The Wavelet Transform

An Introduction and Example

Shaw Talebi

Published in Towards Data Science · 6 min read · Dec 20, 2020

This is the final post in a 3-part series on Fourier and Wavelet Transforms. In previous posts both the Fourier Transform (FT) and its practical implementation, the Fast-Fourier Transform (FFT) are discussed. In this post, a similar idea is introduced the Wavelet Transform. Once you have a solid understanding of how the FT works, wrapping your head around the Wavelet Transform is straightforward. I finish this post with a concrete example to show just one of many possible applications.

The Wavelet Transform | Introduction & Example Code

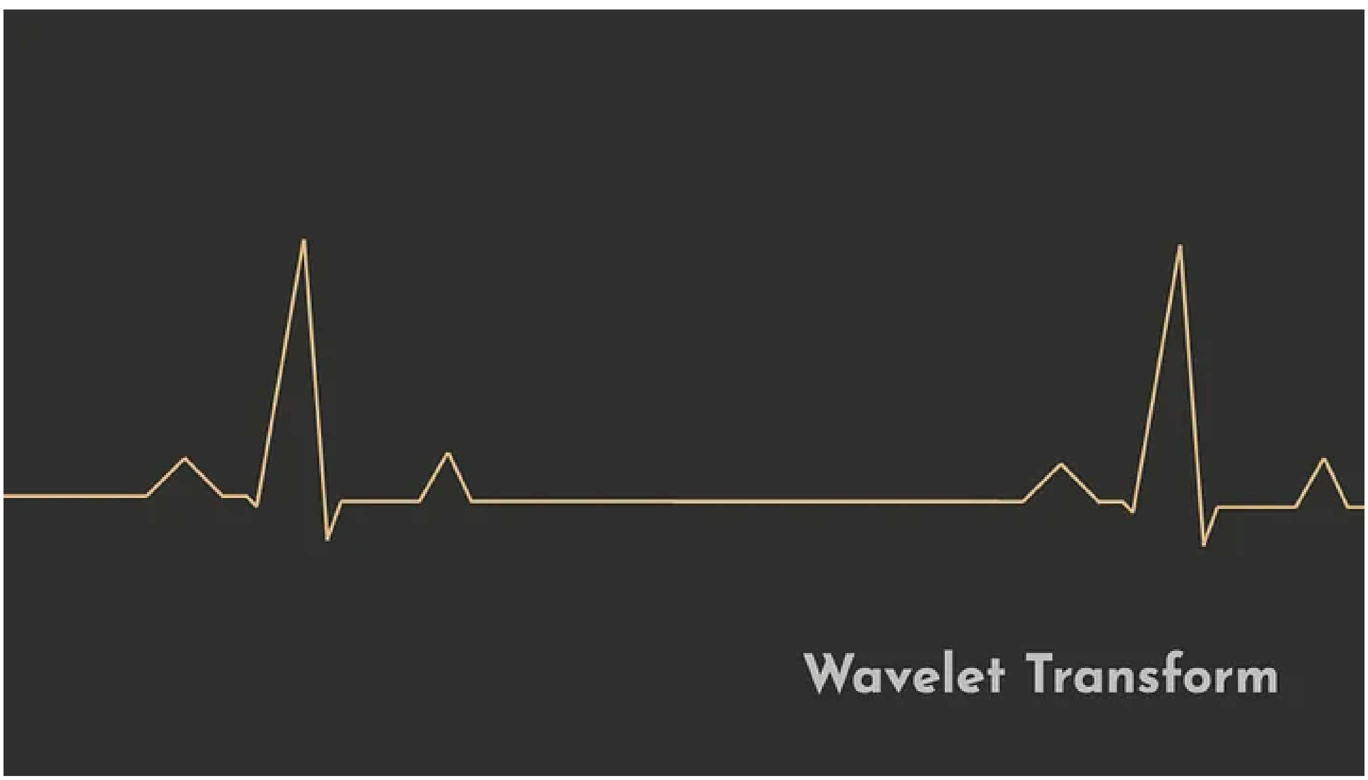
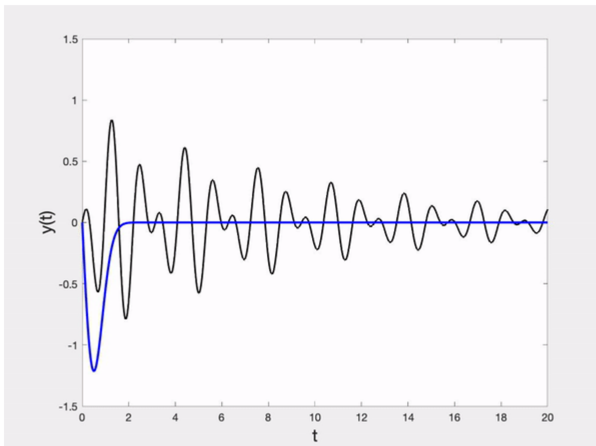


Image by author.

Wavelet Transform

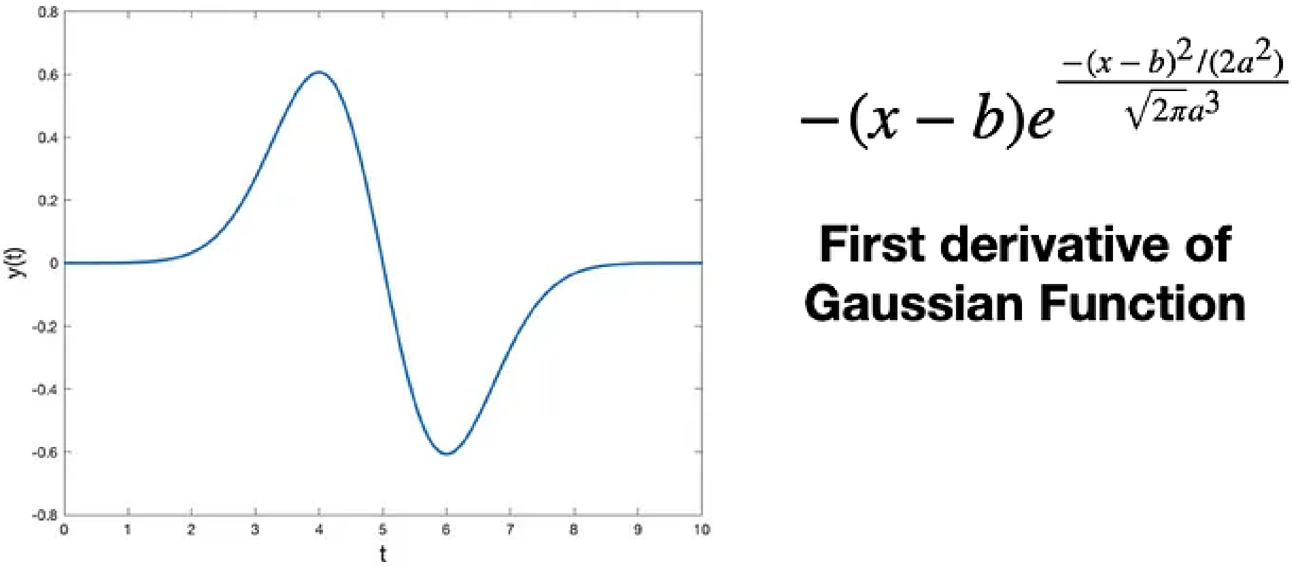
A major disadvantage of the Fourier Transform is it captures *global* frequency information, meaning frequencies that persist over an entire signal. This kind of signal decomposition may not serve all applications well (e.g. Electrocardiography (ECG) where signals have short intervals of characteristic oscillation). An alternative approach is the **Wavelet Transform**, which **decomposes a function into a set of wavelets.**



Animation of Discrete Wavelet Transform. Image by author.

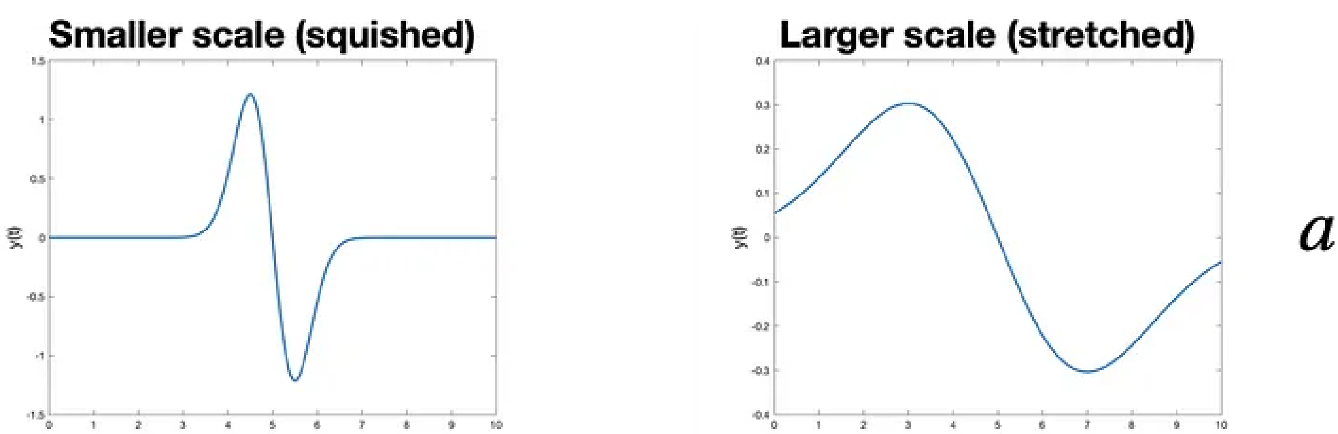
What's a Wavelet?

A **Wavelet** is a **wave-like oscillation that is localized in time**, an example is given below. Wavelets have two basic properties: scale and location. **Scale** (or dilation) defines how "stretched" or "squished" a wavelet is. This property is related to frequency as defined for waves. **Location** defines where the wavelet is positioned in time (or space).



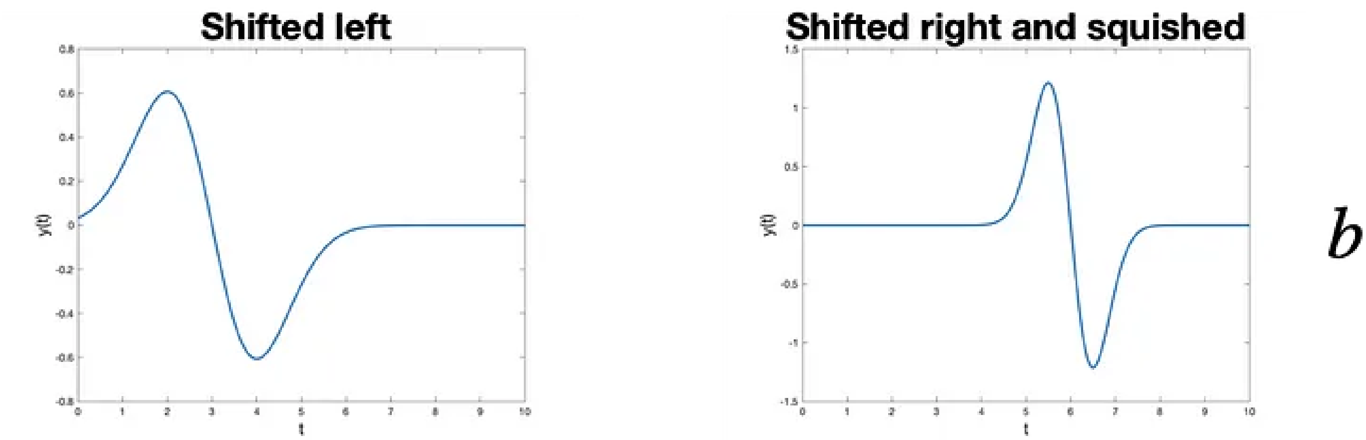
Example Wavelet: The first derivative of Gaussian Function. Image by author.

The parameter “a” in the expression above sets the scale of the wavelet. If we decrease its value the wavelet will look more squished. This in turn can capture high-frequency information. Conversely, increasing the value of “a” will stretch the wavelet and captures low-frequency information.



Left: Example wavelet with decreased scale. **Right**: Example wavelet with increased scale. Image by author.

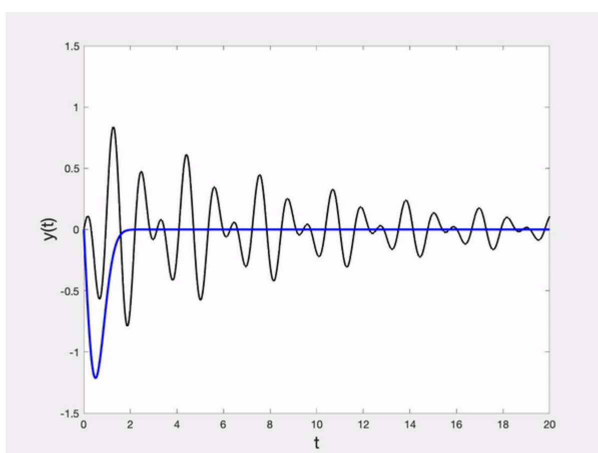
The parameter “b” defines the location of the wavelet. Decreasing “b” will shift the wavelet to the left. Increasing “b” will shift it to the right. Location is important because, unlike waves, wavelets are only non-zero in a short interval. Furthermore, when analyzing a signal we are not only interested in its oscillations, but where those oscillations take place.



Left: Example wavelet with decreased location. **Right**: Example wavelet with increased location and decreased scale. Image by author.

How does it work?

Let’s take another look at the same animation from before.



Animation of Discrete Wavelet Transform (again). Image by author.

The **basic idea** is to **compute** ***how much*** **of a wavelet is** ***in*** **a signal** for a particular scale and location. For those familiar with convolutions, that is exactly what this is. A signal is convolved with a set wavelets at a variety of scales.

In other words, we pick a wavelet of a particular scale (like the blue wavelet in the gif above). Then, we slide this wavelet across the entire signal i.e. vary its location, where at each time step we multiply the wavelet and signal. The product of this multiplication gives us a coefficient for that wavelet scale at that time step. We then increase the wavelet scale (e.g. the red and green wavelets) and repeat the process.

Continuous Wavelet Transform (CWT)

$$T(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t)\, \psi^{*} \frac{(t-b)}{a} dt$$

Discrete Wavelet Transform (DWT)

$$T_{m,n} = \int_{-\infty}^{\infty} x(t)\, \psi_{m,n}(t)\, dt$$

Definitions of Continuous and Discrete Wavelet Transforms. Image by author.

There are two types of Wavelet Transforms: Continuous and Discrete. Definitions of each type are given in the above figure. The key difference between these two types is the Continuous Wavelet Transform (CWT) uses every possible wavelet over a range of scales and locations i.e. an infinite number of scales and locations. While the Discrete Wavelet Transform (DWT) uses a finite set of wavelets i.e. defined at a particular set of scales and locations.

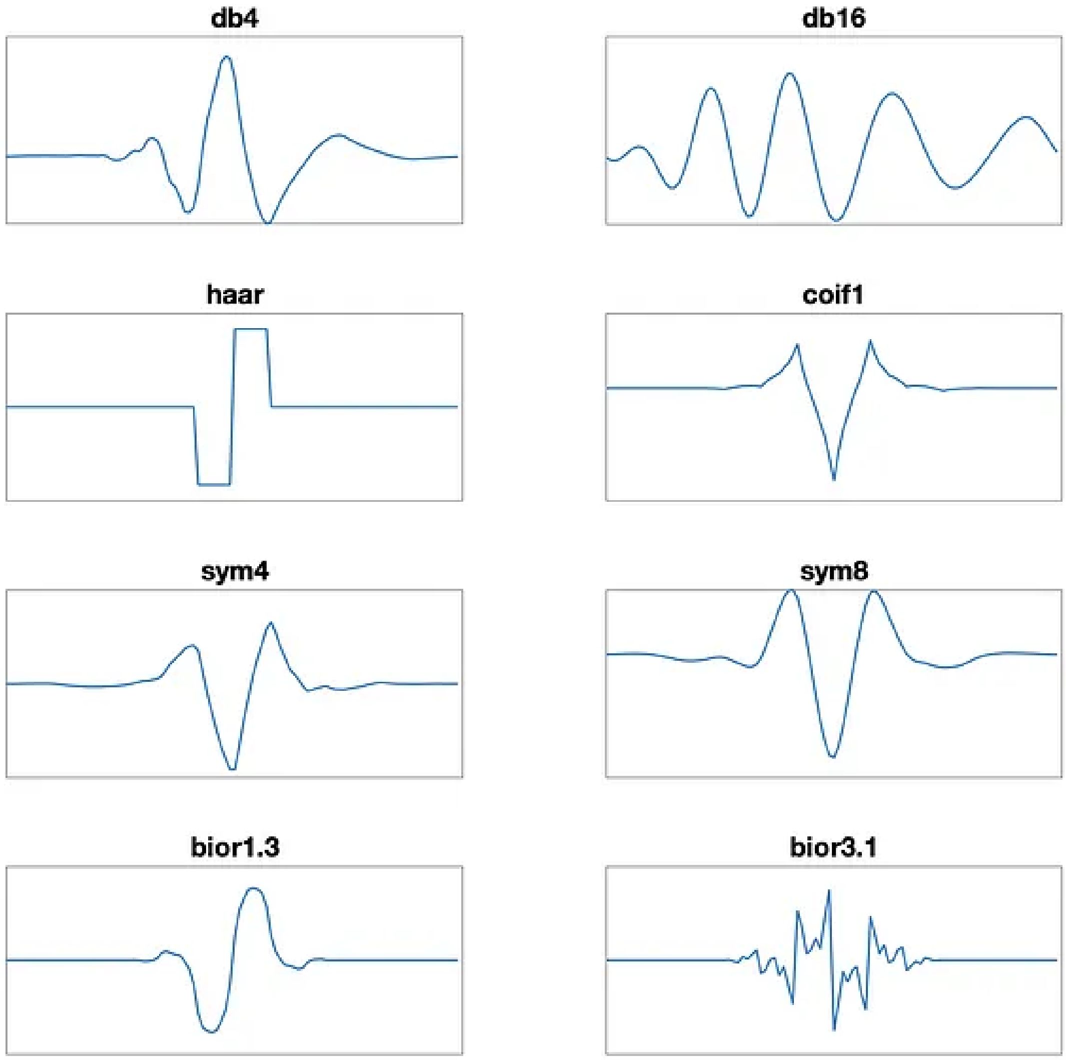
Why wavelets?

A couple of key advantages of the Wavelet Transform are:

- **Wavelet transform** can extract local spectral **and** temporal information simultaneously
- **Variety of wavelets** to choose from

We have touched on the first key advantage a couple of times already. This is probably the biggest reason to use the Wavelet Transform. This may be preferable to using something like a Short-Time Fourier Transform which requires chopping up a signal into segments and performing a Fourier Transform over each segment.

The second key advantage sounds more like a technical detail. Ultimately, the takeaway here is if you know what characteristic shape you are trying to extract from your signal, there are a wide variety of wavelets to choose from to best *match* that shape. A handful of options are given in the figure below.



Some wavelet families. From top to bottom, left to right: Daubechies 4, Daubechies 16, Haar, Coiflet 1, Symlet 4, Symlet 8, Biorthogonal 1.3, & Biorthogonal 3.1. Image by author.

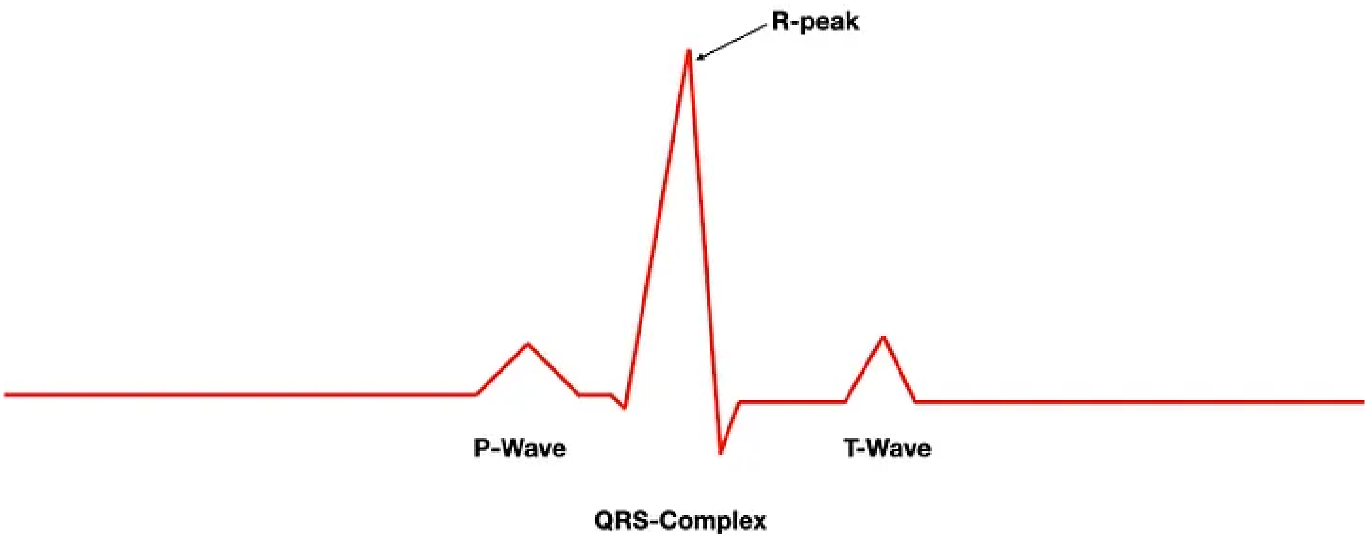
Time Series, Signals, & the Fourier Transform

An introduction and primer for future posts

towardsdatascience.com

Example: Detecting R-peaks in ECG Signal

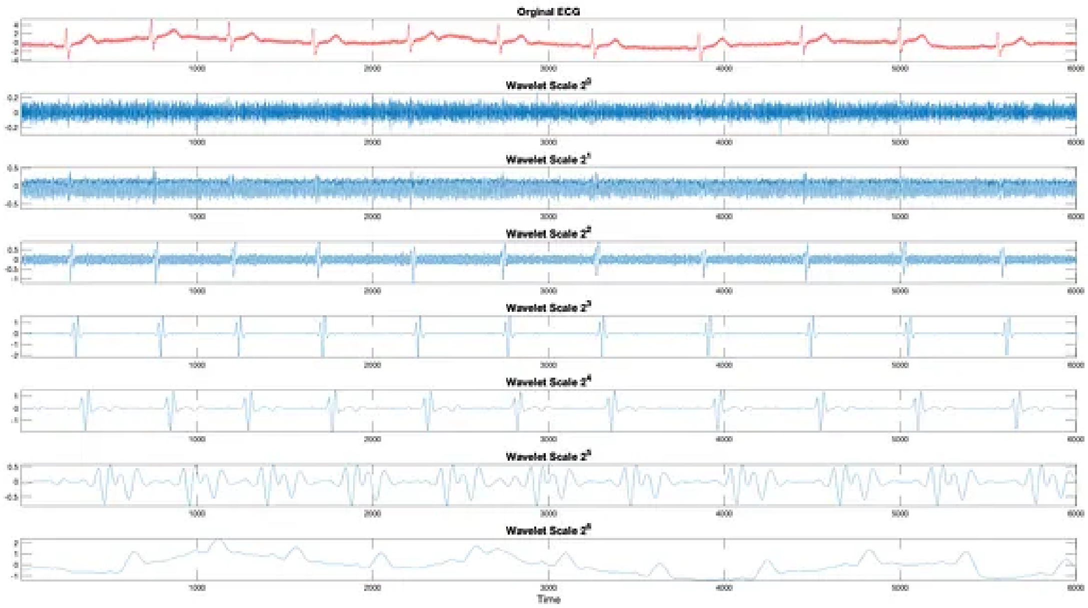
In this example, I use a type of discrete wavelet transform to help detect R-peaks from an Electrocardiogram (ECG) which measures heart activity. R-peaks are typically the highest peak in an ECG signal. They are part of the QRS-complex which is a characteristic oscillation that corresponds to the contraction of the ventricles and expansion of the atria. Detecting R-peaks is helpful in computing heart rate and heart rate variability (HRV). Example code can be found in the GitHub repo.



Sketch of a typical ECG signal resulting from heartbeat. Image by author.

In the real world, we rarely have ECG signals that look as clean as the above graphic. As seen in this example, ECG data is typically noisy. For R-peak detection, simple peak-finding algorithms will fail to generalize when applied to raw data. The wavelet transform can help convert the signal into a form that makes it much easier for our peak finder function.

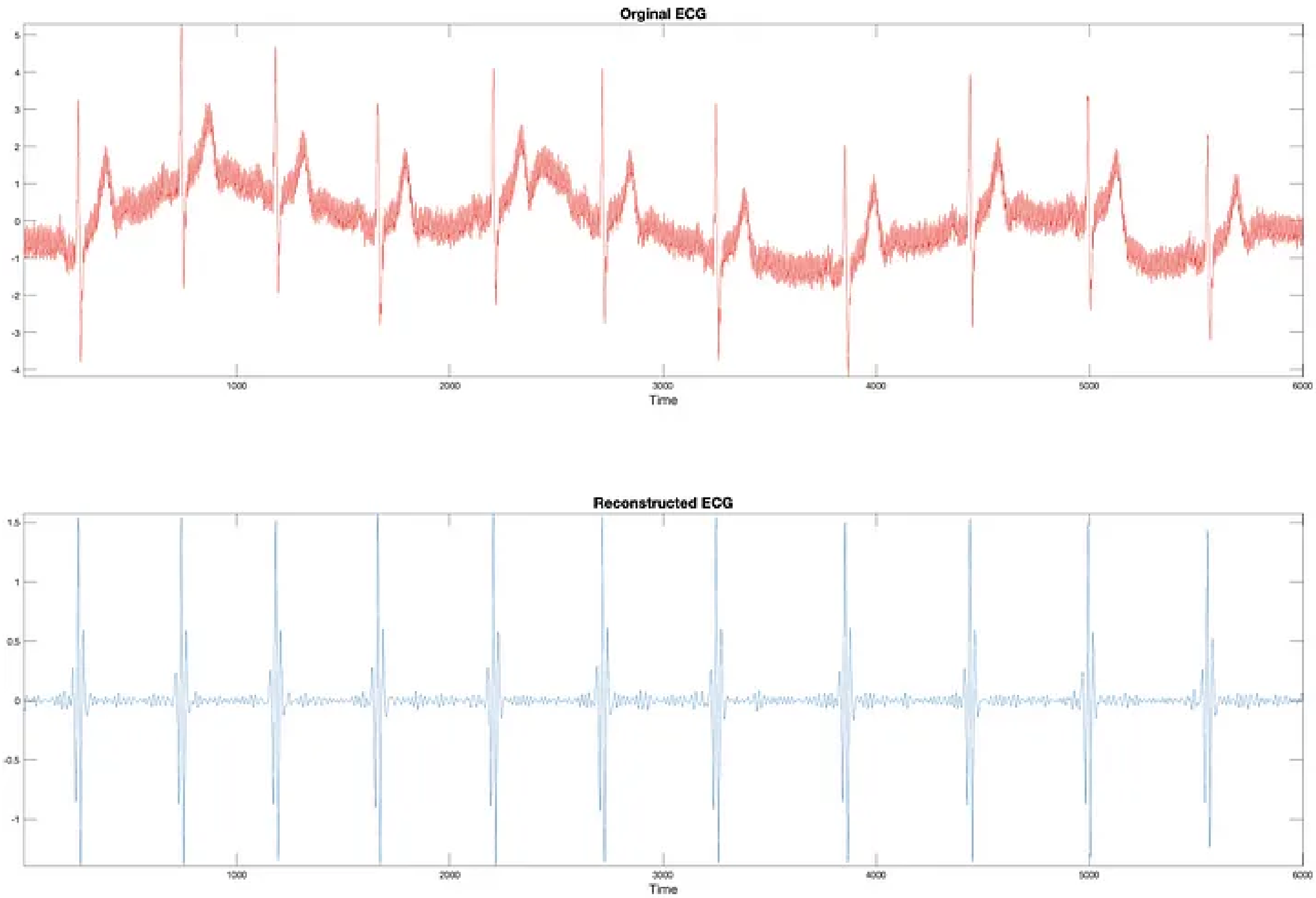
Here I use the maximal overlap discrete wavelet transform (MODWT) to extract R-peaks from the ECG waveform. The Symlet wavelet with 4 vanishing moments (sym4) at 7 different scales are used. Below the original ECG signal is plotted along with wavelet coefficients for each scale over time.



ECG signal and corresponding wavelet coefficients for 7 different scales over time. Image by author.

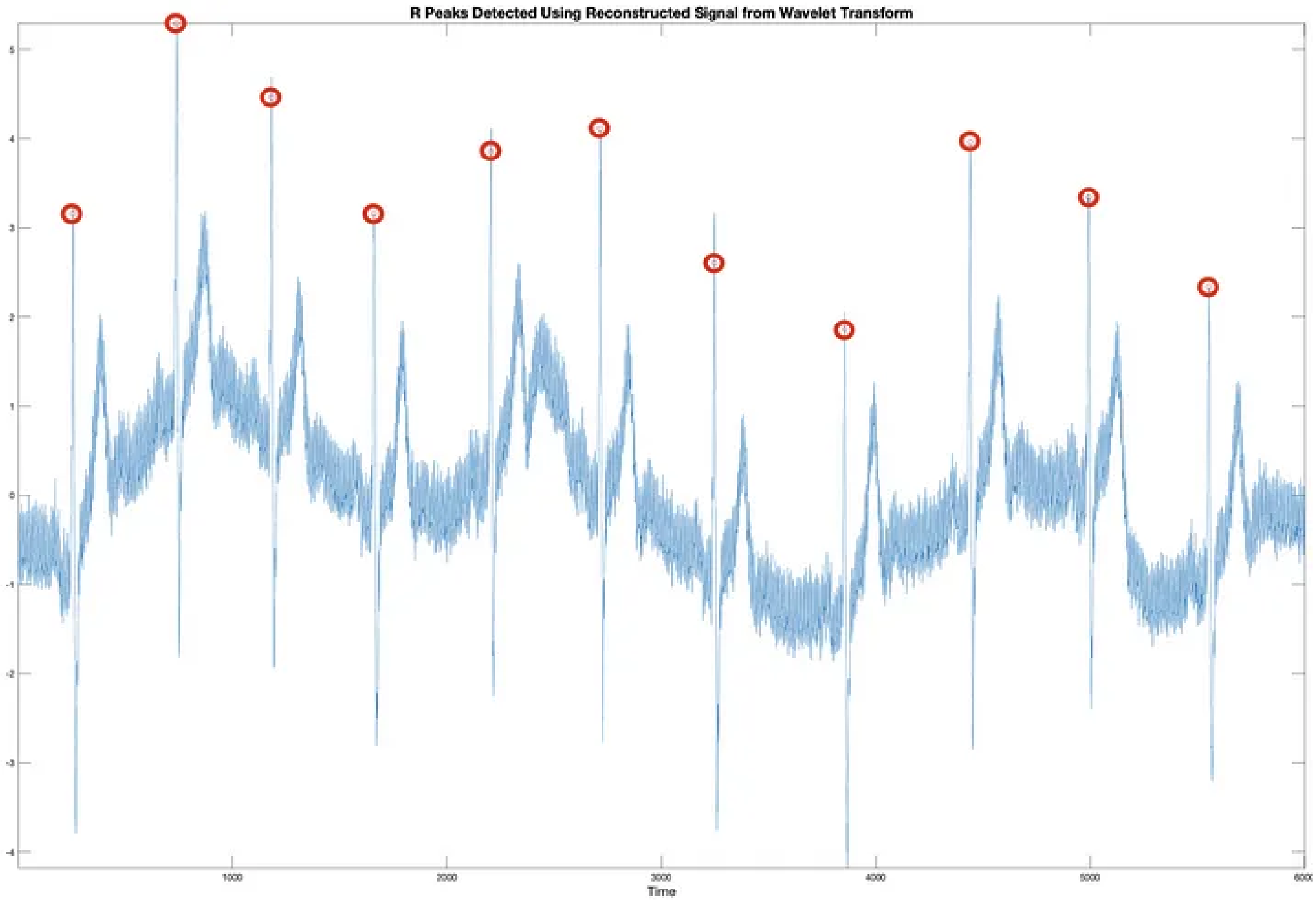
The smaller scales such as 2^0 and 2^1 correspond to high frequencies thus predominantly consist of noise in this example. As we go up in scale, we see blips emerge from the noise that corresponds to R-peaks, i.e. in 2^2, 2^3, and 2^4. We then lose the signal in the larger scale coefficients i.e. 2^5 and 2^6, which are associated with low-frequency information.

We can then reconstruct the original signal with information from a subset of our wavelet scales. Here I only keep information from one scale, 2^3. Below the original and reconstructed signals are plotted. We see the peaks in the reconstructed ECG (lower plot) line up reasonably well with the R-peaks. Additionally, applying a peak finder to the reconstructed ECG seems much more promising than to the original ECG.



Top: Original ECG signal **Bottom**: Reconstructed ECG from wavelet transform. Image by author.

The final step is to apply a find peaks function to the reconstructed signal. This will approximately give the timestamps of each R-peak. To evaluate the performance we plot the detected R-peaks on top of the original signal.



Detected R-peaks plotted on top of original ECG signal. Image by author.

Smoothing Financial Time Series with Wavelets

A real-world use case and example in Python

shawhin.medium.com

Conclusion

In this post, the Wavelet Transform was discussed. The key advantage of the Wavelet Transform compared to the Fourier Transform is the ability to extract both local spectral and temporal information. A practical application of the Wavelet Transform is analyzing ECG signals which contain periodic transient signals of interest.

This post concludes a 3-part series on Fourier and Wavelet Transforms. Videos, code, and additional blog posts can be found in the Resources section below.

👉 **More in this series:** Fourier Transform | FFT

Resources

Connect: My website | Book a call | Ask me anything

Socials: YouTube 🎥 | LinkedIn | Twitter

Support: Buy me a coffee ☕️

The Data Entrepreneurs

A community for entrepreneurs in the data space. 👉 Join the Discord!

medium.com

[1] Addison, P.S. (2005). Wavelet transforms and the ecg: A review. Physiological Measurement, 26 (5), p. R155

[2] https://www.mathworks.com/help/wavelet/ug/r-wave-detection-in-the-ecg.html