

A System and Method for Adaptive Algorithmic Trading via Real-Time Parameter Optimization and Dynamic Reconciliation

Trisagion Developers

support@trisagion.xyz
www.trisagion.xyz

September 8, 2025

Abstract

Herein is disclosed a system and method for algorithmic trading, hereinafter referred to as the Tactical Trend Trader (3T), designed to execute trades based on a quantitative, adaptive framework. The system operates as a persistent, stateful process that continuously evaluates market conditions to inform trading decisions. The core operational loop comprises four distinct stages: data ingestion, multi-parameter optimization, trade evaluation and execution, and state reconciliation. A primary contribution of this work is a massively parallel optimization engine that tests thousands of parameter permutations in real-time to identify locally optimal configurations. For market state classification, the system utilizes Permutation Entropy to distinguish between predictable, trending regimes and stochastic, non-trending noise, thereby filtering conditions unsuitable for trading. A key innovation is its departure from the classical Kelly Criterion for position sizing; instead, it employs a performance-based heuristic that dynamically modulates a base position size according to the profitability of the most recent trading epoch, or 'block'. Furthermore, risk management is abstracted from individual trades and managed at the portfolio level via a scheduled reconciliation process, which continuously monitors and adjusts open positions, obviating the need for conventional stop-loss orders. This synergistic combination of market regime filtering, real-time parameter optimization, adaptive position sizing, and dynamic risk management provides a robust and comprehensive framework for systematic trading.

1 Introduction

The domain of algorithmic trading is characterized by a persistent challenge: developing systems that can adapt to non-stationary and often chaotic financial markets.

Many conventional systems rely on statically configured parameters, rendering them brittle and susceptible to performance degradation as market dynamics shift. The Tactical Trend Trader (3T) is a computer-implemented system engineered to address these limitations by integrating real-time adaptability into its core architecture. It is designed to optimize performance by leveraging statistical analysis for market regime detection and a novel set of heuristics for risk and position management.

The 3T system operates in a continuous loop, analyzing market data to first identify predictable, trending environments and then deploying capital with a position size dynamically calibrated to recent system performance. This paper details the mathematical framework, system architecture, and unique operational logic of the 3T. The primary contributions of this method are threefold:

1. **A Real-Time Optimization Engine:** A mechanism that continuously stress-tests thousands of parameter permutations against live market data to inform the primary trading algorithm.
2. **A Hybrid Risk Management Model:** A novel heuristic for position sizing based on recent performance, coupled with a dynamic, reconciliation-based approach to position control that replaces static, per-trade stop-losses.
3. **Dual Entry Logic:** A configurable system that supports both momentum-following ("breakout") and mean-reverting ("swing") entry strategies, allowing for greater tactical flexibility.

2 System Architecture

The 3T system is a modular architecture comprising four main components: a Data Ingestion Module, a Parameter Optimization Engine, a Trade Evaluation and Execution Engine, and a State Reconciliation Module. This separation of concerns ensures robustness and scalability.

2.1 Data Ingestion Module

This module serves as the sensory input for the entire system, acquiring market data from two primary sources:

- **Real-time Websocket Stream:** Provides low-latency access to immediate price updates (tick data), crucial for the Parameter Optimization Engine.
- **Historical API Endpoint:** Fetches historical Open/High/Low/Close (OHLC) data, which is subsequently aggregated into one-minute volatility bars. This provides the necessary historical context for macro-level analysis like Permutation Entropy calculation.

The module includes logic for data normalization, cleaning, and ensuring data integrity before it is passed to the analytical engines.

2.2 Parameter Optimization Engine

This engine operates as a continuous, massively parallel simulation layer that runs concurrently with the live trading system. Its purpose is to find the most profitable set of parameters for the core trading algorithm in the current market environment. It cyclically tests thousands of permutations of a core set of variables against real-time data to determine the most profitable configurations. The key parameters under optimization include:

- **Permutation entropy parameters:** The embedding dimension d and time lag τ used for market state analysis.
- **Market state thresholds:** The specific value of Permutation Entropy, H_{trend} , below which the market is considered 'trending'.
- **Max duration:** The maximum time, in seconds, that a single simulation (a 'run') is allowed to process data before it is terminated. This constrains the computational search space.
- **Max direction reversal:** The maximum time, in seconds, that a run will attempt to find a market direction before exiting. This parameter manages capital exposure during periods of indecisive price action.
- **APR target:** The target Annual Percentage Rate (APR) that the system aims to achieve. This serves as a benchmark for performance evaluation and the fitness function of the optimization.
- **Rolling APR minutes:** The time window, in minutes, over which the system's APR is calculated. This determines the sensitivity of the system to recent performance.
- **Decision distance seconds:** The minimum time interval between successive trade evaluations to allow for the maturation of the market's response to the previous action.
- **System type:** The fundamental entry logic, supporting "breakout" or "swing" methodologies.

The most profitable parameter set discovered by this engine is then propagated to the Trade Evaluation and Execution Engine to be used in the live environment.

3 Mathematical Framework and Operational Logic

The core of the 3T is a continuous loop executed by the Trade Evaluation and Execution Engine, which acts upon the optimal parameters supplied by the Parameter Optimization Engine. Each iteration performs a rigorous sequence of data analysis and trade management operations.

3.1 Market Regime Analysis via Permutation Entropy

To prevent trading in stochastic, unpredictable market conditions, the system first performs a market regime analysis. This methodology is chosen for its robustness to noise and its model-free nature.[1]

1. A time series is constructed from the most recent N price points derived from the aggregated volatility bars.
2. Permutation Entropy, $H(d)$, is calculated on this time series using the optimized parameters (d, τ) . The formula for Permutation Entropy is given by:

$$H(d) = - \sum_{i=1}^{d!} p_i \log_2 p_i$$

where d is the embedding dimension and p_i is the relative frequency of the i -th permutation ordinal pattern.

3. If the calculated value $H(d)$ is below the optimized threshold H_{trend} , the market is classified as **trending** and algorithmically predictable. The algorithm proceeds to the next stage. Otherwise, the market is classified as **noisy**, and no new trading action is initiated.

3.2 System Entry Logic: Breakout and Swing Types

Once a trending market is confirmed, the system employs its configured entry logic based on the `system_type` parameter.

- **Breakout Logic:** This is a momentum-following strategy. A long entry is considered if the current price exceeds the highest high of the last N periods, and a short entry is considered if the price falls below the lowest low of the last N periods. This logic seeks to capitalize on the continuation of a strong trend.
- **Swing Logic:** This is a mean-reversion strategy. It identifies short-term price extensions away from a central tendency (e.g., a moving average). A long entry is considered when the price has dipped significantly below the mean and shows signs of reverting, while a short entry is considered after a significant rally above the mean.

3.3 Position Sizing Heuristic and APR Trend Analysis

When a valid entry signal is generated, the system determines whether to increase its risk exposure. This decision is governed by a strict gating condition based on the system's profitability trend, as measured by its Annual Percentage Rate (APR), calculated over the `rolling_apr_minutes` window.

3.3.1 APR Gating Condition

A core rule of the risk management framework is that position size will **only** be increased if the system’s rolling APR is trending positively and is above the `apr_target`. If the APR is flat, negative, or below target, the system will not increase its base position size, ensuring that risk is escalated only during periods of sustained, proven profitability.

3.3.2 Performance-Based Sizing Heuristic

If the APR gating condition is met, the system calculates the appropriate position size. It deliberately eschews a direct implementation of the Kelly Criterion[2][4], which can be sensitive to estimation errors in probabilities and payoffs. Instead, it employs a novel performance-based heuristic that modulates a base risk position size (S_{base}) based on the performance of the most recently closed trade (the ‘top block height’, H_{top}).

The performance factor, P_{kelly} , is calculated as a normalized deviation from a benchmark block height, B_{kelly} :

$$P_{kelly} = 1 - \frac{H_{top}}{B_{kelly}}$$

This performance factor is then bounded by a configurable threshold, T_{kelly} :

$$P'_{kelly} = \max(-T_{kelly}, \min(P_{kelly}, T_{kelly}))$$

The final position size, S_{final} , is then determined by adjusting the base size by this bounded performance factor:

$$S_{final} = S_{base} + (S_{base} \cdot P'_{kelly})$$

This mechanism creates a homeostatic effect: it increases position size when the last trade underperformed its benchmark ($H_{top} < B_{kelly}$), effectively ”doubling down” on the strategy, and decreases it when the last trade overperformed ($H_{top} > B_{kelly}$), thereby realizing gains and reducing exposure.

3.4 Bootstrapping Logic

Upon initial deployment, the system exists in a ”cold start” state with no historical block heights to inform the sizing heuristic. To mitigate this, the system uses a pre-configured, conservative default position size for its first trade. Once this initial trade is closed, its resulting profit or loss establishes the first block height ($height = 1$). This value, along with subsequent block heights, forms the statistical basis for the benchmark B_{kelly} , enabling the heuristic for all future trades.

3.5 Execution and Risk Management

1. **Take-Profit:** A take-profit goal is considered at a pre-configured, fixed percentage gain from the portfolio value at start of the current block. When this order is executed, the position is closed, and the realized profitable runs are recorded as part of the newest block height.
2. **Risk Management:** A key innovation of the 3T is its explicit omission of traditional, per-trade stop-loss orders, which are often susceptible to being triggered by transient market volatility or "stop hunting." Instead, risk is managed holistically through the State Reconciliation Module.

4 State Reconciliation Module

The State Reconciliation Module functions as a dynamic, portfolio-level risk manager. Its operation is not event-driven by price action but is scheduled to run at a configured interval (e.g., every 10 minutes), a cadence related to the `decision_distance_seconds` parameter. This prevents high-frequency, reactive adjustments and allows positions time to mature.

During each reconciliation cycle, the module assesses the aggregate health of all open positions. It references the ongoing simulations within the Parameter Optimization Engine to determine the expected performance of the current live configuration. If the actual portfolio performance deviates significantly from the simulated expectancy, or if the market regime has shifted to 'noisy', the module takes corrective action based on the simulated success value of the permutation. This may involve partially reducing a position to decrease exposure or closing a position entirely, even if its take-profit level has not been reached.

To prevent excessive trading fees from minor adjustments ("churn"), the module will only execute a change if the required adjustment in position size is above a minimum financial or percentage threshold (e.g., \$100 or 5% of the position size). This ensures that only strategically meaningful risk adjustments are made, serving as an intelligent, system-wide stop-loss mechanism.

In order to apply risk changes only when the reliability of the current risk-on positions are sound, multiple observer nodes are utilized to ensure consensus.[3] This prevents the situation of a runaway process continuously adding or removing from a position without checks, balances and reconciliations at each iteration.

5 Conclusion

The Tactical Trend Trader (3T) provides a robust and complete framework for adaptive algorithmic trading. Its novelty lies in the synergistic combination of its core components: (1) the use of Permutation Entropy for robust market regime filtering,

(2) a massively parallel, real-time Parameter Optimization Engine that ensures the system remains adapted to current market dynamics, (3) a unique performance-based heuristic for dynamic position sizing, and (4) an active reconciliation module for portfolio-level risk management that replaces brittle, traditional stop-loss orders.

By systematically filtering for predictable market environments, continuously optimizing its own parameters, and intelligently adjusting risk based on real-time performance and expected outcomes, the 3T system is designed to navigate complex financial markets, with the objective of maximizing risk-adjusted returns while diligently managing downside exposure.

References

- [1] Christoph Bandt and Bernd Pompe, *Permutation Entropy: A Natural Complexity Measure for Time Series*, Phys. Rev. Lett. **88**, 174102 (2002).
- [2] J. L. Kelly, Jr., *A New Interpretation of Information Rate*, Bell System Technical Journal, **35**(4), 917-926 (1956).
- [3] Leslie Lamport, Robert Shostak, and Marshall Pease, *The Byzantine Generals Problem*, ACM Transactions on Programming Languages and Systems, **4**(3), 382-401 (1982).
- [4] Van Tharp Institute, *Peak Performance 101 Course for Traders and Investors*. Available online: <https://vantharpinstitute.com/course/peak-performance-course-for-traders-and-investors/>. (Accessed on: September 8, 2025).