



André Dupke

Hamburg, August 7, 2026

scaletimedynamics.com

info@scaletimedynamics.com

Orcid | Academia | YouTube | Facebook

Version: 1.0

© 2026 André Dupke. Licensed under CC BY 4.0

DOI: doi.org/10.5281/zenodo.21833505

STT Framework Link: Scale-Time Dynamics

STAR DRIVE

Sampled Transport by Alias Ratcheting

A Lock-Window Pulse-Scheduling Architecture for Conventional Propulsion

EN-002 EVIDENCE-PACK RELEASE SPECIFICATION

This document lives with the code, not in the paper. It specifies the release artifacts that close Stage 0b of EN-002 §7. Before EN-002 is released, both A1 and A2 must exist in the evidence pack as their own reproducible routines, each with regenerated tables in `results.json` alongside E1–E5 (which currently stores widths only) and its own automated verification test of the same form the pack applies there. Neither substitutes for the other. Until both ship, every six-decimal A1/A2 value in EN-002 - and every design rule resting on them, including rules 1, 2, 8, 9, and 11 - is provisional.

v1.0 revisions (audit pass): the operational-guard requirement is restated as a Stage-1 gate with a Stage-0b branch map and verifier (§A2); A1 gains explicit orbit-pair and convergence-failure requirements; the uniformity search requirement is restated as branch-covering; and a pinned environment and hardened regeneration workflow are now release conditions.

A1 artifact - the anchor-gain map

The routine must record:

- the map and its sign convention, stated explicitly for both the substrate and loop forms, since the two appear in different sections of the note;
- the function evaluating $g = f^q(x) - x - p$ and the detection rule: a root candidate is detected either by a sign change between adjacent grid nodes or by a node satisfying $|g| \leq \text{tol}$ - alternative mechanisms, not a conjunction, since a sign-change test alone misses tangent roots and roots landing on grid nodes (the $1/2$ orbit's points $z = 0, \frac{1}{2}$ land on nodes at its window centre). Candidates are then refined and independently verified;
- the Ω and x grids and refinement tolerances, with window boundaries solved from the simultaneous saddle-node system

$$f_{\Omega}^q(x) - x - p = 0, \quad \frac{\partial f_{\Omega}^q(x)}{\partial x} - 1 = 0,$$

residual tolerances recorded for both equations; the dense-grid multiplier check is retained as independent detection and bracketing, not as the final tangency verification;

- the stable/unstable orbit pair inside every window, with the attracting branch identified - all roots refined, grouped into cycles under the map, classified by multiplier, and the attractor confirmed by iteration from a generic seed with the match distance recorded. The check runs at more than one parameter per window (released implementation: centre and $\pm$ half the half-width) so the pair is established across the window, not at a single point. A window in which no pair is found, or in which the attractor is not the stable branch, is a release failure;
- fail-safe orbit convergence: an orbit iteration that exhausts its budget without meeting the convergence tolerance must raise or record a failure carrying the parameter, the residual and the iteration count. Returning an unverified iterate is prohibited, and callers must not swallow the failure;
- rotation-number classification at the rational-label test points - $\Omega = 1/4$ (expected $W_0 = 5/24$) and $\Omega = 1/3$ (expected $W_0 = 8/25$) at $\kappa = 0.95$ - with convergence and

plateau-identification criteria recorded, since EN-002 §2.4 states these wrong-lock outcomes as numerical claims;

- the transient length and sign convention for the event-phase computation and its dependence on the declared realization, at $\delta_g = 0$;
- the event-phase error functional written mathematically - cyclic indexing of the $m\tilde{q} + \tilde{p}$ events within a superperiod, the uniform grid, and the minimisation over the constant offset b defining $\text{RMSE}_{\text{shape}}$;
- a root solve for the $\tilde{p} = 2$ uniformity condition, and for $\tilde{p} \geq 3$ a branch-covering global minimisation: the objective kinks where an event changes interval assignment, so the window must be partitioned along the zeros of an exact assignment indicator (the distance of the closest event to a fiducial - a counts-based signature is blind to cyclic permutations and must not be used), a bounded minimisation run on every branch, and the competing branch minima, the branch boundaries, the optimality checks (interior gradient and curvature, or boundary attainment) and the sampling density all recorded. A single bounded call over a multi-branch objective does not satisfy this requirement. Any search that is a plain scan must be labelled as a scan in the output, and global nonexistence must not be claimed from either.

A2 artifact - guard, continuation, and the robust controller

The routine must record:

- the complete branch structure of the guard law over the declared domain $0 \leq \delta_g < 1$: every branch enumerated by walking from breakpoint to breakpoint $\delta_{g,j}^{\text{bp}} = x_{n+1} - \ell_j$ (kept notationally distinct from the shape-optimal guard $\delta_g^{\text{opt}} = -C/V$), with each branch's validity interval, its own (R_0, C, V) relative to its own reference guard, an in-branch verification of the closed form against direct computation, and its own admissibility test for $-C/V$. Enumeration must be reported as complete (domain exhausted) and contiguous;
- the operational guard. EN-002 rule 7 fixes δ_g from the first-passage budgets, which are preregistered before Stage 1; the value is therefore not fixed at Stage 0b. What Stage 0b must supply instead is (i) the complete branch map above and (ii) a verifier that, given a candidate guard, returns its branch, that branch's coefficients, the law value there, and an out-of-domain rejection otherwise. Running the verifier on the preregistered

guard is a Stage-1 entry condition, and its output must be recorded beside the preregistration. Quoting coefficients for a guard without this check is prohibited;

- δ_g as an explicit parameter, with shape deviation reported both at $\delta_g = 0$ (the A1 reference) and at any operational guard once fixed, together with the fitted C and V ;
- an adaptive or densified κ scan, with the continuation scheme that tracks each window as it narrows, and anchor consistency: the continuation evaluates the anchor $\kappa = 0.95$ with the same boundary/orbit solver as A1 and must reproduce the canonical anchor values within stated tolerance; scan-accuracy outputs are labelled as such and printed at commensurate precision;
- the robust operating-point computation: $\Omega_{p/q}^{\pm}(\kappa)$ over the κ range spanned by the prediction ratio, the intersection test over ρ , and the bisection returning $\epsilon_{\max}$, with the ρ grid and interpolation scheme recorded, for both the rate-blended and phase-step realizations - reporting the guarded null as window-containment-invariant with its operational tolerance set by the effective-gain cap (rate-blended) and as not prediction-limited (phase-step);
- the margin-constrained command optimizer, in the staged order of EN-002-S §S3: $\mathcal{G}_n$ (window intersection, gain cap, slew limits, and the explicit hard-feasibility predicate HF_{n,H_c} covering count, minimum separation, actuator, deadline, and library-domain constraints); the ceiling $\mathcal{M}_n^{\text{ceil}}$ computed on that pre-margin set so the calibration is not circular; the fixed preregistered floor $\mathcal{M}_{\min}^{\text{design}}$ compared against it; and $\mathcal{F}_n$ last - with $\mathcal{G}_n = \emptyset$ and $\mathcal{F}_n = \emptyset$ reported as distinct faults, $\mathcal{M}_n$ kept notationally and functionally distinct from the ledger's M_n , and suspension enforced on the ledger: while suspended the controller must refuse new threshold commitments and preserve the existing ledger intact. Then the cost: J_n with its normalizing scales R_{ref} , P_{ref} and weights; the definition of P_{control} ; the declared R_n (target, horizon H_c , guard and its branch, state propagation, phase source); the path set $\mathcal{P}_{n,H_c}$ and future setpoint policy; the optimization solver or minimization grid; and the declared five-step total tie order. Release decision: v1.0 releases the controller explicitly as a family. A2 verifies the feasible-set architecture, window intersections, margin ceiling, and analytic anchors; the unique operational instantiation is deferred to the Stage-1 preregistration, and no quoted setpoint inherits [S:A2] from the solver architecture alone;
- the lock-loss evaluator with onset/decision indexing and, for finite records, an explicit undetermined outcome: an episode whose recovery horizon extends past the end of

the record, or a branch-offset run still active at the final sample, must be reported as undetermined rather than as failed or survived.

Regression tests

A static test that only validates checked-in JSON verifies the file, not the code. The suite must therefore include regenerating tests that recompute A1/A2 quantities in-process and compare them with the released `results.json` - at minimum: a window boundary solved from an independent bracket, an event-phase deviation, an orbit pair with its classification, a guard-branch coefficient set, and an $\varepsilon_{\max}$ value. In addition:

- verify $(\Omega_n^{\text{cmd}}, \kappa_n) \in \mathcal{F}_n$ at every simulated update, and recompute the returned worst-case margin independently - from the definition, on a finer ρ grid, without calling the routine under test;
- verify slew-limit compliance $|\Delta\Omega^{\text{cmd}}| \leq s_\Omega$, $|\Delta\kappa| \leq s_\kappa$;
- verify the five-step total tie order on constructed cases in which the tie is demonstrated to survive steps (i)–(iii) - the tied margins asserted equal independently - with one case resolved at step (iv) (minimum κ) and one at step (v) (minimum ω), each also asserted deterministic across repeated calls;
- verify correct detection of $\mathcal{G}_n = \emptyset$ and $\mathcal{F}_n = \emptyset$ as distinct faults, and that an attempted commitment while suspended raises and leaves the ledger unchanged, with commitment restored after recovery;
- verify that a rollout violating HF_{n,H_c} is excluded from $\mathcal{G}_n$ rather than scored on R_n ;
- verify that the margin floor is never lowered outside a declared degraded mode;
- verify the orbit-convergence failure path: an inadequate budget raises, carrying parameter, residual and iteration count, and the failure propagates to callers;
- verify the lock-loss evaluator on a recovering-then-failing pair of episodes, on a case where recovery precedes the excursion decision time, and on a finite record that ends mid-episode, which must be reported undetermined while the same excursion in a longer record is reported failed;
- reproduce the fixed-gain static integer-window ceiling $\Omega^{\text{cmd}} = m + \varepsilon\kappa_n/2\pi$ with margin $\kappa_n(1 - \varepsilon^2)/2\pi - \varepsilon m$ as an analytic regression check under its own conditions - not as a test of the general $\mathcal{M}_n^{\text{ceil}}$ - together with the numerical procedure behind the quoted 2/3 example.

Environment, workflow, and outputs

- **Pinned environment.** The interpreter and library versions are pinned in ``pyproject.toml`` and ``requirements.lock``; the regeneration driver refuses nonmatching versions; and the exact versions are recorded in ``results.json/environment`` and asserted by the test suite. The released artifacts were generated under CPython 3.14.7, NumPy 2.5.1, SciPy 1.18.0, pytest 9.1.1.
- **Cache integrity.** Multi-phase regeneration must stamp each phase cache with a specification hash over the routine sources, environment-declaration files, and the complete numerical runtime fingerprint (interpreter and library versions, Python platform, operating system and architecture), and refuse to combine caches whose hash differs from the current one, naming both hashes in the error.
- **Atomic outputs.** Every artifact is written to a temporary file in the destination directory and renamed into place, so an interrupted or failing run never leaves a partial ``results.json`` or ``manifest.json``.
- **Nonzero exit on failure.** The driver exits nonzero on any manifest mismatch and on artifact-level failures (a window without an orbit pair or a stable attractor, an incomplete guard enumeration, an unverified guard branch).
- **Outputs.** Regenerated ``results.json`` (schema versioned; A1: boundaries, centres, widths, saddle-node residuals, orbit pairs, unguarded event phases, rational-label classification, uniformity results with solver, bounds, convergence and optimality records; A2: guard branch enumeration with validity intervals, κ -continuation with accuracy labels, $\epsilon_{\max}$ for both realizations, optimizer example outputs) and a manifest mapping every EN-002 manuscript value to its generating artifact - one canonical value per number, scan-accuracy duplicates labelled.
- **Clean-room verification.** Before release, the package is regenerated in an empty directory on the same pinned runtime and recorded platform and the result compared with the released ``results.json``; the comparison must be identical apart from the generation timestamp. Cross-platform equality is not implied: the platform fingerprint deliberately invalidates caches and distinguishes artifacts produced by different numerical builds.
- **Release rule.** EN-002 is not released with the A1/A2 values unless the A1 routine, the A2 routine, their verification tests, and the regenerated ``results.json`` ship at the same time.

Scope note

The pack remains one-mode and one-observer, Gaussian readout noise on the estimation side, E5's circle-map coupling on the invertible domain, finite-run ensembles. It supports exactly the [S:En]-tagged claims of EN-002, extended by the deterministic analyses A1 and A2 of the same underlying model, and nothing more: it does not test momentum exchange, reservoir coupling, actuator response, hardware event timing, dynamic acquisition, dynamic lock retention under a moving cadence, long-horizon slip statistics, controller-tick realization effects, super-critical coupling, or an assembled thrust loop. Every claim of that kind in EN-002 is tagged [C] and paired with a stage in its §7.