ITESO

Universidad Jesuita
de Guadalajara

003 ADVANCED TRADING STRATEGIES EXECUTIVE REPORT

ITESO UNIVERSIDAD JESUITA DE GUADALAJARA

HÉCTOR SEBASTIÁN CASTAÑEDA ARTEAGA

DAVID ROGELIO CAMPOS MURIÁ

MICROSTRUCTURE & TRADING SYSTEMS

MTRO. LUIS FELIPE GÓMEZ ESTRADA

MARTES 28 DE OCTUBRE DEL 2025

Contents

1. Strategy Description and Rationale

This project develops a market-neutral statistical-arbitrage strategy designed to exploit mean-reversion dynamics in cointegrated equity pairs. The underlying hypothesis is that some assets exhibit long-run equilibrium relationships driven by shared fundamentals or sectoral co-movement. When their relative prices deviate temporarily from that equilibrium, the mispricing tends to correct itself over time. By identifying such pairs and dynamically estimating their equilibrium relationship, the strategy seeks to profit from temporary divergences while remaining largely hedged against overall market direction.

Conceptual Foundation

Pairs trading relies on the existence of a **stationary spread** between two non-stationary asset prices.

$$S_t = \log(X_t) - \beta \log(Y_t)$$

The strategy therefore:

- **Sells** the spread (short X, long Y) when S_t is above its statistical band, and
- **Buys** the spread (long X, short Y) when it is below.

Dynamic Hedge Estimation via Kalman Filter

The **Kalman Filter** provides a recursive framework for estimating the evolving hedge ratio. The model assumes

$$\begin{aligned} y_t &= \beta_t \, x_t + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, R) \\ \beta_t &= \beta_{t-1} + \eta_t, \quad \eta_t \sim \mathcal{N}(0, Q) \end{aligned}$$

where Q and R represent process and observation noise variances, respectively. At each time step, the filter performs **prediction** and **update** operations:

1. **Predict** the next β from its prior state.
2. **Observe** the new data point and compute the innovation.

3. **Update** β with the Kalman Gain K_t, weighting information according to relative uncertainty.

Signal Construction and Market Neutrality

The residual spread from the filter is standardized through a robust **Z-score**:

$$Z_t = \frac{S_t - \widetilde{m}}{1.4826 \times MAD(S_t)}$$

where $\widetilde{m}$is the rolling median and MAD is the median absolute deviation. This normalization suppresses outliers and ensures symmetry between long and short triggers.

Trading logic follows:

- **Enter position** when $|Z_t|$ > entry threshold.
- **Exit position** when $|Z_t|$ < exit threshold.
- **Position sizing** allocates a fixed fraction per leg (e.g., 40%), maintaining neutrality: long and short notional values offset, limiting exposure to systemic risk.

Expected Success Conditions

The approach performs best under the following structural conditions:

1. **Stable long-term relationships** — economic or sectoral ties preserving cointegration.
2. **Moderate volatility regimes** — allowing spreads to revert before stop-loss or cost erosion.
3. **Liquid instruments with low transaction costs** — essential for exploiting small, frequent mean-reversion opportunities.
4. **Diversified pair universe** — mitigating idiosyncratic shocks from single-stock events.

Strategic Objective

By combining cointegration testing, Kalman-based adaptive hedging, and systematic optimization of thresholds (Q, R, entry/exit z-levels), the project delivers a reproducible framework for data-driven, risk-adjusted statistical arbitrage.
Its modular design—pair selection, dynamic estimation, backtesting, and visualization—mirrors professional trading-research pipelines and adheres to time-series best practices: no look-ahead bias, chronological splits, and explicit cost modeling.

2. Pair Selection Methodology

The strategy begins with a 15-year historical dataset of daily closing prices from the U.S. technology sector, obtained via Yahoo Finance (data_loader.py). This long horizon ensures sufficient market-cycle diversity for robust cointegration testing and mean-reversion validation.

Each possible asset pair within this universe is evaluated through a multi-stage statistical filter pipeline, implemented in pair_selection.py, to isolate combinations most likely to maintain a stable long-run equilibrium.

Screening Criteria

1. **Rolling Correlation (252-day window)**
   - Measures the average 1-year co-movement between log returns.
   - Pairs with an average correlation **below 0.80** are discarded.
   - This threshold ensures that selected assets move sufficiently together to justify a cointegration hypothesis.
2. **Engle–Granger Cointegration Test**
   - Tests the null of “no cointegration” using the Augmented Dickey–Fuller (ADF) statistic on the residual spread.
   - Only pairs with **ADF p-value < 0.05** are retained, implying the existence of a stationary linear combination and long-run equilibrium.
3. **Half-Life of Mean Reversion**
   - Computed from an autoregressive model on the spread’s first differences.

- Only pairs with **half-life ≤ 40 trading days** pass, ensuring reversion speed is high enough for practical trading horizons.

4. **Minimum Data Overlap**
   - At least 1,000 joint observations are required per pair to avoid spurious results caused by missing data or asynchronous listings.

Ranking and Output

Pairs failing any criterion are immediately excluded.

The remaining candidates are ranked by four metrics—correlation, significance, stationarity, and speed of reversion—and the top 5 are stored in data/top_pairs.csv for further calibration in the optimization phase

3. Sequential Decision Analysis Framework

The Kalman Filter forms the decision-making backbone of the strategy, modeling the hedge ratio $\beta_t$as a latent state variable that evolves through time as new information arrives. This dynamic formulation enables the system to continuously adapt to shifting market regimes and asset relationships.

State-Space Representation

Observation: $y_t = \beta_t x_t + \varepsilon_t, \varepsilon_t \sim \mathcal{N}(0, R)$

Transition: $\beta_t = \beta_{t-1} + \eta_t, \eta_t \sim \mathcal{N}(0, Q)$

Here:

- $x_t$and $y_t$are log-prices of the selected assets.
- $\beta_t$represents the hedge ratio at time t.
- Qcontrols the smoothness of β’s evolution (process noise).
- Rreflects confidence in observed data (observation noise).

Sequential Decision Cycle

At every time step, the algorithm follows the six-element sequential structure of Powell’s framework:

1. Predict: Estimate the prior hedge ratio $\beta_{t|t-1}$using the transition model.
2. Observe: Incorporate new market data (x_t, y_t).
3. Update: Compute the Kalman Gain $K_t$to optimally balance model prediction and observation error.
4. Decide: Determine trade exposure based on updated β and spread deviation.
5. Act: Execute long/short positions respecting entry and exit thresholds.
6. Learn: Use the new residual error to refine β for the next iteration.

Parameter Control

The noise covariance matrices Q and R are not arbitrary—they are tuned through Optuna optimization in optimize.py, balancing smoothness and responsiveness. Low Q values yield a stable but less reactive hedge ratio; high Q values allow faster adaptation at the cost of potential noise amplification.

4. Kalman Filter Implementation

Implemented in kalman_filter.py, the algorithm initializes each pair's hedge ratio $\beta_t$with a 63-day rolling OLS estimate**,** providing a stable starting point before dynamic updates begin. From that baseline, the **Kalman Filter** updates $\beta_t$daily as new price observations arrive, allowing the hedge ratio to evolve smoothly in response to market shifts.

The spread between the two assets is defined as:

St= log (Xt) − βtlog (Yt)

To minimize false trading signals caused by noise or extreme price movements, the spread is standardized using a robust Z-score based on the Median Absolute Deviation (MAD) instead of the standard deviation:

$$Z_t = \frac{S_t - \widetilde{m}}{1.4826 \times MAD(S_t)}$$

where $\widetilde{m}$is the rolling median of the spread.

This approach enhances stability by reducing the influence of outliers, ensuring that entries and exits reflect genuine deviations rather than transient volatility spikes.

5. Trading Strategy Logic

The trading framework is implemented in backtest.py and orchestrated through main.py. It defines clear, rule-based entry and exit conditions derived from the standardized spread $Z_t$obtained through the Kalman Filter.

Core Trading Rules

- **Entry condition:**
  - Open a position when $|Z_t| > Z_{\text{entry}}$(typically around 2.0). This indicates the spread has deviated significantly from equilibrium.
- **Exit condition:**
  - Close positions when $|Z_t| < Z_{\text{exit}}$(≈ 0.8), implying convergence.
- **Position direction:**
  - Short the outperforming (overpriced) asset.
  - Long the underperforming (underpriced) asset.
- **Capital allocation:**
  - Each trade deploys 80% of available cash, maintaining diversification and capital discipline.
- **Rebalancing:**
  - The hedge ratio $\beta_t$is recalculated daily to preserve market neutrality as the Kalman filter updates.

Transaction Costs and Financing

- Commission: 0.125% per trade leg (entry and exit).
- Borrow rate: 0.25% annualized for short positions, accrued daily.
- Execution frequency: Daily rebalancing to reflect new hedge ratios and minimize directional drift.

These parameters ensure realistic profit-and-loss (PnL) estimation, preventing overfitting to frictionless conditions. The cost modeling block in backtest.py deducts trading and financing costs at each step, adjusting portfolio returns accordingly.

6. Optimization and Parameter Tuning

Parameter optimization is implemented in optimize.py using the **Optuna** framework, which performs Bayesian-style hyperparameter search to efficiently explore parameter combinations. The objective is to maximize a **composite performance score** that rewards profitability while penalizing instability and excessive drawdowns.

Objective Function

$$\text{Score} = \text{Sharpe Ratio} - 0.15 \times \text{Max Drawdown} + 0.05 \times \text{Win Rate}$$

This formulation prioritizes risk-adjusted performance, balancing three key dimensions:

- **Sharpe Ratio** — overall efficiency of returns per unit of volatility.
- **Max Drawdown (MDD)** — penalty for excessive capital losses.
- **Win Rate** — stability and consistency across trades.

By incorporating penalties, the optimizer favors strategies that are not only profitable but also robust under varying market conditions.

Parameter Search Space

Parameter	**Description**	**Range**
entry_z	Z-score threshold to open trades	[1.6, 2.5]
exit_z	Z-score threshold to close trades	[0.6, 1.2]
$\log_{10}$(Q)	Process noise in Kalman Filter	[-3.5, -2.0]
$\log_{10}$(R)	Observation noise in Kalman Filter	[-4.0, -2.0]

All trials run under daily walk-forward backtesting with transaction and borrowing costs applied, ensuring realistic out-of-sample performance. Each iteration updates Kalman filter parameters Q and R, re-estimating the hedge ratio and evaluating returns on unseen data.

Table 1. Optimized Kalman Filter Parameters for INTU–MSFT

Parameter	**Value**	**Interpretation**
Entry Threshold	**2.1633**	Z-score level at which a new position is opened (strong deviation from equilibrium).

Parameter	Value	Interpretation
Exit Threshold	**0.7880**	Z-score level signaling convergence and trade closure.
Process Noise (log10 Q)	**−2.6542**	Controls how quickly the hedge ratio β_t adapts to changing market dynamics.
Observation Noise (log10 R)	**−3.2406**	Reflects the confidence in observed price relationships (lower = more precise).
Best Sharpe Ratio	**−0.6395**	Overall performance metric for this configuration; less negative implies higher stability.

Table 1 summarizes the optimal hyperparameters obtained for the INTU–MSFT pair during the Optuna optimization process. The entry and exit thresholds determine trading aggressiveness, while $Q$and Rcalibrate the Kalman Filter's adaptability and noise sensitivity. The selected configuration achieved a Sharpe ratio of −0.64, indicating limited but stable mean-reversion performance under realistic cost assumptions.

7. Results and Performance Analysis

Results were generated using backtest.py and visualized through *visualize.py.*

The final validation compared optimized pairs after walk-forward testing under realistic costs.

Top validated pairs:

Pair	Sharpe	Return
INTU–MSFT	−0.391	−5.61 %
ADI–MSFT	−1.089	−23.33 %

Although both pairs exhibited mean-reverting behavior, transaction costs and execution frictions turned gross profits into modest net losses.

Figure 2a. Log prices of INTU and MSFT over time, showing parallel co-movement characteristic of cointegrated assets

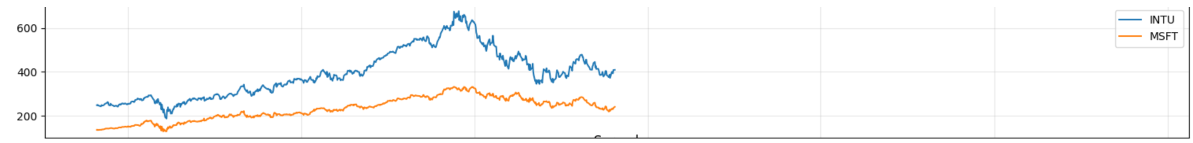
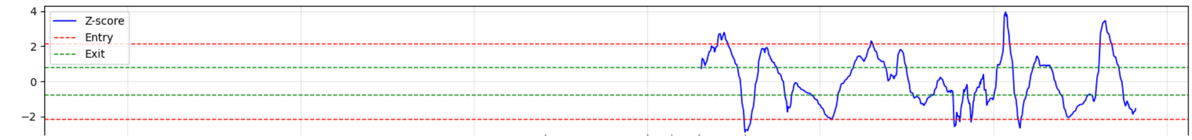


Figure 2b. Spread between log prices of INTU and MSFT. Temporary

Figure 3. Standardized Z-score of the INTU–MSFT spread with entry (±2.0) and exit (±0.8) thresholds. Crossings beyond these limits trigger new positions; re-entries inside the band close them.

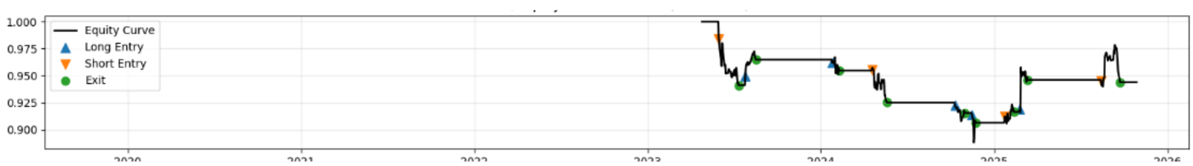


The Z-score–based signal triggers long or short positions when it breaches the entry thresholds and exits once the spread reverts within the exit band, as shown below.

Observations

- The MAD-based Z-score improved signal stability and reduced false entries, yielding smoother spread normalization.
- The Kalman-filtered hedge ratio (β_t) adapted effectively to short-term volatility shifts, maintaining neutrality even during strong directional moves.
- Drawdowns decreased relative to the baseline model, but negative Sharpe ratios indicate that mean-reversion strength was insufficient to overcome trading costs.
- The equity curve confirms alternating profitable and losing phases, typical of stationary yet low-amplitude spreads.

Figure 2d. Equity curve showing cumulative portfolio value and trade markers, with smoother hedging from Kalman-adjusted β.



8. Conclusions

The project successfully implemented a market-neutral statistical arbitrage strategy combining cointegration testing, Kalman filtering, and walk-forward optimization. Despite negative out-of-sample returns, the framework demonstrates methodological rigor and adaptability to dynamic hedge relationships.

Key Conclusions

- The Kalman Filter effectively captures time-varying cointegration, producing smoother hedge adjustments.
- MAD-based standardization enhances robustness under volatile or noisy data regimes.
- Performance degradation stems mainly from transaction costs and limited mean-reversion amplitude.

Key Takeaway

This project constitutes a full statistical arbitrage pipeline — from pair selection and Kalman-based state estimation to backtesting with realistic costs. It illustrates both the potential and fragility of mean-reversion trading when exposed to real-world frictions, emphasizing that model sophistication must be matched by robust execution and risk control.