NAVIGATION ARCHITECTURE

NashMark AI Nav

Ambiguity-Sensitive Continuity and Drift Recovery in Navigation

Equilibrium-Governed Model | Label-Light Traversal | Degraded Signal Recovery

EXECUTIVE SUMMARY

Core Proposition

Fundamental Shift

NashMark AI begins from the premise that **navigation is a continuity problem under uncertainty**, not merely a point-estimation challenge. The system must preserve coherent traversal when observations are incomplete, distorted, delayed, or structurally ambiguous.

Benchmark Position

Conventional Kalman filtering remains strongest in clean labelled-route tracking. HMM-style models retain value in discrete branch-state interpretation. **NashMark AI shows its clearest advantages in ambiguity-sensitive continuity, drift recovery, and degraded traversal conditions**, where equilibrium preservation matters more than immediate local correction.

Narrow but Defensible Claim: NashMark AI is a viable navigation and recovery architecture with strongest current evidence in ambiguous, degraded, and continuity-sensitive environments.

Key Capabilities

1. **Preserve State Coherence**
   Maintain lawful traversal under uncertainty
2. **Resist False Collapse**
   Avoid premature commitment to wrong states
3. **Restore After Drift**
   Recover from degraded observations
4. **Defer Unsafe Commitment**
   Wait for structural confidence

Target Environments

- ✓ Deep-sea traversal
- ✓ Urban canyons
- ✓ GNSS dropouts
- ✓ Orbital blackout

PART I

Mathematical Foundation

The theoretical framework governing equilibrium-based navigation

CHAPTER 1 — 2

Navigation as Continuity, Not Mere Fix Estimation

Conventional Approach

Most navigation systems are constructed as **recursive estimators**. They attempt to infer latent state from noisy observations through prediction and correction. This is effective when:

- Observation stream remains sufficiently stable
- Environment is well-behaved estimation problem

NashMark Position

Treats navigation as **preservation of lawful traversal under uncertainty**. The system should:

Preserve Coherence
Maintain state consistency

Resist Collapse
Avoid false state transitions

Restore After Drift
Recover equilibrium

Defer Commitment
Wait for confidence

Latent Path Inference

Let the latent traversal state at step t be defined over a candidate set of route states. A probabilistic inference layer assigns likelihood over those states rather than forcing immediate collapse.

$$\delta_t(j) = \max_i[\delta_{t-1}(i) + \log a_{ij}] + \log b_j(t)$$

where a_{ij} is transition structure, $b_j(t)$ is observation likelihood

$$\Gamma_t = \{ j : \gamma_t(j) \geq \tau \}$$

Corridor retention threshold τ maintains structured ambiguity

> "Under degraded observation, the system does not have to collapse instantly to one route edge or one path label. It can retain structured ambiguity while remaining traversally coherent."

CHAPTER 3 — 4

Equilibrium as Governing Principle

Net Traversal Imbalance

$$\Delta_t = \text{Destabilising Load} - \text{Stabilising Structure}$$

Equilibrium when: $\Delta_t = 0$

↑ Destabilising

- Observation drift
- Structural ambiguity
- Breach pressure
- Inconsistency

↓ Stabilising

- State coherence
- Governance stability
- Route continuity
- Restoration potential

Governance Stability

Navigation under uncertainty is not only a probability problem—it is a **decision-governance problem**. A model that commits too fast can fail structurally even when local evidence looks strong.

$$G(t) = w_c C(t) + w_p P(t) - w_v V(t)$$

$C(t)$	$P(t)$	$V(t)$
Avg. cooperation	Policy alignment	Volatility

Key Insight: The model asks not merely which state is most likely, but whether the system is sufficiently governed to commit safely.

Recovery & Degradation Curves

The model represents not merely current state, but expected recovery capacity and residual degradation pressure over time—essential in tunnel, dropout, drift, or ambiguity-heavy settings.

$R(t) = E^* - e^{-kt}$
Recovery potential over time

$D_h(t) = D_0 e^{-\lambda t}$
Degradation intensity decay

Safe Operating Envelope

$MSS(t) = C(t) / (C(t) + D(t))$
Moral/Structural Stability

$R_{sys}(t) = w_d d(t) + w_v V(t)$
Systemic Risk

Safe Conditions:
$MSS(t) \geq \theta_M$ AND $R_{sys}(t) \leq \theta_R$

CHAPTER 5 — 6

Governance Stability and Safe Operating Envelope

The Governance Layer

A model that commits too fast can fail structurally even when its local evidence looks strong. NashMark introduces a **governance layer** that prevents premature state transitions.

Cooperation
$C(t)$

Alignment
$P(t)$

Volatility
$V(t)$

$$G(t) = w_c C(t) + w_p P(t) - w_v V(t)$$

Safe Operating Envelope

The model is constrained by a **safe operating envelope**. Navigation systems should not be forced into hard commitment when structural conditions are unsafe.

Moral/Structural Stability

$MSS(t) = C/(C+D)$

Must satisfy: $MSS(t) \geq \theta_M$

Systemic Risk

$R_{sys}(t) = w_d d + w_v V$

Must satisfy: $R_{sys}(t) \leq \theta_R$

Gating Logic

Branch commitment, transition hardening, and aggressive restoration are **gated by lawful state**, not by raw probability alone.

1. **Check MSS**
   Is moral/structural stability above threshold?
2. **Check Risk**
   Is systemic risk within acceptable bounds?
3. **Allow Commit**
   Only then permit hard state transition

Safety First

This prevents the system from making irreversible commitments when internal conditions suggest instability.

PART II

Applied Navigation Architecture

Implementation layers and operational mechanisms

CHAPTER 7 — 9

NashMark AI Architecture Overview

Six-Layer Architecture

The applied NashMark navigation model is composed of six integrated layers that distinguish it from plain recursive estimators.

1. **Latent Traversal Inference**
   Probabilistic inference over candidate route states
2. **Corridor Retention**
   Maintain plausible traversal set under uncertainty
3. **Equilibrium Refinement**
   Inner convergence under live observation
4. **Governance Stability**
   Multi-agent convergence assessment
5. **Safe-Envelope Gating**
   MSS and risk threshold enforcement
6. **Dynamic Restoration**
   Recovery curves and commit control

Key Distinction

The system is not simply asking where the point estimate lies. It is evaluating:

- **Coherence**
  Does current traversal remain structurally coherent?
- **Governability**
  Is the system sufficiently governed to commit safely?
- **Safety**
  Is it safe to harden the current state?

Live Step Signals

At each step, the system derives key state signals from observations:

Pressure	**Ambiguity**
Stable Obs.	**Breach**

CHAPTER 10 — 11

Dynamic Gain Modulation and Temporal Commit Gating

Dynamic Gain Modulation

NashMark AI does not apply a single fixed correction gain. It **modulates the navigation update** according to equilibrium state, ambiguity, governance stability, recovery potential, and degradation load.

Observation Gain
Controls how strongly live observation pulls current state

Target Gain
Controls corridor/route target pull strength

Recovery Gain
Controls restoration toward coherent traversal

Action-State Conditioning

COOPERATE
Stronger target and recovery structure

HOLD
Balanced correction behavior

DEFECT
Increased observation pull, reduced commitment

Temporal Commit Gating

A single-step dominance spike should not force branch switching. The system requires **persistence before hard commitment**—a hysteresis mechanism preventing branch flapping.

$$\text{Commit at } t \Leftrightarrow D_t \wedge C_t \wedge G_t \wedge \Sigma_t$$

- D_t: Dominance validity
- C_t: Persistence streak
- G_t: Governance sufficiency
- Σ_t: Safe envelope

Four Questions Before Commit

1. Is the candidate genuinely dominant?
2. Has that dominance persisted?
3. Is the system governable enough?
4. Is the system inside safe envelope?

Behavioral Outcomes

Sharp
Stable conditions

Cautious
Under ambiguity

Restorative
Under drift

Conservative
Under breach

PART III

Benchmark Results

Empirical validation across five challenging scenarios

CHAPTER 14 — 18

Benchmark Scenarios and Results

Five Test Scenarios

1. **Clean Route**
   Benign conditions, baseline performance
2. **Urban Canyon**
   Structured distortion, multipath effects
3. **GNSS Dropout / Tunnel**
   Signal absence, blackout conditions
4. **Ambiguous Junction** ★
   Structural ambiguity, branch uncertainty
5. **Drift Recovery** ★
   Severe drift, restoration dynamics

Benchmark Objective

Test whether NashMark AI improves **continuity-sensitive traversal** under degraded, ambiguous, or drift-heavy conditions relative to conventional baselines.

Comparative Results

Ambiguous Junction (Key Win) — Best

Kalman	HMM	NashMark
3.58 Mean Error	5.73 Mean Error	3.08 Mean Error ✓

0% false branch commits | 87.7% continuity

Drift Recovery (Strong Performance) — Best

Kalman	HMM	NashMark
1.97 Mean Error	5.88 Mean Error	1.84 Mean Error ✓

100% continuity | 100% restoration efficiency

GNSS Dropout / Tunnel — Strong

Kalman	HMM	NashMark
1.60 Mean Error	7.85 Mean Error	2.56 Mean Error

98.5% continuity | 0% false branch rate

Note: Kalman remains strongest in clean conditions. NashMark excels in degraded/ambiguous scenarios.

Conclusion & Product Direction

Established Claim

NashMark AI is a **benchmarked navigation and recovery architecture** with strongest evidence in ambiguity-sensitive continuity and drift-recovery tasks.

Product Architecture

- Rooted label authority where available
- Sharp continuous estimator where useful
- **NashMark as equilibrium governor**

Dual Deployment Modes

Standalone

Navigation logic in label-light space

Governing Layer

Higher-order control around rooted labels

"Product-grade architecture, even if further tuning remains"