# Post-Actuation Verification Through Discrepancy Classification

by Nick Clark | Published April 25, 2026

## What Verification Through Discrepancy Classification Specifies

After an actuator command commits, the verification stage observes: the sensed effect through whatever readback channels are available (encoder feedback, sensor response, environmental observation, downstream telemetry), and the predicted effect from the planning and control models that produced the command. The discrepancy classifier produces one of: nominal (sensed matches predicted within expected envelope), expected-noise (discrepancy within sensor and environmental noise floors), anomaly (discrepancy exceeds noise but does not match a fault or adversarial pattern), fault (discrepancy matches a known fault signature), adversarial-interference (discrepancy matches an adversarial-attribution pattern from the disruption-modeling library).

Each classification is itself a credentialed observation: signed by the verifying agent, with the supporting sensed and predicted values recorded in lineage. The classification propagates through the mesh as a credentialed observation, available for consumption by neighboring units, infrastructure agents, and regulatory authorities.

## Why Actuation Should Not Terminate at Commit

Current autonomous architectures typically treat actuator commit as the terminal step. The command is sent, the actuator responds, and the planning loop proceeds. Sensor feedback flows into the next planning cycle but is not architecturally tied to the previous actuation as verification of that specific commit.

This produces structural blindness to failure modes that depend on actuation-effect mismatches. An actuator that drifts from its command, a control loop that introduces delay, an adversarial intervention that displaces the effect, all produce sensed-vs-predicted discrepancies that the architecture has no first-class concept for. Post-actuation verification gives the discrepancy a structural home.

## How Verification Composes With Mode Selection

The verification stage runs after every commit regardless of mode. In full mode, verification confirms the action proceeded as predicted. In partial mode, verification confirms the partial commit produced the expected partial effect. In stage-gated mode, verification at each stage gates progression to the next stage. In emergency-overridden mode, verification produces audit lineage of the override's actual effect.

Discrepancy classification feeds back into the next admissibility evaluation. A unit that has been producing anomalies receives reduced confidence weighting in subsequent admissibility computations. A unit producing adversarial-interference classifications triggers cross-system alerts through the mesh. The feedback closes the loop between expected behavior and observed behavior structurally.

## What This Enables for Cross-System Coordination

When verification observations propagate through the mesh, downstream consumers gain awareness of upstream actuation reality. A vehicle whose lane-change actuation

produced an anomaly broadcasts the anomaly; following vehicles consume the anomaly observation through composite admissibility and modulate their own response. Cross-system coordination becomes structural rather than reconstructed from telemetry.

Audit and incident reconstruction become tractable. The lineage of a crash includes the sensed-vs-predicted discrepancies of every actuation in the moments before the event, with their classifications and the conditions under which each was committed. The architecture produces what current accident-investigation processes reconstruct manually from log archaeology, structurally and at scale.