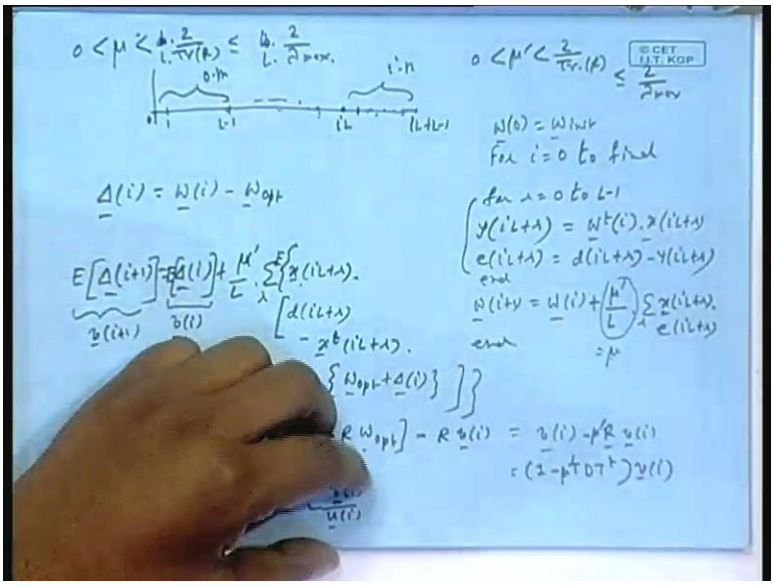
Adaptive Signal Processing

Prof. M. Chakraborty
Dept of Electronics & Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 14
Block LMS Algorithm

(Refer Slide Time: 00:57)



LMS algorithm yesterday we are almost finished it, then just to recap just like this. I mean input was block like this, you know 0th block was 0 1 up to L minus 1, this was say 0th block dot, dot, dot ith block was from i L, L is the block length dot, dot up to I L plus L minus 1 ith. Over each block we are keeping the filter weight constant, unlike the LMS algorithm where filter weights were changing at every index we are doing iteration from block to block and within the block, we use a same weight. So, basically the algorithm was like this I mean for say I equal to, I is a block index, I equal to 0 to 0th block to say final block, final could be anything of your choice, final block within the block, what are the block indices for the ith block?

i L, i L plus 1 i L plus 2 dot, dot, dot i L plus L minus 1 So, I can introduce the variable r, r will take value 0 to L minus 1. For r equal to 0 to L minus 1 y I L plus r, is it not? I L plus r means I, i either here or here or here or here or here at this points, filter what is the output? Output is w transpose I, it is not I L plus r, because that is constant for the block that is the difference into the data vector I L plus r. And remember when r equal to 0, you have data from the not only here, but last L minus 1 sorry, last n minus 1, capital N is the block I mean filter length, N minus data from the previous block. So, you have to have those data stored also.

Then here also, here also progressively less, but we need to have maximum of n minus 1 data from this the last n minus 1 data of the previous block stored. So, this was our output then, I L plus r was your desired response this is fine, this is 1 end, filtering an e r computation you are doing at every index within the block. Once that is done now you update, go to the next block that starts at w i plus yesterday we have seen, it is some mu prime by L mu prime by L into this entire quantity you can rename as equal mu, some mu into there is a summation. So, you need more computation actually there is a summation, but summation was over what?

Simply the data vector at each index I L plus R into corresponding in the case of LMS you had only 1 data vector followed, multiplied by the error. Now, it is you have do the same thing at all the points and averaging that is why L factor comes up and here r, r equal 0 to L minus 1. This is 1 loop and this is at loop and initial condition you can say w the 0th iterate equal to something, w in it, which is usually 0 vector this algorithm. You see the algorithm the problem is weight update computation is now more complex, more computation is it not? but all this can be made faster by using a FFT which will be discussed.

But before that, let us do a little bit of convergence analysis just for convergence it mean only, not for that mean square error analysis which is very lengthy you know, for this algorithm and that will be again very much like what we did in the LMS case. That is suppose, we define the error vector, weight error vector for the Ith iteration. Earlier it was delta n, now it is delta I this is the only difference, because weights are changing over I, block to block not over n. So, it was w i minus w opt, w opt remains as usual that is an optimal filter r inverse p.

So from the, in this equation from both side if you subtract in this weight update equation, from both side you subtract delta I, you get sorry you, you subtract w opt you get delta I plus 1 as delta I plus say mu prime by L into this thing. But I am told you that this error vector itself was contains information about w. So, that has to be taken out is it not?, so this summation remains as it is over r, when I say summation over r, I am not given the index r equal to 0 to L minus 1 that is understood. We have given the range of r somewhere.

So, this first component I L plus r remains as it is and then this guy, this guy is minus w transpose x or x transpose w was same and is a real case. x transpose I L plus r into w I L w i, but w i is what w opt plus delta I, is it not? This together is delta I plus w opt is w I, this plus this that is why I am doing here. Now apply E, apply E over this, so again E over this and this quantity also E over this, sorry I am putting you know I mean for you it could be tedious, but I am using the same equation I need inside easier. Just to save time, but you are familiar with these things. This expected weight adder vector, I give it a name last time I think v or delta what, what name I gave, v.

So, again I gave v i v i plus 1 this v i this quantity is v i plus 1. So, this is v i plus mu prime by L summation over r, so expected value of this data vector times d is absolute similar to before. Data vector or any index times the desired response, expected value of that is p vector and p irrespective of the index. So, the first term will be p and then x, x transpose and w opt, w opt is a constant. So, if you take that out x x transpose expected value of that is r irrespective of the index So, r as before you know r w opt and you understand this term will cancel, because what is w opt r inverse p. So, this will cancel this will become equal to 0.

So, this is 1 term this becomes equal to 0, but there is 1 more term minus E data vector times is transpose time delta and here I will be using that independence assumption, that w i is independent of the data vectors, you see it is more we have done earlier. Because, earlier because earlier w was independent of the current data vector, but now there are so many data vectors, is it not? what the entire index, is it not? W i actually was obtained from the previous data vectors, the data vectors for the previous block, but there is a correlation in the data and what will be the data vector here consist of data from the previous block, there is overlap.

So, w i actually you cannot say strictly the w i is purely independent of the data vectors present here for all r. Because, firstly w i was obtained by the update phenomena by using the data vectors from the previous block, but there is a correlation in the input. So, that way through that correlation w i depends on the data vectors or data for the current block itself. Furthermore when r equal to 0, r equal to 1, r equal to 2, I mean this part of the data vector overlaps with the previous block and that was used in the evolution of w I, is it not? Still we make this assumption and it works that w i is independent of not just uncorrected, statistically independent of the data vectors corresponding to the current block.

Why statistically depend that means, I told you uncorrelated does not help us much that is the uncorrelated means what? If there are 2 variables x and y E x y equal to E x into E y. But if I then come across situation that E of x square y or E of some f x into g y, I cannot separate out. But if they are statistically independent, if it is given that probability density of x y is equal to p x into p y, I can then have E of any f x into g y is E of f x into E of g y. That kind of thing will come here because obviously, you see x and x transpose there are those square terms coming out, is it not? And then delta I, that is why statically independence is used and therefore, what will that give?

If I to separate out delta because of the statistical independent assumption delta from this x spot. Then x x transpose will gives rise to r and E of delta I, E of delta is v i. Now, this is summed L times, you see this is summed L times and L and L will cancel. So, you get back what we had got earlier, after this we do not need to proceed minus r v i which is nothing but I minus R. You remember r was TDT transpose v I, what you did after that we defined in the LMS case, what you did. We defined T transpose v I, T is unitary matrix T transpose v i you have forgotten, what you did in pre-multiplied this by T transpose. T transpose v i plus 1, we give it a name u i plus 1 and norm square of u and norm square of v they are same, because T transpose is unitary and I was written as Ti T transpose.

Student: ((Refer time: 12:38))

Oh, there is mu here thank you. Where is that mu thing, mu prime, yes thank you very much. Mu prime is a scalar. So, T can be move backward and forth.

Student: ((Refer time: 12:57))

I will be written as TT transposed, Ti T transpose. So, you can take the 2 T common absolutely same as the same equation, mu prime D into T transpose v I, T transpose v i is there is a vi which is your u I, just a minute. And then if you take the norm of this vector, that is same as what, T times a diagonal matrix into u I. You remember that I told you that, unitary matrix types of vector does not change the norm, that is why norm of u or norm of v are same. So, if I can show that norm of u tending to 0 that means, norm of v is tending to 0 and norm of a vector 0 tending to 0 means each component is tending to 0, but what is norm of this or norm square of this norm square of this part.

Because T is unitary and then, norm square of that part you evaluate this diagonal matrix, that will be the first component squared times 1 minus mu prime lambda 0 whole square. There is second component square plus 1 minus mu prime lambda 1 square and dot, dot, dot and you may say that if each of this 1 minus mu prime lambda say, lambda k. If it is kept within 1 to minus 1 then this total norm will be less than the norm square of u I, this is repetition, from that we derive the condition for mu prime and that will be same as the equations are same. So, for mu prime again we will get the same condition. So, in terms of mu what will it be, so mu prime should be say 2 by lamb, 2 by if you can put this way trace R which is less than equal to 2 by lambda max.

So, in terms of mu it will be in terms of mu what, what will be mu, because when you run the algorithm you do not write mu prime L, you give it a name mu. So, that is the step size mu will be just L times that is mu. So, this algorithm also convergence in mean convergence, it means is established now people have done it you know lot of research has done into LMS. Then they try to do that same analysis that mean square analysis which you did at length for 3, 4 classes you remember? You found out the mean square error in the steady state that have been even extra component call x s mean square.

And we wrote that was evaluated in terms of weight error covariance matrix k and other k prime n, we found over the dynamics of k prime under some assumption and all that from that evaluated that x s mean square and adjustment are just made. Same thing can be done here people have done, but no need to repeat only steps are more or less common. So, we will not go on doing it for all elements, because this just gives you test. I will do now something which is more DSP, that is its implementation.

Student: ((Refer time: 16:33))

1 by L or?

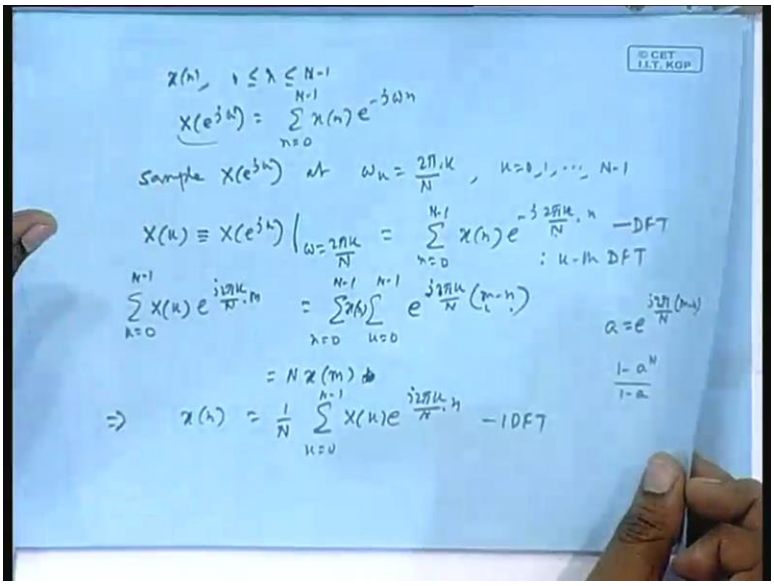
Student: Sir, mu by L is

Mu prime by it could be 1 by L or there let me see, whether 1 by L, say I wrote quickly. So this this is very simple thing, mu prime this yes, mu prime by L you are right it will be by L, thank you. You divided by L mu prime by L is mu, so divided by L. So, 2 by L trace r 2 by L, thank you. This is the, these corrections you make anyway, since it is video recording to tell me also for its no officially the correct thing is recorded. If it is a class room I would have been ignore such comments.

So, now I am interested in the implementation of this, that is why this algorithm so important you know, that is very. I mean 1 full block output and weight update for that block can be carried out, for all the integers filter output error output and weight update, all these things for the entire block can be carried out much faster than in an LMS algorithm using FFT and all. So, now we have to get into implementation and again I do not know how much FFT or DFT you know. So, again you know as is my practice I start from the basic. You all know discrete time Fourier transformation course DTFT all of you know, I mean please do not ask me to start from DTFT.

Given, a finite length sequence x n and some other DSP guys into its common, I have just taught in your class are, they. Suppose, I have got a finite length sequence now for the time being forget all this adaptive filter and all, this is pure DSP we are back to, we are go we are using finite length sequence of length capital N. This N has nothing to do with that filter length of capital N, this is a separate topic I am starting. And then I will use this result to for the effective, efficient implementation of that logarithms algorithm.

(Refer Slide Time: 18:38)



Suppose there is a sequence x n for n finite length. What is this DTFT? DTFT, you know summation x n discrete time Fourier transform all of you know this is a periodic function of omega. You plot over 2 pi, you either plot it from 0 to 2 pi or minus pi to pi and this will repeat all those things you know, it which is the magnitude for phase part both will repeat because it is periodic. Omega is the digital frequency its unit is radiant, all that is fine it is the final length sequence I this. But for finite length sequence, my claim is you do not need this n, see omega is a continuous variable here, from minus pi to pi. But my claim is you do not need the enter envelope, this continuous function from minus pi to pi, to have information about x n.

We can leave happily by just taking capital N, number of samples or more than capital N number of samples. N is the duration of the sequence of this equi space samples from 0 to 2 pi could be from minus pi to pi also, but 0 to 2 pi I speak to that is enough for us. Why you can very easily see, actually this is 1 short curt we have teaching DFT, but actually way is you know when I will be teaching vector space there we will see. There are 2 signals can be decomposed I mean can it has a linear combination of basis vectors, and capital N number of basis vectors. And choice of basic vectors gives you D DFT or DCT or wavelet transform or hard transform, hadamard transform all those things you know. But any way when I teach vector space that time, again I make just make some remark, but here quickly.

Suppose, I take the, I sample it sample at N point capital N over points at these frequencies omega k equal to. So, what is the range 0 to 2 pi capital N number of points, so each the gap is 2 pi by capital N I am sampling in frequency domain mind you. It is still mathematically it is nothing but sampling operation is done, but in frequency domain, function is function of omega, but actually sampled. So, that is why here also early as in and other things can come up, I will not going to all that detail. 2 pi by n this is the basic thing basic step times k, k can be 0 1 dot, dot, dot.

So, you understand 0 then 2 pi by n then 4 by n 6 pi by n these are the points I will take the sample, Kth sample will be denoted simplify capital x within bracket k, is nothing but this guy at omega equal to 2 pi k by n. This is then what, simply e to the power minus j replace omega by that formula, this particular discrete frequency 2 pi into 2 pi by n times k, k times k 2 pi by a into small n. At this specific omega you are finding out the DTFT and you have simply calling it x k, X k is called the kth DFT, DFT means discrete Fourier transform DFT. So, how many such DFT'S are there, DFT coefficients? Capital N and my claim is this N samples of DTFT, which I call the n DFT coefficients.

They are enough for me to get back x n and from x n, if I know x n I can get back these also. So, the n omega can be written in terms of it samples, in time domain sampling also you know when you do this sampling and all that. If you follow that non-aliasing condition of NY Quist then, I do not know how much you studied then the entire function of time band limited function, function of time itself can be represented by its discrete samples, by the samples taken at 0 capital T 2 T 3 T T is a sampling period. So, you do not need the entire information, that is the beginning of DSP, that band-limited functions can be represented just by the samples taken at a appropriate rate.

Entire f of t can be written as you know that formula NY Quest interpolation formula in terms of sin function sin can delete, sin can delete, sin can delete, sin can delete, sin at each sampling point there will be a sin. Sample times this sin, another sample time this sin, another sample time this sin. So, and if you sum them you will get the entire n value. So, which means you know I mean you do not need that n value to be stored, you it is enough if you need the, if you have the samples. Similarly, here also same at mathematically sins are same, because when it is mathematics it does not care which is time or frequency variable is a function of variable is sampled.

Similar thing will have happen here, here also my claim is this guy, but here I have not established this band-limited thing and all those things. Instead of band limited with time limited, that is why is time limited you see. Then it was band-limited, this is time limited I wish I could teach more, if you want to read more you see my video cassettes available in the net on multi-rate DSP and wavelet or something. Read the, I mean see the lecture number 1 to 13, that covers most of the basic of DSP. So, these things I dealt with at lengthier as you can see, but this is not a class on that. In fact, I should never even gone into DFT I would have simply used background knowledge of DFT, but I know how strong the background knowledge is. So, I am studying from DFT.

Anyway my, I come back to my claim my claim is this n samples are enough to get back x n and therefore, this. That is very simple, this is x k equal to this suppose an x n duration is capital N, out of this suppose I want to find out a particular sample say mth sample. What I will simply do you know, this is common in all Fourier transform another Fourier transform, inverse transform DDFT inverse DFT. You, if you want to want find out the particular thing say x, small m you multiply both side e to the power plus j 2 pi k by n into small n and sum over k, k again how many capital N. Here also you will have double summation then, you multiply this by e to the power j 2 pi k m by n and then, sum over k double summation, I will interchange the you know very obvious step always.

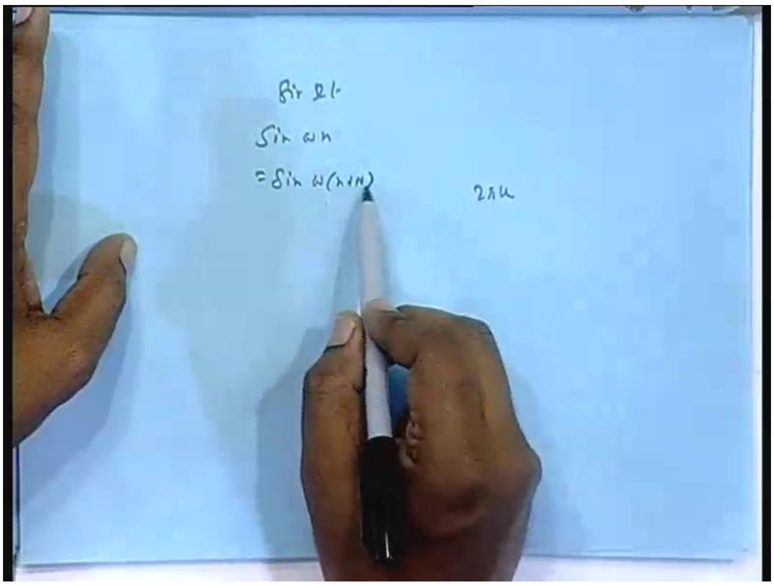
x n is dependent on n only, so it comes here, k here and e to the power minus this is already there and e to the power plus the same thing with m. So, you can say j 2 pi k by n look at this summation, m is your choice small n is your choice. Now, n is varying for each n you are running the summation for each n fix from outside. So, when n becomes equal to m here m, m because n will become a, because m is taken from this range only. When m hits m, that time it is 0 e to the power 0 1 1 1 1 1 added n, n times x m. So, 1 term is n times x m when small n equal to m and now, consider the situation where small n takes other values other than m. That time it is an integer, positive or negative I do not care and then you know this is a gp series in k.

If you say a equal to the e to the power j 2 pi by n into m minus n and m minus n is nonzero now, if you call it a simply it is e to the power k. So, 1 minus summation will give rise to what 1 minus e to the power n by 1 minus a. This summation e to the power n means N, N cancels e to the power j 2 pi into integer is 1 1 minus 1 0, so that we will be given rise to 0. So, we will get back this which means x m is nothing but 1 by n times this. So, you can get back any sample actually, the reason why it becomes 0 it is like you know it you know in the case of continuous signals, if you have cosine sinusoidal signal and you integrate it over a integral number of period it is 0.

Here also for this choice of this 2 pi k time by N, with this kind of choice of omega or 2 pi into some integer by N, K if you go on adding you are basically know, you will be completely 1 or more that 1 period exactly. And that it why it is become 0, it is not for any small omega here. Do you know this that sin analog frequency I denote it by capital omega, sin capital omega T or cos capital omega T was periodic, period is 2 pi by capital omega. But sin, small omega N it is not periodic in general, do you know?
Student: ((Refer time: 28:16))
Sin omega is not always periodic, because for that we of again getting divert it sin omega N, omega is radiant here earlier I said sin capital omega T radiant per second analog. This always periodic, but here is not in general periodic, if is to periodic then periodic then period suppose is equal to capital N. So, there must be some capital N so that, this. So, omega into capital N should be other 2 pi or 4 pi or 6 pi or something solving for those kind of choices.

Student: ((Refer time: 28:46))

(Refer Slide Time: 28:51)



But it is their problem, now it is coming. They are looking here and there, they are not.

Student: N cannot be a multiple.

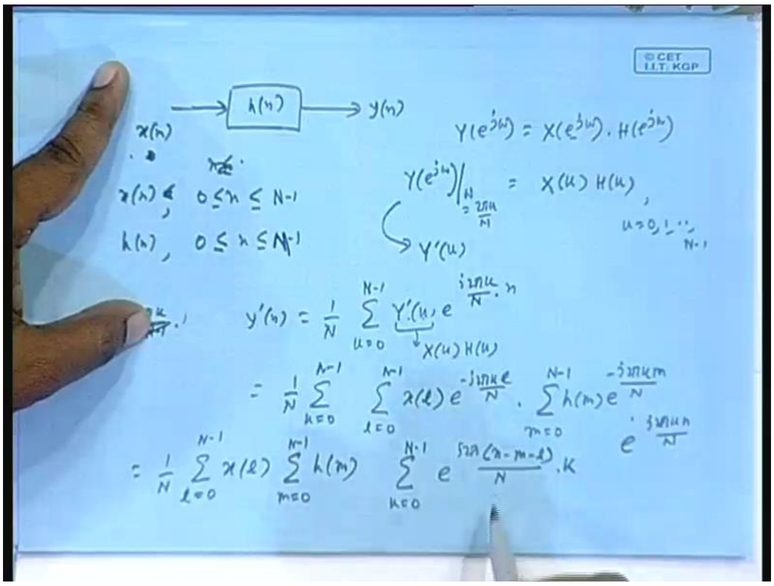
So, capital N is a integer. So, omega into capital N as to be your 2 pi or 4 pi or 6 pi or something, is it not? Integral multiple of 2 pi. So, 2 pi into k, omega n should be 2 pi k that is omega should be 2 pi k by N and that what is coming here, what is that? You understood that, is it not?? So, that is why this periodic thing comes up and that is I made the statement that you will really integrate the full number of periods for this kind of choice, but mathematically you can see it is 0. So that means, instead small m now I write the general form for any x n, any sequence x n it is 1 by n this is called inverse DFT formula, from the DFT coefficient.

So, how to get x n, but remember you treat this formula and that is how it should always be thought as what, for each k this is the signal. I call it carrier or a basis function for each k here, this is a sequence n takes various values this is a sequence from 0 to n minus 1. K equal to 0, you can view thought it is a dc, k equal 1 first harmonic, k equal to 2 second harmonic, k equal to 1 fundamental k equal to 2 first harmonic like that. So, various sequences they are multiplied by some coefficients and added. So, it is a linear combination of some basic sequences. One sequence is for k equal to 0, another for k equal to 1 like that.

So, it is a basically combination of some basis sequences. Their linear combination gives any x n, only the coefficients carry the information over x n. This is the vector space statement, which we will steady later that any x n can be written as a linear combination some basis sequences. Basis sequences have to basis sequences, they should sorry, in fact, they are orthogonal sequence all those things will study later, I will comeback I mean when I teach all that in the contacts of orthogonal projection and all for estimation problems. I will again take this example that look I did this that time this is what happens, this IDFT this is your DFT this is IDFT, fine.

So, I prove my claim that from the sample of the DTFT I can get back x n and from x n putting back this x n here, I can get back capital X e to the power j omega. So, DTFT itself can written by its own samples, DTFT samples are we will have to write the express the DDFT mathematically.

(Refer Slide Time: 31:40)



Now, look at this that suppose I have got a sequence, finite length sequence here n this is from, this is h n this is say y n, x n and h n are finite length. For the time being suppose you assume that both are same length, if not you can add some zeros are being themselves same length, length is say y n. That is here n is less than equal to that is x n or if you want you can wrote like this, x n is for n equal in the most general case say capital N minus 1 0 h n, not n m this is m length n length m. For the time being, I come back make them equal to N, that will be easier for explanation. So, both are length N.

Now, you know 1 thing y e to the power j omega, this DFT this see is x e to the power j omega into H e to the power j omega, that I think all of you know. This much DSP was covered, so this product you know, where this equal to this. Therefore, if I take sample this and if I sample this, I know if I sample this function, its length is capital N like this capital N. So, if I take N samples here N samples here those samples are good enough for this. Now, suppose I take this product only, so this km is nothing, but DTFT at the particular frequency DTFT the particular frequency.

These 2 if you may multiply, you should get back this at the same frequency, is it not? 2 pi k by N and k is 0 1 dot, dot N minus 1. Question is if you call them say for the time being some instead of Y k, I am writing Y prime k. If I used them in the inverse DTFT relation, this from the sample inverse DTFT why get back y n. Answer is no, why?

Student: ((Refer time: 34:18))

He understood, y n length of y n is what, not N 2 N minus 1. So firstly, its DTFT should be sampled at least 2 N minus 1 times and they are the basis function should have been, e to the power j 2 pi k by 2 N minus 1, not N times N, multiplied by the corresponding coefficient y k or what do by you say. And k will go from 0 to 2 n minus 2, total is 2 n minus 1 then it is okay, but I am not having that. I am have a only n number of sample and like a full, I am just running the IDFT as do the sequence length was capital N. So, summation will be from 0 to n, n minus 1 this this thing same 2 pi k by N only here not 2 n minus 1 0 to n minus 1 only not 2 n minus 2.

So obviously, I will not get back the correct y n you understood this problem, I see that the fine I would not get, then what do I get? We can evaluate that do I get, what I get will not be y n, y n obtained by linear convolution with the x n and h n. You all know linear convolution graphical we have doing it please do not ask me to do that. What y n will have, will have not only the linear convolution then be another component there is the another component which is the error component we will see. Together will be the total y n, this y n the way it is generated that is 2 DFT multiplied and taking inverse DFT, it is called circular convolution or periodic convolution.

Which will have a component of linear convolution plus an erroneous component, I have to then may do some manipulations to through I that to do with that component, that to make sure that extra component is 0. But before that let us revaluate what we will be get, if you still stick to the range 0 to N minus 1 only, what doing it. So, suppose I call them y prime k that is why I am not calling them y k. If it is strictly y k, I should take 2 N minus points of samples of its DTFT and all that I am not, I am not, I am simply calling the product y prime k I will like a full, I am putting them in the IDFT formula hoping that I will get back y n, which I will not.

So, what I will get back it is not y n, so let me call it y prime N and I will see how much difference is there in y prime N from y n, I will told you linear convolution plus an extra components will be there in y prime n. So, what is y prime n is 1 by N that IDFT formula, but with these coefficients range is as before not 2 N minus 2, but 2 N minus, but N minus 1 only, times e to the power j 2 pi k by N into small n IDFT. And now, y prime k you replace by this x k, H k and x k H k, because I want to finally relate everything in time domain, is it not? Linear, how it is related to actual linear convolution, linear convolution is time domain.

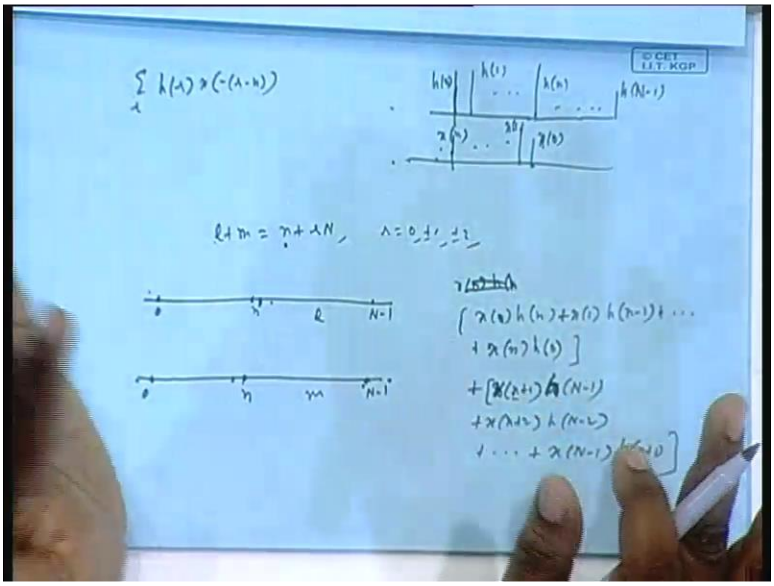
If I put y prime k equal to x k, H k see on the left hand side I have got N, right hand side x k H k not it is a small x n not h n. So, again x k has to be written by its DFT expression involving x n, H k is to be replaced by its DFT expression involving H n I will triple summation kind of thing. So, that will have at least relate y prime sequence with small x n, small h n and that I will simplify and I will see that there will be some component which is coming from the linear convolution, between the 2 and an extra component. So, y prime k is you for next step is you replace by these 2. Simply that is put this equal to x k into h k and next step, will be replace x k by its formula DFT H k by its for further formula that DFT sample, DTFT sample.

I might go fast here actually, because by now it should be familiar with what I was doing, this is x do not use n here, n is fix by your choice, X k. So, use x l e to the power minus j 2 pi k, k is your, k is fixed from outside summation K l by N, l from 0 2 N minus. This is your x k, we are something that DTFT of x similarly, here h k same thing, but use another index h m. So, x k h k and this other term e to the power j 2 pi k n by N. Now, 3 summations we have to interchanged the summation, this k summation push it absolutely inside, that in the inner most. That this over x summation or l, so only x l remains here, this involves k this term in involves not only l, but k if I have a inner summation over k.

So, this will go they are because it has k, this k this exponential solves going us inside, because the inner more summation is over k, are you following this? This outer was becomes inner most whichever has k that those are pushed in, this has k, this has k, this has k. Firstly, all of you have seen earlier that whenever this argument N minus m minus l is non-zero, then this is DP series. Because, series in terms of k you call the entire thing e, e to the power k, because will be 0. So, this will have value only when m and l they are taking various values, now they are taking value so that either m plus n equal to n or n plus capital N or n plus capital 2 N 2 capital N and all that.

Then, this will become equal to 1 e to the power 0 1 and 1 1 1 1 1 N times that N will come, this N will cancel and we have got this things. Some term of x, some term of h depending on the particular l and m we had at that time, the product 1 x 1 h fine, isn't otherwise 0.

(Refer Slide Time: 41:13)



So that means, in the summation what we have to do, only when l plus m is equal to n plus r into N, r can be anything. r can be 0 plus minus 1 plus minus 2 you understand, n or n plus capital N or n minus capital N and you put back here, the entire thing will be e to the power either 0 or e to the power j 2 pi integral multiplied 2 pi and all that you will get one. Only then this will be non-zero 1 1 1 N times and N will cancel x of particular l and particular h or particular m as coming from this it will be multiplied. So, now let us see what happens then, please see that books do not discuss this individual, you just work out a formula, but this will tell you handled DSP.

Let us draw to equal, axis, this is for l 0 to N minus 1 this is for m, N is also varying from 0 to N minus 1, so thus m. Suppose l is 0, then m is varying 0 1 2 3 only when m equal to in this range small m, because small n you are speaking from the range 0 to capital N minus only. So, when only when m hits small n this is satisfied or when n hits n plus capital N, but can small m be m plus capital N answer is no. Because, all the indices have got limited range 0 to n minus 1, small n plus capital N means we have jumping beyond the range so that, that will not occur. So, only when, so for l equal to 0 small n equal to n that is enough.

So, x 0 into h n then again this will be satisfied you will get 1 1 1 1 at it N times and N cancels. So, 1 term will be there is suppose, this is the n this with this, 1 term will be x 0 into sorry, x 0 into h n then l takes 1, l takes 1 the moment l takes 1, m should be what n minus 1. So that, this is satisfied l is 1 n minus 1 you understand this is increasing this this means what summation small n. M is n minus 1 or n minus 1 plus capital N, but that will go again beyond the range. So, only m minus 1 is fine dot, dot, dot l is increasing when l touches small n, that is x n m should be 0, 0 plus n still n 0 or if you jump again 0 plus capital N you are beyond this, not allowed.

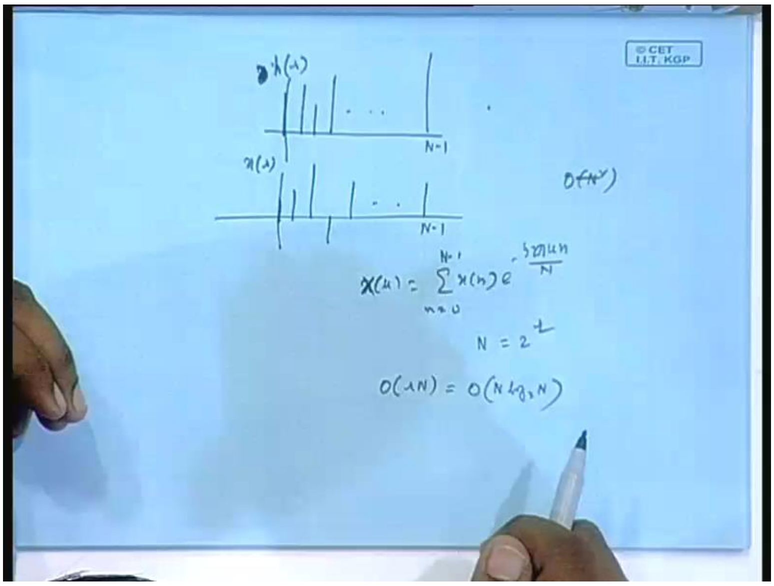
So, x n into h 0 other terms, but look at this, what is this, what is this expression? Linear convolution between x and h evaluated at nth index, but I will, I will again come to this. Let us come to this further, now l is increasing n plus 1. So, n plus 1 means it should be m should be 0 minus 1 then minus 1 and 1 cancels, it become a small n, but minus 1 again beyond the range and what do I do? From there if I jump to right by capital N I hit back a valid point capital N minus 1, is it not? So that means, h n plus 1 into x h or x sorry, this x this is h, X n then x n plus 1 h n minus 1 and n as you go on increasing, if it is n plus 2, this would be capital N minus 2.

So, on and. So, forth easiest way to remember is, if you add the 2 indices they should amount to small n plus capital N here. Here if you add the indices it should amount to small n. This is, this extra component that is coming, this is the extra component that is coming out. What is this component those who do not, cannot see that is circular linear convolution look at 2 sequences x n h n. Say, h when you convolved you know the formula, you know how to convolve graphically you, all of you know this. Say h r x n minus r, n minus r is r minus n summation over the available range. So, before r minus n you are given say x of n. So, reverse it say x of r reverse it, so it become x of minus r the instead of r replace by r minus n that is shift it by n.

If n is positive shift it to the right by n and then multiply that 2 sample wise add this is the convolution and suppose it is h 0 h 1 dot, dot, dot, dot, dot, h n dot, dot, dot h n minus 1. X also was a sequence like this from here to here, first you reverse it because it is minus and then shift it by small m. If you reverse it x 0 x 1 x 2 dot, dot, dot on this side and then if shift it to the right by small n. So, x 0 comes below this x 1 comes here just shift it dot, dot, dot x n comes here where if it is 0, so if it is 0 it is n and if you multiply then, another terms here get multiplied by 0, 0 multiplied these terms. So, forget them only these part you multiply, you get back x 0 h n, you understand.

These are linear type convolution, but that is that is what I wanted, but I am getting the extra term. These together is called circular convolution, before I can eliminated this terms by some trick and all that, I should have a graphical way of interpreting this and that graphical way of carrying this computation you know, graphical way like this is a graphical way of linear convolution. So, graphical way of doing the entire thing I will now present then, you can verify it.

(Refer Slide Time: 48:35)



Just follow the procedure that I suggest, that suppose you are giving 2 equations, 2 sequences x and h. See I have given 2 sequences say h r and also x, shall I just dictate or shall I write? Before you do the circular convolution, what is circular convolution actually you are doing n point DFT n point DFT multiplying inverse DFT. You will get a end point sequence, what is that? That is not linear convolution, just a kind of complicated thing which you evaluate it, how to get that graphically that is what you are saying. That is called that sequence is call the circular convolution output between a periodic convolution output, that is obtained by periodic convolution between the 2 or circular problem between the two.

Now, first step is hold 1 sequence as it is do not change h say, next step is make a periodic version of this. That is construct a sequence by repeating this again, again, again to the right and to the left. I am not doing all those you know. Third is reverse, third is reverse this that is unlike linear convolution where you just reverse this, here do not reverse this the given 1 make a periodic version then reverse. After reversing again the same step shift it to the right by small n, these 2 are common reverse and shift, these 2 are common, but after that when you shift to the right by n still these are infinitely long sequence.

Now we use a chopper, that is our window that is to right of capital N minus 1, whatever you have in that shifted periodic sequence through away 0 to the left of origin through away 0. Again you got a finite length sequence of length n, now multiply sample wise from h sum you should get that, that is my claim. And that you can verify I do not know how much time we have today. Suppose you do this, you make I mean I am not showing the intermediate steps. Suppose you make periodic repetition of this, reverse it and now you will go on shifting, you will go on shifting. So, today there is no time I would suggest that you take it up on your own.

And next time we have to consider, because you once you start it, it will take about ten minutes also is graphical business, but that is very important. Once that is done, then some property of this we will see of the circular convolution and that will be using in the filtering part of that block adaptive filter. Then, again we come back to the DFT some property of circular shift, not originally shift circular shift, we will have to see, that effect will be used in the computation of the weight update. That weigh update part you remember, there is a summation some data vector times errors it is actually some kind of correlation.

So, using DFT what will the using you know circular correlation property of DFT and circular convolution will be in the filtering part, convolution, linear convolution I will use circular convolution cleverly to get linear convolution error. Similarly, I will use DFT and the circular correlation property to use to get that correlation. And this DFT remember, why I am may be just 2 3 minutes why you are bothered about DFT, because there are very fast algorithms available. Like you know I mean if you see this simply DFT computation, x k for each k how many complex multiplications? Capital N, is it not? n samples have complex multiplications and n minus 1 addition.

Each complex multiplication means, how many actual multiplication 4 a plus j b into c plus j d 4, but cleverly you can reduce it to. So, that I will tell you some other day. People have you know found out way of adding some term and subtract some term and it can be reduce to 3 some to 2 adders will go up, but even they that is quite huge. So, if I assume order I mean n multiplier and how many k, n k. So, order n square and suppose you are having a sequence of length 500 2 L, you understand how much computation 502 L square. But there are first algorithms where, say if N length N is power of 2 say 2 to the power r then, it should have r into N or equivalently order of N log 2 N this can be reduced.

So you understand 500 2 L means, 500 2 L square and here it become 5 2 L into 2 to the power 8, 8 wrote down to that level that makes it very fast. Unfortunately, I cannot things this here you say FFT and all that, but there is a reason why you go for this and there are plenty of excellent architectures available to implement it, there is a motivation for going into DFT. So, I stop here today please come with some preparation of this. So, that you know the recapitulation time is less.

Thank you very much.