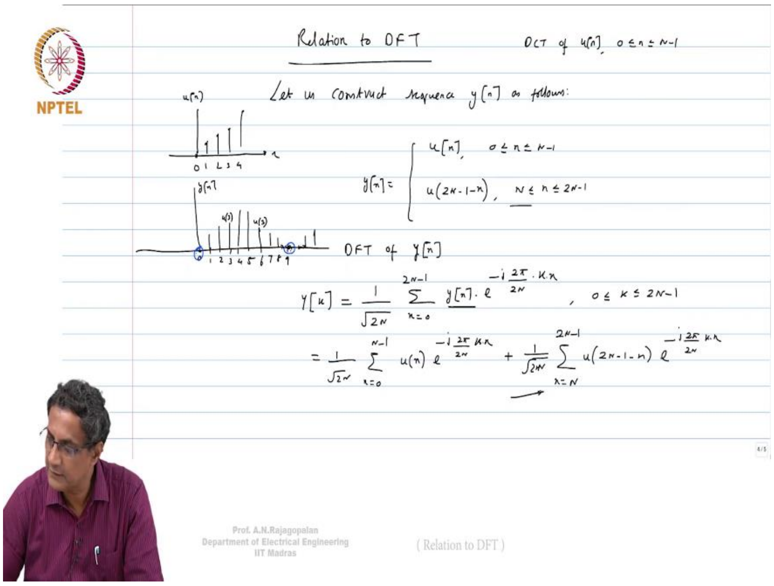
Image Signal Processing
Professor. A.N. Rajagopalan
Department of Electrical Engineering
Indian Institute of Technology,
Lecture 49
Relation to DFT

(Refer Slide Time: 00:16)



Let us kind of, so like I told you. So, we can, we have to do, we have to show that this is related to, to a symmetrically extended sequence of the actual sequence you started with. So, what we will do, so its relation to the MN, so the claim is that it can be computed at the same speed at which you compute the DFT. Separable, of course, it is also separable. So, when you go to higher dimensions, again you do not get hit. All your computations wise we appear quite good and on top of that it has these other properties. So, naturally that makes it attractive.

So relation to DFT, so which then means that if you can show that this is related to DFT then it means computation wise, computation wise you can in fact use a DFT to now arrive at this. But for that we have to first of all establish what is that, what is that, you see the relation between that I, that this has with respect to DCA with respect to DFT.

So, in order to in order to establish that, we will have to go through a small little sort of, sort of a proof. So, let us construct, let us construct sequence yn as follows. So when you construct it, construct it as yn is equal to un. So, that is, so you started with the sequence un. So, you are looking at DCT of u of n from 0 plus n, n minus, so set as a length n and we want to show that this is somehow related to the CDFT of this guy extended in some manner.

So, y of n, so we construct y of n to be equal to u of n for 0 less than or equal to n less than or equal to N minus 1 and then it will be u of 2N minus 1 minus n. So that is a symmetric extension for N greater than or equal to n and less than or equal to 2N minus 1. So, you have sort of extended the sequence such that it has a length 2N now.

So now, if you compute DFT of yn compute DFT of y of n, while that be, let us call this as yk, and what will this be? 1 by root 2N. Because you have 2Ns like the sequence is length 2N, if it was N length, we would have written 1 by root N, there is length 2N now.

So, 1 by root 2N summation going all the way from 0 to 2N minus 1, because yn has 2N length, 1 minus 1 then yn e raised to minus j 2 pi by, I cannot write n now, I have to write 2 pi by 2N, whatever is the length here. e power minus j 2 pi by 2N kn and k going from 0 to 2N minus 1.

See, it might appear to you that initially on the face of it, it might look like, Oh then right, does this mean that I have to take a hit of a 2N point DFT in order to which may look like more work and sort of n point DFT. But we will show eventually that you do not actually compute anywhere at 2N point DFT. This is only to start the exercise and then along the way we will show that actually it is not, it is not required to compute at 2N point DFT.

So, now if you look at it, suppose I have this sequence, let us go back to the sequence, where you had 0, 1, 2, 3, 4, 2 3 4, and this was your un, let us say, like versus n. So, what you have done is, so this u0, u1, u2, u3, u4, those are the values. So, you are computing your yn now, you are plotting yn versus n. So, from 0 to 0, 1, 2, 3, 4, it looks exactly like un and at 5, if you look at it at 5 or come here to this below guy, so what is the length of this? It has a length n equal to 5, so it will be 10 minus 1 9 minus 5 4. So, which means that this is like u4.

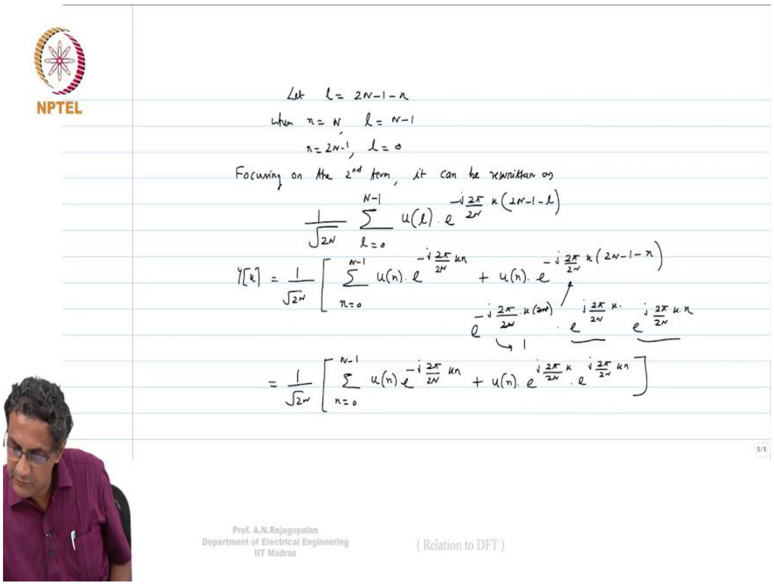
Then 10 minus 1 is 9 minus then 6 is u3 and so it will be like u3. So, this u3 and then u3 u2 u1 and then 6, 7, 8, wait a minute 8 and then 9. So, at 9 you would have hit u0. So, computing a 2D DFT rate of the sequence means that this guy will start to again, so this whole sequence will start to repeat. So, it means there are 10, you will have this and then it will start to go up. So now, you can say of course on this one also repeat from the left.

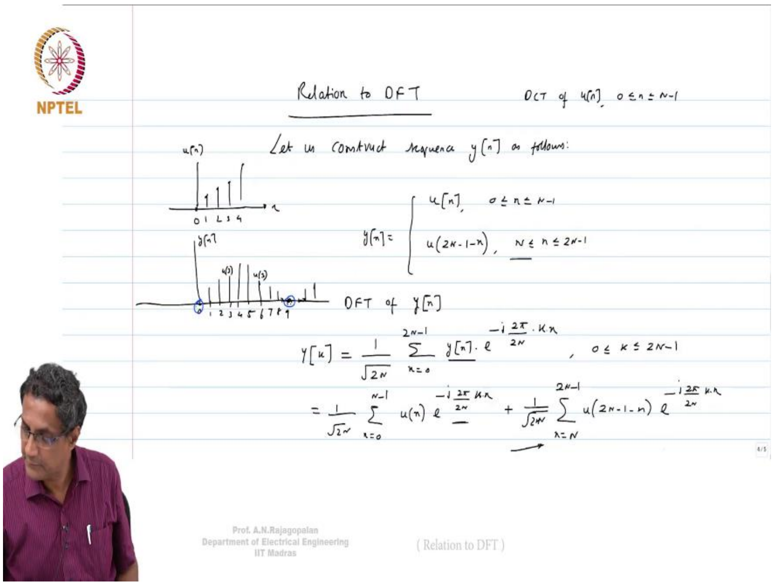
So, you see that, you have made sure that, that this and your and that value, which is the beginning and the end value of y of n. They are, they match. They are equal and then and that is one way to see that this false jump that you got otherwise is gone. So, un can be anything arbitrary but then once you extend it in this manner then it's clear that I get a yn sequence wherein I would not see this unnecessary jump happen.

Though I been, see different people call it in different ways, they call it spectral leakage for a DFT, different people call it by different names. Now that we have this, what we will do is we will, now yk. Now let us just, let us just split this as 1 by root 2N summation n equals 0 to n minus 1 and for that yn is un, e power minus j 2 pi by 2N and in a hurry, we will not try to cancel these two.

Do not be in a hurry to cancel it out, because you need that 2N when you look at that cos, in the bottom it has 2N. So, that is why we do not knock off this 2N. We keep it, we carry it along with us. Plus, then 1 by root 2N to 2N summation and N whatever you want to call it, let us call it M or N, it does not matter, we know that, N is equal to see N equal to N to 2N minus 1, correct and in that range, it is u of 2N minus 1 minus n e raised to minus j 2 pi by 2N kn and so, we have only split it into, into what we know, what yn is like over 0 to n minus 1 or what yn is like over n to n, n to 2N minus 1. Now, if you focus on this expression.

(Refer Slide Time: 07:51)





Let us say, let l be equal to what is that? 2N minus 1 minus n. So, when n is equal to, when n is equal to what is that? The lower limit is N, so l is N minus 1, correct and when n is 2N minus 1, l is 0. So, if so focusing on the second term, focusing on the second term, what do you get? So, you see that, turns out that, that can be written, it can be rewritten as, be rewritten as 1 by root 2N summation anyway l equals 0 to N minus 1.

I can write it as going from because that is all, those are all only terms that are involved and then, what do you have? u of l e raised to minus j 2 pi by 2N k and then replace n by l. So, that will be 2N minus 1 minus l, k still goes from 0 to 2N minus 1. This is okay or In other words we can come back to the expression and say that yk is 1 by root 2N, 1 by root 2N and then you have, then we can actually sum it up.

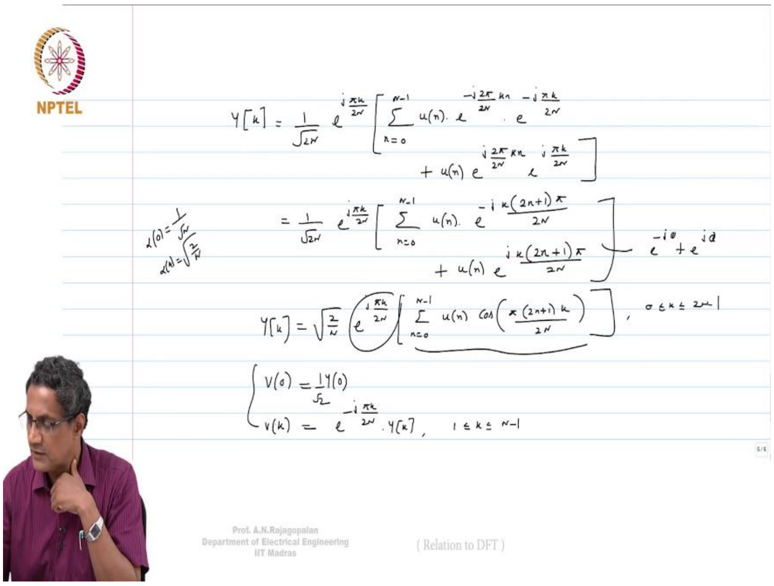
Because this sum is also going from 0 to N minus 1, that sum was also going from 0 to N minus 1. So, simply write it as n equal to 0 to N minus 1 except that first case you had un e raised to minus j 2 pi by N kn and now, you have plus un e raised to minus j what can we do, 2 pi by 2N by, no wait a minute, k 2N minus 1 minus n. Now, if you, if you look at this one, e raised to minus j 2 pi by 2N k into 2N into e raised to j 2 pi by 2N into k ej 2 pi by 2N kn.

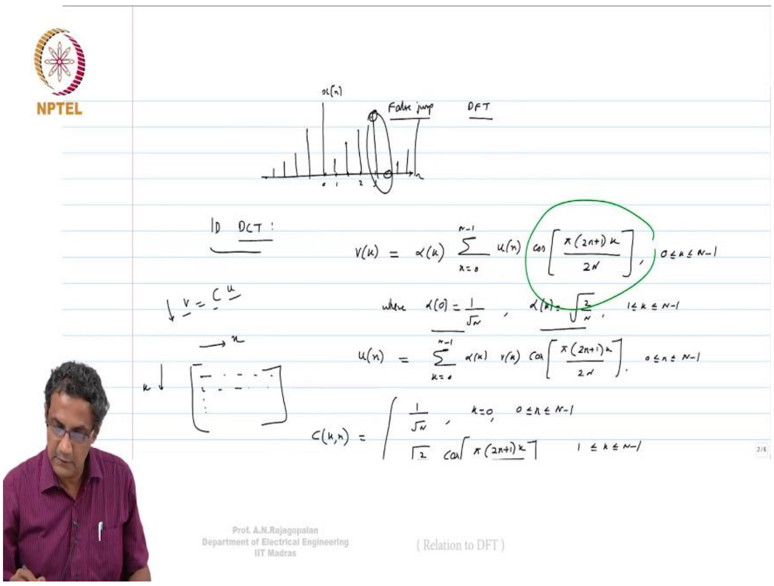
Student: (())(10:17)

Professor: 2 pi by 2N, yeah. So, omit is here? Thanks. I missed that. Yeah, that has 2 pi by 2N. Yeah, here it is, 2 pi by 2N. Yeah so, now if you look at it, this 2N 2N will cancel off and ej minus 2 pi k, for k integer is 1. So, this just goes out and then you are simply left with this term and you are left with this term.

Even this becomes 1 by root 2N, summation n equals 0 to N minus 1, un e raised to minus j 2 pi by N, 2 pi by 2N kn plus un e raised to j 2 pi by 2 N k e raised to j 2 pi by 2N kn, and now, you play a small trick and all this done because to bring it in that form. So, what would you do? You would take out pi, what is that? pi k by see 2N out. Take pi k by 2N.

(Refer Slide Time: 11:34)





So yk, so e raised to, where is that? 1 by root 2N. So, you take out ej pi k, pi k by 2N, take it out. Then you have a sum going from n equals 0 to N minus 1 and then you have un e raised to minus j 2 pi by 2N kn, but though this guy never had any pi k by 2N term therefore, it will become e raised to minus j pi k by 2N. Because it does not have anything like that. Next one will be un and what did it have.

It had ej 2 pi by 2N kn. So, e raised to j 2 pi by 2N kn. The other term, the right is this one which is j 2 pi k by 2N and you have already taken ej pi k by 2N out, so this becomes ej pi k by 2N because of this term. So, this is 1 by root 2N e raised to j pi k by 2N summation n equals 0 to N minus 1 un e raised to minus j k 2n plus 1.

See, so here it is 2n 2 pi by 2N when I pulled out k is common and pulled out k pi, I pulled out as common. So one case, you will get 2N and other case you will get 1 and minus j of course is common, plus un e raised to j, I mean you should do the same thing here k, this is k it looks like u, should be kn. ejk 2n, the pi we will take out.

So, ejk and then pi on this side by 2N. So, now right you start to see the similarity, so like e power minus j this one theta plus ej theta, un is common. So, this e raised to, form e raised to minus j theta plus e raised to j theta, theta is the same, minus theta plus theta. So, therefore, this will make it, so you will get 2 cos theta, so the 2, lets knock off here.

So, you will get root 2 by N and then e raised to j pi k by 2N summation, n equals 0 to N minus 1 un cos pi into 2N plus 1k by 2N. Less than or equal to 2N minus 1, still. This is yk. So, now you see that right, I mean see, when I, when we kind of say derive yk, we simply took actually a DFT.

We did not take a DCT at all, we took a DFT. But then eventually what we have ended up showing is that, is that, that this expression sitting inside this is of course a DCT of obviously of u of n. You can see a DCT sitting there. So, which actually, which actually means that, which actually means that if you, if you are interested in vk, which is actually, which is this. This is what you are interested in.

So, what you can do is if you want v0, let us say, because v0 will mean that this guy will drop out actually because if this is become 1 and then, so what is that relation that we had? So, 1 by root N. alpha 0 was what? So, you need mn. So, you see, your v0, no this involve the alpha 0 was 1 by root N, whereas alpha k for other values was, what is that? Root 2 by N.

Student: (())(16:16)

Professor: What is that1? Root 2 by N or 1 by, root 2 by N, I thought.

Student: (())(16:23)

Professor: What did I write there?

Student: Root 2 by N

Professor: Root 2 by N. So, for other, other alpha. So, now keeping that in mind, so if you look at it, so your alpha k which is multiplying it from outside. So, this expression had an alpha k, extra, if you go back, here we had this alpha k sitting here. So, this inner one is what you are seeing out there, this expression you are seeing and there is an alpha k that also we have to account for. So, the alpha k is such that for k equal to 0 is 1 by root N. So, if you put k equal to 0 and if you, if you want, so it's like saying that if I take y0, so y0, if I want to express it in terms of y0, but then y0 has this root 2 by N.

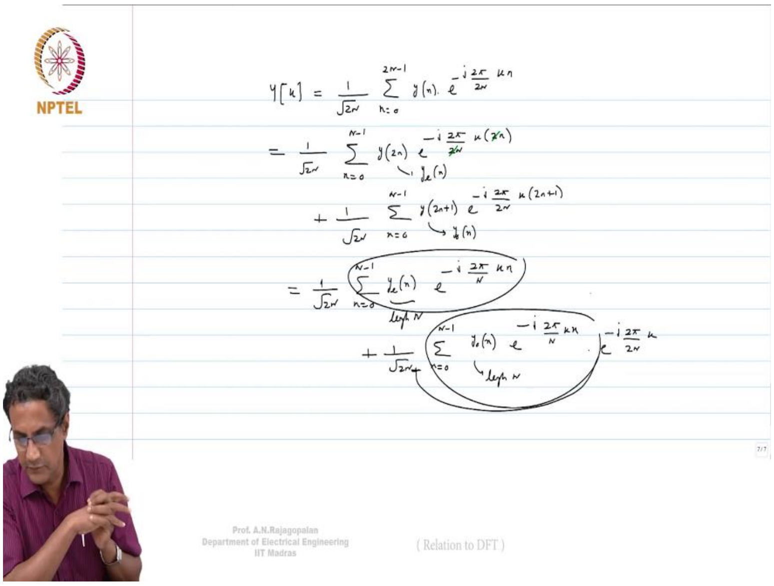
So, if I want to knock this off, so I should simply multiply it by 1 by root 2. If I do 1 by root 2 times y0, then it will give me actually v0 because then it becomes like 1 by root N e power j pi k is gone, because k equal to 0 and then the other term is what we had there. So, for V0, this is what I can do if I want to derive from yk and for other k, vk, there anyway, I need a root 2 by N, that alpha k. So, all that I will do is cancel this guy off. So, e raised to minus j pi k by 2N yk, it will give me for 1 less than k less than or equal to N minus 1.

Correct, because I need k, v, for v, I need only up to N minus 1. I am not, I do not need beyond that. That is all I need to compute my, compute my vk. So, you see that, so the explanation that we had, why this can actually pack more energy in the first few questions is because implicitly computing a DCT, you can also implicitly visualize it as computing DFT of the symmetrically extended sequence.

So, if you are computing DCT of u of N is equivalent to extending it to a length of 2N and making it a symmetric sequence and then computing a, computing a DFT of that and because you know that, computing a DFT in that form, it does not involve jumps and so on. Therefore, you expect those questions to die faster.

So, one way to kind of interpret the fact that DCT can pack more is because, is because of its inherent relation with respect to with respect to DFT. Based upon what happens when you compute a DFT, you can now go back and explain as to why a DCT can be expected to do, can be expected to be superior, then actually a DFT. Although, there is also this relation with the DFT but that is in fact, what actually throws light on this. Now, the only thing that really remains is it looks like that I have to compute, compute a 2N point DFT, but that is also not true. So, this point is, the second point is easy.

(Refer Slide Time: 19:27)



Because, you know that, computing, computing yk, when you have yk going from, what is this? 1 by root 2N divide summation n equals 0 to see 2N minus 1 yn e raised to minus to j 2 pi by 2N kn. When you have this, we know that we can split this into, into even and odd, so we can write this as 1 by root 2N in summation n equals 0 to N minus 1. Let us say, y of 2N that will capture all the odd values. So, some I have changed from 0 to N minus 1, e raised to minus j 2 pi by 2N k into 2, see 2n, because n should be replaced by 2n.

Student: (())(20:15)

Professor: Plus 1 by root 2N, then you again let the summation go from 0 to N minus 1 to cover all the values. So, y of 2N plus 1, all, pick up all the odd values, e raised to minus j 2 pi by 2N k into 2, see, 2n plus 1. Now, this is simply some ye of n, if you wish, we can call this as all the even values sitting in y, these are all the odd values sitting in, sitting in y.

So these two have length n and now, you can cancel off this 2, so this 2 will knock off this 2. Here, of course, when you will get one extra term coming out, but the point is, let me write it down. So, this is like 1 by root 2N summation n equals 0 to N minus 1, ye of n, this is length N because the whole length is 2N minus 1.

So, this is length N e raised to minus j 2 pi by N kn, because as 2 and 2 got cancelled plus 1 by root 2N summation n equals 0 to N minus 1 y odd of n this is like the odd value, so this again length N and then e raised to minus j 2 pi by N kn into, so this other term is what? Into e raised to minus j 2 pi by 2N k and anyway, this is independent of n.

So, therefore, this can be brought out, this you know, this is the standard way in which you do sort of decimation in time. So, this can be brought out because it is not, it is not dependent on n, so it just comes out and therefore, clearly you can simply and computing this is equal to computing an n point DFT, computing this is again equal to computing n point DFT. 2 pi by N kn lengths N. So, it is the unit computing n point DFT and therefore, if we simply compute this and find out why just for the first N minus 1 values of for, just for k running from 0 to N minus 1, you have your vk.

So, what this shows is that, so because all operations can be done using a Fast Fourier Transform in fact, even though to begin with it was not so apparent but because all this can be done using an FFT. Therefore, the, what do you call? The order for this is N log N to the base 2, which you can compute the same speed at which you would compute an FFT. So, the computational complexity, if you talk about a DFT, a DCT computational complexity is in fact, it has real values. So, that way it’s even faster, but the order is like N log, N log N to the base 2.