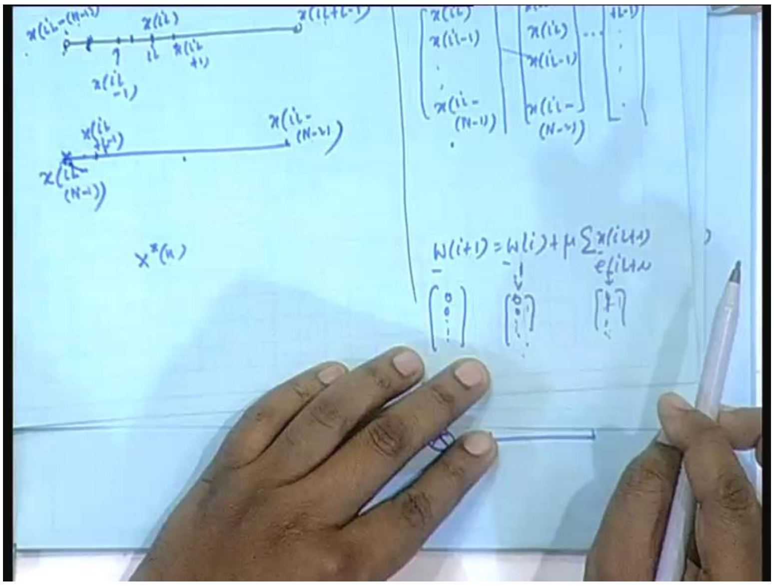
Filter output is available here you ignore those terms first N minus 1 terms. On the weight update side what I said same x k I can to use, but I have to use a conjugate of that. Then only it amounts to what DFT of this vector which is flipped except for the first element everything is flipped. Assuming a conjugate, but there is no real data the conjugate has no meaning for that if you take the DFT it is nothing but x star k.

So, I simply take x star k, x star k times that is the circular convolution between these and these error sequence with a 0 here. If you carry that out and take only first N cases, rest will be garbage will come and some garbage will go into and sample wise multiplication error, but I do not need them. So, take only the first N then I will get that weight update to the update term in the weight update formula.

That means here I conjugate, but this weight has come. So, after ignoring first term you get actually vector parallel form. So, you can assume. So, this is an output. You can take out here out, but also I have to find out now error vector error. So, sorry this is minus this is plus d n minus y n that is the error, y n output is coming the parallel form in a kind of a total vector has come out.

So, d n also, if I write s by p are you only been buffering not that is d of I L d of I L plus 1 d of I L plus 2 dot, dot d of I L plus L minus 1. So, this error. So, I get at error sequence error sequence which kind of error sequence from e I L to this. I have to know add extra 0’s. How many N minus 1 0’s in the beginning. So, add N minus 1 0’s in the front. So, again length is L plus N minus 1 equal to M here. This M point, now you take the DFT of that that 2 DFT s are to be multiplied. There will be having multiplication by mu also somewhere here, mu I will do later 2 things multiplied. Then I DFT. So, by I DFT what you get circular convolution between the 2, but I told you between that 2 I will written only the first N 1’s remaining 1’s I will put, say 0’s.

(Refer Slide Time: 36:41)



If we see why 0 because my weight update formula is all these are length L, sorry length N length N, but remember in that circular convolution filter I mean I added N minus I mean I added extra 0’s. How many L minus 1 0’s total length is L plus N minus 1 length N. So, L minus 1 0’s I added. Then only I took this DFT that DFT I used in the filtering part. So, this vector I will add here extra 0’s to all of them.

Here 0 0 0 I will add here also 0 0 0 here also. I will get therefore, 0 0 0, are you following what I am saying this notation. This vector as it is length N I will have L minus 1 0 L minus 1 L minus 1 0 here L minus 0 in the bottom. If you add them naturally here also in the last L minus 1 positions will be 0 that should speak because if I take the DFT of that that can be used in the filtering part is the circular convolution. Whether it was needed that last N minus L minus 1 guy should be 0.

See how beautiful the thing is things are fitting in all things know all directions. So, in the weight update part, then to carry out these I told you that I will take circular convolution between these and these, but I need only the first term remaining 1s I do not need, but I put it 0's there. So, that this 0 vector comes below. Suppose, the previous block previous, I means weights have been updated like that, the last L minus 1 guys are 0.

See, if you add, if you do the update these also will have 0. Now, DFT of this is DFT of this plus DFT of this, but DFT of this means what. This sorry, yes that is correct. So, what I will do. Let me first of the write down I am making one mistake. Let me first write down these then make last how many points.

Student: N minus1.

No, L minus 1. In this figure I would explain that, whatever explaining yes. So, I got this term you see I got this term. I did the circular convolution how by the 2 DFT and then I DFT I got the circular convolution between the 2, but their I am saying in time domain I do not use I use only first N points afterwards other terms are no use. I do not, but then I am deliberately putting 0's there I am putting 0's. Then take I am I am taking DFT of this. So, DFT of this plus DFT of this is the DFT of the next future weight vector.

So, the future weight vector. So, there will be a loop now this is again schematic you know. This is delay actually it is a block delay. Not 1 sample delay say block delay because filter wards are held constant over a block. So, it is a block delay. How to draw this, I mean this is the space is too less. It is becoming very clumsy somewhere I should have multiplication by mu. May be earlier only, here only you do multiplication by mu my drawing is a bit clumsy.

Please excuse for this, you know I am trying to rear a space what I am doing is current with this k means mind you this k is the FFT DFT index k. This W actually for the I th block, there should have been a I index here, but in this figure in order to avoid complication as not to bring in so many symbols I have avoided, but here everything is with respect to the I th block. So, the W that is shown that is from is I th is our k only indicates it is in a FFT domain.

So, FFT of these for the I th block that that is the correct 1 which is the used for filtering part in the filtering part. That plus this update term DFT of this entire stuff has come from here they are added held back for future use. After a block then only it will appear here. This is the first algorithm you see how I mean by cleverly using the data flow structure and all we have red reduce the total number of DFT operations.

If it were a complex case still this will work fine is the complex case in the complex case what we have. We have W harmesian I I L plus r this kind of thing, for r equal to 0 r equal to 1 r equal to 2. That is one thing in the filtering part that was the term adding the weight update part term it is x I L plus r e star I L plus r e star would have come plus this. Suppose, since in the filtering part you are not taking W I, but you are taking W harmesian means basically column vector to row vector, but actually conjugate of each element is used.

Suppose, in the weight update itself I take the conjugate. This time domain operation, I we know weight update itself I am interesting this, because I instead of finding out W I plus 1 and then taking the conjugate. Let us update the conjugates only, suppose I do that so; that means, this is star and this star will go this will come here. Now, remember 1 thing I told you that this property said what that actually be there will be a star here. Then I said it is a real value because star has no meaning, but when this is complex then the star comes in. Data vector has the star. That means, in all these convolutions in all these convolutions is there will be a star everywhere on the data. Star, star, star so that means, data conjugate and then this flipped version. With the star and flipped version, if you take DFT then you get x star k. Earlier I bye pass that saying that why data is real. So, conjugating data has no meaning that I, but actually when data is complex you remember I proved it.

I proved it that, if it is x star k the corresponding data vector is not just flipped version. There is a conjugate sign also that time I took real data. So, I found this has no use, but in the actual case when this complex, there is a conjugate here. There is a conjugate here. Again there is no problem, because in the weight update formula, if I put a star here already data becomes conjugated.

So, that means, in all this circular convolutions after flipping and all, star already comes in. So, again I can take x star k to represent the DFT of this. So, then that times this fellow will be what will be already W it will be represent that DFT of the W harmesian vector that is DFT of this side left hand side. Conjugate was already there DFT of that. That DFT and this DFT circular convolution and I DFT, it will correspond to the correct linear convolution with W harmesian these thing.

This W harmesian comes back and therefore, mu into this part should come. So, that is coming because of this, this conjugate times this and I DFT. I told you that amounts to what circular convolution between the 2 which has star here. So, this part comes here and added and held back. This is the very exciting result. That is why I gave you, this is very shows how you know how this elegance of DSP how things fit into each other and you know I mean you can reduce complications, computational complications.

Is this understood. Well actually after these, you have exam I know, but after this after this what I have to start is. Again you know mathematically very exciting, it is a new topic. That is a new approach to adaptive filtering with a more general, very general fundamental approach. I would say very exciting because that algebra is really great that is vector space treatment. So, I do not know how much vector space you know, but I will be covering some good deal of linear vector space call actually Hilbert’s space here know.

I will go up to orthogonal decomposition of subspaces and all that will be in the context of towards the end in the context of Hilbert’s space of random variables. Then this again will take care relook into this adaptive filtering optimal filtering as orthogonal projection. I already mentioned that, but that was using some analogy in the 3 d world, but now this time will be revolve us. From that onwards I will be building up some other kind of adaptive filters using that frame work.

That Hilbert’s space frame work one of them is a lattice filter. Lattice filter has the advantage that you know I mean here you have got ordered fixed. Filter length as either n capital N weights only capital N number of weights. If you want to see, how the filter performance would be if I had used N plus 2 or N plus 3 number of times.

You have to again re do the redone the elimination algorithm and all that, but lattice is 1 thing which is order recursive. These are not only time recursive from WN to WN plus 1 recursive in time lattice filter adaptive filter I am talking now. It is recursive both in ordinary lattice is recursive only in order non adaptive lattice a lattice filter actually is one realization of fir filter .Forget about adaptive lattice suppose I have got I got fir filter. So, low pass filter or high pass filter fifth order.

We have got a fifth order lattice also fifth order low pass filter means fifth order lattice. Suppose, you want to say that look if I had designed the sixth order low pass filter, how would it have been. Then you have to again redesign, destroy those five coefficients you used earlier you get a redesign and you get new six values.

You see this, but in the case of lattice it permits you to retain those five I mean fifth order lattice means is a cascade of five basic stages. There is a basic fundamental lattice stage. You just cascade it five times you get fifth order. So, to moving to sixth order you do not have to destroy those five you just cascade one more, another one.

So, together it will be a sixth order. So, you can tap at the first you can tap at the second third fourth fifth everywhere using first order performance, second order performance third order performance, fourth order performance simultaneously. There is lattice filter, but as a lattice filter like fir filter can be non-adaptive then can be adaptive and why it should be adaptive we have all explained.

Similarly, lattice can be non-adaptive non adaptive I am not considering, but it can be adaptive also. So, I will basically must choose adaptive lattice, but there is another region, another you know very what to say sound application of lattice filters. There is in the case of speech processing. Do you know this auto regressive modeling a r process a r modeling in times it is modeling.

Some of you may know, some of you may not, which is useful in power spectrum estimation also. There is one kind of model called auto regressive model. It is a model that has a transfer function with only poles and no 0's. So, it is transfer function will like this. So, if y n is output and input is x n, then your y n only pole no 0's. So, y n plus may be A 1 into y n minus 1 plus A 2 into y n minus 2 plus dot, dot plus A p into y n minus p A p into y n minus equal to some x n.

This has no 0 only pole. This kind of models are called auto regressive models or pole model. You know I, it can be shown that many random processes most of the random processes can be modeled. As though they are generated by driving a model like this with a wide noise. These all very good statistical theory and all that. Similarly, you can have a defined models also. Moving average model, where it is only 0 y n is some b 0 x 1 plus b 1 that is FIR filter b 0 x n plus b 1 x n minus 1 dot, dot b q x n minus q. Only 0 all 0 model called ma model called moving average model.

There can be a normal pole 0 model called aroma model, auto regressive moving average model. You know aroma we do not we are not particularly excited because the corresponding equations are non-linear, because input is not known that white, white known samples are known. So, if it is b 0 into x n b 1 into x 1 minus 1 b 0 b 1 model parameters are ought to be identified.

So, we do not know them and input x n, x n minus 1 drive wide noise samples they are not known. So, product of 2 unknown it makes non-linear all the algorithms will go away and is all approximate, but in the case of a r y n times a 1 or y n minus 1 times a 1 y n is the filter output is the data I am receiving.

I am modeling that. So, those samples are known to me. So, that is here it is a linear product it leads to beautiful fast algorithms and all and very good properties. This is used for modeling many random processes, particularly speech humanity it has been established also the way we generalize speech you know. Actually in our voice box here there is an air model air pore actually there is a filter.

It parameters change, number of poles again pole location, everything change. That is excited by a wide noise produced by ours. Corresponding random processes generated which gives rise to some sound or other. So, for all speech modeling, speech characterization, compression and all this e r modeling is very useful. You know this LPC know linear predictive coding this is related to linear prediction this coefficients are identified by linear prediction the entire thing is to identify the model.

This is all related to air model this lead it to lattice filters. So, zeroth order model first order model second order model third order model. Similarly, zeroth order lattice first order lattice second order it leads to lattice filter. So, from the next class, I will get into these more I would not say rigorous, but or may be rigorous is the word is mathematical treatment using.

They are there will be nothing analogous nothing based on analogy and everything you know, we mean visual sound logic, but again without requiring any advance math and all. I will spend quite a few lectures on this Hilbert's space treatment, vector space treatment, vector space with a dot product is called a Hilbert's space. Dot product is something we call inner product actually. So, that we spend on these from then onwards after that I will go into lattice, because there are other version of deriving lattice, but I do not like that, because mine is a unified treatment. When you go to recursive least squares not only elements recursive, those kind of adaptive filter same approach can be used there. Projection based orthogonal projection based approach. So, that is all and good luck for your mid sem after.

Thank you very much.