Signal Processing Algorithms & Architecture

Dr. Anirban Dasgupta
Department of Electronics & Electrical Engineering
Indian Institute of Technology Guwahati

Lecture 10

Applications of FFT

Hello, everyone. Welcome to a new lecture. On the applications. Of Fast Fourier Transform. So, this is a part of the course. Signal processing algorithms. And architectures. I am Dr. Anirban Dasgupta. And let us get started. Okay, So here we are trying. understand three applications of FFT. Now applications I mean to say that. Not only that. We can use this FFT. The spectra for Some practical uses. But also how FFT works. Is reducing the complexity. Okay! So first, we filter long sequences. Second is frequency estimation. And the third is. Power spectrum. Density estimation. So, filtering is long. Sequences. What does it mean? Now, when I speak, Probably there. There may be some noise. Maybe the AC is making some noise. And I want to filter that out. And we already discussed it. That is when we want it. To filter such sequences. We can simply use it. A linear filter which does convolution. Now, what is the problem? So, suppose my signal is of length L_x and My filter is of length L_h. Now, if this signal is of length L_x, which is long. My number of multiplications is. It will be quite large. So, what is the problem? Let us not consider it. Additions as of now. Because of multiplications. There is more time. taking operation. So, The sequence. If it is long, then it is. will be computationally Expensive. So, can we do better? So, what are we? We will do it. will use The FFT, how? So we will break the signal into Small sub-signals and Then we will do the filtering. In the frequency domain. So what advantage do we have? I get by by filtering. The frequency domain. So, convolution in time. become multiplication In frequency, that is one thing. And the second thing is that the Conversion from time to time. Frequency can be measured. in faster way using the FFT. So, The two popular methods of Achieving this is called. The overlap save method and The overlap add method. These two are collectively. called sectional convolution. Because we are dividing them. Signal in sections. then performing The convolutions are okay. So, what are we doing? The first method is. overlap save method. So in this, we will. First, divide the signals. into overlapping sub-signals or overlapping blocks as The terminology states. And then what will we do? We will filter. Each subsequence. Now there is a rule For doing the overlap. So, first, let the input in. Sequence of length L_x. How? Is this L_x formed? We will provide an illustration. So this is it. whole signal L_x, okay? And we have the filter. Length m. Fine, fair enough. So now we will take a break. The input signal is into. Overlapping segments. And how will we overlap? So, say this is the signal. which is starting from this point. So, what is there before this? Since this is a signal, technically. This is the span of the signal. or the support of the signal, but beyond this the signal has 0 values. So, if I define a signal x[n], and which Is 0 to $L_x - 1$? Then. Beyond this index 0. The signal values are. 0 and beyond the index. L_x, the signal value is 0. So, the logic is that I will. Take m minus values. From the previous block. And in this case, there. There are no previous blocks. For the previous values. All of the signals are 0. So, for the first sub signal $x_1[n]$, we will pad m minus. 0s to the front. For the second sub-signal. $x_2[n]$, we will take $m-1$ value from the last. Minus values of this block. or $x_1\ [n]$. Similarly, for the third. sub signal $x_3[n]$, we will Take m minus values. From the previous block, $x_2[n]$. So, this is from the input side. What about the output side? Now, on the output side, Since these are formed. 1 block is technically redundant. So we can discard this. Discard this; discard this. And then just concatenate. the signal which is my Finally, that is it. But it might sound complex, so let us take an example. So, let us say this is my sequence and what it is. The length of the sequence.

$$x[n] = [3, -1, 0, 1, 3, 2, 0, 1, 2, 1], \text{ and } h[n] = [1, 1, 1]$$

So, my L_x is 10, and m is 3; this is the length of the signal. Now, practically, you would not. Get a length of 10, and you have it. To do sectionals. Convolution occurs because. You can do this directly. And that is why I have it. Taken a length of 10 because. I can show you that. Result of direct linear analysis. Convolution is the same as the result of. overlap save method. But in practice, you may not get. Get 10,000 or even 100,000 point signals, which is huge. So, what will I do? Now I will form the subsequences. So, first sub-sequence, what should it be? Subsequence length? What is my L? Now how? Many-point FFT or circular convolution. I want to perform that is very important. So here, let us decide that I want to perform 8. So here, let us decide that I want to perform 8 point circular convolution and my M are 3, which is the filter length. So what will L be?

$$L = N - M + 1 = 6$$

What does it mean? We will take six values from the signal. Then, $m-1$, $m\ -1$ is 2, so there are 2 zeros. 2 zeros or 2 values from the Previous block and 6 values from. The signal, this is my $x_1[n]$. What about $x_2[\ n]$? I will take these two values. is not the signal, this is the signal. So, let me erase this to avoid confusion. So, this is my signal. So, I take the six values from the signals. Similarly, these two values, The first 2m-1 values will come. From the last $m-1$. The previous block, which is 3 and 2, is correct. And the remaining values will be 0, 1, 2, and 1. And since I am out of signal, what is there after one? There are zeros up to infinity. So, I take as many zeros as needed to make it 8 or n. Now I will perform a circular convolution. Now I am in the time domain. Doing this circular convolution because. This is just an 8-point sequence, but practically. I want you to know that circular convolution is important. will be done as a multiplication. Frequency domain and this conversion. To the frequency domain will be. performed using FFT and IFFT which will take $O(n\ log\ n)$ time complexity. So, these two will be your results. So again, I see this is my$y_1[n]$. And this is my $y_2[n]$. Now I have to do something called. Overlap and save to find the final subsequence. So this is my first subsequence, which I call y[1]. And this is my second subsequence. Which is y[1]. Now, what should we do next? As I said, I will discard $m - 1$ values of y[1] and y[2]. This means I will discard these. Two and save the remainder. I will discard these two and save the remainder. And then if I concatenate, I get this signal and just check that this is correct. Exactly the same as the signal output that I got from direct linear convolution. So overlap and save works for this specific example and If you check out any problems, this will work. What about the next method, overlap addition? So here is what we are doing: we are not. Overlapping anything from the input side. So we will just take L blocks, L sized blocks, as x1, x2, and x3. But with L, I cannot perform the endpoint. Circular convolution or endpoint DFT. So, I have to pad $M - 1$ zeros at the end. Now, as I pad this $M - 1$ zero at the end, Now my sub signal length becomes of size capital N. Now, when I do the convolution, circular convolution, say, So I will get these output blocks. Now what I will do is just. overlap this $m - 1$ points and As the name suggests, it is so. Overlap added; I will directly add them. Again, if it is confusing, let us take an example. Let us take the very same example. So this is it. is my signal and this is the result of linear convolution and We have all this chalked out easily. But what we are doing here is taking 6. So 6, 1, 2, 3, 4, 5, and the remainder will be this. Of course, these two are since there are zeroes after this. So I have taken 0, 0, and these $m - 1$ zeros are padded at the ends. And now, do the math; now do the circular convolution. you get these two signals or sub signals as outputs. And as I explained, we have to overlap these $m's - 1$, $m - 1$ of the last and $m - 1$ of the first. If there were more signals, so you have To overlap the $y_2[n]$ and $y_3[n]$ again. I try to keep this as a simple exercise. And then, these are. directly coming, only these two portions will be added. So, $5 + 0$ is 5; $2 + 1$ is 3. And you check the result: this is your $y[n]$, which is verified. So overlap add also works. And here is just a summary of the comparison. So here you have overlapping inputs. Segments in overlap save. In the overlap, you have added. Non-overlapping input sides. But on the output side, you need to overlap and add. So here you are, overlapping. $M - 1$ samples are between the blocks. And here is the $M - 1$. Samples overlap at the output. Overlapped in this case means. You are taking from the previous one. Block, and here you are discarding the $M - 1$ samples because. These are coming with time domain aliasing, and there is no need for it. Discarding, you are just keeping them, but adding, and this is direct. Concatenation, and this is. Overlapping and then adding to get the results. So, these are the four main differences. With more practice, you will find things easier. So, let us move on to the second one. Example, which is frequency estimation. So, what is the problem with frequency estimation? Now, this is the analogue case. where I represent $x(t)$ as being composed of some components sine waves that are varying at different frequencies Having different phases and different amplitudes. Well, this looks just like the Fourier series in continuous time. But here we have something else: we have noise. So the problem is that I want to find exactly which frequencies are present. In the signal, or I would say dominant frequencies, because there are are something that is called SNR, or the signal-to-noise ratio. So if the noise is very high, you may know, and if the amplitude of a specific signal is also high, The frequency component is comparable to the noise or lower than it then that signal is buried under the noise. A practical example is when you are talking over the phone. And then some trains go and make a very loud noise; your signal is lost. Kind of like that analogy. So, what is the challenge? And this is exactly what I was talking about. That noise is very random. You cannot always distinguish that. which is noise, which is true signal. So, it is difficult to estimate the frequency. The second is the spectral resolution. What is spectral resolution? So, in this case, I said that in the Fourier assumption. Is that what this F1 says is your fundamental, and these are all harmonics? So, if you know F1, you can just say F2. F3 and others are just integer multiples. But there can be two frequencies that are very close to one another. Like I worked on a project to find Say a broken rotor is caused by a fault from the armature current. So your healthy motor will have a 50-hertz component and a. A broken rotor can have components that are 50.5 hertz. Now, seeing these two components separately is often a necessity. Challenge because they are very close, and then the leakage effect occurs. Adds to the problem because, say, if I want a 50 hertz signal, I should Ideally, we get a peak, but in reality, we do not receive a 50 hertz signal. We get a sign that is of a definite window, which means. Multiplying that sign by a window function. And we know that because of this windowing, instead of getting. A sharp peak gives you a spread-out peak, and probably this is 50.5. It will be spread out like this, so it will sometimes appear as a single peak. And this is the same problem as the closely spaced frequencies that I mentioned. And of course, aliasing is again a sampling problem. So, if you are sampling at a lower rate anyway, you get these problems. So you have to know very precisely about which frequencies you You want to detect, and you should appropriately sample at that frequency. So there are a lot of methods here, since this is an application. I will discuss the FFT-based method. But there are challenges, such as this closely spaced sinusoid problem. Where there are even better methods, like the autoregressive. Model-based methods, then music is a very nice method, multiple. Signal classification: these are all advanced signal processing topics. Then S-SPIRIT, which is an estimation of Signal parameters through rotational invariance. So let us see how the FFT will solve this problem. For simplicity's sake, I like to start things with simple problems. So here I see we have a signal

$$x(t) = A\sin(2\pi f_0 t)$$

and if you can always have a noise term, but for simplicity's sake. I said I do not have noise right now. And neither do I have a phase right now, but this. I will work with the phase. And the noise is such that the SNR is good enough. So, now what I will do is, of course, theoretically, this is again. Having a span from negative infinity to positive infinity. But we will consider a window, maybe from 0. To some value t, I will not say t. And then we are sampling this as Ts with a sampling rate. sampling duration. So, this makes our discrete time equivalent sequence as $A\, sin(2\pi f t)$ not n by fs. And now we will compute the DFT, or in other words, we are. Computing the FFT is typically this sequence. The length will be large, so if this sequence length is 10,000. We have to compute a 10,000-point FFT, which is. It is easier with FFT than with DFT. Another important thing is that. You can always zero-pad this length. Make it $2^{V\ or\ R}$ whichever. So we get the FFT; FFT means, suppose, in continuous time. Spectra, considering this is of infinite duration, we. Should get a specific peak at F, not possibly minus. If not, let us consider just the positive half. because for a real signal this will be Kind of a symmetric spectrum. Now the question is, will it? Do we get this kind of ideal situation? No, we will experience some spectral leakage, as I said. So there will be a lot of, let me use Some other color, if it is available. So, we will get a lot of these samples. But the good thing is that we learned. the peak detection algorithm. So, we will use peak detection to find and Most importantly, this is not f; this is in k. Because we are doing FFT, we are in k space. We have to find the value of k where there is a peak. I call that K, not. Now, I have found a peak at K. So, I need an estimate of F, not. So, using this formula, we can actually map it. This K is not corresponding to which value. If not, considering that there is only one sinusoid. Now, you can repeat this with two sinusoids with sufficient spacing. You should try experimenting with different values. Of frequencies and how they work. Then add noise, add phase, and check how it works. So, why is this coming? Because k0 is for length. n up to because we are doing n point FFT. Why is n divided by 2? Because you get two peaks, one at k and one at n minus k. And similarly, f0 can be represented at its maximum. Up to $\frac{F_s}{2}$, which is the folding frequency. So this is one example; I have a signal in the. Time domain, and probably this is, I think, 50 hertz. Maybe. So this is a 50 hertz signal, a sinusoidal signal, and if I Do the FFT; this is the frequency plot. This is the magnitude spectrum. So in the magnitude spectrum, I see that this is the dominant one. From this, we can estimate that the dominant frequency is 50 hertz. And the length of this sequence may be very large. is precisely why we have this 50.0, but in practical cases We do not get exactly 50 Hertz; we might not get it. Some estimates are close. Why is that? I will give you an example of what this value of n is. Affecting the spectra or the spectral leakage is evident. from this plot. So, if I take n equals to 64, I See, there is a peak, but there is also all the leakage. And this frequency spreads out in this area. The higher the value of n, the more I increase it. I see that this leakage has been narrowed down. So this peak is kind of precise; I increased it. Furthermore, it is becoming more precise. And 512 points are giving me a very precise frequency. So, for practical purposes, let us try to see an example. So here we have this signal of 50 hertz, and I have a sample. A rate of 1000 and 1000 is much greater than 100. So, the Nyquist criterion is valid here. I am taking 256 samples. Great. So, what should my frequency resolution be? which is f_s/n, which is roughly 3.91 hertz. So, technically, if I see it, it corresponds to a k. Not the value that is 12.8, but k is an integer. So, either the peak will appear at 12 or at 13, and since This is closer to 13, assuming the peak appears at k not equal to 13. So, what we will get is a peak at k, not 13, which is corresponding to it. To a frequency of 50.78 hertz, and that is why. I say this is an estimate of that, not the exact amount. Or in other words, I have a sine wave; I do not know what it is. Where it originates, I just take some of this in. In the discrete time domain, I perform the FFT, and then from the FFT, I Find the peak, and from that peak value, I remap it to the analog. Frequency and find this estimate, which is much closer to this. Now, naturally, this is very good for finding the signal if it has high precision. It is not a big factor, as you can always increase this n and get a better result. Precision, but if you are trying to find sinusoids, it is broken. Rotor bar problem at 50 and 50.5 hertz separately, or 50 and 50.5. 51 hertz separately; this method may not be very good. Because you might need a huge value of n. So, there are these other methods, like music. And esprit will come to the rescue. So, another problem is the power spectral density. Estimation; this is another very good problem. So, what is the power spectral density? What is energy spectral density? So, we have already classified the signals. as power signals and energy signals. So, for power signals, we have power. Spectral density. So, how is the power of the signal? It is for each frequency value, and from this, White noise is also defined where we have a. Flat power spectral density, and similarly for Energy signals have energy spectral densities. So, how do we estimate this? Again, I say estimation because we do not have it. The whole periodic signal, we just get the duration. From that, we will make an estimate. So, this is a four-step process. First you perform FFT of the signal, Then calculate the magnitude spectrum. And then you take the square of the magnitude. Now, this is giving you the power, but this is not a normalized PSD. So, you have to divide by n, n is the You can use the point FFT or the signal length. Say, and this is giving you the estimate of the PSD. For example, if your $x[n]$ signal is white noise, you can estimate. This PSD and then see the distribution of power over the. different frequencies, and if they are almost the same, Then you can say this is kind of white noise. This is one example, so this is kind of a sine wave that is corrupted with noise. So if I do a PSD estimate using FFT, we see That although this is very noisy, you can clearly see a peak. At some value near 50, and probably this is the significance. Of this fundamental period of the wave, buried with. Noise to be a 50 hertz sinusoid. And now this can be improved by methods. Like averaging periodograms and There are also advanced methods. So the averaging periodogram is called the Welch. method that is often used in packages Like in MATLAB and Python. So, that is it. Thank you very much. We will meet again. Have a nice day.