Financial Signal Processing

Part 1

Pablo Marchesi

Abstract

Digital signal processing (DSP, for short) is a branch of electronic and telecommunications engineering that focuses on analyzing, interpreting, and manipulating signals to extract useful information. This is done using digital electronic devices. DSP is usually applied in fields such as communications, medicine, and acoustics. It is also commonly used to process temporal signals that are often random. The techniques used in this field can also be applied to financial markets due to the similarities in signal type (as financial signals are also random and vary over time).

This paper aims to apply the most common DSP techniques (signal filtering, Fourier Transforms, convolutions, etc.) to financial markets and determine if they provide us with helpful information about the signal. The paper is presented in a way that would be understood by people with a non-STEM background (e.g., readers with a finance or business degree). However, some technical knowledge about Signals and Systems would facilitate the comprehension of the text.

Contents

Chapter 1

Sampling Financial Time Series

1.1 Continuous-Time vs Discrete-Time Signals

In this chapter, we will explain the implications of working with discrete-time and continuous-time signals. As DSP deals with discrete signals, sampling will be a critical part.

While financial time series are not strictly continuous, they are often treated as such for analytical purposes. This is because they are typically recorded at fixed intervals (e.g., daily, hourly, or every minute) and can be interpolated or extrapolated to fill in any gaps in the data. In general, for most practical purposes, financial data collected at millisecond or even microsecond intervals is sufficient to capture the relevant information and trends in the market. Thus, we can define the signal as continuous-time if the interval is short enough.

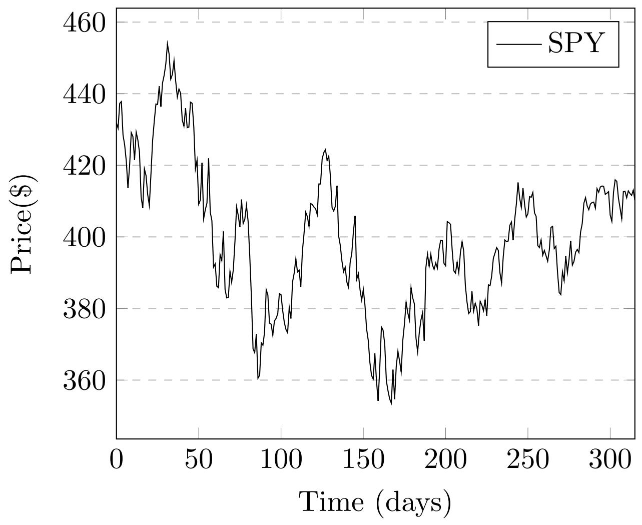

Figure 1.1: A continuous-time financial signal

However, as it is easier to obtain data in larger intervals (e.g., daily), from now on, we will define financial time series as a sampled continuous signal (in daily intervals), consequently, a discrete-time signal.

For simplicity, we will use the closing price as our primary data source. This implies sampling the daily time series once per day. Thus, our sampling frequency will be one sample/day. If we consider the Nyquist–Shannon sampling theorem, we have that the maximum bandwidth of our signal will be 0.5 samples/day (or a period of two days).

Theorem 1.1.1 (Nyquist–Shannon sampling theorem) *To accurately reconstruct a continuous-time signal, the sampling frequency must be greater than twice the maximum frequency present in the signal.*

$$fs \geq 2B_{max} \tag{1.1}$$

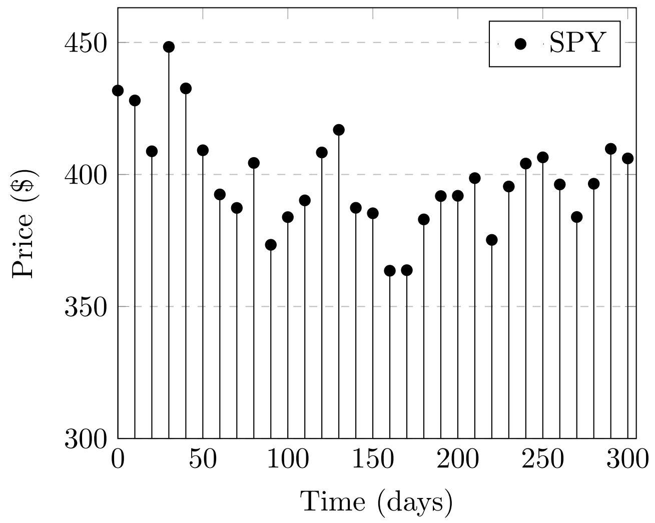

Figure 1.2: A discrete-time financial signal

The main purpose of applying DSP techniques to financial data is to extract useful information from its frequency domain, as it contains extra information about the signal that is not present in the time domain. Sampling will modify the frequency domain representation of the signal, making the spectrum periodic, with a frequency domain ranging from 0 Hz to 0.5 Hz.

1.2 Frecuency Domain Representation

Once the sampling is done, we can proceed to work with the signal in its frequency domain. To accomplish this, we must transform the signal using the Discrete-Time Fourier Transform (DTFT, for short).

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n] \cdot e^{-j\omega n} \tag{1.2}$$

The DTFT of our signal will be an even periodic complex function. For simplicity, from now on, we will only represent one period of the signal and the positive frequency components. In addition to this, as we will be focusing more on the module of the DTFT, we will represent this part of the signal, ignoring the phase information.

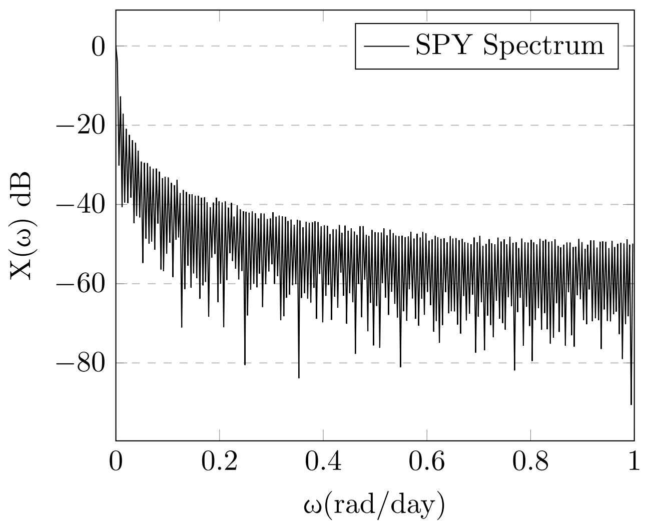

Figure 1.3: DTFT of a financial signal

One of the advantages of analyzing the signal in the frequency domain is that we can easily eliminate some parts of the spectrum. Thus, we can modify the frequency components of our time series. This is usually called filtering, and it can be used, for example, to remove noise from the signal by eliminating the high-frequency components.

1.3 Aliasing

1.4 Market Noise and Aliasing Noise

Chapter 2

Filtering

2.1 Simple Moving Average (SMA)

This chapter will review some of the most common digital filters that can be applied to financial signals. In addition, we will introduce some concepts that will be key to developing our trading applications, such as the cutoff frequency or the step response.

A filter that selects the low frequencies of the signal (and discards the high frequencies) is called a low-pass filter, commonly used for smoothing purposes. One of the most popular low-pass filters is the simple moving average (SMA, for short), and it is widely used to analyze financial signals.

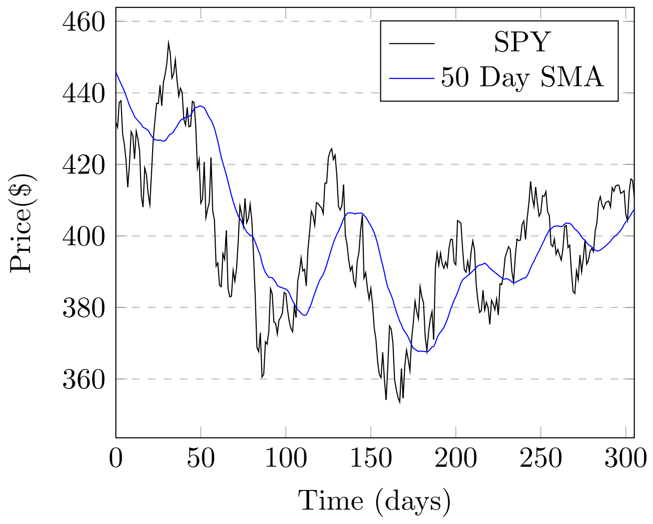

Figure 2.1: The 50 day SMA smoothes the signal

Note that although we have represented both lines continuously, they are discrete-time signals. From now on, we will display financial signals this way.

This section will define the SMA filter in both the time and frequency domain. In the time domain, filtering is done by the convolution operation between the signal and the filter. The discrete convolution of two sequences $x[n]$ and $h[n]$ is denoted as $y[n]$, and it is defined as:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] \tag{2.1}$$

The filter is characterized by its impulse response, which we will call $h[n]$. Therefore our signal will be $x[n]$. The convolution of both functions gives the filtered signal, $y[n]$. The impulse response of an SMA filter of length N is given by:

$$h[n] = \frac{1}{N} \sum_{k=0}^{N-1} \delta[n-k] \tag{2.2}$$

The SMA is a sequence of pulses starting from $n = 0$ with an amplitude of $1/N$. Consequently, the SMA is a non-recursive causal filter, as its inputs are only past values. It is, in fact, an average of the N past values. We can also conclude that the simple moving average is a finite impulse response filter or FIR filter (as the $h[n]$ domain is limited to a couple of values). Filtering with a simple moving average yields a smoothed version of the signal, and the smoothing degree is proportional to the length N of the SMA.

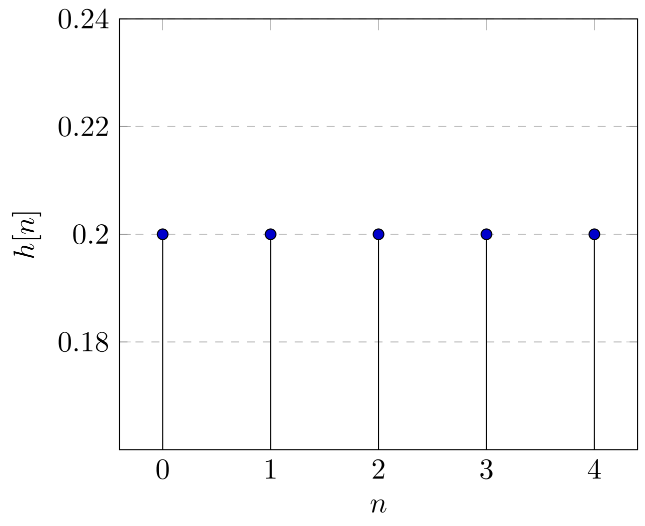

Figure 2.2: Impulse response of a 5 day SMA

Another way to obtain extra information about the filter is to work with its frequency, domain representation. Thus, our $h[n]$ (impulse response) will become $H(w)$ (frequency response) after taking the DTFT. Working in this new domain will enable us to determine which frequencies are attenuated or eliminated by the filter. The frequency response of a moving average filter is defined as:

$$|H(\omega)| = \frac{1}{N}\left(\frac{\sin(\omega N/2)}{\sin(\omega/2)}\right) \tag{2.3}$$

In this case, high frequencies will be rejected because the SMA is a low-pass filter.

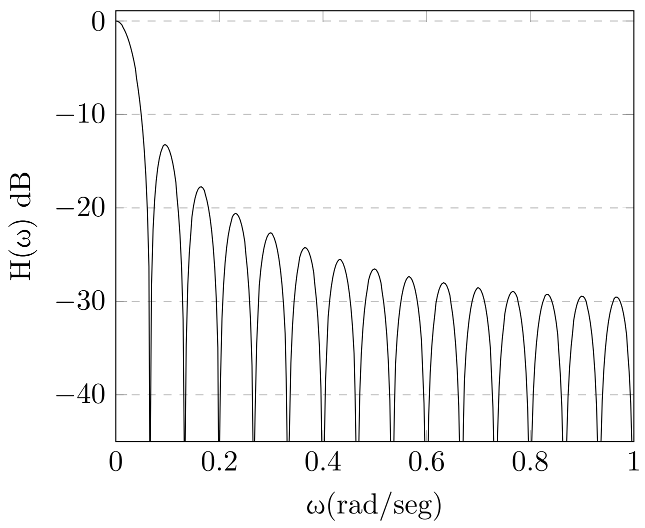

Figure 2.3: Frequency Response of a 50 day SMA

Working with the frequency response instead of the impulse response has many advantages. One of them is that convolutions in the time domain become multiplications in the frequency domain. Therefore filtering in this last domain is a much simpler and more intuitive operation. It is also computationally efficient to filter signals in the frequency domain due to the fact that the FFT, or Fast Fourier Transform, allows us to perform such calculations with a reduced number of operations (compared to the convolution). The filtered signal in the frequency domain is defined as:

$$Y(\omega) = X(\omega)H(\omega) \tag{2.4}$$

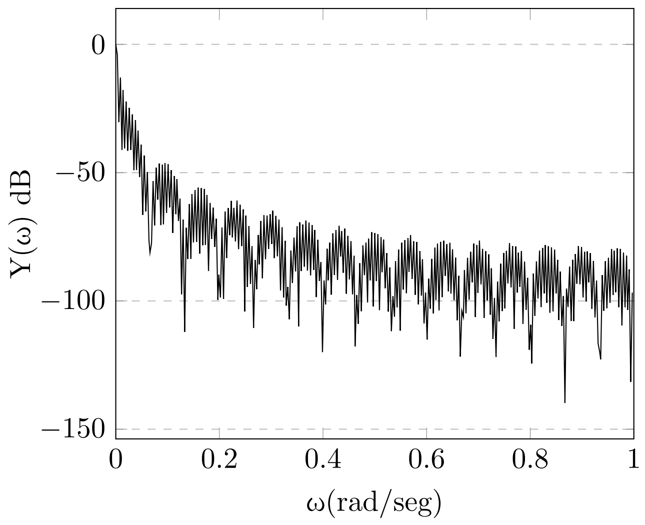

Figure 2.4: Frequency Response filtered signal (with a 50 day SMA)

2.2 Exponential Moving Average (EMA)

The exponential moving average, or EMA, is an extension of the simple moving average that places more weight on recent data points and less weight on older data points, making it more responsive to changes in the input signal. The difference equation of an exponential moving average filter is straightforward:

$$y[n] = \alpha x[n] + (1 - \alpha)y[n - 1] \tag{2.5}$$

The filter is characterized by the smoothing factor α (alpha). It can also be understood as the weight given to the current input value, and it lies between 0 and 1. A typical value for α is often around 0.1 to 0.3. Due to this ponderation of recent data points, the EMA is able to adapt quicker to market fluctuations than the SMA.

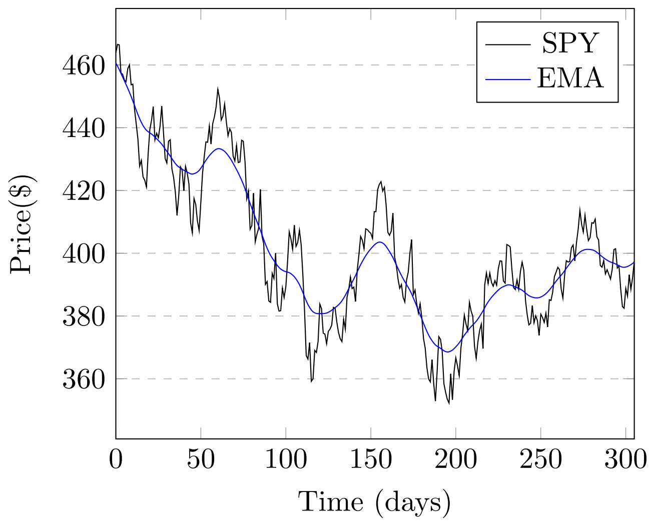

Figure 2.5: The EMA adapts quicker to market fluctuations

From equation 2.5, we can observe that the EMA is a recursive filter, as the current value is computed using previous filter calculations. Its impulse response has the following equation:

$$h[n] = \alpha(1-\alpha)^n \tag{2.6}$$

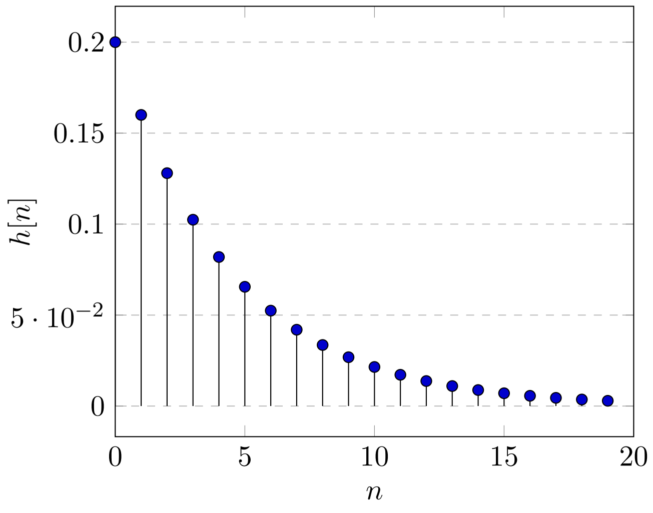

Figure 2.6: EMA impulse response ($\alpha = 0.1$)

It is clear that the EMA has an infinite impulse response, thus, it is a FIR filter. With regard to the frequency domain representation, the EMA is also a low-pass filter, as it smoothes the signal by removing the high-frequency components. It has the following equation:

$$H(\omega) = \frac{\alpha}{1-(1-\alpha)\cdot e^{-i\omega}} \tag{2.7}$$

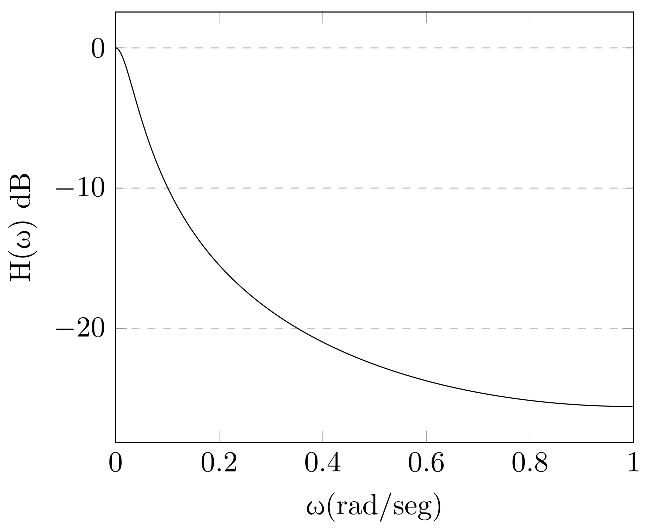

Figure 2.7: Frequency Response EMA ($\alpha = 0.1$)

Lastly, here is the spectrum of the filtered signal:

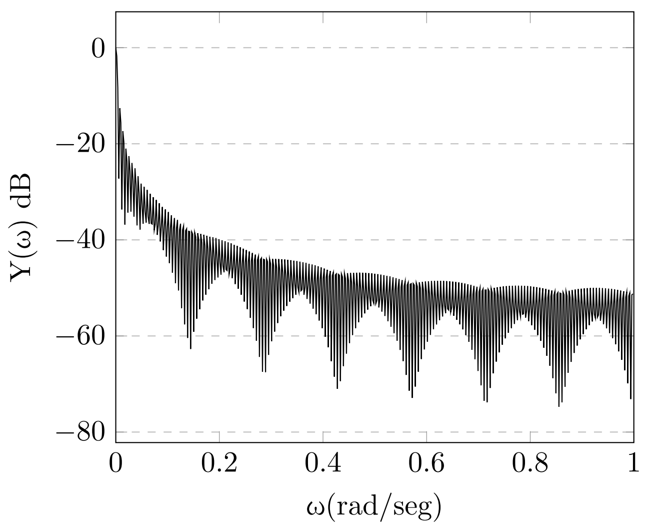

Figure 2.8: Frequency Response filtered signal ($\alpha = 0.1$)

2.3 Cutoff Frequency

One way to characterize a filter is by its cutoff frequency, in other words, the frequency at which the filter's power or amplitude response is reduced to half (-3dB) of its maximum value. It marks the point where the filter starts to attenuate the signal significantly. It's often called the -3dB point, because:

$$\log_{10}(\frac{1}{2}) \approx -3.01dB \tag{2.8}$$

The cutoff frequency is a fundamental parameter in the design and analysis of filters, such as low-pass, high-pass, band-pass, and band-stop filters. It determines the range of frequencies that the filter allows to pass through (passband) or attenuates (stopband).

In the following chapters, we will compare the performance of different kinds of filters, and we will be taking the cutoff frequency as our point of reference. In addition, we will use it as our main optimization parameter in order to improve our trading systems.

2.4 Step Response

Another way to characterize a filter is by looking at its step response. We have previously discussed the impulse response $h[n]$, which gives us crucial information about how the filter behaves. However, if we analyze the system's response to a unit step input, we can compute the step response $s[n]$ and get extra information about the filter. This approach enables us to determine new parameters like rise time (the time it takes for the output to reach a certain percentage of the final value) or the settling time (the time taken for the output to settle within a specific tolerance of the final value).

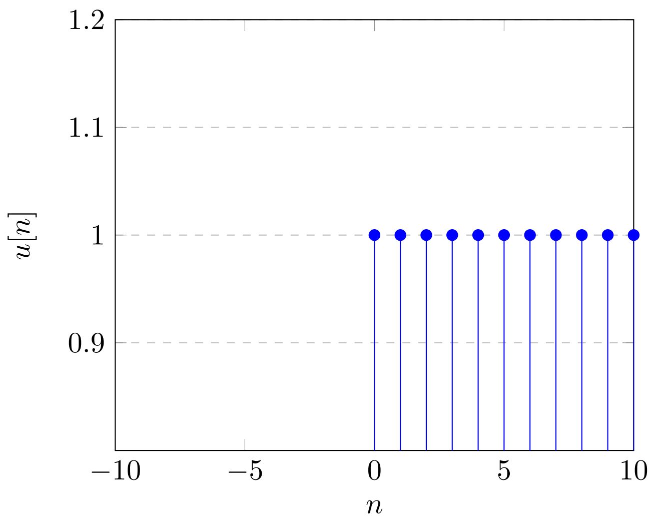

Figure 2.9: Unit Step Function (Discrete-Time)

2.5 Butterworth Filter

2.6 Elliptic filter

2.7 Chebyshev filter

Chapter 3

Filter Implementation

3.1 Difference Equation

3.2 Transfer Function

3.3 Python Implementation

3.4 Trading Indicators and Strategies

3.5 Design of a Trading System

Chapter 4

Backtesting, Optimization and Statistical Analysis

4.1 Backtesting Process

4.2 Results and Key Metrics

4.3 Bayesian Optimization

4.4 Hypothesis Testing

4.5 Final Results

Chapter 5

Conclusion

Appendix A

Dashboard

Appendix B

Bibliography