Kalman Filtering for Pairs Trading

Kimberly Kho

Delft University of Technology

TU Delft

Kalman Filtering for Pairs Trading

by

Kimberly Kho

Student number: 5071887
Instructor: Dr. R.C Kraaij
Faculty: Faculty of Electrical Engineering, Mathematics and Computer Science

TUDelft

Lay summary

Consider two companies operating in the same industry, such as Coca-Cola and Pepsi, whose stock prices usually exhibit the same movements. If Pepsi's price suddenly drops while Coca-Cola's stays the same, a financial trader might expect the gap to close again. By buying the cheaper stock and selling the more expensive one, the trader can profit once prices revert to their original pattern.

The difficulty is that markets are noisy: prices fluctuate constantly, making it hard to tell real trading opportunities from random signals. To address this, this thesis applies the Kalman filter, a mathematical tool that smooths out data and filters away noise. In theory, this should generate clearer and more accurate trading signals.

The method was tested on two related cryptocurrencies, Ethereum (ETH) and NEO, during 2018–2019. Three trading approaches were compared: trading on the raw price difference, on Kalman-filtered data with default settings, and on Kalman-filtered data with optimized parameters.

The results showed that trading on raw data outperformed the default filter, highlighting the need for parameter optimization. Once implemented, the filter with optimized parameters achieved the strongest overall performance.

Interestingly, the price gap between Ethereum and NEO did not consistently return to historical averages, but behaved unpredictably. Still, the study demonstrates that filtering techniques combined with optimization can improve trading performance and uncover hidden opportunities.

Summary

This thesis explores the application of Kalman filtering techniques to enhance pairs trading strategies in financial markets. Pairs trading is a statistical arbitrage strategy that exploits temporary price divergences between historically correlated assets by taking opposite positions with the expectation of mean reversion. The study addresses a fundamental challenge in pairs trading: accurately modeling the underlying spread dynamics in the presence of market noise.

The research implements a state-space model framework where the observed spread between asset prices is treated as a noisy measurement of a mean-reverting process. A default Kalman filter is applied to estimate the true underlying spread by filtering out market noise, with the goal of generating more reliable trading signals. To optimize the Kalman filter's performance, the Expectation-Maximization (EM) algorithm is employed to estimate the model's latent parameters, including process noise and observation noise covariances.

The methodology is tested on a cryptocurrency pair (Ethereum-NEO) identified from existing literature using the distance method for pair selection, covering the period from January 2018 to December 2019. To test the performance of Kalman filtering three approaches are compared: trading on unfiltered spreads, trading on Kalman-filtered spreads with default parameters, and trading on spreads filtered using EM-optimized parameters.

The empirical results reveal several key findings. Surprisingly, the unfiltered spread strategy initially outperformed the default Kalman filter approach, generating $725.73 in profits across 4 trades compared to $523.95 across 3 trades for the filtered approach. However, when EM optimization was applied, the Kalman filter strategy achieved the highest performance with $750.83 in profits across 4 trades.

A notable discovery is that the estimated state transition coefficient consistently converged to 1, indicating random walk behavior rather than the expected mean-reverting dynamics. This suggests that the theoretical assumption of mean reversion may not always align with empirical data, highlighting the importance of model validation in quantitative finance applications.

The study demonstrates that while Kalman filtering can enhance pairs trading strategies, parameter optimization through EM is crucial for achieving superior performance. The research contributes to the understanding of noise reduction techniques in financial time series and provides insights into the practical challenges of implementing statistical arbitrage strategies. Future work could explore larger asset universes, explicit mean-reversion constraints, and the incorporation of transaction costs and risk management considerations.

Contents

1
Introduction

Pairs trading is a well-known trading strategy used in financial markets. It is a form of statistical arbitrage, which entails that deviations in the price relationship between two historically correlated financial instruments, such as stocks, bonds, or derivatives, are exploited with the aim of generating returns.

Executing a pairs trading strategy involves two key steps: first, selecting a pair of financial instruments that demonstrate a historical relationship, and second, detecting moments when their prices diverge from this equilibrium [12]. When such moments occur, the trading strategy aims to profit by taking opposite positions while expecting that the relationship will eventually revert to its mean. These steps will be explained shortly in this introduction by the means of an example.

The first step in pairs trading, identifying a correlated pair, is crucial as it heavily determines the strategy's success. It involves selecting pairs whose prices move together historically and predictably [13]. A well-known, simple technique that is used in literature and industry is the distance method. This method assumes that two assets whose normalized price series stay close together are good candidates for pairs trading. A simplified example of such pair could be Pepsi (PEP) and Coca-Cola (KO), two distinct companies that produce similar products within the soft drink industry. Historically, their stock prices have exhibited similar patterns, rising and falling in response to broader market trends affecting the soda market. In Figure 1.1 we can see that, in general, the prices of PEP and KO move similarly.

The second step is to identify discrepancies in the prices of the pair chosen in the first step. Referring back to the Pepsi and Coca-Cola example, in Figure 1.1 it can be observed that Pepsi's asset price experiences a sudden drop in May. A potential reason for this could be that Pepsi experiences internal issues, leading to financial losses which does not have a direct effect on the other company, Coca Cola. When such a moment occurs, the difference between the prices of the assets can exceed a historical mean. This triggers trading signals for the pairs trader, as he anticipates that the prices of the securities will revert to their typical relationship. These trading signals can be formalized using statistical methods on historical data.

Figure 1.1: Price relationship between Coca-Cola (KO) and Pepsi (PEP).

After performing these two steps, the trader will enter opposite positions in the assets; short (sell) the relatively overpriced stock, expecting its value to fall back to its mean, and go long (buy) on the under priced stock, expecting its value to rise, see Figure 1.2. When the prices of the securities revert back to their historical pattern, the traders can make a profit by the correction of the deviation in the prices by exiting his positions.

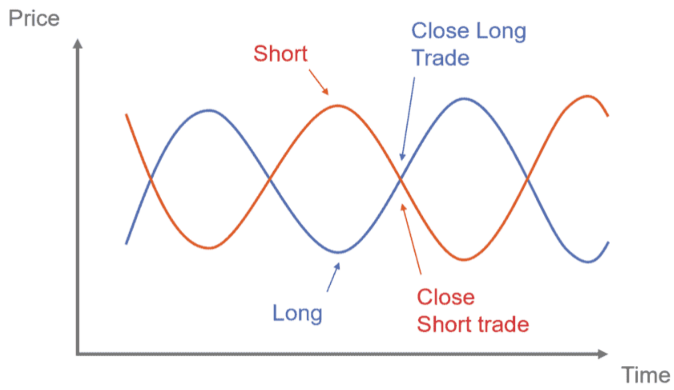


Figure 1.2: General idea of pairs trading.

A key advantage of this strategy is that it reduces exposure to overall market movements by focusing on relative price changes between assets. The trader aims to keep their portfolio market neutral, which entails that the portfolio will profit from both rising and falling prices in a market. In other words, the trader can generate returns regardless of whether the overall market is going up or down making pairs trading an attractive trading strategy.

A downside of the strategy is that it heavily relies on the assumption that the relationship between the asset prices is stable, following a stationary process. In practice, however, financial time series are often affected by noise, caused by for example market volatility or trading activity. The observed prices on the market can therefore be seen as 'noisy'. This noise can obscure the true underlying relationship between two assets, which can interfere with the accuracy of generating trading signals in the pairs trading strategy.

In this thesis, the focus lies on the second step of pairs trading, where the aim is to more accurately model the dynamics of the asset prices. This will be done by analyzing the price difference between the two securities, referred to as the spread and using statistical methods for determining characteristics of this spread, such as the historical mean, and minimizing noise from the noisy market observations.

There exist multiple useful mathematical tools for reducing noise in the observed prices. In order to use these tools, the spread will be modeled as a mean reverting process. Mean reversion is a property of a time series which entails that there is a long-term average value around which the series may fluctuate over time but eventually will revert back to [4]. As previously mentioned, this characteristic is fundamental to pairs trading.

The mean-reverting process described above can be reformulated as a state space model, which is a way of representing a dynamic system where the true underlying state evolves over time but is not directly observed. Instead, only noisy measurements related to that state are observed. In the context of pairs trading, the "state" refers to the true value of the spread between the asset prices, while the "observations" are the actual noisy spread measurements seen in the market. A state space model captures this setup with two equations: the state equation, which governs the evolution of the hidden spread, and the observation equation, which links the hidden spread to the measured data. This formalism provides a way to deal with uncertainty and noise in both the dynamics of the process and the observations.

To estimate the hidden state from noisy observations, the Kalman filter provides a useful tool. It is a recursive mathematical algorithm that can be used to filter out the market noise from the observed spread such that the true underlying spread prices can be revealed. It combines prior knowledge about the system with new incoming data to produce the best possible estimate of the current state, in this case the true underlying spread prices. As a result, the Kalman filter produces smoother and more reliable estimates of the true spread, improving trading signals.

Although the Kalman filter is a powerful tool for estimating hidden states from noisy data, its accuracy depends on certain parameters that describe how much uncertainty there is in the system and in the observations. These parameters are often difficult to choose correctly in practice, and poor choices can lead to inaccurate results. To improve this, the Expectation-Maximization (EM) algorithm can be used to automatically fine-tune these parameters based on the observed data, leading to more accurate state estimates [3] [5].

Chapter 2 provides a more detailed discussion of the pairs trading strategy. The pair selection process will be briefly explained using the distance method and a pair will be selected from existing literature, as the process of pair selection it outside the scope of this thesis. The chapter then continues with a formalization of the trading strategy itself, using standard techniques such as the Z-score.

In Chapter 3, the Kalman filter is introduced step by step, beginning with an explanation of state space models and a simple illustrative example to build intuition. Once the foundational concepts are clear, a mathematical derivation of the Kalman filter equations will be given. The final part of the chapter demonstrates how the filter can be applied to historical data of the previously chosen asset pair to identify trading opportunities.

Chapter 4 focuses on optimizing certain model parameters of the Kalman filter using the EM algorithm. It starts by building intuition for how the EM algorithm works, followed by a compact mathematical explanation of the method. The chapter concludes by applying the algorithm to the same historical financial data, now using the optimized parameters to potentially improve trading performance.

Finally, Chapter 5 presents the key findings of the thesis and discusses other points for consideration and future research.

2
Pairs Trading

The concept of pairs trading was developed at Morgan Stanley in 1985 by Gerry Bamberger and Nunzio Tartaglia [1]. Bamberger and Tartaglia developed statistical methods to identify pairs of stocks that exhibit correlated price movements of which the difference reverts to some long-term mean. Even though the method was not yet formalized, it was widely used in practice. The strategy gained formalization and prominence through the pioneering work of Gatev et al [6]. They introduced a statistical framework for pair selection using a distance-based approach; stocks are coupled into pairs by matching them based on a minimum-distance principle in terms of their normalized historical prices. This is one of the most used and simple methods of pair selection. These pairs are then used in the pairs trading strategy, where one shorts (sells) the overpriced assets and goes long (buys) on the under priced asset.

This chapter describes the methods of pair selection used in this thesis and provides the theoretical foundation for the trading strategies. In Section 2.1 the distance method for pair selection will be explained. In Section 2.2 the trading strategies used in this thesis are to be elaborated on in detail.

2.1. Pair selection

As mentioned in the introduction of this thesis, the first step of making a selection of asset pairs highly influences the effectiveness of the strategy. By using historical data, traders can develop a better understanding of securities within the option pool before making a selection. While the pair selection process is an important part of the overall strategy, it lies outside the scope of this thesis as the primary focus of this work lies on the second step of the pairs trading strategy. For the purposes of this study, the selection of pairs has been based on results from existing literature. These pairs have been found by using the distance method. Even though the process of pair selection is not performed in this thesis, the distance method used in the literature the pairs have been selected from will be explained.

2.1.1. Distance Method

The distance method, introduced by Gatev et al. [6], selects pairs based on the distance between the cumulative returns of two assets as measured over a certain period of time. More concretely, pairs that exhibit a smaller distance during this period are assumed to be more likely to revert back to the mean of the spread.

The distance is described as the total sum of squared differences between the standardized price series of the two assets, see (2.1). The residual series is obtained by taking the difference between their normalized prices [11].

$$D = \sum_{i=1}^{n} (\rho_{xi} - \rho_{yi})^2. \tag{2.1}$$

where ρ_{xi} and ρ_{yi} are normalized asset prices based on their mean and standard deviations. The normalized asset price is given by

$$\rho_{xi} = \frac{\rho_{xt} - \mathbb{E}[\rho_{xt}]}{\sigma_i} \tag{2.2}$$

where ρ_{xt} is the price of asset x at time t, $\mathbb{E}[\rho_{xt}]$ is the mean or expected value of ρ_{xt} and σ_i is the volatility or standard deviation of asset x. Pairs with the smallest distances are selected for trading, under the assumption that assets with similar historical standardized behavior are more likely to exhibit a stable, mean-reverting spread.

2.1.2. Pair Selection Based on Existing Literature

In order to maintain the focus of the thesis on the application of Kalman filtering on pairs trading, the selection of trading pairs is based on empirical results from existing literature. More specifically, this thesis adopts the results of the paper *Pairs Trading in Cryptocurrency Market: A Long-Short Story* by Nair (2021) [11], which explores the profitability of pairs trading in the cryptocurrency market using multiple statistical techniques, including the distance method. The study considers four major cryptocurrencies: Bitcoin (BTC), Ethereum (ETH), Litecoin (LTC), and Neocoin (NEO), which together represent a combination of high-cap (large total market value) and low-cap (smaller total market value) assets and offer diversity in price behavior, liquidity, and volatility. The choice of these coins ensures a mix of established and emerging assets and aims to study whether price co-movements and arbitrage opportunities persist across different market capitalizations.

In the paper, the author evaluates all six unique pairs formed from these four cryptocurrencies over four non-overlapping six-month periods between January 2018 and December 2019. For each subperiod (A-D), the normalized prices of the assets are used to compute the distance as in (2.1) between the pairs.

The following table summarizes the distance values for all six pairs across the first sample period (Panel A: January 1, 2018 – June 30, 2018), as reported in the paper:

Table 2.1: Distance Between Cryptocurrency Pairs (Panel A)

Cryptocurrency Pair	**Distance**
BTC – ETH	102.073
BTC – LTC	252.973
BTC – NEO	155.134
ETH – LTC	157.088
ETH – NEO	**37.782**
LTC – NEO	100.458

As the table indicates, the Ethereum – Neocoin (ETH–NEO) pair has the lowest distance in this period, suggesting it is the most suitable pair for a trading strategy based on price convergence. Similar patterns were observed in other periods as well. Other pairs such as BTC–NEO and BTC–ETH also show favorable distance profiles in other subperiods considered in the study. Based on these findings, the pair ETH–NEO is selected in this thesis to serve as inputs for testing the Kalman filter-based trading strategy to be implemented in Section 3.3.

2.2. Trading Strategy

After selecting suitable pairs for trading in Section 2.1, recall that the second step of pairs trading consists of generating trading signals based on the spread dynamics of these pairs with respect to its long-term equilibrium. In this thesis, the trading framework as proposed by Gatev et al. (2006) [6] is implemented.

The strategy is based on the mean-reversion assumption of the spread; it assumes that there exists a long-term equilibrium to which the spread, over time, will return to. Trading signals are generated when the spread diverges two standard deviations from the equilibrium; more specifically, a long position will be triggered when the spread exceeds the lower threshold (-2 std) and a short position will be triggered when the spread exceeds the upper threshold (+2 std). Positions will be closed when the spread returns back to its equilibrium. Whenever a position remains open and the spread remains to exceed the threshold, no new position will be opened until the open position has been closed.

To standardize signal generation across different pairs and time periods, the Z-score of the spread is used. This allows for fixed thresholds of $+2$ and -2 to be used consistently. The Z-score of the spread S_t at time t is defined as

$$Z_t = \frac{S_t - \mu_S}{\sigma_S} \tag{2.3}$$

where μ_S and σ_S represent the historical mean and standard deviation of the spread, respectively. Trading signals are thus formally defined as:

Long Signal :	$Z_t < -2$
Short Signal :	$Z_t > 2$
Close Long/Short :	$Z_t = 0$

To better adapt the pairs trading strategy to changing market conditions, a rolling-window approach using 6-month periods is used. The full timeline is automatically segmented into consecutive half-year intervals. For each 6-month period, the mean and standard deviation of the spread are computed independently. These values are used to calculate Z-scores for that specific period:

$$z_t = \frac{\mathrm{S}_t - \mu_{\text{period}}}{\sigma_{\text{period}}}$$

This rolling recalibration ensures that trading signals are based on more localized statistics, making the strategy more responsive to structural changes in the spread's charateristics.

While this Z-score-based framework is followed in this thesis, it is worth noting that the paper from which the selected trading pairs are taken implements a slightly different version of the pairs trading strategy. In that paper, the author adopts a daily trading strategy where the position is opened at the beginning of the trading day and closed at the end of the same day. Rather than using fixed statistical thresholds, the strategy relies on identifying the relatively undervalued and overvalued asset within the selected pair, based on the hedge ratio obtained through regression during a prior formation period. Despite these differences, the underlying principle remains the same: exploiting temporary deviations from a stable long-term relationship between asset prices. For the purpose of this thesis, however, the Z-score threshold-based strategy of Gatev et al. (2006) [6] is preferred for its simplicity, interpretability, and widespread use in literature.

3

Fundamentals of the Kalman Filter for Linear Systems

An important assumption of pairs trading is that the spread between the two asset prices follows a mean-reverting process. However, the spread observed in financial markets is typically affected by various sources of noise, making it a noisy rendition of the true underlying process.

The goal is to estimate this true mean-reverting spread from the noisy observations. To achieve this, the Kalman filter can be used. It is a widely used mathematical algorithm to estimate states of a dynamic system from indirect and uncertain measurements. The notion of the filter is to recursively estimate the state of a system at each time step and predict future states of the system based on noisy measurements. This makes the filter useful for a wide range of applications such as navigation and signal processing, where uncertainty in measurements can play a large role.

In the context of pairs trading, the Kalman filter can be used to filter out market noise and obtain a smoother, more reliable estimate of the spread, which can improve the timing and accuracy of trading signals.

In this chapter, we aim to introduce the fundamental notions of Kalman filtering. To explain the use-fullness of the Kalman filter, a basic understanding of state space models is needed. Therefore, in Section 3.1 an introduction will be given about state space models, including an example. After introducing the conceptual framework of Kalman filtering, the mathematical derivation will be discussed in Section 3.2.2. Lastly, the filter will be applied to financial data in Section 3.3.

3.1. State space models

Before introducing the concepts of Kalman filtering, it is necessary to define state-space models, as they provide the mathematical framework on which the filter operates. State space models are mathematical models that describe how an unobserved state evolves over time and how it relates to noisy observations. More often than not, one can only partially observe the state of a system and the parameters of the system can adapt over time. The Kalman filter is specifically designed to optimally estimate the state of such systems.

Definition 1. ***[State-space model]*** *Let $x_t \in \mathbb{R}^n$ be the unobservable state vector and let $y_t \in \mathbb{R}^n$ be the observable measurements. The following difference equations govern the dynamics between the state vector and the measurement vector:*

State Equation*:*

$$x_t = F_t x_{t-1} + w_t, \quad w_t \sim \mathcal{N}(0, Q_t) \tag{3.1}$$

where:

- $F_t \in \mathbb{R}^{n\times n}$ *is the state transition matrix,*
- $w_t \in \mathbb{R}^n$ *is the process noise, assumed to be Gaussian with zero mean and covariance* $Q_t \in \mathbb{R}^{n\times n}$*.*

Observation equation:

$$y_t = H_t x_t + v_t, \quad v_t \sim \mathcal{N}(0, R_t) \tag{3.2}$$

where:

- $H_t \in \mathbb{R}^{p \times n}$ *is the observation matrix,*
- $v_t \in \mathbb{R}^p$ *is the measurement noise, assumed to be Gaussian with zero mean and covariance* $R_t \in \mathbb{R}^{p \times p}$.

The state equation (3.1) outlines the system's hidden state that cannot be observed, and the observation equation (3.2) describes how the underlying state relates to the observable data. The system consisting of (3.1) and (3.2) becomes stochastic by incorporating the process noise w_t and measurement noise v_t. We assume v_t and w_t are white and mutually independent:

$$w_t \sim \mathcal{N}(0, Q_t), \qquad v_t \sim \mathcal{N}(0, R_t),$$

with $\{w_t\}$ and $\{v_t\}$ and independent over t.

To fully specify the model, an initial state x_0 is required. Suppose that there is an initial state $x_0 \sim \mathcal{N}(m_0, P_0)$ where m_0 is the mean value of the multivariate normal distribution of the initial state and P_0 the variance-covariance matrix of the multivariate normal distribution of the initial state . The goal is to further estimate the subsequent states of the system $x_1, x_2, \ldots$. At the same time, data points y_t are observed which shall be used to further estimate x_t. This particular scenario, where one aims to estimate states in real time for dynamic scenarios occurs often in (financial) engineering. In the next section an example will be given that will show the use of state space equations in an engineering setting.

3.1.1. Example: A simple spacecraft

To further deepen the understanding of state space models a short example will be presented [10]. At the end of the example a problem arises which will introduce the need for using Kalman filtering.

In applications related to spacecraft guidance and navigation, the objective is to determine a spacecraft's position and velocity as it travels through space, keeping Newton's laws of motion in mind. Depending on the determined position and velocity of the spacecraft, decisions about subsequent actions must be made. In such scenarios, the aim is to to make use of all available information to generate the most accurate estimates possible.

Consider a spacecraft that is heading towards the moon, launched into a carefully planned path by the thrust of its engines. As it travels through space, its position relative to Earth must be continuously estimated. To achieve this, periodic observations of surrounding stars are used to update and refine the spacecraft's location.

Let x_t be the radial distance of the spacecraft from Earth. For any acceleration, it holds by Newton's law that if the spacecraft's position at time $t-1$ is p_{t-1} and it's velocity is v_{t-1}, then it's position at t is:

$$p_t = p_{t-1} + v_{t-1}\Delta t + w_t \tag{3.3}$$

Here Δt is the time elapsed between time points and w_t represents the noise where $w_t \sim N(0, \tau^2)$. The equation represents the idea that 'a body in motion stays in motion'. Using vector and matrix notation we can rewrite (3.3) as:

$$\begin{bmatrix} p_t \\ v_t \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_{t-1} \\ v_{t-1} \end{bmatrix} + \begin{bmatrix} w_{1t} \\ w_{2t} \end{bmatrix} \tag{3.4}$$

So, if there is no acceleration the velocity from time $t-1$ to time t will clearly not change. Define the following variables:

$$x_t = \begin{bmatrix} p_t \\ v_t \end{bmatrix}$$

$$F = \begin{bmatrix} p_{t-1} \\ v_{t-1} \end{bmatrix}$$

$$w_t = \begin{bmatrix} w_{1t} \\ w_{2t} \end{bmatrix}$$

Then the state equation will simply be:

$$x_t = Fx_{t-1} + w_t \tag{3.5}$$

This is of the form as in (3.1). So far, any observed data has not been incorporated. Without any measurements from the system, all trust lies solely on the state equation to describe how the system evolves. Therefore, if the initial state x_0 is known, the best prediction for the future states is determined by the following model alone:

$$\hat{x_1} = F\hat{x_0}, \hat{x_2} = F\hat{x_1}, \ldots$$

It is important to note that these future states are better described as predictions rather than estimates, as no observational data has yet been incorporated. They are projections of the system's behavior based solely on its internal dynamics.

Naturally, the question that follows is what happens if there is data y_t available that is observed at time t? And what is the impact of the amount of measurements on the estimation of our state at time at future time steps?

Assume that at discrete intervals, onboard instruments provide measurements of the system's state. Thus, at time t an observation of the current position y_t is given by:

$$y_t = p_t + v_t \tag{3.6}$$

where v_t $N(0, \sigma^2)$. Thus, y_t is a noisy representation of the true position of the spacecraft. As before, rewrite (3.6) in matrix vector form as:

$$y_t = \begin{bmatrix} 1 & 0 \end{bmatrix} x_t + v_t \tag{3.7}$$

By defining $H = \begin{bmatrix} 1 & 0 \end{bmatrix}$ (3.7) can be simply rewritten in the standard observation equation format as in (3.2):

$$y_t = Hx_t + v_t \tag{3.8}$$

Here one can wonder what the result on x_t is by observing y_t, the focus will revolve around three key types of inference by using observations:

1. Prediction: Forecasting subsequent values of the state
2. Updating: Estimating the current values of the state from past and current observations

The Kalman filter can be used to carry out the different types of inference described above.

3.2. The Kalman filter

The Kalman filter is a mathematical algorithm that was first developed by Rudolf Kalman in the late 1950s [8]. The Kalman filter was originally designed for the guidance and navigation of spacecrafts (an example of which we have seen in the previous section), but its applications have since expanded to many other fields such as economics, control systems, and signal processing due to it's adeptness in extracting useful information from noisy data. It supports estimations of past, present and even future states when the underlying nature of the modeled system is unknown. The filter is able to propagate state estimations through time with minimal variance.

In this section, the mathematical foundations of the Kalman filter will be explained.

3.2.1. Methods of derivation

To derive the Kalman filter, we first consider a fundamental problem in estimation theory: given an observed noisy signal, how can we best estimate the underlying true state? Consider a state space model as defined in Definition 1. The key challenge is that we do not observe x_t directly, but rather only its noisy measurement y_t. Our goal is to find an estimator $\hat{x}_t$ that best approximates x_t, which will be obtained by minimising the mean squared error.

Eventually in Subsection 3.2.2 the filter is constructed as a mean squared error minimiser and some Bayesian techniques will be used. In this section an introduction of the MSE will be given. In addition, an alternative derivations of the filter is briefly provided showing how the filter relates to Bayesian methods.

Consider a cost function which if often used in optimization problems. A cost function quantifies the error or loss associated with a particular choice of parameters or estimates. The goal is typically to minimize this function to achieve the best possible estimate. In the context of the Kalman filter, we define a cost function to measure the discrepancy between the estimated state $\hat{x}_t$ and the true state x_t. One way to measure the accuracy of our estimate is through the error:

$$e_t = x_t - \hat{x}_t. \tag{3.9}$$

A cost function should ideally satisfy the following properties:

- It should be positive, ensuring that larger errors result in higher penalties.
- It should increase as the error magnitude grows, reinforcing the idea that larger deviations are worse.

A common choice for the cost function is the squared error as it is positive and monotonically increasing:

$$f(e_t) = ||e_t||^2. \tag{3.10}$$

Since we deal with multiple measurements over time, we take the expectation of the squared error, leading to the definition of the mean squared error (MSE):

$$\Phi(t) = E[e_t^2]. \tag{3.11}$$

Minimizing this function provides the optimal estimate of x_t in the least-squares sense if all the noise is Gaussian. From this minimization process, the Kalman equations can be derived.

Apart from minimizing the mean squared error, the Kalman filter can also be derived using Bayesian inference. These will be discussed briefly and some Bayesian techniques will be used in deriving the filter equations in Subsection 3.2.2.

The Kalman filter can be derived as the optimal Bayesian estimator under linear-Gaussian assumptions by updating the estimated state x_t based on new observations $y_{1:t}$. Using Bayes' theorem:

$$p(x_t|y_{1:t}) \propto p(y_t|x_t)p(x_t|y_{1:t-1}), \tag{3.12}$$

the term $p(x_t|y_{1:t-1})$ represents the predicted state distribution based on the system dynamics, while $p(y_t|x_t)$ describes how likely the observation is given the state. Under Gaussian assumptions, this results in a normal distribution update, leading to the Kalman filter equations. Under Gaussian assumptions, both distributions are normal, and their product (via Bayes' rule) results in another normal distribution. This Bayesian update naturally leads to the Kalman filter equations, where the prediction step computes $p(x_t|y_{1:t-1})$ and the update step computes $p(x_t|y_{1:t})$.

This approach reinforces that the Kalman filter is optimal under linear-Gaussian assumptions, providing the minimum mean square error estimator that coincides with the Bayesian posterior mean.

3.2.2. Kalman Filter Derivation

Define the state equation of a system as in Definition (1):

$$x_t = F_t x_{t-1} + w_t \tag{3.13}$$

$$y_t = H_t x_t + v_t \tag{3.14}$$

Recall that the random variables w_t and v_t are assumed to be Gaussian, i.e. $w_t \sim \mathcal{N}(0, Q_t)$ and $v_t \sim \mathcal{N}(0, R_t)$. Furthermore, we assume that w_t and v_t are independent of each other. This provides us with the following property which is needed in deriving the filtering equations:

Lemma 1. *If the random variables w_t and v_t are independent of each other, then $E[w_t v_t] = E[w_t]E[v_t]$*

The goal of the Kalman filter is to estimate the state x_t given the measurements y_t. As mentioned in the Section 3.1.1, the Kalman filter consists of a prediction step and an update step. In the prediction step, we predict the value of the state at the next time step. In the update step, we adjust our prediction obtained in the prediction step using obtained measurements. We define these estimates of the state x_t as follows:

Definition 2. *Let* $\hat{x}_t^{'}$ *denote the* a priori *state estimate at time step* t*, defined as the conditional expectation of the state given all observations up to time* $t-1$*:*

$$\hat{x}_t^{'} = \mathbb{E}[x_t \mid y_{1:t-1}].$$

Similarly, let $\hat{x}_t$ *denote the* a posteriori *state estimate at time step* t*, defined as the conditional expectation of the state after incorporating the new observation* y_t*:*

$$\hat{x}_t = \mathbb{E}[x_t \mid y_{1:t}].$$

To measure the accuracy of the estimates defined in Definition 2 , we need to keep track of the error between predicted values and true values. This is done by using the same error function as (3.9) applied to the a priori and posteriori estimates:

Definition 3. *Define the a priori estimate error (3.15) and a posteriori estimate error (3.16) as*

$$e_t' = x_t - \hat{x}_t' \tag{3.15}$$

$$e_t = x_t - \hat{x}_t \tag{3.16}$$

As discussed in Section 3.2.1 the Kalman filter minimizes the mean squared error (MSE) between the true state and the estimated states. The MSE is given by the estimates error covariances that are defined as below:

Definition 4. *The a priori estimate error covariance and the a posteriori estimate covariance are given by*

$$P_t' = E[e_t' e_t^{T'}] \tag{3.17}$$

$$P_t^- = E[e_t e_t^T] \tag{3.18}$$

, respectively.

We now derive the five equations which form the Kalman filter. To this end, suppose that we are at time $t \in \{1, \ldots, T\}$. This means that we know what our a priori and a posteriori hidden state estimates, $\hat{x}_t'$ and $\hat{x}_t$ are up to t. Furthermore, we have obtained measurements $y_1, \ldots y_t$. With this knowledge, the first Kalman filter equation can be obtained by taking the expectation of the state equation 3.27, this results in the state extrapolation equation. Note that this is a Bayesian approach.

Definition 5. *Suppose F,* x_t *and* $\hat{x}$ *are known. Then the state extrapolation equation is known as*

$$\hat{x}_{t+1} = F\hat{x}_t \tag{3.19}$$

Proof: Let $\mathcal{Y}_t$ *be all observations up to and including* t*, so* $\mathcal{Y}_{t-1} = \{y_1, y_2, \ldots, y_{t-1}\}$*. Taking the conditional expectation of the state equation given* $\mathcal{Y}_t$*:*

$$\begin{aligned}
\hat{x}_{t+1} &= \mathbb{E}[Fx_t + w_{t+1} \mid \mathcal{Y}_t] \\
&= F\mathbb{E}[x_t \mid \mathcal{Y}_t] + \mathbb{E}[w_{t+1} \mid \mathcal{Y}_t] \\
&= F\hat{x}_t + 0 \\
&= F\hat{x}_t,
\end{aligned}$$

since $\mathbb{E}[w_{t+1} \mid \mathcal{Y}_t] = 0$ *and* w_t *is independent of* $\mathcal{Y}_{t-1}$*. Thus, the state extrapolation equation is:* $\hat{x}_{t+1} = F\hat{x}_t$

The state extrapolation equation is used to predict the state at the next time step, assuming the system is fully deterministic (i.e. the process noise w_t is ignored). Apart from keeping track of the state extrapolations, we aim to keep track of the uncertainty of these extrapolations as well. To do so, also assume that up to time t, we know what both estimates for the error covariance matrices, P_t' and P_t are. We compute

$$
\begin{aligned}
P'_{t+1} &= \mathbb{E}\left(e^{-}_{t+1}e^{-T}_{t+1}\right) \\
&= \mathbb{E}\left((x^{-}_{t+1} - x_{t+1})(x^{-}_{t+1} - x_{t+1})^T\right) \\
&= \mathbb{E}\left((F\hat{x}_t - (Fx_t + w_t))(F\hat{x}_t - (Fx_t + w_t))^T\right) \\
&\quad \text{Plugging in (3.15) and (3.19)} \\
&= \mathbb{E}\left((F(\hat{x}_t - x_t) - w_t)(F(\hat{x}_t - x_t) - w_t)^T\right) \\
&= \mathbb{E}\left((F(\hat{x}_t - x_t) - w_t)((\hat{x}_t - x_t)^T F^T - w_t^T)\right) \\
&= \mathbb{E}\left(F(\hat{x}_t - x_t)(\hat{x}_t - x_t)^T F^T - F(\hat{x}_t - x_t)w_t^T - w_t(\hat{x}_t - x_t)^T F^T + w_t w_t^T\right) \\
&= F\mathbb{E}\left((x_t - \hat{x}_t^{-})(x_t - \hat{x}_t^{-})^T\right) F^T \\
&\quad - F\mathbb{E}((x_t - \hat{x}_t^{-})w_t^T) - \mathbb{E}(w_t(x_t - \hat{x}_t^{-})^T)F^T + \mathbb{E}(w_t w_t^T) \\
&\quad \text{Since } e'_t \text{ and } w_t \text{ are independent, and } \mathbb{E}(w_t) = 0, \\
&= FP_tF^T - 0_{n\times n} - 0_{n\times n} + Q \\
&= FP_tF^T + Q.
\end{aligned}
$$

This prediction equation is known as the covariance extrapolation equation where P'_t captures the uncertainty in our prediction. It is the second equation of the Kalman filter.

Proposition 1. *Suppose P_t is known. Then the Covariance extrapolation equation is given by*

$$P'_{t+1} = FP_tF^T + Q \tag{3.20}$$

Proof: See derivation above.

Now that we have predicted our next state $\hat{x}'_{t+1}$ and it's covariance matrix P'_{t+1}, we update the subscripts such that we move forward in time in time $t \to t+1$, so $\hat{x}'_{t+1} \to x_t$ and $P'_{t+1} \to P_t$. Right after this forward move, we collect our measurement y_t about the true state x_t.

The matrix P'_t has the property that it is symmetric, since from linear algebra we have the fact that a matrix ABA^T is symmetric if B is symmetric (given that A and B are of fitting dimensions). Also Q is symmetric, so the Covariance extrapolation equation (3.20) ensures that P'_t is symmetric. Then, the total variance of the a priori state estimate x'_t equals the trace of P'_t, since the off-diagonal elements denote covariances and are not of interest in the minimization problem. To recall, the definition of a trace of a matrix is given below.

Definition 6. *Let $A \in \mathbb{R}^{n\times n}$ be a square matrix. The* trace *of A, denoted* Tr(A)*, is defined as the sum of its diagonal elements:*

$$\text{Tr}(A) = \sum_{i=1}^{n} A_{ii}.$$

As previously mentioned, the goal of the filter is to minimize the state's estimator variance. Each diagonal entry of P'_t is the variance of the error in estimating a single component of the state. Using the trace of P'_t we can obtain the total mean squared error of the state prediction, i.e. the total variance of the a priori state estimate x'_t.

If $Tr(P_t)'$ is small, more faith is put into the predicted states. However, it could be the case that the total variance of the measurement noise R is smaller than $Tr(P_t)'$ This implies that the measurements tend to be more accurate than our the a priori hidden state estimates. Hence, whenever updating our prediction $\hat{x_t}'$ we tend to move 'towards' our measurement value if $Tr(R)$ is smaller than $Tr(P_t)'$ and vice versa. To put the concept of putting 'trust' in estimates versus the measurements into concrete formulas, we propose the following definition:

Proposition 2. *Let $y_t - H\hat{x}_t$ be the innovation, which is the difference between the measurement value and the prediction of our measurement value based on $\hat{x}_t$.*

Note that if the a priori estimates are precise, the innovation should be small and vice versa. Consider a linear combination of the a priori estimate and the innovation. This is the updated estimate of the hidden state. By assigning a weight to the innovation, one is able to quantify how much 'trust' is put into the estimate. The assigned weight is known as the Kalman gain K_t, which will be derived later on in the third Kalman filter equation. For now, assume K_t is known. This enables us to describe the fourth filter equation, the state update equation, in the following manner.

Proposition 3. *Suppose $\hat{x}_t, H, z_t$ are known. Then, the state update equation is*

$$\hat{x}_t = \hat{x}'_t + K_t(y_t - H\hat{x}'_t) \tag{3.21}$$

where K_t is the Kalman gain.

This equation plays a large role in determining the last Kalman equation, the covariance update equation. This equation is part of the correction step, where we refine the certainty of our state estimate after incorporating the latest observation. First, rewrite the state update equation (3.21) using the observation equation (3.28):

$$\begin{aligned}\hat{x}_t &= \hat{x}'_t + K_t(y_t - H\hat{x}'_t)\\ &= \hat{x}'_t + K_t(Hx_t + v_t - H\hat{x}'_t),\end{aligned}$$

which we substitute into the a posteriori error expression (3.16) by

$$\begin{aligned}e_t &= \hat{x}_t - x_t\\ &= \hat{x}'_t + K_t(Hx_t + v_t - H\hat{x}'_t) - x_t\\ &= (I_n - K_tH)\hat{x}'_t + (K_tH - I_n)x_t + K_tv_t\\ &= (I_n - K_tH)(\hat{x}'_t - x_t) + K_tv_t\\ &= (I_n - K_tH)e'_t + K_tv_t.\end{aligned}$$

Now compute the a posteriori error covariance P_t, defined as:

$$P_t = \mathbb{E}[e_te_t^\top].$$

Substitute the expression for e_t:

$$\begin{aligned}P_t &= \mathbb{E}\left[((I - K_tH)(x_t - \hat{x}'_t) - K_tv_t)\,((I - K_tH)(x_t - \hat{x}'_t) - K_tv_t)^\top\right]\\ &= (I - K_tH)\mathbb{E}[(x_t - \hat{x}'_t)(x_t - \hat{x}'_t)^\top](I - K_tH)^\top + K_t\mathbb{E}[v_tv_t^\top]K_t^\top\\ &\quad - (I - K_tH)\mathbb{E}[(x_t - \hat{x}'_t)v_t^\top]K_t^\top - K_t\mathbb{E}[v_t(x_t - \hat{x}'_t)^\top](I - K_tH)^\top.\end{aligned}$$

By assumption, the process noise $x_t - \hat{x}'_t$ is independent of the measurement noise v_t, so the cross terms vanish:

$$\mathbb{E}[(x_t - \hat{x}'_t)v_t^\top] = \mathbb{E}[v_t(x_t - \hat{x}'_t)^\top] = 0.$$

Furthermore, recall:

$$P'_t = \mathbb{E}[(x_t - \hat{x}'_t)(x_t - \hat{x}'_t)^\top], \quad \mathbb{E}[v_tv_t^\top] = R.$$

So the updated covariance becomes:

Proposition 4. *Suppose that P'_t, H and R are known. Then, the covariance update equation is given by*

$$P_t = (I_n - K_tH)P'_t(I_n - K_tH)^T + K_tRK_t^T \tag{3.22}$$

where K_t is the Kalman gain.

This is the general form of the covariance updatey. In practice, a simplified version is often used which can be derived by plugging in the (upcoming) definition of the Kalman gain:

$$P_t = (I - K_tH)P'_t,$$

Finally, we shall give the proper derivation of the Kalman gain K_t. Recall that it determines how much we trust the new measurement versus our current prediction. A high gain means we trust the measurement a lot; a low gain means we trust our prediction more. For example, consider the state update equation (3.21). For $K_t = 0$, we have that $\hat{x}_t = \hat{x_t'}$ which means that $K_t = \mathbf{0}$ corresponds to full trust into our prediction.

We want the a posteriori estimate $\hat{x_t}$ to be as accurate as possible. This means minimizing the total variance of the error $\hat{x}_t$, which as we recall equals $Tr(P_t)$. The objective is then to choose K_t such that $Tr(P_t)$ is minimized. We now derive the corresponding formula for K_t, using the covariance update equation (3.22). Rewrite as follows:

$$\begin{aligned}
P_t &= (I_n - K_t H)P_t'(I_n - K_t H)^\top + K_t R K_t^\top \\
&= (I_n - K_t H)P_t'(I_n - H^\top K_t^\top) + K_t R K_t^\top \\
&= (P_t' - K_t H P_t')(I_n - H^\top K_t^\top) + K_t R K_t^\top \\
&= P_t' - P_t' H^\top K_t^\top - K_t H P_t' + K_t H P_t' H^\top K_t^\top + K_t R K_t^\top \\
&= P_t' - P_t' H^\top K_t^\top - K_t H P_t' + K_t (H P_t' H^\top + R) K_t^\top,
\end{aligned}$$

The trace operator has a linearity and cyclic property which we can use in the following way:

$$\begin{aligned}
\mathrm{Tr}(P_t) &= \mathrm{Tr}\left(P_t' - P_t' H^\top K_t^\top - K_t H P_t' + K_t (H P_t' H^\top + R) K_t^\top\right) \\
&= \mathrm{Tr}(P_t') - \mathrm{Tr}(P_t' H^\top K_t^\top) - \mathrm{Tr}(K_t H P_t') + \mathrm{Tr}\left(K_t (H P_t' H^\top + R) K_t^\top\right) \\
&= \mathrm{Tr}(P_t') - 2\,\mathrm{Tr}(K_t H P_t') + \mathrm{Tr}\left(K_t (H P_t' H^\top + R) K_t^\top\right),
\end{aligned}$$

We minimize this expression with respect to K_t. For that, we differentiate $\mathrm{Tr}(P_t)$ w.r.t. K_t and set the derivative to zero. This is a matrix calculus problem. Using matrix derivative rules, we find:

$$\frac{\partial}{\partial K_t} \mathrm{Tr}(P_t) = -2P_t' H^\top + 2K_t (H P_t' H^\top + R).$$

Now, for our minimization problem set the gradient to zero:

$$-2P_t' H^\top + 2K_t (H P_t' H^\top + R) = 0 \tag{3.23}$$

$$K_t (H P_t' H^\top + R) = P_t' H^\top. \tag{3.24}$$

And finally solve for K_t by using the fact that $H P_t' H^\top + R > 0$ is invertible, we obtain the Kalman gain.

Proposition 5. *Assume P_t', H, R are known. Then the Kalman gain is given by*

$$K_t = P_t' H^\top (H P_t' H^\top + R)^{-1}. \tag{3.25}$$

This is the optimal Kalman gain that minimizes the trace of the posterior covariance P_t, i.e., minimizes the total mean-squared estimation error.

3.2.3. Summary

Now that we have derived the equations which form the Kalman filter, let us consider a brief summary. The Kalman filter algorithm can be summarized as follows:

1. Start with initial estimates $\hat{x}_0'$ and P_0'.
2. Compute the Kalman gain K_t using the prior error covariance matrix P_k' and the measurement noise covariance R.
3. Update the state estimate $\hat{x}_t$ using the measurement y_t.
4. Update the error covariance matrix P_t.
5. Project the state and error covariance matrix to the next time step.
6. Repeat the process for each new measurement.

The Kalman filter propagates the state estimate by the following five equations.

State extrapolation equation (3.19)	$\hat{x}_t' = F_t \hat{x}_{t-1}$
Covariance extrapolation equation (3.20)	$P_t' = F_t P_{t-1} F_t^\top + Q_t$
Kalman gain (3.25)	$K_t = P_t' H_t^\top (H_t P_t' H_t^\top + R_t)^{-1}$
State update equation (3.21)	$\hat{x}_t = \hat{x}_t' + K_t (y_t - H_t \hat{x}_t')$
Covariance update equation (3.22)	$P_t = (I - K_t H_t) P_t' (I - K_t H_t)^\top + K_t R_t K_t^\top$

Table 3.1: The five equations which form the Kalman filter

3.3. The Kalman Filter applied to financial data

This section presents the empirical results obtained from implementing the Kalman filter-based pairs trading strategy on the ETH-USD and NEO-USD cryptocurrency pair over the period from January 2018 to December 2019, as discussed in Section 2.2. The analysis is structured to evaluate both the filtering performance and the trading strategy effectiveness on the raw and filtered spread.

3.3.1. Mean-reverting process

The theoretical foundation for applying the Kalman filter to pairs trading assumes that the spread between two asset prices follows a mean-reverting process, capturing the idea that deviations from a long-run equilibrium tend to decay over time. In [5] a standard way to model mean-reverting behavior is done using the Ornstein–Uhlenbeck (OU) process, which is a stochastic process that captures both the tendency to revert to a mean and the presence of random fluctuations. The spread is modeled as follows:

$$x_t = \mu + \rho(x_{t-1} - \mu) + \eta_t, \quad |\rho| < 1, \tag{3.26}$$

where x_t denotes the latent spread at time t, μ is the long-term mean, ρ is the mean reversion coefficient, and η_t is white noise with zero mean and variance σ_η^2.

To ensure linearity in the state-space formulation, we define the state as the deviation from the mean: $\tilde{x}_t = x_t - \mu$. The deviation follows:

$$\tilde{x}_t = \rho\tilde{x}_{t-1} + \eta_t, \quad |\rho| < 1$$

In this setting, we assume that the observed spread y_t is a noisy measurement of the true spread:

$$y_t = \tilde{x}_t + \mu + \varepsilon_t$$

where ε_t is measurement noise, assumed to be Gaussian and independent of η_t, with variance σ_ε^2. This way, the spread is mainly driven by the mean-reverting process and slightly by the effect of a Gaussian noise term. Due to this noise term, we end up with the familiar state space model.

$$\tilde{x}_t = F_t\tilde{x}_{t-1} + w_t \tag{3.27}$$

$$y_t = H_t\tilde{x}_t + \mu + v_t \tag{3.28}$$

Here:

- $\tilde{x}_t$ is the latent state (deviation from mean) at time t;
- y_t is the observed spread;
- $F_t = \rho$ is the state transition coefficient;
- $H_t = 1$ is the observation matrix;
- $w_t = \eta_t$ is the process noise;
- $v_t = \varepsilon_t$ is the observation noise;
- μ appears as a known constant in the observation equation.

3.3.2. Kalman Filter Performance Analysis

We begin the empirical analysis by examining the historical price development of the selected cryptocurrency pair, Ethereum (ETH) and NEO, over the sample period from January 2018 until December 2019. Figure 3.1 displays the adjusted daily closing prices of both assets showing the high volatility characteristic of cryptocurrency markets during this period.

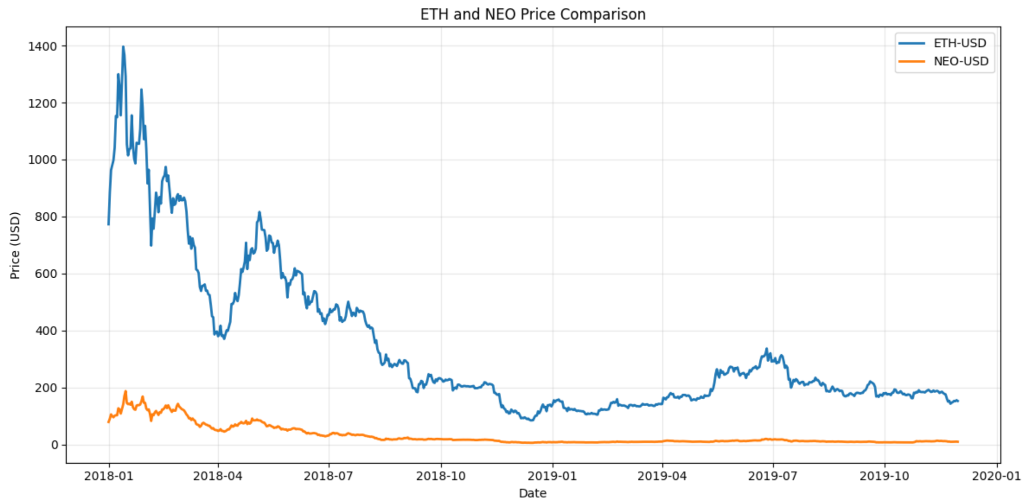


Figure 3.1: Adjusted daily closing prices of ETH and NEO

The Kalman filter can be applied to estimate the latent spread from the observed historical data. In practice, Python libraries such as `PyKalman` provide a convenient interface for implementing the filter. When initializing the Kalman filter, the library requires values for several parameters, including the state transition coefficient F_t, the observation matrix H_t, and the covariances of the process and observation noises w_t and v_t.

By default, `PyKalman` uses generic starting values for these parameters. It assumes $F_t = 1$, $H_t = 1$, and assigns initial variances of 1 to the noise terms. While these defaults allow the filter to operate, they do not reflect the theoretical properties of a mean-reverting spread. In our model, the spread is assumed to follow a mean-reverting proces, where the latent deviation from the long-term mean $\tilde{x}_t = x_t - \mu$ evolves as

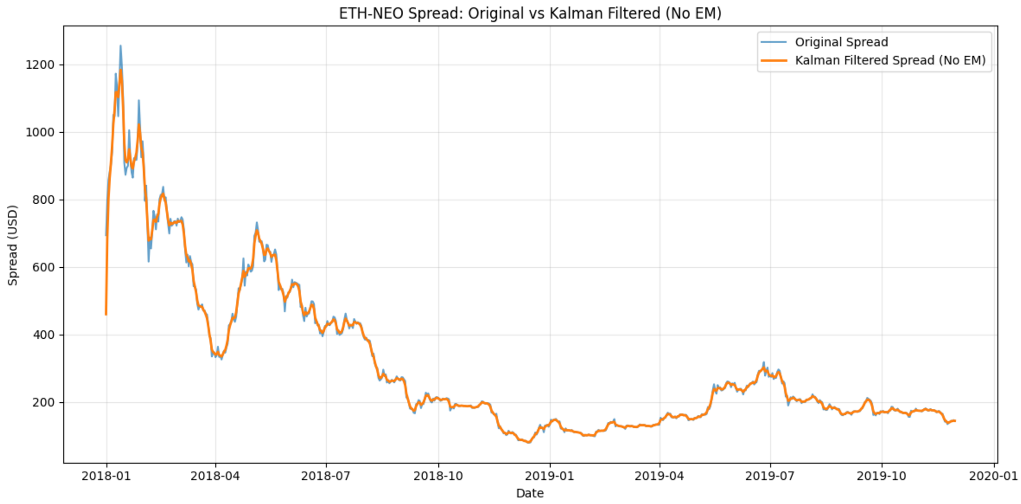
$$\tilde{x}_t = \rho\tilde{x}_{t-1} + \eta_t, \quad |\rho| < 1.$$

Consequently, the state transition coefficient should be set to ρ rather than 1, capturing the tendency of the spread to revert toward its long-term mean. Similarly, the variances of the process and observation noise, σ_η^2 and σ_ε^2, should reflect the characteristics of the spread and the measurement noise.

When first implementing the Kalman filter on historical spread data the default parameters provided by `PyKalman` will be used for practicality, only the long-term mean μ of the spread is determined from historical averages. These defaults allow the filter to run without requiring extensive preliminary analysis of the data. While this approach does not guarantee optimal parameter values, it provides a simple and functional baseline implementation for the Kalman filter.

Figure 3.2 demonstrates the comparison between the original spread and the Kalman filtered version. The filtered spread shows significantly reduced volatility while preserving the main price movement patterns.

Figure 3.2: Kalman filter applied to the spread between ETH and NEO



3.3.3. Trading Strategy Performance

The filtered spread was subjected to the same Z-score-based trading strategy described in Section 2.2, using ±2 standard deviations as entry thresholds and a rolling 6-month window for parameter recalibration. To compare the effect of using Kalman filtering on the spread, the trading strategy will be applied to both the unfiltered spread and the filtered spread.

In Figure 3.3 the trading strategy is applied to the unfiltered spread. It illustrates the trade entry and exit points overlaid on the filtered spread, showing the timing and magnitude of each trading opportunity identified by the algorithm.

It can be seen that 4 completed trades were performed during the sample period achieving the following performance metrics:

- Total profits and losses: $ 725.73
- All trades were profitable
- Average profits and losses $ 181.43 per trade
- Individual trade results: $396.17, $234.83, $6.9, $87.83

Figure 3.3: Trade entry and exit points over time, overlaid on the unfiltered spread.

In Figure 3.4 it can be seen that 3 completed trades were performed during the sample period achieving the following performance metrics:

- Total profits and losses: \$ 523.95
- All trades were profitable
- Average profits and losses \$ 174.65 per trade
- Individual trade results: \$427.62, \$5.34, \$90.99

Figure 3.4: Trade entry and exit points over time, overlaid on the filtered spread.

Interestingly, the highest profits are obtained when trading on the original spread, while trading on the Kalman-filtered spread produced lower profits.

This result can be explained by the interaction between noise, smoothing, and trading signals. The Kalman filter with default parameters appears to over-smooth the data, removing not only noise but also legitimate price movements that could generate profitable trading signals. While this produces more statistically reliable estimates and reduces false signals [7, 5], it also eliminates some short-term movements that may have been profitable in the historical data. As a result, fewer trades are triggered or trades occur with smaller deviations, leading to lower realized profits.

The underperformance of the default Kalman filter suggests that the assumed noise structure does not match the actual characteristics of the spread data. The equal weighting of process and observation noise ($Q = R = 1$) may not reflect the true volatility patterns in cryptocurrency markets, leading to inappropriate smoothing behavior.

Furthermore, the assumption of a random walk ($F = 1$) rather than mean reversion may fundamentally misrepresent the spread dynamics, causing the filter to track long-term trends rather than identify mean-reverting opportunities.

To address this, the Expectation-Maximization (EM) algorithm can be employed. The EM algorithm iteratively updates the initial parameters to maximize the likelihood of the observed data under the Kalman filter model. By using EM, we can obtain parameter estimates that more faithfully reflect the true dynamics of the spread, improving the filter's performance and the reliability of trading signals.

4

Expectation Maximization

In the Chapter 3, we introduced the Kalman filter as a recursive method for estimating the hidden state of a linear dynamical system from noisy observations. While the Kalman filter is powerful, its performance heavily relies on the specification of certain model parameters. Most notably, the covariance matrices of the process noise and measurement noise. In practice, however, these covariances are often unknown and difficult to estimate manually. Moreover, the state transition matrix and observation matrix may not be fixed or known a priori, especially in real-world applications like pairs trading where the system dynamics can evolve over time. To overcome these limitations, one can apply the Expectation-Maximization (EM) algorithm, an iterative method for maximum likelihood estimation in models with hidden variables. By alternating between estimating the hidden states (given current parameter estimates) and updating the parameters (based on these estimated states), the EM algorithm can estimate the parameters of a state-space model from data, without requiring prior knowledge of the noise structure.

In Section 4.1 an intuitive explanation of the EM algorithm will be given. Then, in Section 4.2 we move on to a short mathematical derivation of the algorithm, with a focus on its application to linear Gaussian state-space models like the Kalman filter. Finally, we revisit the financial setting of the previous chapter and demonstrate how incorporating EM-based parameter estimation into the Kalman filter framework can improve performance when applied to real pairs trading data in Section 4.3.

4.1. Intuition Behind the EM Algorithm

The EM algorithm provides a systematic approach to maximum likelihood estimation in models containing latent variables. In the context of Kalman filtering, the algorithm addresses a fundamental limitation: the requirement to specify system parameters that are often unknown in practice.

Consider the challenge faced in Section 3.3, where default parameter values were used for the Kalman filter implementation. The process noise covariance Q and observation noise covariance R were arbitrarily set to 1, while the state transition matrix F was assumed to equal 1 (random walk). These choices, while allowing the filter to operate, do not necessarily reflect the true underlying dynamics of the financial data.

The EM algorithm resolves this issue by treating the model parameters $\theta = F, H, Q, R$ as unknown quantities to be estimated from the observed data. The latent state sequence $\{x_1, x_2, ..., x_t\}$ plays the role of missing data, creating an incomplete data problem where direct maximum likelihood estimation becomes intractable.

The algorithm operates through two alternating steps:

- **E-Step (Expectation)**: Given current parameter estimates θ^t, compute the expected value of the complete-data log-likelihood with respect to the posterior distribution of the latent states. This step effectively ”fills in” the missing state information using the current parameter estimates and all available observations
- **M-Step (Maximization)**: Maximize the expected log-likelihood computed in the E-step to obtain updated parameter estimates θ^{t+1}.

This process is repeated until convergence, and each iteration is guaranteed to increase the likelihood (or leave it unchanged).

In the context of the Kalman filter, the hidden state sequence plays the role of the latent variables, and the EM algorithm helps estimate the unknown parameters of the state-space model, such as the state transition matrix F, the observation matrix H, and the noise covariance matrices Q and R. Instead of fixing these matrices in advance, the EM algorithm allows us to learn them directly from observed data by leveraging the structure of the Kalman smoother in the E-step and closed-form updates in the M-step. This makes EM an essential tool for improving the robustness of the Kalman filter in practical financial applications like pairs trading, where the system dynamics are not static and parameters must be inferred from historical price data.

4.2. Mathematical Derivation of the EM Algorithm

While a complete mathematical derivation is beyond the scope of this thesis, the essential concepts needed to understand its application to Kalman filtering are provided. For a comprehensive treatment of the EM algorithm's theoretical foundations, see Dempster et al. [2] and McLachlan and Krishnan [9].

Let Y be the observed data, X be the latent (unobserved) variables, and θ be the parameters of the model. The goal is to maximize the log-likelihood of the observed data:

$$\log p(Y \mid \theta)$$

However, this is often difficult to compute directly because it involves integrating over the latent variables X:

$$\log p(Y \mid \theta) = \log \int p(Y, X \mid \theta)\, dX$$

The EM algorithm circumvents this difficulty by iteratively maximizing a lower bound on the log-likelihood.

In the E-step, we compute the expected value of the complete-data log-likelihood, conditioned on the observed data X and the current estimate of the parameters $\theta^{(t)}$. This expectation is denoted by:

$$Q(\theta \mid \theta^{(t)}) = \mathbb{E}_{X \mid Y, \theta^{(t)}} \left[\log p(Y, X \mid \theta)\right]$$

Here, $Q(\theta \mid \theta^{(t)})$ is a function of θ, and $\theta^{(t)}$ is the current estimate of the parameters at iteration t.

In the M-step, we maximize the function $Q(\theta \mid \theta^{(t)})$ with respect to θ to obtain a new estimate of the parameters:

$$\theta^{(t+1)} = arg_{max_\theta Q(\theta \mid \theta^{(t)})})$$

This new estimate $\theta^{(t+1)}$ is used in the next iteration of the E-step.

The EM algorithm guarantees that the log-likelihood of the observed data $\log p(X \mid \theta)$ increases (or remains constant) with each iteration. This is because the E-step constructs a lower bound on the log-likelihood, and the M-step maximizes this lower bound. The algorithm converges when the change in the log-likelihood between iterations becomes negligible.

4.2.1. EM Algorithm for the Kalman Filter

The EM algorithm can be used to estimate the parameters of the Kalman filter, such as the process noise covariance Q and the measurement noise covariance R. The states x_t are treated as latent variables, and the observed data are the measurements y_t.

In the E-step, we compute the expected value of the complete-data log-likelihood, which depends on the latent states x_t. This involves computing the following quantities:

- The state estimates $\hat{x}_t$ and their covariances P_t.
- The cross-covariance between consecutive states $P_{t,t-1}$.

These quantities are computed using the current estimates of the parameters $Q^{(t)}$ and $R^{(t)}$.

In the M-step, we update the parameters Q and R by maximizing the expected complete-data log-likelihood. The updates are given by:

$$Q^{(t+1)} = \frac{1}{N} \sum_{k=1}^{N} (\hat{x}_k - F\hat{x}_{k-1}) (\hat{x}_k - F\hat{x}_{k-1})^T + FP_{k-1}F^T - P_{k,k-1}F^T - FP_{k,k-1}^T$$

$$R^{(t+1)} = \frac{1}{N} \sum_{k=1}^{N} (y_k - H\hat{x}_k) (y_k - H\hat{x}_k)^T + HP_k H^T$$

Here, N is the number of time steps, F is the state transition matrix, and H is the measurement matrix.

The convergence of the log-likelihood is important for several reasons:

- **Optimality**: When the log-likelihood converges, it indicates that the algorithm has found a (local) maximum of the likelihood function. This ensures that the estimated parameters are optimal given the observed data.
- **Stopping Criterion**: The convergence of the log-likelihood provides a natural stopping criterion for the EM algorithm. Once the change in the log-likelihood between iterations falls below a predefined threshold, the algorithm can be terminated.
- **Model Evaluation**: The value of the log-likelihood at convergence can be used to compare different models or parameterizations. A higher log-likelihood indicates a better fit to the data.

The EM algorithm is a powerful tool for parameter estimation in the Kalman filter. It iteratively improves the estimates of the parameters Q and R by alternating between the E-step (computing expected values) and the M-step (maximizing the expected log-likelihood). The convergence of the log-likelihood ensures that the algorithm has found an optimal set of parameters, making it a reliable method for parameter estimation in systems with latent variables.

4.3. The EM algorithm applied to financial data

In Section 3.3, we applied a Kalman filter with the default parameters provided by `pykalman`, which essentially assumes a random walk for the state variable.

To further improve the estimation of the spread's long-term equilibrium (and therefore trading signals), a Kalman filter combined with the Expectation-Maximization (EM) algorithm is applied. The algorithm converged after several iterations, yielding the following optimized parameter estimates:

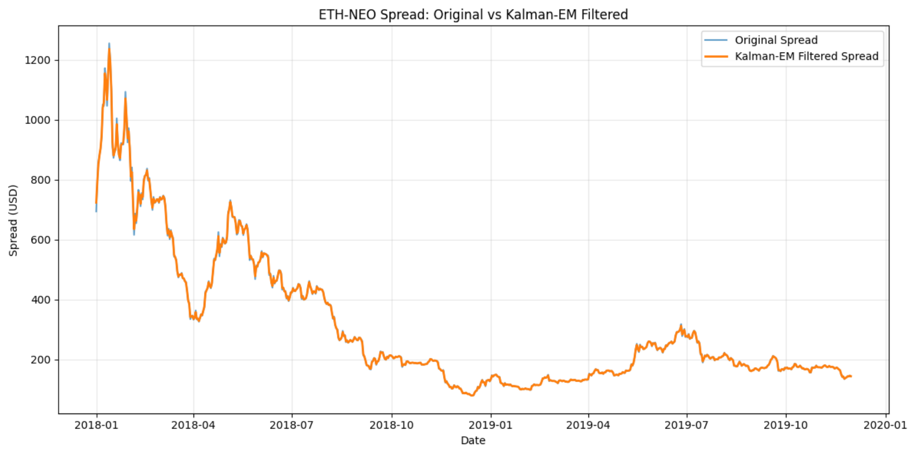
- $F = 1$ (transition matrix)
- $H = 1$ (observation matrix)
- $Q = 467.28$ (process noise covariance)
- $R = 66.11$ (observation noise covariance)

The convergence of F to exactly 1 is mathematically significant, as it indicates that the spread does not seem to follow the mean-reverting dynamics typically assumed in pairs trading models. This result was consistent across multiple initializations of the EM algorithm.

The substantial increase in the estimated noise parameters compared to the default values $(Q = 1, R = 1)$ reflects the high volatility present in cryptocurrency markets. The ratio $Q/R \approx 7.07$ indicates that the process noise dominates the observation noise, meaning that most of the uncertainty in the system stems from the underlying dynamics rather than measurement errors.

Figure 4.1 compares the original spread with the EM-optimized Kalman filtered version.The filtered spread exhibits reduced short-term volatility while preserving the major price movements, demonstrating effectiveness in noise reduction.

Figure 4.1: ETH-NEO Spread: Original vs Kalman-EM Filtered



When applied to the same rolling 6-month trading strategy used previously, the EM-optimized Kalman filter generated improved performance metrics:

- Total profits and losses: $750.83
- All trades were profitable
- Average profits and losses $187.71 per trade

Figure 4.2 illustrates the trading signals generated by the EM-optimized filter overlaid on the filtered spread. The improved parameter estimates resulted in more responsive signal generation, producing an additional profitable trade while maintaining the perfect win rate.

Figure 4.2: Trade entry and exit points over time, overlaid on the Kalman-EM filtered spread.

Table 4.1 provides a comprehensive comparison between the default and EM-optimized approaches. The EM optimization achieved a 43.3% improvement in total profits.

The superior performance of the EM-optimized filter can be explained to its more accurate modeling of the noise characteristics in the data. By estimating Q and R from the data rather than using arbitrary default values, the filter achieved a better balance between smoothing noise and preserving genuine price movements.

Metric	Default Kalman (No EM)	EM-Optimized Kalman
Number of trades	3	4
Total P&L ($)	523.95	750.83
Average P&L per trade ($)	174.65	187.71
Win rate (%)	100	100
Transition matrix F	1	1
Process noise Q	1	467.2
Observation noise R	1	66.11

Table 4.1: Comparison of trading performance using Kalman filter with default parameters versus EM-optimized parameters.

This allowed the trading algorithm to identify profitable opportunities that were obscured by the over-smoothing effect of the poorly calibrated default parameters. However, it should be noted that the absence of mean reversion (F = 1) raises questions about the theoretical foundation of the observed profits. The trading strategy's success may reflect the filter's superior noise reduction capabilities rather than exploitation of mean-reverting dynamics.

5
Discussion and conclusion

This thesis investigated the use of Kalman filtering techniques in the context of statistical arbitrage, with a focus on pairs trading. Building on the framework of Gatev et al. [6] and using a suitable pair for trading from Saji, T G [11], the trading strategy was implemented on the raw spread and on the filtered spread using a Kalman filter with default parameters. The study was further extended by applying the Expectation-Maximization (EM) algorithm to estimate the latent parameters of the Kalman filter.

First, the default Kalman filter performed worse than trading on raw spreads, generating \$523.95 compared to \$725.73 in profits. This demonstrates that poorly specified filtering parameters can actually harm performance by over-smoothing potentially profitable signals. However, when parameters were optimized using the EM algorithm, the Kalman filter achieved the best performance with \$750.83 in total profits and higher average returns per trade.

A particularly surprising finding was that the state transition coefficient F consistently converged to 1, indicating that the spread behaved as a random walk rather than exhibiting the mean-reverting properties assumed in pairs trading theory. This challenges the fundamental assumption that the asset spread will revert to their historical mean, suggesting that shocks to the spread are permanent rather than temporary, violating the stationarity assumption underlying the pairs trading model. The high volatility observed in the cryptocurrency market during this period may have contributed to this non-stationary behavior, as the estimated noise parameters (Q = 467.28, R = 66.11) were substantially high.

The superior performance of the EM-optimized filter, despite the absence of mean reversion, indicates that the Kalman filter's value lies primarily in noise reduction rather than capturing mean-reverting dynamics. The dramatically higher estimated noise parameters (Q = 467.28, R = 66.11) compared to defaults highlight the importance of proper parameter calibration in noisy financial markets.

From a practical perspective, this study demonstrates that parameter optimization is crucial when implementing filtering-based trading strategies. The 43% improvement in returns achieved through EM optimization represents a substantial enhancement that could significantly impact real trading performance. However, the absence of mean reversion raises questions about the sustainability of such strategies and emphasizes the need for careful model validation.

The study has several limitations that should be acknowledged. The analysis focused on a single cryptocurrency pair over a limited time period, which may not generalize to other assets or market conditions. Additionally, the backtesting did not account for realistic trading costs, bid-ask spreads, or market impact, which could reduce actual returns. The perfect win rate achieved across all strategies suggests the analysis may not fully capture real-world trading challenges.

For further research, other methods for parameter optimization for Kalman filtering could be considered. In the literature, a common approach is to fit an autoregressive model of order one (AR(1)) to the historical spread. An AR(1) process assumes that the current value of a time series depends linearly on its immediately preceding value, plus a random noise term. This makes it a simple tool to capture mean-reverting behavior in financial time series. Fitting such a model provides estimates for the state transition coefficient and the process noise variance. For example, Elliott et al. [5] and Gatev et al. [6] demonstrate that fitting a simple autoregressive model allows for more accurate initialization of the mean-reversion coefficient ρ and the process noise variance σ_η^2.

Moreover, regarding the trading strategy could be expanded by testing multiple asset pairs across different markets and time periods, incorporating realistic trading constraints, and exploring alternative filtering techniques or signal generation methods. Additionally, investigating methods to explicitly enforce mean reversion in state-space models or examining the relationship between market conditions and filtering effectiveness could provide valuable insights.

In conclusion, this study demonstrates that Kalman filtering can be a valuable tool for pairs trading when parameters are properly estimated through techniques like the EM algorithm. While the observed spread did not exhibit the mean-reverting behavior typically assumed in pairs trading literature, the filtering approach still provided improved performance through effective noise reduction. The findings suggest that careful parameter estimation is essential for implementing filtering-based trading strategies, and highlight the importance of validating model assumptions against empirical data. This work provides a practical example of how state-space models can be applied to financial time series, contributing to the broader understanding of filtering techniques in quantitative finance applications.

References

[1] Richard M Bookstaber. “A demon of our own design: Markets, hedge funds, and the perils of financial innovation”. In: *(No Title)* (2007).

[2] Arthur P Dempster, Nan M Laird, and Donald B Rubin. “Maximum likelihood from incomplete data via the EM algorithm”. In: *Journal of the royal statistical society: series B (methodological)* 39.1 (1977), pp. 1–22.

[3] Binh Do, Robert Faff, and Kais Hamza. “A new approach to modeling and estimation for pairs trading”. In: *Proceedings of 2006 financial management association European conference*. Vol. 1. 2006, pp. 87–99.

[4] Douglas S Ehrman. *The handbook of pairs trading: strategies using equities, options, and futures*. John Wiley & Sons, 2006.

[5] Robert J Elliott, John Van Der Hoek*, and William P Malcolm. “Pairs trading”. In: *Quantitative Finance* 5.3 (2005), pp. 271–276.

[6] Evan Gatev, William N Goetzmann, and K Geert Rouwenhorst. “Pairs trading: Performance of a relative-value arbitrage rule”. In: *The Review of Financial Studies* 19.3 (2006), pp. 797–827.

[7] Andrew C. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, 1990.

[8] Rudolph Emil Kalman. “A new approach to linear filtering and prediction problems”. In: (1960).

[9] Geoffrey J McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*. John Wiley & Sons, 2008.

[10] Roger D Peng. *A very short course on time series analysis*. 2020.

[11] TG Saji. “Pairs trading in cryptocurrency market: A long-short story”. In: *Available at SSRN 5135627* (2021).

[12] Kevin Manhau Wong. “Pairs trading-A quantitative approach”. In: (2015).

[13] Fenghui Yu et al. “Optimal pairs trading strategies: A stochastic mean–variance approach”. In: *Journal of Optimization Theory and Applications* 196.1 (2023), pp. 36–55.