Signal Processing Algorithms & Architecture
Dr. Anirban Dasgupta
Department of Electronics & Electrical Engineering
Indian Institute of Technology Guwahati
Lec 3: Zero-crossing Detection

Hello everyone, welcome back to a new lecture on the topic. Zero crossing detection as part of the course. signal processing algorithms and architecture. I am Dr. Anirban Dasgupta, your instructor, and let us get started. So, time domain signal processing is a summarization of the topics.

So we have already discussed time shifting and scaling. Amplitude shifting and scaling, modulation, correlation, and convolution. So, today we are going to start the topic of zero crossing detection. And in the subsequent lecture, we will be talking about peak detection.

Envelope detection, signal smoothing, and pattern matching. So, what is a zero crossing? Zero crossing is as the name suggests. It is a point where the sign of the signal changes. For example, the signal value can change. a positive to a negative and vice versa.

So this is represented on the time axis has a value of 0, as you can see in the graph. So these points where the signal value touches 0 and switches the sign, so this looks very straightforward. So now that signal was a kind of continuous time signal. Now let us see what happens in a discrete-time signal. So here we have specific values at specific integer indices.

Now, here we see that in this case, the signal value is rarely 0 at a 0 crossing, but we can see that in consecutive. Samples like this sample, this sample, or this sample. And in this sample, the sign of the amplitude changes. So, in simple words, a zero crossing can be either an exact Value of n at the point where the signal value is 0 and the other sides have different signs or zero crossing. can lie between two integer indices like n and n minus 1, where the zero crossing is somewhere in between.

Now, technically, as I said, a signal is a physical phenomenon. A zero crossing may indicate some event in the signal. So we want to capture it in the original time sequence rather than the sampled signal. So, we need to estimate the value of n and then finally map. It is on the analog time axis t, and as I said, it can be specific.

The integer value of n, or it can lie between two integer values. So, let us understand some zero-crossing detection terminology. The first is the zero crossing point. And as I said, this is the point in the signal where. The signal value changes from positive to negative or vice versa.

And depending on the zero-crossing points, how many such points are there? The rate at which events occur in unit time is called the zero crossing rate. And finally, we have something called a threshold. So, threshold considers that this is a small value near 0. Some value, like epsilon, is there to ignore small fluctuations of the signal. Because such fluctuations often occur in real signals.

of noise, and these are often called noise-induced false 0 crossings. So, to reduce that effect, a thresholding technique is used, which we will discuss. So, what is the need for zero-crossing detection? The first is frequency estimation, like if you see a sine wave. And if I tell you what the frequency is, how will you do it? So, you will basically see how many zero crossings there are. or exactly where the zero crossings are occurring and based On that, you can actually find the time period or the fundamental frequency of that signal.

In speech processing, zero-crossing detection is of paramount importance. It is often used to distinguish different segments of the speech signal. Signal classification is another application. Where the zero crossing detection is used to classify various forms of signals. Then edge detection; now this is also a very important application.

Zero crossing detection specifically in image processing. Where edges are found using the concepts of zero crossings. So, considering these applications, what are the real problems with zero crossings? So in the above figure, you see that this is a beautiful, clear sine wave. It is very easy to find the zero crossing; even a school student can do it easily. But if you are given a signal like this which is often the real world case, So real-world signals are not as clean as the above figure.

So, how do we find the zero crossings in such a case? Because here we see that there are many noisy regions. which make that belief that these are zero crossings. But actually, they are just noise-induced zero crossings. and it does not truly signify any event in the signal. So what is the concept? What should the naive approach to detection be? So let us try to define the cases where the zero crossings occur.

Can happen, and based on that, we will try to formulate the algorithms. So, as I said, zero crossings are detected when the signal changes sign. Changes between two consecutive sample points. So, suppose we have a discrete time signal $x[n]$ and say for a specific value of. n, $x[n]$ is 0, and we also check that the previous sample and the remaining samples like the previous sample and the other samples So these two samples are of opposite polarity.

If $x[n]$=0:

$$sign\ (x[n]) \neq sign(x[n+1])$$

Else if $x[n] \neq$0:

$$sign\ (x[n-1]) \neq sign(x[n+1])$$

So, if two successive points are there, for example, as I said. In the last case, so this is my $n-1$, and this is my n then. In this case, this zero crossing will be determined by checking the polarity of successive samples. But if a specific point at n, the signal is 0, then we will Check the previous sample and the next sample.

This is the sine function which has a value of 1 if x is positive and a value of 0 if x is negative.

The signum $sign(x)$ can be defined as:

$$sign(x) = \begin{cases} 1, & x > 0 \\ 0, & x < 0 \end{cases}$$

So if there are p consecutive 0 signal values, like the values are, say, 0s. For, say, some p times, and then you have one negative value. and on the other hand one positive value. So you have to check if consecutive p values have zeros.

Then the previous sample and the next sample are apart from This group of zeros should be of different polarities. And finally, once we get the count of the zero crossings, We can easily calculate the zero-crossing rate. So, this is the total number of zero crossings in the signal. What we will do is divide by NTS. Why NTS? Because these are the crossings in a number of samples.

And if we divide by T_s, we get that per unit second. So, this will give me the number of zero crossings per second. So, let us try to demonstrate each of these cases pictorially. So the first case is that n is between two consecutive. Samples $n-1$ and $n-2$, and here we check the signs are different, and this is one case. So here we see that. At n equals 2, the signal is positive, and at n equals 3, the signal is negative. So the 0 will be somewhere in between these two samples. The second case is exactly where we have a value where my signal value is 0.

So we check for one value before that point that is n equal to 2 and After that point, where n equals 4, see whether these two are of different polarities. If they were of the same polarity, like say this is also positive, then this is just touching 0. It is not crossing 0, and the third case is that I have some p number of values. Like here, p is 4.

So, 4 values are having 0. So, I check the value previous to this. group which is at n equals 2, which is positive, and a value after this group Which is n equals 7, which is negative, and hence we can say there is a zero crossing. So now let us try to understand the detection methods and detection algorithms. So typically there are a lot of algorithms available, but mostly there are very few. Books that actually discuss these concepts.

I have mostly taken this from research papers, the methods include dot product methods. The methods of interpolation, windowing, and hysteresis. So, we will quickly discuss all these methods. First is the dot product method. And again, it is a very naive algorithm, whereas I said in the definition that the two successive samples, and this is mostly the case in practice because.

In practice, you do not get the signal value to be exactly 0. It can be some small positive, and the other end can be a small negative. For this case, we will do the main analysis, but you are free to select the other cases as well. But this is the most common case, and in this we see that if one value is positive. And if one other value is negative, what is the best way? Just multiply them.

So, that is why it is called the dot product method. So just multiply them and check whether they are negative. So if this is negative, that means there is a zero crossing. So this should be just less than 0.

But again, if you consider this. equals 0; it actually takes care of that case where exactly n is at $x[n]$ at equals 0. So this is typically an indication. Whether there is a zero crossing present in the signal or not. And once you have detected the zero crossing, the next task is to. Find where the zero crossing is, and that is also evident from this.

Formula. But wait a minute. So I know that at n and at $n-1$, the product. is negative. So, which means that my zero crossing is typically lying in between these two samples, I need to find the value where it is lying. And more precisely, I want to know where that zero is on the analog time axis.

Crossing is happening. So, what do we do? In this case, we can take the average. So the average of $n-1$ and n will be $n-0.5$ or $n-\ half$. So, this is typically taken as the estimated zero-crossing value. Now, you might be wondering why this is not an integer value and how it is possible.

Now, since we are mapping to analog, ultimately when we multiply with the Sampling time Ts, it will make sense. So, now about the complexity, since this is. an algorithm, So, each comparison let us say it is a constant time operation. So, there will be n minus 1 such comparisons for a signal of length n because. We are comparing successive samples; hence, we can say the complexity of this algorithm is $O(n)$. So, let us take an illustration of this problem. So, we have a small signal small just to show for illustration purposes and We have the values as 3, 4, 5, negative 6, and negative 7, and clearly we know. That this is the region where my zero crossing will lie. So what do we do? We take the index pairs like this, and these are the corresponding signal values. To the index pairs, we find the dot product, which is nothing but the multiplication.

And basically, we see that at this point, or between these two samples, the value is negative. And which means that yes, there is a zero-crossing detected between the samples at 2 and 3. And as I said, in this method, we will take the unbiased estimate as the midpoint, which is 2.5 and now we can map that based on the sampling time. Now you may argue that the The middle point may not be a very good estimate, like considering a zero crossing where. The signal value at $n-1$ is a very small value, which is -2. It is having a very high value, say plus 8. So, you will see that the portion and that should be very much closer to -2 and very much farther away from $+8$.

So, what is this? This is a linear interpolation instead of taking the midpoint. and this is giving me a reasonably better accuracy of the zero-crossing point. So, this is the new algorithm that is zero crossing detection with interpolation. So, in this, the only change is that I am using the same algorithm for detection. Whether zero crossing is there or not, but for marking the zero crossing instead.

Of taking the midway, I will use this linear interpolation formula, which is

$$n_{\text{cross}} = n - 1 - \frac{x[n-1]}{x[n]-x[n-1]}$$

And this you can easily derived by using this diagram, and using this, we will take the same. Example, but here we have the points $x[n-1]$ as 2 and $x[n]$ as 3.

So we will use the interpolation formula, and we find that the value is 2.45 which is different from the previous algorithm 2.5, and this is keeping in mind the fact.

That 5 and negative 6 are different heights. So this is the algorithm, and we will compare the complexity. So, as you know, this Comparison will take $O(n)$ time, like the same dot product method. So, interpolation, of course, you will do the interpolation only when these zero crossings are detected.

But if, say, we have a very noisy zigzag signal. Where every pair of points has a zero crossing, we still get the worst. Case to be $O(n)$. So, overall we can say that the complexity will be $O(n)$ linear time complexity. And this is what I was referring to in the zero-crossing detection with interpolation. So, since this value has a higher height and this value has a lower height.

So, this zero crossing is not exactly midway, but is biased towards the lower height sample. Now, these two algorithms look pretty naive in nature It is easy to understand, but there are some shortcomings, such as noise. induced false zero crossing. As I said that there will be lot of like this signals where the zero crossings may appear due to the formula, but they are typically caused by noise. Another is the interpolation error, as we discussed regarding these two points.

So we will typically do a linear interpolation, but the actual signal may be too. Curvy in nature, a better interpolation method can be quadratic or. A higher-order polynomial, you never know. Like if it were originally a sign, Maybe if the spacing is large, linear interpolation may not give you. The exact precise point, then multiple zero crossings between two points.

Now this is not a problem with the algorithm; rather, it is a problem of understanding. Sampling, and this is violating the Nyquist rate, says that if there are Some oscillatory nature, so we will miss some zero crossings. But this is natural. Because then this is a high-frequency signal, and we have probably sampled. It is at a lower frequency, and that is why we are missing this zero crossing.

So this is an undersampling problem. This is an aliasing problem. So what are the other issues? The boundary issues, like the boundary we have. of a signal, and we cannot apply that at the boundary because we have to see that we have to neglect the boundary pixels because we cannot find an n minus 1 for the first sample and n plus 1 for the last sample. So in that case, Often, the boundary is neglected, and the boundary pixel may be a zero crossing. You never know. So, how do we solve these issues? So, one is noise-induced false zero crossings, which can be removed by smoothing or filtering the signal before applying this algorithm.

Now, we will discuss smoothing techniques in subsequent lectures. But if you are facing this problem, you can smooth the signal. first before applying this zero crossing detection algorithm. Interpolation errors, as I said, can be solved if you are like this. I am very much interested in that kind of fine precision, then probably you are too. You may consider higher-order interpolation, like quadratic or cubic splines.

But if you are okay with a small error in precision, then linear is good enough. Multiple zero crossings between two points, as I said, are a sampling problem. So, you should use a higher sampling frequency at the time of sampling itself. To get a very precise measurement and not miss any zero crossings, specifically in high-frequency signals.

For the boundary issues, one way now is to use mirroring. So, you can extend the signal by mirroring the last few or the first few values. so that the boundary issues are handled carefully. Now, here 0π adding may not be a good option, a way to solve it because in this case, zero padding. will not help you determine whether there was any actual zero crossing or not.

But mirroring might get, of course, mirroring is also a very accurate technique. but this will ensure that your boundary values are handled. Now some signals may have some switching nature. Switching nature means So the signal values are, say, having a trend like this. So this is often called a non-stationary signal where the signal mean is deviating from the baseline is also very common in biomedical signals and it is called baseline wonder. So here instead of doing globally what you can do is perform a windowed zero-crossing detection. So you can take small windows, then you normalize the signal and then apply the zero crossing methods. So the algorithm is you have to divide the signal into either overlapping or non-overlapping windows.

Now what to use will depend on your application and usually, the naive method is to use fixed-size windows. And then for each window, each window behaves like a sub-signal. So in each signal you apply the methods discussed earlier. And then you can combine the results from each window. And this method will give you the zero crossings, so if we talk about the time complexity.

So, dividing the signal into windows will take $O(n)$ time because you are assigning. Each point to a specific window, and considering the windows are non-overlapping. You get $O(n)$ times because ultimately you have n sample points and then you want to aggregate; also, you have to check $O(n)$. So, $O(n)$, $O(n)$, O. $O(n)$ is a complexity of $O(n)$. So, although this method is better for non-stationary signals, but there are issues like the window size.

So what window size do you need? To select in such a way that you get a very nice, accurate result. So one way is to have dynamically changing window size, and that is also a research problem. And this is specific to the signal that you are trying to analyze. The overlapping window is also a concern, like if you have very high.

If the amount of overlap is, say, 90 percent, then the computations will be pretty high. It will not be n of n that overlaps. It might become nearly n^2 because If you are using 90 percent overlapped windows. Less overlap, like 25% or 0% non-overlapping windows. Then you may also miss some portions. So, what percentage? The overlap you should take is again a critical issue, and windowing artifacts.

It is again that boundary conditions are appearing here and here since you. Have a window, so each window will have a boundary problem, and it will require a boundary check, so this is one result with the windowed zero crossing. Now, of course, this signal does not have any trend, but still, you can see that.

We are able to detect most of the zero crossings. Of course, in some regions. We have spurious zero crossings because of some fluctuations in the signal. which is typically solved using the method called the hysteresis method. So, in hysteresis, if you see there is a zero line, this is the time axis, you say t or n anything you say does not really matter. So, you see that now here technically, I would say this is my actual zero crossing or prominent zero crossing.

But these are kind of noise-induced zero crossings. So, how do we solve this? So, we can have two lines, which I call thresholds. One is the upper threshold. which is for the positive cycle and the lower threshold, which is for the negative cycle. So, typically these two values are taken to be equal but there is no hard and fast rule.

You can take your own thresholds; so, in this method, you can actually remove. The noise induced zero crossings by properly selecting the threshold value. And all these algorithms, as I have explained, are solving a specific problem. So, which problem are you having based on that you are free to select your algorithm and you can devise your own algorithm which can be a combination of all these steps. So, the algorithm is: first, I will assign the zero crossing state to 0 that means initially there are no zero crossings. And then we will check whether the signal exceeds the high threshold, which is a mark. Of a positive crossing, and if the signal falls below a lower threshold, then it is a sign of a negative zero crossing, and then we will just check. For conditions like in the dot product case. And again, this is a complexity because we are just checking or comparing.

Each sample against the thresholds, so possibly you are having. Two comparisons for each sample for the high threshold and the low threshold. So this is all about zero crossing detection. Thank you. We will return with new topics.