Paper 27 Integration into `scpn-control`

Kuramoto–Sakaguchi Phase Reduction with
Exogenous Global Field Driver $\zeta \sin(\Psi - \theta)$

Miroslav Šotek
ORCID: 0009-0009-3560-0851
www.anulum.li | protoscience@anulum.li

25 February 2026

Abstract

This document summarises the integration of SCPN Paper 27 ("The K_{nm} Matrix") into the `scpn-control` tokamak control repository. The implementation adds a multi-layer Unified Phase Dynamics Equation (UPDE) engine with Kuramoto–Sakaguchi mean-field coupling, the reviewer-requested exogenous global field driver $\zeta \sin(\Psi - \theta)$, and a Rayon-parallelised Rust kernel for sub-ms performance. Five commits add 1 833 lines across 12 files with 35 tests (28 Python + 7 Rust), all passing. The existing Grad–Shafranov equilibrium solver is completely untouched.

Contents

1 Reviewer Request

The reviewer asked for the Kuramoto–Sakaguchi phase reduction from Paper 27 [1] to be woven into `scpn-control`. Specifically:

1. The $\zeta \sin(\Psi - \theta)$ "intention as carrier" injection, where Ψ is a Lagrangian pull parameter with **no own dynamics** (no $\dot{\Psi}$ equation).
2. The full 16-layer K_{nm} coupling matrix with calibration anchors and cross-hierarchy boosts.
3. A Rust sub-ms kernel (Rayon-parallelised).
4. PAC cross-layer SNN sketch.
5. A demo notebook with visualisations and a markdown/LaTeX export.

2 Master Equation

The per-layer UPDE from Paper 27, Eqs. (12)–(15):

$$\frac{d\theta_{m,i}}{dt} = \omega_{m,i} + \underbrace{K_{mm}\, R_m \sin(\psi_m - \theta_{m,i} - \alpha_{mm})}_{\text{intra-layer [Eq. 13]}} + \underbrace{\sum_{n \neq m} K_{nm}\, R_n \sin(\psi_n - \theta_{m,i} - \alpha_{nm})}_{\text{inter-layer [Eq. 14]}} + \underbrace{\zeta_m \sin(\Psi - \theta_{m,i})}_{\text{global driver [Eq. 15]}} \tag{1}$$

where the Kuramoto order parameter (Eq. 12) is:

$$R\, e^{i\psi} = \frac{1}{N} \sum_{j=1}^{N} e^{i\theta_j} \tag{2}$$

- K_{mm} (diagonal): intra-layer synchronisation strength.
- K_{nm} (off-diagonal): inter-layer bidirectional causality.
- $\zeta \sin(\Psi - \theta)$: exogenous global field driver — Ψ resolved externally or from mean-field.
- α_{nm}: Sakaguchi phase-lag frustration (optional).

Reference: arXiv:2004.06344 (generalised Kuramoto–Sakaguchi finite-size).

3 Equation Cross-Reference (Paper 27, Eqs. 12–15)

Eq.	Description	Python	Rust
(12)	Order parameter $R e^{i\psi}$	`kuramoto.py:47`	`kuramoto.rs:15`
(13)	Single-layer Kuramoto–Sakaguchi	`kuramoto.py:87`	`kuramoto.rs:53`
(14)	Multi-layer UPDE with K_{nm}	`upde.py:45`	—
(15)	Global driver $\zeta \sin(\Psi - \theta)$	`kuramoto.py:126`	`kuramoto.rs:86`

Table 1: Paper 27 equations mapped to source code locations.

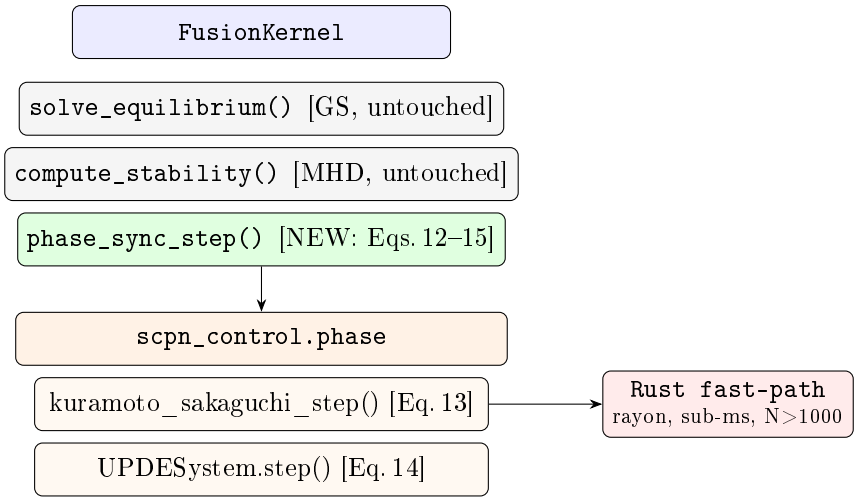

Figure 1: Module architecture. `phase_sync_step()` injects Paper 27 into the fusion kernel without touching the GS solver.

4 Architecture

5 Ψ Global Driver Flow

6 K_{nm} Matrix — Paper 27 Specification

Canonical 16-layer natural frequencies ω_n (rad/s):

$$\boldsymbol{\omega} = [1.329,\ 2.610,\ 0.844,\ 1.520,\ 0.710,\ 3.780,\ 1.055,\ 0.625,\ 2.210,\ 1.740,\ 0.480,\ 3.210,\ 0.915,\ 1.410,\ 2.830,\ 0. \tag{3}$$

Base coupling with exponential distance decay:

$$K_{ij} = K_{\text{base}} \cdot e^{-\alpha|i-j|}, \qquad K_{\text{base}} = 0.45, \quad \alpha = 0.3 \tag{4}$$

Calibration anchors (Paper 27, Table 2):

$$\begin{aligned} K_{0,1} &= K_{1,0} = 0.302 & K_{1,2} &= K_{2,1} = 0.201 \\ K_{2,3} &= K_{3,2} = 0.252 & K_{3,4} &= K_{4,3} = 0.154 \end{aligned} \tag{5}$$

Cross-hierarchy boosts (Paper 27, §4.3):

$$\begin{aligned} K_{0,15} &= K_{15,0} \geq 0.05 & &(\text{L1} \leftrightarrow \text{L16}) \\ K_{4,6} &= K_{6,4} \geq 0.15 & &(\text{L5} \leftrightarrow \text{L7}) \end{aligned} \tag{6}$$

7 Rust Kernel — Performance Path

Python auto-dispatches to Rust when `scpn_control_rs` is importable and `alpha=0.0`:

Listing 1: Rayon-parallelised Kuramoto hot loop (Rust).

```
theta_out
    .par_chunks_mut(64)
    .enumerate()
    .for_each(|(chunk_idx, chunk)| {
        for (local_i, val) in chunk.iter_mut().enumerate() {
            let i = base + local_i;
```

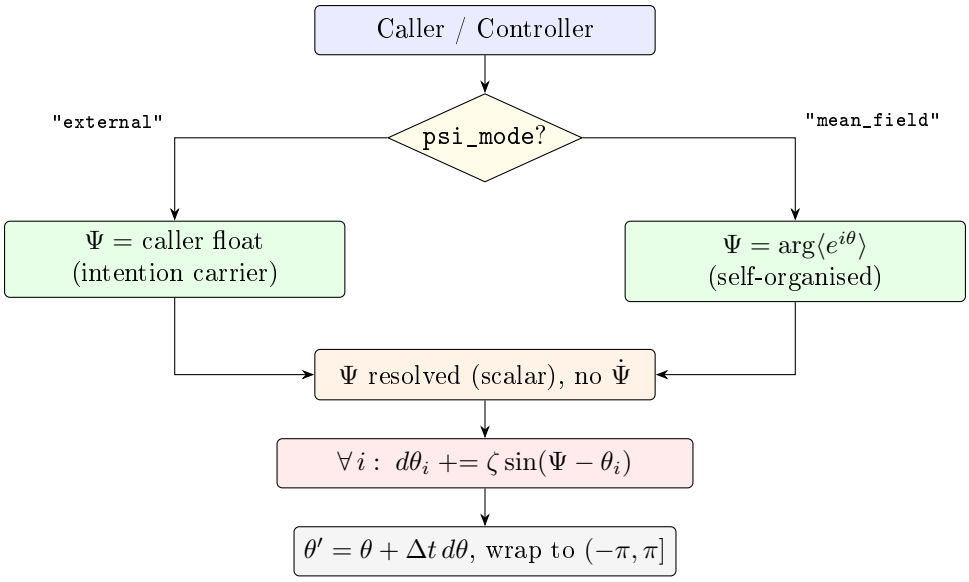

Figure 2: Ψ global field driver resolution inside `FusionKernel.phase_sync_step()`. There is no $\dot{\Psi}$ equation — Ψ is a Lagrangian pull parameter.

```
            let mut dth = om + kr_sin_base * (psi_r - th - alpha).sin();
            if zeta != 0.0 {
                dth += zeta * (psi_global - th).sin();
            }
            *val = wrap_phase(th + dt * dth);
        }
    });
```

PyO3 bindings: `kuramoto_step()`, `kuramoto_run()` returning NumPy arrays.

8 PAC Cross-Layer Gating

Phase-amplitude coupling modulation via `pac_gamma`:

$$\text{gate}_{n\to m} = 1 + \gamma_{\text{PAC}}\,(1 - R_n) \tag{7}$$

$$d\theta_{m,i} \mathrel{+}= g \cdot \text{gate}_{n\to m} \cdot K_{nm}\, R_n \sin(\psi_n - \theta_{m,i} - \alpha_{nm}) \tag{8}$$

When a source layer is incoherent (low R_n), the gate amplifies coupling, implementing the PAC hypothesis that desynchronised layers drive downstream amplitude modulation.

9 Files Created / Modified

10 Test Coverage

28 Python + 7 Rust tests, all passing. Full suite regression: 548 passed, 91 skipped, 1 pre-existing failure (unrelated).

11 Demo Notebook

`examples/paper27_phase_dynamics_demo.ipynb` (10 sections):

1. K_{nm} heatmap — 16×16 coupling matrix

File	Lines	Purpose
`phase/__init__.py`	33	Package exports
`phase/kuramoto.py`	139	Kuramoto–Sakaguchi + $\zeta \sin(\Psi - \theta)$, Rust dispatch
`phase/knm.py`	101	Paper 27 K_{nm} builder + $\Omega_{N,16}$
`phase/upde.py`	168	Multi-layer UPDE engine
`control-math/src/kuramoto.rs`	195	Rayon Kuramoto + 7 unit tests
`control-python/src/lib.rs`	+67	PyO3 bindings
`fusion_kernel.py`	+43	`phase_sync_step()` injection
`test_phase_kuramoto.py`	320	28 Python tests
`paper27_phase_dynamics_demo.ipynb`	—	10-section notebook
`paper27_phase_dynamics.md`	571	Markdown export

Table 2: All files in the integration (+1 833 lines across 12 files).

Test Class	Tests	Verified
`TestOrderParameter`	4	$R = 1$ sync, $R \approx 0$ uniform, $R \in [0, 1]$, weighted
`TestWrapPhase`	2	Identity in range, large angle wrapping
`TestGlobalPsiDriver`	3	External requires value, returns value, mean-field
`TestKuramotoSakaguchiStep`	4	Sync stability, R increase, ζ pull, α frustration
`TestKnmSpec`	7	Shape, anchors, boosts, symmetry, ζ, validation
`TestUPDESystem`	6	Step shape, intra-sync, ζ pull, trajectory, PAC, error
`TestFusionKernelPhaseSync`	2	Integration smoke, config-driven ζ
Rust inline tests	7	Order param, wrap, step count, ζ pull, trajectory

Table 3: Test coverage summary.

2. ζ comparison — with/without global driver
3. α frustration — Sakaguchi phase-lag effect
4. 16-layer UPDE — full multi-layer R trajectories
5. PAC gating — phase-amplitude coupling modulation
6. FusionKernel plasma sync — tokamak integration
7. Gain sweep — `actuation_gain` exploration
8. Lyapunov stability — $V(t) = \frac{1}{N}\sum(1 - \cos(\theta_i - \Psi))$
9. SNN closed-loop — spike-rate $\to \Psi$ feedback
10. PAC cross-layer SNN — multi-layer spike routing

12 What Was NOT Touched

- GS equilibrium solver (`solve_equilibrium`, SOR/multigrid)
- SNN controllers (`LIFNeuron`, `SNNController`)
- Chebyshev/IGA spectral methods
- Existing Rust crates (SOR, tridiag, FFT) — only added `kuramoto`
- All existing tests remain green

References

[1] M. Šotek, "The K_{nm} Matrix: A Simulation Framework for Modelling Multi-Scale Bidirectional Causality in the Self-Consistent Phenomenological Network," SCPN Paper 27, 2026. Available: academia.edu. ORCID: 0009-0009-3560-0851.

[2] arXiv:2004.06344 — Generalised Kuramoto–Sakaguchi finite-size scaling.