HC-SLAM “Hot–Cold”: Navigation via Cyclic Foliation from a Scalar Measurement

Mirado Rajaomarosata Luc Jaulin Lionel Lapierre Simon Rohou

December 17, 2025

Setup: a robot with a single scalar sensor

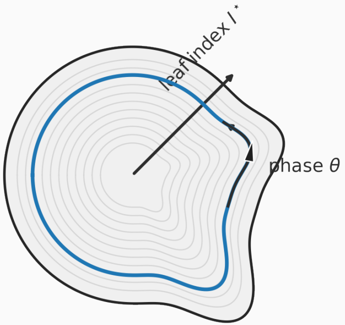
- Robot is *lost*: no GPS / lidar / camera.
- Only one scalar stream remains:
  - temperature, potential, RSSI, acoustic RMS, concentration, ...
- Goal: navigate *without* building a global metric map.

Cyclic foliation induced by a controller

- A reactive controller stabilizes the robot on a *limit cycle*.
- Varying the setpoint yields a family of nested cycles:

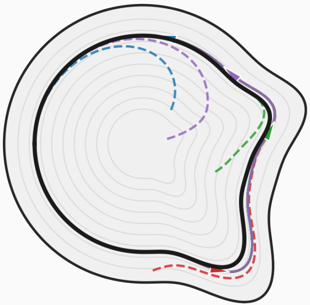
$$\{\Gamma_I\}_{I \in \mathcal{I}} \;\Rightarrow\; \text{cyclic foliation.}$$

- Leaf coordinate: setpoint (leaf index) $I^\star$.
  Intra-leaf coordinate: phase $\theta \in \mathbb{S}^1$.

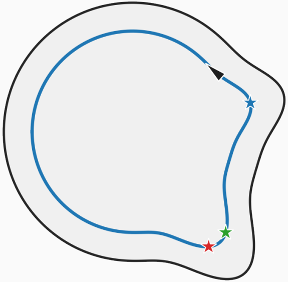


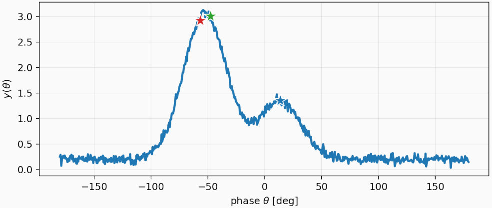
Step 1: reactive setpoint control $\Rightarrow$ converge to a limit cycle

- Apply a standard feedback to reach the target leaf $\Gamma_{I^\star}$.
- From diverse initial conditions, trajectories contract toward the same cycle.
- Near geometric “corners”, tracking is imperfect: deviations create *informative events*.



Step 2: on the cycle, the scalar trace contains localized events





Key intuition

Most of the useful information is *localized* (peaks / "hot–cold" events) rather than spread everywhere.

Leaf-level SLAM on $\mathbb{S}^1$: measurement + evolution + gauge

Leaf-level state: phase $\theta_t \in \mathbb{S}^1$.

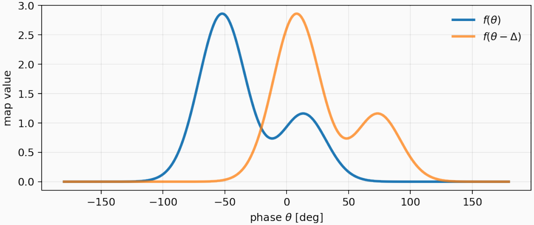
Evolution model (prediction):

$$\theta_{t+1} = \mathrm{wrap}(\theta_t + \omega_t \Delta t + \nu_t)\,,$$

Measurement model:

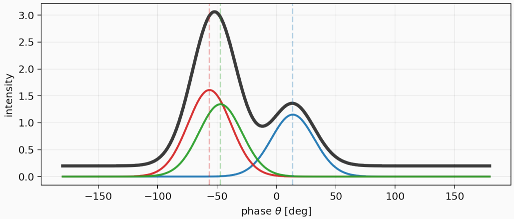
$$y_t \approx b + G\, f(\theta_t; x) + \varepsilon_t,$$

Gauge (rotation) invariance: only the profile shape modulo a global shift is observable.

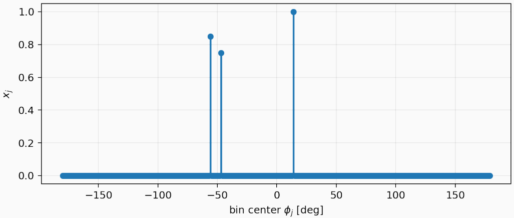


Map model: sparse periodic kernels (physics-agnostic)

$$f(\theta;x)=\sum_{j=1}^{N} x_j\,K(\theta-\phi_j)\quad\triangleq\quad a(\theta)^\top x,\qquad \|x\|_0\ll N.$$

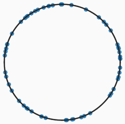


localized kernels ⇒ peaks / events



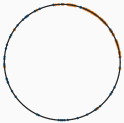
sparse coefficients x (interpretable landmarks)

Phase localization with a particle filter on $\mathbb{S}^1$



Initialize: $\{(\theta_0^{(i)}, w_0^{(i)})\}_{i=1}^{N_p}$, $w_0^{(i)} = 1/N_p$.

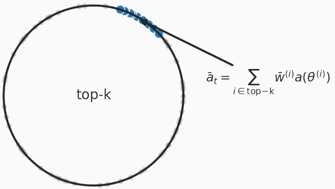
Predict: $\theta_{t+1}^{(i)} \leftarrow \mathrm{wrap}\Big(\theta_t^{(i)} + \omega_t \Delta t + \nu_t^{(i)}\Big)$.



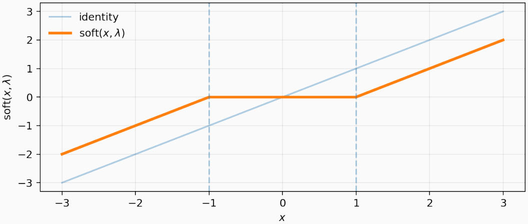
Weight / resample: $r(\theta) = \frac{y_t - \left(b + G\, a(\theta)^\top x\right)}{\sigma_y}$.
$w^{(i)} \propto w^{(i)} \exp\big(-\rho_\delta(r(\theta^{(i)}))\big)$.

Estimate (circular mean): $\hat{\theta}_t = \arg\Big(\sum_i w^{(i)} e^{\mathrm{i}\theta^{(i)}}\Big)$

Online map learning: top-k expectation + proximal ℓ_1 update



gray: all particles blue: top-k

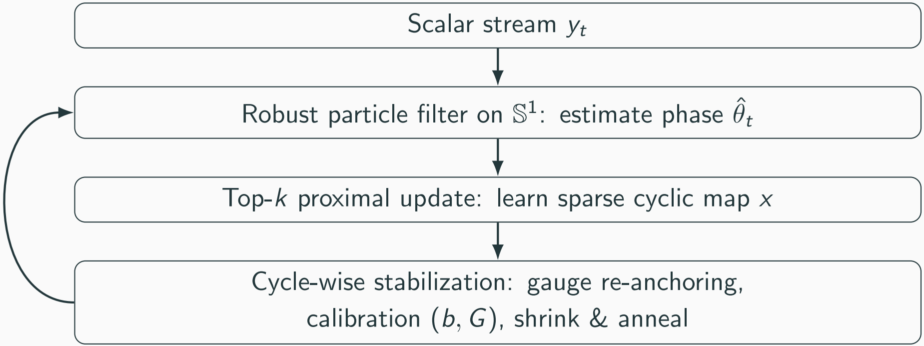


proximal step uses soft-thresholding

$$\bar{a}_t = \sum_{i \in \text{top-}k} \bar{w}^{(i)} a(\theta_t^{(i)}).$$

$$x^+ = \mathrm{soft}\left(x + \eta \frac{G}{\sigma_y} \psi(r_t)\, \bar{a}_t,\ \eta\lambda\right).$$

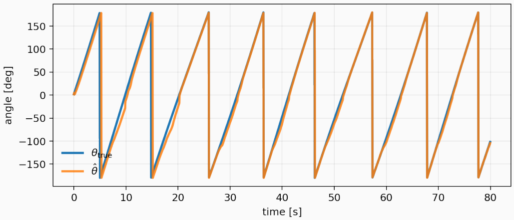
HC-SLAM in one slide



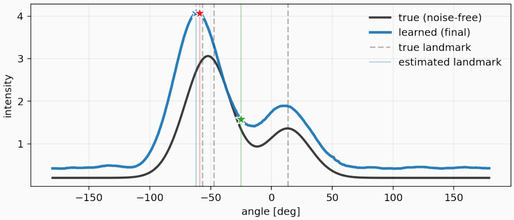
Outcome

Each stabilized leaf becomes *navigable*: online phase localization + a compact hot–cold landmark map.

Simulation results (single leaf)



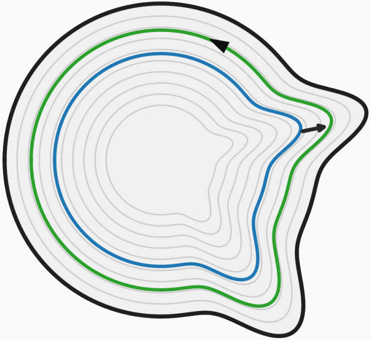
phase tracking over multiple cycles



learned profile and estimated landmarks vs. ground truth

Navigation via foliation (no metric map)

- **Leaf selection:** change setpoint $I^\star$, wait for convergence.
- **Intra-leaf navigation:** run HC-SLAM to localize θ and target a hot/cold landmark.
- Task space becomes $(I^\star, \theta)$, not a full 2D metric map.



Takeaways and perspectives

- **Main message:** if the closed-loop motion contracts to a cycle, the relevant state becomes $\theta \in \mathbb{S}^1$ and a scalar trace can support *1D SLAM on a loop*.
- **Hot–Cold idea:** leverage *localized* events (peaks) as landmarks; enforce sparsity for interpretability and robustness.
- **Beyond robotics maps:** many "natural motions" (nonlinear modes / invariant sets) can be seen as low-dimensional invariant manifolds; foliations suggest SLAM on *unknown* motion primitives.
- **Next steps:** extend to back-and-forth trajectories; exploit richer path parameterizations (e.g., Fourier descriptors) to infer aspects of the loop geometry.

Thank you !

Questions?