SAGE: Semantic and Adaptive Generative Ergodicity for Autonomous Underwater Inspection

Markus Buchholz
Institute of Sensors, Signals and Systems
Heriot-Watt University, Edinburgh, UK

Abstract—**Underwater vehicles rarely get the luxury of a clean map before they have to act on it. Acoustic returns are sparse, vision is limited by turbidity and range, and there is no GPS to anchor a plan against drift. Most inspection missions still fall back on lawnmower patterns or hand-placed waypoints, which spend exactly as much time on empty seabed as on the pipe joint that actually matters. This note describes SAGE (Semantic and Adaptive Generative Ergodicity), a control architecture that instead asks the vehicle to spend its time in proportion to where the mission believes something is worth looking at, and lets that belief come directly from onboard perception rather than from a human at a planning terminal. A vision agent and a mission description are fused into a spatial probability distribution, which an ergodic controller tracks by matching the time-averaged statistics of the vehicle's trajectory to that distribution, rather than chasing a fixed reference path. We derive the ergodic metric used to close this loop, describe how the underlying distribution is built and updated online, and show that the same machinery can be inverted to turn a hazard map into an avoidance behaviour with no extra planning layer. The architecture was implemented in ROS 2 and validated on a BlueROV2 Heavy model in the Stonefish simulator, on a pipeline inspection task where several candidate anomalies compete for the vehicle's attention. The results suggest that ergodic coverage, grounded in semantics rather than geometry alone, is a natural fit for the kind of soft, uncertain, and constantly revised information that underwater sensing actually produces.**

I. Introduction

An AUV inspecting a pipeline, a reef, or a wreck is not solving a path-planning problem so much as an attention-allocation problem. The vehicle has a fixed energy and time budget, a belief about where interesting things are that is never fully resolved, and a sensor suite that keeps revising that belief as it moves. Classical trajectory tracking is the wrong abstraction for this: it asks the vehicle to converge on one committed path, and any deviation from it is treated as an error to be corrected rather than as new information to be exploited. In practice this means missions are planned offline, executed rigidly, and re-planned from scratch whenever a sonar return or a camera detection changes what the operator thought was down there.

Ergodic control offers a different contract. A trajectory is ergodic with respect to a distribution ϕ if the fraction of time it spends in any region of the workspace converges, as the horizon grows, to the probability mass ϕ assigns to that region. Instead of asking "where should the vehicle be at time t", the controller asks "has the vehicle, averaged over its history, spent time where the information is". This reframing has a property that matters a great deal underwater: the objective degrades gracefully. Interrupt the vehicle at any point in the mission and the path flown so far is already a reasonable, non-wasteful coverage of what was known up to that point, because the controller was never committed to finishing a specific route.

What was missing from the ergodic control literature, as far as we could find during this work, was a clean way to get ϕ itself from the vehicle's own senses rather than from a hand-drawn density. SAGE closes that gap: a vision agent turns camera or sonar frames into object detections, a mission description supplies the operator's intent in the form of which classes or regions matter right now, and a semantic grounding module fuses the two into the distribution that the ergodic controller tracks. The distribution is therefore a live quantity, not a static prior, and it can be inverted just as easily to keep the vehicle away from a region as to draw it toward one. This note lays out the mathematics behind that loop, the architecture that implements it, and a simulated pipeline-inspection experiment run on a BlueROV2 Heavy in Stonefish that exercises all three behaviours: covering a static distribution, avoiding one, and tracking one that moves.

The formulation builds directly on the ergodic control framework developed by Mathew and Mezić [1] and on the spectral multiscale coverage algorithm of Abraham and Murphey [3], whose reference implementation, the ergodic-control-sandbox released by the Murphey Lab at Northwestern University [6], served as the starting point for the controller used here [7], [8].

II. Related Work

Ergodic control was introduced as an alternative to potential-field and frontier-based exploration by treating coverage as a distribution-matching problem in the frequency domain [1]. Miller and Murphey formulated it as a trajectory optimization [2], and Abraham and Murphey later gave it a closed-form, receding-horizon control law, spectral multiscale coverage (SAC), that is cheap enough to run online and that this note relies on directly [3]. Underwater coverage has historically leaned on lawnmower sweeps, occupancy-grid frontier exploration, or potential fields with hand-tuned obstacle terms, none of which naturally consume a soft, multi-modal belief the way ergodic control does. Semantic scene understanding, meanwhile, has matured mostly in the image-processing literature, largely disconnected from how that understanding then shapes low-level motion. SAGE sits at the join of these two lines: it treats semantic perception as the source of the coverage objective rather than as a separate downstream annotation step.

III. Ergodic Coverage: Formulation

Let $X \subset \mathbb{R}^n$ be the (normalized) workspace and let $\phi : X \to \mathbb{R}_{\geq 0}$ be a target distribution over it, with $\int_X \phi(x)\,dx = 1$. We use the Fourier basis

$$F_k(x) = \frac{1}{h_k} \prod_{i=1}^{n} \cos\left(\frac{k_i \pi x_i}{L_i}\right), \tag{1}$$

where $k = (k_1, \ldots, k_n)$ indexes frequency and h_k is a normalizing constant so that each F_k has unit L^2 norm on X. The distribution's spectral coefficients are

$$\phi_k = \int_X \phi(x)\, F_k(x)\, dx, \tag{2}$$

and the trajectory $x(\cdot) : [0,T] \to X$ produces its own time-averaged coefficients

$$c_k\big(x(\cdot)\big) = \frac{1}{T} \int_0^T F_k\big(x(t)\big)\, dt. \tag{3}$$

The ergodic metric is the weighted distance between the two spectra,

$$\mathcal{E}\big(x(\cdot)\big) = \sum_k \Lambda_k \big(c_k - \phi_k\big)^2, \qquad \Lambda_k = \big(1 + \|k\|^2\big)^{-\frac{n+1}{2}}, \tag{4}$$

and it is this quantity, not a positional error, that the controller drives to zero. The weight Λ_k decays with frequency, so coarse, low-frequency features of ϕ (roughly, where the big blobs of probability sit) are matched before fine detail is. That is a useful bias for a search vehicle: it commits to the general area first and only refines within it once the coarse coverage error is already small, which mirrors what a human operator would do with the same sonar picture. Fig. 1 contrasts this against ordinary trajectory tracking, both as block diagrams and as trajectories overlaid on a density: the tracked path corrects toward a fixed setpoint, while the ergodic path spends its effort proportional to the density itself.

Because $\mathcal{E}$ only depends on time-averages, it never asks for a specific pose at a specific time, which is precisely why interrupting the mission early still leaves a sensible partial coverage behind, unlike a lawnmower path cut short partway through a row.

A. Control law

Following the spectral multiscale coverage law [3], we do not optimize $\mathcal{E}$ over the full horizon, which is expensive, but instead compute at each instant the direction that most rapidly decreases it given the current dynamics. Define the coverage gradient

$$S\big(x(t), t\big) = -\sum_k \Lambda_k \big(c_k(t) - \phi_k\big)\, \nabla_x F_k\big(x(t)\big), \tag{5}$$

which points toward the region the trajectory has under-visited relative to ϕ, weighted by how much that region matters spectrally. The controller then solves a small, local quadratic program

$$u^\star(t) = \arg\min_{u \in \mathcal{U}} \; S\big(x(t), t\big)^\top f\big(x(t), u\big) + \lambda \|u\|^2, \tag{6}$$

where f is the vehicle's dynamics and λ trades off aggressiveness against control effort. Solved at a few tens of hertz, this yields a smooth, receding-horizon feedback law that reacts immediately to changes in ϕ, since ϕ only enters through ϕ_k and the running average $c_k(t)$, both cheap to update online.

IV. Vehicle Model

The ergodic layer commands a desired planar velocity, which a lower-level controller converts into thruster commands over the full six degrees of freedom, matching the split shown in Fig. 2 between the ergodic controller and the low-level controller. The BlueROV2 Heavy is modeled with the standard marine-craft equations [5],

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau, \tag{7}$$

$$\dot{\eta} = J(\eta)\, \nu, \tag{8}$$

with η the pose in the inertial frame, ν the body-frame velocity, M the rigid-body plus added-mass inertia, C Coriolis and centripetal terms, D hydrodynamic damping, and τ the thruster wrench allocated by the low-level controller from the desired velocity handed down by the ergodic layer. The ergodic controller itself reasons over a reduced surge/sway/yaw state, which is what the coverage metric in Section III is defined over; depth and roll/pitch stabilization are handled independently by the low-level loop, which is the arrangement Stonefish's ROS 2 interface expects.

V. The SAGE Architecture

Fig. 2 is the architecture this note builds. A vision agent processes the camera or sonar stream and returns a set of detections, each with a class label, an estimated pose (from stereo geometry or from acoustic range and bearing), and a confidence. Separately, a short mission description, for instance "inspect the pipe joints and flag anomalies", is parsed into a semantic command: a set of weights over the classes and regions the operator currently cares about. The semantic grounding module combines the two into a live spatial distribution as a weighted Gaussian mixture,

$$\phi(x, t) = \frac{1}{Z} \sum_i w_i(t)\, \mathcal{N}\big(x;\, \mu_i,\, \Sigma_i\big), \tag{9}$$

where μ_i, Σ_i come from the detection's estimated pose and its positional uncertainty, w_i combines detection confidence with the mission weight for that class, and Z normalizes the mixture to a proper density on X. This is the distribution the ergodic controller in Section III tracks: not a fixed prior drawn by hand, but something the vehicle's own perception keeps producing. Fig. 3 shows the same pipeline end to end, from operator and sonar input through the semantic module to the resulting three-dimensional coverage surface, with the vehicle's trajectory threading between the density peaks it induces.

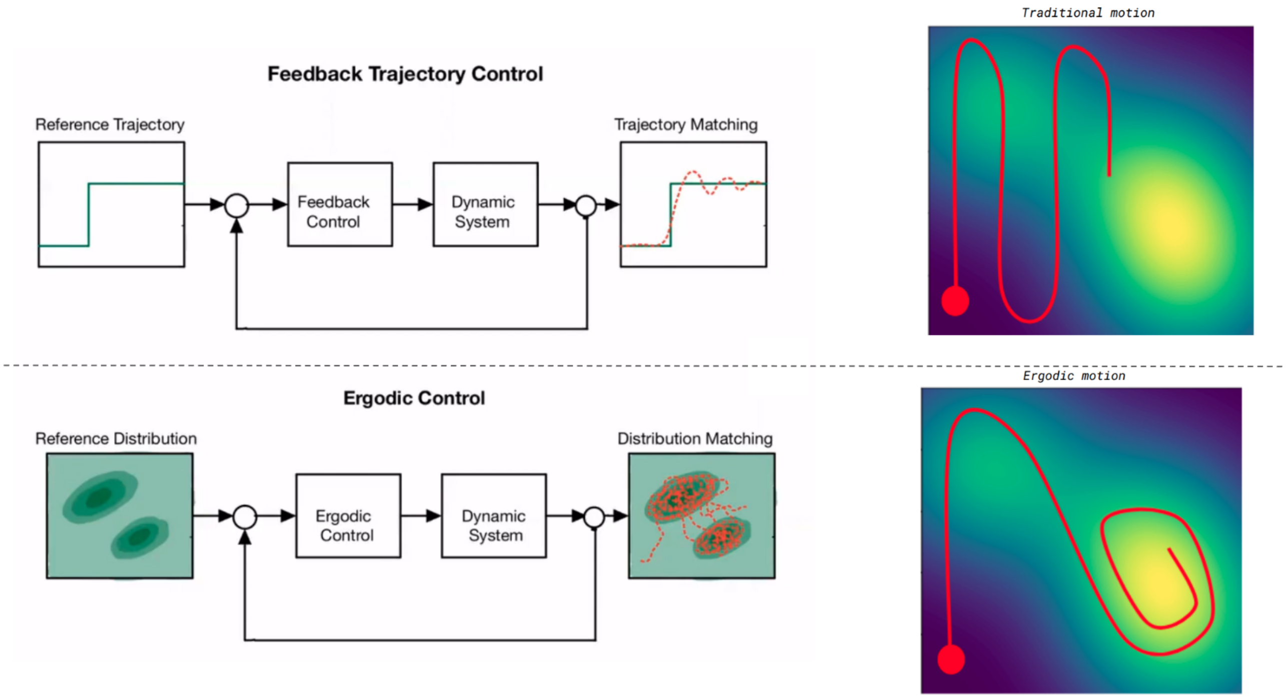

Fig. 1. Trajectory tracking asks the vehicle to converge on a fixed reference and penalizes deviation from it (top). Ergodic control asks the vehicle to match the statistics of a reference distribution, so the error signal is a coverage error rather than a positional one (bottom). On the right, a traditional sweep pattern spends time uniformly and only grazes the region of highest density, while the ergodic trajectory spirals inward and lingers exactly where the density is highest.

Because ϕ is recomputed rather than planned once, new detections change the vehicle's behaviour without any explicit re-planning step: they simply shift where the mixture puts its mass, and the coverage gradient in (5) responds on the next control cycle.

VI. Dynamic and Reversed Distributions

Two extensions fall out of the same objective with no new machinery. First, a target distribution does not have to be updated only by adding mass; it can decay, which is the natural way to represent a belief that gets stale once the vehicle has already looked somewhere, or that should be refreshed as new frames arrive:

$$\phi_t(x) = (1-\lambda)\,\phi_{t-1}(x) + \lambda\,\phi_{\text{obs},t}(x), \tag{10}$$

with $\lambda \in (0,1]$ a forgetting rate. The rightmost panel of Fig. 4 shows the effect: as one mode's mass fades and a second one strengthens, the trajectory migrates with it, with no discrete re-plan trigger anywhere in the loop.

Second, the same metric can be asked to minimize time spent in high-density regions instead of maximizing it, simply by tracking the complement of ϕ,

$$\phi_{\text{avoid}}(x) = \frac{1-\phi(x)}{\int_X \left(1-\phi(x')\right) dx'}. \tag{11}$$

This turns a hazard map, a diver exclusion zone, or a fragile habitat patch flagged by the same vision agent into an avoidance behaviour for free, without a separate potential-field or virtual-obstacle term bolted onto the controller. The center panel of Fig. 4 shows this reversed behaviour against the same bimodal density used for normal coverage on the left: the trajectory now threads the low-density corridor between the two modes rather than looping through them.

VII. Simulation Setup and Results

The architecture was implemented in ROS 2 and tested in Stonefish [4], an open-source marine robotics simulator with a full hydrodynamics model, sonar and camera rendering, and a native ROS 2 interface, running a BlueROV2 Heavy model. The scenario, shown in Fig. 5, places the vehicle above a pipe section with several candidate anomaly locations along its length; each location contributes a Gaussian mode to ϕ with weight set by simulated detection confidence. Rather than being handed an inspection order, the vehicle's ergodic controller distributes its trajectory across all four modes in proportion to their weight, spending visibly longer near the two higher-confidence detections while still returning to check the other two, which is exactly the qualitative behaviour a human operator would want and precisely what a fixed waypoint tour cannot reproduce without being told the ranking in advance. A supplementary animation of this run (sage_1.gif) and a recorded walkthrough of the full system (SAGE_web.mp4) accompany this note.

Across the static, reversed, and dynamic cases in Fig. 4 and the pipeline scenario in Fig. 5, the qualitative pattern held: the vehicle's time allocation tracked the density's shape rather

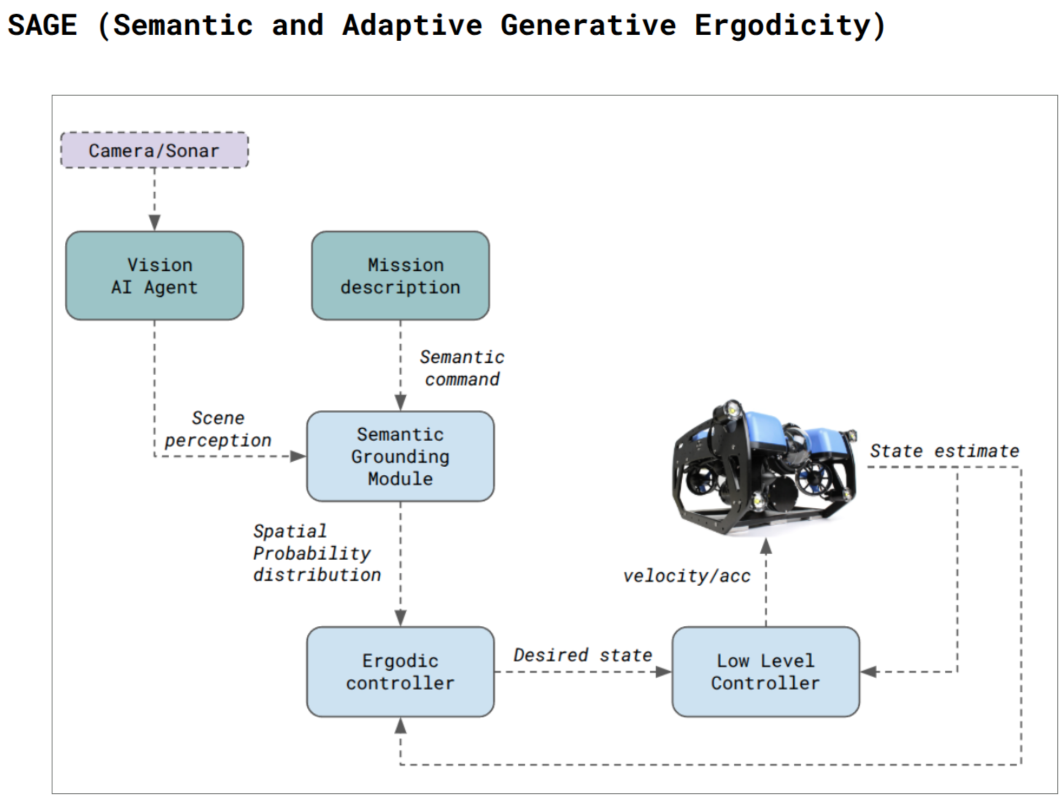

Fig. 2. The SAGE pipeline. A vision agent turns camera or sonar frames into scene perception, while a mission description supplies operator intent as a semantic command. The semantic grounding module fuses both into a spatial probability distribution, which the ergodic controller tracks by issuing a desired state to the low-level controller; the vehicle's state estimate closes the loop back into the ergodic layer.

than a scripted order, it adapted within a control cycle when the density moved, and inverting the sign of the objective was enough to turn coverage into avoidance.

VIII. Discussion: Why This Fits Underwater Autonomy

The case for ergodic coverage gets stronger, not weaker, once the sensing is bad, which is the normal condition underwater rather than the exception. Acoustic and optical returns are sparse and noisy, so committing to a discrete waypoint list built from a single noisy detection pass is fragile in a way that committing to a soft density is not: a wrong detection shifts a Gaussian's weight rather than sending the vehicle to the wrong place outright. Replanning a waypoint tour every time a new sonar ping revises the picture is also expensive and produces visibly jerky behaviour; here, a revised ϕ just changes the coverage gradient on the next cycle, so the vehicle's response is continuous rather than a discrete jump. Multi-modal, ambiguous scenes, several plausible defects on a pipeline, say, are handled directly by a mixture density rather than by solving a combinatorial ordering problem, and the vehicle's actual visitation frequency reflects the confidence in each mode automatically. The same statistical framing also means that current and disturbance rejection do not need to be perfect for the mission to still make sense: since the objective is a time-averaged match rather than a path to be exactly retraced, small deviations self-correct over the remaining horizon instead of accumulating as tracking error. And because the objective degrades gracefully, a vehicle that has to abort early, common enough with limited battery or a lost acoustic link, still leaves behind the best coverage it could have produced given the time it had, which is a meaningfully different guarantee from a partially completed lawnmower sweep.

IX. Limitations and Future Work

This work was carried out entirely in simulation, and the largest open question is how much of it survives contact with real acoustic noise, real currents, and a real vision pipeline running on degraded, murky footage. The semantic grounding module here weights detections heuristically; a proper treatment would fold detection uncertainty into ϕ through a Bayesian update rather than the exponential-forgetting heuristic used in (9), and would let a language model parse the mission description directly rather than through fixed class weights. Extending the coverage metric from the planar reduction used here to the full water column, and from a single vehicle to a small team sharing one distribution, are natural next steps, as is a real trial with Heriot-Watt's BlueROV2 platform to close the sim-to-real gap this note leaves open.

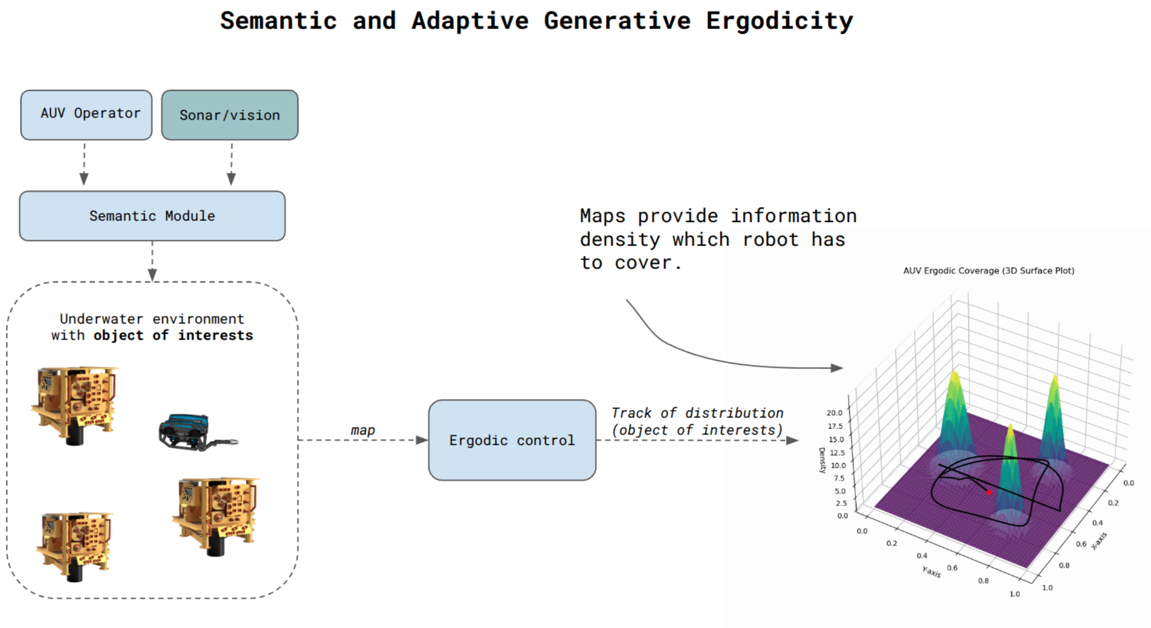

Fig. 3. End-to-end view: operator intent and sonar/vision perception feed the semantic module, which produces a map of information density over the workspace. That map is what the ergodic controller tracks, shown here as a trajectory weaving between three density peaks in a synthetic scene.

ACKNOWLEDGMENT

The ergodic controller used here builds on the ergodic-control-sandbox released by the Murphey Lab at Northwestern University [6], [7], and this work benefited from the underwater robotics group at Heriot-Watt University.

REFERENCES

[1] G. Mathew and I. Mezić, "Metrics for ergodicity and design of ergodic dynamics for multi-agent systems," *Physica D: Nonlinear Phenomena*, vol. 240, no. 4-5, pp. 432–442, 2011.

[2] L. M. Miller and T. D. Murphey, "Trajectory optimization for continuous ergodic exploration," in *American Control Conference (ACC)*, 2013, pp. 4196–4201.

[3] I. Abraham and T. D. Murphey, "Decentralized ergodic control: distribution-driven sensing and exploration for multiagent systems," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2987–2994, 2018.

[4] P. Cieslak, "Stonefish: An advanced open-source simulation tool designed for marine robotics with a ROS interface," in *OCEANS 2019 - Marseille*, 2019, pp. 1–6.

[5] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, Wiley, 2011.

[6] Murphey Lab, "ergodic-control-sandbox," GitHub repository, https://github.com/MurpheyLab/ergodic-control-sandbox.

[7] Murphey Lab, Northwestern University, https://murpheylab.github.io/.

[8] T. D. Murphey, Faculty profile, Northwestern University Robotics, https://robotics.northwestern.edu/people/profiles/faculty/murphey-todd.html.

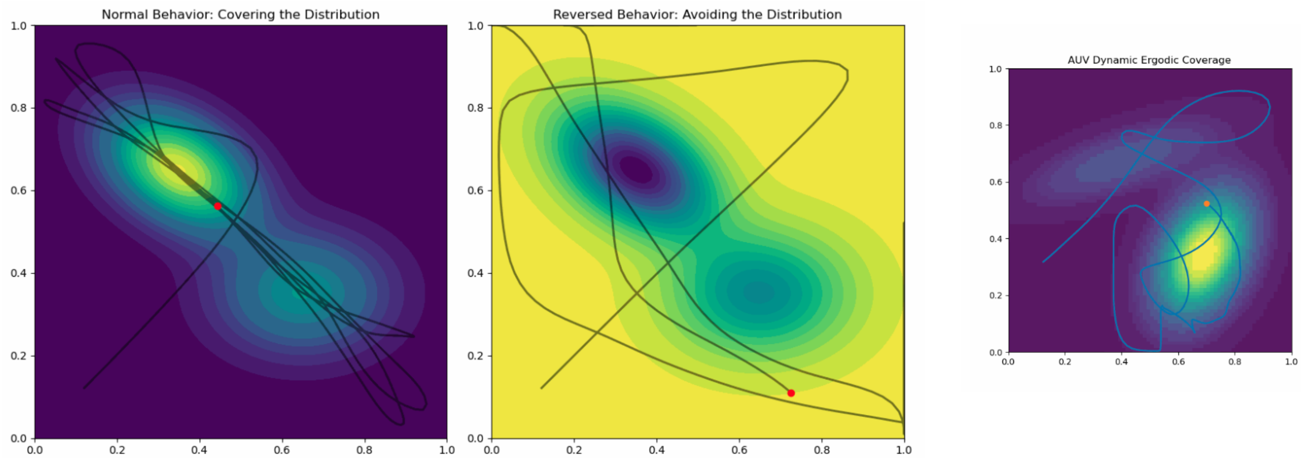

Fig. 4. Left: ergodic coverage of a static bimodal distribution, the vehicle threading a figure-eight between the two modes. Center: the same machinery with the distribution complemented, so the vehicle is repelled from the high-density regions instead of drawn to them, useful for marking hazard or no-go zones without a separate obstacle-avoidance layer. Right: coverage of a distribution that itself evolves, the trajectory shifting from one mode to a second as it strengthens.

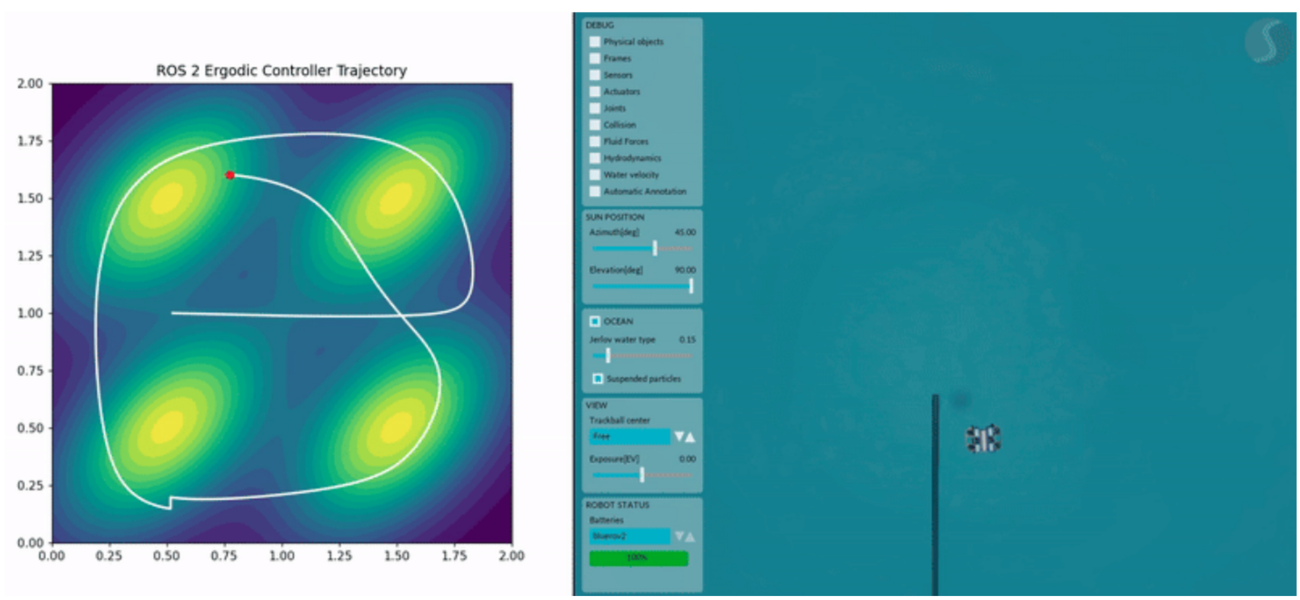

Fig. 5. Left: the ROS 2 ergodic controller's trajectory over a synthetic four-mode density representing candidate pipeline anomalies. Right: the corresponding BlueROV2 Heavy run in Stonefish, inspecting a pipe section under a physically simulated water column, current, and lighting model.