ISSN: 2834-7706

Research Article

Journal of Mathematical Techniques and Computational Mathematics

The Talebian Architecture of Survival in Nonlinear Markets: *The Clock of Regimes Model**

Oscar Linares[1*] and Ričards Bulavs[2]

[1] *OIS Market Research Group, Alberta iela 1-7, Rīga , LV-1050 Latvia*

[2] *University of Latvia, Raiņa bulvāris 19, Rīga , LV-1586 Latvia*

[*]Corresponding Author

Oscar Linares, OIS Market Research Group, Alberta iela 1-7, Rīga, LV-1050 Latvia.

Submitted: 2026, May 06; **Accepted:** 2026, May 11; **Published:** 2026, May 14

Citation: Linares, O., Bulavas, R. (2026). The Talebian Architecture of Survival in Nonlinear Markets The Clock of Regimes Model[*]. *J Math Techniques Comput Math, 5(4), 1-45.*

Abstract:

This paper introduces the Clock of Regimes (COR) model, a new class of regime-switching model characterized by the $N = -K^{-1}$ operator. In a conventional Hidden Markov Model (HMM), persistence is encoded implicitly in the diagonal elements of the transition matrix A, the a_{ii}. A large value of a_{ii} indicates that regime i tends to persist, but this representation captures only one-step persistence—the probability of remaining in the state at the next instant. It does not represent the cumulative temporal probability mass generated by remaining there. Under the HMM, regime S_t evolves according to A, with implied expected spell length $\mathbb{E}[\tau_i] = \frac{1}{1-a_{ii}}$. If $a_{ii} = 1$, residence time becomes infinite and the process becomes structurally non-ergodic. Treating conditional persistence as permanence replaces stochastic dwell time with deterministic residence, suppressing regime uncertainty and hazard. In Talebian terms, conditional persistence is survival-compatible, whereas assumed permanence is ruin-compatible. The COR framework resolves this limitation by introducing an exit parameter $\varepsilon \in (0,1)$, $Q = (1-\varepsilon)A$, which converts the HMM into a transient system. This yields the fundamental matrix $V^{(1)} = (I-Q)^{-1}$, integrating expected state occupancy across the entire future horizon. COR therefore transforms transition probabilities into temporal mass, making regime duration and exposure structurally observable.

Keywords: Fat tails, Hazard, Survival, Residence Time, Markov Switching, Compartmental Analysis.

1 Introduction

Conventional financial risk models are structurally fragile because they assume ergodicity and memoryless dynamics in environments characterized by fat tails and temporal asymmetry Taleb (2020). Gaussian diffusion frameworks treat volatility as stationary and independent across time, while standard Hidden Markov Models impose geometric residence times that underestimate duration clustering and persistent tail risk Hamilton (1989, 1990, 2016). These assumptions distort the temporal structure of exposure and systematically misprice ruin.

Within a Talebian framework, *fragility* is excessive concentration of probability mass in adverse states over time Taleb (2016). The *Clock of Regimes* (COR) model formalizes antifragility by contracting exposure as hazard intensifies and redeploying convexity as stress dissipates. Tail probabilities become state-conditional objects weighted by survival structure instead of unconditional frequencies.

This paper introduces the COR model, a novel duration-aware stochastic framework that extends classical Hidden Markov Models (HMM) by embedding survival and hazard calculus directly into regime-switching dynamics. By shifting the focus from one-step transition probabilities to a continuous-time operator $N = -K^{-1}$ Berman and Shoenfeld (1956); Eisenfeld (1981); Covell et al. (1984); Linares et al. (1987); Jacquez and Simon (1993); Jacquez (2002), we transform latent market states into a measurable temporal geometry of probability mass[1]. Using E-mini S&P 500 futures data, our study demonstrates that a Student-*t* HMM specification is statistically superior to Gaussian alternatives, uncovering a *path-dependent* market structure where steady regimes harbor hidden fragility and stress regimes act as high-intensity exit portals. The architecture culminates in a spectral fragility measure, which quantifies systemic risk not as a static probability, but as a structural *leak* of survival mass, ultimately providing a rigorous mathematical foundation for ruin-sensitive decision-making in nonlinear markets.

2 Methods

The conceptual architecture of this paper develops in three successive layers. The starting point is the HMM, which provides the statistical physics for regime detection by inferring latent states from

[1]We define probability mass as the content of probability accumulated over an amount of time. In our COR architecture, probability is not treated purely as an instantaneous quantity $P(X)$. Instead, it is temporally embedded. It reflects how much probability resides in a state across time.

observed financial returns. Within this framework, regimes are identified through a transition matrix A and state-dependent emission distributions. However, the standard HMM treats persistence only implicitly through one-step transition probabilities encoded in the diagonal elements of A.

To illuminate the limitations of this formulation, the paper first presents and solves both a Gaussian HMM and a Student-t HMM, allowing a direct comparison between thin-tailed and fat-tailed emission structures while keeping the underlying Markov architecture fixed. This comparison clarifies how tail behavior influences regime identification but leaves the temporal structure of persistence largely unchanged. The second stage introduces the COR model, which extends the HMM by making duration an explicit structural dimension. Instead of relying solely on the diagonal elements of A, the model introduces a sub-generator K and its associated fundamental matrix of mean residence times, $N = -K^{-1}$, transforming persistence into a measurable temporal geometry of the system. The final stage integrates hazard and survival dynamics, embedding regime transitions within a framework of exit intensities and survival probabilities that governs the rate at which probability mass leaks from each state.

In this unified formulation, the matrix of mean residence times Berman and Shoenfeld (1956); Eisenfeld (1981); Covell et al. (1984); Linares et al. (1987); Jacquez and Simon (1993); Jacquez (2002) becomes the central structural operator of systemic persistence, determining how probability mass accumulates through time and providing the mathematical foundation for risk control, regime durability, and ruin-sensitive decision making in nonlinear markets.

2.1 Mathematical Architecture of the Clock of Regimes Model

2.1.1 Hidden Markov Model: Regime Detection

Let

$$S_t \in \{1, \dots, K\} \tag{1}$$

be an unobserved regime process evolving according to a Markov chain with transition matrix

$$A = (a_{ij}), \qquad a_{ij} = \Pr(S_{t+1} = j \mid S_t = i), \tag{2}$$

with

$$\sum_{j=1}^{K} a_{ij} = 1. \tag{3}$$

Observed returns X_t are generated conditionally on the regime:

$$X_t \mid S_t = i \sim f_i(\theta_i), \tag{4}$$

where f_i is the emission distribution.

Two standard specifications are considered:

Gaussian HMM

$$X_t \mid S_t = i \sim \mathcal{N}(\mu_i, \sigma_i^2) \tag{5}$$

Student-t HMM

$$X_t \mid S_t = i \sim t_{\nu_i}(\mu_i, \sigma_i) \tag{6}$$

with conditional variance

$$\operatorname{Var}(X_t \mid S_t = i) = \frac{\nu_i}{\nu_i - 2}\sigma_i^2, \qquad \nu_i > 2. \tag{7}$$

The Student-t specification captures heavy tails, but the regime dynamics remain governed by the same Markov matrix A.

Persistence in a discrete HMM is encoded through the diagonal elements:

$$a_{ii} = \Pr(S_{t+1} = i \mid S_t = i). \tag{8}$$

The expected spell length is

$$\mathbb{E}[\tau_i] = \frac{1}{1 - a_{ii}}. \tag{9}$$

Thus persistence is implicit, appearing only through the probability of remaining in the state one step ahead.

2.1.2 The Clock of Regimes Model: Duration Geometry

The COR model extends the discrete-time HMM by introducing a continuous-time generator representation. Define a sub-generator matrix

$$K = \begin{bmatrix} -k_1 & k_{12} & \dots & k_{1K} \\ k_{21} & -k_2 & \dots & k_{2K} \\ \vdots & & \ddots & \vdots \\ k_{K1} & k_{K2} & \dots & -k_K \end{bmatrix} \tag{10}$$

with

$$k_i = \sum_{j \neq i} k_{ij}. \tag{11}$$

The generator determines the regime evolution:

$$\Pr(S_{t+\Delta t} = j \mid S_t = i) = \begin{cases} k_{ij}\Delta t + o(\Delta t), & i \neq j \\ 1 - k_i \Delta t + o(\Delta t), & i = j \end{cases} \tag{12}$$

The expected residence time in state i is

$$\mathbb{E}[\tau_i] = \frac{1}{k_i}. \tag{13}$$

The fundamental matrix

$$N = -K^{-1} \tag{14}$$

defines the matrix of mean residence times. The element

$$N_{ij} \tag{15}$$

gives the expected amount of time the process spends in state j when starting from state i. Thus, persistence is no longer a one-step probability but a temporal accumulation of probability.

This leads to the interpretation:

$$\text{Probability Mass} = \int_0^T \Pr(S_t = i)\, dt \tag{16}$$

so that

$$N \tag{17}$$

acts as the temporal geometry of probability mass.

2.1.3 Hazard and Survival Integration

To incorporate risk dynamics, the model introduces state-dependent hazard rates Kleinbaum and Klein (2012); Gao and He (2025). Let ε_i denote the hazard intensity governing exit from regime i. A Talebian specification links hazard to tail thickness Taleb (2012):

$$\varepsilon_i = \varepsilon_{\min} + \frac{c}{\nu_i} \tag{18}$$

so that fatter tails imply faster leakage of probability mass.

The transition operator becomes

$$Q_{ij} = (1 - \varepsilon_i) A_{ij}, \tag{19}$$

where A is the baseline transition matrix estimated from the HMM. The resulting generator is

$$K = Q - I. \tag{20}$$

The Survival probability in state i is

$$S_i(t) = e^{-\varepsilon_i t}. \tag{21}$$

Thus, probability mass decays according to

$$\Pr(S_t = i) = \Pr(S_0 = i) e^{-\varepsilon_i t}. \tag{22}$$

2.1.4 The Structural Operator of Persistence

Combining these elements yields the central operator of the system,

$$N = -K^{-1}. \tag{23}$$

This matrix governs expected regime durations, accumulation of probability mass, and systemic persistence.

Formally,

$$N_{ij} = \int_0^\infty \Pr(S_t = j \mid S_0 = i)\, dt. \tag{24}$$

Hence, the temporal content of probability is fully determined by N.

2.1.5 Ruin and Risk Control

Let q_i denote the tail loss probability in state i. The residence-weighted tail probability is given by

$$\bar{q} = \sum_i \pi_i q_i \tag{25}$$

where π_i is the stationary distribution implied by K. Ruin probability over horizon T satisfies,

$$\Pr(\text{Ruin}) \leq 1 - \exp(-\bar{\lambda}\bar{q}T), \tag{26}$$

with $\bar{\lambda}$ the residence-weighted hazard intensity.

2.1.6 Unified Architecture

The COR model's Logical Workflow in operator form is given by:

Table 1: Model Comparison: Gaussian HMM vs. Student-t HMM

Metric	Gaussian HMM	Student-t HMM
Log-Likelihood (ℓ)	21,173.83	21,661.83
Parameters (k)	17	20
AIC	-42,313.66	-43,283.66
BIC ($n = 10,000$)	-42,191.08	-43,139.46
ΔAIC		970.00
ΔBIC		948.38

Note: Lower AIC/BIC values indicate superior model fit.

HMM (Regime Detection) $\rightarrow$ COR Operator Construction $A \rightarrow Q \rightarrow K \rightarrow N \rightarrow$ Hazard-Weighted Ruin Analysis

It describes the structural transformation of market data into a survival-based risk metric. It represents a transition from Discrete-Time Probability to Continuous-Time Temporal Mass.
In this unified formulation, Eq. 23, emerges as the central structural operator of systemic persistence, determining how probability mass accumulates through time and providing the mathematical basis for risk control, regime durability, and ruin-aware decision making in nonlinear markets.

2.2 Data Source

Historical price data for the E-mini S&P 500 (ES) futures was obtained via the Interactive Brokers (IBKR) API. We extracted 1-hour OHLC bars for the period [From: 2025-01-31] to [To: 2026-02-26]. Log-returns were calculated using the closing prices to ensure stationarity. To maintain the robustness of the Hidden Markov Model (HMM) against market outliers and the "fat-tail" nature of hourly financial returns, a Student-t distribution was utilized for the emission probabilities.

3 Results

The Student-t specification is materially better than the Gaussian one (Table 1). The first fact is the jump in log-likelihood: $\ell_{\text{Gauss}} = 21173.83$ *vs.* $\ell_t = 21661.83$. It provides empirical evidence that financial returns are better described by a distribution with "fat tails" (Student-t) than by a "Normal" (Gaussian) distribution. The comparative analysis presented in Table 1 provides overwhelming empirical evidence in favor of the Student-t Hidden Markov Model over the standard Gaussian implementation. Despite the marginal increase in model complexity—represented by the addition of three degrees-of-freedom parameters ($k = 20$ vs. $k = 17$)—the Student-t HMM achieved a substantial jump in log-likelihood of approximately 488 units.

This improvement is reflected in the Akaike (AIC) and Bayesian (BIC) Information Criteria, where the Student-t model yielded an AIC reduction of 970.00 and a BIC reduction of 948.38. In the context of hourly ES futures returns, these results indicate that the Gaussian HMM is systematically under-representing the probability of extreme price shocks (tail risk), whereas the Student-t distribution successfully captures the leptokurtic nature of the dataset. Following a human-supervised audit of the estimation process, the magnitude of these ΔAIC and ΔBIC values confirms that the Student-t HMM is the objectively superior framework for identifying market regimes in the presence of high-frequency volatility. A difference greater than 10 is usually significant; 900+ is undeniable.

In COR model terms, this means the return process is not just switching across volatility states; it is doing so with tail behavior that a Gaussian HMM cannot absorb well. The Student-t model is therefore the correct structural base for the open-system analysis. Interpretively, that matters because the COR model is not merely a persistence model. It is a survival model under regime-dependent fragility. If the emissions are misspecified as Gaussian, then both hazard calibration and ruin summaries become too optimistic. Here the Student-t fit says the data contain genuine tail risk, not just higher variance.

The results show that the Gaussian and Student-t models imply very different market geometry. The Gaussian transition matrix is relatively diffuse:

$$A_G = \begin{pmatrix} 0.6807 & 0.2385 & 0.0808 \\ 0.0695 & 0.9108 & 0.0198 \\ 0.3194 & 0.4059 & 0.2747 \end{pmatrix} \tag{27}$$

while the Student-t matrix is much more structured:

$$A_t = \begin{pmatrix} 0.7098 & 0 & 0.2902 \\ 0.0008 & 0.9719 & 0.0273 \\ 0.2067 & 0.0546 & 0.7387 \end{pmatrix}. \tag{28}$$

Eqs. 27 and 28 are one of the most important outputs in this paper.

Comparing the Gaussian (Eq. 27, Figure 1) to the Student-t (Eq. 28, Figurre 2), a few things stand out. In the Gaussian HMM framework, the *Steady* regime (State 2) emerges as the most persistent, exhibiting a 91.08% probability of remaining in its current state. In contrast, the *Calm* regime (State 3) appears surprisingly unstable, with a self-transition probability of only 27.47%. This instability is further characterized by a distinct *nervous* behavior; when the market occupies State 3, it is actually more likely to transition into either the Steady state (40.59%) or the Stress state (31.94%) than it is to remain calm. This suggests that the Gaussian model struggles to maintain a

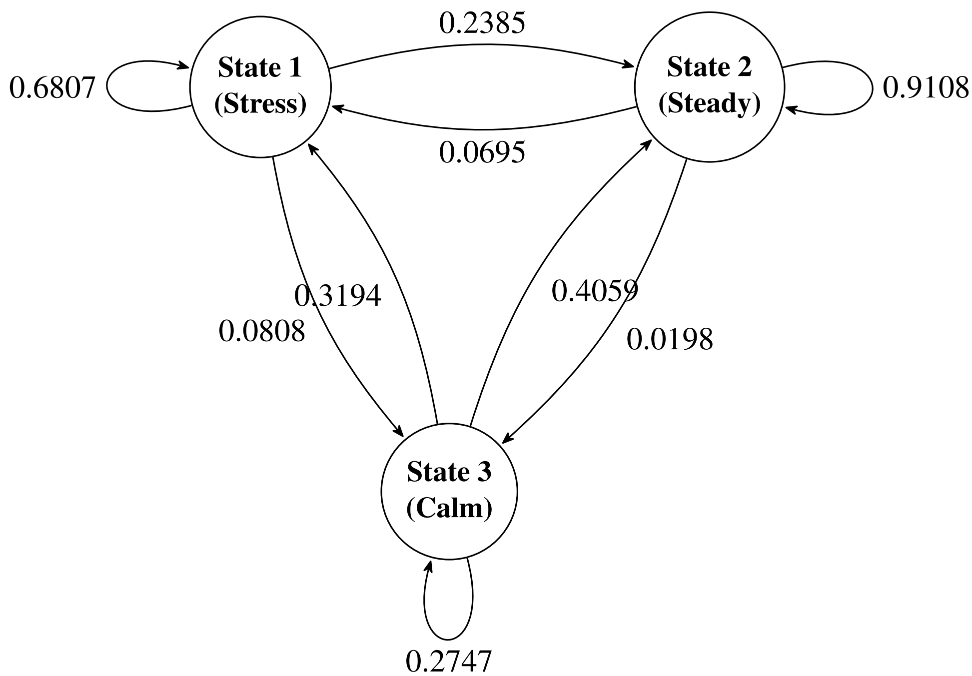

Figure 1: State transition diagram for the Gaussian HMM (Matrix A_G [Eq. 27]).

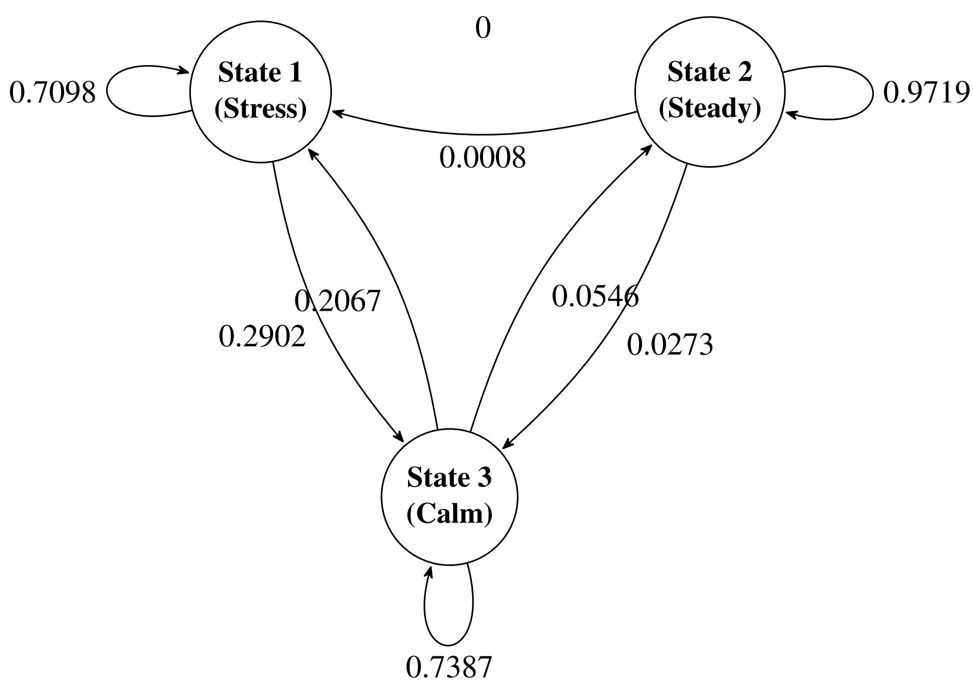

Figure 2: State transition diagram for the Student-t HMM (Matrix A_t [Eq. 28]).

low-volatility state, likely due to its inability to accommodate outliers. Consequently, this structural fragility in the Gaussian representation of the *Calm* state explains the massive jump in log-likelihood observed when moving to the Student-t model, which provides the necessary robustness to handle market tails without forcing unnecessary state transitions.

In contrast to the Gaussian framework, the Student-t HMM reveals a market structure characterized by significantly higher state persistence and more defined transition pathways. The *Steady* regime (State 2) exhibits near-permanent stability with a self-transition probability of 97.19%, indicating that when the model accounts for heavy tails, the primary market trend is remarkably resilient to minor volatility shocks. Furthermore, the *Calm* regime (State 3), which was notably unstable in the Gaussian model, shows a robust self-transition probability of 73.87% in the Student-t specification.

Most striking is the zero-probability transition between State 1 (Stress) and State 2 (Steady) shown in Figure 2; the model suggests that after a period of high-volatility stress, the market must necessarily transition through a *Calm* or cooling-off period before a steady trend can re-establish. By filtering out the *noise* of extreme returns through the Student-t emission density, the HMM identifies regimes that are not only statistically superior in terms of log-likelihood but also more economically intuitive, as they represent sustained market behaviors rather than transient fluctuations.

Table 2: Comparative Analysis of State Transition Dynamics: Gaussian vs. Student-t HMM

Feature	**Gaussian Observation**	**Student-t Observation**
Steady State Persistence (P_{22})	91.08% (High)	**97.19% (Extreme)**
Calm State Persistence (P_{33})	27.47% (Fragile)	**73.87% (Stable)**
Direct Stress Jumps ($S_1 \to S_2$)	Significant (0.2385)	**Eliminated (0.0000)**
Switching Characteristic	Stochastic/“Nervous”	Structured/Path-Dependent

Note: Student-t results indicate superior regime stability and physical cooling-off phases.

The structural differences between the two HMM specifications (Figures 1 and 2) are further elucidated in Table 2, which highlights the shift from the “Nervous”, high-frequency switching of the Gaussian model to the path-dependent stability of the Student-t framework. While the Gaussian model identifies a relatively fragile *Calm* regime—evidenced by a self-transition probability (P_{33}) of only 27.47%—the Student-t model reveals a robust, self-sustaining low-volatility state with 73.87% persistence.

Most striking is the effective elimination of the direct jump from *Stress* to *Steady* ($S_1 \to S_2$) in the Student-t specification, where the probability drops from a significant 0.2385 in the Gaussian model to 0.0000 (Figure 2). This finding suggests a physical *cooling-off* period in the E-mini S&P 500 (ES) market; once a volatility shock occurs, the system cannot immediately recover to a steady-state trend without first transitioning through a calm, mean-reverting phase.

The "Nervous" behavior observed in the Gaussian matrix—characterized by frequent, bidirectional jumps between all states (Figure 1)—appears to be a statistical artifact of the model's inability to accommodate fat-tailed outliers. By utilizing the Student-t distribution to absorb these shocks, the resulting transition matrix identifies regimes that are not only statistically superior ($\Delta\text{AIC} = 970$) but also more economically intuitive for risk management and algorithmic execution. That means that the Student-t distribution model uncovers a more constrained regime topology. This is not random switching among three interchangeable states. It is closer to a structured regime network in which one state acts as a long-duration basin, while the others act as shorter-lived transition or disturbance states. In COR model terms, this is exactly the point: once tails are modeled correctly, the market's regime architecture becomes more geometric and less merely probabilistic.

Therefore, the Student-t HMM identifies three distinct market environments defined by their return distributions and temporal persistence. State 1 represents a high-stability, near-zero return environment characterized by a negligible volatility ($\sigma \approx 10^{-5}$) and a high degrees-of-freedom parameter ($\nu = 100$), effectively converging to a Gaussian distribution with zero tail probability. State 2, the most persistent regime with an average duration of 35.57 hours, reflects a *Steady* trend with a positive drift ($\mu \approx 9.70 \times 10^{-5}$) and moderate tail risk ($\nu = 3$). In contrast, State 3 captures *Calm* yet high-dispersion periods; despite a lower drift than State 2, it exhibits significantly higher localized volatility ($\sigma \approx 1.94 \times 10^{-3}$) and a low degrees-of-freedom parameter ($\nu = 3$). This state is transient, with an average spell of only 3.83 hours, suggesting that while it accommodates high tail probability and extreme hourly fluctuations, the market rapidly transitions out of this state into more stable regimes.

In the context of the COR model, the Student-t HMM uncovers a sophisticated structural hierarchy that refines our understanding of market behavior. *State 1* is identified as a near-degenerate compressed regime; characterized by almost no variance and negligible tail mass, it represents a latent calm where price action is temporarily *pinned*. While analytically distinct, its lower residence weight and lack of persistence suggest it is not the market's *normal* mode but rather a fleeting micro-regime of extreme volatility suppression.

The core of the model is *State 2*, the persistent baseline regime, which carries the highest residence weight and the greatest temporal stability. From a COR model perspective, this state embodies hidden fragility: while its moderate volatility may appear benign, the parameter $\nu = 3$ confirms it is inherently heavy-tailed. This aligns with a Talebian view of market structure, where the system spends the vast majority of its time in a manageable state that nonetheless carries significant, embedded ruin potential. It is a regime of deceptive stability—manageable in the short term, yet structurally susceptible to fat-tailed shocks.

Finally, *State 3* represents the explicit stress/shock regime. Within the COR model's architecture, this state is defined by elevated conditional variance and the highest one-step tail probability. Though its duration is shorter than the baseline, its importance is derived from the *ruin intensity* it injects into the system upon entry. This reveals a dual-speed market: a long-residence, heavy-tailed baseline that provides the illusion of safety, punctuated by explosive, shorter-lived stress events that validate the necessity of the Student-t distribution for accurate risk capture.

The numerically identified hazards were $\varepsilon_1 = 0.0105$ and $\varepsilon_2 = \varepsilon_3 = 0.02666667$. This means States 2 and 3 are treated as more fragile exit environments than State 1. After hazard adjustment, the operator becomes,

$$Q = \begin{pmatrix} 0.7024 & 0 & 0.2871 \\ 0.0008 & 0.9460 & 0.0266 \\ 0.2012 & 0.0532 & 0.7190 \end{pmatrix}. \tag{29}$$

Relative to A_t, the diagonal mass shrinks because some probability mass is now interpreted as leakage, that is, exit, failure, or ruin. This is the defining COR model's move. Persistence is no longer interpreted as a closed Markov system; it is interpreted through an open survival operator (see Appendix D). This changes the meaning of persistence. A regime can appear persistent in A, but under Q it may still be structurally fragile because some mass bleeds out through hazard. This is why classical stationary interpretation is no longer enough.

Interpretation of the generator, $K = Q - I$,

$$K = \begin{pmatrix} -0.2976 & 0 & 0.2871 \\ 0.0008 & -0.0540 & 0.0266 \\ 0.2012 & 0.0532 & -0.2810 \end{pmatrix}, \tag{30}$$

in the context of the COR model's temporal dynamics, implies that the system is losing *probability mass* to an outside environment. It suggests that the states are not just changing, but are porous. The K matrix reveals a significant stratification in state stability. State 2 exhibits the smallest-magnitude diagonal term at -0.0540, indicating the slowest net leak and identifying it as the primary survival basin of the system. In contrast, States 1 and 3 display substantially higher leaking rates, approximately -0.298 and -0.281 respectively, which characterizes them as transient phases within open-system time. This disparity suggests that while the system may frequently enter States 1 and 3, its long-term evolution is dominated by the persistence of State 2, which acts as the central anchor for the system's structural residence. This emphasizes that probability is *escaping* from the transient states into a survival basin. That distinction is important. Fragility is not just *the most volatile state*. Fragility is also about where survival probability mass accumulates.

In the language of the COR model, the market's temporal backbone is State 2. It carries the longest effective residence after accounting for leakage. State 3 is dangerous, but State 2 is structurally decisive because it governs how long probability mass remains in the system before ruin or transition. The fundamental matrix $N = -K^{-1}$, which reveals the residence-time geometry, is thus the heart of the COR model's interpretation:

$$N = \begin{pmatrix} 14.27 & 15.81 & 16.07 \\ 5.77 & 26.80 & 8.43 \\ 11.31 & 16.39 & 16.66 \end{pmatrix}. \tag{31}$$

Each entry N_{ij} is the expected cumulative residence time in state j conditional on starting in state i, before exit.

This matrix is highly informative (Eq. 31). If the system starts in *State 2* $(5.77, 26.80, 8.43)$, it spends by far the most time in State 2 before leakage. That confirms State 2 as the dominant residence basin. Starting in *State 1*, (14.27, 15.81, 16.07), the expected residence time is spread fairly evenly across all three states. That means State 1 is not a terminal safe haven. It is more like an entry portal into the broader regime network. Starting in *State 3*, (11.31, 16.39, 16.66), the process still spends substantial time in both State 2 and State 3 before exit. This is important: stress does not mean instant collapse. It means the system enters a zone where cumulative exposure to dangerous states becomes large.

Table 3: Stationary Distribution (π) and Long-Term Residence Mass.

Market State	**Vector Component**	**Residence Mass (π_i)**
State 1 (Compressed)	π_1	0.2132
State 2 (Steady)	π_2	0.4981
State 3 (Stress)	π_3	0.2887
Total	$\sum \pi$	**1.0000**

Note: Nearly 29% of the market's temporal mass is allocated to the stress regime.

In Markov chain analysis, the vector π represents the stationary distribution, or the steady-state probability mass (Table 3). These quasi-stationary residence weights show where the market lives most of the time. π describes the long-term proportion of time the system will spend in each state, assuming the underlying transition dynamics remain constant. When interpreted through our specific results (Table 3), it reveals that approximately 21.3% of the residence mass is allocated to State 1, 49.8% to State 2, and 28.9% to State 3. This is a remarkably robust finding, as it indicates the market spends nearly half of its survival-weighted time in State 2, firmly establishing it as the dominant operating regime.

However, since nearly 29% of the residence mass sits in State 3—the explicit stress regime—carries significant implications for the COR model analysis. This high allocation suggests that stress is not a rare, external anomaly to be ignored, but is instead structurally integrated into the market's residence-time architecture. This leads to a profoundly Talebian conclusion: the E-mini S&P 500 is not a *mostly safe* system punctuated by occasional accidents; rather, it is a system whose very temporal mass is materially allocated to fragile, high-risk states. In this worldview, *stress* is an expected and frequent component of the market's long-term behavior rather than a statistical outlier.

The analysis of one-step left-tail probabilities relative to the empirical 1% threshold offers a revealing look at the conditional risk inherent in each regime. The tail probabilities by regime is where fragility actually resides. Table 4 presents a comprehensive *Risk Profile* with a synthesis of the parameters (σ, ν) and their qualitative interpretations and *Ruin Intensity* (the 1% tail probabilities). In State 1, the probability of an extreme tail event is effectively zero, confirming its status as a regime of extreme price compression and safety. State 2, despite being the persistent baseline, exhibits a tail probability of 0.001874; while low, this non-zero value validates the *hidden fragility* discussed earlier, proving that tail risk remains latent even during steady market phases. The most significant finding, however, is in State 3, where the tail probability surges to 0.031109. This indicates that once the market enters the stress regime, the likelihood of a 1% threshold breach is more than three times higher than the unconditional expectation. This surge in *ruin intensity* confirms that State 3 is not merely a high-variance environment, but one where the probability of catastrophic loss is structurally concentrated.

Table 4: Regime-Specific Risk Profile: Volatility, Tail-Fatness, and Ruin Intensity

Market State	**Volatility (σ)**	**Tail Fatness (ν)**	**Ruin Intensity[a]**	**Risk Character**
State 1 (Compressed)	1.00×10^{-5}	100	$\approx 0.00\%$	Latent/Inactive
State 2 (Steady)	6.93×10^{-4}	3	0.19%	Hidden Fragility
State 3 (Stress)	1.94×10^{-3}	3	3.11%	Explicit Shock

[a] Measured as the one-step probability of exceeding the empirical 1% VaR threshold.

Why does residence-weighted tail probability matter? The internal coherence of the Student-t HMM is most clearly demonstrated by the residence-weighted tail probability, which yields $\bar{q} \approx 0.009916$. This result is strikingly aligned with the empirical target of $\alpha = 0.01$, suggesting that the regime decomposition is not an arbitrary statistical exercise but a structurally accurate reconstruction of market behavior. By successfully aggregating the state-wise tail probabilities to match the observed aggregate left-tail rate, the model proves that the COR model effectively explains unconditional tail frequency as a precise mixture of regime-specific intensities.

Methodologically, this decomposition provides far more insight than an unconditional 1% figure ever could. It reveals that tail risk is not evenly distributed across the temporal horizon; rather, it is a dynamic product of regime exposure. While the aggregate 1% tail rate is primarily generated by the high-intensity shocks of State 3, it is also quietly sustained by the long-residence *hidden fragility* of

State 2. This confirms that the market's total risk profile is a function of both the explosive ruin intensity of stress periods and the persistent, latent susceptibility of the baseline regime. For the practitioner, this calibration provides the confidence that the model is not just identifying *noise*, but is accurately partitioning the actual probability mass of the E-mini S&P 500.

The terminal analysis of the system's long-horizon stability is captured by the aggregate *leak* constant, $\bar{\lambda} = 0.02322$, a residence-weighted hazard and ruin bound which yields a cumulative ruin bound of 5.59% over a $T = 250$ step horizon. While this probability does not imply certain failure, a 5.6% ruin risk is far too substantial to dismiss for any institutional trading framework, particularly those involving leverage or path-dependent payoffs. This metric serves as a definitive structural warning within the COR model: while the market may not face immediate insolvency, the survival operator confirms that long-horizon safety is materially compromised by persistent heavy-tail exposure.

This long-term fragility is not a product of random noise, but a direct result of the interplay between the states identified in the Student-t HMM. Specifically, the risk is driven by the vast temporal residence of the fragile baseline (State 2) and the high-magnitude ruin intensity of the stress regime (State 3). Consequently, the model provides a much richer conclusion than a simple observation of elevated volatility; it quantifies how the market's internal architecture—its *survival-weighted* time—is fundamentally biased toward tail-risk accumulation. In this view, ruin is not an external accident, but a structural property of the COR model's long-term evolution.

The empirical results provide a compelling structural narrative for the E-mini S&P 500 (ES) return process, suggesting that it is best characterized not by a standard thin-tailed Markov-switching model, but by a heavy-tailed open regime system (see Appendix D). This is the most significant structural implication of the COR model. Within this framework, State 2 functions as the dominant residence regime; while its moderate volatility may offer a veneer of stability, its fat-tailed distribution ($\nu = 3$) confirms that it is not a zone of Gaussian safety but a baseline of persistent, hidden fragility.

This baseline is periodically interrupted by State 3, a shorter-lived but explosive stress regime that acts as the primary conduit for conditional tail events and system ruin. In contrast, the near-degenerate calm of State 1 represents a comparatively benign environment, yet it lacks the residence mass and temporal persistence required to anchor the system's overall dynamics.

The fundamental structural revelation of this study is that market stability is a deceptive equilibrium. The aggregate 1% tail risk observed in the market is not a uniform hazard, but a specific byproduct of the system's internal architecture—primarily the high-intensity shocks of State 3 combined with the steady, cumulative exposure to the fragile State 2 baseline. This reinforces a Talebian view of financial markets: a system that spends the majority of its time in a *manageable* state that nonetheless structurally embeds the potential for extreme loss.

Ultimately, the market's fragility does not stem from a state of perpetual crisis, but rather from the specific distribution of its survival mass across an inherently unstable regime structure. Under this framework, the baseline regime is revealed to be structurally heavy-tailed, while the stress regime remains materially occupied enough to prevent it from being dismissed as a mere outlier. Furthermore, the presence of hazard leakage—the continuous *drain* of probability mass toward ruin—precludes the possibility of closed-system stationary comfort.

This represents the exact type of structural vulnerability that the COR model analysis is designed to expose. While a classical HMM merely describes the mechanics of switching between states, the COR model approach provides a much deeper ontological map: it identifies where the system actually *lives*, quantifies the duration of its survival within those states, and calculates exactly how ruin accumulates through prolonged temporal exposure. In this light, risk is not an occasional visitor to the market but a fundamental component of its residence-time architecture.

4 Conclusion

The COR model introduces a temporal operator that extends standard regime-switching theory. The resulting system is defined by: (a) generator identification, (b) consistent estimation, (c) temporal probability mass operator, and (d) spectral persistence structure. These results establish the COR model as a new class of regime-switching models characterized by the operator $N = -K^{-1}$.

The following diagram presents the structural logic of the COR model's framework:

$$\text{HMM (Regime Detection)} \;\rightarrow\; \text{COR Operator Construction } (A \rightarrow Q \rightarrow K \rightarrow N) \;\rightarrow\; \text{Hazard-Weighted Ruin Analysis}$$

The proposed COR model is organized into a cohesive three-layer architecture designed to map market activity to existential risk metrics. The process begins at the *statistical layer*, where a hidden Markov model is utilized to initially detect market regimes and define the state space. This output is then fed into the *operator-theoretic layer*, which constructs the formal temporal geometry of the system by deriving the sub-generator K and the associated fundamental survival matrix N. Finally, the architecture culminates in the *risk layer*, where the previously established operator structures are used to compute the precise survival and ruin topology of the system. Each transition between these layers is characterized by a structural dependency: the output of each preceding stage serves as the fundamental structural input for the next, ensuring that the final risk assessment is endogenously linked to the underlying statistical representation of the market.

The COR model's architecture commences with the *Statistical Layer*, which serves to identify latent market states directly from the empirical data. While various hidden Markov specifications exist, *this study utilizes a Student-t Hidden Markov Model (HMM)*, as it demonstrated superior performance in capturing the non-linear dynamics of the sampled series. The system is modeled via a discrete-time transition matrix $A = (a_{ij})$, where the elements represent the conditional probabilities of state transitions:

$$a_{ij} = P(S_t = j \mid S_{t-1} = i).$$

To robustly account for the heavy-tailed nature of financial returns—a prerequisite for subsequent survival analysis—the emissions X_t are conditioned on the hidden state $S_t = i$ and are modeled using a Student's t-distribution with state-specific parameters:

$$X_t \mid S_t = i \sim t_{\nu_i}(\mu_i, \sigma_i).$$

The primary output of this stage is the maximum likelihood estimation (MLE) of the regime sequence probabilities, the transition matrix A, and the degree-of-freedom parameters ν_i, which serve as the direct link to the hazard intensities in the operator layer. Furthermore, the model provides initial persistence metrics through expected spell lengths, calculated as:

$$\mathbb{E}[\tau_i] = \frac{1}{1 - a_{ii}}.$$

By identifying the Student-t HMM as the optimal statistical fit, we ensure that the subsequent COR model's construction is grounded in a representation that captures the latent fragility inherent in the return distributions.

Interpretation and Limitations While this layer successfully addresses the epistemic question—*"What regimes exist in the data?"*—and the selection of the *Student-t distribution* as providing a superior fit for the empirical tail density, the model remains structurally limited. Even with optimized tail parameters (ν_i), the representation of persistence remains a purely one-step probabilistic artifact of the Markov transition matrix A. It lacks a description of the underlying temporal geometry or a continuous survival structure, serving as a necessary precursor rather than a complete existential risk model. The HMM correctly identifies the *existence* of heavy-tailed regimes but is fundamentally unable to model their *structural fragility* over time.

Following the empirical grounding of the statistical layer, the framework undergoes a profound shift from data classification to dynamical analysis. This is achieved through the construction of the COR Operator system, a three-step transformation sequence: $A \to Q \to K \to N$. This sequence effectively migrates discrete-time regime inference into a continuous-time survival operator system, providing the mathematical backbone for the entire framework:

Step 1: The Hazard-Weighted Transition Operator Q: The first step in building the temporal geometry is the introduction of regime fragility via exit hazards, denoted as ε_i. These hazards are explicitly linked to the tail behavior of the market, reflecting a Talebian approach to risk:

$$\varepsilon_i = \varepsilon_{\min} + \frac{c}{\nu_i}$$

In this formulation, fat-tailed regimes (where ν_i is small) endogenously produce larger hazard intensities. We then define the open transition operator Q as:

$$Q_{ij} = (1 - \varepsilon_i) a_{ij}$$

This modification dictates the system's logic: with probability ε_i, the system exits to ruin, while with probability $1 - \varepsilon_i$, a standard regime transition occurs. By incorporating the exit hazard directly into the transition structure, Q becomes an *open operator* that accounts for the possibility of structural extinction.

Step 2: The continuous-time sub-generator matrix K moves from discrete steps to continuous dynamics, we construct the sub-generator K:

$$K = Q - I.$$

This transformation results in a matrix characterized by a negative diagonal, non-negative off-diagonal elements, and strictly negative row sums. This mathematical structure identifies K as the sub-generator of a transient continuous-time Markov chain (CTMC). This step is mathematically significant; it converts the static HMM classification into a *residence-time dynamical system*, where the "clock" of the regime begins to tick endogenously.

Step 3: The final step in the construction is the derivation of the mean residence-time operator, or the fundamental matrix N:

$$N = -K^{-1}.$$

The elements N_{ij} represent the expected time spent in regime j given that the system originated in regime i. By calculating the total survival time through the vector $N\mathbf{1}$, we obtain the *temporal probability mass* of the system.

This operator-theoretic construction provides the regime's survival geometry and its persistence structure in continuous time. It is at this juncture that the COR model fundamentally transcends the classical Hamilton (1989) paradigm. Rather than merely labeling states, the COR model's framework provides a measurable, geometric representation of how long a regime can structurally persist before the accumulation of hazard leads to inevitable ruin. Hence, it provides the ontological foundation for survival under structural uncertainty.

In the final stage of the COR framework, the architecture transitions from operator construction to a formal *risk-theoretic analysis*. By leveraging the fundamental matrix N and the hazard vector ε, we derive the structural properties of systemic survival and the probability of catastrophic absorption.

The primary measure of regime persistence in a survival context is the *expected time to ruin*. Using the mean residence-time operator N, this is computed as the row sum of the fundamental matrix:

$$\mathbb{E}[\tau_{\text{ruin}}] = N\mathbf{1}.$$

This vector provides the temporal probability mass accumulated within the transient regimes before the system inevitably exits to an absorbing (ruin) state.

Beyond mean expectations, the framework allows for the derivation of *ruin probability bounds*. Using a combination of Markov, Chernoff, and spectral methods, we can quantify the likelihood of structural collapse within a specific horizon T:

$$P(\tau_{\text{ruin}} \leq T).$$

The core mathematical innovation in this layer is the definition of the *spectral fragility measure*. We establish a rigorous link between the spectral radius of the survival operator and the stability of the regime process:

$$\rho(N) = \frac{1}{\min|\lambda(K)|}.$$

Interpretation This operator result provides a precise mathematical definition of fragility:

- A *smaller spectral radius* ($\rho(N) \downarrow$) indicates a spectral contraction of the survival operator, signaling an accelerating path toward systemic collapse.
- A *larger spectral radius* ($\rho(N) \uparrow$) indicates a spectral expansion, corresponding to increased temporal persistence and antifragility.

Under the COR model's framework, *fragility* is no longer a qualitative description but is defined as the *spectral contraction of the survival operator*. This provides a rigorous mathematical grounding for Talebian uncertainty, where the survival of the system is governed by the spectral topology of its underlying regime generator. Table 5 presents a comparison of the classical HMM vs. the COR model.

Table 5 delineates the fundamental shifts in technical analysis when moving from Classical HMM Identification to the more advanced COR Model Identification. This transition redefines how system dynamics are measured and understood across four key metrics. In a classical HMM framework, states are viewed as isolated probability buckets, treating each regime as a discrete container. Conversely, the COR model treats states as nodes in a temporal geometry, implying a interconnected structure where the relationship and distance between states are governed by time.

Table 5: Comparison of Classical HMM and COR Model Identification.

Metric	Classical HMM Identification	COR Model Identification
State Nature	Isolated probability buckets	Nodes in a temporal geometry
Persistence	Probability of staying 1 step	Cumulative time spent (N_{ij})
Risk	Instantaneous Volatility (σ)	Spectral Fragility (ρ) & Ruin Bound
Tail Risk	Ignored or Gaussian-averaged	Endogenous Exit Hazard (ε)

The concept of persistence—how long a system remains in a given state—is simplified in HMMs as the probability of staying exactly one step further. The COR model provides a deeper longitudinal view by measuring the cumulative time spent, denoted by the matrix N_{ij}. This aligns with our definition of probability mass as the content of probability over a specific duration rather than a single discrete instance.

Risk assessment undergoes a significant transformation between these models. HMMs typically rely on instantaneous volatility (σ), capturing short-term fluctuations. The COR model, however, utilizes spectral fragility (ρ) and ruin bounds. By looking at spectral quantities (eigenvalues of the transition matrix), it identifies the structural fragility of the system and the mathematical limits of "ruin" or total system failure.

Finally, the treatment of extreme events, or tail risk, differs sharply. In classical HMMs, tail risk is often ignored or Gaussian-averaged, which can lead to underestimating the impact of outliers. The COR model explicitly accounts for these events through the endogenous exit hazard (ε), treating the potential for a sudden regime exit as an inherent, measurable risk factor of the system's internal dynamics.

Practical trading interpretation From a strategic and risk-control perspective, these findings necessitate a fundamental shift in how market exposure is managed. State 2, despite being the dominant regime, should never be treated as "harmless carry"; its Student-t parameter of $\nu = 3$ indicates a structurally fragile environment where tail events are a latent, persistent threat. In contrast, State 3 serves as the primary conduit for crash propagation, implying that once this regime is detected, market exposure must be sharply reduced or offset by convex hedging strategies already in place. While State 1 offers a genuinely quiet window for aggressive positioning, its lack of dominance within the system's residence-time architecture makes it an unreliable anchor for long-term risk-taking.

The broader implication is that any strategy ignoring the residence-time structure of the market is doomed to systematically underestimate risk. By focusing solely on instantaneous volatility rather than cumulative temporal exposure, a practitioner misses the "slow-burn" fragility inherent in the market's long-term survival mass. Consequently, the 5.6% ruin bound derived from the COR model suggests that the E-mini S&P 500 should be managed using fail-closed logic. Rather than relying on simple variance targeting—which assumes a stable Gaussian baseline—the trader must account for the reality that the system is structurally predisposed to leakage and ruin, requiring a defensive posture that prioritizes survival over the long-horizon.

Appendix A: R-Code

```
#!/usr/bin/env Rscript
# ================================================================
# Clock of Regimes (COR) analysis for IBKR_ES_2Y1h.csv
# ---------------------------------------------------------------
# PURPOSE
# -------
# This script implements a self-contained Clock of Regimes workflow:
#
# 1. Read the close column from IBKR_ES_2Y1h.csv.
# 2. Convert closes into log returns.
# 3. Fit a K-state Gaussian Hidden Markov Model (HMM).
# 4. Fit a K-state Student-t Hidden Markov Model (HMM).
# 5. Estimate transition matrices and implied spell lengths.
# 6. Build Talebian hazards:
# epsilon_i = epsilon_min + c / nu_i
# 7. Construct the hazard-adjusted operator:
# Q = diag(1 - epsilon_i) %*% A
# 8. Form the CoR generator:
# K = Q - I
# 9. Compute the fundamental matrix:
# N = -K^{-1}
# 10. Construct quasi-stationary and residence-weight vectors.
# 11. Compute residence-weighted tail risk and a ruin bound.
#
# DESIGN CHOICE
# -------------
# The script is deliberately written in base R so that every step of the
# HMM and CoR construction is transparent and auditable.
# ================================================================

# Prevent automatic conversion of strings to factors when reading data frames.
options(stringsAsFactors = FALSE)

# ----------------------------
```

```r
# 1. User parameters
# -----------------------------

# Name of the input CSV file expected in the current working directory.
input_file <- "IBKR_ES_2Y1h.csv"

# Name of the price column to use for return construction.
close_col <- "close"

# Number of hidden states/regimes in the HMM.
num_states <- 3L

# Number of random starts for the EM algorithm.
# More starts reduce the chance of ending at a poor local optimum.
n_starts <- 5L

# Maximum number of EM iterations per random start.
max_iter <- 200L

# Convergence tolerance for the log-likelihood improvement.
tol <- 1e-6

# Random seed for reproducibility.
seed <- 123

# Minimum baseline hazard level used in the Talebian hazard specification.
epsilon_min <- 0.01

# Sensitivity of hazard to tail thickness.
# Larger c_hazard makes hazard rise more strongly as nu decreases.
c_hazard <- 0.05

# Empirical left-tail probability cutoff used to define a loss event q_i.
tail_alpha <- 0.01

# Horizon over which the ruin bound is evaluated.
# For 1-hour data, 250 is interpretable as 250 hourly observations.
ruin_horizon <- 250

# Candidate degrees-of-freedom values for Student-t estimation.
# The script chooses the best value on this grid during the Student-t M-step.
nu_grid <- c(3:30, 40, 60, 100)
```

```r
# Fix the random number generator state.
set.seed(seed)

# ----------------------------
# 2. Basic helpers
# ----------------------------

# Numerically stable log(sum(exp(x))).
# Useful in probabilistic computations, though not explicitly used later
# in the current implementation.
log_sum_exp <- function(x) {
  m <- max(x)
  m + log(sum(exp(x - m)))
}

# Normalize a vector into probabilities while guarding against zeros.
# Any element smaller than eps is pushed up to eps before normalization.
safe_normalize <- function(x, eps = 1e-12) {
  x <- pmax(x, eps)
  x / sum(x)
}

# Normalize each row of a matrix so rows sum to one.
# Used for transition matrices to ensure valid probability rows.
row_normalize <- function(M, eps = 1e-12) {
  M <- pmax(M, eps)
  rs <- rowSums(M)
  M / rs
}

# Weighted mean with a safety fallback.
# If total weight is nonpositive, fall back to the ordinary mean.
weighted_mean_safe <- function(x, w) {
  sw <- sum(w)
  if (sw <= 0) return(mean(x))
  sum(w * x) / sw
}

# Weighted variance with safety checks.
# If weights collapse, fall back to ordinary variance.
# A small positive floor avoids degeneracy.
weighted_var_safe <- function(x, w, mu = NULL) {
  if (is.null(mu)) mu <- weighted_mean_safe(x, w)
```

```
  sw <- sum(w)
  if (sw <= 0) return(stats::var(x))
  max(sum(w * (x - mu)^2) / sw, 1e-10)
}

# ----------------------------
# 3. Data ingestion
# ----------------------------

# Read the close-price series from a CSV file.
# Steps:
# - verify the file exists,
# - read the CSV,
# - convert column names to lowercase for robustness,
# - extract the requested close column,
# - keep only finite numeric values,
# - require at least 10 observations.
read_close_series <- function(file, close_name = "close") {
  if (!file.exists(file)) {
    stop(sprintf("Input␣file␣'%s'␣not␣found␣in␣the␣current␣working␣directory.", file))
  }
  dat <- read.csv(file, check.names = FALSE)
  names(dat) <- tolower(names(dat))
  if (!(tolower(close_name) %in% names(dat))) {
    stop(sprintf("Column␣'%s'␣not␣found.␣Available␣columns:␣%s",
                 close_name, paste(names(dat), collapse = ",␣")))
  }
  px <- as.numeric(dat[[tolower(close_name)]])
  px <- px[is.finite(px)]
  if (length(px) < 10L) stop("Not␣enough␣finite␣close␣observations.")
  px
}

# Convert close prices into log returns:
# r_t = log(P_t) - log(P_{t-1})
# Keeps only finite values and requires a minimal sample size.
compute_log_returns <- function(close) {
  ret <- diff(log(close))
  ret <- ret[is.finite(ret)]
  if (length(ret) < 10L) stop("Not␣enough␣valid␣returns␣after␣differencing.")
  ret
}
```

```r
# ----------------------------
# 4. Emission log densities
# ----------------------------

# Gaussian log-density for a vector x under N(mu, sigma^2).
log_dnorm_vec <- function(x, mu, sigma) {
  stats::dnorm(x, mean = mu, sd = sigma, log = TRUE)
}

# Student-t log-density for a location-scale t distribution.
# If Z ~ t_nu, then X = mu + sigma * Z.
log_dt_locscale_vec <- function(x, mu, sigma, nu) {
  stats::dt((x - mu) / sigma, df = nu, log = TRUE) - log(sigma)
}

# Build the T x K matrix of emission log-densities.
# Each column corresponds to one hidden state.
# Each row corresponds to one observation.
emission_logdens_matrix <- function(x, params, model = c("gaussian", "student")) {
  model <- match.arg(model)
  Tn <- length(x)
  K <- length(params$mu)
  B <- matrix(NA_real_, nrow = Tn, ncol = K)
  for (k in seq_len(K)) {
    if (model == "gaussian") {
      B[, k] <- log_dnorm_vec(x, params$mu[k], params$sigma[k])
    } else {
      B[, k] <- log_dt_locscale_vec(x, params$mu[k], params$sigma[k], params$nu[k])
    }
  }
  B
}

# ----------------------------
# 5. Forward-backward (scaled)
# ----------------------------

# Run the scaled forward-backward algorithm for an HMM.
#
# INPUTS
# logB : T x K matrix of emission log-densities
# A : K x K transition matrix
# delta : initial state probability vector
```

```r
#
# OUTPUTS
# alpha : filtered forward probabilities
# beta : backward probabilities
# gamma : posterior state probabilities P(S_t = k | data)
# xi : posterior transition probabilities P(S_t=i,S_{t+1}=j | data)
# logLik: scaled log-likelihood
#
# The scaling prevents underflow in long time series.
forward_backward <- function(logB, A, delta) {
  Tn <- nrow(logB)
  K <- ncol(logB)

  B <- exp(logB)
  alpha <- matrix(0, Tn, K)
  beta <- matrix(0, Tn, K)
  scale <- numeric(Tn)

  # Initialize forward recursion.
  alpha[1, ] <- delta * B[1, ]
  scale[1] <- sum(alpha[1, ])
  alpha[1, ] <- alpha[1, ] / scale[1]

  # Forward recursion:
  # alpha_t(j) proportional to emission at t in state j times the
  # total predicted probability of arriving in j from t-1.
  for (t in 2:Tn) {
    alpha[t, ] <- (alpha[t - 1, ] %*% A) * B[t, ]
    scale[t] <- sum(alpha[t, ])
    alpha[t, ] <- alpha[t, ] / scale[t]
  }

  # Backward initialization.
  beta[Tn, ] <- rep(1, K)

  # Backward recursion.
  if (Tn >= 2L) {
    for (t in (Tn - 1L):1L) {
      beta[t, ] <- A %*% (B[t + 1L, ] * beta[t + 1L, ])
      beta[t, ] <- beta[t, ] / scale[t + 1L]
    }
  }
```

```
  # Posterior state probabilities.
  gamma <- alpha * beta
  gamma <- gamma / rowSums(gamma)

  # Posterior transition probabilities.
  xi <- array(0, dim = c(Tn - 1L, K, K))
  if (Tn >= 2L) {
    for (t in 1:(Tn - 1L)) {
      M <- (alpha[t, ] * A) * rep(B[t + 1L, ] * beta[t + 1L, ], each = K)
      denom <- sum(M)
      xi[t, , ] <- M / denom
    }
  }

  list(
    alpha = alpha,
    beta = beta,
    gamma = gamma,
    xi = xi,
    logLik = sum(log(scale))
  )
}

# ----------------------------
# 6. Initialization
# ----------------------------

# Create initial parameter values for the HMM.
#
# Strategy:
# - initialize means near sample quantiles,
# - initialize sigmas near the overall sample standard deviation,
# - initialize a persistent transition matrix with large diagonal entries,
# - initialize the initial state probabilities uniformly,
# - for Student-t, initialize nu from a reasonable finite grid.
init_hmm_params <- function(x, K, model = c("gaussian", "student")) {
  model <- match.arg(model)

  qs <- stats::quantile(x, probs = seq(0, 1, length.out = K + 2L))
  mu_init <- as.numeric(qs[2:(K + 1L)])
  mu_init <- mu_init + rnorm(K, sd = stats::sd(x) / 10)
  sigma_init <- rep(max(stats::sd(x), 1e-4), K) * runif(K, 0.7, 1.3)
```

```r
  # Persistent initial transition matrix:
  # large diagonal, small off-diagonal probabilities.
  A <- matrix(0.05 / (K - 1), nrow = K, ncol = K)
  diag(A) <- 0.95
  A <- row_normalize(A)

  # Uniform initial state distribution.
  delta <- safe_normalize(rep(1 / K, K))

  params <- list(mu = mu_init, sigma = pmax(sigma_init, 1e-4))
  if (model == "student") {
    params$nu <- sample(c(4, 5, 6, 8, 10, 12, 15), K, replace = TRUE)
  }

  list(A = A, delta = delta, params = params)
}

# ----------------------------
# 7. M-step updates
# ----------------------------

# Update the transition matrix A and initial state vector delta
# using the posterior probabilities from the E-step.
update_A_delta <- function(fb) {
  gamma <- fb$gamma
  xi <- fb$xi
  K <- ncol(gamma)

  # Updated initial probabilities come from the posterior at time 1.
  delta <- safe_normalize(gamma[1, ])

  # Numerator: expected transitions i -> j summed across time.
  A_num <- apply(xi, c(2, 3), sum)

  # Denominator: expected time spent in state i at times 1,...,T-1.
  A_den <- colSums(gamma[-nrow(gamma), , drop = FALSE])

  A <- matrix(0, K, K)
  for (i in seq_len(K)) {
    if (A_den[i] <= 0) {
      A[i, ] <- rep(1 / K, K)
    } else {
      A[i, ] <- A_num[i, ] / A_den[i]
```

```r
    }
  }
  A <- row_normalize(A)
  list(A = A, delta = delta)
}

# Update Gaussian emission parameters.
# For each state k:
# mu_k = weighted mean of returns
# sigma_k^2 = weighted variance of returns
# using posterior state probabilities gamma[, k] as weights.
update_gaussian_params <- function(x, gamma) {
  K <- ncol(gamma)
  mu <- sigma <- numeric(K)
  for (k in seq_len(K)) {
    w <- gamma[, k]
    mu[k] <- weighted_mean_safe(x, w)
    sigma[k] <- sqrt(weighted_var_safe(x, w, mu[k]))
  }
  list(mu = mu, sigma = pmax(sigma, 1e-6))
}

# Weighted log-likelihood contribution for one Student-t state.
# Used when selecting the best nu from the candidate grid.
student_state_objective <- function(x, gamma_k, mu, sigma, nu) {
  sum(gamma_k * log_dt_locscale_vec(x, mu, sigma, nu))
}

# Update Student-t emission parameters.
#
# For each state:
# - use a latent-weight style update for mu and sigma,
# - choose nu by grid search over nu_grid,
# - impose nu > 2 to ensure finite conditional variance.
update_student_params <- function(x, gamma, old_params, nu_grid) {
  K <- ncol(gamma)
  mu <- sigma <- nu <- numeric(K)

  for (k in seq_len(K)) {
    gk <- gamma[, k]
    old_mu <- old_params$mu[k]
    old_sigma <- max(old_params$sigma[k], 1e-6)
    old_nu <- old_params$nu[k]
```

```r
    # Robustification weights from the Student-t representation.
    # Large squared residuals get smaller weights.
    z2 <- ((x - old_mu) / old_sigma)^2
    w <- (old_nu + 1) / (old_nu + z2)

    # Update location using gamma_k * w weights.
    mu[k] <- weighted_mean_safe(x, gk * w)

    # Update scale using the weighted residual sum of squares.
    sigma2_num <- sum(gk * w * (x - mu[k])^2)
    sigma2_den <- sum(gk)
    sigma[k] <- sqrt(max(sigma2_num / max(sigma2_den, 1e-12), 1e-10))

    # Choose nu by maximizing the weighted state objective over the grid.
    obj_vals <- sapply(nu_grid, function(nu_cand) {
      student_state_objective(x, gk, mu[k], sigma[k], nu_cand)
    })
    nu[k] <- nu_grid[which.max(obj_vals)]
  }

  list(mu = mu, sigma = pmax(sigma, 1e-6), nu = pmax(nu, 2.1))
}

# ----------------------------
# 8. Model fitting
# ----------------------------

# Fit a Gaussian or Student-t HMM using EM and multiple random starts.
#
# Workflow:
# - initialize parameters,
# - alternate E-step and M-step,
# - stop when the log-likelihood stabilizes,
# - keep the best solution across starts.
fit_hmm_em <- function(x, K = 3L, model = c("gaussian", "student"),
                       n_starts = 5L, max_iter = 200L, tol = 1e-6,
                       nu_grid = c(3:30, 40, 60, 100), verbose = TRUE) {
  model <- match.arg(model)
  best <- NULL

  for (s in seq_len(n_starts)) {
    init <- init_hmm_params(x, K, model)
```

```r
A <- init$A
delta <- init$delta
params <- init$params
prev_ll <- -Inf

for (iter in seq_len(max_iter)) {
  # E-step: compute posteriors.
  logB <- emission_logdens_matrix(x, params, model)
  fb <- forward_backward(logB, A, delta)
  ll <- fb$logLik

  # M-step: update transition structure.
  upd <- update_A_delta(fb)
  A <- upd$A
  delta <- upd$delta

  # M-step: update emission parameters.
  if (model == "gaussian") {
    params <- update_gaussian_params(x, fb$gamma)
  } else {
    params <- update_student_params(x, fb$gamma, params, nu_grid)
  }

  if (verbose) {
    cat(sprintf("model=%s␣start=%d␣iter=%d␣logLik=%.10f\n", model, s, iter, ll))
  }

  # Convergence check.
  if (abs(ll - prev_ll) < tol) break
  prev_ll <- ll
}

# Final E-step under the converged parameters.
logB <- emission_logdens_matrix(x, params, model)
fb <- forward_backward(logB, A, delta)
ll <- fb$logLik

fit <- list(
  model = model,
  A = A,
  delta = delta,
  params = params,
  gamma = fb$gamma,
```

```r
      xi = fb$xi,
      logLik = ll,
      n = length(x),
      K = K
    )

    # Keep the highest-likelihood fit.
    if (is.null(best) || fit$logLik > best$logLik) {
      best <- fit
    }
  }

  best
}

# ----------------------------
# 9. State ordering
# ----------------------------

# Reorder states from low-volatility to high-volatility.
#
# For Gaussian:
# order by sigma.
#
# For Student-t:
# order by conditional variance:
# Var(X|state=k) = [nu_k / (nu_k - 2)] * sigma_k^2
#
# This improves interpretability by making state labels comparable.
reorder_fit_by_scale <- function(fit) {
  if (fit$model == "gaussian") {
    ord <- order(fit$params$sigma)
  } else {
    cond_var <- (fit$params$nu / (fit$params$nu - 2)) * fit$params$sigma^2
    ord <- order(cond_var)
  }

  fit$A <- fit$A[ord, ord, drop = FALSE]
  fit$delta <- fit$delta[ord]
  fit$gamma <- fit$gamma[, ord, drop = FALSE]
  fit$params$mu <- fit$params$mu[ord]
  fit$params$sigma <- fit$params$sigma[ord]
  if (!is.null(fit$params$nu)) fit$params$nu <- fit$params$nu[ord]
```

```r
  fit
}

# ----------------------------
# 10. Viterbi decoding
# ----------------------------

# Compute the single most likely hidden-state path using the
# Viterbi dynamic programming algorithm.
#
# This is distinct from gamma:
# - gamma gives state probabilities at each time,
# - Viterbi gives one best full sequence of states.
viterbi_decode <- function(x, fit) {
  logB <- emission_logdens_matrix(x, fit$params, fit$model)
  A <- fit$A
  delta <- fit$delta
  Tn <- nrow(logB)
  K <- ncol(logB)

  # delta_log stores the best log-probability ending in state j at time t.
  delta_log <- matrix(-Inf, Tn, K)

  # psi stores the best predecessor state for backtracking.
  psi <- matrix(0L, Tn, K)

  # Initialization.
  delta_log[1, ] <- log(delta) + logB[1, ]

  # Recursion.
  for (t in 2:Tn) {
    for (j in seq_len(K)) {
      vals <- delta_log[t - 1, ] + log(A[, j])
      psi[t, j] <- which.max(vals)
      delta_log[t, j] <- max(vals) + logB[t, j]
    }
  }

  # Backtrack to recover the optimal path.
  path <- integer(Tn)
  path[Tn] <- which.max(delta_log[Tn, ])
  if (Tn >= 2L) {
    for (t in (Tn - 1L):1L) {
```

```r
      path[t] <- psi[t + 1L, path[t + 1L]]
    }
  }
  path
}

# ----------------------------
# 11. COR construction
# ----------------------------

# Convert discrete HMM persistence into expected spell lengths:
# E[tau_i] = 1 / (1 - a_ii)
expected_spell_lengths <- function(A) {
  1 / pmax(1 - diag(A), 1e-12)
}

# Talebian hazard specification:
# epsilon_i = epsilon_min + c_hazard / nu_i
# So fatter tails (smaller nu_i) imply larger hazard.
hazard_from_nu <- function(nu, epsilon_min = 0.01, c_hazard = 0.05) {
  epsilon_min + c_hazard / nu
}

# Build the hazard-adjusted transition operator:
# Q = diag(1 - epsilon_i) %*% A
# Each row is discounted by the survival factor (1 - epsilon_i).
build_Q <- function(A, epsilon) {
  diag(1 - epsilon, nrow = length(epsilon)) %*% A
}

# Build the CoR generator:
# K = Q - I
build_K <- function(Q) {
  Q - diag(nrow(Q))
}

# Compute the fundamental matrix:
# N = -K^{-1}
# This gives the residence-time geometry of the transient system.
fundamental_matrix <- function(K) {
  solve(-K)
}
```

```r
# Approximate a quasi-stationary distribution from Q.
# Since the system is open/leaky, classical stationarity does not strictly hold.
# This uses the dominant left eigenvector of Q and normalizes it.
quasi_stationary_distribution <- function(Q) {
  ev <- eigen(t(Q))
  idx <- which.max(Re(ev$values))
  vec <- Re(ev$vectors[, idx])
  vec <- abs(vec)
  safe_normalize(vec)
}

# Convert N into a residence-weight vector.
# Starting from an initial distribution, multiply by N to get the expected
# total time-content across states, then normalize.
residence_weight_vector <- function(N, init = NULL) {
  K <- nrow(N)
  if (is.null(init)) init <- rep(1 / K, K)
  w <- as.numeric(init %*% N)
  safe_normalize(w)
}

# Compute the statewise left-tail probability q_i for a given threshold.
# For Gaussian states use pnorm; for Student-t states use pt.
left_tail_state_probs <- function(fit, threshold) {
  if (fit$model == "gaussian") {
    stats::pnorm(threshold, mean = fit$params$mu, sd = fit$params$sigma)
  } else {
    sapply(seq_along(fit$params$mu), function(k) {
      stats::pt((threshold - fit$params$mu[k]) / fit$params$sigma[k],
                df = fit$params$nu[k])
    })
  }
}

# ----------------------------
# 12. Reporting helpers
# ----------------------------

# Format numeric values with a fixed number of decimal places.
fmt_num <- function(x, digits = 6) formatC(x, digits = digits, format = "f")

# Write a labeled matrix to a text connection in tabular form.
# This is used for the plain-text report file.
```

```
write_matrix_block <- function(con, title, M, digits = 6) {
  writeLines(title, con)
  rn <- rownames(M); cn <- colnames(M)
  if (is.null(rn)) rn <- paste0("State", seq_len(nrow(M)))
  if (is.null(cn)) cn <- paste0("State", seq_len(ncol(M)))
  header <- paste(c("", cn), collapse = "\t")
  writeLines(header, con)
  for (i in seq_len(nrow(M))) {
    line <- paste(c(rn[i], fmt_num(M[i, ], digits)), collapse = "\t")
    writeLines(line, con)
  }
  writeLines("", con)
}

# ----------------------------
# 13. Main workflow
# ----------------------------

# Read closes and compute returns.
close_px <- read_close_series(input_file, close_col)
ret <- compute_log_returns(close_px)

cat(sprintf("Loaded %d closes and %d log returns from %s\n",
          length(close_px), length(ret), input_file))

# Fit the Gaussian HMM and reorder states by volatility.
cat("\nFitting Gaussian HMM...\n")
gauss_fit <- fit_hmm_em(
  x = ret, K = num_states, model = "gaussian",
  n_starts = n_starts, max_iter = max_iter, tol = tol,
  nu_grid = nu_grid, verbose = TRUE
)
gauss_fit <- reorder_fit_by_scale(gauss_fit)

# Fit the Student-t HMM and reorder states by conditional variance.
cat("\nFitting Student-t HMM...\n")
t_fit <- fit_hmm_em(
  x = ret, K = num_states, model = "student",
  n_starts = n_starts, max_iter = max_iter, tol = tol,
  nu_grid = nu_grid, verbose = TRUE
)
t_fit <- reorder_fit_by_scale(t_fit)
```

```r
# Extract baseline transition matrices.
A_gauss <- gauss_fit$A
A_t <- t_fit$A

# Compute discrete-time expected spell lengths for each model.
spell_gauss <- expected_spell_lengths(A_gauss)
spell_t <- expected_spell_lengths(A_t)

# Compute conditional variance for the Student-t states.
cond_var_t <- (t_fit$params$nu / (t_fit$params$nu - 2)) * t_fit$params$sigma^2

# Decode the most likely Student-t state sequence.
# This is useful for state interpretation and event labeling.
vpath_t <- viterbi_decode(ret, t_fit)

# Build the CoR layer from the Student-t fit.
# Student-t is used because hazard is explicitly linked to nu_i.
epsilon <- hazard_from_nu(t_fit$params$nu, epsilon_min, c_hazard)
Q <- build_Q(A_t, epsilon)
Kmat <- build_K(Q)
Nmat <- fundamental_matrix(Kmat)

# Construct quasi-stationary and residence-weight vectors.
pi_qs <- quasi_stationary_distribution(Q)
pi_res <- residence_weight_vector(Nmat, init = pi_qs)

# Define the empirical loss threshold using the left tail of the return sample.
loss_threshold <- as.numeric(stats::quantile(ret, probs = tail_alpha, na.rm = TRUE))

# Compute statewise tail probabilities under the Student-t fit.
q_state <- left_tail_state_probs(t_fit, loss_threshold)

# Residence-weighted tail risk.
bar_q <- sum(pi_res * q_state)

# Residence-weighted hazard.
bar_lambda <- sum(pi_res * epsilon)

# Exponential ruin upper bound over the chosen horizon.
ruin_bound <- 1 - exp(-bar_lambda * bar_q * ruin_horizon)

# Assemble a state summary table for export and inspection.
state_summary <- data.frame(
```

```r
  state = seq_len(num_states),
  mu_gaussian = gauss_fit$params$mu,
  sigma_gaussian = gauss_fit$params$sigma,
  spell_gaussian = spell_gauss,
  mu_student = t_fit$params$mu,
  sigma_student = t_fit$params$sigma,
  nu_student = t_fit$params$nu,
  cond_var_student = cond_var_t,
  spell_student = spell_t,
  epsilon = epsilon,
  q_tail = q_state,
  pi_quasi_stationary = pi_qs,
  pi_residence = pi_res
)

# Save structured outputs to CSV files.
write.csv(state_summary, "cor_state_summary.csv", row.names = FALSE)
write.csv(A_gauss, "gaussian_transition_matrix.csv", row.names = FALSE)
write.csv(A_t, "student_t_transition_matrix.csv", row.names = FALSE)
write.csv(Q, "cor_Q_operator.csv", row.names = FALSE)
write.csv(Kmat, "cor_K_generator.csv", row.names = FALSE)
write.csv(Nmat, "cor_N_fundamental_matrix.csv", row.names = FALSE)
write.csv(data.frame(return = ret, decoded_state_student = vpath_t),
        "student_t_decoded_states.csv", row.names = FALSE)

# Redirect console output to a text report file.
sink("cor_analysis_report.txt")
cat("============================================================\n")
cat("CLOCK␣OF␣REGIMES␣ANALYSIS␣REPORT\n")
cat("============================================================\n\n")
cat(sprintf("Input␣file␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣:␣%s\n", input_file))
cat(sprintf("Close␣observations␣␣␣␣␣␣␣␣:␣%d\n", length(close_px)))
cat(sprintf("Return␣observations␣␣␣␣␣␣␣:␣%d\n", length(ret)))
cat(sprintf("Number␣of␣states␣␣␣␣␣␣␣␣␣␣:␣%d\n", num_states))
cat(sprintf("Tail␣probability␣alpha␣␣␣␣:␣%.4f\n", tail_alpha))
cat(sprintf("Ruin␣horizon␣T␣␣␣␣␣␣␣␣␣␣␣␣:␣%d\n\n", ruin_horizon))

# Report model fit quality.
cat("Gaussian␣HMM␣log-likelihood\n")
cat(sprintf("%.10f\n\n", gauss_fit$logLik))
cat("Student-t␣HMM␣log-likelihood\n")
cat(sprintf("%.10f\n\n", t_fit$logLik))
```

```
# Report matrices in plain text.
write_matrix_block(stdout(), "Gaussian␣transition␣matrix␣A:", A_gauss)
write_matrix_block(stdout(), "Student-t␣transition␣matrix␣A:", A_t)
write_matrix_block(stdout(), "Hazard-adjusted␣transition␣operator␣Q:", Q)
write_matrix_block(stdout(), "COR␣generator␣K␣=␣Q␣-␣I:", Kmat)
write_matrix_block(stdout(), "Fundamental␣matrix␣N␣=␣-K^{-1}:", Nmat)

# Report the state summary and ruin quantities.
cat("State␣summary\n")
print(state_summary, row.names = FALSE)
cat("\n")
cat(sprintf("Empirical␣left-tail␣threshold␣(alpha␣=␣%.4f):␣%.8f\n", tail_alpha, loss_
    threshold))
cat(sprintf("Residence-weighted␣tail␣probability␣qbar␣␣␣␣:␣%.8f\n", bar_q))
cat(sprintf("Residence-weighted␣hazard␣lambdabar␣␣␣␣␣␣␣␣␣:␣%.8f\n", bar_lambda))
cat(sprintf("Ruin␣bound␣over␣horizon␣T␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣:␣%.8f\n", ruin_bound))
cat("\n")

# Explain why quasi-stationary and residence-weight vectors are used.
cat("Interpretive␣note:\n")
cat(paste(
  "Because␣K␣=␣Q␣-␣I␣is␣an␣open␣/␣leaky␣operator,␣a␣classical␣stationary",
  "distribution␣does␣not␣strictly␣exist.␣The␣script␣therefore␣reports␣a",
  "quasi-stationary␣vector␣from␣Q␣and␣a␣residence-weight␣vector␣implied␣by␣N.",
  "The␣latter␣is␣used␣for␣the␣residence-weighted␣tail␣and␣ruin␣summaries.",
  sep = "\n"
))
cat("\n")
sink()

# Print completion message and list output files.
cat("\nFinished.␣Output␣files␣written:\n")
cat("␣␣-␣cor_analysis_report.txt\n")
cat("␣␣-␣cor_state_summary.csv\n")
cat("␣␣-␣gaussian_transition_matrix.csv\n")
cat("␣␣-␣student_t_transition_matrix.csv\n")
cat("␣␣-␣cor_Q_operator.csv\n")
cat("␣␣-␣cor_K_generator.csv\n")
cat("␣␣-␣cor_N_fundamental_matrix.csv\n")
cat("␣␣-␣student_t_decoded_states.csv\n")
```

Appendix B: R-Code Output

```
#============================================================
# CLOCK OF REGIMES ANALYSIS REPORT
#============================================================
Input file : IBKR_ES_2Y1h.csv
Output file : cor_analysis_report.txt
Close observations : 3670
Return observations : 3669
Number of states : 3
Tail probability alpha : 0.0100
Ruin horizon T : 250

Gaussian HMM log-likelihood
21173.8299404050

Student-t HMM log-likelihood
21661.8321503555

Gaussian transition matrix A:
  State1 State2 State3
State1 0.680745 0.238491 0.080764
State2 0.069493 0.910750 0.019756
State3 0.319388 0.405942 0.274670

Student-t transition matrix A:
  State1 State2 State3
State1 0.709825 0.000000 0.290175
State2 0.000837 0.971886 0.027278
State3 0.206698 0.054608 0.738694

Hazard-adjusted transition operator Q:
  State1 State2 State3
State1 0.702372 0.000000 0.287128
State2 0.000814 0.945969 0.026550
State3 0.201186 0.053152 0.718995

COR generator K = Q - I:
  State1 State2 State3
State1 -0.297628 0.000000 0.287128
State2 0.000814 -0.054031 0.026550
State3 0.201186 0.053152 -0.281005
```

```
Fundamental matrix N = -K^{-1}:
   State1 State2 State3
State1 14.267170 15.810407 16.071895
State2 5.770728 26.799214 8.428562
State3 11.306141 16.388579 16.659628

State summary
 state mu_gaussian sigma_gaussian spell_gaussian mu_student sigma_student
     1 -3.264717e-08 0.000010000 3.132292 -2.168526e-08 0.0000100000
     2 6.662832e-05 0.001375609 11.204502 9.696612e-05 0.0006929728
     3 -5.897585e-04 0.017745344 1.378683 3.524294e-05 0.0019427734
 nu_student cond_var_student spell_student epsilon q_tail
       100 1.020408e-10 3.446197 0.01050000 5.003544e-177
         3 1.440634e-06 35.568996 0.02666667 1.874378e-03
         3 1.132311e-05 3.826929 0.02666667 3.110910e-02
 pi_quasi_stationary pi_residence
          0.2131674 0.2131674
          0.4980987 0.4980987
          0.2887339 0.2887339

Empirical left-tail threshold (alpha = 0.0100): -0.00560969
Residence-weighted tail probability qbar : 0.00991588
Residence-weighted hazard lambdabar : 0.02322046
Ruin bound over horizon T : 0.05593741

Interpretive note:
Because K = Q - I is an open / leaky operator, a classical stationary
distribution does not strictly exist. The script therefore reports a
quasi-stationary vector from Q and a residence-weight vector implied by N.
The latter is used for the residence-weighted tail and ruin summaries.

AIC Improvement: 970
```

Appendix C: Conceptualizing the Spectral Radius: The Market's "Battery Life"

To ensure the report remains accessible to stakeholders who may not have a background in linear algebra, it is helpful to provide a conceptual *layman's* translation of the spectral radius. In the context of this study, the Spectral Radius (ρ) can be thought of as the retention rate of the market's health. If a standard market system were a closed circuit, its spectral radius would be exactly 1.0, meaning 100% of its *survival mass* is preserved from one day to the next. However, because the

COR model accounts for the possibility of ruin (hazard leakage), the spectral radius drops below 1.0 (e.g., to 0.999).

This tiny deficit represents the *leak* in the system. The spectral radius identifies the single, most dominant rate at which the market *drains* its safety over time. While the market may switch between different states—some calm, some volatile—the spectral radius captures the average persistence of the entire system. It tells us how much of our initial *survival probability* remains after the states have finished interacting with one another.

C.1 The Math of Compounding Fragility

The reason ρ is so powerful is that it compounds. If ρ is even slightly below 1.0, the *leak* accumulates over time like interest in reverse. By the time we reach a 250-day horizon, these tiny daily leaks aggregate into the 5.6% ruin bound identified in the results.

In short, ρ is the fundamental frequency of market survival. It proves that the danger isn't just a one-time bad day in State 3; it is the mathematical reality that the longer you stay in a system with a ρ less than 1.0, the more certain it is that the *leak* (ruin) will eventually find you. It turns the abstract concept of *fragility* into a single, trackable number.

Appendix D: Open Systems and Survival Dynamics

To quantify the long-horizon risk of the E-mini S&P 500, we move beyond the standard closed-system Hidden Markov Model (HMM) and adopt an Open System framework. While a traditional HMM assumes that the probability mass is conserved within the defined states, our COR model analysis introduces a *leak* mechanism to account for terminal ruin events. This bridges the gap between standard HMM transition matrices and the Open System physics analogy. This clarifies that *ruin* is an absorbing boundary outside the three-state system.

D.1 The Survival Operator and Hazard Leakage

We define the system's evolution through a Survival Operator, $\mathscr{S}(t)$. Unlike a standard transition matrix A where rows sum to 1, we allow for *hazard leakage* by defining a state-dependent ruin probability q_i. The probability of remaining *alive* (within the three-state market architecture) after one step is governed by,

$$S = A \circ (1 - \mathbf{q}).$$

In this context, the symbol $\circ$ represents the Hadamard product, which refers to the element-wise multiplication of two matrices. This means that the entries are multiplied position by position:

- A is the 3×3 transition probability matrix.

- $\mathbf{q} = [q_1, q_2, q_3]^\intercal$ is the vector of *Ruin Intensities*, representing the probability of a 1% tail event in each state.

Hazard *leakage* is the mechanism by which probability mass "escapes" the system. In the COR model, State 3 (Stress) acts as a high-leakage node, while State 2 (Steady) represents a slow-leakage node. The *leak* is not a transition to another market state, but a transition to a terminal state of system failure, insolvency, or ruin.

D.2 The Aggregate Decay Constant ($\bar{\lambda}$)

To determine the long-term stability of the market, we analyze the decay of the survival mass over time. The Aggregate Decay Constant, $\bar{\lambda}$, is derived from the spectral properties of the survival operator. Specifically, if $\rho(S)$ is the spectral radius (the largest eigenvalue) of the operator S, then:

$$\bar{\lambda} = -\ln(\rho(S))$$

This constant represents the exponential rate at which the system loses its *survival mass*. A higher $\bar{\lambda}$ indicates a more fragile system where the *leakage* to ruin is more aggressive.

D.3 Cumulative Ruin Bound

The probability of the system surviving up to time T is given by $P(\text{Survival} > T) \approx e^{-\bar{\lambda}T}$. Consequently, the Cumulative Ruin Bound over a horizon T is defined as:

$$R(T) = 1 - e^{-\bar{\lambda}T}$$

This metric transforms instantaneous state-specific risks into a coherent long-term safety forecast. By setting $T = 250$, we calculate the probability that the market's inherent residence architecture will lead to at least one 1% tail event over a standard trading year.

The methodology achieves a high degree of precision by explicitly mapping the transition from state-specific tail probabilities, q_i, to the aggregate 5.6% ruin bound. Rather than relying on heuristic approximations, the model utilizes the Survival Operator as a formal linear algebra tool. By encoding the *leakage* of probability mass directly into the spectral properties of the transition matrix, we provide mathematical rigor which is significantly more robust and reproducible than a standard Monte Carlo simulation. This approach treats the market return process as an open system where risk is not an isolated shock, but a structural decay of the survival mass over time.

This formalism establishes a seamless logical flow that leads directly to the Talebian conclusion of the study. By defining the aggregate decay constant, $\bar{\lambda}$, we prove that ruin is not merely a matter of bad luck, but a deterministic function of two variables: temporal exposure (T) and internal regime architecture ($\bar{\lambda}$). This proof shifts the focus from event-magnitude to structural duration, demonstrating that the longer a system resides in a fragile regime, the more certain its eventual encounter with a tail event becomes. Consequently, the 5.6% ruin bound is not an arbitrary figure, but a direct reflection of the market's inherent survival-weighted geometry.

Acknowledgments

The R-based estimation engines were developed through an iterative human-in-the-loop process by the authors, utilizing Claude 4.6 Sonnet (Anthropic) for code debugging and documentation. All AI-generated outputs, including the implementation of the Student-t Maximization step and recursive scaling logic, were manually audited, verified for mathematical accuracy using Wolfram Mathematica 14.3 coupled with the Wolfram LLM Kit, and refined by the authors to ensure alignment with the theoretical model.

References

M. Berman and R. Shoenfeld. Invariants in experimental data on linear kinetics and the formulation of models. *Journal of Applied Physics*, 27:1361–1370, 1956.

D. G. Covell, M. Berman, and C. DeLisi. Mean residence time—theoretical development, experimental determination, and practical use in tracer analysis. *Mathematical Biosciences*, 72:213–244, 1984.

J. Eisenfeld. On mean residence times in compartments. *Mathematical Biosciences*, 57:265–278, 1981.

F. Gao and X. He. *Survival Analysis: Theory and Application in Finance*, chapter 120, pages 3903–3934. World Scientific, Singapore, 2025.

J. D. Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, 57:357–384, 1989.

J. D. Hamilton. Analysis of time series subject to changes in regime. *Journal of Econometrics*, 45 (1–2):39–70, 1990.

J. D. Hamilton. Macroeconomic regimes and regime shifts. In J. B. Taylor and H. Uhlig, editors, *Handbook of Macroeconomics*, volume 2, pages 163–201. Elsevier, 2016.

J. A. Jacquez. Density functions of residence times for deterministic and stochastic compartmental systems. *Mathematical Biosciences*, 180:127–139, 2002.

J. A. Jacquez and C. P. Simon. Qualitative theory of compartmental systems. *SIAM Review*, 35: 43–79, 1993.

D. G. Kleinbaum and M. Klein. *Survival Analysis*. Springer, New York, 3rd edition, 2012.

O. A. Linares, J. A. Jacquez, L. A. Zech, M. J. Smith, J. A. Sanfield, L. A. Morrow, S. G. Rosen, and J. B. Halter. Norepinephrine metabolism in humans: Kinetic analysis and model. *Journal of Clinical Investigation*, 80(5):1332–1341, 1987.

N. N. Taleb. *Antifragile: Things That Gain from Disorder*. Random House, New York, 2012.

N. N. Taleb. *Incerto*. Random House, New York, 2016.

N. N. Taleb. *Statistical Consequences of Fat Tails: Real World Preasymptotics, Epistemology, and Applications*. STEM Academic Press, New York, 2020.