

Eclares: Energy-Aware Clarity-Driven Ergodic Search

Kaleb Ben Naveed¹, Devansh Agrawal², Christopher Vermillion³ and Dimitra Panagou^{1,2}

Abstract—Planning informative trajectories while considering the spatial distribution of the information over the environment, as well as constraints such as the robot’s limited battery capacity, makes the long-time horizon persistent coverage problem complex. Ergodic search methods consider the spatial distribution of environmental information while optimizing robot trajectories; however, current methods lack the ability to construct the target information spatial distribution for environments that vary stochastically across space and time. Moreover, current coverage methods dealing with battery capacity constraints either assume simple robot and battery models, or are computationally expensive. To address these problems, we propose a framework called **Eclares**, in which our contribution is two-fold. 1) First, we propose a method to construct the target information spatial distribution for ergodic trajectory optimization using clarity, an information measure bounded between $[0, 1]$. The clarity dynamics allows us to capture information decay due to lack of measurements and to quantify the maximum attainable information in stochastic spatiotemporal environments. 2) Second, instead of directly tracking the ergodic trajectory, we introduce the energy-aware (**eware**) filter, which iteratively validates the ergodic trajectory to ensure that the robot has enough energy to return to the charging station when needed. The proposed **eware** filter is applicable to nonlinear robot models and is computationally lightweight. We demonstrate the working of the framework through a simulation case study. [Code]^a

I. INTRODUCTION

Autonomous robots are widely used in tasks that involve data acquisition over long time horizons, such as search-and-rescue [1], [2], characterization of ocean currents such as the Gulf Stream to study cross-shelf exchanges [3], [4], water body exploration [5], [6] and wildfire monitoring [7]. Coverage planning for data acquisition poses diverse challenges, including considering the spatial distribution of environmental information while optimizing search trajectories [8]–[10]. This is important because some regions have higher density of information as compared to others. Other challenges involve persistently monitoring a *spatiotemporal environment* that varies across space and time [7], [9], [11], [12], and ensuring *task persistence* by enabling the robot to return to the charging station and recharge when/as needed during persistent coverage tasks [13]–[16]. In this paper, we consider these problems and propose a solution framework **Eclares**. Moreover, we also consider the case when a

^aThe authors would like to acknowledge the support of the National Science Foundation (NSF) under grant no. 2223845 and grant no. 1942907.

¹Department of Robotics, University of Michigan, Ann Arbor, MI, 48109 USA. {kbnaveed@umich.edu}

² Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, 48109 USA.

³ Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, 48109 USA.

^aCode: <https://github.com/kalebbennaveed/Eclares.git>

spatiotemporal environment has process uncertainty, i.e., we consider a class of *stochastic spatiotemporal environments*. Probabilistic methods can be used to compute estimates of the environmental variation. However, in general the stochastic variations in the environment can result in information decay without continuous monitoring.

Recent advances in ergodic search [8]–[10], [17], [18] consider the spatial distribution of the information over the environment while generating coverage trajectories. This is achieved by constructing a *target information spatial distribution* (TISD) over the environment and then adjusting the time spent in specific regions according to the TISD. However, these methods either assume *spatiostatic environment* [8], [10], which only varies through space, or spatiotemporal with known variation model [9], [19], [20]. Therefore, these methods can not incorporate potential information decay, which will result from the environment’s stochastic variation across space and time. To address this problem, we use clarity [21], an information measure bounded between $[0, 1]$, to construct the TISD. The proposed method to construct the TISD captures information decay due to a lack of measurements, and quantifies the maximum attainable information in stochastic spatiotemporal environments.

In order to ensure persistent monitoring of a stochastic spatiotemporal environment for a long time horizon, the robot’s limited battery capacity must be taken into consideration so that it can return to a charging station when needed. Most of the recent work on task persistence mainly either uses control barrier function (CBF)-based methods [13]–[15], [22] or Hamilton-Jacobi reachability based methods [16], [21], [23]. CBF-based methods are computationally efficient; however, they only assume simple robot and battery models. Alternatively, reachability-based methods can handle nonlinear systems and are robust with respect to external disturbances, but they are mostly used offline, suffer from curse of dimensionality, and are computationally expensive. The proposed online filter **eware** inspired by [24] addresses these problems. **eware** iteratively validates the ergodic trajectory based on the robot’s current battery levels, ensuring that the robot is aware of its remaining battery energy during the coverage task and returns to the charging station before the energy is depleted. The paper is organized as follows: **Section II** introduce preliminaries, **Section III** formulates the problem statement, **Section IV** present the proposed solution framework **Eclares**, and **Section V** provide results.

II. PRELIMINARIES

1) **Notation:** Let $\mathbb{N} = \{0, 1, 2, \dots\}$. Let $\mathbb{R}, \mathbb{R}_{\geq 0}, \mathbb{R}_{> 0}$ be the set of reals, non-negative reals, and positive reals re-

spectively. Let \mathcal{S}_{++}^n denote set of symmetric positive-definite in $\mathbb{R}^{n \times n}$. Let $\mathcal{N}(\mu, \Sigma)$ denote a normal distribution with mean μ and covariance $\Sigma \in \mathcal{S}_{++}^n$. The space of continuous functions $f : \mathcal{A} \rightarrow \mathcal{B}$ is denoted $C(\mathcal{A}, \mathcal{B})$. The norm of a vector $x \in \mathbb{R}^n$ is denoted $\|x\|_Q = \sqrt{x^T Q x}$.

2) Dynamics Model: Consider the continuous-time system dynamics comprising the robot and battery discharge dynamics:

$$\dot{\chi} = \begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = f(\chi, u) = \begin{bmatrix} f_r(x, u) \\ f_e(e, x, u) \end{bmatrix} \quad (1)$$

where $\chi = [x^T, e]^T \in \mathcal{Z} \subset \mathbb{R}^{n+1}$ is the system state consisting of the robot's state $x \in \mathcal{X} \subset \mathbb{R}^n$ and robot's State-of-Charge (SoC) $e \in \mathbb{R}_{\geq 0}$. $u \in \mathcal{U} \subset \mathbb{R}^m$ is the control input, $f : \mathcal{Z} \times \mathcal{U} \rightarrow \mathbb{R}^{n+1}$ defines the continuous-time system dynamics, $f_r : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}^n$ define robot dynamics and $f_e : \mathbb{R}_{\geq 0} \times \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$ define battery discharge dynamics.

3) Ergodic Search: Ergodic search [8], [9] is a technique to generate trajectories $x : [t_0, T] \rightarrow \mathcal{X}$ that cover a rectangular domain $\mathcal{P} = [0, L_1] \times \cdots [0, L_s] \subset \mathbb{R}^s$, matching a specified *target information spatial distribution* (TISD) $\phi : \mathcal{P} \rightarrow \mathbb{R}$, where $\phi(p)$ is the density at $p \in \mathcal{P}$. Moreover, the spatial distribution of the trajectory $x(t)$ is defined as

$$c(x(t), p) = \frac{1}{T - t_0} \int_{t_0}^T \delta(p - \Psi(x(\tau))) d\tau \quad (2)$$

where $\delta : \mathcal{P} \rightarrow \mathbb{R}$ is the Dirac delta function and $\Psi : \mathcal{X} \rightarrow \mathcal{P}$ is a mapping such that $\Psi(x(\tau))$ is the position of the robot at time $\tau \in [t_0, T]$. In other words, given a trajectory $x(t)$, $c(x(t), p)$ represents the fraction of time the robot spends at a point $p \in \mathcal{P}$ over the interval $[t_0, T]$. Then, the *ergodicity* of $x(t)$ w.r.t to a TISD ϕ is

$$\Phi(x(t), \phi) = \|c - \phi\|_{H^{-(s+1)/2}} \quad (3)$$

where $\|\cdot\|_{H^{-(s+1)/2}}$ is the Sobolev space norm defined in [8], i.e., Φ is a function space norm measuring the difference between the TISD ϕ and the spatial distribution of the trajectory c . In [9] a Projection-based Trajectory Optimization (PTO) method is proposed to generate ergodic trajectories. It optimizes (over the space of trajectories $x(t) \in C([t_0, T], \mathcal{X})$, $u(t) \in C([t_0, T], \mathcal{U})$)

$$\begin{aligned} \min_{x(t), u(t)} \quad & \left[\Phi(x(t), \phi) + J_b(x(t)) + \int_{t_0}^T \|u(\tau)\|^2 d\tau \right] \\ \text{s.t.} \quad & \dot{x} = f(x, u) \\ & x_0 = x(t_0) \end{aligned} \quad (4)$$

where J_b penalizes trajectories that leave the domain, and x_0 is the robot's state at time t_0 . The algorithm uses Fourier decomposition of c, ϕ to numerically evaluate Φ and its gradients with respect to $x(t)$ and $u(t)$. This enables an efficient gradient descent algorithm to optimize (4). In this paper, we do not modify the algorithm in [9]. Instead, we focus on developing a strategy to systematically construct the TISD ϕ given the environment model and the robot's

sensing model, as discussed later.

4) Clarity: Consider a stochastic variable (quantity of interest) $h \in \mathbb{R}$ governed by the process and output (measurement) models:

$$\dot{h} = w(t), \quad w(t) \sim \mathcal{N}(0, Q) \quad (5a)$$

$$y = C(x)h + v(t), \quad v(t) \sim \mathcal{N}(0, R) \quad (5b)$$

where $Q \in \mathbb{R}_{\geq 0}$ is the known variance associated with the process noise, $y \in \mathbb{R}$ is the measurement, $C : \mathcal{X} \rightarrow \mathbb{R}$ is the mapping between robot state and sensor state^b, and $R \in \mathbb{R}$ is the known variance of the measurement noise.

In [21], we introduced the notion of clarity q of the random quantity h , which lies between $[0, 1]$ and is defined such that $q = 0$ represents h being unknown, and $q = 1$ corresponds to h being completely known. The clarity dynamics for the subsystem (5a), (5b) are given by [21]

$$\dot{q} = \frac{C(x)^2}{R}(1 - q)^2 - Qq^2 \quad (6)$$

If $C(x) = C$ is constant, (6) admits a closed-form solution for the initial condition $q(0) = q_0$:

$$q(t; q_0) = q_\infty \left(1 + \frac{2\gamma_1}{\gamma_2 + \gamma_3 e^{2kQt}} \right) \quad (7)$$

where $q_\infty = k/(k+1)$, $k = C/\sqrt{QR}$, $\gamma_1 = q_\infty - q_0$, $\gamma_2 = \gamma_1(k-1)$, $\gamma_3 = (k-1)q_0 - k$.

Notice that as $t \rightarrow \infty$, $q(t; q_0) \rightarrow q_\infty \leq 1$ monotonically. Thus q_∞ defines the maximum attainable clarity. Equation (7) can be inverted to determine the time required to increase the clarity from q_0 to some q_1 . This time is denoted $\Delta T : [0, 1]^2 \rightarrow \mathbb{R}_{\geq 0}$:

$$\Delta T(q_0, q_1) = t \text{ s.t. } q(t; q_0) = q_1 \quad \text{for } q_1 \in [q_0, q_\infty] \quad (8)$$

For $q_1 < q_0$, we set $\Delta T(q_0, q_1) = 0$ while $\Delta T(q_0, q_1)$ is undefined for $q_1 \geq q_\infty$.

III. PROBLEM FORMULATION

A. Environment Specification

Consider the coverage space \mathcal{P} . We discretized the domain into a set of N_c cells each with size V .^c Let $m_c : [t_0, \infty) \rightarrow R$ be the (time-varying) quantity of interest at each cell $c \in \mathcal{C} = \{1, \dots, N_c\}$. We model the quantities of interest as independent stochastic processes:

$$\dot{m}_c = w_c(t), \quad w_c(t) \sim \mathcal{N}(0, Q_c) \quad (9a)$$

$$y_c = C_c(x)m_c + v_c(t), \quad v_c(t) \sim \mathcal{N}(0, R) \quad (9b)$$

where $y_c \in \mathbb{R}$ is the output corresponding to cell c . R is the measurement noise, and $Q_c \in \mathbb{R}_{>0}$ is the process noise variance at each cell c . Since m_c varies spatially and temporally under process noise Q_c for each cell $c \in \mathcal{C}$, the environment becomes a *stochastic spatiotemporal environment*.

^bC is set to 1 when sensed by the robot and 0 otherwise.

^cSize is width in 1D, area in 2D, and volume in 3D.

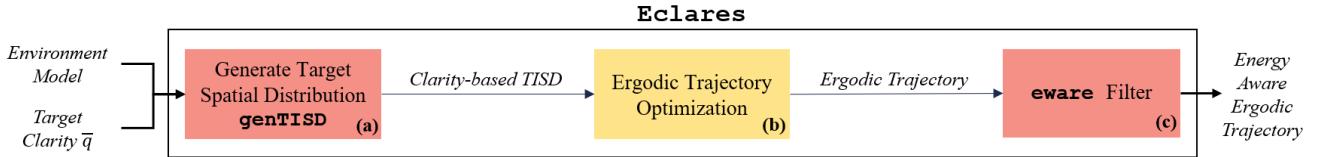


Fig. 1. The high-level overview of the proposed framework. Blocks (a) and (c) are the proposed modules of the framework, while block (b) is borrowed from [9].

B. Problem Statement

Consider a robotic system performing persistent coverage of a stochastic spatiotemporal environment (9), i.e., over a time interval $[0, \infty)$. Assume the desired quality of information at each cell has been specified. This is encoded using a *target clarity* $\bar{q}_c < q_{\infty,c}$ for each cell $c \in \mathcal{C}$. The target clarity can be different at each cell, indicating a different desired quality of information at each cell, but must be less than $q_{\infty,c}$, the maximum attainable clarity of the cell. If $\bar{q}_c \geq q_{\infty,c}$ for a cell $c \in \mathcal{C}$, then the robot would try to spend infinite amount of time at a cell c , which is undesirable.

We use clarity as our information metric since it is particularly effective for stochastic spatiotemporal environments:

- The clarity decay rate in cell c , i.e. $-Q_c q_c^2$, is explicitly dependent on the stochasticity of the environment Q_c in (9). This allows the information decay rate to be determined from the environment model, and not set heuristically [21]. Furthermore, spatiostatic environments are a special case: by setting $Q_c = 0$, clarity cannot decay.
- While taking measurements of cell c , clarity q_c monotonically approaches $q_{\infty,c} < 1$ for $Q_c, R > 0$. This indicates that maximum attainable information is upper bounded.

In this persistent task, the robot replans a trajectory every $T_H \in \mathbb{R}_{>0}$ seconds, i.e., at times $\{t_0, t_1, \dots\}$ for $t_k = kT_H$, $k \in \mathbb{N}$. At the k -th iteration, the objective is to minimize the *mean clarity deficit* $q_d(t_k + T_H)$, which is defined as

$$q_d(t_k + T_H) = \frac{1}{N_c} \sum_{c=1}^{N_c} \max(0, \bar{q}_c - q_c(t_k + T_H)) \quad (10)$$

where $q_c(t_k + T_H)$ is the clarity at time $t_k + T_H$ of cell $c \in \mathcal{C}$. However, in order to persistently monitor a stochastic spatiotemporal environment over a long time horizon, the robot's energy constraints must be taken into consideration. Thus, the overall problem can be posed as follows:

$$\min_{x(t), u(t)} q_d(t_k + T_H) \quad (11a)$$

$$\text{s.t. } \chi(t_k) = \chi_k \quad (11b)$$

$$\dot{\chi} = f(\chi, u) \quad (11c)$$

$$\dot{q}_c = g(x, q_c) \quad \forall c \in \mathcal{C} \quad (11d)$$

$$e(t) \geq e_{\min} \quad (11e)$$

where $q_d(t_k + T_H)$ is the mean clarity deficit at the end of system trajectory $\chi(t; t_k, \chi_k)$, $\forall t \in [t_k, t_k + T_H]$ given by (10), $g : \mathcal{X} \times [0, 1] \rightarrow \mathbb{R}_{\geq 0}$ define the clarity dynamics (6), and e_{\min} is the minimum energy level allowed for the robot.

IV. ENERGY AWARE CLARITY DRIVEN ERGODIC SEARCH (**ECLARES**)

In this section, we introduce **Eclares** as a solution to the optimization problem (11). We first provide the solution overview and discuss its motivation. Next, we provide details on the method to construct the TISD and lastly, we introduce the **eaware** filter.

A. Method Motivation & Overview

In order to solve problem (11), we take inspiration from ergodic search [8], [9]. As discussed in Section II-3, ergodic search can generate trajectories by solving problem (4). If the construction of TISD ϕ is based on the current clarity $q_c(t)$ and the target clarity \bar{q}_c at each cell, then ergodic search can be used to minimize the mean clarity deficit (10). Therefore, in this work, we propose a method to construct ϕ using clarity. However, the ergodic trajectory optimization in (4) does not consider the energy constraint (11e). One approach would be to add the energy constraint in (4), but the non-convexity of (4) implies that guaranteeing convergence and feasibility is challenging. Therefore, we propose the framework **Eclares**, shown in Figure 1, as an approximate solution to the problem (11).^d

Our solution involves decoupling (11) into two subproblems: (A) we design a trajectory that maximizes information collection without the energy constraint (referred to as the *ergodic trajectory*), and (B) we construct a trajectory that tracks part of the ergodic trajectory but also reaches the charging station before the robot's energy depletes (referred to as the *committed trajectory*). The low level controller of the robot always tracks the last committed trajectory. In this way, we ensure that the robot persistently explores the domain without violating the energy constraint.

We run these steps at different rates. The ergodic trajectory is replanned every T_H seconds, while the committed trajectory is updated every $T_E < T_H$ seconds.^e That is,

- at each time $t_k = kT_H$, $k \in \mathbb{N}$,
 - recompute the TISD ϕ using **genTISD**.
 - recompute an ergodic trajectory using PTO [9].
- at each time $\tau_j = jT_E$, $j \in \mathbb{N}$,
 - update the committed trajectory using **eaware**.

In this work, we adopt PTO directly from [9]. Before describing **genTISD** and **eaware**, we establish notation

^dIt is also important to note $q_d(T)$ is not differentiable so computing gradients for optimization problem (11) can be challenging. However, the problem (4) is differentiable and can be approximately solved using PTO [9].

^e $T_E, T_H \in \mathbb{R}_{>0}$ are user-defined parameters.

Algorithm 1 The **genTISD** algorithm

```

1: function genTISD ( $q_c, \bar{q}_c$ , environment model (9))
2:   for  $c \in \{1, \dots, N_c\}$  do
3:      $k \leftarrow 1/\sqrt{Q_c R}$ 
4:      $q_\infty \leftarrow k/(k+1)$ 
5:      $\bar{q} \leftarrow \min(\bar{q}_c, q_{\infty,c} - \epsilon)$ 
6:      $\phi_c \leftarrow \Delta T(\bar{q}, q_c)$ 
7:   end for
8:    $\phi_c \leftarrow \phi_c / (\sum_{c=1}^{N_c} \phi_c), \quad \forall c \in \{1, \dots, N_c\}$ 
9:   return  $\phi_c \forall c \in \{1, \dots, N_c\}$ 
10: end function

```

Algorithm 2 The **eaware** algorithm

```

1: function eaware( $\tau_j, x_k^{ergo}, x_{j-1}^{com}$ )
2:   Solve back-to-base trajectory (12a)
3:   Solve initial value problem (13a)
4:    $\tau_f \leftarrow \tau_j + T_N + T_B$ 
5:   if  $e(t) \geq 0 \forall t \in [\tau_j, \tau_f]$  and  $x(\tau_f) = x_c$  then
6:      $x_j^{com} \leftarrow x_j^{can}$ 
7:   else
8:      $x_j^{com} \leftarrow x_{j-1}^{com}$ 
9:   end if
10:  return  $x_j^{com}$ 
11: end function

```

for trajectories. Let $x([t_k, t_k + T_H]; t_k, x_k)$ be the ergodic trajectory generated at time t_k starting at state x_k and defined over a time horizon T_H . For compactness, we refer to $x([t_k, t_k + T_H]; t_k, x_k)$ as x_k^{ergo} . We use similar notation to describe other trajectories later on.

B. Generate Target Spatial Distribution (**genTISD**)

The **genTISD** algorithm is described in [Algorithm 1](#). Let ϕ_c denote the target information density evaluated for cell c . At the k -th iteration (i.e., at time $t_k = kT_H$), we set ϕ_c to be the time that the robot would need to increase the clarity from $q_c(t_k)$ to the target \bar{q}_c by observing cell c (Lines 3-6). This is determined using (8). The small positive constant $\epsilon > 0$ in Line 5 ensures that target clarity is always less than the maximum attainable clarity, i.e., $\bar{q}_c < q_{\infty,c}$. Finally, we normalize ϕ_c such that the sum of $\sum_{c \in \mathcal{C}} \phi_c = 1$ (Line 8). Once ϕ is constructed, PTO method [9] is used to generate the ergodic trajectory.

C. Energy-Aware (**eaware**)

Inspired by [24], **eaware** is a lightweight filter to ensure that the robot reaches the charging station before running out of energy. To do so, it constructs a *candidate* trajectory that follows the ergodic trajectory for a short time horizon T_N before attempting to return to the charging station. If the candidate trajectory is valid (defined below), it becomes a *committed* trajectory. The low-level tracking controller always tracks the last committed trajectory. [Algorithm 2](#) describes the **eaware** algorithm.

The j -th iteration of **eaware** starts at time $\tau_j = jT_E$. Let the robot state at time τ_j be $x_j \in \mathcal{X}$ and the system system

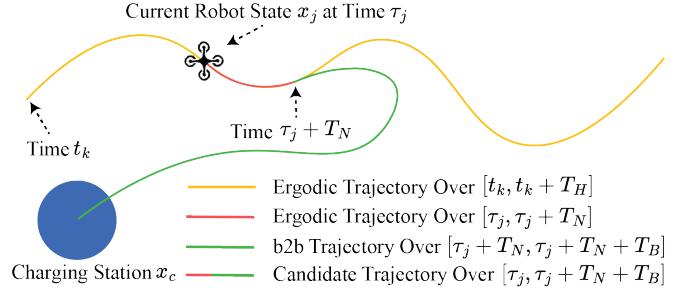


Fig. 2. This figure illustrates the generation of candidate trajectory at time τ_j , combining a small portion of the ergodic trajectory (shown in red) and the b2b trajectory (shown in green).

at time τ_j be $\chi_j \in \mathcal{Z}$. Suppose the last ergodic trajectory generated is x_k^{ergo} (defined over the interval $[t_k, t_k + T_H]$).

(Line 2) We construct a back-to-base (b2b) trajectory x_j^{b2b} , defined over $[\tau_j + T_N, \tau_j + T_N + T_B]$, by solving the optimal control problem:

$$\min_{x(t), u(t)} \int_{\tau_j + T_N}^{\tau_j + T_N + T_B} \|x(\tau) - x_c\|_{\mathbf{Q}}^2 + \|u(\tau)\|_{\mathbf{R}}^2 d\tau \quad (12a)$$

$$\text{s.t. } x(\tau_j + T_N) = x_j^{ergo}(\tau_j + T_N), \quad (12b)$$

$$\dot{x} = f_r(x, u) \quad (12c)$$

where $x_c \in \mathcal{X}$ is the state of the charging station, $\mathbf{Q} \in \mathbb{S}_{++}^n$ weights state cost, and $\mathbf{R} \in \mathbb{S}_{++}^m$ weights control cost.

(Line 3): Once b2b trajectory x_j^{b2b} is generated, we numerically construct the candidate trajectory by solving the initial value problem

$$\dot{x} = f(x, u(t)), \quad (13a)$$

$$x(\tau_j) = \chi_j \quad (13b)$$

$$u(t) = \begin{cases} \pi(\chi, x_k^{ergo}(t)), & t \in [\tau_j, \tau_j + T_N] \\ \pi(\chi, x_j^{b2b}(t)), & t \in [\tau_j + T_N, \tau_j + T_N + T_B] \end{cases} \quad (13c)$$

where $\pi : \mathcal{Z} \times \mathcal{X} \rightarrow \mathcal{U}$ is a control policy to track the portion of the ergodic trajectory and the b2b trajectory. [Figure 2](#) illustrates the generation of a candidate trajectory.

(Lines 5-9): Once the candidate trajectory is constructed, we check whether the robot can reach the charging station before running out of energy. If it can, the candidate replaces the committed trajectory.

V. SIMULATION CASE STUDY

A. Simulation Setup

In this section, we evaluate **Eclares** using a simulation case study. We synthesize a hypothetical scenario where a quadrotor modeled with nonlinear dynamics described in [25, Eq. (10)] is persistently monitoring a stochastic spatiotemporal environment modeled according to the dynamics described in (9). The coverage domain is 20×20 meter in area with grid cells of dimension 0.20×0.20. We use the PTO method [9] with double integrator dynamics to solve the optimization problem (4) for the time horizon $T_H =$

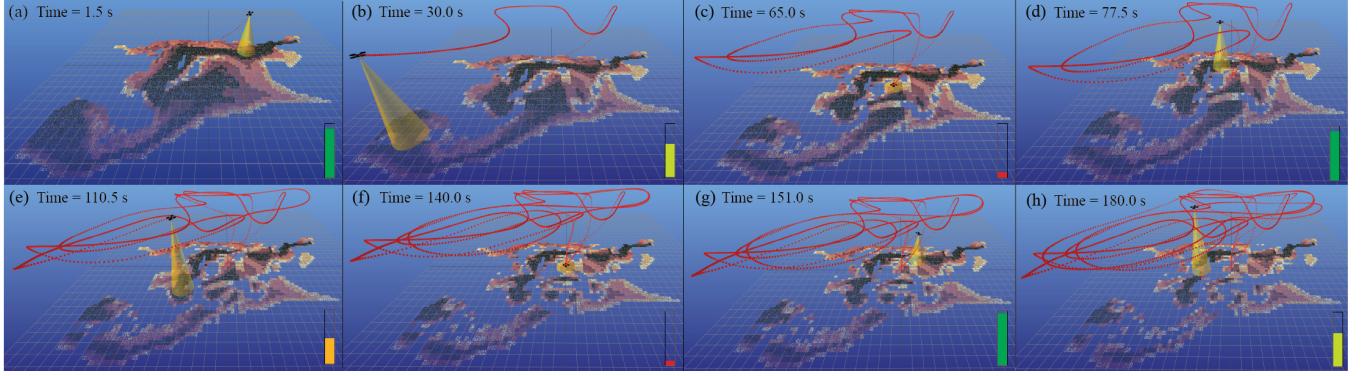


Fig. 3. The still frames from quadrotor simulation are shown where the quadrotor with a downward-facing camera and a gimbal is persistently monitoring a stochastic spatiotemporal environment. The TISD is shown as a point cloud in each frame and the colored bar in each frame shows the battery level. Frame (a) shows the quadrotor starting the coverage mission with full energy. Moving across frames (b) and (c), the quadrotor energy depletes and returns to the charging station (modeled as the origin of the world frame). After battery swapping, the quadrotor resumes the coverage mission as shown in the frame (d). The same trend follows in the subsequent frames. Moreover, it can be seen that as the quadrotor explores the environment, the TISD becomes more sparse, which reflects increase in clarity in the region. For full simulation video, please see <https://youtu.be/i2yD3A2b5nA>

30.0s. To generate b2b trajectory, we solve the problem (12) using model predictive control (MPC) with reduced linear quadrotor dynamics proposed by [25]. Another MPC problem with the same reduced linear quadrotor dynamics was posed to track the ergodic trajectory for shorter time horizon $T_N = 2.0$ s. Then the committed trajectory is generated using the **eware** algorithm. The committed trajectories are generated in the receding-horizon manner at 0.50 Hz (i.e., $T_E = 2.0$ s) and tracked at 20.0 Hz with zero-order hold.^f If one of the committed trajectory returns quadrotor back to the charging station, it stays there to simulate recharging. Note that no assumptions are made about the charging method; it may either involve stationary charging or battery swapping. We use the battery dynamics described in [14, Eq. (6)] as it directly incorporates robot control input as opposed to the worst-case approximation used in other works [13], [15]. Throughout the simulation, we use the Runge-Kutta 4th order (RK4) integration scheme. Figure 3 shows still frames from the light UAV simulator where a quadrotor with downward-facing camera is exploring a stochastic spatiotemporal environment.

B. Results and Discussion

1) Performance Comparison between Spatiostatic and Stochastic Spatiotemporal Environment: Figure 5(a) and Figure 5(b) show the evolution of mean clarity deficit q_d for the spatiostatic and stochastic spatiotemporal environments respectively. For the proposed method, in the case of the spatiostatic environment, the q_d approached almost zero at time $T = 780$ s; however, for the stochastic spatiotemporal environment, the q_d reduced to a non-zero value. This is expected due to the inherent stochasticity present in the environment as the information is decaying constantly due to the term $-Qq^2$ in (6), and it is not possible to achieve zero

^fFor more details on algorithm implementation, trajectory generation, and on hypothetical stochastic environment, please see <https://github.com/kalebbennaveed/Eclares.git>.

clarity deficit. However, the proposed method does provide an ability to set realistic expectations on the maximum possible information that can be collected in a stochastic environment by computing $q_{\infty,c}$, $\forall c \in C$.

2) Comparison to Baseline methods: We compare the performance of the proposed methodology **genTISD** to the uniform TISD method used in most of the ergodic literature [8], [10]. The uniform TISD assigns equal importance to all cells $c \in C$ in the domain. As evident from Figure 5(a) and Figure 5(b), the proposed method shows a faster convergence rate as compared to the uniform method. This is explained by the robot trajectory shown in Figure 4 as the robot spends more time in the region with higher clarity deficit as compared to the uniform method. It is important to note that the past ergodic literature does not consider the stochastic nature of the environment in their construction of target distribution. We also evaluate the performance of the proposed coverage method against the classic lawnmower coverage method. In the lawnmower coverage method, the robot follows the prescribed path which covers the whole domain. The lawnmower coverage pattern results in a repetitive paths as the robot covers same areas multiple times while leaving gaps in some areas. This causes clarity deficit to stabilize to a non-zero value as shown in Figure 5(a). In case of using lawnmower coverage method for stochastic spatiotemporal environment, the clarity deficit eventually starts increasing as the information constantly decays due to the term $-Qq^2$ in (6).

3) Evaluation of **eware Filter:** Figure 5(c) compares the quadrotor flight with and without the **eware** filter. Evidently, with the **eware** filter, the quadrotor safely returns to the charging station. However, in the absence of **eware** filter, quadrotor unaware of its remaining battery energy keeps on following the nominal ergodic trajectory, leading to a crash. Moreover, we observed that although the minimum SoC specified was 0%, the quadrotor returns to the charging station with almost 3 ± 1 % remaining SoC. This number can

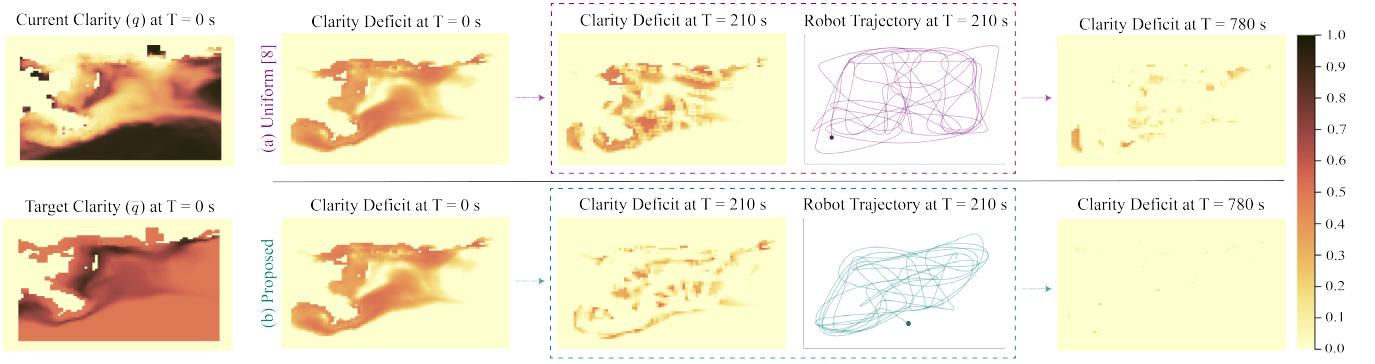


Fig. 4. The clarity deficit Map for the **spatiostatic environment** at time $T = 0$ s, $T = 210$ s, and $T = 780$ s is shown comparing two methods, uniform distribution, and proposed method, with the same clarity q and target clarity \bar{q} at $T = 0$ s.

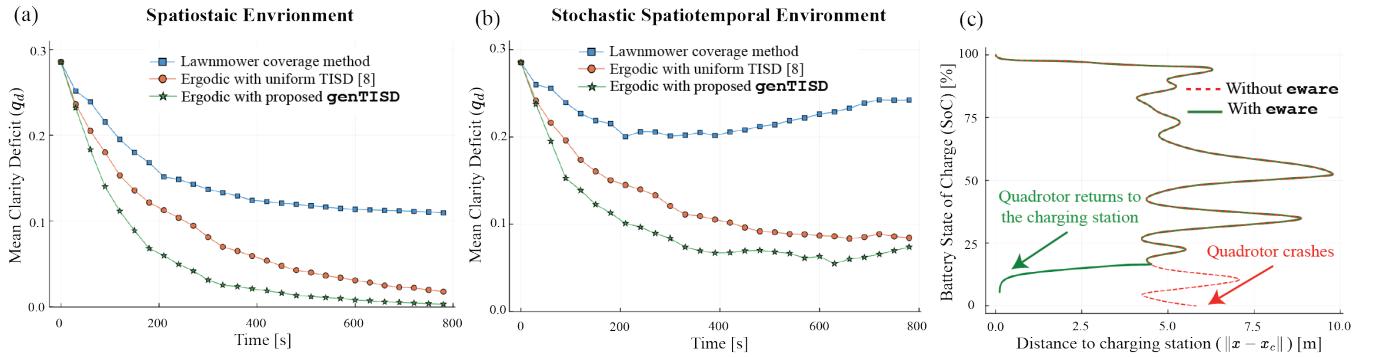


Fig. 5. (a) Plot of the mean clarity deficit (q_d) against time is shown for the **spatiostatic environment** for three methods. (b) The plot of the q_d against time is shown for the **stochastic spatiotemporal environment** for three methods. It can be seen from both plots that the proposed method outperforms other methods: Ergodic search with uniform target spatial distribution and classic lawnmower coverage method. Moreover, it is important to note that in the stochastic spatiotemporal environment, it is not possible to acquire zero clarity error as information is always decaying due to term $-Qq^2$ in (6). (c) Plot of the quadrotor's distance to charging station ($\|x - x_c\|$) against battery's State of Charge (SoC) is shown for one quadrotor flight, where x_c is the position of the charging station. In case of **eaware**, quadrotor safely returns to the charging station as energy depletes with almost 3 % SoC. However, without **eaware**, quadrotor unaware of its remaining energy keep on following the nominal path, leading to a crash.

be brought down by increasing the frequency of the candidate trajectory generation from 0.5 Hz to a larger number; however, this comes at the expense of overall computational time. We also compared the compute time of proposed method **eaware** with that of the CBF-based methods used for robotic task persistence since both are online methods. The **eaware** filter, updated at 0.5 Hz, took average time of 30.2 ms to numerically forward integrate the candidate trajectory through quadrotor and battery's nonlinear dynamics. Alternatively, the CBF-based method used in [13], updated at 20 Hz, took average time of 0.0551 ms to solve the optimization problem described in [13, Eq. (2)] with single integrator dynamics. While considering the update frequency, the CBF-based method [13] is almost 13 times faster than the proposed method; however, it assumes simple robot and battery models. Conversely, the proposed **eaware** filter achieved real-time performance with nonlinear quadrotor and battery dynamics. This allows for more flexibility with robot and battery models and thus increases generalizability of the proposed method.

VI. CONCLUSION

In this work, we introduced a framework **Eclares**, in which we proposed a method to update target information spatial distribution for the ergodic trajectory optimization, which considers the stochastic spatiotemporal environments. We also integrated a lightweight filter called **eaware** between the high-level ergodic planner and low-level tracking controller, which iteratively ensures that the robot has enough battery energy to continue the coverage task. Through simulation case study, we demonstrated that the proposed framework incorporates potential information decay and quantifies maximum attainable information in a stochastic environment while ensuring robot's safe return to the charging station before the battery energy depletes. However, We note that the decoupling used in **Eclares** can lead to a suboptimal solution, since the b2b trajectory is not optimized for information collection, which we hope to address in future work. Other future directions include considering environmental disturbances and extension of the proposed method to multi-agent exploration scenarios.

VII. APPENDIX

A. System Dynamics Model

1) *3D Quadrotor Dynamics*: We modeled the quadrotor using the dynamics described in [25, equation (10)]. The dynamics have the form:

$$\dot{x} = \begin{bmatrix} \dot{r} \\ \dot{q} \\ \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ (\frac{1}{2})L(q)H\omega \\ (\frac{1}{m})^W F(x, u) \\ J^{-1}(B\tau(x, u) - \hat{\omega}J\omega) \end{bmatrix} \quad (14)$$

where $x \in \mathbb{R}^{13}$ and $u \in \mathbb{R}^4$ are the state and control vectors, $r \in \mathbb{R}^3$ is the position vector, $q \in \mathbb{H}$ is attitude where \mathbb{H} is space of quaternions, $v \in \mathbb{R}^3$ is the linear velocity, $\omega \in \mathbb{R}^3$ is the angular velocity, m is the mass, $J \in \mathbb{R}^{3 \times 3}$ is the inertia matrix, ${}^W F(x, u) \in \mathbb{R}^3$ are the forces in the world frame, $B\tau(x, u)$ are the moments in the body frame, $H \in \mathbb{R}^{4 \times 3}$ is the matrix used to convert the 3-dim vector into 4-dim quaternion with zero scalar part,

$$H = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad (15)$$

$L(q) \in \mathbb{R}^{4 \times 4}$ is the matrix used for left multiplication of quaternion,

$$L(q) = \begin{bmatrix} q_s & -q_v^T \\ q_v & q_s I + \hat{q}_v \end{bmatrix} \quad (16)$$

where $q_s \in \mathbb{R}$ is the scalar part of the quaternion, $q_v \in \mathbb{R}^3$ is the vector part of the quaternion, and \hat{x} is the skew-symmetric matrix operator

$$\hat{x} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \quad (17)$$

2) *Battery Dynamics*: We modeled the battery dynamics using the dynamics described in [14, equation (6)]. The dynamics have the form:

$$\dot{e} = -\eta C^{-1} \alpha(\|u\|^2) \quad (18)$$

where C is battery's rated capacity, η is its Columbic efficiency, α is a class \mathcal{K} function, and $\|u\|$ is the magnitude of the control input.

B. Ergodic Trajectory Generation

We use the Projection-based Trajectory Optimization (PTO) proposed by [9] for ergodic trajectory optimization. For implementation sake, discrete approximation of system dynamics (1) is used. The optimization problem to generate the T_H horizon discrete trajectory $\tilde{x}_{1:N_h}$ is defined as

$$\begin{aligned} \min_{\tilde{x}_{1:N_h}, \tilde{u}_{1:N_h-1}} \quad & \{\Phi(\tilde{x}_{1:N_h}, \phi) + J_b(\tilde{x}_{1:N_h}) + \frac{\Delta t}{2} \sum_{k=1}^{N_h-1} \tilde{u}_k^T \mathbf{R} \tilde{u}_k\} \\ \text{s.t.} \quad & \tilde{x}_1 = \tilde{x}_{ic}, \\ & \tilde{x}_{k+1} = \tilde{x}_k + f(\tilde{x}_k, \tilde{u}_k) \Delta t \end{aligned} \quad (19)$$

where $\mathbf{R} \in \mathbb{S}_{++}^m$ weights control cost, \tilde{x}_{ic} is the initial robot state, and $J_b(\tilde{x})$ is the boundary penalty to keep the robot within the boundary of the rectangular coverage space \mathcal{P} . The $J_b(\tilde{x})$ is given as

$$J_b(\tilde{x}_{1:N_h}) = c_b \sum_{k=1}^{N_h} \sum_{i=1}^s (\max(\tilde{x}_{k,i} - L_i, 0)^2 + \min(\tilde{x}_{k,i}, 0)^2) \quad (20)$$

where $c_b > 0$ weights the boundary penalty, L_i is the i^{th} length of the rectangular domain, and $\tilde{x}_{k,i}$ is the i^{th} element of state \tilde{x}_k . We used the double integrator dynamics while solving the problem (19). Comprehensive details on the PTO method can be found in [9]. We iteratively generate ergodic trajectories of time horizon $T_H = 30$ sec with $dt = 0.2$ at a frequency of $\frac{1}{30}$ Hz. If the quadrotor returns to the charging station while following the nominal trajectories due to energy depletion, the new ergodic trajectory is generated based on the current clarity level after recharging.

C. Candidate Trajectory Generation

The candidate trajectories are constructed by using the method described in Section IV-C. The candidate trajectories are generated at 0.5 Hz and tracked with zero-order hold at 20.0 Hz. Essentially, the candidate trajectories are generated by stitching the shorter T_N horizon ergodic trajectory with the T_B horizon b2b trajectory. Here, we describe the method to generate the T_N horizon ergodic trajectory where $T_N < T_H$ and the T_B horizon b2b trajectory.

1) *T_N horizon Ergodic Trajectory*: The shorter horizon ergodic state trajectory $\tilde{x}_{1:N_n}$ and the control trajectory $\tilde{u}_{1:N_n-1}$ are generated using the convex Model Predictive Controller (MPC) formulation given as

$$\begin{aligned} \min_{\tilde{x}_{1:N_n}, \tilde{u}_{1:N_n-1}} \quad & \sum_{k=1}^{N_n-1} \left[(\tilde{x}_k - \tilde{x}_k^{ref})^T \mathbf{Q} (\tilde{x}_k - \tilde{x}_k^{ref}) \right. \\ & \left. + (\tilde{u}_k - \tilde{u}_k^{ref})^T \mathbf{R} (\tilde{u}_k - \tilde{u}_k^{ref}) \right] \\ \text{s.t.} \quad & \tilde{x}_1 = \tilde{x}_{ic}, \\ & \tilde{x}_{k+1} = f(\bar{x}_k, \bar{u}_k) + \left[\frac{\partial f}{\partial x} \Big|_{\bar{x}_k, \bar{u}_k} \right] (\tilde{x}_k - \bar{x}_k) \\ & + \left[\frac{\partial f}{\partial u} \Big|_{\bar{x}_k, \bar{u}_k} \right] (\tilde{u}_k - \bar{u}_k) \end{aligned} \quad (21)$$

where (\bar{x}_k, \bar{u}_k) is the k^{th} equilibrium point, $\mathbf{Q} \in \mathbb{S}_{++}^m$ weights state cost, $\mathbf{R} \in \mathbb{S}_{++}^m$ weights control effort, \tilde{x}_k^{ref} is the k^{th} reference state from the reference T_H horizon ergodic state trajectory $\tilde{x}_{1:N_n}^{ref}$, \tilde{u}_k^{ref} is the k^{th} reference

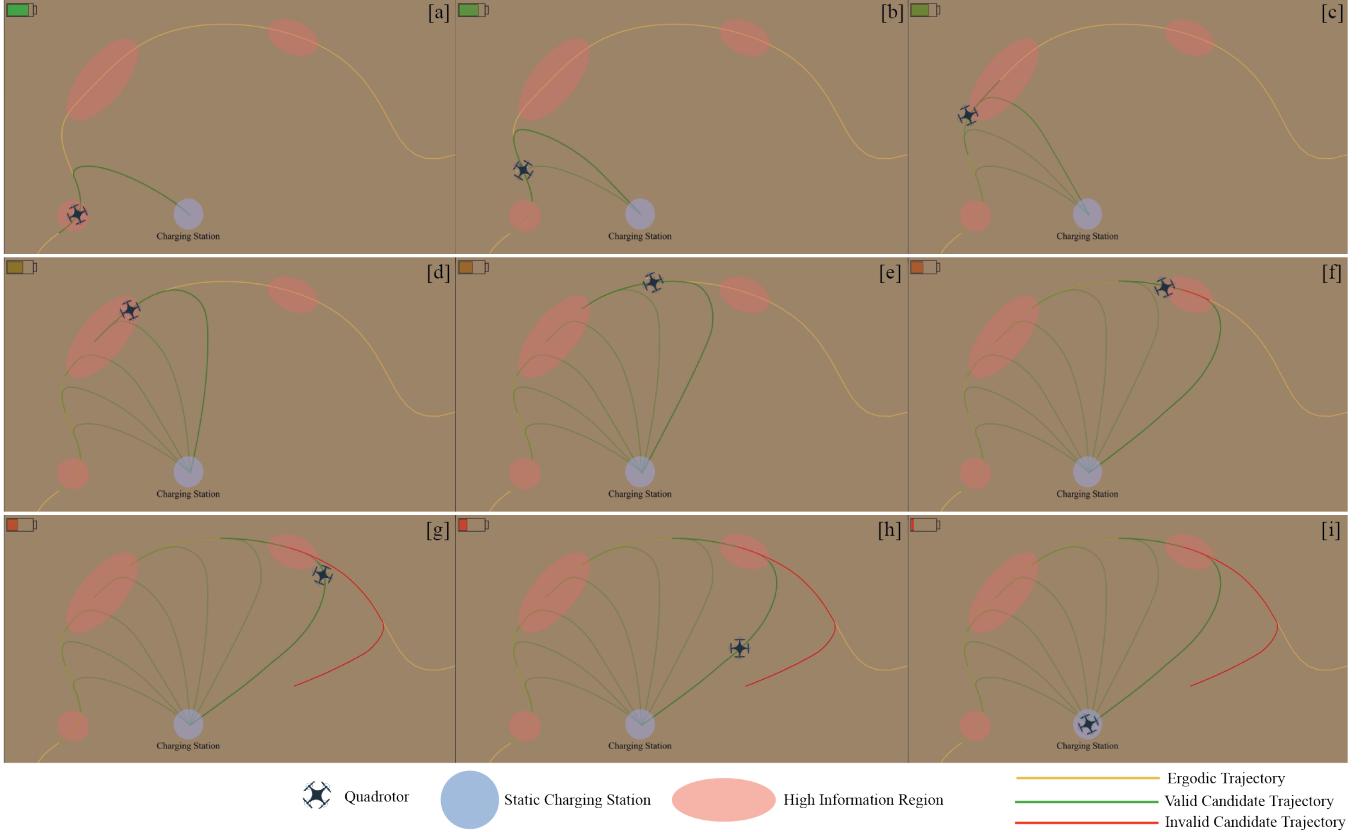


Fig. 6. This figure illustrates the process of generating candidate trajectories iteratively. The green trajectories are the valid candidate trajectories or the committed trajectories. These trajectories follow part of the ergodic trajectory (shown in yellow) and then go to the charging station. If the candidate trajectory cannot ensure the quadrotor's safe return to the charging station, it is considered invalid (shown in red) and the quadrotor takes the last committed trajectory back to the charging station.

control input from T_H horizon ergodic control trajectory $\tilde{u}_{1:N_b}^{ref}$, and $N_n = \frac{T_N}{dt}$. For the convex MPC problem, we use the linearized version of the nonlinear quadrotor dynamics described in (14). It is important to note that the linearized quadrotor system with $A \in \mathbb{R}^{13 \times 13}$ and $B \in \mathbb{R}^{13 \times 4}$ is not controllable with rank 12. Therefore, we reduce the system to $\bar{A} \in \mathbb{R}^{12 \times 12}$ and $\bar{B} \in \mathbb{R}^{12 \times 4}$ using the attitude Jacobin for error dynamics proposed by [25]. For more details, please see [25]. The reference state trajectory and reference control trajectory used in the MPC problem (21) are formed using the double integrator ergodic trajectory with unit quaternion. Apart from this method, differential flatness property can be used to track the double integrator trajectory using the nonlinear geometric controller [26]. We generate the nominal trajectory for time horizon $T_N = 2.0$ sec with sampling time $dt = 0.05$.

2) back-to-base(b2b) Trajectory: The b2b robot state trajectory $\tilde{x}_{1:N_b}$ and control trajectory $\tilde{u}_{1:N_b-1}$ are generated using the convex MPC formulation given as

$$\begin{aligned} \min_{\tilde{x}_{1:N_b}, \tilde{u}_{1:N_b-1}} \quad & \sum_{k=1}^{N_b-1} \left[(\tilde{x}_k - x_c)^T \mathbf{Q} (\tilde{x}_k - x_c) + (\tilde{u}_k)^T \mathbf{R} (\tilde{u}_k) \right] \\ \text{s.t.} \quad & \tilde{x}_1 = \tilde{x}_{N_h}, \\ & \tilde{x}_{k+1} = f(\bar{x}_k, \bar{u}_k) + \left[\frac{\partial f}{\partial x} \Big|_{\bar{x}_k, \bar{u}_k} \right] (\tilde{x}_k - \bar{x}_k) \\ & \quad + \left[\frac{\partial f}{\partial u} \Big|_{\bar{x}_k, \bar{u}_k} \right] (\tilde{u}_k - \bar{u}_k) \end{aligned} \quad (22)$$

where x_c is the state of the charging station and the initial state \tilde{x}_1 is the last state of the T_N horizon ergodic trajectory. We use the same reduced linearized quadrotor dynamics to generate a b2b trajectory. We use the values of $T_B = 10.0$ sec with sampling time $dt = 0.05$.

Figure 6 illustrates the process of generating candidate trajectories iteratively. The green trajectories are the valid candidate trajectories or the committed trajectories. These trajectories follow part of the ergodic trajectory (shown in yellow) and then go to the charging station. If the candidate trajectory cannot ensure the quadrotor's safe return to the charging station, it is considered invalid and the quadrotor

takes the last committed trajectory back to the charging station.

D. Construction of the hypothetical stochastic spatiotemporally varying environment

We constructed a hypothetical stochastic spatiotemporally varying environment using the Salinity data from the Gulf of Mexico [27]. Salinity, one of the oceanographic quantities, refers to the concentration of dissolved salts in water. In the oceanographic community, salinity evolution in a water body can be used to study cross-shelf exchanges. Cross-shelf exchanges refer to the movement of nutrients from the shallow waters to the open ocean. We extracted the salinity mean values along with the standard deviation for a specific region and then scaled down the environment spatially and temporally to form a hypothetical environment.

REFERENCES

- [1] S. Mayer, L. Lischke, and P. W. Woźniak, “Drones for search and rescue,” in *1st International Workshop on Human-Drone Interaction*, 2019.
- [2] S. Waharte and N. Trigoni, “Supporting search and rescue operations with uavs,” in *2010 international conference on emerging security technologies*. IEEE, 2010, pp. 142–147.
- [3] G. Gawarkiewicz, R. E. Todd, W. Zhang, J. Partida, A. Gangopadhyay, M.-U.-H. Monim, P. Fratantoni, A. M. Mercer, and M. Dent, “The changing nature of shelf-break exchange revealed by the ooi pioneer array,” *Oceanography*, vol. 31, no. 1, pp. 60–70, 2018.
- [4] R. E. Todd, “Export of middle atlantic bight shelf waters near cape hatteras from two years of underwater glider observations,” *Journal of Geophysical Research: Oceans*, vol. 125, no. 4, p. e2019JC016006, 2020.
- [5] P. Sujit, J. Sousa, and F. L. Pereira, “Uav and auvs coordination for ocean exploration,” in *Oceans 2009-Europe*. IEEE, 2009, pp. 1–7.
- [6] S. Manjanna, A. Q. Li, R. N. Smith, I. Rekleitis, and G. Dudek, “Heterogeneous multi-robot system for exploration and strategic water sampling,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4873–4880.
- [7] K. D. Julian and M. J. Kochenderfer, “Distributed wildfire surveillance with autonomous aircraft using deep reinforcement learning,” *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 8, pp. 1768–1778, 2019.
- [8] 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, pp. 432–442, 2011. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S016727891000285X>
- [9] L. Dressel and M. J. Kochenderfer, “Tutorial on the generation of ergodic trajectories with projection-based gradient descent,” *IET Cyber-Phys. Syst.: Theory & Appl.*, vol. 4, no. 2, pp. 89–100, 2019. [Online]. Available: <https://doi.org/10.1049/iet-cps.2018.5032>
- [10] D. Dong, H. Berger, and I. Abraham, “Time optimal ergodic search,” *arXiv preprint arXiv:2305.11643*, 2023.
- [11] B. Haydon, K. D. Mishra, P. Keyantuo, D. Panagou, F. Chow, S. Moura, and C. Vermillion, “Dynamic coverage meets regret: Unifying two control performance measures for mobile agents in spatiotemporally varying environments,” in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 521–526.
- [12] Y. Sung, D. Dixit, and P. Tokekar, “Environmental hotspot identification in limited time with a UAV equipped with a downward-facing camera,” *CoRR*, vol. abs/1909.08483, 2019. [Online]. Available: <http://arxiv.org/abs/1909.08483>
- [13] G. Notomista, S. F. Ruf, and M. Egerstedt, “Persistification of robotic tasks using control barrier functions,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 758–763, 2018.
- [14] G. Notomista, C. Pacchierotti, and P. R. Giordano, “Online robot trajectory optimization for persistent environmental monitoring,” *IEEE Control Systems Letters*, vol. 6, pp. 1472–1477, 2022.
- [15] H. Fouad and G. Beltrame, “Energy autonomy for robot systems with constrained resources,” *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3675–3693, 2022.
- [16] W. Bentz and D. Panagou, “Energy-aware persistent coverage and intruder interception in 3d dynamic environments,” in *2018 Annual American Control Conference (ACC)*, 2018, pp. 4426–4433.
- [17] I. Abraham, A. Prabhakar, and T. D. Murphy, “An ergodic measure for active learning from equilibrium,” *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 917–931, 2021.
- [18] H. Coffin, I. Abraham, G. Sartoretti, T. Dillstrom, and H. Choset, “Multi-agent dynamic ergodic search with low-information sensors,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 11480–11486.
- [19] A. Rao, A. Breitfeld, A. Candela, B. Jensen, D. Wettergreen, and H. Choset, “Multi-objective ergodic search for dynamic information maps,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 10197–10204.
- [20] A. C. Garza, “Bayesian models for science-driven robotic exploration,” Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, September 2021.
- [21] D. R. Agrawal and D. Panagou, “Sensor-based planning and control for robotic systems: Introducing clarity and perceivability,” *IEEE Control Systems Letters*, vol. 7, pp. 2623–2628, 2023.
- [22] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.
- [23] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, “Hamilton-jacobi reachability: A brief overview and recent advances,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 2242–2253.
- [24] D. Agrawal, R. Chen, and D. Panagou, “gatekeeper: Safety verification and control for nonlinear systems in unknown and dynamic environments,” 2023.
- [25] B. E. Jackson, K. Tracy, and Z. Manchester, “Planning with attitude,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5658–5664, 2021.
- [26] T. Lee, M. Leok, and N. H. McClamroch, “Geometric tracking control of a quadrotor uav on se (3),” in *49th IEEE conference on decision and control (CDC)*. IEEE, 2010, pp. 5420–5425.
- [27] “Ncss for grids (grid as point dataset),” accessed: 2023-07-13. [Online]. Available: <https://ncss.hycom.org/thredds/ncss/grid/GOMI0.04/expt.32.5/2019/dataset.html>