Politecnico di Milano GSoM

Master's Degree in Business Analytics and Data Science

Machine Learning and Stochastic Filtering Techniques for Statistical Arbitrage

Supervisor:
Prof. Rocco Mosconi

Giovanni Piva

Academic Year 2024–2025

Abstract

In this thesis, a quantitative statistical arbitrage strategy based on pairs trading is developed and tested.

The methodology primarily integrates two advanced tools: the Kalman filter, employed to dynamically estimate the coefficients of the linear relationship between the two assets, and Hidden Markov Models, used to model market regimes and manage risk.

The aim is not to propose a strategy capable of "beating the market" or generating extraordinary profits, but rather to outline a rigorous methodological framework for the design, estimation, and validation of advanced trading strategies, with particular emphasis on avoiding common biases. Indeed, many strategies presented in the literature or online often display seemingly exceptional results that are not replicable due to improper data usage or insufficient methodological robustness.

Contents

Introduction **1**

1 Theoretical Foundations **3**
1.1 The Ornstein–Uhlenbeck Process 3
1.2 Cointegration 5

2 Spread Estimation **8**
2.1 The Kalman Filter 8
2.1.1 State equation: 9
2.1.2 Observation equation: 9
2.1.3 Steps 10
2.2 The EM algorithm for estimating Q and R 12
2.3 Standardization of the Spread through Rolling Window 14

3 Handling Market Regimes **16**
3.1 The Hidden Markov Model Framework 17
3.1.1 Markov Chains 17
3.1.2 Hidden Markov Model 18
3.1.3 Viterbi Algorithm 19
3.2 Regimes Identification 20
3.2.1 Training 20
3.2.2 Results 21

4 The Strategy **23**
4.1 Trading Signals and Execution Rules 23
4.2 Backtest and Performance Evaluation 24

5 Conclusions **29**

Bibliography **30**

Introduction

Since the emergence of organized financial markets in the seventeenth century, arbitrage opportunities have played a central role in trading activities. From the Amsterdam Stock Exchange in 1602 to the modern era of electronic markets, the principle of the "Law of One Price" has guided both traders and economists: identical or closely related assets should not persistently trade at different prices.

During the twentieth century, the development of mathematical finance — from Bachelier's pioneering work (1900) to the breakthrough of the Black–Scholes model (1973) — provided the theoretical foundations for modern quantitative strategies. In the 1980s and 1990s, the advent of computing power and electronic trading fostered the rise of statistical arbitrage, with pairs trading standing out as one of the earliest and most influential techniques. This period also saw the emergence of major quantitative funds, such as Renaissance Technologies, D.E. Shaw, and Optiver, which continue to shape the industry today.

Pairs trading is one of the simplest forms of statistical arbitrage: it seeks to exploit, through long and short positions, temporary deviations in the relationship between two highly correlated assets in order to generate profits. While the underlying intuition is straightforward, the strategies and technologies employed today are highly sophisticated and studied by some of the most prominent minds in the field. At the same time, financial markets have become increasingly competitive and, at least in appearance, more efficient. According to the Efficient Market Hypothesis (Fama, 1970), asset prices should fully reflect all available information, leaving little room for consistent arbitrage opportunities. In practice, however, market frictions, behavioral biases, and structural inefficiencies still create windows of opportunity. The result is an ongoing "arms race", in which traders and institutions compete through ever more advanced strategies, faster execution systems, and increasingly sophisticated technologies.

This thesis builds upon that tradition by developing and testing a pairs trading strategy that integrates advanced probabilistic tools, in particular the Kalman filter and Hidden Markov Models, with the aim of designing a rigorous and methodologically robust framework. The contribution is not to propose a strategy capable of consistently "beating the market", but rather to emphasize the importance of rigorous statistical design in ensuring robustness and reliability, while avoiding common pitfalls such as look-ahead bias, data-snooping, and overfitting.

The empirical development initially focuses on a single pair of stocks, Visa and Mastercard, given their strong economic and business similarity. The analysis is conducted over a five-year period, from January 1, 2020, to January 1, 2025, a timeframe marked by significant events that deeply affected global financial markets, including the COVID-19 pandemic, rising inflation, and heightened volatility across asset classes. This setting provides an ideal testing ground for implementing the methodology and assessing its effectiveness under controlled conditions. Subsequently, the analysis is extended to additional case studies in order to evaluate the robustness and generalizability of the proposed approach. The results highlight how the integration of probabilistic modeling and machine learning techniques can, under certain conditions, enhance both the stability and the risk-adjusted performance of pairs trading strategies when compared to more traditional implementations.

The structure of the thesis is as follows. Chapter 1 introduces the theoretical foundations, focusing on mean reversion and cointegration. Chapter 2 presents spread estimation through the Kalman filter, a tool widely used in physics and engineering as well as in finance for handling noisy observations. Chapter 3 addresses the identification of market regimes using a specific machine learning approach: the Hidden Markov Model. The key idea is to distinguish between two regimes: a "safe", mean-reverting one, and a "high-volatility" regime, in which the relationship may break down or the Kalman filter estimation becomes unreliable. Finally, Chapter 4 presents the definition and implementation of the strategy, together with the backtesting procedure and the performance evaluation. The purpose of this last chapter is mainly to demonstrate the correct functioning of the models and the overall strategy, rather than to focus on the absolute value of the results obtained.

The overall research process and strategy design can be summarized in the following steps:

- **Pair(s) selection** based on economic similarity, correlation, and cointegration tests.
- **Spread estimation** using the Kalman filter to dynamically model the relationship between the assets.
- **Spread standardization** through a rolling-window procedure.
- **Regime identification** with Hidden Markov Models to distinguish mean-reverting phases from volatile or unreliable periods.
- **Trading rule definition** based on the standardized spread (z-score) and regime classification.
- **Backtesting and performance evaluation** on historical data.

Chapter 1

Theoretical Foundations

In this chapter, we present the theoretical foundations underlying pairs trading strategies. Although these concepts may appear more abstract than the subsequent applied sections, they are essential for correctly interpreting the results and understanding the rationale behind the proposed methodology.

We begin with the Ornstein–Uhlenbeck process, which formalizes the concept of mean reversion and provides the theoretical basis for analyzing the dynamics of the spread. Subsequently, we introduce the concept of cointegration, which offers the statistical framework to verify the long-term relationship between two time series and to identify the most suitable pairs of assets for the strategy.

1.1 The Ornstein–Uhlenbeck Process

The profitability of pairs trading relies on the assumption that the spread between two assets fluctuates around a long-term mean: when the relationship deviates from this equilibrium, it is expected to revert back to it. This behavior, known as mean reversion, can be formally described by the Ornstein–Uhlenbeck (OU) process.
The Ornstein–Uhlenbeck process, named after the Dutch physicists Leonard Ornstein and George Eugene Uhlenbeck, is a continuous-time stochastic process that describes a variable subject to random fluctuations while exhibiting a tendency to revert towards a long-term mean μ. It is defined by the following stochastic differential equation:

$$\mathrm{d}Z_t = \kappa(\mu - Z_t)\mathrm{d}t + \sigma \mathrm{d}B_t$$

where Z_t denotes the variable of interest (in this context, the spread), $\kappa > 0$ is the parameter governing the speed of mean reversion towards μ, $\sigma > 0$ measures the intensity of the random component, and B_t represents a standard Brownian motion (continuous white noise).
Intuitively, the OU equation captures the balance between two forces: a deterministic component that pulls the process back towards the mean μ with speed κ, and a stochastic component, scaled by σ, that introduces random fluctuations over time.

The Ornstein–Uhlenbeck SDE admits the explicit solution:

$$Z_t = \mu + (Z_0 - \mu)\, e^{-\kappa t} + \sigma \int_0^t e^{-\kappa(t-s)}\, \mathrm{d}B_s$$

From this expression, it follows that Z_t is a Gaussian process, being the sum of a deterministic component and a Gaussian stochastic integral. Specifically:

$$\mathbb{E}[Z_t] = \mu + (Z_0 - \mu)e^{-\kappa t}, \quad \mathrm{Var}(Z_t) = \frac{\sigma^2}{2\kappa}\big(1 - e^{-2\kappa t}\big)$$

As $t \to \infty$, the process converges in distribution to a stationary normal law:

$$\boxed{Z_t \sim \mathcal{N}\left(\mu,\ \frac{\sigma^2}{2\kappa}\right)}$$

This result shows that the spread does not diverge to infinity, as in the case of a random walk, but rather fluctuates around the long-term mean μ. Moreover, the stationary variance $\frac{\sigma^2}{2\kappa}$ provides a measure of the typical amplitude of these fluctuations. Finally, the fact that the spread is normally distributed in equilibrium allows it to be normalized and to define trading thresholds in probabilistic terms, in our case via the z-score.
The solution of the OU stochastic differential equation can be derived using Itô calculus and the method of variation of constants. The analytical derivation lies beyond the scope of this thesis; what matters for our purposes is that the solution reveals the Gaussian nature of the process, its mean-reverting behavior towards μ, and a variance that converges to a stationary value.

After computing the spread, we can estimate the three parameters κ, μ, and σ for our pair. Their estimates provide valuable insights into the spread dynamics. We use the discrete-time representation of the OU, which results in an AR(1) model:

$$Z_{t+\Delta t} \;=\; a \;+\; b\, Z_t \;+\; \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}\big(0,\, \sigma_\varepsilon^2\big)$$

with the continuous-to-discrete mapping:

$$b \;=\; e^{-\kappa \Delta t}, \qquad a \;=\; \mu\big(1 - e^{-\kappa \Delta t}\big), \qquad \sigma_\varepsilon^2 \;=\; \frac{\sigma^2}{2\kappa}\Big(1 - e^{-2\kappa \Delta t}\Big)$$

After estimating the AR(1) parameters $(a, b, \sigma_\varepsilon^2)$ via OLS, the OU parameters follow:

$$\kappa \;=\; -\frac{1}{\Delta t}\ln b, \qquad \mu \;=\; \frac{a}{1-b}, \qquad \sigma \;=\; \sqrt{\frac{2\kappa\, \sigma_\varepsilon^2}{1 - e^{-2\kappa \Delta t}}}$$

The stationary (long-run) volatility of the process is:

$$\sigma_{\text{stat}} \;=\; \sqrt{\frac{\sigma^2}{2\kappa}}$$

Finally, the half-life of mean reversion—the average time for the spread to cover half the distance back to the mean after a shock—is:

$$t_{1/2} = \frac{\ln 2}{\kappa}$$

expressed in the same time units as Δt. Practical note: for daily data, $\Delta t = 1$ day; for other sampling frequencies, adjust Δt accordingly and interpret κ and $t_{1/2}$ in those units.

It is crucial to stress that the estimation of these parameters strongly depends on how the spread is defined. In our analysis, the parameters were computed both using the spread obtained from an OLS regression (which implicitly incorporates future information, and was indeed used solely for exploratory purposes), and the spread estimated dynamically through the Kalman filter, leading to markedly different results. For example, the half-life estimated with the OLS spread turned out to be around 20 days, whereas the one estimated using the dynamic spread from the Kalman Filter was just one day. The reasons behind these discrepancies are discussed in detail in the chapter on the Kalman filter.
Therefore, the estimation of OU parameters should be regarded mainly as an exploratory tool or as a quick benchmark for comparing pairs, rather than as a basis for direct trading decisions. This highlights that the OU process serves primarily as a theoretical foundation to motivate the strategy, while its direct parameter estimation plays a secondary, exploratory role.

While the Ornstein–Uhlenbeck process provides a theoretical framework for describing mean reversion, in practice we need statistical tools to verify whether such a property actually holds for a given pair of assets. Cointegration offers precisely this: it allows us to test whether two non-stationary series exhibit a stable long-term relationship, making the spread suitable to be modeled as a mean-reverting process.

1.2 Cointegration

The following chart shows the price dynamics of our selected pair, Visa and Mastercard, over the period 2020–2025:

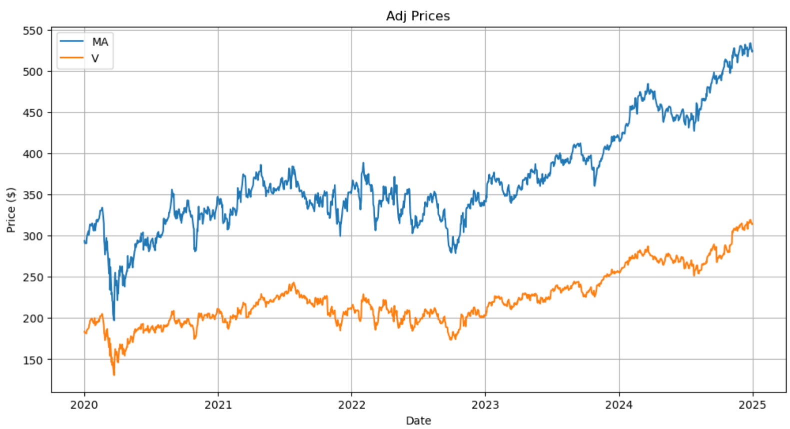


Figure 1.1: Adjusted Prices of Visa and Mastercard

After computing the linear correlation, which in this case amounts to 98.3%, we apply the Engle–Granger test to verify whether the relationship between Mastercard and Visa remains stable in the long run.
In this type of analysis, both prices and log-prices can be used, without substantially affecting the results. In literature, both approaches are found; given the scale similarity of Visa and Mastercard and the strong evidence of cointegration, we opted for raw prices for simplicity and interpretability, without loss of generality.

Cointegration between two non-stationary series is defined as the existence of a linear combination that is stationary. In other words, although both series may be integrated (for instance of order 1), their linear combination can reveal a long-run equilibrium relationship.

Formally, we consider the relationship:

$$y_t = \alpha + \beta x_t + \varepsilon_t$$

where Y_t denotes Mastercard prices and X_t Visa prices. The objective is to verify that the residuals ε_t (i.e., the spread) are stationary, using a unit root test (ADF test).
The ADF is a unit root test that assesses the null hypothesis that a time series contains a unit root (i.e., is non-stationary) against the alternative that it is stationary. In practice, one estimates the regression:

$$\Delta\varepsilon_t = \rho\,\varepsilon_{t-1} + \sum_{i=1}^{p} \phi_i\,\Delta\varepsilon_{t-i} + u_t$$

where $\Delta\varepsilon_t = \varepsilon_t - \varepsilon_{t-1}$. The lag length p is typically selected using information criteria such as AIC or BIC, or through sequential testing, in order to ensure that the residuals u_t are not serially correlated. In our case, the adfuller procedure automatically selected 10 optimal lags according to the AIC criterion.

The null hypothesis is $H_0 : \rho = 0$ (unit root), while the alternative is $H_1 : \rho < 0$ (stationarity). The decision is based on comparing the test statistic with critical values tabulated by Dickey and Fuller; if the statistic is lower than the critical value, or equivalently if the p-value is below the chosen significance level (commonly 5%), the null hypothesis is rejected.

In the context of the Engle–Granger procedure, the ADF is applied to the residuals $\varepsilon_t = Y_t - (\alpha + \beta X_t)$. If these residuals are found to be stationary, then Y and X are cointegrated, implying the existence of a valid long-run equilibrium relationship.

The results are as follows:

- **ADF Statistic:** -4.687
- **p-value:** 8.9×10^{-5}
- **Critical Values:** $1\% : -3.44;\ 5\% : -2.86;\ 10\% : -2.57$

Since the p-value is well below the 5% significance threshold, we reject the null hypothesis of non-stationarity of the residuals, thereby confirming the presence of

cointegration between the two series.
Methodological note: the critical values reported by statsmodels are the standard ADF ones, not the specific Engle–Granger values (which are usually more negative). Nevertheless, our statistic (–4.687) is well below even the 1% threshold, thus confirming cointegration.

The Visa–Mastercard pair is therefore suitable for the application of the strategy and is, not surprisingly, widely employed in the literature and in empirical studies on this class of models.
An important remark is that, at this stage, the cointegration test has been performed on the spread estimated using the entire time window, via OLS (with fixed α and β parameters). In fact, the purpose of this section is solely to validate the suitability of the pair for the strategy.
In the subsequent analysis, the spread will be estimated dynamically and recursively, on a day-by-day basis, using the Kalman filter, in order to avoid look-ahead bias.

Chapter 2

Spread Estimation

In this chapter, we focus on the calculation of the spread between the two assets. As previously mentioned, the estimation is carried out dynamically, updating the alpha and beta parameters on a daily basis using only the information available up to the previous day. To achieve this, we employ a Kalman Filter, a model that balances the reliance on the current estimate with the incorporation of noisy new observations. Intuitively, the filter answers the question: "How much should I trust my estimate compared to the actual measurement?" We first estimate the key parameters of the filter, in particular the Q and R matrices, using a training window; the filter is then applied to provide daily estimates of the spread. Finally, a rolling window is used to standardize the spread and compute the z-score, which serves as the basis for generating trading signals.

2.1 The Kalman Filter

The Kalman Filter is a widely used tool in engineering and applied sciences for balancing the reliance on model-based estimates with new noisy observations. A classical example often used to illustrate its functioning is GPS localization: if a car is moving at a constant speed along a straight line, the filter combines the theoretical prediction of its position with the noisy measurement provided by the GPS after one hour. The result is a more accurate estimate than relying solely on either the model or the observation.
In our setting, rather than estimating position and velocity, the filter is applied to dynamically estimate the parameters α_t and β_t, which capture the linear relationship between the two assets. The spread is therefore defined as:

$$\epsilon_t = y_t - (\alpha_t + \beta_t x_t)$$

In our example:

$$\text{spread}_t = \text{MA}_t - (\alpha_t + \beta_t \text{V}_t)$$

The Kalman Filter is based on a linear dynamic model with Gaussian noise. It is defined by two components: the state equation and the observation equation.

2.1.1 State equation:

The state equation describes the time evolution of the unobserved parameters α_t and β_t, which follow a stochastic process perturbed by noise:

$$\boxed{s_t = As_{t-1} + w_t}$$

where s_t denotes the hidden state of the system, defined as:

$$s_t = \begin{bmatrix} \alpha_t \\ \beta_t \end{bmatrix}$$

These parameters have a clear practical meaning: α_t is the intercept, expressed in price units (dollars), capturing the level differences between the two assets; β_t is the hedge ratio, so the number of units of X required to hedge one unit of Y. These two parameters are not only essential for constructing the spread, but will also be used later to determine the actual trading quantities of Mastercard and Visa.
The A matrix is the transition matrix, which describes how the system evolves from one step to the next, while $w_t \sim N(0, Q)$ represents Gaussian noise with covariance matrix Q.
In our case, the transition matrix is chosen as the identity matrix:

$$A = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

This assumption implies that the evolution of α_t and β_t over time is driven exclusively by noise, with no drift term. In other words, we model α_t and β_t as following a random walk: the two parameters remain constant from one period to the next, except for small random fluctuations.

The covariance matrix of the noise term w_t is square and can be heuristically defined as: $Q = (1 - \delta)I, \quad \delta \in (0, 1)$.
The parameter δ governs the dynamics of the coefficients α_t and β_t: the closer δ is to 1, the more persistent the parameters remain over time, making the model behave similarly to an OLS regression. In fact, when $Q = 0$, the coefficients are fixed, and the Kalman Filter reduces exactly to an OLS estimate.
Nevertheless, relying solely on a heuristic specification of Q may be restrictive. To achieve a more data-driven and realistic estimate, we implement the Expectation-Maximization (EM) algorithm, which will be presented in the following section.

2.1.2 Observation equation:

The observation equation specifies the link between the estimated parameters and the observed prices: at each point in time, the price of Y is explained by $\alpha_t + \beta_t X_t$, plus a noise term accounting for market and measurement imperfections:

$$\boxed{y_t = Hs_t + v_t}$$

where H is the observation matrix and $v_t \sim \mathcal{N}(0, R)$ denotes the observation noise. The magnitude of R controls the weight given to the observation relative to the prediction: a large R indicates very noisy observations, leading the filter to rely more heavily on the model estimate rather than the observed data.

In our setting, since only one variable is observed—the asset price—the matrix H is of dimension 1×2 and takes the form:

$$H_t = \begin{bmatrix} 1 & x_t \end{bmatrix}$$

so that the observed value depends linearly on α_t and β_t. More specifically:

$$y_t = Hs_t + v_t = \begin{bmatrix} 1 & x_t \end{bmatrix} \begin{bmatrix} \alpha_t \\ \beta_t \end{bmatrix} + v_t = \alpha_t + \beta_t x_t + v_t$$

As with Q, we opted to use the Expectation-Maximization (EM) algorithm to estimate R directly from the data, rather than specifying it heuristically.

It is crucial to note that, in our case, the measurement of the asset price y_t is not subject to instrumental errors, unlike the GPS example. Nevertheless, we do not set $R = 0$. Doing so would cause the filter to rely exclusively on the observed values, resulting in excessively abrupt parameter updates at each new observation. This would make the estimates unstable and the model considerably less robust.

The key idea is that observed prices are "contaminated" by undesirable components such as market microstructure effects, bid-ask spread, noise trading, or idiosyncratic events. In contrast, the "true value" of the underlying process evolves more smoothly. For this reason, we treat the price measurement as noisy and include a nonzero R, even though the data appear to be precise.

2.1.3 Steps

The operation of the Kalman Filter is structured around two main iterative phases: the prediction step and the update step.

1. **Prediction (prior)**
   The state at time t is predicted using information available up to $t - 1$:

   $$\hat{s}_{t|t-1} = A\hat{s}_{t-1|t-1}$$

   with predicted covariance:

   $$P_{t|t-1} = AP_{t-1|t-1}A^\top + Q$$

   In this step, the filter "imagines" how the parameters α_t and β_t should evolve, relying solely on the model and past estimates.

2. **Update (posterior)**
   Once y_t is observed, the estimate is corrected with the new information:

   - **Kalman gain**:

   $$K_t = \frac{P_{t|t-1}H_t^\top}{H_t P_{t|t-1} H_t^\top + R}$$

- **State update**:
$$\hat{s}_{t|t} = \hat{s}_{t|t-1} + K_t\big(y_t - H_t\hat{s}_{t|t-1}\big)$$
- **Covariance update**:
$$P_{t|t} = (I - K_tH_t)P_{t|t-1}$$

Once a new observation arrives, the filter compares it with the prediction and corrects the parameters, striking a balance between trusting the model and trusting the data.

The balance between prediction and measurement is therefore determined by the Kalman gain, which specifies how much the estimate should be corrected based on the new observation. Intuitively:

- if R is small (very reliable measurement), the filter strongly relies on the new observation;
- if the prediction uncertainty, $P_{t|t-1}$, is large, the filter also gives more weight to the measurement, since the prior estimate is less trustworthy.

Thus, the Kalman gain acts as an adaptive weight that dynamically balances trust between the model and the observed data: when it is close to 1, the filter relies heavily on new observations, while values close to 0 indicate greater reliance on the model's prior estimate.
The updated state estimate $\hat{s}_{t|t}$ and its covariance $P_{t|t}$ then serve as the new prior for the next iteration, ensuring that the filter continuously refines its predictions as new data become available.

Example. Suppose that at time t the filter predicts $\alpha_t = 3.0$. The new observation indicates a value of 3.2, yielding a prediction error of $+0.2$. With a Kalman gain of $K_t = 0.3$, the correction amounts to $0.2 \times 0.3 = 0.06$. The updated estimate is therefore: $\alpha_t = 3.0 + 0.06 = 3.06$, a compromise between the prediction and the noisy observation.
Suppose further that the prior uncertainty of the estimate is $P_{t|t-1} = 0.5$, with $H_t = 1$ (for simplicity, in this example we are considering H_t as a scalar). In this case, the covariance update is: $P_{t|t} = (1 - 0.3) \times 0.5 = 0.35$. Thus, in addition to correcting the state from 3.0 to 3.06, the filter also reduces the associated uncertainty, from 0.5 down to 0.35.

Here are the results of the estimation:

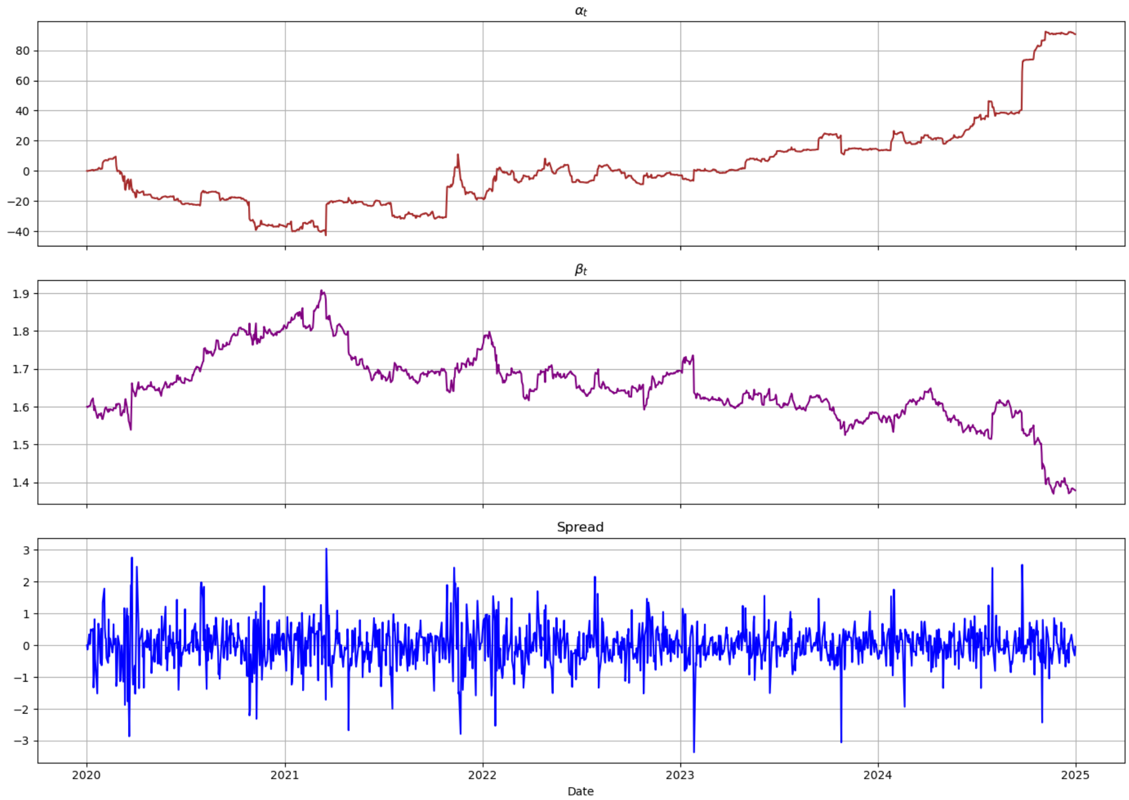


Figure 2.1: Parameters and Spread estimation

2.2 The EM algorithm for estimating Q and R

As anticipated, in order to achieve more accurate results, we estimate the matrices Q and R using the Expectation–Maximization (EM) algorithm, as these matrices play a fundamental role in shaping the dynamics of the filter and strongly influence the outcomes.
To avoid making the project overly complex, the estimation is performed only once on a training window, after which the parameters remain fixed throughout the analysis. Nevertheless, there exist extensions of the Kalman Filter, such as the adaptive Kalman filter, in which Q and R are periodically re-estimated or updated in real time, allowing the model to adapt to possible structural changes in the data.
The estimation of Q and R remains an active area of research and is of considerable practical relevance, particularly in the financial sector, where banks and large funds devote significant effort to improving the robustness of forecasting and risk management models.

As previously seen, the Kalman Filter process is defined by the following state-space model:

$$\begin{aligned} s_t &= A s_{t-1} + w_t, \quad w_t \sim \mathcal{N}(0, Q) \\ y_t &= H_t s_t + v_t, \quad v_t \sim \mathcal{N}(0, R) \end{aligned}$$

The covariance matrices Q and R are specified as:

$$Q = \begin{bmatrix} \mathrm{Var}(\alpha_t) & \mathrm{Cov}(\alpha_t, \beta_t) \\ \mathrm{Cov}(\beta_t, \alpha_t) & \mathrm{Var}(\beta_t) \end{bmatrix}, \qquad R = \begin{bmatrix} \sigma^2_{\mathrm{obs}} \end{bmatrix}$$

Their interpretation is as follows:

- If Q is small, the parameters α_t and β_t change little over time, making the model similar to a static OLS regression.
- If Q is large, the parameters become more dynamic, autocorrelation decreases, and the estimated spread differs significantly from the OLS one.
- If R is small, the filter strongly relies on the observed data, adapting quickly to new information.
- If R is large, the observations are treated as highly noisy, so the filter relies more on the model forecast, leading to slower updates.

The estimation of these parameters, $\theta = \{Q, R\}$, aims at maximizing the likelihood of observing the series $\{y_t\}_{t=1}^{T}$. Formally, the log-likelihood is:

$$\ell(\theta) = \sum_{t=1}^{T} \log f(y_t \mid y_{1:t-1}, \theta)$$

where $f(y_t \mid y_{1:t-1}, \theta)$ denotes the predictive density produced by the filter at time t. Since this optimization problem does not admit a closed-form solution, the estimation is carried out using the Expectation–Maximization (EM) algorithm, which iteratively updates Q and R until convergence.

The EM algorithm consists of two main phases:

- Expectation step (E-step): the Kalman smoother, that exploits the entire sequence of available observations (unlike the filter), is applied over the training window to estimate the latent parameters α_t and β_t, their covariance, and cross-covariances, given the current values of Q and R.
- Maximization step (M-step): Q and R are updated to maximize the expected log-likelihood obtained in the E-step. Concretely, they are updated as average variances of the estimated residuals:

$$Q = \frac{1}{T-1} \sum_{t=2}^{T} (s_t - A s_{t-1})(s_t - A s_{t-1})^\top$$

$$R = \frac{1}{T} \sum_{t=1}^{T} (y_t - H_t s_t)(y_t - H_t s_t)^\top$$

Expanding with the smoothed estimates:

$$Q^{(t+1)} = \frac{1}{T-1}\sum_{t=2}^{T}\left(P_{t|T} - AP_{t,t-1}^{\top} - P_{t,t-1}A^{\top} + AP_{t-1|T}A^{\top}\right)$$

$$R^{(t+1)} = \frac{1}{T}\sum_{t=1}^{T}\left((y_t - H_t\hat{s}_t)(y_t - H_t\hat{s}_t)^{\top} + H_t P_{t|T} H_t^{\top}\right)$$

Intuitively, Q is the average variance of the state difference $\Delta s_t = s_t - s_{t-1}$: if the states evolve smoothly, Q is small; if they undergo sudden unpredictable changes, Q grows. Conversely, R is the average variance of the observation error, which combines the visible part (squared residuals) plus the uncertainty carried by the latent state.

These two steps are repeated iteratively until the log-likelihood converges, so that the final estimates of Q and R maximize the probability of the observed data.

In our example, the EM algorithm yielded the following covariance estimates:

$$Q_{\text{MA-V}} = \begin{bmatrix} 1.6279 & -0.0019 \\ -0.0019 & 0.0001 \end{bmatrix}, \qquad R_{\text{MA-V}} = 2.28$$

The variance of α_t is relatively high, suggesting that this parameter changes significantly over time. In contrast, β_t exhibits a very small variance, indicating much greater stability.
This result suggests that, while the intercept may adjust to transient shocks and short-term fluctuations, the slope remains essentially stable, reflecting the robustness of the long-term relationship between the two assets.

The EM algorithm will be also used in the Hidden Markov model section for regimes identification.

2.3 Standardization of the Spread through Rolling Window

Once the spread has been estimated, it must be standardized in order to obtain the z-score, which serves as the normalized measure used to define trading signals.

The standardization procedure relies on a rolling window of length N. For each day t, the moving average μ_t and the moving standard deviation σ_t of the spread over the past N days are computed. The z-score is then given by:

$$z_t = \frac{\epsilon_t - \mu_t}{\sigma_t}$$

This transformation converts the spread into a dimensionless variable, enabling the detection of significant deviations from its recent behavior.
As for the choice of N, the literature does not provide a universally optimal value.

A common criterion is to set N as a multiple of the estimated half-life of the mean-reverting process, typically between two and four times. In our case, we opted for a 60-day window; however, this choice did not have a significant impact on the final results.

The figure 3.1 shows a comparison between the spread computed via OLS and that obtained through the Kalman Filter (along with their standardized versions). A marked difference can be clearly observed between the two, which is entirely expected. The adaptive parameters estimated by the Kalman Filter, α_t and β_t, dynamically capture much of the variability that, in the OLS approach, remains embedded in the residuals. With OLS, the parameters are fixed and do not adapt to changes over time, causing the spread to display wider fluctuations.

For the purposes of the strategy, what matters is not the absolute value of the spread, but its standardized version (z-score), which is crucial for identifying statistically significant deviations. The goal is therefore not to replicate the OLS spread (which is itself estimated using future information), but rather to obtain a dynamic measure that can be employed in real time.

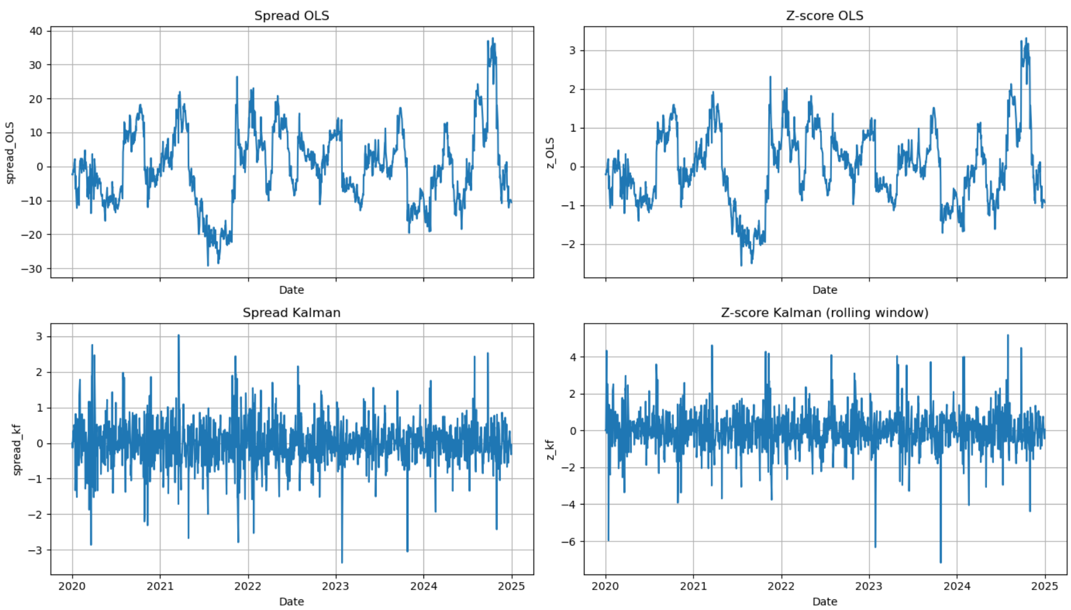


Figure 2.2: Comparison of the spreads

In summary, the dynamic estimation of the spread through the Kalman Filter, combined with its standardization into the z-score, provides the operational basis for generating trading signals. At the same time, it is essential to determine the prevailing market regime, a task carried out using the Hidden Markov Model, which will be introduced in the next chapter.

Chapter 3

Handling Market Regimes

As previously mentioned, pairs trading relies on the assumption that the spread between assets is both stationary and mean-reverting. Under these conditions, temporary deviations represent profitable opportunities because convergence back to equilibrium is expected. However, empirical evidence shows that financial markets are not governed by a single, stable regime. Instead, the statistical properties of spreads vary over time, often due to structural breaks, liquidity shocks, or macroeconomic events.

As Fanelli, Fontana, and Rotondi (2024) emphasize in their study of crude oil futures, "the cointegration vector depends significantly on the time window over which the relationship is estimated" and structural breaks may undermine the stability of the equilibrium relation. This directly challenges the core assumption of pairs trading: that the spread reliably mean-reverts. If trades are initiated during periods of instability, what appears as a short-term mispricing may instead reflect a fundamental regime shift, leading to losses rather than arbitrage profits.

Fanelli et al. (2024) show that accounting for regime shifts is crucial when designing statistical arbitrage strategies. Their study highlights that spreads may behave differently across time, alternating between phases in which the mean-reversion property is reliable and phases of instability or heightened volatility where trading signals are less trustworthy. Strategies that explicitly distinguish between such regimes were found to perform significantly better than those based on static assumptions about spread dynamics.

In the context of this thesis, the motivation for distinguishing regimes is mainly one of risk management. By identifying when the spread dynamics are stable and when they are not, the strategy can avoid entering positions in periods where the underlying assumptions fail. This filtering role reduces exposure to adverse conditions and improves robustness, ensuring that trades are initiated only when market behavior is consistent with the statistical arbitrage framework. In the following section, we introduce hidden Markov models as a practical tool for capturing and classifying these latent regimes.

A natural question arises: why use a Hidden Markov Model instead of simply defining two regimes through a fixed z-score threshold, such as "stable" for $|z_t| < 1$ and "risky" for $|z_t| \geq 1$?

While intuitive, this approach treats each observation independently and ignores the temporal structure of the data. Financial regimes, however, tend to persist over time, and a threshold-based rule would cause frequent and unrealistic regime switches driven by random fluctuations in the z-score. Moreover, a fixed cutoff provides no probabilistic interpretation, as it cannot quantify the likelihood of remaining in or transitioning between regimes, nor can it adapt to changes in volatility.

3.1 The Hidden Markov Model Framework

3.1.1 Markov Chains

A Markov chain is a stochastic process $(X_t)_{t\geq 0}$ that evolves through a finite set of states $\mathcal{S} = \{0, 1, ..., N-1\}$, where the probability of transitioning to the next state depends only on the current state, i.e. the states before the current state have no impact on the future except via the current state. It's as if to predict tomorrow's weather you could examine today's weather but you weren't allowed to look at yesterday's weather.

Formally, the Markov property is:

$$\mathbb{P}(X_{t+1} = j \mid X_0, X_1, \ldots, X_t) = \mathbb{P}(X_{t+1} = j \mid X_t) \quad \forall j \in \mathcal{S} \tag{3.1}$$

From now on, we will refer to the entire series of a random variable X from time 0 to time t as:

$$X_{0:t} := (X_0, X_1, \ldots, X_t)$$

The dynamics of the transitions themselves are described by a transition probability matrix:

$$P = [p_{ij}], \quad \text{where } p_{ij} = \mathbb{P}(X_{t+1} = j | X_t = i), \quad \forall i, j \in \mathcal{S}$$

with each row summing to one:

$$\sum_{j=0}^{N-1} p_{ij} = 1$$

and p_{ij} represents the probability of transitioning from state i to state j.

This formulation allows us to compute the probability of being in any state after k steps via:

$$\mathbb{P}(X_{t+k} = j | X_t = i) = (P^k)_{ij}$$

In application to our research, the states i are interpreted as market regimes (e.g., "stable" or "volatile"), and the transition probabilities describe how persistent each regime is. For example, the expected number of consecutive periods spent in regime i is $\mathbb{E}[T_i] = \frac{1}{1-p_{ii}}$, where T_i is the sojourn length. For example, if $p_{00} = 0.9$, this equals 10 periods (Hamilton, 1989).

To visualize better the transition probability matrix, consider:

$$P = \begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix}$$

$$\text{with} \quad p_{00} + p_{01} = 1, \quad p_{10} + p_{11} = 1$$

Furthermore, the definition we have for the transition probability p_{ij} is a generalized version, as it satisfies both time-homogeneous (the transition probability depends only on the current state X_t) and time-inhomogeneous (the transition probability is also a function of time t) Markov chains. In this project we will exclusively utilize time-homogeneous Markov chains, therefore, in the homogeneous case the transition probabilities are time-invariant:

$$\mathbb{P}(X_{t+1} = j | X_t = i) = p_{ij}, \quad \forall t \geq 0$$

3.1.2 Hidden Markov Model

In many real-world situations, the state of the system is not directly observable. For example, in our case, we cannot directly observe whether the market is in a stable or volatile regime: we can only observe quantities such as prices, returns, or spreads. This motivates the use of a Hidden Markov Model (HMM).

An HMM extends the idea of a Markov chain by introducing two layers:

1. **Hidden states** (X_t): these evolve according to a Markov chain, exactly as before. They represent the underlying regimes (e.g., stable vs. volatile) that we cannot directly observe.

2. **Observations** (Z_t): at each time step we observe some data point Z_t, which is generated according to a probability distribution that depends on the hidden state X_t. In our case observations correspond to the z-score: $Z_t = z_t$.

Formally, the hidden states evolve according to the same transition matrix introduced earlier:

$$\mathbb{P}(X_{t+1} = j \mid X_t = i) = p_{ij}, \quad \forall i, j \in \mathcal{S}, \quad \sum_{j=0}^{N-1} p_{ij} = 1.$$

Conditional on the hidden state $X_t = i$, the observation Z_t is drawn from an emission distribution:

$$Z_t \mid X_t = i \sim f_i(\cdot\,; \theta_i),$$

where f_i denotes the emission density (or probability mass function), parameterized by θ_i and associated with state i.

In this work, we specialize to Gaussian emissions:

$$Z_t \mid X_t = i \sim \mathcal{N}(\mu_i, \sigma_i^2),$$

where each hidden state i is associated with its own mean μ_i and variance σ_i^2.

For example:

- In a **low-volatility regime**, Z_t might be drawn from a normal distribution with small variance.

- In a **high-volatility regime**, Z_t could come from a normal distribution with much larger variance.

Thus, the hidden state controls the statistical behavior of the observed data.

To build intuition, consider the classic analogy of weather conditions (hidden states) that follow a Markov chain: sunny vs. rainy. You cannot directly observe the weather, but you can observe whether people carry umbrellas (observations). If many people carry umbrellas, it is more likely that the hidden state is 'rainy.' In the same way, in financial markets, we cannot observe the regime directly but we can infer it from the distributional properties of returns or spreads.

3.1.3 Viterbi Algorithm

The Viterbi algorithm (Viterbi, 1967) is a dynamic programming method used to determine the most probable sequence of hidden states given a sequence of observations. In the context of this work, it identifies the most likely market regime (stable or risky) at each point in time, conditional on the observed z-scores.

Formally, the algorithm finds the sequence of hidden states $(X_1, X_2, \ldots, X_T)$ that maximizes the joint probability of the observations and state transitions:

$$\boxed{\arg\max_{X_1, X_2, \ldots, X_T} \prod_{t=1}^{T} p(Z_t \mid X_t)\, p(X_t \mid X_{t-1})}$$

where $p(Z_t \mid X_t)$ is the emission probability (the likelihood of observing Z_t given state X_t), and $p(X_t \mid X_{t-1})$ is the transition probability between consecutive states.

The algorithm proceeds recursively:

1. **Initialization:** set the starting probabilities for each state given the first observation Z_1;
2. **Recursion:** for each time t, compute the most probable path ending in each state using the transition and emission probabilities;
3. **Backtracking:** once the recursion is complete, reconstruct the most probable sequence of hidden states.

This approach is computationally efficient, avoiding the need to evaluate all possible state paths (which would grow exponentially with T). In our case, it allows for real-time regime classification while preventing look-ahead bias, since only information up to time t is used in estimating the current regime.

In simple terms, the Viterbi algorithm answers the question: *"Given what I have observed (the z-scores), what is the most likely sequence of hidden states?"* For each time step, the algorithm computes:

- "If I came from state 0, what is the total probability of being here now?"
- "If I came from state 1, what is the total probability of being here now?"

and keeps only the most probable path at each step. Through the backtracking phase, it then reconstructs the entire most likely sequence of hidden states. However, since in our implementation the Viterbi algorithm is applied on a rolling window on a day-by-day basis, we only retain the final inferred state of each window — that is, the regime estimated for the most recent day.

3.2 Regimes Identification

3.2.1 Training

Now that the intuition behind HMMs is clear, we formalize how they are used in our strategy. The model assumes that the observed time series $(z_{0:T})$, representing the z-scores of the spread estimated by the Kalman filter, is generated by an unobserved hidden state process (X_t).
Each hidden state i corresponds to a regime with its own statistical properties, captured through a Gaussian distribution with mean μ_i and variance σ_i^2:

$$z_t \mid (X_t = i) \sim \mathcal{N}(\mu_i, \sigma_i^2).$$

We estimate the HMM on an initial training window of 315 days. Estimation is performed using the Expectation-Maximization (EM) algorithm, which iteratively adjusts both the transition probabilities p_{ij} and the Gaussian emission parameters (μ_i, σ_i^2) to maximize the likelihood of the observed sequence of z-scores:

$$\ell(\theta) = \log \mathbb{P}(z_{0:T} \mid \theta), \quad \theta = (P, \mu_i, \sigma_i^2).$$

The model is trained only once on the initial 315-day window, after which the parameters are kept fixed for the entire out-of-sample period.
After training, the model provides:

- The transition probability matrix $P = [p_{ij}]$.
- The means and standard deviations (μ_i, σ_i) associated with each hidden state.
- A mapping between hidden states and observed volatility levels.

We then classify the two hidden states by comparing their estimated variances: the state with the lower variance is interpreted as the stable regime (mean-reverting), while the state with the higher variance corresponds to the risky regime.

Inferring the regime in real time

At each time t, we feed the model the observed sequence $z_{0:t}$. The model applies the Viterbi algorithm, which finds the most likely sequence of hidden states $(\hat{X}_0, \ldots, \hat{X}_t)$. The last element $\hat{X}_t$ is then taken as the regime estimate for today.

This procedure ensures that no look-ahead bias is introduced: only past and present data are used when inferring today's regime. Formally, we define the regime indicator R_t as

$$R_t = \begin{cases} 0 & \text{if the inferred state } \hat{X}_t \text{ corresponds to the low-variance (stable) regime,} \\ 1 & \text{if the inferred state } \hat{X}_t \text{ corresponds to the high-variance (risky) regime.} \end{cases} \tag{3.2}$$

Example. Suppose we are at day $t = 400$ of the time series, and we observe a standardized spread value (z-score) of $z_t = +0.45$, i.e., close to the mean. The HMM, trained on the initial 315-day window, has already learned:

- the transition probability matrix P;
- the Gaussian emission parameters (μ_i, σ_i^2) for each hidden state, estimated via the Baum–Welch (EM) algorithm.

These parameters remain fixed throughout the out-of-sample period. At time $t = 400$, the model uses a rolling window of the most recent 316 observations (from day 85 to 400, today included) and applies the Viterbi algorithm to this segment. By combining the information from the observed z-scores and the transition probabilities, it finds the most probable sequence of hidden states within this window. The last element of this sequence, $\hat{X}_{400}$, represents the inferred regime for the current day.

3.2.2 Results

Once trained, the Hidden Markov Model identifies two distinct regimes based on the volatility of the z-score series. The estimated parameters clearly reflect a separation between a low-variance and a high-variance state:

$$\begin{aligned} \text{Empirical State 0: } & \mu = 0.02, \quad \sigma = 0.7136 \\ \text{Empirical State 1: } & \mu = -0.01, \ \sigma = 1.7325 \\ \text{State 0: } & \mu = 0.02, \quad \sigma = 0.699 \\ \text{State 1: } & \mu = -0.01, \ \sigma = 1.745 \end{aligned}$$

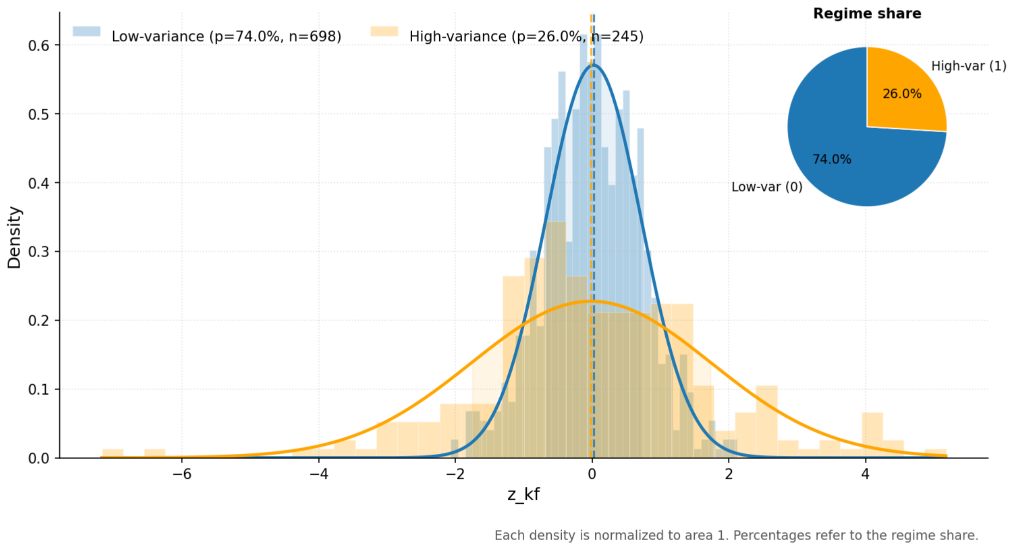


Figure 3.1: HMM Estimated State Distributions

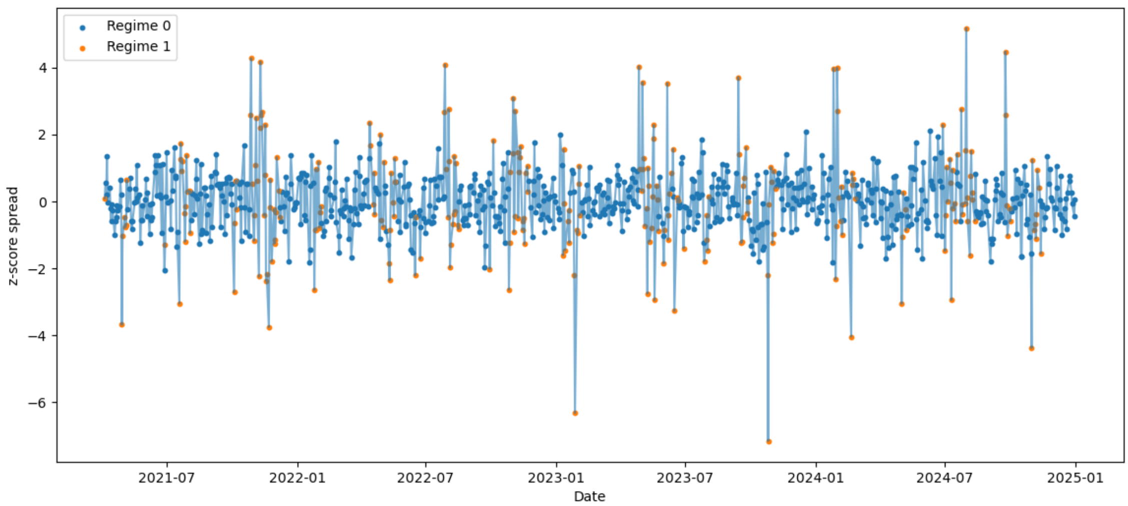


Figure 3.2: Identified regimes on the z-score

This is the transition matrix P, learned automatically during the training phase, which describes the transition probabilities between the two regimes

$$P = \begin{bmatrix} 0.97 & 0.03 \\ 0.06 & 0.94 \end{bmatrix}$$

The high diagonal values indicate strong persistence in both regimes: the stable state tends to prevail over time, while transitions to the volatile regime are rare but persistent once they occur.

With the regimes identified, the model is now ready to generate trading signals and implement the strategy.

Chapter 4

The Strategy

Once the z-score and the market regime have been defined, trading signals can be generated.
This chapter first introduces the conditions and operational rules that govern the generation of trades, building upon the theoretical and methodological tools developed so far. The second part of the chapter is devoted to the backtesting of the strategy, applied not only to the Visa–Mastercard pair but also to other asset combinations. It should be emphasized that the backtesting phase, being strongly dependent on input parameters and operating conditions, serves primarily as a validation and exploratory tool, rather than as a definitive measure of the strategy's robustness.

It is important to remember that both models require an initial training period, of 315 days in our case: the Kalman Filter to estimate the covariance matrices Q and R, and the HMM to learn how to distinguish between the two market regimes based on the z-score.
During this period, the Kalman Filter collected the observations and estimated its parameters, allowing the spread and z-score to be computed. The z-score was then used as input for training the HMM.
As a result, the strategy becomes operational only after this burn-in period: the first trading signals are generated starting from April 5, 2021.

4.1 Trading Signals and Execution Rules

The possible signals are three: long (buy Mastercard, sell Visa), short (buy Visa, sell Mastercard), or hold (no trade).
A signal is generated only if both of the following conditions are met:

- the z-score exceeds one of the predefined thresholds (entry or exit threshold);
- the market regime identified by the HMM is classified as safe (mean-reverting).

To avoid look-ahead bias, trades are executed on the following day rather than on the same day the signal is triggered.
It is important to emphasize that the choice of entry and exit thresholds, which

have a strong impact on the profitability of the strategy, was made in an arbitrary manner. The thresholds reported in the results table are the outcome of a grid search performed ex post. In more advanced strategies, however, these thresholds can be estimated dynamically, evolving over time in response to market conditions, volatility, or specific risk metrics.

The final dataset, used for strategy implementation and backtesting, is therefore structured on a daily basis as follows:

- Date
- MA: Mastercard price
- V: Visa price
- α_t: coefficient estimated by the Kalman Filter
- β_t: coefficient estimated by the Kalman Filter
- rolling mean: moving average of the spread
- rolling std: moving standard deviation of the spread
- z-score: computed through rolling window
- regime: 0 = safe (mean-reverting), 1 = risky (high-volatility)
- signal: 1 = long, –1 = short, 0 = hold
- position: effective position (equal to the previous day's signal)

4.2 Backtest and Performance Evaluation

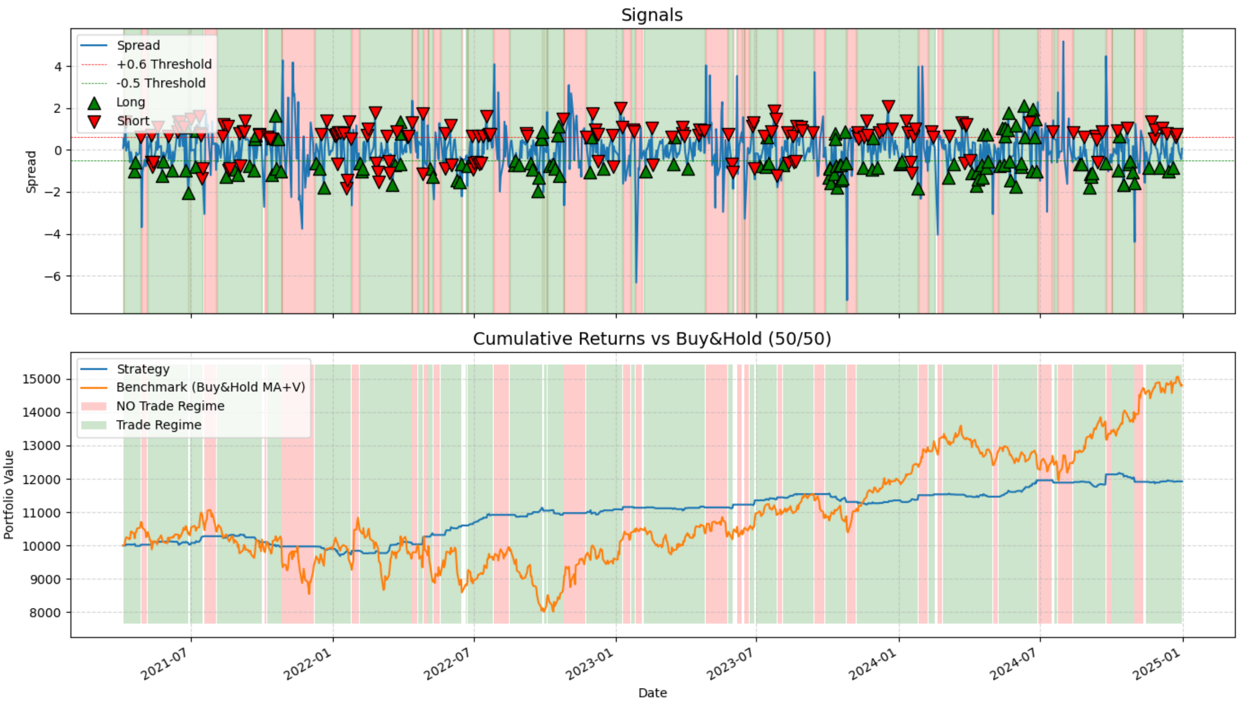
The backtest starts from a fixed initial capital (its absolute value is irrelevant for the computation of returns). Daily asset returns are computed as percentage price changes.
The position on the spread is determined by the trading signal and by the hedge ratio β_{t-1}, estimated through the Kalman Filter.
We then construct a long–short portfolio with unit gross exposure, defined on the spread $Y - \beta X$. In practice, allocations are scaled so that the sum of the absolute positions is always equal to 1, regardless of the value of β. The portfolio weights are:

$$w_Y = \frac{\text{pos}}{1 + |\beta|}, \qquad w_X = -\frac{\text{pos}\,\beta}{1 + |\beta|}$$

The gross exposure is therefore fixed at 1 (so that $|w_Y| + |w_X| = 1$), while the allocation between the two legs depends on $|\beta|$: if $|\beta|$ is large, the weight on asset X increases and the weight on asset Y decreases, and vice versa.
The strategy returns are computed as the weighted combination of the two asset returns, net of transaction costs. Transaction costs are modeled as 1 basis point on

total turnover (the sum of daily absolute changes in positions).
Finally, trades are executed with a one-day lag with respect to the signal, using the position variable (lagged signal), in order to avoid look-ahead bias.

The figure below presents the results of the strategy backtest.



The upper panel shows the z-score with trading signals generated when thresholds are crossed: green indicates a long signal and red a short signal. Some signals appear inverted because, as discussed earlier, trades are executed on the following day to avoid look-ahead bias.
The lower panel reports the cumulative returns of the strategy (in blue), compared with a benchmark consisting of an equally-weighted (50/50) portfolio of the two assets. Green and red backgrounds indicate the different market regimes identified by the HMM.

To assess the performance of the strategy, several standard risk–return metrics were computed:

- Sharpe ratio: evaluates excess return over the risk-free rate per unit of total risk (volatility).
- Sortino ratio: similar to the Sharpe ratio, but penalizes only downside volatility, making it more suitable for asymmetric strategies.
- CAGR (Compound Annual Growth Rate): measures the annualized growth rate of the invested capital.

- Annualized volatility: estimates the variability of returns on a yearly basis, using the standard deviation of daily returns.
- Max Drawdown: captures the maximum peak-to-trough decline in cumulative capital.
- Calmar ratio: relates CAGR to Max Drawdown, highlighting the return achieved relative to drawdown risk.

In our example, the strategy achieved a Sharpe ratio of 1.26 and a Sortino ratio of 1.51. Without the HMM, the Sharpe ratio dropped to 1.10 (using the same thresholds), suggesting that the model effectively acted as a risk management tool in this case. However, in other tests on different pairs of assets, results were sometimes better without the HMM. This may be due to suboptimal classification of regimes during training, or simply because the riskier version of the strategy happened to be more profitable in those specific cases.

Regarding risk, the 50/50 benchmark strategy proved more profitable in the long run, but also more volatile, highlighting a different balance between risk and return. The correlation between the strategy returns and the 50/50 benchmark is 72%. This value indicates a substantial, though not excessive, relationship: while the strategy naturally shares part of the dynamics of the two underlying assets, it also introduces a distinct component that differentiates it from a simple equally-weighted portfolio.

The following figures display the heatmaps of the results obtained for different combinations of entry and exit thresholds. The first figure reports the outcomes with the HMM activated, while the second figure shows the results with the HMM turned off.

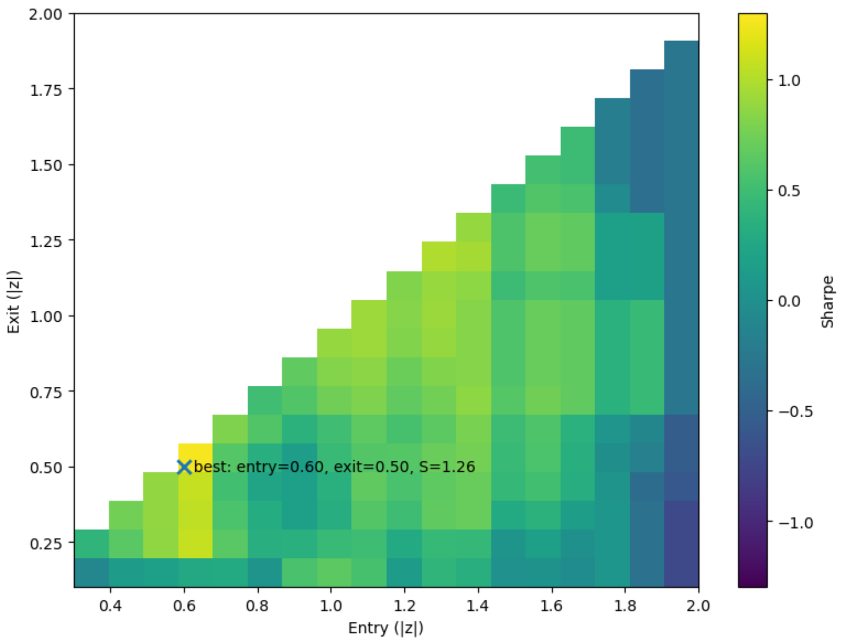


Figure 4.1: Results with HMM

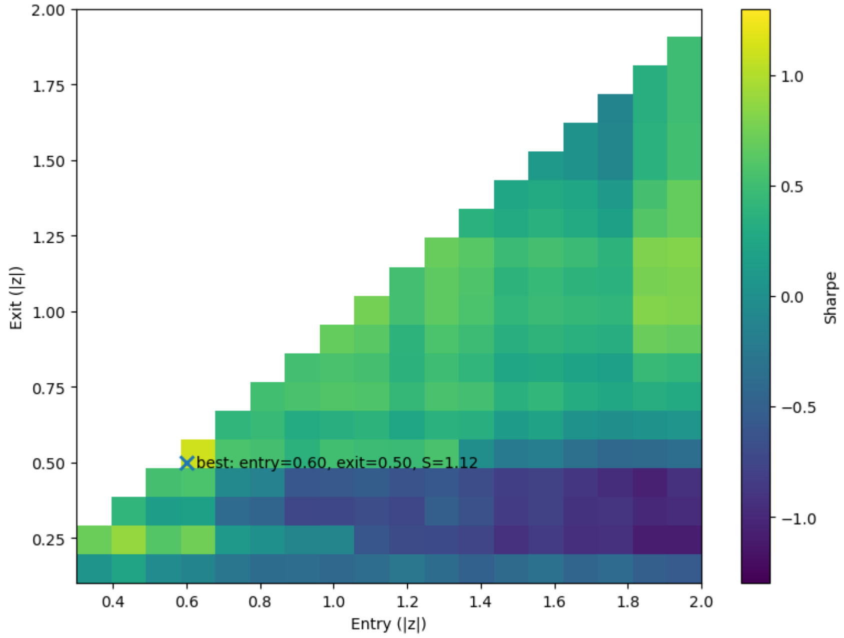


Figure 4.2: Results without HMM

The grid search analysis of entry and exit thresholds shows that the strategy generally performs better when the HMM is active. Specifically, with the model enabled, the mean Sharpe ratio is 0.37 and the median is 0.48; with the HMM switched off, the mean drops to 0.08 and the median to 0.30. This indicates that, on average, the inclusion of the HMM enhances the stability and robustness of the strategy, despite the variability observed in individual cases.
It is also important to highlight the significant impact of transaction fees. In our case, they substantially lowered the profitability of the strategy: for example, the Sharpe ratio drops from 1.48 without costs to 1.26 once costs are included, a reduction of 0.22 points.

The following section presents the results of the strategy applied to seven asset pairs. The evaluation considers different combinations of entry and exit thresholds, as well as the use of the HMM as a regime filter. The reported values correspond to the optimal configurations identified through grid search.

- **IBM–NVDA:** two large technology firms, combining a mature company (IBM) with a highly volatile growth stock (NVIDIA).
- **EWA–EWC:** exchange-traded funds tracking the Australian and Canadian markets, typically used as geographically diversified but correlated equity exposures.
- **SHEL–MPC** and **SHEL–VLO:** oil and energy companies operating in the downstream segment, characterized by strong correlation driven by global crude oil prices.
- **BP–COP:** another energy pair (BP and ConocoPhillips) with similar business models but different regional exposures.

- **AMZN–DLR:** a cross-sector pair combining Amazon (consumer and cloud services) and Digital Realty (data center REIT), included to test the method on assets with weaker fundamental linkage.

This selection allows the evaluation of the model's performance under different correlation structures, volatilities, and sectoral behaviors.

Pair	**Corr**	**Coint p-value**	**Entry/Exit**	**Sharpe**	**Sortino**	**Ann. Vol.**	**CAGR**	**MaxDD**	**Calmar**	**HMM**
MA-V	98.35%	8.9e-05	0.6 – 0.5	1.26	1.51	3.80%	4.8%	-6.2%	0.77	ON
IBM-NVDA	93.3%	0.0026	0.8 – 0.2	0.78	0.82	15.71%	11.6%	-19.9%	0.58	ON
EWA-EWC	96.12%	0.042	1.7 – 0.8	1.34	0.93	2.90%	3.9%	-2.35%	1.71	OFF
SHEL-MPC	96.18%	0.0007	0.6 – 0.1	0.91	1.16	13.50%	12.1%	-13.9%	0.87	ON
SHEL-VLO	96.0%	0.0012	1.1 – 0.1	1.23	0.96	5.05%	6.3%	-3.4%	1.83	ON
BP-COP	92.4%	0.0034	1.1 – 0.3	1.03	0.67	8.10%	8.4%	-9.8%	0.86	ON
AMZN-DLR	87.4%	0.0004	0.8 – 0.6	1.26	1.65	16.5%	21.5%	-19.5%	1.10	OFF

Table 4.1: Results

The strategy appears to deliver acceptable results in most cases. The MA–V pair proved to be one of the most solid, thanks to its very high correlation and cointegration values. As previously mentioned, the use of the HMM was not always beneficial: in the case of EWA–EWC, the pair with the lowest annual volatility, activating the filter reduced the Sharpe ratio from 1.34 to 0.61. Another case worth highlighting is AMZN–DLR, which achieved strong performance without the HMM but also showed the highest volatility among all pairs.

It is important to note that the backtesting phase is not considered an essential component of this project, as results are strongly dependent on parameter choices (entry/exit thresholds, training window length, Kalman Filter configuration, time window, training set etc.). Moreover, since thresholds were selected ex post, the results are subject to overfitting and should not be interpreted as indicative of future performance.
For these reasons, backtesting is used here primarily as an exploratory and validation tool to verify the correct functioning of the models and the strategy, rather than as a definitive measure of robustness or profitability.

Chapter 5

Conclusions

In this project, we presented the development and implementation of a quantitative trading strategy based on two main models: the Kalman Filter and the Hidden Markov Model. The former was used to dynamically estimate the relationship between two assets, while the latter served as a tool for assessing and managing trading risk through the identification of market regimes.

The empirical results obtained across several pairs confirmed the correct functioning of the models, producing coherent and generally acceptable performances (arguably more realistic than those reported in some papers and online sources).
The Kalman Filter proved to be a highly valuable tool for dynamic parameter estimation and remains today a cornerstone of even the most advanced strategies used by major funds. Ongoing research continues to explore new variants, improved methods for estimating the covariance matrices, and techniques to enhance the model's stability and computational efficiency. The HMM, in turn, added an additional layer of risk control and, in most cases, contributed positively to the robustness of the strategy.

From a methodological standpoint, the thesis highlights both the importance and the difficulty of achieving dynamic, unbiased modeling. During the design phase, we paid particular attention to avoiding the so-called "seven sins of quantitative investing," including look-ahead bias, overfitting, poor outlier management, the omission of transaction costs in backtesting etc.
At the same time, the study emphasized the inherent limitations of the backtesting phase, which is strongly influenced by numerous parameters and therefore cannot be considered indicative of future performance. The results should thus be interpreted primarily as exploratory evidence and as a validation of the models' functioning rather than as a definitive performance benchmark.

Future work could focus on the implementation of dynamic entry and exit thresholds, a deeper optimization of model parameters, or the inclusion of additional risk management layers. Each element developed in this project represents only a baseline implementation—every one of these tools could be further refined and expanded, forming the foundation for increasingly sophisticated quantitative trading strategies.

Bibliography

Chan, E. P. (2009). *Quantitative trading: How to build your own algorithmic trading business*. John Wiley & Sons.

Chan, E. P. (2013). *Algorithmic trading: Winning strategies and their rationale*. John Wiley & Sons.

Fanelli, L., Fontana, R., & Rotondi, T. (2023). A hidden markov model for statistical arbitrage in international crude oil futures market.

Gatev, E., Goetzmann, W. N., & Rouwenhorst, K. G. (2006). Pairs trading: Performance of a relative-value arbitrage rule.

Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, *57*(2).

Iain L. MacDonald, R. L., Walter Zucchini. (2016). *Hidden markov models for time series*. Chapman & Hall.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems.

Khan, Y., & Dutt, D. (2006). An expectation-maximization algorithm based kalman smoother approach for event-related desynchronization (erd) estimation from eeg.

Palomar, D. P. (2025). *Portfolio optimization: Theory and application*. Cambridge University Press.

Vidyamurthy, G. (2004). *Pairs trading: Quantitative methods and analysis*. John Wiley & Sons.