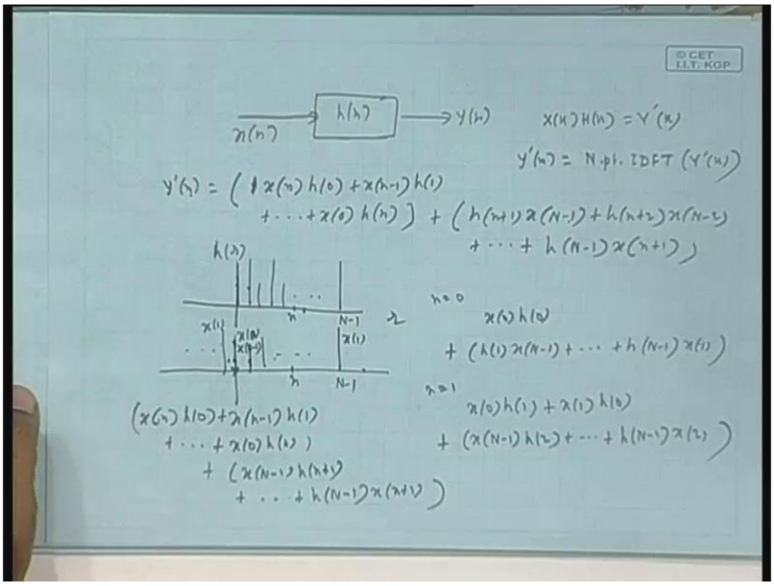
Adaptive Signal Processing
Prof. M. Chakraborty
Department of Electrical and Electronics Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 15
Fast Implementation of Block LMS Algorithm

Last class we are discussing DFT. In fact, we have discussed block LMS algorithm and I told you that I going to a fast implementation of block LMS algorithm using FFT. Then I started with the definition, I mean theory of DFT. How DFT comes and then I said that if there are say 2 finite length sequences x n and h n, one of length capital N another of length m capital M and they may not be same then x k.

(Refer Slide Time: 01:29)



If you take say if I said that if there is x n, and there is h n and there is y n. Suppose, x n and h n they are of different length, but I add 0's to make them equal length I started I add 0's to that some extra 0's that both are same length that length is equal to capital N. So, I take n point DFT n point DFT multiply the 2 DFT, but that will be again a sample of the DFT of this, but if I take the inverse DFT. I DFT of those samples I do not get y n because y n is a longer sequence its range is not capital n, 2 n minus 1 is the length 0 to N minus 2.

Then question was even if, I do not care for that I still persist with these. That I will do x k into h k and take I DFT what do you get for y n I get an n point sequence, but what is it. Is it how much of it is uh coming out to be you know linear convolution and how much is the error and all that. So, you work it out. We found out that, if you do x k h k and call it say y prime k and take end point say y prime n as N point I DFT of y prime k.

This y prime n expression we found out. Remember there was a linear convolution component and there was another component y prime n was, one was this say, may be you can write in these way what was which do I wrote x n h 0 or x 0 h n, x n h 0. These are the linear convolution part. Then there was another term as h crosses go beyond n plus 1, you have got x N minus 1 summation should be small n plus capital N.

Then h n plus 2 x n minus 2 plus dot plus h n minus 1 x, what n plus 1 is not it. X is coming down from capital N minus 1 to capital N minus 2 plus dot up to small n plus 1. This is going up further right. This first part is the linear convolution component you have seen, but second part is the error. That is why together, it is not linear convolution together, it is called periodic convolution or circular convolution between x n and h n.

Now, I will show you a graphical way I think that is where I stopped last time a graphical way of evaluating it. I will just dictate the steps out of the 2 sequences x n and h n both are of length same length. You hold one of them as it is and make a periodic version of the other one. So, that means, whatever do you have when one period of length capital N go on repeating it in left and right, to the left side and right side it becomes infinitely long sequence.

After that the next 2 steps are common in linear convolution. You reverse it reverse the entire infinitely long sequence mind you after reversal also it remains periodic. A periodic sequence when you reverse it time reverse it it still remains periodic. Then you shift these by that amount at which you want to carry out the convolution. Like here you are finding out the convolution at n. N can be 0 1 up to capital N minus 1 that is our range. So, shift it to the right by small n

After that again you still have an infinite length sequence. Now, you throw away the components outside the window from 0 to capital N minus 1 to the right and to of capital N minus and to left of 0 whatever you throw it away. So, they can becomes a finite length sequence. So, 2 finite length sequence will have h n as it is and 1 obtained from x n in these manner

Now, go on multiplying samples from top and bottom, add them you will get the circular convolution. That is my claim and will have it verified that will intend give you this expression. So, let us do that and mind you I will not go through all the steps I rely only on intelligence. So, that you can visualize how I am getting a sequence like this and all that. Otherwise it will becomes too lengthy and clumsy also you have to draw too many diagrams.

So, h I write as it is say may be h r where n I reserve for computing the convolution. I mean n is the index at which you have to find out the convolution. So, n is your choice which means fixed n cannot be variable. So, call it h r then. Plot h sequence as a function of 1 not as a function of n because n is your choice here, you want to find the convolution at specific choice of value of n. So, n is fixed That is why I am writing at n r.

Then what will I do, whatever I have for x n x r rather x r I will make it periodic fine. Then I will shift reverse it and then shift it by an amount and then chop of some portion to right of this to the left of these and sample wise multiplication addition. So, for the time being I will start with 0 shift then 1 shift 2 shift. So, 0 shift only.

If it is 0 shift what I get here is x 0, then to the right I had x 1 x 2 x 3 like that, then it reversal it will go this way x 1 dot what will be here, at N minus 1 again this x 1 will come is not it, x 0 this is x 0 at zero th point again at n th point there will be x 0. So, here x 1 up to x capital N minus 1 like x 0 x 1 x capital N minus 1.

So, this is x I am sorry it is becoming clumsy that is the problem with this N minus 1 can you see this N minus 1. I am sorry it cannot be bigger here 0 shift. Now, I throw away to the any component to the right of this to the left of this to the right of capital N minus 1 to the left of this is just multiply. Obviously, you have one term x naught into sorry x naught into h 0 x naught into h 0. This term plus you have you, see x capital N minus 1 h 1 x capital N minus 1 h 1. So, on and so forth h 2 x capital N minus 2, are you with me. Small n is 0 here. So, 0 plus 1 0 plus 1. You are getting it.

Now, if I shift it by 1 and then again chop of quantity to the right of this point and to the left of this point. This was for n equal to 0 for n equal to 1 we do once or twice. It will be very clear. For n equal to 1 this x 0 will come below h 1. See that that part is like linear convolution x 0 comes below h 1 x 1 comes below h 0 and sample wise multiplication.

You I have shift it. So, you understand h 0 has come here the x 0 into h 1 x 1 into h 0. That part is like linear convolution. So, far so good this is ok. Then n minus 1 x n minus 1 will go below h 2. This x 1 will move out, are you following me x 1 will move out. So, you have got again this form. In general now you can see, in general if you shift it to the right by n bit x 0 will come below h n.

Therefore, below h 0 will be x n distance is same. So, that part will be obviously, x n h 0 x n will be here h 1. That means, before x h n x n, there is x are you getting me, this very simple h 1. This x 0 has gone over to the n th point this is n this is where x 0 is now. So, x 0 into h n I have shifted. Then to the right of this is x capital N minus 1, x 0 is here to the right of this is this fellow to the right of x 0 is always x N minus 1.

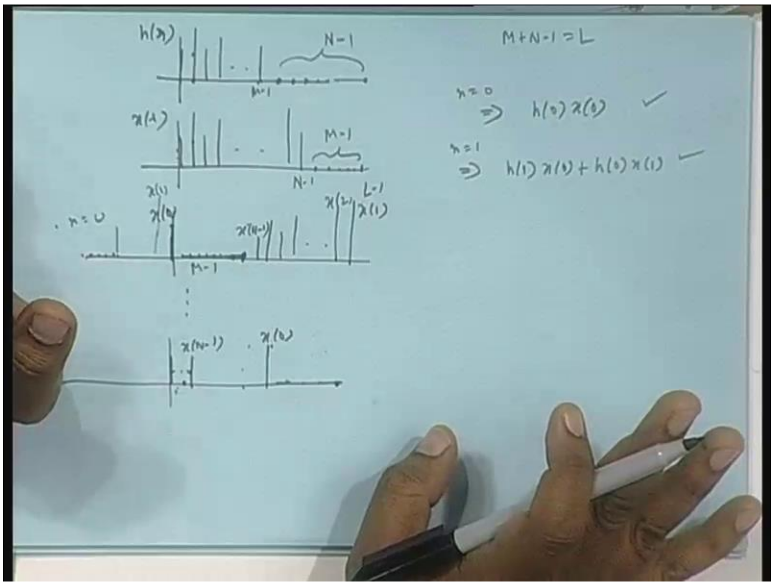
So, x this fellow this is x n minus 2 x n minus 1 will multiply what h n plus 1 th. If it is n this is n plus 1 th. So, h and in this order in this order h n plus 1 h n plus 2 up to this. This will go up this will go down, this will go up this will go down summation remains. So, you get these. If you have question you can ask me. I did not show so many diagrams and all I thought by means you clear.

So, this is the graphical way of doing convolution, but you understand there is a problem. This error things comes though it is not part of this adaptive filter, but just for educating you people, let me tell you how to get rid of this quantity. So, this is not something we need here. So, that is why I draw it on the back side. That is by using this DFT technique only I want to find out the linear convolution. Then what we do suppose x n is of length of capital N, this is of length capital M and N is say greater than M. What I will do I will be adding some 0's to both because what is the length of y n. If it is length capital N if it is capital M y n length is N plus M minus 1. So, I want to make this and y n same length.

That means this was length capital N. So, I add M minus 1 0 here, capital M minus 1. This was length capital M. So, I add capital N minus 1 0 here. Now, if I convolve them I still get some sequence because 0 does not change any result. So, all there of the same range, but then the question is does it, if I carry out then now I mean the I DFT, DFT means how many point DFT N plus M minus 1 current length. So, you call it capital L.

So, L point DFT, L point DFT L point DFT multiply L point inverse DFT. If I do I get a L point sequence, is it the linear convolution. Is it that if I carry out the graphical method I will have this this this appearing then, that is the thing. You want to verify if this disappears wonderful then that will intend give us this these part only linear convolution. So, this this we need to verify. So, what we did.

(Refer Slide Time: 14:02)



Your h n h r rather, it was up to M minus 1 and then I have 0 0 0 0 0 0. How many 0 length M and N minus 1 0 because M plus N minus 1 is the length of linear convolution between x n and h n. That is output y n has a length of this. You can call it capital L. So, I want to make in both the input signals to be both h n and x n to be of the same length. So, I appended extra 0 already it was M, now N minus 1. So, M plus N minus 1 it is length is L. Similarly, I do the same thing for x, x plus N minus 1 how many 0 M minus 1 0. I have to now do circular convolution between the 2, very simple I will follow the graphical method.

I will hold this top 1 as it is make a periodic version of this reverse shift. Then chop of the right side and left side right of this point and left of this point. This point you can call as L minus 1 0 to L minus 1 length is L. There is sample wise multiplication addition. See if you do that for 0 shift say for N equal to 0, what will happen. This x 0 will remain here please see this figure is very important figure.

This x 0 will remain here this entire thing will be folded to that side. So, that means, this samples will go and to the in the front will be these 0 pad. If you fold this that way, this part will be folded back right x 0 x 1. So, here will be x 1 here will be x 1 up to some point then 0 0 0 0 0 same thing will repeat here periodicity.

So, this 0 pad M minus 1 will be here M minus 1 is a 0 pad will coincide with this point length is N. It will coincide with that point and then there will be some data. This will be x 1. To the left of it x 0 I have x 1. So, again from periodicity this is coming from this side x 2. This is x N minus 1 and now chop of right and left multiply.

This 0 will take up this fellows and this 0 will take up this fellows. So, only you get x 0 h 0. So, n equal to 0 this is called 0 padding technique, 0 padding. That means n equal to 0 means only h 0 which is a linear convolution correct linear convolution result. You understand these 0, how many M, M minus 1.How many samples here 1 is already here. So, N minus 1. How many 0's N minus 1. Now, you shift it to the right by one.

The 0 will shift here, one data will be out, rest of the data will be again multiplied by 0. Only these 2 samples will be taken care of by this which 1 x 0 will be from below h 1 now x 1 will come below h 0 linear convolution. All correct result and I go on doing it. After as I go on shifting some data again some data will go out these 0 keeps moving. This 0 keeps moving, 0 comes to this side and data comes.

As long as these data comes here no problem. Then will multiply like you know x 0 will move further to the right below h 2 x 1 below h 1 x 2 below h 0, you get convolution result at 2. Subsequently, there is only 0 part of the samples will multiplied by 0's part of this samples will be this samples will be multiply part of these 0. So, on and so forth.

After a while what will happen these data dot after a while. As I go on shifting you will have a situation where these data has come here to the right of this. I mean this is, the suppose data block. So, that means, here also and I have got some 0, here this entire data in 1 block means remaining points have to be 0. Some 0 and remaining 0 here.

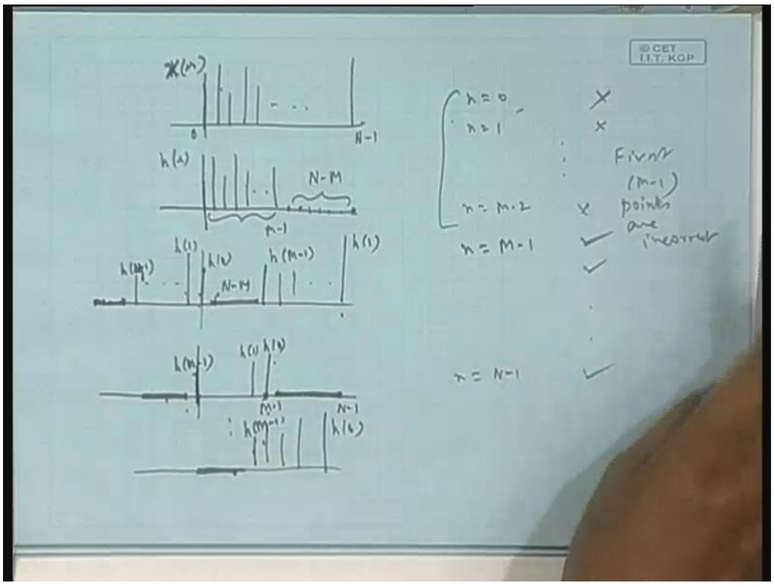
This 0 will make sure that no that nothing comes from top and these 0 will take care these 0 are colliding with these 0 you have no problem. Essentially you have x 0 coming with the appropriate I mean you will get linear convolution. Normalize you do in the linear convolution.

Finally, this will come over here to the right at this point x 0 finally, come here and that time x n minus 1 will be last sample, last here to the left will be 0's and that will last result of the convolution. So, you all understand it will give you perfect linear convolution result. This is called 0 padding technique in your DFT you should have studied this. I am sure you have all been studied, taught, circular convolution by.

Student: ((Refer Time: 20:17))

Anyway, another result now I will come back to my context. Another result we will just present the result and store that result somewhere for our future conjunction. Suppose, I do not do 0 padding in both which way is the shorter one I add those extra 0's only to that guy. So, that both the lengths are same. Then do circular convolution. Obviously, I will not get correct linear convolution result, but is still something better than not doing any 0 padding at all. So, what will I get that is the question.

(Refer Slide Time: 20:54)



Suppose, I have got a situation like this. Say, h r say x r input I always take to be bigger and this h fellow up to these length M, length N length M. What ideally I should do, I will add M minus 1 0's here and N minus 1 0's here, then make a bigger sequence convolve. I mean if I if I do this circular convolution I will get the correct linear convolution result fine, but suppose I do not do that

I say the bigger guy I will not touch, let it be as it is I will add some 0's only here. How many 0's length N length M. So, obviously, N minus M 0's. I say that I will do a circular convolution between that 2 means, I will take n point DFT n point n point DFT multiply n point I DFT. What will I get that I will that is what I get graphically by that circular convolution I want to examine what it is.

Whatever I get there is n point sequence is it that part of that at least is correct or no part no point I will get correct linear convolution. Everywhere I have got the error component. Will see that, part of that output gives you correct data correct linear convolution result we will see that and some portion you will get wrong result. We will identify where correct where wrong.

So, suppose I do this circular business. So, I will be making periodic version of it reverse it and then 0 shift, first shift and always chop of portions to the right of n minus 1 and to the left of 0 sample wise multiplication top and bottom and add. That is also simple formula. So, if I do that h 0 and then 0's this is a 0 pad here. That means, here is a 0 pad 0 pad of how long N minus M.

This sample N minus M and then I have got other samples which 1. If it is h N minus 1 h what h M minus 1, sorry this N M minus 1. So, here also I have got h 1 this is my area of interest. Now, you can easily see I will not get correct result why because correct result means for the 0 shift I should have only h 0 into x 0. This samples have taken care of some samples coming from here this 0's, but these guys are not taken care of by 0 here because I need not place those 0's.

So, I am getting incorrect result. So, at n equal to 0 incorrect n equal to 1. I am not bothered about the correct expression. I just want to find out where correct where not correct. Correct means against linear convolution. So, at n equal to 0 I will get erroneous result as far as linear convolution is concerned at n equal to 1.This 2 will go up this 0 1 sample will be moved out, but still this fellows are there. They will multiply with the samples from top and then will give raise to error.

This left of 0 pad is fine h 0 below x 1 h 1, below x 0 that is a linear convolution thing, but this fellows are getting trouble. So, this will be incorrect it will continue as long as I mean till the point where this 0 pad reaches here. From that moment at that moment I have what h 0 h 1 dot. How many, how many shift.

Student: M shifts.

M shifts because h 0 I had this much h 1 M shift of course obviously, total M minus N 0's. Here this point is my M N minus 1 th point. So, this way I can find out. What is this point M shift. After M minus M shifts, I will have a situation like this. What will be this point h M minus 1 no h M minus 1. This folded portion, this will come out, 0 shift you are here 1 shift you are here 2 shift after m shift this will be here if you at 0th point M minus 1, this is M minus 1th point length is M all the M data have to fit in.

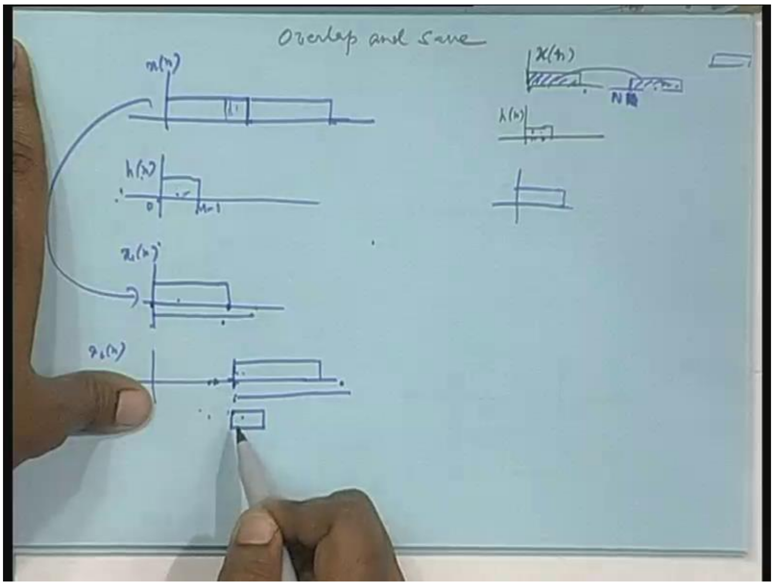
So, that for fit in and that is why only then you have got that extra 0's. This entire 0 covering off the remaining place. This entire thing number of data how many M that of come here and that is why M minus N this 0 block perfectly covers the remaining part of these. That is the time when no such data will create trouble because they are multiplying 0's. Earlier to the right of the 0 pad I had a sample. Like to the right of the 0 had a samples they are multiplied by samples from here they are creating trouble, but now this is finally, pushed to this length N minus M total is N

So, how many samples here M, M means the entire block has come here. So, h 0 h M minus 1. So, at 0th location M minus 1 is sampled such M minus 1 location 0th sample. So, this will be what, this will give you the output from n equal to M minus 1 onwards correct output. After this you shift it to the right, but from here again the 0 pad. Finally dot or time will come, when again h 0 will come here h M minus 1 will come here.

This 0 pad will fully fill up these place. Length wise you see, total length has to be N, this is M this is N minus M and again this will take care of this fellows. So, from this point onwards till n, till the last point. Last point is n equal to n minus 1, I have got correct result. So, how many for n equal to M minus 2. That means first is incorrect. First M minus 1 points are incorrect.

So, again let us sum up. There were 2 sequences, 1 shorter 1 bigger. Shorter 1 I added extra 0's to make its length equal to the bigger ones and then I followed a circular convolution. In that circular convolution I still get a length n sequence, out of which first M minus 1 points will give you wrong data. First M minus 1 points after that that is from zeroth to M minus 2th. From m minus 1th you will start getting correct linear convolution result, till n equal to capital N minus 1. Remember this this result you will store somewhere. I will making use of it later. Now, let me something else from very basics of DSP.

(Refer Slide Time: 29:13)



There is a technique called overlap and save, overlap and save method. Suppose, you have got a very long, forgot about all these even DSP forgot about for the time being means. Suppose, you have got a very long input sequence where there is a FIR filter you want to find out the output. At each point you have to go on doing convolution that. So, many points refer to convolution. So, much of computation you have to do and do find out the output.

So, whole task can be make much simpler and faster computationally by some technique. One of them is called overlap and save method. First before I go into that, certain basic things of DSP you see whether you can force it. Suppose, there are 2 sequences. One is here another is here you call it say x r or x n does not matter you call this h n. You get a convolution. See I have drawn it like this causal.

It is not necessary that they have to start at origin they can go beyond to the left also. My result has not depend on this save. Just a sequence, 2 sequence if you convolve you get something. We know where would locate all that all those things we will know. My point is if x n is suppose instead of these if this is this x n is shifted here. That is this goes to a point say capital K not K, K have used for DFT say again N this entire thing N. This guy is come here and I am convolving the 2. My convolution will be result will be same because data has not changed, but the location at which the output will be placed will be what?

Student: Shift by the same amount .

Shift by the same amount. Earlier if the output where shift say position here, it will never shifted by same amount if you do not cannot see this graphically. You can just suppose go into the z domain or z domain say convolution in time domain means multiplying using z domain, z transform of this will be z transform of this is the z transform of the output here. Fine what I am doing here I delay it by capital N.

So, same z transform multiplied by z to the power minus N into z transform of here. Now, z to the power minus N you keep aside then again then z transform of this would be the z transform of this, which is the same output, that multiplied by z to the power minus n means this output will shifted by capital N

In z domain it very easy to work out. So, I can conversely, if I am convolving 2 guys which are separated like these, then I can make life simpler by simply shifting it aligning this with origin. Convolve calculate the output and again shift it back to by simplified by capital N. So, again next time I can convolve with some other person here. I will again shift it back to origin, convolve get the result and again place the output. There this way I can do. Another important thing, suppose and this is what you know books do not mention these aspects and all list out this point.

Suppose, this has data this data say I am segmenting like this and there is 1 sequence I am convolving with fine. You know how to do linear convolution will be reversing it and then multiply sample wise shift it again multiply sample wise shift it go on doing it. You will get an output from here to some point to the right of these. You all know convolution suppose this data I break like 2 components. One component x 1 n this was x n another is and this is h n. This length is 0 to M minus 1. This is x 2 n I am saying that first I will convolve between these 2 find out the output.

Take the output from here to here. Output will be though convolution, output will be what output will be going to the right. Again I will convolve between h n and this I start here and I go to the right. Then here if I take data from here to here only output from here to here only. Next output I take from here you understand. If I convolve between h and x result will be something like this up to this point. If I convolve between h and x 2 result will be like this.

If I say that I will take the output from here to here. Then I will go here and take the output up to N will it be correct. Answer is wrong because when I convolve between h and this I am reversing it, but to the back of these to the left of these there is no data. Where as in effect when I was doing convolution there was data coming from this side that was getting multiplied by part of this that we will be using.

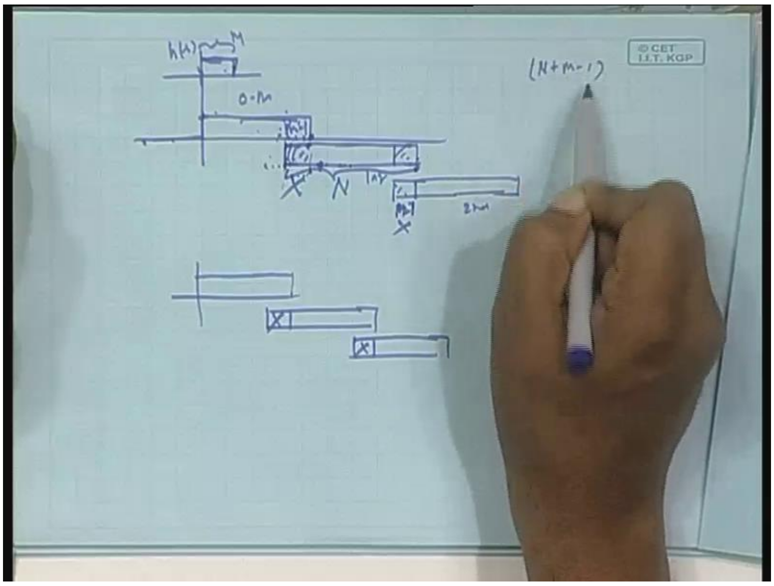
That means here if I do this convolution at this point will be wrong. Still again at this if I reverse, if I suppose reverse it if I reverse it 0 to minus of M minus 1 that entire thing comes here. They multiplied by 0 here and only 1 sample 1 sample. So, you get some result, but that is wrong because in effect there was no 0 this was data, but you have made it 0 here.

Fine you see you understood there is wrong there is something that goes wrong. You go for the next one. Still partly it will be 0 here and only 2 value that are coming. So, still that output will be wrong. When will the output be correct when after getting shifted this entire thing has come below this. Then you do not need the any data from the previous block. When will that be, after how many shifts.

Student: M

From M th shift onward. So, that means, if the origin where here from the 0th sample first sample, second sample third sample, if I align everything to origin say. Then here o th , first, second, third, fourth up to M minus 1th will be wrong M th will be correct. Remember there was also a different context figure was same M minus 1 up to M minus 1 was wrong M th was correct. Two different context, but just figures are, look at the figures. Now, I propose a technique to propose this long convolution.

(Refer Slide Time: 36:39)



That suppose, I break this data like this, I will have an overlap. This is my 0th block next block. In the case of block adaptive filter this will be that block right away. Next block instead of starting here I come back. How many points inside M minus 1 points inside that go like this. That is here I have overlap data is same I am just showing as a 2 kind of strips. This is one strip of data and then last M minus 1 component again repeated in the front of this next block.

This is block number one again this guys repeated M minus 1 length. Second block dot. Now, if I do that and suppose I am convolving with a small filter, h n whose length is capital M. Here again to make life simple, suppose you really started here to the left is 0. So, to the left of the problem. You can reverse it this is your h n this is your h n. So, you can reverse it. Indeed here I have 0. So, there is no problem, now you go on doing convolution to get result.

Here also at this point, at this point, at this point you go on doing the convolution. You get correct result because all the previous data needed for this convolution is already available in the block fine. So, you get 1 set of output place it here output you placed it.

Now, you are convolving, now you take this block with this block. You want to convolve. Firstly you can first shift the block alignment align with the 0 thing. Then convolve whatever result comes that you place back here just for making life simple. So, suppose I am now this is the 0. I mean this is vertical axis this is the origin you are convolving between the two. That time again at this point at this point at this point you get result, but will that be correct result answer is no because to the left of this I have no zero, but originally there was data.

So, that those output I ignore. Who will supply me the output, from the previous block computation I already computed them I stored them saved them. That is why overlap and save them. So, at this points I get incorrect result, but I throw I ignore. Only from this point onwards I get correct result I stored them till here. So, this will be my adaptive filter block adaptive filter this was zeroth block 1 filter weight.

Slightly different from this context because I am keeping the filter to be same here convolving always with these guy, whether these guy, this block or this block or this block. In the case of adaptive filter when I convolve with filter I have got 1 filter weight when I go to the next block weights value have changed, but when values So, their blocks will be overlapping like this.

At this points I need output, but that output will correspond to the filter weight for these block and that I already obtained. Are you following me. Filter weight for this block I already adopted for that I mean I will take the output at these points corresponding to the filter weight. The weight assigned for or for zeroth block. That I have already computed in this convolution I saved them.

Now, weights have changed I start again at this points. Firstly this points gives me wrong result even if any weights did not change. Now, weights also have changed. So, at this points the output really I do not want are you getting me, but I will not need them because they are already computed correct desired things already computed. I will throw them away, but from this point onwards I need the output for this block only with the filter weight obtained for this block.

That I am doing because for that I do not need any data from to the left of these. So, that is available and I happily do it. That means, whatever is here that I will discard whatever is here at the output I will discard. That is output if you place like this position you discard again output if you place like these this portion you discard. Now, I want to do it by FFT. What will I do one thing is you can tell me, sir you want to do, after all you are doing filtering between these, I mean convolution between this block and this entire block.

This is of length say N, this entire length is N, this entire length is M. So, you can tell me sir we all know 0 padding technique, as long as you are convolving between these and these you will first you will finally, throw away the first 2 samples I agree, but before that let us carry out the convolution between the two. Only the first N minus 2 samples are wrong, we know because they are assume assuming 0 values to the left. Whereas actually there was data.

They are assuming same filter weight or else filter weight has changed. I am putting back the correct, but even then first you calculate the linear convolution. Then only I will throw away this part. So, you can tell me look fine. You want to do a linear convolution between this block of length capital N and a block of length M. You add 0's to this guy also in our calculation M minus 1 0's. You add 0's to this guys also N minus 1 0's.

So, M plus N minus 1 is the length fine that will work DFT, I DFT you will get a full output and there you throw away this sample. I agree, but instead of N plus M minus 1. This many point of DFT do you have more computation. Then n th point DFT much less computation N plus M minus 1. I say that I can use simply n th point DFT because what I will do, I suppose this block you are taking this block you are convolving with this guy.

What I will do this is the shorter sequence this is the bigger sequence, longer sequence. I will add only 0's to this fellow. N minus 1 0's and do circular convolution between the 2 block. I just shown you previously that even there you get incorrect result only in the first M minus 1 points only. From M th point you get the real linear convolution result between this block and this block.

That is what I want because from this point onwards convolution between this and this is enough I do not need any data from here. So, two things are matching that also M minus 1 points you have to ignore and I am already ignoring. After wards you get the real convolution linear convolution output between this and this. I am interested in that linear convolution because at this point, at this point, at this point when I convolve if I knowing the data is contiguous and all there is no contribution from this tail. This much is enough. Are you following me.

When you suppose this is the origin you first reverse it all 0's are coming because on this side 0's incorrect result, but as you moved it here as you moved it here. From this point onwards you need purely linear the linear convolution between this and this is enough. I do not need any data from here that you can say that look originally there was some data. Now, you have got some 0's only. So, you are getting incorrect result that does not apply here.

So, simply linear convolution between the two block can be acceptable in the final result. So, again that figure M minus 1. So, M minus 1 I do not care either way. This is called overlap and save technique you see I would have just leave happily with n point DFT only not so much N plus M minus 1. You see these are the DSP tricks and you know I mean wish I could read some papers. Some papers have come out recently where it can be still you know manipulated to be less than this.

Some papers are there did not study under the assumption that data is real and all that can further I mean some further optimization are possible. Let me see if I can time to read some recent papers they have come out. This is overlap and save. So, you understand it will how, it will be used in the adaptive filter case not weight updating I am talking up now. I am only talking about the filtering part.

For this block 1, set of filter weights for this block another set of filter weights, but for this block say this is the filter weight. I have to convolve between this and this block. That time at this point, from this point onwards to get the output I do not need data, any data from these block because this previous M minus 1 as available here is enough for me.

Then what will happen for computing output here. Firstly, we have to compute the output at this this points. I will target this block, this points comes under this block. So, therefore, filter weight for this block, for that I have already calculated in the previous round that have saved. Are you getting my point. So, this is how you will can simplify this computation in the forward part filtering part, but the more challenging thing is the weight update part.

Weight update part I will have to continue till tomorrow. I can just briefly show certain things.

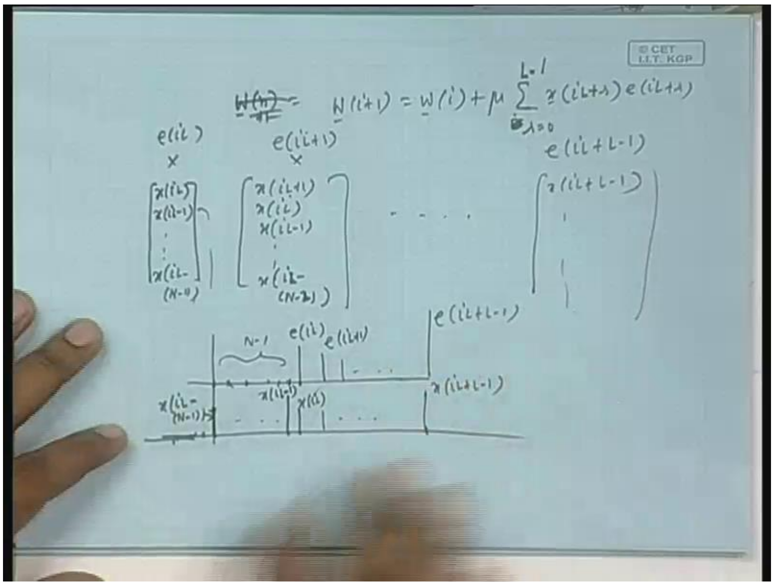
Student: ((Refer Slide: 47:22))

Yeah right, zero th means starting block.

Student: Let us say yes.

Yeah in starting block there is no question of throwing out something because there is no more to replace that weight. In the block LMS algorithm there are there are 2 things. One is filter weights are fixed for the block and you are doing filtering means. There is a block of data it passes through the FIR filter that is what we did now. We have to do linear convolution. You can use FFT by choosing the blocks in that overlapping manner. There are one block another block there is an overlap of M minus 1 samples. If the filter length is M, M minus 1 samples. That way, that much is clear. Now, weight update part. In the weight update part you can see one thing that let me write down elaborately.

(Refer Slide Time: 48:28)



W n plus 1 not n W I plus 1 because is block LMS, it is not question of updating with time n. Updating with block index I plus was it mu by L what notation you use mu by L mu prime by L or something.

Student: mu by L.

 Call it mu because that is off no use here just any constant will be it is the computation I am bothered about. So, it was I not I which notation you use here j is it because I is used for the block index r, r equal to 0 to capital L minus 1, capital L was the length of the data block and capital N was the length of the filter. L was suppose you have taken to be long longer than N though I do not see if necessary.

We have got this things x vector, by the way if it were a complex LMS algorithm complex, LMS algorithm then you can see one thing, whole derivation would be same, but there will be a star here x star, star on x or star on e

Student: Star on e.

Star on e, we will be considering that extension because when I do this FFT finally, I will take care of that also. For the time we are assuming real. Just look at the data structure here what kind of thing is happening. These are all data vectors e. So, this is one vector I write this is the starting point of the block, but before this how many points.

So, this is that overlap zone. I will take the last N minus not M any more filter length is not M here its N capital N. So, last N minus 1 data from the previous block they correspond to that. In my modified scheme they have got appended with correct data block. Correct data block earlier was starting here, but I have appended this part. Only thing output at these indices will be ignored, because of the thing which have been discussed. So, long this is my x this data vector of r equal to 0. This will be multiplied by this is multiplied by e I L. Next block is, you understand I L will go on I L plus 1 dot.

See one thing, consider the first row because after all multiplying will be adding them. You have multiplied this entire column. This entire column by this and so and so add you will get a column vector. So, consider the first row or first element of the column, how will you get that. Just the first entries. This fellow into this fellow and this fellow into this fellow dot dot dot this fellow into this fellow.

Next entry will be what, this sequence remains same, but this sequence gets shifted to the right by 1 you see. I L have moved here I L plus 1 here and I L minus 1 here and multiplication it is like this. That suppose I have some 0's how many N minus 1. N minus 1 0's and then I have got e I L I put e I L plus 1 dot dot dot. All these terms e I L plus L minus 1, suppose I do that. This entire data starting from here I go up I put this at the origin, then go up this manner put them here. I L minus N minus 1 then dot dot dot dot up I L then I L plus 1 I L plus 2 and further put them here. No books will explain like these please be with me.

Student: ((Refer Slide: 53:27))

I have put some 0's.

Student: I L minus N minus 1.

N is the length of the filter not M any more in adaptive filter it is N not M. I will what I am doing now I am taking this data from here, 0th data first data second data t like these. So, N minus 1 and then L. So, this data will be here dot dot this is and then your normal this fellows x I L this guy.

Suppose, I multiply the 2 sample wise and add first what will you get, what was getting here this into this because after all e I L only multiply x I L other fellows are 0. This will multiply the next guy. So, this fellows will not come into operation. I will get the first entry. This in to this plus, this in to this plus dot, this in to this first N for second entry what we will have to do.

If I shift the entire thing to the right by 1 bit and send in the 0's. Suppose, as though there was 0's here. Send in 0's and then within this zone, I will multiply top to bottom and add will I not get. So, it is like that intermediate step of graphical way of circular convolution even you have reversed it and then shifting and chopping off. So, this is the reversed version, what is the original version. You understand, I have to use a circular convolution are you following this thing.

Any way time is up. So, I will just stop with this question. This is what we do in that intermediate stage of circular convolution first no shift multiply then shift I want bit, but chop and confine within these only. From here if I can send in a 0, then 0 comes in 1 data goes out. This data goes out, this fellow comes here this goes out. So, this fellow comes here and again you multiply add and go on doing. That is what you do in the circular convolution the later stage in the computational stage. So, that means, this is obtained by reversing something. So, what is that something that have to look there. So, I stop here today. So, tomorrow I will begin here.

Thank you very much.