ChatGPT

# Patent Draft Revision: Abstract, Claims, and Prosecution Shield


## Abstract

A **quantum-inspired financial analysis system** is disclosed that detects and responds to complex patterns in time-series data (such as market price sequences) through specialized computational techniques. The system **computes a coherence metric and a stability metric** for an incoming binary or numerical sequence using Shannon entropy and state transition analysis [1]. For example, a sequence's Shannon entropy *H* is calculated and **Coherence** is defined as *1 – (H / H<sub>max</sub>)* while **Stability** is defined as *1 – (transition_rate / max_transitions)* [1]. Using these metrics, the system classifies the sequence into pattern categories (e.g. an *AlternatingBlock* pattern if Coherence ≥ 0.8 and Stability ≥ 0.75, versus a *RandomNoise* pattern if Coherence ~0.3 and Stability ~0.2 [2]). The system further identifies rapid **"quantum collapse" events** by detecting when the coherence value drops by more than a **collapse threshold** (for instance, >0.2) within a brief time window (e.g. 3 consecutive data ticks) [3]. In response to such events or identified patterns, the system adaptively adjusts its predictive models in real-time. Additional components include a multi-asset signal fusion module that weighs signals by cross-asset correlation strengths to enhance prediction confidence [4] [5], and a market regime classifier that adjusts thresholds based on volatility regimes (e.g. treating volatility above a high threshold as "High Volatility" regime) [6]. Overall, the invention provides an **improved computer-implemented method** for pattern discovery and decision support in financial trading, leveraging discrete prime-number time steps for updates (prime-indexed "heartbeat" intervals) and **concrete parameter thresholds** to ensure technical robustness and real-world reliability.


## Algorithm Implementation Summary

To **enable reproducible implementation**, the patent document includes detailed algorithmic descriptions with pseudocode and equations:

- **Quantum Field Harmonics (QFH) Algorithm:** The system performs a harmonic decomposition of bit/value sequences to detect latent patterns. A **damping factor** λ is computed as a function of sequence entropy and coherence:

```
λ = k1 * Entropy + k2 * (1 - Coherence)
V_i = Σ (p_j - p_i) * e^(-λ * (j - i))
```

where *k1* and *k2* are weighting constants (e.g. 0.3 and 0.2 respectively) [7]. This exponential damping emphasizes recent divergences; the computed values *V_i* highlight transitions. The QFH module flags specific **pattern types** when coherence and stability metrics meet predefined thresholds. For instance, an **AllFlip** pattern (complete alternation) is identified at Coherence ≈0.9 and Stability ≈0.85, whereas a

**MeanReversion** cycle might show Coherence ~0.75, Stability ~0.7 [2]. These concrete threshold values are explicitly provided in the specification to guide implementation.

- **Quantum Bit State Analysis (QBSA):** This module validates and further analyzes patterns by measuring the **Shannon entropy** of the sequence and deriving the **Coherence** and **Stability** as defined above [1]. A high coherence (close to 1.0) indicates an orderly pattern with low entropy, while lower values indicate randomness. **Quantum collapse detection** logic monitors sudden drops in coherence: if the coherence declines by more than a set threshold (e.g. 0.20) within a short timeframe (e.g. <5 ticks), a **collapse event** is registered [3]. This signals that a previously stable pattern has broken down abruptly, prompting the system to react (such as by reducing confidence in related predictions or switching to a different strategy).

- **Multi-Asset Correlation and Signal Fusion:** In deployments with multiple data streams (e.g. several asset prices), the system computes the **Pearson correlation coefficient** between assets (with optional lag optimization) to quantify inter-asset influence [8]. A dynamic correlation weight *ρ_dynamic(t)* is calculated as a blend of short-term and long-term correlations [8]. Using these, each asset's signal is assigned a weight: `weight_i = correlation_strength_i * (1 + confidence_modifier_i)` [4]. The engine then performs a **weighted voting** across assets, summing weighted "buy" vs "sell" indications [9]. A **consensus direction** is chosen by comparing the aggregate weights of buy, sell, and hold signals [10]. The fusion module also computes a cross-asset agreement **coherence** (fraction of agreeing signal-pairs) and boosts the final confidence if multiple assets agree (e.g. final confidence = base_confidence * (0.7 + 0.3 * coherence)) [5]. These formulae ensure that the combined signal accounts for corroborating evidence across assets, thereby improving reliability.

- **Market Regime Adaptation:** The system adapts its thresholds based on detected market conditions. For example, it continuously calculates a normalized volatility (e.g. using ATR divided by price) and classifies the regime as **High**, **Medium**, or **Low** volatility by comparing to predefined numeric thresholds [6]. If volatility exceeds a **HIGH_THRESHOLD** value (defined in the specification, e.g. representing the 80th percentile of historical volatility), the system marks a High-volatility regime; if below a **LOW_THRESHOLD** (e.g. 20th percentile), a Low-volatility regime is assumed [11]. Trend strength is measured (such as percent price change over a window and slope of a moving average), and a composite trend metric is calculated [12]. The patent document provides example equations for these calculations, ensuring that a practitioner can replicate how the system adjusts pattern detection sensitivity in trending vs. ranging markets. In high-volatility regimes, the algorithm may raise its confidence thresholds (demanding stronger pattern coherence before acting) to reduce false signals, as explicitly described in the specification's **Adaptive Threshold Theory** (e.g. adding +0.10 to confidence threshold during turbulent periods) [13].

- **Prime-Number Gated Updates:** To maintain computational stability and avoid resonance artifacts, **fundamental state updates occur only at prime-numbered time steps** [14]. In other words, the engine uses a **prime-step clock** (2, 3, 5, 7, 11, …) to trigger major pattern integration events. This unique timing scheme, detailed in the draft, prevents cyclical feedback loops by introducing non-repeating intervals between full updates. The specification includes a discrete action minimization formula: *L<sub>SEP</sub>(p) = C(p) – I(p)* (computational cost minus information gain at prime step *p*), summing over primes to optimize the system's evolution. By incorporating this prime-gating mechanism, the invention improves upon conventional periodic update systems, contributing to more robust pattern convergence.

Each of the above algorithmic components is **fully described with concrete parameters, threshold values, and example pseudocode** in the patent draft. This ensures that the reader is enabled to implement the system and highlights the specific technical **improvements in computational analysis** (e.g. faster anomaly detection, noise-resistant signal fusion, and adaptive thresholding) offered by the invention.

## Independent Claims Revision

Below are rewritten versions of the independent **system** and **method** claims, now incorporating specific threshold values and mathematical expressions to enhance clarity and specificity:

### Independent System Claim (Revised Draft)

- **1.** A **computing system for real-time pattern analysis in financial data**, comprising:
- **(a)** a data intake module configured to continuously receive a time-series data stream comprising sequential data points from one or more financial market sources;
- **(b)** a **quantum-inspired pattern analysis engine** operatively coupled to the data intake, the engine including:
    - **(i)** an entropy calculation unit that computes a Shannon entropy *H* of the incoming sequence at each update interval;
    - **(ii)** a coherence module that computes a **Coherence value** defined as *1 – (H / H<sub>max</sub>)* for the sequence, where *H<sub>max</sub>* is a predefined maximum entropy for normalization, and a **Stability value** defined as *1 – (transition_rate / max_transitions)* based on observed state transitions in the sequence;
    - **(iii)** a pattern classification unit that compares the Coherence and Stability values to multiple preset threshold pairs and classifies the current sequence state into a **pattern category** when the values meet or exceed a corresponding threshold pair, wherein the threshold pairs are concrete numeric values stored in memory – **for example**, a Coherence threshold of *0.80* and Stability threshold of *0.75* for identifying an **AlternatingBlock pattern**, and a Coherence threshold of *0.95* with Stability *0.90* for a **NullState pattern**;
    - **(iv)** a collapse detection module that monitors changes in the Coherence value over successive data points and triggers a **collapse event signal** if the Coherence value decreases by more than a predetermined fraction within a predetermined short time window, wherein the predetermined fraction is a specific percentage (configured to about **20%** drop) and the time window is a fixed small number of data intervals (configured to, e.g., **3 consecutive intervals**);
- **(c)** a **multi-asset fusion module** that, in response to receiving multiple parallel data streams, calculates pairwise correlation coefficients between the data streams and assigns a **weight** to each stream's signal proportional to its correlation strength with other streams, the module further computing a weighted aggregate signal by summing contributions from each stream and determining a consensus direction as **buy**, **sell**, or **hold** based on which aggregate weight (buy vs. sell vs. hold) is greatest;
- **(d)** a **regime classification module** that continuously evaluates a volatility metric of the data stream (or streams) and labels the operating regime as **High Volatility** if the volatility metric exceeds a high threshold value stored in the system (and as **Low Volatility** if the metric falls below a low threshold

value), the high and low threshold values being concrete numeric levels derived from historical volatility distribution (for example, the system sets *HIGH_THRESHOLD = 1.5%* daily change and *LOW_THRESHOLD = 0.3%* daily change as default values); this regime classification module dynamically adjusts at least one parameter of the pattern analysis engine based on the regime (such as raising the pattern classification Coherence threshold during High Volatility regimes to reduce false positives);

- **(e)** a **processor and memory** configured to implement the modules (a)–(d) and to execute the pattern analysis engine in discrete time steps keyed to prime numbers, such that *fundamental pattern state updates occur only at prime-indexed intervals* while intermediate data points are buffered for the next prime interval update, thereby preventing periodic feedback loops; and
- **(f)** an output interface that generates a real-time **trading signal or alert** based on the identified pattern category or collapse event, the consensus direction from the fusion module, and the current market regime, wherein the output signal is generated only when predetermined confidence conditions are satisfied (e.g. the fused signal confidence exceeds a set threshold like **70%**, which may be further increased to **80%** during High Volatility regimes as adjusted in (d)).

**Overall,** the system of claim 1 is characterized by the utilization of specific mathematical thresholds and update rules to improve the reliability of pattern detection and prediction in computerized financial trading, the system thereby providing a technical improvement in how trading signals are generated by dynamically adapting to detected pattern coherence levels, abrupt state changes, cross-asset influences, and volatility regimes.

## Independent Method Claim (Revised Draft)

- **1.** A **computer-implemented method for detecting emergent patterns and anomalies in time-series data** in real time, the method comprising the steps of:

- **Ingesting data** from at least one streaming source of sequential data points, the data representing a variable observed over time in a computing system memory;

- **Calculating entropy and normalized coherence:** at each prime-indexed time step (2nd, 3rd, 5th, 7th, … data point), computing a Shannon entropy *H* of the sequence of the most recent N data points, and then computing a **Coherence value = 1 – (H / H<sub>max</sub>)** ***for that sequence, where H<sub>max</sub> is a reference entropy for a uniform distribution over the observed value range; in tandem, computing a **Stability value = 1 – (transition_rate / max_transitions***), where *transition_rate* is the fraction of data point-to-point changes in the recent sequence and *max_transitions* is a predefined maximum possible transition rate;
- **Classifying pattern state:** comparing the Coherence and Stability values to a set of stored threshold pairs corresponding to defined pattern states, and if the Coherence and Stability meet or exceed a particular threshold pair, assigning the current sequence a **pattern classification** label; wherein the pattern classifications and threshold pairs include at least:
  - a **NullState** classification when Coherence $\geq$ **0.95** and Stability $\geq$ **0.90**, indicating a stable equilibrium state,
  - an **AllFlip** classification when Coherence $\geq$ **0.90** and Stability $\geq$ **0.85**, indicating a sustained alternating pattern, and
  - a **RandomNoise** classification when Coherence $\leq$ **0.30** and Stability $\leq$ **0.20**, indicating an absence of any discernible pattern (near-total randomness),

along with one or more intermediate pattern classifications for values between these extremes;

- **Detecting collapse events:** monitoring changes in the Coherence value on a step-by-step basis and **triggering an alert** if a drop in Coherence exceeds a predefined threshold **ΔC** within a short span of successive steps, wherein in an example configuration **ΔC = 0.20** (a 20% decrease in coherence) and the span is **within 5 consecutive data intervals**; responsive to detecting such a coherence collapse event, flagging the current pattern classification as **unstable** and prompting an adjustment in subsequent analysis (including optionally skipping or shortening step 3 for the next few intervals to allow recalibration);
- **Fusing multi-stream signals (if applicable):** if data from multiple sources are being ingested, computing at each prime-indexed step pairwise correlation values among the sources and updating a **weighted signal vote** by multiplying each source's latest pattern signal by a weight proportional to its correlation with other sources; summing weighted "buy", "sell", and "hold" signals across all sources to determine a dominant suggested action (buy, sell, or hold) by whichever sum is greatest, and calculating a combined confidence score for that action (boosting the confidence when multiple sources agree, for instance by a factor that increases the confidence by 30% of the cross-source agreement rate);
- **Adapting to market regime:** evaluating a volatility metric over a recent window (for a financial price series, using metrics like Average True Range or standard deviation) and comparing it against preset **High** and **Low volatility thresholds** stored in the system (specific numeric examples being a high-volatility threshold of 1.5% daily change and a low-volatility threshold of 0.3% daily change); labeling the current regime as High, Medium, or Low volatility, and adjusting at least one parameter of the pattern classification or collapse detection algorithm accordingly – **for example,** in High volatility regimes increasing the collapse alert threshold ΔC or requiring higher Coherence to classify a stable pattern, to account for noisier fluctuations;
- **Outputting a decision or alert:** generating an output signal from the computing system based on the results of the above steps, the output signal including at least the identified pattern classification (from step 3), any collapse event alert (from step 4), and for multi-source data the consensus action (from step 5), along with an indication of the confidence level and current volatility regime. This output can be used to **automatically execute a trade**, raise an alert to human operators, or feed into further processing in a trading platform.

In the method of claim 1, the use of **specific mathematical formulas and thresholds** (including the entropy-based coherence calculation, the ≥0.90/0.85 etc. pattern criteria, the ΔC > 0.20 collapse trigger, and fixed volatility breakpoints) is configured to provide a **technical improvement** in pattern recognition accuracy and timeliness, as the system dynamically adjusts to significant changes and varying market conditions rather than relying on static or purely qualitative analysis. The prime-number-based update scheduling in step 2 ensures that the computations avoid harmonic periodicities, thereby reducing systemic bias and improving the robustness of the detected patterns.

*(Each of the above steps is described in enabling detail in the specification, with pseudocode examples and exact parameter values to illustrate one embodiment. Variations and optimizations (e.g., using different threshold values or additional pattern categories) are also discussed, ensuring that a skilled practitioner can adapt the method to various data environments without undue experimentation.)*

# Prosecution Shield (Supporting Arguments & Strategy)

## 35 U.S.C. §101 Positioning – Technical Improvement Over Conventional Systems

- **Not an Abstract Idea, but a Concrete Technological Solution:** The claimed invention is rooted in computer technology and **improves the functioning of computer-based financial analysis systems**. Unlike an abstract mental process or a generic algorithm, it introduces specific structural and functional constraints (prime-number gating of computations, entropy-based coherence metrics, concrete threshold triggers) that tailor how the computer processes data in a way that **no human mind could perform mentally** with the same speed or precision. For instance, the prime-indexed update schedule and the real-time coherence calculations utilize computing capabilities to detect millisecond-level pattern shifts – a clear improvement in computer processing of streaming data.

- **Improvement to Performance and Accuracy:** Traditional trading algorithms might rely on fixed technical indicators or simple statistical thresholds. In contrast, our invention provides a **quantum-inspired analytic framework** that **significantly enhances predictive accuracy and responsiveness** [15] [16]. The specific combination of features – such as dynamic signal weighting based on correlation and adaptive threshold adjustment for volatility – results in faster recognition of regime changes and anomalies, thereby reducing latency in trading responses. This is an improvement to the technological field of algorithmic trading systems, akin to how an improved image processing algorithm was deemed patent-eligible due to enhancing computer functionality (drawing an analogy to cases like *McRO* or *Trading Technologies*, where domain-specific algorithmic improvements were found eligible). Here, the "feedback-loop resistant" prime-step scheduling and entropy-coherence mechanism make the computer system **more effective at processing financial data** than conventional systems.

- **Integrated, Specific Implementation (No Preemption of Abstract Ideas):** The claims are narrowly focused on a **particular arrangement of modules and computations**. By reciting **specific equations (e.g., Coherence = 1 – H/H_max)** and **specific threshold values** used in the analysis, the claims do not monopolize all ways of detecting patterns or all financial analytics – they claim one **specific engineered solution**. The inclusion of concrete numeric examples in the claims underscores that this is a practical application of mathematical principles, not a disembodied formula. The invention is **"applied" in the context of computerized trading systems** to solve problems like false signals and pattern collapses, thereby satisfying the **Alice/Mayo** step 2 as significantly more than an abstract idea (it implements an inventive concept in software to achieve better functioning of a trading platform).

- **Comparable to Recognized Computer Improvements:** The inventive features can be likened to improvements in computer capabilities recognized as patentable. For example, the dynamic re-weighting of signals and adaptive thresholding is analogous to improving a CPU's task scheduling or memory caching algorithm – it **optimizes how the computer handles data under varying conditions**. Similarly, the prime-number gating of updates is a novel timing control that improves system stability (preventing feedback oscillations), comparable to improvements in computer network protocols or time synchronization algorithms. These are **technical solutions** to technical problems (signal noise, pattern stability, computational resonance), not just business or financial abstractions.

## Enablement and Written Description Support – Key Points

- **Full Algorithm Disclosure:** The patent specification provides **extensive pseudocode, equations, and parameter examples** for each novel aspect of the invention (QFH, QBSA, signal fusion, regime adaptation, etc.). For instance, the specification explicitly shows how to calculate the entropy and coherence (with a formula) and gives numeric values for weighting factors (*k1 = 0.3*, *k2 = 0.2*, etc.) [7], as well as listing representative threshold values for different pattern types [2]. This level of detail ensures that a person of ordinary skill can **practically implement the claimed methods** without guesswork or undue experimentation.

- **Concrete Example Scenarios:** Multiple **illustrative examples** are described to demonstrate the system's operation. The draft includes scenarios such as detecting an *AllFlip* pattern in a synthetic alternating sequence, spotting a *VolatilityBreakout* when coherence suddenly drops by >0.2, and how the system reacts in a **High Volatility regime** by tightening its confidence requirements. By walking through these examples with actual numbers and outcomes, the specification confirms that the inventors had possession of the claimed invention and delineates the boundaries of the invention's behavior under various conditions.

- **Threshold Values Justified and Adjustable:** The inclusion of specific thresholds (e.g., 0.8 vs 0.75 for a pattern classification, or a 5-step collapse window) is accompanied by an explanation in the description that these values were chosen based on empirical tuning (e.g., through an internal **threshold_tuning.py** testbed or similar experiments) and can be adjusted for different applications. This both **demonstrates enablement** (by showing one workable set of values) and provides guidance for skilled readers to adjust the parameters if needed. The draft avoids pure functional claiming by pinning key steps to real values, which shows that the inventors have reduced the invention to practice in at least one concrete form.

- **Combining Modules is Fully Described:** Since the invention integrates several sub-systems (pattern analysis, collapse detection, multi-asset fusion, etc.), the specification carefully describes how these parts interface. There are diagrams and descriptions (e.g., the pipeline from QFH to QBSA to output [17]) illustrating data flow, ensuring the **cumulative system is enabled as a whole**. The roles of each module are clearly delineated, and their cooperative function is explained step-by-step. This prevents any concern that the claims are more than what's disclosed – every claim element is traceable to a portion of the write-up with sufficient detail.

- **Hardware and Software Implementation Details:** The draft also touches on implementation considerations, such as using **GPU acceleration** for the intensive calculations (mentioned as an option) [18], and memory management strategies (tiered memory for short-term vs long-term pattern storage in the engine). While the claims are at a higher level, the presence of such details in the spec gives the Examiner confidence that the inventors have thoroughly thought through **how to make and use** the invention on actual computing hardware. The description even references using prime-step updates to avoid clock drift and discusses how the system could be deployed in a live trading environment (network architecture, data latency handling), indicating there are no hidden implementation hurdles.

- **No Undue Experimentation Required:** In sum, a skilled practitioner (such as a quantitative finance engineer or a data scientist with knowledge of algorithms) would be able to read this patent and

**immediately start building a working prototype** of the system. All critical formulas and decision criteria are provided. Any tuning (like adjusting thresholds for a specific stock or market) falls within routine calibration, not inventive labor. Therefore, the application meets the enablement and written description requirements of §112 – the inventors have **fully conveyed how to practice the invention** and demonstrated possession of the inventive concepts through exhaustive detail.

## Examiner Interview Script (Key Discussion Points)

1. **Opening and Invention Overview:** *"Thank you for taking the time to meet. I'd like to start by briefly summarizing our invention. We have developed a quantum-inspired algorithmic trading system that fundamentally changes how pattern detection in financial data is done. Specifically, our system uses entropy-based coherence calculations, threshold-driven pattern classification, and a prime-number timing mechanism to identify market patterns and anomalies in real-time. This isn't just a theoretical concept – it's a concrete software system with specific modules and parameters, aimed at improving the performance of computer-based trading platforms."*

2. **Emphasize Technical Improvement:** *"One thing I want to highlight is the technical nature of the improvement. This is not an attempt to patent a financial idea or just an outcome; it's about how we process data within a computer. For example, by updating patterns only at prime-numbered intervals, our system avoids a lot of noise and feedback issues that plague high-frequency trading algorithms. That's a timing control at the system architecture level – a very technical solution. Also, using coherence (1–H/H_max) as a real-time signal provides a new kind of input to the computer's decision-making that wasn't available in standard systems. We've effectively expanded the toolbox of computing metrics for time-series analysis. These features collectively make the computer analyze data more efficiently and accurately."*

3. **Addressing §101 Concerns:** *"If there are any concerns about abstract ideas, I'd like to address them head-on. Our claims are carefully crafted to recite specific steps and components – for instance, calculating a Shannon entropy, computing a coherence value, checking it against a threshold (0.2 drop), etc. These are not broad ends-justifying-means claims; they are tied to a particular way of doing pattern detection on a computer. We believe this aligns well with the guidance for patent eligibility. Like in the* Electric Power Group *case, purely gathering and analyzing data can be abstract, but here we're doing it in a very inventive, constrained way that solves a technical problem in computer-driven trading. It's more akin to cases where a new algorithm improved computer functionality and was allowed. We'd be happy to walk through how each claim element maps to the novel technical behaviors in the system."*

4. **Novelty and Non-Obviousness (if raised):** *"We'd also like to point out where the novelty lies, in case that's useful. To our knowledge, no prior art combines these elements – the prime-step update gating is unique, and while entropy and correlation are known concepts, using them in this specific integrated fashion with set thresholds is new in the trading domain. Traditional systems use either simple technical indicators or black-box AI; none uses a hybrid quantum-inspired approach that is transparent and rule-based like ours. We have prior art search results that haven't found anything with a similar entropy-coherence metric driving trading signals. If needed, we can discuss specific prior art references and why they don't teach or suggest our combination of features."*

5. **Examiner's Input and Next Steps:** *"At this point, I'd appreciate your thoughts. Are there particular claim elements you feel need clarification or adjustment? We're open to amendments – for example, if specifying certain threshold values or ranges in the claims helps clarify the invention's scope, we are willing to do that*

*(we tried to include concrete numbers to underscore specificity). Our goal is to secure protection for the core innovative concepts (the pattern coherence/collapse detection and the prime-step control) while making sure the claims are clear. Please let us know if you have any suggestions or if there are any lingering concerns about eligibility or clarity that we can address."*

6. **Closing:** *"Thank you for the productive discussion. We believe this invention represents a genuine advance in fintech technology, and we're eager to work with you to ensure the claims clearly reflect its technical merit. We will follow up with any additional materials or claim amendments based on today's conversation. Thanks again for your time and insight."*

---


1 2 3 4 5 6 7 8 9 10 11 12 13 14 THEORY.md
https://github.com/SepDynamics/sep/blob/6feec479c95f866ceb17e3bb080e774442ae3e84/docs/THEORY.md

15 16 17 18 00_MASTER_INVENTION_SUMMARY.md
https://drive.google.com/file/d/1FXZ4-i1cU-8KgM_yZ89_AJ-tGODKYloj