Ergodic Graph Exploration via Markov Chain for Active Robotic Information Acquisition

Benjamin Wong

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2026

Reading Committee:
Santosh Devasia, Co-Chairs
Ashis G. Banerjee, Co-Chairs
Joseph L. Garbini

Program Authorized to Offer Degree:
Mechanical Engineering

University of Washington

Abstract

Ergodic Graph Exploration via Markov Chain for Active Robotic Information Acquisition

Benjamin Wong

Co-Chairs of the Supervisory Committee:
Santosh Devasia
Mechanical Engineering

Ashis G. Banerjee
Mechanical Engineering
Industrial & Systems Engineering

Many robotic applications can be considered as information acquisition, including surveillance, environmental monitoring, disaster response, and robotic learning. These tasks are often a combination of being repetitive, time consuming, time sensitive, or dangerous, which are unsuitable for human to perform. Specifically, this work consider inspection of confined spaces as the primary example. This work presents a semi-autonomous robotic system for assisting human to perform inspection in such hazardous environment. Challenges arise for robots to operate autonomously in these cluttered, poorly illuminated environment with complex connectivity such as localization, mapping, and navigation. Teleoperation also poses challenge as communication is limited with the confined space being enclosed in large metallic structures. This work focus on autonomous inspection with regard to foreign object debris (FOD) detection. First, an statistical FOD detection method is presented, accounting for mapping uncertainty in various location of the tank. The result is verified by the operator at the end of each inspection session. Second a hierarchical planning method is presented for optimizing the detection rate of FOD while handling the complex connectivity and limited navigation capability. Last, the planning method is generalized to a multi-robot system for collecting information in a large complex environment.

TABLE OF CONTENTS

Nomenclature

CAD	Computer-Aided Design
EKF	Extended Kalman Filter
FMMC	Fastest Mixing Markov Chain
FOD	Foreign Object Debris
GESCE	Graph-based Ergodic Search in Cluttered$_{E}nvironments$
HEDAC	Heat Equation Driven Area Coverage
HEMaP	Hierarchical Ergodic Markov Planning
MAB	Multi-Armed Bandit
MC	Main Contribution
MPC	Model Predictive Control
PRM	Probabilistic Road Map
REMC	Rapidly Ergodic Markov Chain
ROS	Robot Operating System
RQ	Research Question
SE(3)	3D Special Euclidean Group

SLAM	Simultaneous Localization and Mapping
SLEM	Second Largest Eigenvalue Modulus
SMC	Spectral Multiscale Coverage
UAV	Unmanned Aerial Vehicle

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

Table Number Page

DEDICATION

to people who has the free time to read a dissertation

Chapter 1

INTRODUCTION

1.1 Motivations

Confined space is defined as place that can be entered by human but has limited entrance and exit, and it is not designed for prolonged human occupancy [97][21], examples include: Vehicle interior such as wing tank in airplane and ballast tank in ships, sewers, and storage tanks such as grain bins. These spaces poses significant hazards for workers required to perform jobs within, including, but not limited to, accumulation of harmful gases, material engulfment such as water and grain submersion, and flammable gases [22]. Numerous worker injuries and deaths have occurred for the past decade due to performing jobs in these confined spaces [121].

Figure 1.1: Example image of a worker inspecting a dry docked ballast tank [1]

Robotic solutions have been a continuous interest for the industry to mitigate these hazards by aiding the workers to perform tele-operations without entering the confined space. One notable challenge being locomotion within these complex structures. Many works have focused on novelty designs to overcome these challenges such as fitting a continuous robots through the entry [32] [34], crawlers that can adhere to or climb over obstacles [110][109], and collision safe drones [52] [108]. Another challenge is the autonomy of these robots in such clustered environments. Research has been done in the areas of localization [20] [111], manipulation [88] [89] [47], navigation, [94] [10] [29] and exploration [6].

This work specifically focuses on the autonomous inspection of a ballast tank (example shown in Fig. 1.1) in the context of exploration of a partially known environment.The inspection robot should inspect different regions stochastically according to some characteristics of the regions. In addition, a connection with designing such random walk can be drawn with a rapidly developing field of control theory named ergodic control, first introduced in [78]. Ergodicity is a concept introduced by Ludwig Boltzmann in statistical mechanics, where he asserted, in short, the time trajectory of a gas molecule in an constant energy surface will visit every region in the phase space with equal amount of frequency [113]. Later formalized by Birkhoff that a system is ergodic if the statistic of a time trajectory of any point in a set is equal to the statistic of the entire set [14]. In the context of ergodic control in robotics, one would like to design a control law such that the statistic along the trajectory of the robot equal to some desired spatial statistics. In particular, the popular application is for coverage where the robot frequency of the robot visiting each point in space matches with the desired visitation distribution, example illustrated in Fig. 1.3.

This work focuses on synthesis of ergodic system on a graph using stochastic transition policy.

---

[1] Image extracted from [64]

[2] Image extracted from [83]

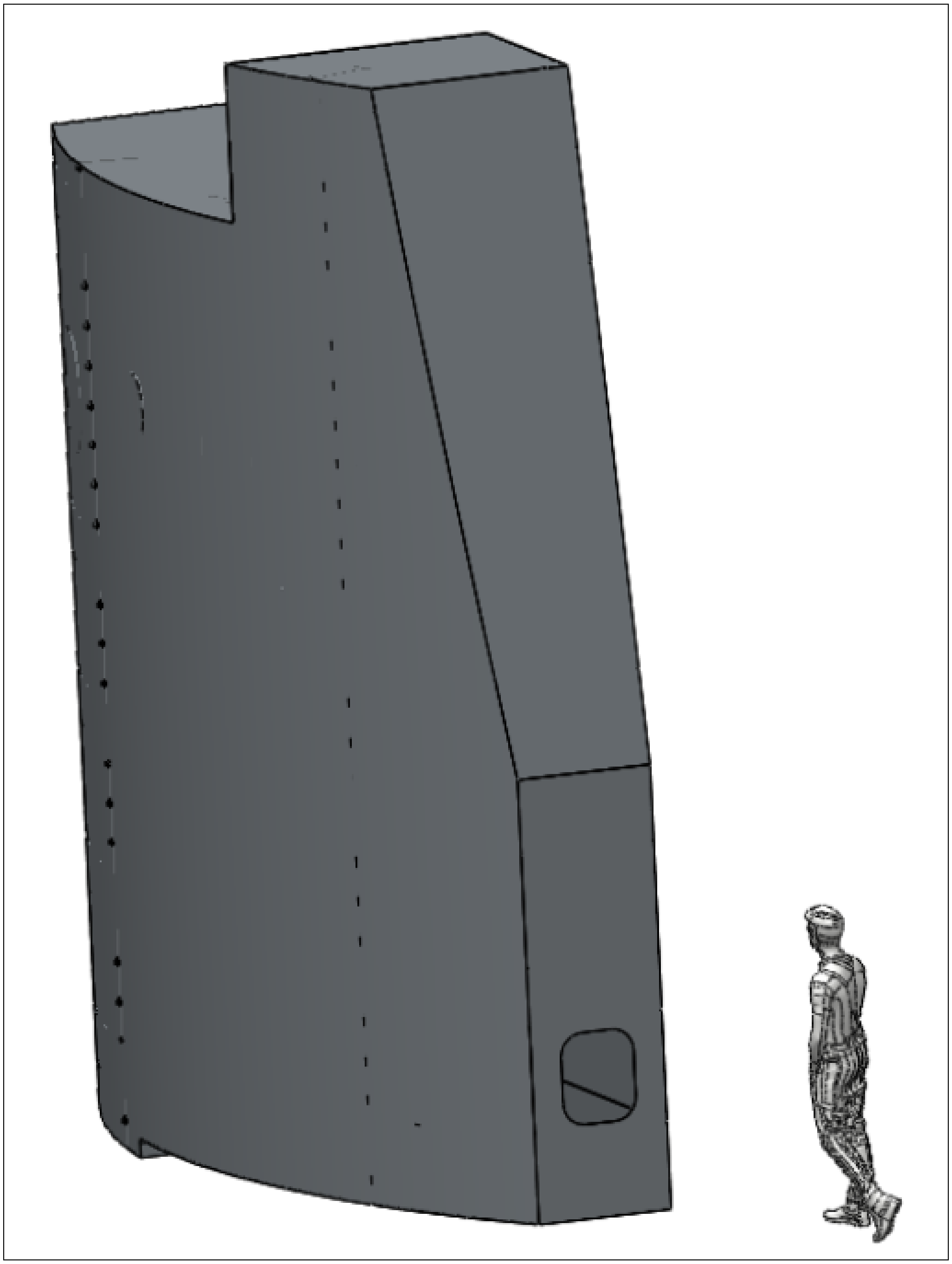


Figure 1.2: CAD model of the ballast tank in comparison to a human model.

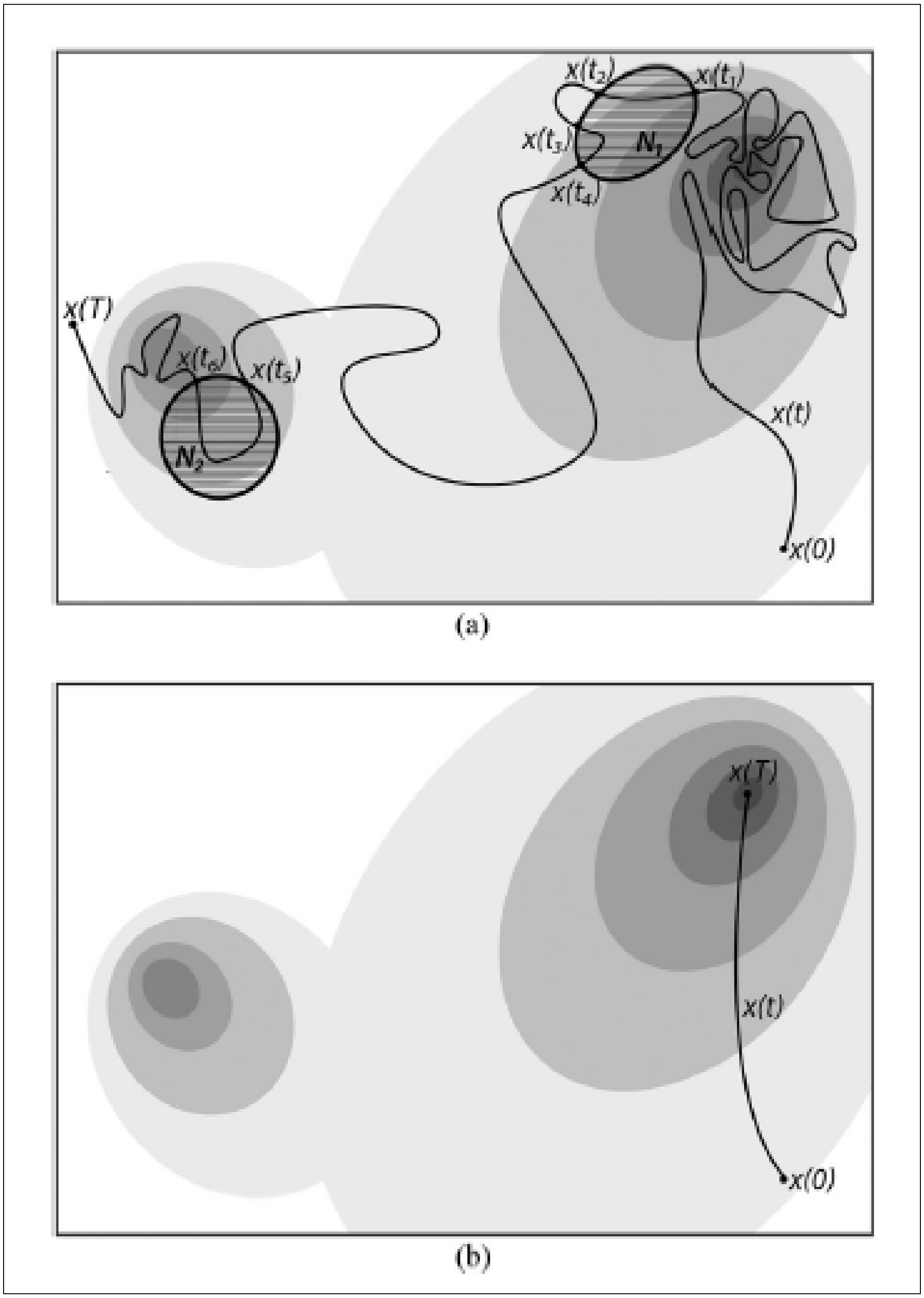


Figure 1.3: Example of ergodic trajectory (a) compared to maximization trajectory (b). [2]

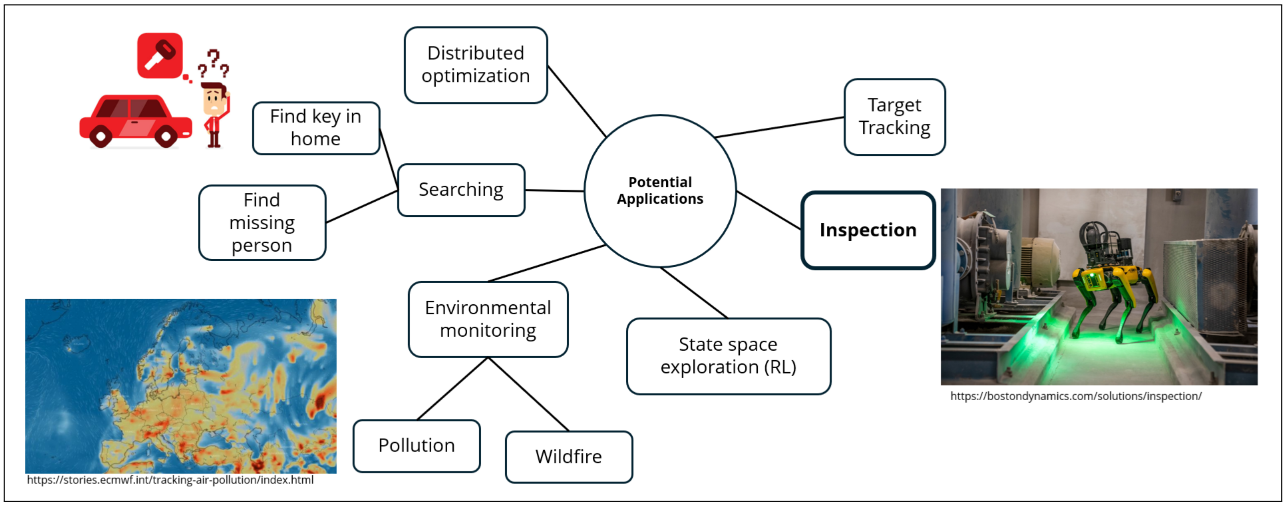


Figure 1.4: Potential Application of ergodic graphs in robotic systems.

1.2 Problem Statement and Research Questions

Problem Statement: How can a robot/a team of robots inspect a partially known ballast tank efficiently and robustly.

1.2.1 ***RQ1:*** *How to detect FOD in a partially known ballast tank?*

FOD can cover a wide range of items that does not belong in the workspace. One can build a exhaustive library of classes of foreign objects and utilize image classification models. Aside from requiring extensive amount of training, it also raise concern performance about unseen classes and general problem of overconfidence of machine learning models. The objective of this question is to design a method that utilize the known nominal condition and can be interpret statistically for human intervention.

1.2.2 ***RQ2:*** *What is an efficient and scalable way to plan a inspection route that works with the FOD detection method?*

Popular planning methods focus on maximizing rate of detection, usually require accurate model of observations and the associated information gain. Sampling methods or elaborate optimizations are used to solve such problem. This question seeks to investigate what planning method can be used to work in tandem with the previous FOD detection method while robust to inaccurate modeling and efficient to be ran on board of the robot.

1.2.3 ***RQ3:*** *How to allocate observation effort with uncertainty of information?*

With the property of ergodicity, the ergodic method in MC2 can be generalized to a multi-robot system without modification, with the stochasticity approaches 0 as the number of robot increases. A good target distribution needs to be investigated to effectively allocating the robots effectively. Existing research, as well as our work in MC2, uses the information gain to generate the target distribution directly. However the information gain itself can also be unreliable due to it being estimated from the observation. This question seeks to investigate a proper target distribution that can account for the uncertainty of information gain.

1.3 Main Contributions

1.3.1 ***MC1****: FOD detection via Mahalanobis Distance*

MC1 answers RQ1 by using the Mahalanobis distance between the CAD model of the ballast tank and the point cloud build by visual simultaneous localization and mapping (V-SLAM). Points with high statistically significant Mahalanobis distances are clustered and photos are taken for human verification.

1.3.2 ***MC2****: Ergodic Graph Traversal with Hierarchical Planning*

MC2 answers RQ2 by partitioning the ballast tank in to regions and arranged them into a graph by connectivity. The robot travel between the regions by sampling from the rapid ergodic Markov chain, where the target distribution is specified by the estimated information from the FOD detection.

1.3.3 ***MC3****: Graph Ergodic Swarm with Annealed Ergodic Traversal*

MC3 answers RQ3 by applying simulated annealing to the information gain to generate the target distribution that starts from uniform (which is good for unreliable estimated information gain) and gradually shift toward the theoretical optimal distribution (which is optimal for accurate information gain).

1.4 Organization

The main contributions are summarized in Fig. 1.5. The remainder of this document is organized as follow: Related works are presented in Chapter 2; The FOD detection method is discussed in Chapter 3; the ergodic graph planning method and the corresponding modification of the FOD detection method is discussed in Chapter 4; and the extension to multi-robot annealed ergodic graph planning is discussed in Chapter 5.

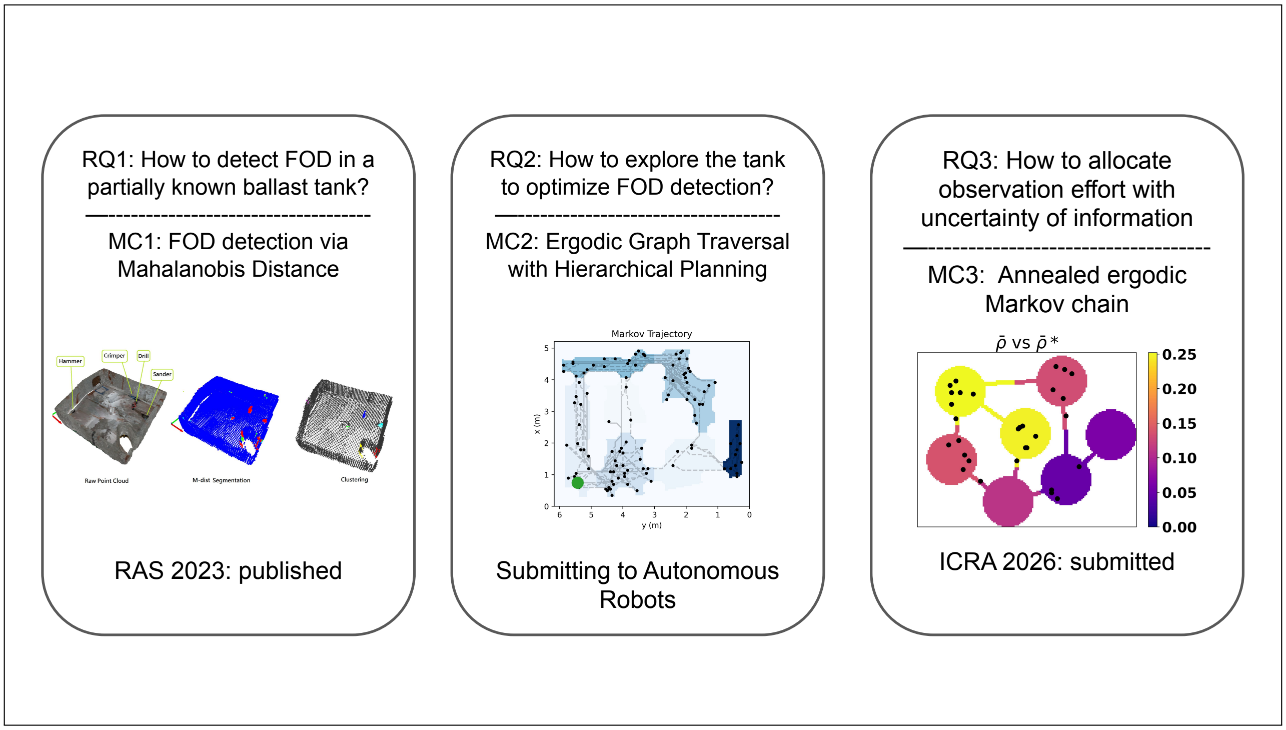


Figure 1.5: Summary of research timeline with main contributions and publications

Chapter 2

RELATED WORKS

2.1 Robotic FOD detection

There are several related works focusing on inspection of confined space using localization, mapping and navigation, including semi-automated inspection of an industrial combustion chamber [111]; submerged building mapping by an autonomous vehicle [94]; UAV-based localization in ballast tanks [20]; intelligent exploration in mines using a small drone [6]; autonomous navigation through manhole-sized confined spaces using a collision-tolerant aerial robot [29]; and, semi-autonomous inspection of underground tunnels and caves [10].

Other open space FOD detection applications include inspection of aircraft damages [119] and power transmission lines [123, 48]; quality control of graphics card assembly lines [67]; and real-time logistics monitoring [117] with visible light cameras. For confined space FOD detection, to the best of our knowledge, there has been only one reported work so far, where Latimer investigated processing of depth camera images for aircraft wing inspection [69]. On a somewhat related note, real-time detection of the differences of industrial parts from their corresponding computer-aided design (CAD) models has been done by processing the point clouds generated by hand-held laser scanners [63].

2.2 Ergodic Control and Markov Chain

2.2.1 Ergodic Control

There are three main approach to develop a ergodic controller The earliest ergodic control method for robotic trajectory generation can be traced back to SMC from [78], where the workspace is assumed to be rectangular. The target distribution is then transformed into the Fourier domain, then the control input is calculated as an LQ optimization to minimize the difference between the selected finite Fourier coefficients between the robot's time distribution and the space distribution. By requiring Fourier transform, this method is limited to flat rectangular domain. Additional, by only considering finite Fourier coefficients of the space distribution, it has no collision guarantee which can be limited in a confined space setting.

The second approach is developed by the group from [70, 4, 83], where the SMC method is expanded as a trajectory optimization problem, where metric from SMC is used as the objective function for a MPC-like nonlinear controller. As an MPC formulation, this method is highly flexible with various cost and constraints such as input cost, specifically, collision avoidance can be handled by control barrier functions. The downside is that, depending on the non-linearity of the problem and the choice of constraints, nonlinear MPC can be computationally intensive, this also compounds with the dimensional complexity of SMC. [9] reduced the computation complexity of the trajecotry optimization framework by sarcastically sampling the optimal input with bayesian optimization. [42] extended the framework to a decentralized multi-agent system.

The third approach, from the group of [55, 58, 54, 57], tackled the problem of SMC requiring a flat, rectangular domain by replacing the Fourier basis with radial basis function. The control law is created by modeling potential field using the heat equation, with the different between the current robot distribution and the target distribution as the heat source. The robot then calculates the control input according to the gradient of the potential field. Similar to our method, by choosing appropriate basis function, HEDAC is flexible with the topology of the workspace. This method requires computationally intense prepossessing such as finite element method for computing the eigen basis function for the workspace and may not be suitable for partially known workspace or dynamically changing workspace such as the ballast tank with unknown FOD scattered around.

In general,these continuous space methods generate complex trajectories that are hard to execute in a clustered confined space and unnecessary for visual inspection. By using graph-based approach one can be ergodic on the region sense while maintain simple trajectory in the continuous space. To the best of our knowledge, there are no ergodic control applied to a graph.

2.2.2 Markov Chain Optimization

Another related field to our approach is optimization of Markov chains. [16] can be considered the seminal work for optimizing a Markov chain. The chapter shows that the rate for a symmetric Markov chain to converge to it's stationary distribution can be optimized by minimizing the second largest eigenvalue modulus, and it forms a convex optimization problem. The method can be extended to optimize for reversible Markov chain, i.e. a Markov chain that satisfies the detailed balance condition. We will show later in this section that while FMMC can be applied to improve the convergence rate of ergodicity, it is not necessarily the optimal solution.

[11] formulated an application of FMMC to robotics, where the group utilized the continuous time formulation of FMMC for task allocation for a team of robots. A discrete set of tasks that requires various amount of robots are arranged into a graph by proximity and reachability. The chapter shows that with a sufficiently large group of robots, the desired proportion of robots to each task can be allocated by broadcasting the FMMC stochastic matrix to all agents. Additionally, a transition rate is modeled between tasks to account for traveling time. Inheriting from FMMC, this work is different from our approach in that it considers the distribution at each time instant but not the time history of the distribution as in ergodic control.

2.3 Information Acquisition

In a broad sense, this work falls in the domain of active planning, where agents plan trajectories that satisfy an objective and continuously adjust their trajectories as new information arrives. In this case, the objective is to obtain information through sampling. A popular choice of doing so is to find a trajectory that covers the entire space [23, 39, 124, 102]. This choice assumes that all the points in the space are equally important and lacks the flexibility to adapt to the quality of information. Alternatively, to consider information density, sensor placement through Voronoi partitioning has been used [25, 71, 33, 75]. This is well-suited for a static placement problem, where the robots converge to the optimal position until a new event arises. This requires a known number of robots and assumes all robots to be active, and does not account for the need of redundancy. On the other hand, in ergodic control, all the agents provide coverage individually, which make the system more robust to sensor and actuation failures and varying numbers of active robots.

Our formulation of obtaining information from discrete locations is essentially a problem of sequential experiment design [95]. A particular form that has been thoroughly studied is the multi-armed bandit (MAB) problem, where the agents maximize their rewards under unknown distributions [106]. Traditionally, MAB assumes that any arm can be sampled at any time. Recently, work has been done on applying the MAB problem constrained on a graph [122], similar to our problem formulation. While we can consider the information as a form of reward in our formulation, the biggest difference with the MAB problem is that the MAB rewards are fungible, where agents can collect solely from the state with the maximum amount of reward. On the other hand, in our formulation, sufficient information has to be collected from all the regions, i.e. missing information in one region cannot be substituted by more information from another region.

Chapter 3

FOD DETECTION VIA MAHALANOBIS DISTANCE

3.1 Technical Approach

To accomplish effective and efficient inspection, this chapter describes the incorporation of human-in-the-loop decision making to assist the semi-autonomous inspection system, based on our published work in [8].

The system is primarily running online under the Robot Operating System (ROS) framework, with various offline pre-processing modules to enhance the online performance. Preprocessing includes defining an inspection route for every inspection session; and generating a reference point cloud from either the CAD model or the collected nominal maps. The online inspection is divided into four major phases, as illustrated in Figure 3.1. In the first phase, the robot collects the query point cloud by performing SLAM either via teleoperation or autonomous navigation using the predefined inspection route. The second phase begins when the SLAM is completed. The point cloud from the visual SLAM is exported and compared with the reference point cloud. Points with high discrepancy are segmented and clustered into FOD candidates. In the third phase, the centroid of each candidate is projected to the navigation cost map to determine the waypoints from where the robot expects to see the candidate cluster. The robot then covers all the waypoints, selecting the one closest to its current location as the next destination, and takes (camera) pictures of the FOD candidates. In the final phase, the pictures taken by the robot are presented to a human to determine whether FODs are present.

3.1.1 SLAM and Navigation

The SLAM functionality is provided by RTAB-Map via ROS [68]. The main configuration follows the default launch file from rtabmap_ros[1], with the modified parameters and nodelets shown in 3.A. The main objectives for the SLAM are self-localization in the confined space, and construction of a point cloud for each inspection session using a depth camera. RTAB-Map is chosen based on its open source ROS implementation, a wide variety

---

[1] `https://github.com/introlab/rtabmap_ros/blob/master/launch/rtabmap.launch`

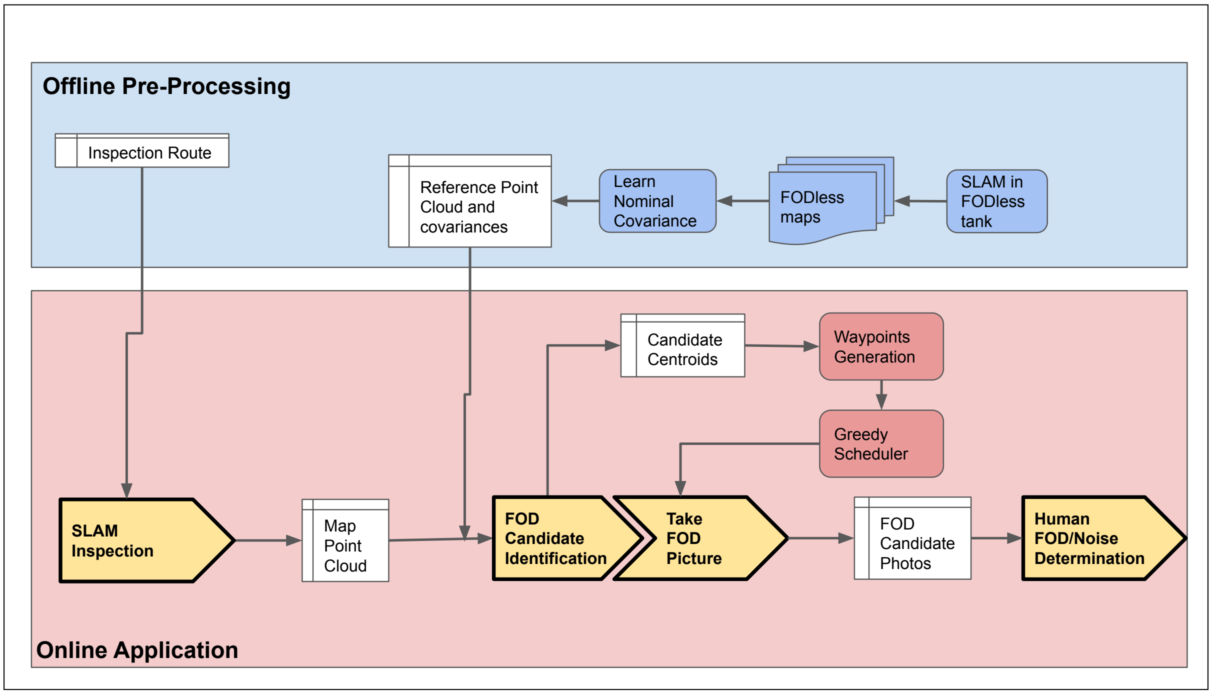


Figure 3.1: Overall pipeline of the FOD detection system

of sensor compatibility, and good reported performance on many different tasks. In addition, RTAB-Map has a modular approach toward odometry, which allows a simple switch between wheel odometry, in-built visual odometry, or any other third party odometry algorithm without affecting the SLAM functionality. The navigation functionality is provided by the ROS navigation stack in the TurtleBot3 package. The navigation stack takes the 2D occupancy grid map generated by the SLAM package as an input and creates a 2D occupancy grid cost-map in real-time. Once a waypoint is published to the waypoint topic, the optimal trajectory from the current location to the specified waypoint is calculated and executed through velocity control.

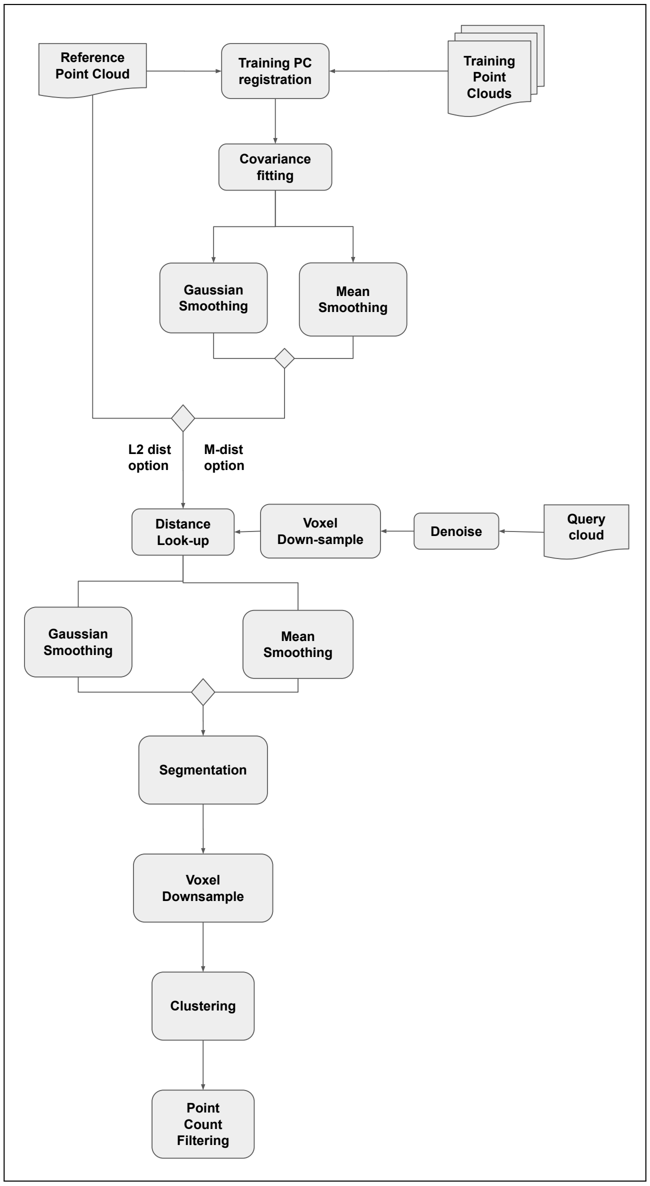


Figure 3.2: FOD candidate identification procedure

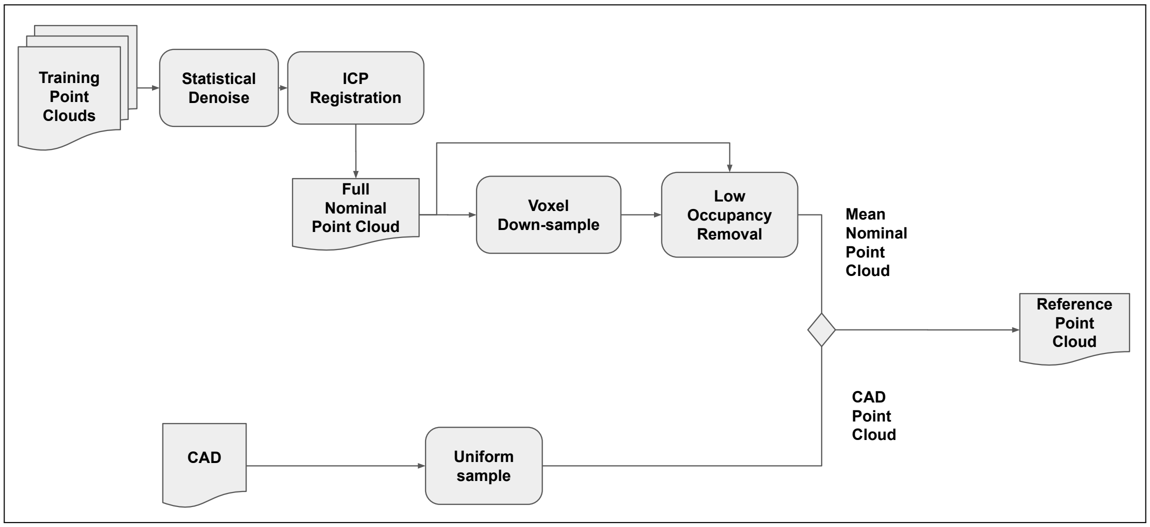


Figure 3.3: Reference point cloud generation procedure

3.1.2 FOD Candidate Identification

We develop our own FOD candidate identification algorithm to detect potential foreign objects from the SLAM point cloud while preventing misclassification due to noise. The core idea is to compare the *discrepancy* of each point in the point cloud to a nominal reference map of the environment. The overall pipeline for the identification process with the various options is shown in Figure 3.2.

Reference Point Cloud Generation

The reference point cloud is a point cloud representation of the water tank that is known to be FOD-free. It is treated as the ground truth and any points that deviate from it by more than the cutoff value are considered FOD candidate points. Here, we consider two options for reference point cloud generation. It is created either using a mesh of the tank CAD model or by collecting sample point clouds in the FOD-free tank. Figure 3.3 shows the procedure of generating the reference point cloud using both the options.

The CAD mesh method assumes that a accurate model of the environment is available, wherein, the CAD model is first exported as a PLY format triangle mesh and the mesh is uniformly sampled to create a dense point cloud. The sample point clouds method, on the other hand, is developed for use in a confined space with a large number of unmodeled structures and/or construction variations from its CAD model. In this case, all the sample point clouds from the inspection training sessions are first de-noised and then registered, either with the CAD model or one of the sample clouds. Next, all the sample point clouds are merged into a single point cloud. The merged point cloud is then voxel down-sampled to calculate the mean points occupying a single voxel. This down-sampling process merges all the points in a voxel into a single point regardless of the number of points. However, such a merger sometimes results in noisy points creating false voxel occupancy conditions. To eliminate this effect, all the points generated from a voxel with occupancy counts less than a threshold quantile are removed.

Covariance Fitting

The discrepancy metric is a scalar value assigned to each point in a point cloud. The metric scales according to the estimated amount of deviation of these points to their corresponding points in the reference point cloud. Euclidean distance for nearest neighbors is used as the *de facto* metric while comparing the two point clouds. In our case, we consider the nearest neighbors of the points in the query point cloud to the nominal point cloud. However, this consideration often leads to noisy mapping of the walls and beams with high deviation values and either a) cause a large number of FOD candidates for a low distance threshold; or, b) are insensitive to small sized FODs for a high distance threshold.

We address this issue by using a probabilistic approach based on local *Mahalanobis distance* (M-distance). Alternatively, a spatial Chi-squared test can be used for local outlier detection [36]; however, it works well only for homogeneous density point clouds.

$$D_M = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)}. \tag{3.1}$$

The M-distance D_M, shown in (3.1), introduces an expected noise in the form of the covariance matrix Σ. It also adds directionality to the expected noise, which is especially useful in distinguishing the walls with horizontal deviations and beams with vertical deviations. While μ is defined as the closest point from the training or query cloud point x to the reference point cloud, Σ is estimated from the training samples collected for the mean point cloud. If the CAD model is used for reference point cloud generation, a set of training point clouds has to be still collected for Σ estimation.

Instead of estimating one single global covariance matrix for the whole tank, we compute local covariances. It is done by first calculating the signed spatial error

$$\Delta x_{i,j} = \begin{pmatrix} x_j \\ y_j \\ z_j \end{pmatrix} - \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \tag{3.2}$$

from each point x_j on the sample point clouds to their nearest neighbor point x_i on the reference point cloud. Next, for each point on the reference point cloud that contains at least one sample point, we calculate the scatter matrix S_i as:

$$S_i = \sum_j (\Delta x_{i,j}) \cdot (\Delta x_{i,j})^T. \tag{3.3}$$

If this quantity is divided by the number of samples, we get the maximum likelihood estimation (MLE) of the covariance for that point on the reference map Σ_i. It is, however, deliberately left undivided to smooth out the covariance matrices within a local region. Otherwise, depending on the number of training maps, the point density of each training map and the voxel size of the nominal map, we risk having a large number of points on the nominal map with zero sample size and many points with a low sample size.

We consider two options for smoothing: mean smoothing and Gaussian smoothing. For mean smoothing, the covariances, $\Sigma_{i,m}$, are calculated regardless of the distance of the neighboring points to the center point using the formula:

$$\Sigma_{i,m} = \frac{1}{\sum_{j=1}^{k} n_j} \sum_{j=1}^{k} S_j. \tag{3.4}$$

Here, k is the number of neighbors around the i-th point on the nominal map, n_j is the numbers of samples of the neighbors, and S_j is the scatter matrix of the j-th neighbor. For Gaussian smoothing, the covariances, $\Sigma_{i,G}$, are computed similarly but with the distances of the neighbors modeled as reliability weights [40]. This weighting method provides a non-frequency based unbiased estimator of the sample covariance. The weights are based on a Gaussian kernel and a tunable roll-off rate parameter σ as:

$$w_j = \exp\left[-\frac{\|\Delta x_{i,j}\|^2}{\sigma^2}\right] \tag{3.5}$$

$$V_1 = \sum_{j=1}^{k} n_j w_j \quad V_2 = \sum_{j=1}^{k} n_j w_j^2 \tag{3.6}$$

$$\Sigma_{i,G} = \frac{1}{V_1 - V_2/V_1} \sum_{j=1}^{k} w_j S_j. \tag{3.7}$$

To define the smoothing neighborhood, we consider two alternatives: k-nearest neighbor and spherical region of interest.

Prior to smoothing, an optional voxel down-sampling step can be performed on the reference point cloud to reduce the computational burden. In that case, a nearest neighbor search is performed to find the closest point from the full point set to the down-sampled point set. The covariances for the points in the full point set are then simply set to the covariances of their nearest neighbors. Figure 3.4 shows an example heat map of the log-determinant of the local covariances in the tank using sample point cloud and mean smoothing. As expected, the highest covariance values occur at the ceiling and behind the column, both of which are not adequately observed by the robot. The access hole area also has high covariance as the tank cover is manually placed, leading to some variations in its location among the different trials.

Discrepancy Query

Once the reference point cloud is generated and the covariances are estimated, a discrepancy metric is applied to query point clouds to identify the (candidate) FODs. Instead of directly

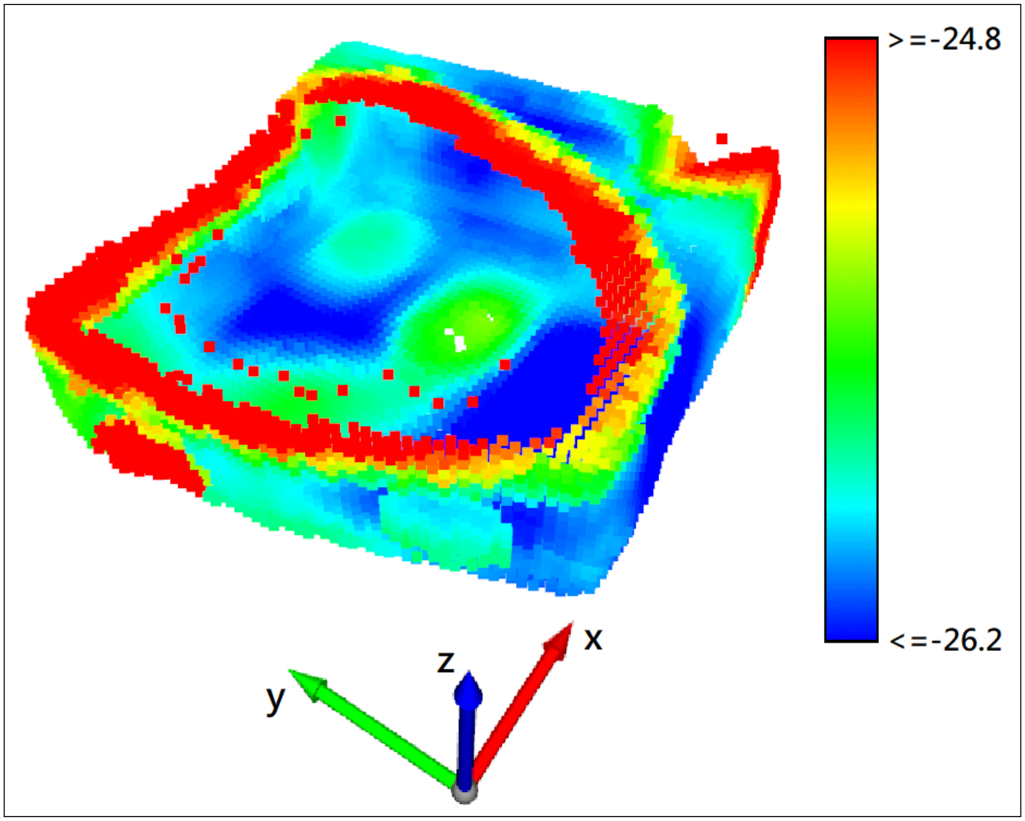


Figure 3.4: Heat map of the log-determinant of the local covariance matrix in the mean point cloud of the FOD-free water tank, with the minimum value clamped to 10^{th} percentile and the maximum value clamped to 90^{th} percentile

computing the metric on the raw point cloud, several steps are applied to reduce noise and improve computation speed. First, a basic statistical outlier removal method is applied to to remove the noisy points that are far away from its neighbors in the query point cloud. A voxel down-sampling is then applied to the point cloud to keep the point count at a manageable level. The number of points corresponding to each voxel, n_i, is also saved as a weight for future use to avoid losing density information from the down-sampling process.

The query process is similar to the training process. The associated point x_j on the reference point cloud is drawn for each voxel point x_i in the query cloud based on the shortest Euclidean distance as the discrepancy metric for the baseline L2-distance method. For the M-distance method, the discrepancy is found with a local version of (3.1) as

$$d_M = \sqrt{(x_i - x_j)^T \Sigma_j^{-1} (x_i - x_j)}, \tag{3.8}$$

where Σ_j is the covariance matrix of the point x_j. Effectively, the noisier (higher covariance) a region is, the smaller is its discrepancy for the query point. For example, in Figure 3.4, the discrepancy of a point with a large distance value near the ceiling is scaled down by the high covariance of the ceiling; a point with the same distance but near the ground has a greater discrepancy due to the smaller covariance of the ground. To reduce high discrepancy values from sparse noisy points, a scalar version of the smoothing used during covariance fitting is applied on the discrepancy metric for each point. Mathematically, it amounts to substituting $n_i d$ for S_i in (3.3), where d is the discrepancy metric. Every point with a discrepancy higher than the threshold is segmented out from the query point cloud and clustered into FOD candidates using hierarchical clustering. The points are agglomerated based on the Euclidean distance to the centroid of the existing clusters. Two clusters are merged if the distance between them is lower than a cutoff value, and any cluster with a fewer number of points than a minimum acceptable point count is rejected from the FOD candidate set. The selection of the threshold, clustering cutoff, and minimum acceptable point count are discussed in the Experiments section. The centroid of each cluster is then sent to the waypoint generation algorithm to proceed to the photo taking phase. Figure 3.5

shows an example of segmented and clustered FOD candidates point cloud, with the points lower than the threshold colored dark grey and each FOD candidate cluster colored using a different hue.

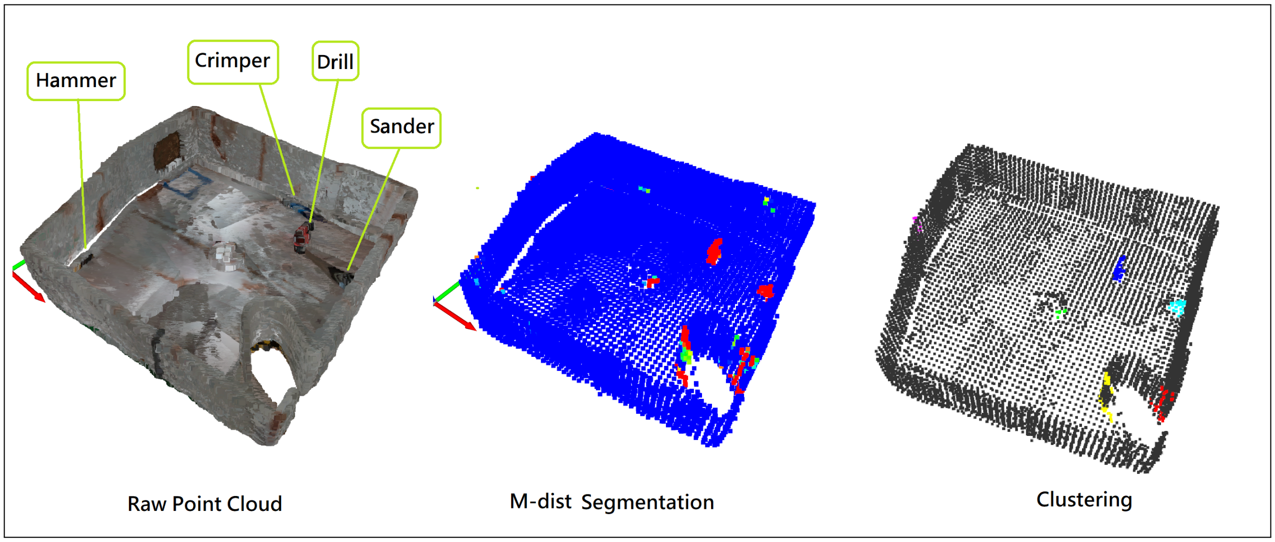


Figure 3.5: An example query point cloud with the high M-distance points segmented and clustered into different candidate FOD clusters with each cluster shown using a distinct hue

3.1.3 Waypoint Generation

Once the FOD candidate locations are computed, they are passed to the waypoint generation module. First, a two dimensional occupancy grid map is created for the query point cloud. To prevent waypoints generation in unreachable locations, a flood fill is performed using the robot location as the seed value. The flood filled image is then subtracted by the original image to remove the walls, and inverted to obtain zero occupancy in the obstacle-free interior of the water tank. The occupancy grid is then inflated into a cost-map according to the ROS navigation stack [77].

Candidate waypoints are created surrounding the FOD location within a minimum and maximum range so as to provide an acceptable image of the FOD. The candidate waypoints are then filtered to avoid colliding with the surroundings and ensure that the FOD is visible. To avoid collisions, the waypoints that overlap with high-cost regions of the cost-map are removed. To ensure that the FOD is visible, each remaining waypoint casts a ray between itself and the FOD. The ray is terminated if it collides with the surroundings in the FOD-less cost-map, and the corresponding waypoint is removed. Each ray records its cumulative cost, which is the sum of the cost in the grid cell under the ray at each time step. The candidate waypoint with the least cost is chosen as the final waypoint. Figure 3.6 shows an example of the waypoint generation process.

3.2 Experiments

The experiments were conducted in two stages. Preliminary testing was done in simulation, followed by extensive physical (hardware) trials on a scaled-down tank prototype. Human subjects studies are included in both the experiment stages.

3.2.1 Simulation

The simulation was done using Gazebo in Ubuntu with ROS Melodic as the framework and Python as the primary programming language. The CAD model of the tank was painted with a rusty white texture and exported as a DAE file using Blender. The DAE file was spawned in Gazebo with all the natural light sources disabled to recreate the dark confined space environment. TurtleBot3 Waffle Pi was chosen as the robot model. The original Pi camera was replaced by two identical cameras using Intel RealSense D435's stereo camera specifications without the active infrared projector. A spot light source was added to the front of the robot to act as a flashlight.

The experiments comprised a total of 30 trials, with 15 trials using the RTAB-Map's built-in visual odometry and another 15 trials using the robot's wheel odometry to study the viability of pure visual odometry. For each trial, the robot was spawned near the access hole with the same pose. Two to five FODs were randomly spawned with random poses from a pool of six FOD types, consisting of drill, screw driver, hammer, wrench, level, and

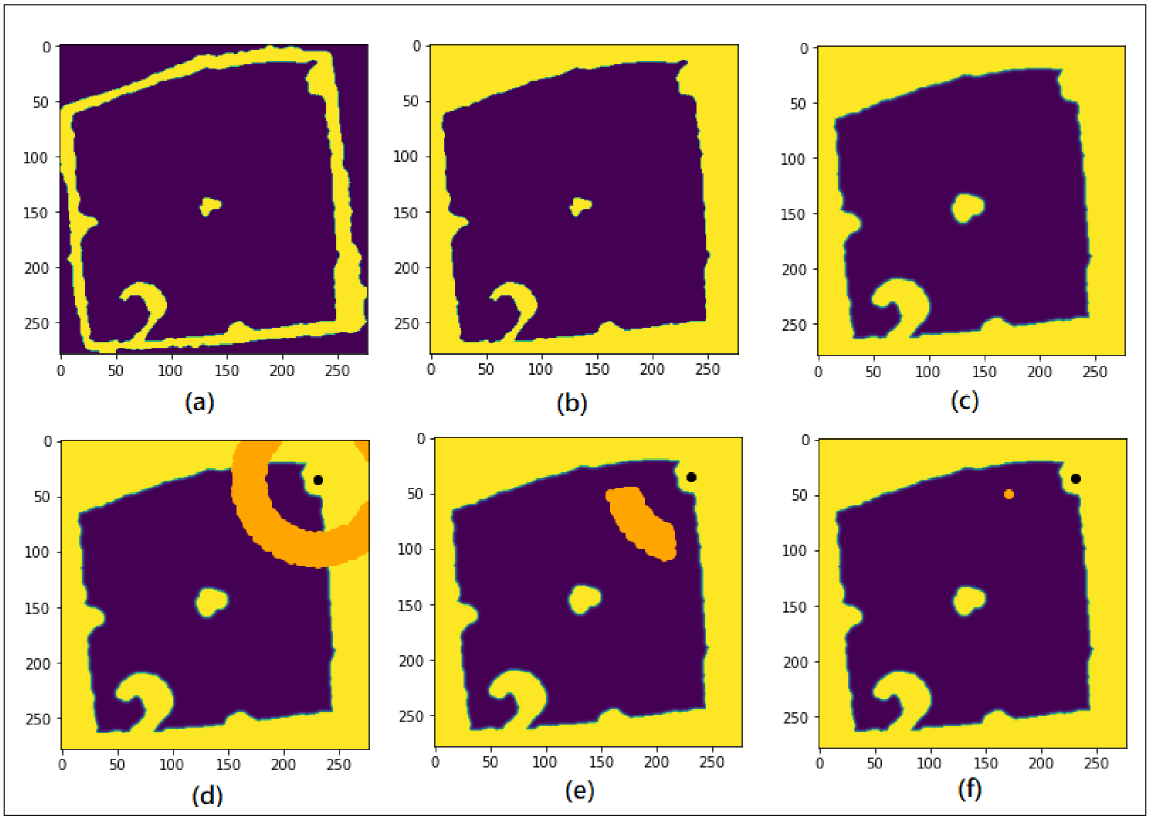


Figure 3.6: Waypoint generation procedure given FOD candidate location: (a) 2D occupancy grid with yellow pixels representing obstacles and purple pixels denoting free space (b) Flood fill to remove unreachable locations outside the tank (c) Cost map generated from the occupancy map (d) Sampling of waypoint candidates along a ring centered at the candidate FOD cluster (e) Filtering of waypoint candidates based on the cost map (f) Final waypoint selected using ray tracing

sander. Point cloud registration was performed using the Open3D iterative closest point (ICP) method. All the parameters were manually chosen. Random down-sampling with a ratio of 0.1 was used in place of voxel down-sample. A CAD model was used as reference point cloud and no denoising was performed. M-distance was used as the metric with Gaussian smoothing ($\sigma = 0.05$), a spherical region of interest (radius of 4σ), a threshold distance of 1.75, and clustering cutoff of 0.275.

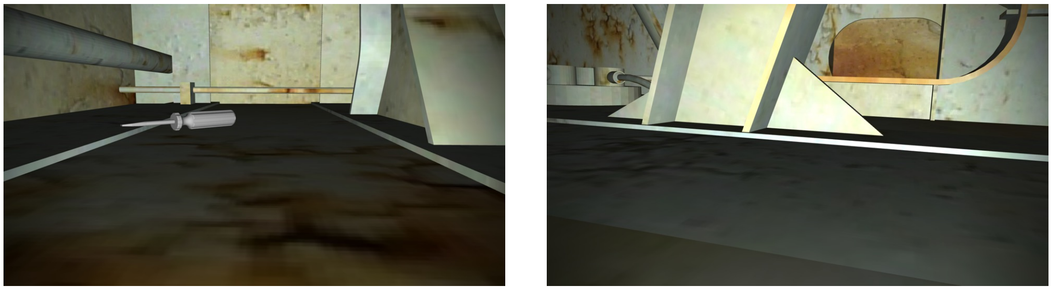
The resultant FOD photos were assembled into online questionnaire surveys, with each survey containing all the photos from a single trial. The administration of the anonymous survey was approved by the University of Washington (UW) Institutional Review Board with the study # STUDY00013902. The surveys were sent out to UW students and Naval Undersea Warfare Center (NUWC) Division Keyport personnel. A total of 23 responses were received, of which 61.9% were engineers, 33.3% graduate students, and 4.8% managers. 57.1% were in the 18-30 age group, 28.6% in the 31-45 group, and 14.3% in the 46-60 group. 81.0% of the participants were males and 19.0% were females.

We first analyzed the performance of the FOD detection approach before looking at the effectiveness of remote humans in making the final decisions. The relevance of the FOD photos is shown in Table 3.1, where the "photo contains FOD" category includes the same FOD appearing in multiple photos and photos containing partial FOD images. The precision is defined as the number of photos containing FODs divided by the total number of photos. It is less than 50% for both the odometry methods, with wheel odometry being slightly lower than visual odometry. These low values are a direct result of high sensitivity by choosing a low M-distance threshold. Figure 3.7 shows an example of a photo containing FOD, with a gray screwdriver at the center of the image, and an example of a photo with no FOD, which is a false positive detection due to the noise associated with the I-beam structure.

The effect of high sensitivity, or recall, is shown in Table 3.2. The first column shows the total number of FODs detected by the photo set, which is the number of FODs present in a session and captured in at least one photo. This differs from the photo containing FOD in that only distinct FODs are counted here. The second column shows the number of missing

Table 3.1: Relevance of Candidate FOD Photos in a Simulation Study

	Contains FOD	**No FOD**	**Photo Precision**
Wheel Odometry	64	104	0.381
Visual Odometry	77	109	0.414



True positive case where a screwdriver is correctly imaged as a candidate FOD

False positive case where an image is taken even though no FOD is present

Figure 3.7: Examples of candidate FOD photos, taken by the inspection robot, that are shown to remote humans for final decision-making.

FODs, which are FODs present in a session but not seen in any of the photos. The detection rates are high as previously discussed, with both the odometry methods having recall greater than 90%. This suggests that if the human are able to spot all the FODs in the photos, over 90% of the FODs can be detected and removed afterward.

The accuracy of identifying the exact FOD type comes out to be 0.860. The confusion matrix for exact FOD labeling is reported in Table 3.3. Unlike standard confusion matrices, the labels contain a “not sure” option for participants that find a FOD but cannot identify it. We also have a “mixed FOD” column to include photos with multiple detected FODs and multiple mislabeled FODs, since it is not possible to deduce which FOD is causing the confusion. As before, the matrix shows a high labeling accuracy with the maximum values

Table 3.2: FOD Detection Rate in a Simulation Study

	Detected FOD	**Missed FOD**	**FOD Recall**
Wheel Odometry	51	3	0.944
Visual Odometry	48	4	0.923

in the diagonal entries.

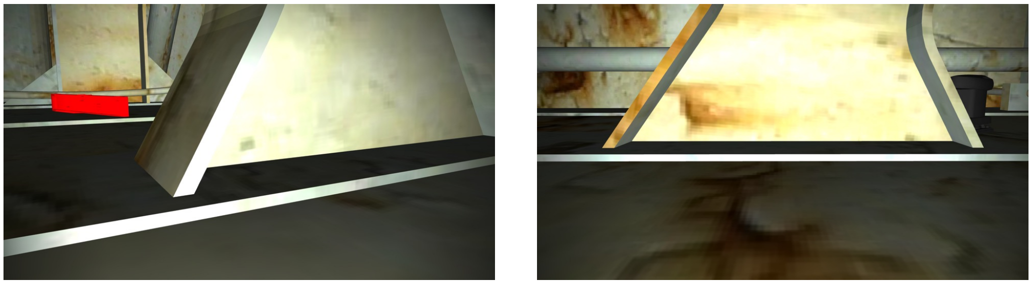
Table 3.3: Confusion Matrix for Human Labeled FOD Photos over 30 Simulation Trials

		Actual Type							
		Hammer	**Level**	**Screwdriver**	**Wrench**	**Sander**	**Drill**	**No FOD**	**Mixed FOD**
Labeled Type	**Hammer**	6	0	0	0	0	0	0	1
	Level	0	17	0	0	0	0	0	0
	Screwdriver	0	0	33	0	0	0	0	0
	Wrench	0	0	0	20	0	0	0	0
	Sander	0	0	0	0	2	0	0	0
	Drill	0	0	0	0	0	11	0	0
	No FOD	0	0	1	4	6	0	163	0
	Not Sure	2	13	0	8	1	2	3	0

As reported in Table 3.3, the spirit level is the most confused FOD, which is likely due to a low-quality rendering of the level. Figure 3.8 shows an example of such a confusing photo, with the level colored uniformly red. While the color red is easy to spot in the photos, the detailed shape is hard to see with uniform coloring, which makes the level appear as a red bar from a distance. On the other hand, the item most commonly mislabeled as “No fod” is the sander. Similar to the spirit level, the sander has a less distinct outline as compared to the other FODs. However, unlike the bright red color of the level, the sander is colored dark gray, as shown in Figure 3.8, which makes it similar to the hue of the rusty tank texture.

Hence, it is quite easy for the participants to completely miss the sander.

These rendering problems are, however, unique to simulation and should not be a concern for real-world deployment, as evident from the physical trials results reported next. More importantly, we observe promising results on detecting FODs using just visual odometry, where both precision and recall are almost identical to that using accurate wheel odometry. Consequently, we choose visual odometry during the physical trials to demonstrate the potential of our detection approach even for non-wheeled robots.



“Not sure” case where the humans are not confident about which FOD is present in the photo (a level tool is rendered in red)

Most mislabeled case where the humans indicate “No FOD” even though a dark brown sander is present behind the beam

Figure 3.8: Candidate FOD photos causing maximum confusion to the human subjects during online questionnaire survey

3.2.2 Physical Trials

Setup

A modified TurtleBot 3 Waffle Pi was used as the robot platform. Instead of using the baseline Raspberry Pi 3B+, an Nvidia Jetson AGX Xavier was added to the robot to provide on-board GPU capability. The default Pi camera was also replaced by a RealSense D435i depth camera. Two portable photography lights were mounted on top of the robot to reduce strong specular reflection from the ground. A scaled down prototype of the water tank was built from ply wood and painted with a white base with rusty brown spots to mimic the actual tank texture. Figure 3.9 shows an image of the FOD-free tank taken from the access hole with the robot at the starting location.

Figure 3.9: Nominal Physical Tank Configuration

All the computation was done in Python 3.7, with several methods parallelized using the CUDA library from Numba 0.53.1. The mean point cloud option was chosen for reference point cloud generation as the physical tank had significant variations from the CAD model. The voxel size of the mean point cloud and the occupancy threshold were chosen to be 0.05 m and 25^{th} percentile, respectively. For down-sampling, we used "voxel down-sample and trace" to handle stochasticity with a voxel size of 0.02 m. For smoothing, we employed the k-nearest neighbors (mean) method to avoid numerical instabilities from the estimated Gaussian weights for large distances and poor performance from searching spherical neighborhoods for highly variable density point clouds. k was selected as 250 for covariance smoothing and 50 for discrepancy smoothing. These parameters were manually tuned using the validation set discussed below. A grid search was used to obtain the threshold and cutoff parameters, which depended on the choice of the discrepancy metric, as discussed in the next section.

15 nominal training maps were collected to generate the mean point cloud and fit the local covariances. In addition, 15 validation maps with FODs were collected for parameters tuning. The FODs were randomly sampled from the following set: hammer, power drill, tape measure, screw driver, sander, and crimper. The FOD samples and their locations were generated as follows. First, the total number of FODs was randomly chosen between 2 to 5. The corresponding number of FODs were then uniformly sampled without replacement from the full FOD set. For each chosen FOD, the x, y location was uniformly sampled from the bounding box of the mean point cloud and z was fixed to the ground plane; if the location was sampled outside the tank, the FOD was placed at the closest interior location. Another 15 FOD-containing test maps were generated in a similar manner. The FOD centroids were adjusted according to their actual locations in the registered point clouds for use as ground truth values.

Parameter Search

The optimal values for the discrepancy threshold, clustering cutoff, and minimum point count were obtained using a grid search method over a set of 15 validation (mapping) trials. 30 cluster cutoff values were generated with a linear spacing from 0.05 to 1. 20 discrepancy values were generated, with the lower bound of the linear space as the minimum 25^{th} percentile of the discrepancies over all the 15 trials. The upper bound was manually chosen to be 3 and 0.05, respectively, for the M-distance and Euclidean distance methods. All the integers from 0 to 9 were used for the minimum point count. To quantify the performance, a cost function c for a single trial was devised as:

$$c = \frac{1}{n}\sum_{i=1}^{n} d_i + \lambda m. \tag{3.9}$$

Here, n is the actual number of FODs, d_i is the Euclidean distance from the i-th FOD centroid to its closest candidate centroid, m is the number of candidates that are not associated to any actual FOD, and λ is a positive weighting factor (chosen to be 0.05).

Note that multiple FODs can be associated to the same candidate if the distance between them is sufficiently low, as multiple FODs can be captured in the same picture taken from a particular candidate location. A representative example of multiple and missing actual-candidate FOD associations is shown in Figure 3.10. Here, the drill and the screw driver are sufficiently close to share a single candidate without incurring a high distance cost. The candidate at the center of the tank is, however, not associated with any FOD, which results in $m = 1$ for this trial. The closest candidate for the hammer is biased toward the high noise region near the column, which constitutes a large distance cost.

We compute the mean cost over all the trials for all possible combinations of the parameters, and select the set of parameter values resulting in the minimum cost for the M-distance and Euclidean distance methods separately. The M-distance method achieves a slightly lower cost of (0.493) than the Euclidean metric (0.508). For the M-distance method, the selected threshold, clustering cutoff, and minimum point count values are 2.75, 0.345 m, and 0, respectively. For the Euclidean method, the corresponding values are 0.030 m, 0.279 m, and 4, respectively.

Evaluation of FOD Candidate Generation Methods

The results for FOD candidate generation in a set of 15 test trials are shown in Table 3.4. We observe that the total number of unassociated candidate points, prior to clustering, is much smaller for the M-distance method as compared to its Euclidean distance counterpart. In fact, a Wilcoxon signed-rank test on the two test results samples yields a single tailed p-value of 0.014. This indicates that the M-distance method has significantly better noise reduction capability at the conventional $\alpha = 0.05$ significance level.

Correspondingly, on an average, the M-distance method also performs better than the Euclidean distance method in terms of the number of unassociated candidates (after clustering) per trial. This indicates that the former method is expected to be more precise, i.e., avoid false positive during FOD detection. However, both the methods have quite large

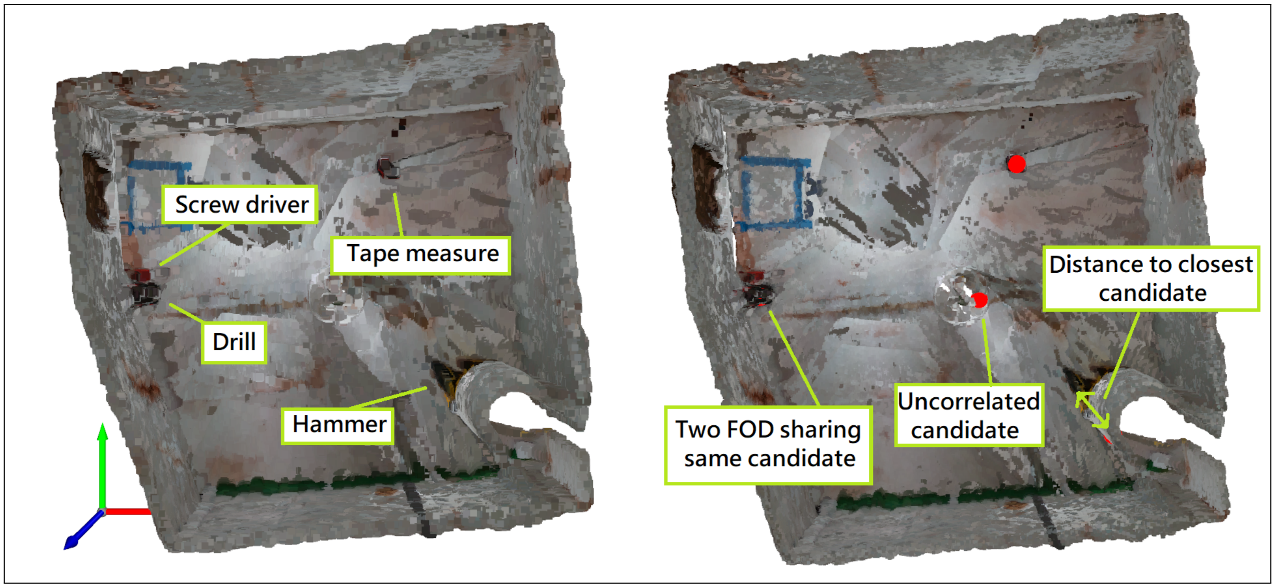


Figure 3.10: A representative example from the physical trials illustrating the associations between the actual and candidate FOD locations. The true FOD locations are shown in the denoised point cloud (left) and the candidates generated using the M-distance method are shown as red dots (right).

standard deviations. A direct pairwise comparison shows that the M-distance method yields a smaller number of unassociated candidates in 10 of the 15 trials. It, however, performs particularly badly in 2 of the trials, as discussed below with examples, thereby causing a large standard deviation. The Wilcoxon signed-rank test, with a resultant single tailed p-value of 0.176, shows that the overall performance difference is substantial, even though it is not statistically significant. It is useful to mention here that the test is under-powered due to the small sample size and large variance.

The two methods are equivalent with respect to the average distance of the actual FODs from their associated candidate FODs. This is also borne out by the Wilcoxon test that yields a single tailed p value of 0.56. Overall, these results suggest that the M-distance method is likely to reduce the number of false positive FOD detections as compared to the baseline Euclidean method, without causing any appreciable decrease in the quality of the FOD photos.

Table 3.4: Performance Comparison over 15 Physical Test Trials with a Total of 54 FODs

	Total Candidates	**Unassociated Points per Trial**	**Unassociated Candidates per Trial**	**Distance Error per FOD (in m)**
M-Distance	72	127.2 ± 152.9	2.2 ± 1.56	0.372 ± 0.348
L2 Distance	78	269.4 ± 312.5	2.6 ± 1.02	0.373 ± 0.328

Figure 3.11 shows selected samples from the test set. Each row comes from a trial and column 1 shows the point cloud with the true FOD location labeled. Columns 2 and 3 show the candidate clusters generated by the M-distance and Euclidean method, respectively. For all the three trials, the M-distance method shows a lower noise level based on the number of unassociated candidate points. Especially for row 1, it successfully eliminates two noise clusters, one on the left and the other behind the column, such that the candidate centroid is closer to the true FOD location. For the trial in row 2, while the column noise is vastly reduced by the M-distance method, the remaining noise is still large enough to form a false

cluster. However, it is able to eliminate one noise cluster at the lower left wall that is present in the Euclidean method, while both the methods miss one FOD on the top left corner. Row 3 shows a trial with substantial localization errors due to a misplaced top wall and large noise on the ground. As expected, both the methods perform quite poorly in this case. Although the M-distance method reduces the noise volume, such reduction causes the noisy points to be more disjoint, which inadvertently leads to more candidate clusters than the Euclidean method. This observation explains why the M-distance method has more unassociated candidates than the Euclidean method in a few trials.

FOD Detection Performance

The test set was further processed along the FOD identification pipeline, where the cost maps and corresponding waypoints were generated for each trial based on the clusters obtained from the M-distance method. Subsequently, the robot took a photo inside the tank at each location specified by the waypoints to capture the associated FOD candidates.

The detection rate of the photo set is shown in Table 3.5. Similar to the simulation study, a FOD is considered detected if any portion of the FOD appears in the photo set of a trial. Both the drill and the sander have a perfect recall rate, most likely due to their large heights that cause substantial deviations from the ground plane. On the other hand, the screw driver has the lowest recall, followed by the crimper. The screw driver has the least volume among all the FOD types, which makes it especially challenging to detect. While the crimper has a relatively large volume, it is mostly planar with a height similar to the screw driver when laid on the ground, thereby yielding small deviation values.

The overall recall rate is found to be 0.77. The relevancy of the photo set is also analyzed similar to the simulation study. Out of the 72 photos taken by the robot from the 15 trials, 40 contain one or more FODs and 32 contain no FOD. This constitutes a precision of 0.56. Thus, we note that the precision is higher than that for simulation, while the recall is lower. This happens potentially due to our choice of $\lambda = 0.05$ in the grid search cost function, which results in a threshold biased toward reducing the number of irrelevant candidates rather than

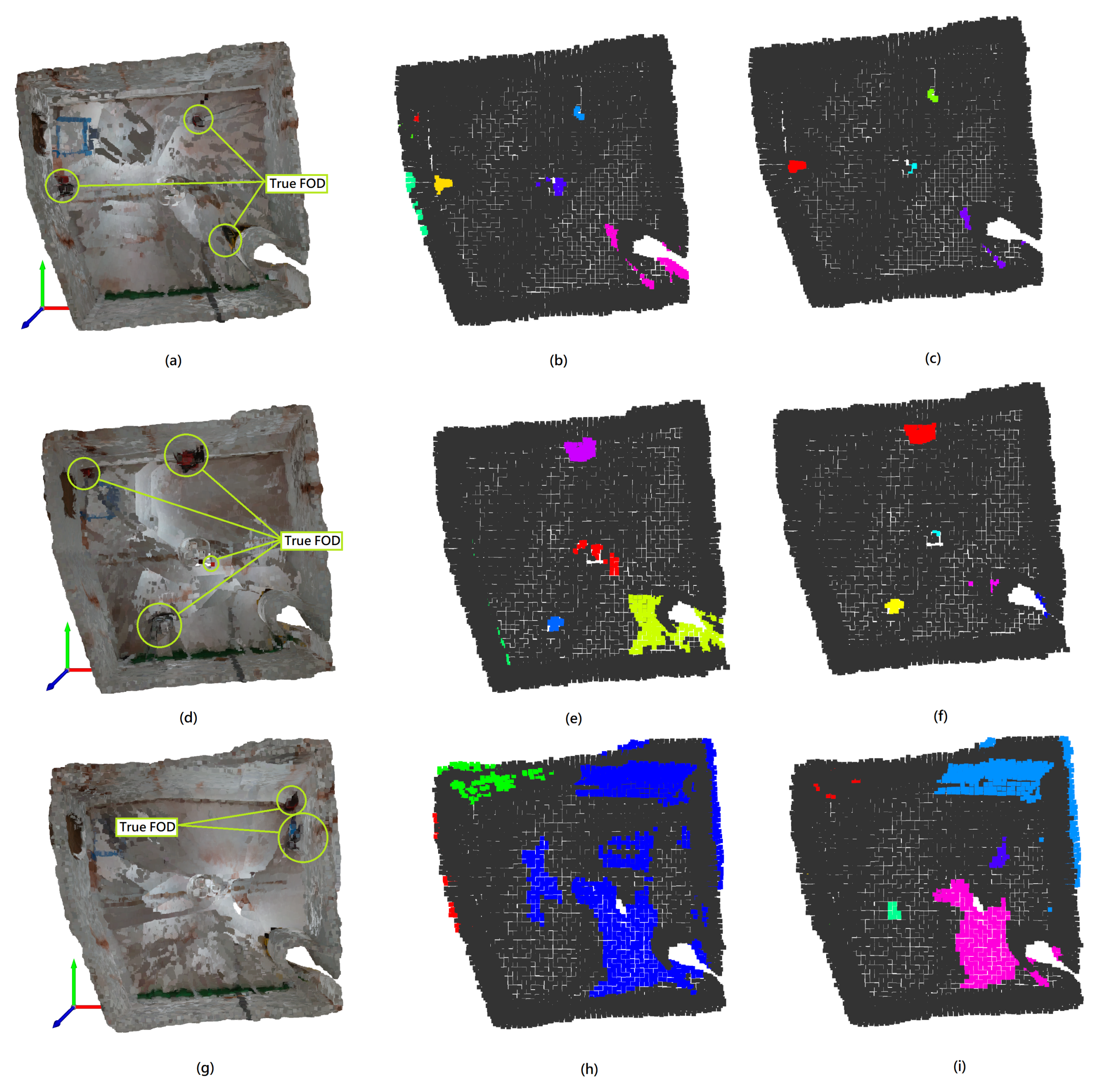


Figure 3.11: Selected examples from physical test trials illustrating the difference in performance between the Euclidean and M-distance methods. Each row represents a single trial. Column 1 shows the denoised point cloud with circled FOD locations; column 2 shows the candidate FOD clusters generated by the Euclidean method, with each cluster highlighted using a different hue; column 3 shows the corresponding clusters generated by the M-distance method. The M-distance method yields substantially less number of candidate points that are not associated with any FOD, and reduces the number of candidate clusters in the first two rows.

Table 3.5: FOD Detection Rates on Physical Test Trials

Type	**Detected**	**Missed**	**Recall**
Drill	9	0	1
Sander	5	0	1
Screw Driver	8	6	0.57
Hammer	7	2	0.78
Crimper	5	2	0.71
Tape Measure	8	2	0.8
Total	42	12	0.77

yielding candidates in close proximity to the true FOD locations.

The resultant FOD photos were assembled into an online anonymous questionnaire survey. Unlike the simulation study, all the trials are included in a single questionnaire and each participant, therefore, labeled the photos from all the trials. A total of 12 responses were received from undergraduate and graduate students, with an age range from 18-30. 83.3% of the responders were males and 16.7% were females. The results are compiled into a confusion matrix and reported in Table 3.6.

As expected, the human labeling performance for actual photos taken during physical trials is highly accurate (0.981), and better than that for simulation due to the absence of rendering issues. The most mislabeled item is the tape measure, which has only 2 instances each (out of 102) of being wrongly identified as a drill and screw driver, respectively. A possible cause for this mislabeling is that all these tools have the same black and red colored body. Just 1 photo containing FODs is mislabeled as having no FOD, where the FODs are far away from the robot. Only 2 photos with no FODs (out of 384) are labeled as “Not Sure”, where both are photos of the loose end of a hose behind the column. These results show that if any FOD is captured in a photo taken by the robot, it is highly likely that a (remotely located) human would be able to detect the FOD just by looking at the corresponding photo.

Table 3.6: Confusion Matrix for Human Labeled FOD Photos Taken over 15 Physical Test Trials ($n = 12$)

		Actual Type							
		Drill	**Sander**	**Screw Driver**	**Hammer**	**Crimper**	**Tape Measure**	**No FOD**	**Missed FOD**
Labeled Type	**Drill**	141	0	1	0	0	2	0	0
	Sander	0	82	0	0	0	0	0	0
	Screw Driver	1	0	123	0	0	2	0	0
	Hammer	1	0	0	108	0	0	0	0
	Crimper	0	2	0	0	68	0	0	0
	Tape Measure	0	0	1	0	0	98	0	0
	No FOD	0	0	0	0	0	0	382	1
	Not Sure	0	0	0	0	0	0	2	0

3.3 Discussion

There are a few limitations of our approach, as seen in the experimental results. The primary assumption of using M-distance is that the deviation from the reference model is Gaussian. However, there are multiple sources of non-Gaussian noise during the SLAM process. One source is localization uncertainty, which causes feature duplication at the erroneous robot location. This poses problem when using the M-distance as it treats the entire duplicate feature as deviation and causes false positives. Another problematic noise type is associated with the depth camera, which occurs around the edge of an object where the depth gradient is very high. Normally, the edge noises are removed during the SLAM process when the robot travels behind the obstacles. However, this does not happen for confined space structures such as the I-beam on the wall. These noise sources can be potentially dealt with by transitioning to non-Gaussian learning methods such as mixture models.

The clustering method also introduces an error in conjunction with this duplicated feature problem. When there is a large patch of false positive points, the hierarchical clustering algorithm sometimes clusters the true FOD points with the false positive points into one group. As a result, the center of the cluster mass meanders more toward the false positive, and sometimes causes the true positive to disappear from the camera's field of view. A work-around for this issue could be including the cluster size information during waypoint generation. Another method would be to include color information in the feature vector and perform a point cloud version of color image segmentation.

The training sample collection process can be tedious. If the mean point cloud is chosen as the reference, then the training sample is readily available and covariance fitting does not result in any extra work. However, if an accurate CAD model is used, collecting the training set only for covariance fitting may not be cost-efficient. This issue can be addressed by moving the learning process online, where a CAD model with Euclidean distance is used as the initial method. After each inspection session, a operator labels and removes all the FOD points, and the remaining points are used as the training samples, where the points in the CAD model are updated by the training points in a Gaussian mixture fashion.

Last but not the least, the trade-off between high FOD recall and photo precision is tricky, since the parameter search cost function does not directly consider the waypoint generation process and camera specifications. While the cost function provides a unified way of comparing different discrepancy metrics, hand tuning might be more advantageous for achieving a specific precision-recall rate.

Appendix 3.A Modified Rtap-Map Configuration

3.A.1 Global Parameters

- frame_id=base_footprint
- odom_frame_id=odom
- approx_rgbd_sync=false
- odom_tf_angular_variance=1

- odom_tf_linear_variance=1

3.A.2 Nodes Parameters

- Node Name: map_assembler
    - cloud_output_voxelized=false
    - Grid/RangeMax=0.5
    - Grid/cloud_subtrack_filtering_min_neighbors=10
    - Grid/DepthDecimation=1
- Node Name: map_optimizer
    - publish_tf=false
    - odom_frame_id=$(arg odom_frame_id)
- Node Name: rtabmap
    - grid_map=/map
    - Grid/FromDepth=true
    - Grid/CellSize=0.01
    - Grid/DepthDecimation=1
    - Grid/MaxGroundHeight=0.05
    - Grid/MaxObstacleHeight=1

Chapter 4

ERGODIC GRAPH TRAVERSAL VIA MARKOV CHAIN

Toward efficient, scalable, and robust inspection, the ballast tank is segmented into regions and the regions are arranged into a graph by connectivity. The goal is to traverse in the region graph such that the relative frequency of the robot visiting each region is equal to the relative information of that region. The main contribution is to achieve such goal by the use of ergodic Markov chain, specifically, this section shows that the optimal symmetric Markov chain can be obtained by convex optimization; and the convergence rate for general Markov chain can be upper bounded with convex optimization. This chapter is based on our published work in [115].

4.1 Preliminaries

4.1.1 Hierarchical Planning

For scalable and interpretable inspection targeted toward anomaly (FOD) detection, planning is decomposed into several levels, as shown in Fig. 4.1. The robot first decides which confined space region to go to according to coarse-grained information μ_r on anomaly uncertainty, defined on a region graph $\mathcal{G}$ that is stated formally below. The robot then selects a predefined number of poses in the underlying pose space $\mathcal{X}$ corresponding to the selected region. This is done according to fine-grained information on anomaly uncertainty in the part of the region that is visible to the robot from each pose. The selected poses are used as waypoints to move the robot to region-wise inspection points based on the robot operating system (ROS) navigation stack [76]. Figure 4.3 shows an example of the pose space $\mathcal{X}$ and the region graph $\mathcal{G}$ of the ballast tank in Fig. 4.2. A graph-based ergodic traversal method is chosen as the region planner, which is discussed in Section 4.3. The estimator is a SLAM-based anomaly detector, whose formulation is discussed in Appendix 4.A. The waypoint placement algorithm is presented in Section 4.4.

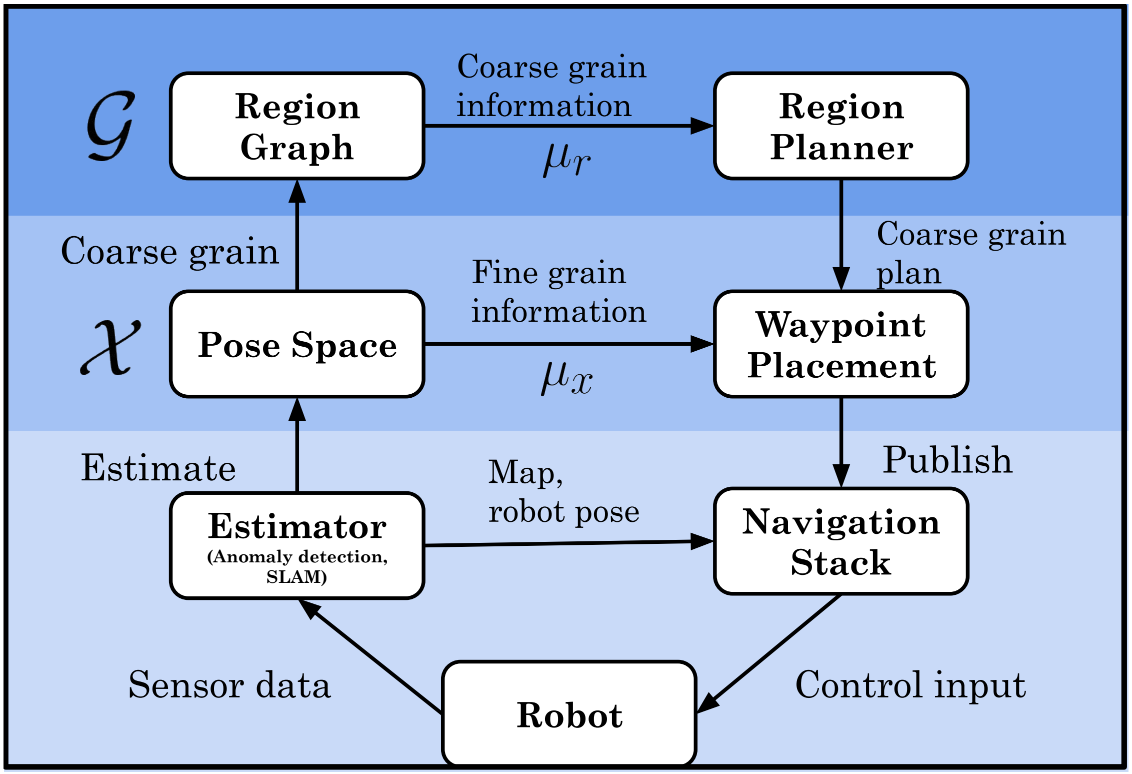


Figure 4.1: Hierarchical Ergodic Markov Planning (HEMaP) framework. At the top level, the graph-level region planner uses coarse-grained information μ_r on the current uncertainty estimates of region-wise anomalies. At the middle-level, fine-grained information on the anomaly uncertainty within the region μ_x is used for waypoint placement inside the region. At the bottom level, robot state estimates and occupancy grid map are provided as inputs for waypoint navigation.

Figures/cad_occupancy.pdf

Figure 4.2: CAD model of an example ballast tank with the roof removed and the corresponding occupancy grid map rendered above. The white grid cells denote feasible 2D poses in the pose space, $\mathcal{X}$, that can be occupied by a ground robot without colliding with the internal structures.

Figures/h_graph.pdf

Figure 4.3: Hierarchical representation of the ballast tank. Top-level is the workspace node, second-level is the region graph $\mathcal{G}$, the lowest level is the pose space $\mathcal{X}$ from Fig. 4.2.

4.1.2 Hierarchical Representation of Workspace

We consider the set of regions in the workspace as an arbitrary coarse graining of the pose space $\mathcal{X} \subset \mathrm{SE}(3)$ into a finite disjoint set $\mathcal{R} = \{r_1, r_2, \cdots, r_n\}$, where the space within each region is connected.

Definition 4.1.1. *Region nodes:*

$$\begin{aligned} r_i \subset \mathcal{X} &\quad \forall r_i \in \mathcal{R} \\ r_i \text{ is connected} &\quad \forall r_i \in \mathcal{R} \\ r_i \cap r_j = \varnothing &\quad \forall r_i, r_j \in \mathcal{R}. \end{aligned} \tag{4.1}$$

For any pair of regions (nodes), if the union is path connected, we denote an edge between the regions.

Definition 4.1.2. *Region edge: If $r_i \cup r_j$ is connected, $(r_i, r_j) \in \mathcal{E}$.*

The graph $\mathcal{G} = (\mathcal{R}, \mathcal{E})$ forms a region graph of the workspace. Additionally, we require the region graph to be strongly connected such that the robot can travel from any region to any other region. Otherwise, each strongly connected component in the graph would be considered a separated workspace. An example of the pose space and the associated graph representation of the ballast tank in Fig. 4.2 is shown in Fig. 4.3.

4.1.3 Continuous Ergodic System

A system is ergodic when the time average of a dynamic system is equal to the space average over the set the system is operating on, as formally defined below.

Definition 4.1.3. *Ergodic System*

The system $(\mathcal{X}, \Sigma, \mu, \Phi)$ is ergodic in a μ-measurable function F if

$$\lim_{T \to \infty} \frac{1}{T} \int_{t=0}^{\mathsf{T}} F(\mathbf{x}(t))dt = \frac{1}{\mu(\mathcal{X})} \int_{\mathcal{X}} F(\mathbf{x})\mu(dx) \tag{4.2}$$

for almost all initial conditions $\mathbf{x}(0)$. Here, the right-hand side is the space average of F over all the elements $\mathbf{x}$ in the set $\mathcal{X}$ with respect to a measure μ over the sigma algebra Σ.

The left-hand side is the (long-term) time average of F *evaluated along the trajectory* $\mathbf{x}(t)$ *generated by an evolution function* $\mathbf{x}(t) = \Phi(t, \mathbf{x}(0)) : (\mathcal{T} \times \mathcal{X}) \rightarrow \mathcal{X}$ *with initial state* $\mathbf{x}(0)$.

4.1.4 Discrete Ergodic System

The following is a similar ergodic system on a graph representation of the space in discrete time.

Definition 4.1.4. *Graph-based Ergodic System*

The system $(\mathcal{G}, \Sigma, \mu, M)$ *is ergodic in a* μ*-measurable function* F *if*

$$\lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} (F(r[k])) = \frac{1}{\mu(\mathcal{R})} \sum_{i=0}^{n} F(r_i)\mu(r_i) \tag{4.3}$$

for all initial conditions $r[0]$*. Here, the right-hand side is the space average of* F *over all elements* r *in the set* $\mathcal{R}$ *with respect to a measure* μ *over the sigma algebra* Σ*. The left-hand side is the (long-term) time average of* F *evaluated along the sequence* $r[k]$ *generated by a dynamical map* $M : \mathcal{R} \rightarrow \mathcal{R}$

$$r[k+1] = M(r[k]) \tag{4.4}$$

with initial state $r[0]$.

4.2 Problem Formulation

For effective and robust inspections, we borrow the notion of ergodic exploration for the region planner at the top level of the hierarchical planning framework in Fig. 4.1. In particular, the relative frequency of the robot visiting each region (i.e., the time average $\hat{\rho}$) is desired to be equal to the relative available information of each region on the graph (i.e., the space average $\bar{\rho}$).

The visitation-based ergodicity can be defined by selecting the measurable function F in

Definition 4.1.4 as the indicator function I

$$I(r_i) \triangleq \begin{bmatrix} \delta_{1,i} \\ \delta_{2,i} \\ \vdots \\ \delta_{n,i} \end{bmatrix} \tag{4.5}$$

with $\delta_{i,j}$ representing the Kronecker delta. Here,

$$\hat{\rho} \triangleq \lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} \left(I(r[k]) \right) \tag{4.6}$$

$$\bar{\rho} \triangleq \frac{1}{\mu(\mathcal{R})} \sum_{r_i \in \mathcal{R}} I(r_i)\mu(r_i). \tag{4.7}$$

The following two subproblems are considered in this work. Subproblem 1: How to synthesize the dynamical map M in (4.4) such that the region sequence $r[\cdot]$ is ergodic in I while respecting graph-connectivity constraints? Subproblem 2: How to optimize M such that the difference between the time average $\hat{\rho}$ in (4.6) and space average $\bar{\rho}$ in (4.7) is minimized for a given finite-time horizon K?

4.3 Graph Ergodic Traversal

We present here that graph ergodic exploration can be achieved by sampling from an ergodic Markov chain. Further, we show that the ergodicity of such a Markov chain can be improved by convex optimization.

4.3.1 Asymptotic Ergodicity

We propose to achieve graph-based ergodicity by using a *Markov chain*, where the transition probability between regions is determined by a stochastic matrix P with the stationary distribution equal to the desired distribution. Then, at the limit, the space average is almost surely equal to the desired distribution based on the ergodic theorem of Markov chains [18, Theorem 1.10.2].

Specifically, consider a linear stochastic dynamical map M, where the transition probability $P_{j,i}$ of the robot from region r_i to r_j forms a stochastic matrix $P \in \mathbb{R}^{n \times n}$. Then, asymptotic ergodicity, in the expected sense, is achieved for any stochastic matrix P that is irreducible, i.e., can go from any region to any other region in the graph $\mathcal{G}$ (not necessarily in one move), with the space average $\bar{\rho}$ as the unique stationary distribution, as shown below.

Theorem 1 (Ergodic Theorem for Ergodic Graph Traversal)**.** *Let the sequence* $r[\cdot]$ *of regions of the strongly connected graph* $\mathcal{G}$ *be sampled according to an irreducible stochastic matrix* P*, i.e.,* $r[k+1] \sim \mathbb{P}(R_{k+1} | R_k = r[k])$*, with the expected value* $\rho_k = \mathbb{E}[I(r[k])]$ *satisfying*

$$\rho_{k+1} = P \rho_k \tag{4.8}$$

and an expected (infinite) time average

$$\mathbb{E}(\hat{\rho}) = \lim_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} \rho_k = \lim_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} \left(P^k \rho_0 \right). \tag{4.9}$$

Then, the stochastic system $(\mathcal{R}, \mathcal{B}, \mu, P)$ *is asymptotically ergodic in expected value, i.e.,*

$$\mathbb{E}[\hat{\rho}] = \bar{\rho} \tag{4.10}$$

for any initial distribution ρ_0*, if the space average* $\bar{\rho}$ *defined in* (4.7) *is a stationary distribution of the stochastic matrix* P*, i.e.,*

$$P \bar{\rho} = \bar{\rho}. \tag{4.11}$$

Proof. The proof follows the standard ergodic theorem of Markov chains [18, Theorem 1.10.2]

□

Remark 1. *The term ergodic Markov chain has multiple definitions in the literature, e.g., when the Markov chain is irreducible [90, p.245], meaning it can transition from any state to any other state in a finite amount of time; or when the Markov chain is both irreducible and aperiodic [81, p.316]. In this document, we adopt the definition most closely aligned with ergodic theory [18, Theorem 4.1], which requires irreducibility (but not aperiodicity) and further stipulates that the time average must equal a specific space average as in Definition (4.1.4).*

4.3.2 Time-Discounted Ergodic Objective

In contrast to the infinite time for asymptotic ergodicity, a time-discounted ergodicity is defined next to formalize the need for faster ergodicity in time.

Definition 4.3.1. *The stochastic system* $(\mathcal{R}, \mathcal{B}, \mu, P, w)$ *is discounted ergodic in expected value if*

$$\mathbb{E}[\hat{\rho}_w] = \bar{\rho} \tag{4.12}$$

where the left hand side is the expected discounted time average of the sequence of regions $r[\cdot]$

$$\mathbb{E}[\hat{\rho}_w] \triangleq \lim_{K\to\infty} \frac{1}{\sum_{k=0}^{K-1} w_k} \left(\sum_{k=0}^{K-1} w_k P^k \rho_0 \right). \tag{4.13}$$

and $w = (w_0, w_1, \cdots, w_k, \cdots)$ *is a sequence of weights. Moreover, the discounted time average operator* f_w *associated with the weight* w *is defined as*

$$f_w(x) \triangleq \lim_{K\to\infty} \frac{1}{\sum_{k=0}^{K-1} w_k} \left(\sum_{k=0}^{K-1} w_k x^k \right). \tag{4.14}$$

Remark 2 (Factorial discounting)**.** *An example of time weighting is a factorial discounting over time,*

$$w_k = 1/k! \tag{4.15}$$

with the associated discounted-time-average operator in (4.14)

$$f_w(x) \triangleq \lim_{K\to\infty} \frac{1}{\sum_{k=0}^{K-1} 1/k!} \left(\sum_{k=0}^{K-1} \frac{1}{k!} x^k \right) = e^{-1} e^{x}. \tag{4.16}$$

In general, discounted ergodicity (4.12) cannot be achieved for an arbitrary initial condition. Instead, the objective is to select the stochastic matrix P optimally to reduce the difference between the expected discounted-time-average $\mathbb{E}[\hat{\rho}_w]$ in (4.13) and the space average $\bar{\rho}$ in the worst case initial distribution ρ_0, normalized over the initial difference. In other words, we want to minimize the normalized ergodicity deviation (NED)

$$J_{\mathbb{E}} = \max_{\rho_0} \frac{\|Q^{1/2} \left(\mathbb{E}[\hat{\rho}_w] - \bar{\rho}\right) \|_2}{\|Q^{1/2} \left(\rho_0 - \bar{\rho}\right) \|_2}, \tag{4.17}$$

where Q is a given weighting matrix.

4.3.3 Symmetric Stochastic Matrix

The following shows that for symmetric stochastic matrices P, time-discounted ergodicity can be exactly optimized by minimizing the second largest eigenvalue of P. This is in contrast to minimizing the second largest eigenvalue modulus (SLEM) of P, which would speed up convergence to the desired stationary distribution, e.g., as in [16].

The theorem below shows that for symmetric stochastic matrices P, under conditions typical for discount factors on the weighting scheme w, optimal time-discounted ergodicity (in the slowest converging direction) can be achieved by minimizing the second largest eigenvalue of P.

Theorem 2. *Let the stochastic matrix P be symmetric and irreducible. Moreover, let the discounted-time-average operator f_w in* (4.14) *be positive semi-definite and strictly monotonic in $[-1, 1]$ namely $f_w(\lambda_1) > f_w(\lambda_2) \geq 0$ if $\lambda_1 > \lambda_2$ for all $\lambda_1, \lambda_2 \in [-1, 1]$. Then, the normalized ergodicity deviation in* (4.17) *with weighting matrix $Q = I$ is minimized for the stationary distribution $\bar{\rho} = \frac{1}{n}\boldsymbol{1}$, if and only if the second largest eigenvalue (SLE) is minimized, i.e.,*

$$\arg\min_{P} \lambda_2(P) \tag{4.18}$$

where $1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \cdots \lambda_n$.

Proof. See Appendix 4.B ☐

Remark 3 (Uniform weighting)**.** *The discounted-time-average operator f_w in* (4.14) *does not satisfy the strictly monotonic condition in Theorem 2 with uniform weighting $w = (1, 1, 1, \cdots)$, since $f_w(\lambda) = 0$ if $\lambda \in [-1, 1)$ and $f_w(\lambda) = 1$ if $\lambda = 1$. Therefore, for any symmetric and irreducible stochastic matrix P satisfying $P\bar{\rho} = \bar{\rho}$ in* (4.11) *with $\bar{\rho} = \frac{1}{n}\boldsymbol{1}$, the ergodic deviation becomes zero from* (4.64) *, i.e.,*

$$\|\mathbb{E}[\hat{\rho}_w] - \bar{\rho}\|_2 = \left\| \sum_{i=2}^{n} f_w(\lambda_i) v_i u_i^{\mathsf{T}} \rho_0 \right\|_2 = 0. \tag{4.19}$$

Thus, the uniformly-weighted ergodicity deviation in (4.17) *is minimized with any symmetric and irreducible stochastic matrix P, as in Theorem 1.*

The symmetric optimal stochastic matrix P, that minimizes the normalized ergodicity deviation as in Theorem 2, can be found by solving a convex optimization problem, as shown in the following corollary.

Corollary 2.1. *The optimal symmetric stochastic matrix P in Theorem 2, with the minimum, second largest eigenvalue, can be obtained by solving the following convex optimization problem.*

$$\begin{aligned}
&\arg\min_{p} \quad \lambda_{\max}\left(P - \frac{2}{n}\mathbf{1}\mathbf{1}^{\mathsf{T}}\right) \\
&\text{s.t.} \\
&\mathbf{1}^{\mathsf{T}}P = \mathbf{1}^{\mathsf{T}} \quad \text{(Stochastic } P) \\
&P_{i,j} \geq 0 \quad \text{(Stochastic } P) \\
&P = P^{\mathsf{T}} \quad \text{(Symmetric } P) \\
&P_{i,j} = 0 \ \text{ if } \ (j,i) \notin \mathcal{E} \quad \text{(Transitions in } \mathcal{G}).
\end{aligned} \tag{4.20}$$

Proof. The optimization follows since the second largest eigenvalue $\max_{i \neq 1} \lambda_i(P)$ is the largest eigenvalue of $P - \frac{2}{n}\mathbf{1}\mathbf{1}^{\mathsf{T}}$, which is a Wielandt deflation [1] of the largest eigenvalue λ_1 of P from $\lambda_1 = 1$ to -1 (the smallest possible eigenvalue of a symmetric stochastic matrix), i.e.,

$$\begin{aligned}
\lambda(P - \frac{2}{n}\mathbf{1}\mathbf{1}^{\mathsf{T}}) &= \{\lambda_1 - 2, \lambda_2, \lambda_3, \cdots\} \\
&= \{-1, \lambda_2, \lambda_3, \cdots\} \\
\Rightarrow \lambda_{\max}(P - \frac{2}{n}\mathbf{1}\mathbf{1}^{\mathsf{T}}) &= \max_{i \neq 1} \lambda_i(P) \ \text{ since } \lambda_i \geq -1.
\end{aligned} \tag{4.21}$$

The minimization of the largest eigenvalue of a symmetric matrix $(P - \frac{2}{n}\mathbf{1}\mathbf{1}^{\mathsf{T}})$ is convex [18, Example 3.10 p.82] and all the constraints are linear, and therefore (4.20) is a convex optimization problem. □

4.3.4 Nonuniform Target Distribution

Although the optimal symmetric stochastic matrix P with the minimum second largest eigenvalue exactly minimizes normalized ergodicity deviation, the only possible target distribution is the uniform distribution $\bar{\rho} = \frac{1}{n}\mathbf{1}$. For a general target distribution $\bar{\rho}$ with no zero entries, if the detailed-balance condition is satisfied, i.e.,

$$P\Pi = \Pi P^{\mathsf{T}}, \quad \Pi \triangleq \operatorname{diag}(\bar{\rho}), \tag{4.22}$$

it is shown that the stochastic matrix P with the minimum second largest eigenvalue minimizes the normalized ergodicity deviation with the weighting matrix Q chosen to be inversely proportional to the target distribution $\bar{\rho}$. In other words, $Q = \Pi^{-1}$, which can be interpreted as normalizing the error vector by the size of the target distribution.

Remark 4 (Detailed balance usage)**.** *The detailed balance condition in* (4.22)*, i.e., that the Markov chain being reversible, has been applied to generate a Markov chain with the desired target distribution in the Metropolis-Hasting algorithm [49], and subsequently required to select the optimal stochastic matrix P for the fastest mixing Markov chain in [16].*

Corollary 2.2. *Selecting the stochastic matrix P to have the smallest, second largest eigenvalue as in* (4.18) *minimizes the normalized ergodicity deviation in* (4.17) *with weighting $Q = \Pi^{-1}$ for a given target distribution $\bar{\rho}$ with no zero entries. Additionally, the optimal stochastic matrix P can be obtained by solving the following convex optimization problem*

$$\arg\min_{P} \quad \lambda_{max}(\Pi^{-1/2}(P - 2\bar{\rho}\boldsymbol{1}^{\mathsf{T}})\Pi^{1/2}) \tag{4.23}$$

with the same conditions as in (4.20) *but with the symmetry $P = P^{\mathsf{T}}$ condition replaced by the detailed-balance condition in* (4.22).

Proof. See Appendix 4.D □

While the discounted time-average used in Corollary 2.2 is not the actual time average with the addition of the discounting weights, the discounted time-average can be made equal to the actual finite time average using unity weights. Specifically, the following lemma shows that any ergodic process generated by Corollary 2.2 is optimally (undiscounted) ergodic if the process is terminated at an even timestep.

Lemma 3 (Finite even-time horizon ergodicity)**.** *Consider the finite horizon time average*

$$\hat{\rho}_K \triangleq \frac{1}{K} \sum_{k=0}^{K-1} P^k \rho_0, \tag{4.24}$$

which can be considered as a discounted time-average with the weighting function

$$w_k = \begin{cases} 1 & \text{if } k < K \\ 0 & \text{if } k \geq K. \end{cases} \tag{4.25}$$

Then, the trajectory $\rho[\cdot]$ produced by the optimal P obtained from (4.23) *is optimally ergodic toward $\bar{\rho}$ for even K, i.e., $K \in \{2, 4, 6 \cdots\}$.*

Proof. See Appendix 4.E. □

4.3.5 Convex Bounding for General Markov Chain

The detailed balance condition in the previous subsection can be restrictive for ergodicity. An obvious example is when transitions are allowed in only one direction between some regions, because the detailed balance condition leads to zero probability even for the available transition. Thus, in such one-way transition cases, the detailed balance condition results in a larger normalized ergodicity deviation compared to the case without the condition, as illustrated in Fig. 4.4.

The detailed balance requirement can be relaxed for general directed graphs (e.g., with potential one-way transitions), by optimizing an upper bound on the normalized ergodicity deviation, as shown in the theorem below.

Theorem 4. *The normalized ergodicity deviation in* (4.17) *with the same weighting $Q = \Pi^{-1}$ as in Corollary 2.2 and a factorial weighting $w_k = \frac{1}{k!}$ in* (4.15) *can be bounded as*

$$\begin{aligned} \max_{\rho_0} & \frac{\|\Pi^{-1/2} \left(\mathbb{E}[\hat{\rho}_w] - \bar{\rho}\right) \|_2}{\|\Pi^{-1/2} \left(\rho_0 - \bar{\rho}\right) \|_2} \\ & \leq \max_{i \neq 1} \lambda_i \left(e^{-1} \exp \left(\frac{1}{2} \left(\tilde{P} + \tilde{P}^{\mathsf{T}} \right) \right) \right) \end{aligned} \tag{4.26}$$

`Figures/Detailed_balance_comp.pdf`

Figure 4.4: Example three-node-graph with one directed edge with the uniform final distribution $\bar{\rho}$ and weighting $w_k = 1/k!$ illustrating the detailed balance condition can be detrimental to ergodicity. When detailed balance is enforced, the REMC stochastic matrix has a zero transition probability on the one-way directed edge, resulting in a normalized ergodicity deviation (4.17) of $J_{\mathbb{E}} = 0.606$ (left), which is substantially higher than $J_{\mathbb{E}} = 0.223$ for the case when detailed balance is not enforced (right).

where $\tilde{P} = \Pi^{-1/2} P \Pi^{1/2}$*, as in* (4.79)*. Moreover, the bound in* (4.26) *can be minimized by reducing the second largest eigenvalue of the symmetric part of* $\tilde{P}$

$$\arg\min_{P} \left(\max_{i\neq 1} \lambda_i \left(\frac{1}{2} \left(\tilde{P} + \tilde{P}^{\mathsf{T}} \right) \right) \right) \tag{4.27}$$

Proof. See Appendix 4.F □

The following corollary states the convex optimization problem for Theorem 4, which is referred in the document as *REMC*.

Corollary 4.1 (Rapidly Ergodic Markov Chain (REMC))**.** *The optimal* P *for the upper-bound in Theorem* 4 *can be obtained by solving the convex optimization problem*

$$\begin{aligned} \arg\min_{P} \quad & \lambda_{\max} \left(\frac{1}{2} \left(\tilde{P} + \tilde{P}^{\mathsf{T}} \right) - 2 \bar{\rho}^{1/2} \bar{\rho}^{\mathsf{T}/2} \right) \\ \text{s.t.} \quad & \mathbf{1}^{\mathsf{T}} P = \mathbf{1}^{\mathsf{T}} \quad \text{(Stochastic } P\text{)} \\ & P \bar{\rho} = \bar{\rho} \quad \text{(Target Distribution)} \\ & P_{i,j} \geq 0 \quad \text{(Stochastic } P\text{)} \\ & P_{i,j} = 0 \quad \text{if} \quad (j,i) \notin \mathcal{E} \\ & \qquad\qquad \text{(Transitions in } \mathcal{G}\text{)} \end{aligned} \tag{4.28}$$

with $\bar{\rho}^{1/2}$ *being the entry-wise square root of* $\bar{\rho}$*.*

Proof. See Appendix 4.G. □

Since REMC is a stochastic method, the region trajectories can vary from each trials. Additional empirical testing of the variance of the trajectories generated by REMC is shown in Appendix 4.H.

4.3.6 Connection to Fastest Mixing Markov Chain

An alternative approach to speed-up ergodicity is to reach the desired stationary distribution $\bar{\rho}$ faster at every time step, consider some finite time horizon, K, i.e.,

$$\arg\min_{P} \max_{\rho_0} \sum_{k=0}^{K-1} \frac{\|\Pi^{-1/2}(P^k \rho_0 - \bar{\rho})\|_2}{\|\Pi^{-1/2}(\rho_0 - \bar{\rho})\|_2}. \tag{4.29}$$

Similar to (4.78), this can be reduced to

$$\begin{aligned}
&\arg\min_{P}\max_{\rho_0}\sum_{k=0}^{K-1}\frac{\|\Pi^{-1/2}(P^k\rho_0-\bar{\rho})\|_2}{\|\Pi^{-1/2}(\rho_0-\bar{\rho})\|_2}\\
&\quad=\arg\min_{P}\max_{\rho_0}\sum_{k=0}^{K-1}\frac{\|\Pi^{-1/2}(P|_{u_1^\perp})^k\rho_{0\perp}\|_2}{\|\Pi^{-1/2}\rho_{0\perp}\|_2}\\
&\quad=\arg\min_{P}\sum_{k=0}^{K-1}\|(\tilde{P}|_{u_1^\perp})^k\|_2.
\end{aligned}\tag{4.30}$$

If the detailed balance condition is enforced, then the 2-norm is equal to the eigenvalue with the largest absolute value, since

$$\begin{aligned}
&\arg\min_{P}\sum_{k=0}^{K}\left\|(\tilde{P}|_{u_1^\perp})^k\right\|_2\\
&\quad=\arg\min_{P}\sum_{k=0}^{K}\max_{i\neq 1}\left|\lambda_i(P)^k\right|\\
&\quad=\arg\min_{P}\left[\max_{i\neq 1}|\lambda_i(P)|\right].
\end{aligned}\tag{4.31}$$

This is equivalent to the fastest mixing Markov chain (FMMC) formulation, where the stochastic matrix P is selected such that the SLEM, $\max_{i\neq 1}|\lambda_i|$, is minimized. The FMMC formulation upper-bounds the REMC by ensuring fast convergence on the discounted-time-average, since

$$\begin{aligned}
\left\|\frac{1}{K}\sum_{k=0}^{K-1}\left(P^k\rho_0-\bar{\rho}\right)\right\| &= \frac{1}{K}\left\|\sum_{k=0}^{K-1}\left(P^k\rho_0-\bar{\rho}\right)\right\|\\
&\leq \frac{1}{K}\sum_{k=0}^{K-1}\left\|P^k\rho_0-\bar{\rho}\right\|.
\end{aligned}\tag{4.32}$$

The difference in ergodicity achieved with the FMMC formulation that minimizes the SLEM and the proposed REMC approach that minimizes the second largest eigenvalue is illustrated in the following 2-node example. An additional example with a 3-node graph is illustrated in Appendix 4.I.

Example 4.3.1. *Comparison of FMMC and REMC on a two node graph: FMMC aims to reach the target distribution as quickly as possible by minimizing the SLEM, while REMC* *aims to reach ergodicity as quickly as possible by minimizing the second largest eigenvalue. To illustrate the difference, both methods are applied to a two-node graph example (where all transitions are possible) for a uniform target distribution, and the resulting optimal Markov chains are shown in Fig. 4.5. The left-hand-side graph in Fig. 4.5 shows the result for FMMC, with the stochastic matrix*

$$P_{\text{FMMC}} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \tag{4.33}$$

which has a second eigenvalue of $\lambda_2 = 0$ *that is the smallest possible SLEM. For the same two-node graph, the right-hand-side graph in Fig. 4.5 shows the result for REMC, with the stochastic matrix*

$$P_{\text{REMC}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{4.34}$$

which has a second eigenvalue of $\lambda_2 = -1$ *that is the smallest possible second eigenvalue.*

With an initial distribution $\rho_0 = (1, 0)$*, the FMMC reaches the target distribution in one step, with the distribution* $\rho_k = P^k_{FMMC}\rho_0$ *at each time step k given by*

$$\{\rho_{\text{FMMC, k}}\} = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \cdots \right\}. \tag{4.35}$$

In contrast, the REMC is cyclic and never reaches the target distribution, and the distribution $\rho_k = P^k_{REMC}\rho_0$ *at each time step k is given by*

$$\{\rho_{\text{REMC,k}}\} = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdots \right\}. \tag{4.36}$$

However, although FMMC reaches the target uniform distribution fast (in one time step), the finite time average $\hat{\rho}_K$ *from* (4.24)

$$\{\hat{\rho}_{\text{FMMC},K}\} = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.75 \\ 0.25 \end{bmatrix}, \begin{bmatrix} 0.66 \\ 0.33 \end{bmatrix}, \begin{bmatrix} 0.625 \\ 0.375 \end{bmatrix} \cdots \right\} \tag{4.37}$$

Figures/two_node_sys.pdf

Figure 4.5: A two-node example illustrating the difference between fastest mixing (left) and optimally ergodic (right) for uniform target distribution.

does not converge as quickly due to the non-zero initial condition that decays slowly. On the other hand, the time average of REMC

$$\{\hat{\rho}_{\mathrm{REMC},K}\} = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 0.66 \\ 0.33 \end{bmatrix}, \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \cdots \right\} \tag{4.38}$$

is exactly the target distribution at every second step. Thus, when considering the time average, instead of having ρ_k reaching the target distribution quickly as in FMMC, the effects of the initial condition can be reduced more rapidly with REMC.

Thus, an ergodic graph traversal for the region planner in Fig. 4.1 can be achieved by sampling from a Markov chain, and a fast convergence ergodic Markov chain can be obtained for any target distribution by using (4.28).

4.4 Hierarchical Ergodic Inspection Algorithm

This section explains how graph ergodic planning is used within the hierarchical planning framework (as shown in Fig. 4.1), which we call hierarchical ergodic Markov planner (HEMaP), to inspect a confined structure such as a ballast tank using the anomaly detection method described in Appendix 4.A. The complete inspection planning algorithm is described in Algorithm 1. The algorithm requires the initial region of the robot $r[0]$; the region graph $\mathcal{G}$; the replanning horizon K; the reference point cloud $\mathcal{P}_{\text{ref}}$; and the initial anomaly entropy μ_p. The robot initializes the graph-level time step k to 0. Then, for every K steps, the robot calculates the region-based measure μ_r by summing the anomaly entropy of the structures in each region. Using the new region-based measure μ_r, the robot calculates the optimal stochastic transition matrix P according to the REMC optimization problem in (4.28). Additionally, to prevent an ill-conditioned target distribution with 0-entries, a smoothing factor δ is added before optimization. The robot samples from this P for the next K steps. After the next region is sampled, the robot picks a waypoint inside the next region according to the structure anomaly entropy in the region (described in the next algorithm), navigates to the waypoint, and updates the anomaly detection via (4.55)-(4.57). This process is repeated until the inspection is finished, either defined by a maximum number of steps or terminated by a human.

Algorithm 1 HEMaP algorithm

```
**Parameter:** number of waypoints per region: $n_{\text{waypoints}}$
**Inputs:** initial region: $r[0]$, region graph: $\mathcal{G}$, time horizon: $K$, reference point cloud: $\mathcal{P}_{\text{ref}}$, information measure: $\mu_p$
**Initialize** $k \leftarrow 0$
while Inspecting do
    if $k \mod K = 0$ then
        Initialize $\mu[n]$
        for $r_i \in \mathcal{R}$ do
            for $p_{ref} \in r_i$ do
                $\mu_r[i] \leftarrow \mu_r[i] + \mu_p(p_{ref})$
            end for
        end for
        $\bar{\rho} = \mu_r / \sum \mu_r$
        $\bar{\rho} = (\bar{\rho} + \delta) / \sum(\bar{\rho} + \delta)$
        $P \leftarrow \text{REMC}(\bar{\rho})$                                ▷ (4.28)
    end if
    $r[k+1] \sim \mathbb{P}(R_{k+1} | R_k = r[k])$
    for $i$ in $n_{\text{waypoint}}$ do
        $\mathbf{x} \leftarrow \text{waypoint\_placement}(r[k+1], \mu_p)$
        navigate_to($\mathbf{x}$)
        $\mu_p \leftarrow \text{anomaly\_detection}(\mu_p)$                  ▷ 4.A.2
    end for
    $k \leftarrow k + 1$
end while
```

Algorithm 2 waypoint_placement

```
Parameter: number of samples: $n_{\text{sample}}$
Inputs: region: $r$, feature information: $\mu_p$
Initialize $\mathbf{x}[n_{\text{sample}}], \mu_x[n_{\text{sample}}]$
for $i$ in $n_{\text{sample}}$ do
    $\mathbf{x}[i] \sim \text{uniform}(r)$        ▷ uniformly sample a pose from the region
    for $p_{\text{ref}} \in \mathcal{P}_{\text{ref}}$ do
        if $p_{\text{ref}} \in \text{camera\_frustum}(\mathbf{x}[i])$ then
            $\mu_x[i] \leftarrow \mu_x[i] + \mu_p(p_{\text{ref}})$
        end if
    end for
end for
$\bar{\mu}_x \leftarrow \mu_x / \sum \mu_x$
$\mathbf{x}^* \leftarrow \text{random\_choice}(\mathbf{x}, p = \bar{\mu}_x)$
Return $\mathbf{x}^*$
```

Algorithm 2 shows the procedure for choosing the waypoint in the designated region from the ergodic planner. A sampling algorithm is used to select the waypoint, where the number of candidate waypoints, n_{sample}, is determined based on the size of the region and the computing power of the robot. The robot samples n_{sample} waypoints from the designated region r. For each candidate waypoint, it then finds the reference points that fall within the robot's camera frustum, defined by the camera field of view and the minimum and maximum depths. The entropies of all the anomalies within the frustum are added together and stored as the waypoint measure. The final waypoint is then selected by randomly sampling from this candidate set, with the selection probability being proportional to the relative measure of each waypoint.

4.5 Experiments

In this section, four sets of experiments, with progressively increasing complexity, are conducted to demonstrate the advantages of REMC and HEMaP at various abstraction levels: 1) Graph world, 2) Grid world, 3) Gazebo simulation, 4) Real-world experiment. The first experiment shows the fast convergence to ergodicity of REMC by testing on a graph world. The second experiment applies HEMaP on a grid representation of the ballast tank and shows that the graph method can be applied to a continuous space partitioned into interconnected regions. In the third experiment, the method is applied in a Gazebo simulation of the ballast tank to study the effectiveness of ergodic graph traversal for FOD detection. In the final experiment, the inspection algorithm is implemented on a TurtleBot in a custom-built mock ballast tank to demonstrate real-world deployment capability. All convex optimization problems were solved using *cvxpy* [43, 27].

4.5.1 Graph World Experiment

The following experiment compares the performance between the ergodicity of the Markov chain generated by FMMC and REMC from Theorem 4 using the region graph in Fig. 4.3.

Methodology

The target distribution $\bar{\rho}$ and initial distribution ρ_0 are generated by sampling from the uniform Dirichlet distribution (i.e. with parameter $\alpha = \mathbf{1}_n$). The expected finite time ergodicity deviation, $\|\mathbb{E}[\hat{\rho}_K] - \bar{\rho}\|_2$, is calculated both for FMMC and REMC.

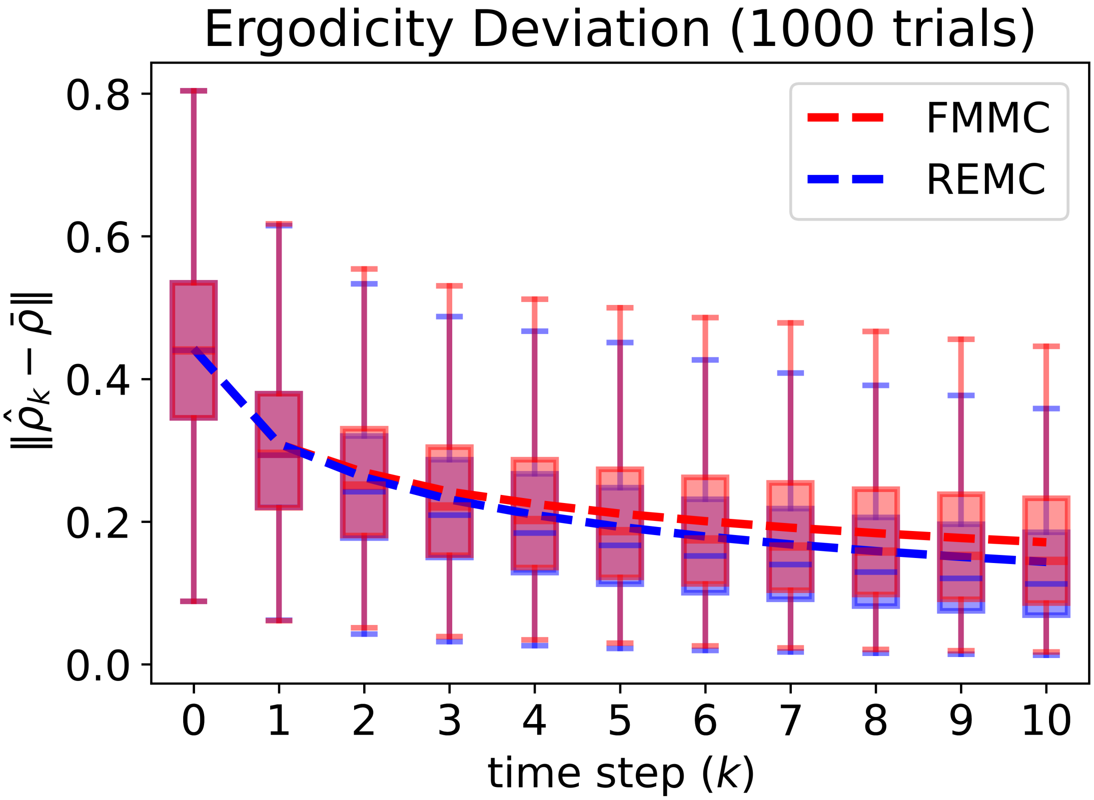


Figure 4.6: Ergodicity comparison between fastest mixing Markov chain (FMMC) and REMC from Theorem 4 tested on the region graph from Fig. 4.3, showing that REMC has a progressively lower ergodicity error on average.

Results

Figure 4.6 shows the expected ergodicity deviations in 10 time steps for FMMC and REMC separately over 1,000 trials. The result shows that the REMC method attains ergodicity faster than FMMC on average. Additionally, it has a substantially lower worst case deviation (shown by the top blue whiskers) than FMMC. Figure 4.7 shows the pairwise difference (FMMC - REMC) between the two methods for the same trials. The result shows that REMC is consistently more ergodic than FMMC, with the first quartiles became larger than 0 after 7-steps. More comprehensive experiments were performed on randomly generated graphs (see Appendix 4.J).

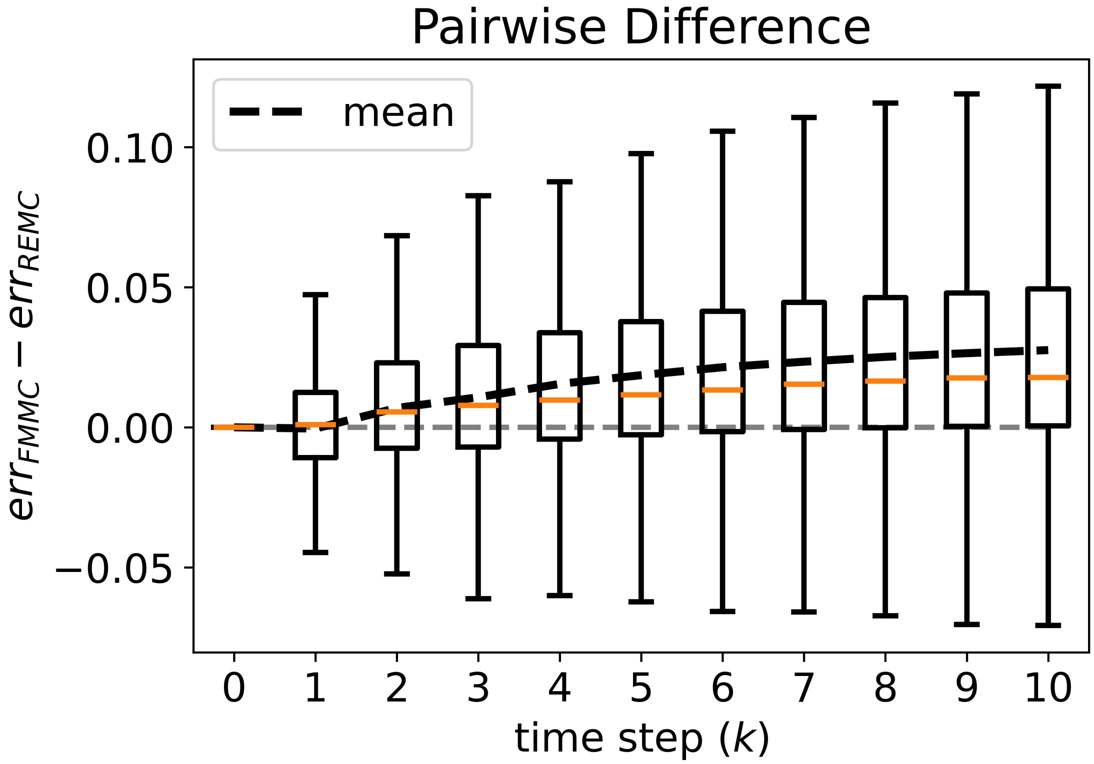


Figure 4.7: Boxplot of the pair-wise error differences for all the trials in Fig. 4.6. For the same initial condition and target distribution, REMC has progressively less error as the time step increases on an average as compared to FMMC.

4.5.2 Grid World Experiment

The following experiments shows the comparison of region-based ergodicity between HEMaP and three other continuous space ergodic planners: spectral multiscale coverage (SMC) [78]; heat equation driven area coverage (HEDAC) [55]; and graph-based ergodic search in cluttered environments (GESCE) [105], in the occupancy grid of the ballast tank shown in Fig. 4.2.

Methodology

The occupancy grid map is created by projecting the ballast tank CAD model onto the ground, with a lower and upper height limit of 0.05 m and 0.2 m, respectively. Additionally, the occupied grids are inflated with the radius of a TurtleBot3 Burger. The robot is modeled as a point mass moving at a constant speed of 0.2 m/s. Specifically, for SMC, the robot follows a single integrator model

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \end{bmatrix} + \Delta t \begin{bmatrix} v_x \\ v_y \end{bmatrix} \tag{4.39}$$

with $\sqrt{v_x^2 + v_y^2} = 0.2$. For HEMaP, the robot is allowed to move to any of the 8 neighboring grid cells centered at the current grid location within the time interval Δt. For HEDAC, the robot follows second-order dynamics according to this GitLab implementation[1]. For GESCE, the robot travels directly according to a probabilistic road map (PRM) planner, i.e., it follows a straight line from one collision-free node to the next without any low-level dynamical constraints.

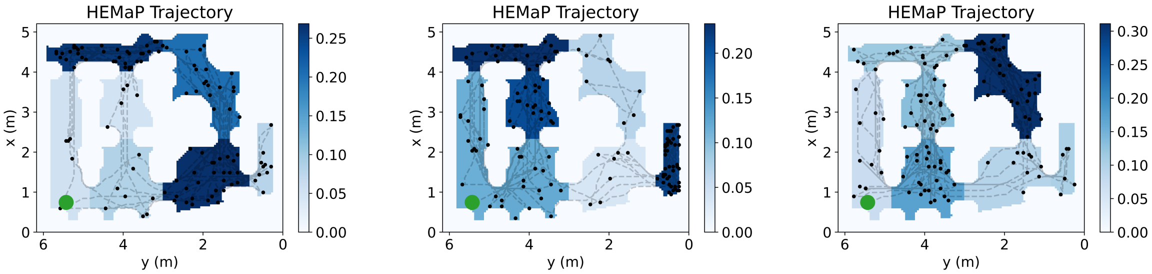
The region-based target distribution $\bar{\rho}$ is generated by sampling from the Dirichlet distribution with the shape parameter $\alpha = \mathbf{1}$. The grid-based target distribution is created by uniformly distributing the associated region-based target distribution on to the grid cells within each region, as shown in Fig. 4.8. To account for collision avoidance, grid cells belong-

---

[1] `https://gitlab.idiap.ch/rli/robotics-codes-from-scratch/-/blob/master/python/ergodic_control_HEDAC_2D.py?ref_type=heads`

ing to obstacles are considered an extra region, with a target visitation frequency of 0. For SMC, 20 Fourier coefficients are used with a time step of 0.01 s. For HEMaP, the waypoints are selected uniformly randomly in the target region indicated by the graph planner. A* is chosen as the path planning algorithm to navigate between the waypoints. For GESCE, 200 nodes are uniformly randomly sampled from the free space for the PRM with any pairs of nodes within 0.6 m connected by an edge. The cost function is identical to the SMC implementation. A total of 500 iterations are allowed during the search cycle to maintain a reasonable runtime. The robot is allowed to run for a total of 1,200 s (except for GESCE), and a total of 30 trials are performed for each method.

Results



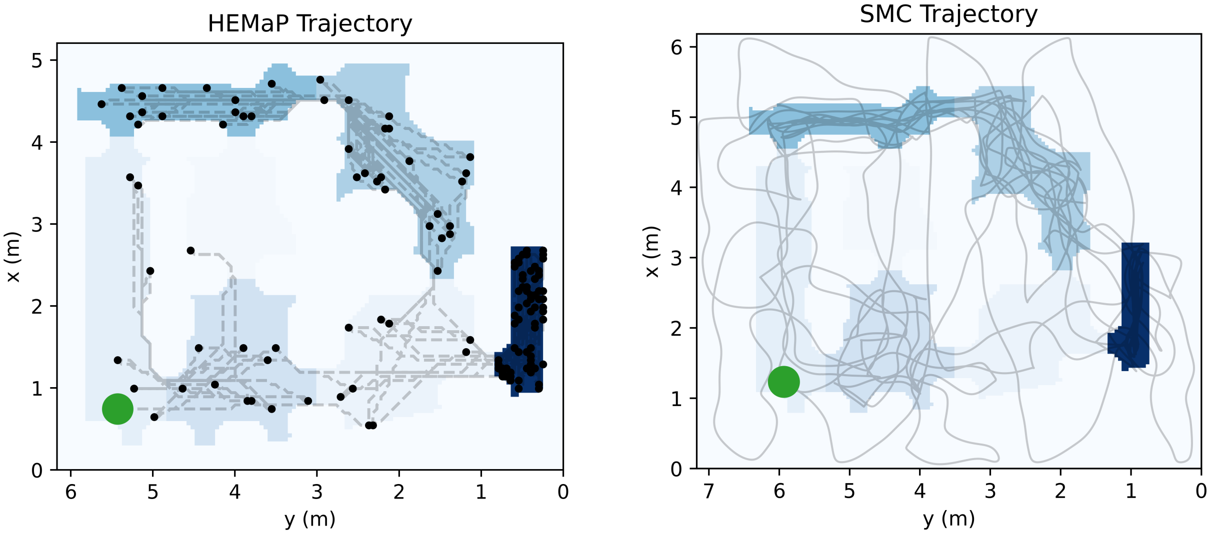
(a)

(b)

(c)

Figure 4.8: Example trajectories generated by HEMaP with A* as the low level planner and uniformly random waypoint placement. This shows that ergodic planning can adapt to various spatial distributions $\bar{\rho}$, where the number of sampled points (shown as black circles) is proportional to the desired region visitation frequency (shown as shade of blue).

A few representative robot inspection trajectories are shown in Fig. 4.8, which demonstrate that HEMaP works well regardless of the target distribution $\bar{\rho}$ that varies according to the region-level anomaly entropy. To provide a contrast with the continuous space ergodic planner trajectories, Fig. 4.9 shows comparisons between HEMaP and SMC for the



(a) HEMaP

(b) SMC

Figure 4.9: Example ergodic trajectories comparison for HEMaP and SMC, with the darker blue shade indicating higher target frequency. The green circles represent the initial location; the black dots in (a) represent the inspection waypoints and dashed gray line shows intermittent trajectory; the gray line in (b) shows continuous trajectory. SMC yields more complex trajectories that enter the obstacle region even if the desired distribution is set to 0.

same choice of $\bar{\rho}$. While HEMaP trajectory is completely contained in the workspace, SMC cannot always avoid the obstacles as it considers a finite number of Fourier coefficients. The Gibbs phenomenon also causes the robot to actively run into obstacles, as seen by the long arcs stretching outside the workspace. In a similar vein, SMC does not have any knowledge of the topology of the space, which allows the robot to pass through obstacles and reach different regions directly. Additionally, the path generated by the A* planner in HEMaP is more straightforward for a ground robot to follow than the complex trajectory generated by SMC.

Figure 4.10 shows region-wise ergodic deviations at each time step for all the ergodic planners averaged over the complete set of trials. Figure 4.11 plots the statistics on the pairwise differences between HEMaP and the continuous space planners on the same set of trials. For SMC and HEDAC, the relative frequency is calculated based on the current location (region) of the robot, with obstacles treated as an extra region. For HEMaP, the relative frequency is calculated for each waypoint and the ergodicity deviation is held constant from the previous waypoint while the robot is in transition. Similarly, the relative frequency for GESCE is calculated for each node in the PRM graph and held constant during the traversal between the nodes.

The results show that HEMaP is able to reduce the region-based ergodicity deviation faster than all the other planners on average. This happens even though SMC does not consider the topology of the explored space, and can be explained by the fact that SMC is designed to handle rectangular domains that are partitioned into regular grid-like regions at varying scales using a Fourier basis. However, the ballast tank regions are irregular and non-rectangular with sharp changes in the desired distribution values at the region boundaries, which are hard to capture using a Fourier series. HEDAC, on the other hand, has a higher error than SMC initially due to obstacle avoidance. However, it is capable of approaching the performance of HEMaP at around 800 s, since it has significantly less collisions compared to SMC[2]. GESCE has similar initial performance as SMC while maintaining obstacle avoidance,

---

[2] Minor collisions, similar to Fig. 4 in [55], can still be observed. This is likely due to discretized second-

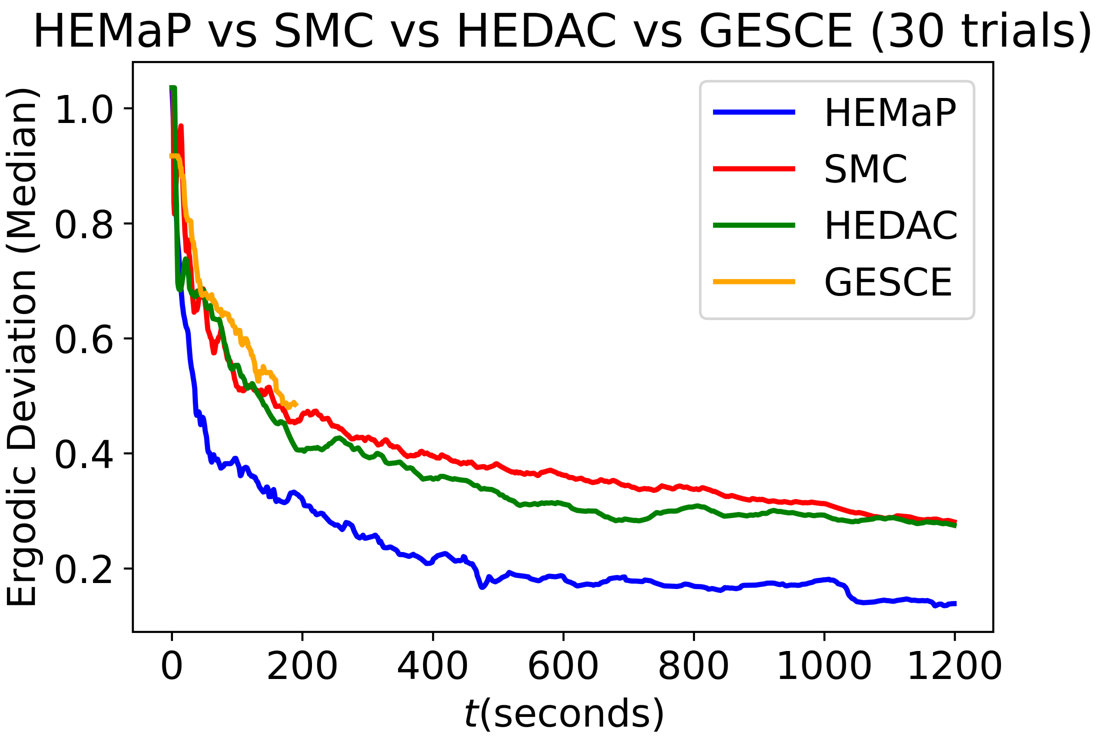
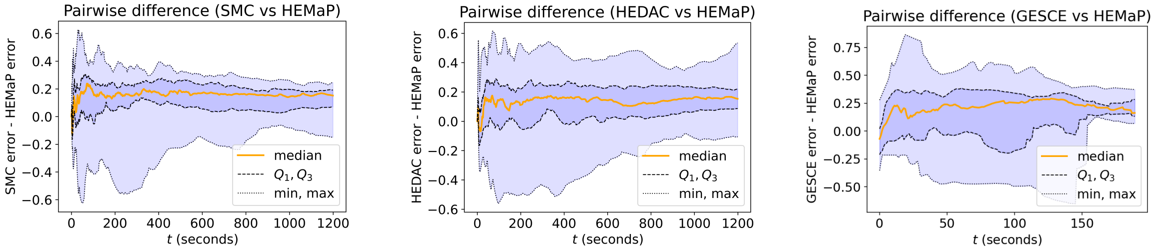


Figure 4.10: Comparison of region-based ergodicity between HEMaP and the various continuous space ergodic planners in the ballast tank (shown in Fig. 4.3) with the robot running at 0.2 m/s for 1,200 seconds. HEMaP has better region-based ergodicity on average.



(a) SMC (b) HEDAC (c) GESCE

Figure 4.11: Statistics on pair-wise differences in ergodicity deviations (errors) between HEMaP and the three continuous space planners over the same ser of trials used in Fig. 4.10. (a) shows that for the same initial condition and target distribution, HEMaP consistently has less error on average compared to SMC, with the 3rd quartile above 0 after 200 seconds. (b) shows a similar trend for HEDAC, where HEDAC has a higher initial error but lower asymptotic error because of its collision avoidance capability. (c) shows GESCE has an almost identical performance to HEMaP in terms of the 1st quartile values (worse with respect to the median and 3rd quartile values though), but is unable to run for more than 200 seconds.

and improves substantially if the search is allowed to run longer. The main challenge is that the trajectory length cannot be specified, due to which none of the trials exceed the 200 second mark, while all the other methods can run indefinitely.

In general, the main advantage of HEMaP is that it discards fine-grained information in pursuit of *rapid* convergence to the target visitation distribution of inspection regions. Therefore, it takes the shortest A* path between the regions, which enhances ergodicity at the region level. This is in contrast to all the continuous space methods that try to provide intra-regional coverage, which affects region-level ergodicity deviation and leads to fractal-like trajectories. If the fine-grained information are crucial and the robot is able to execute complex trajectories, then one can consider a continuous space method. Alternatively, since HEMaP is modular and hierarchical, we envision the possibility of replacing the A* planner with any compatible continuous space method, which is further discussed in Section 4.6.

4.5.3 Gazebo Simulation

The following experiment compares the performance of two different graph traversal methods against HEMaP, in the context of FOD detection in the ballast tank of Fig. 4.2. Specifically, we consider random walk, where the planner samples from all the edges in each region uniformly; and a greedy method, where the planner chooses the edge that leads to the neighboring region with the highest estimated information measure. Additionally, the waypoint placements within the regions are changed accordingly, with random walk choosing the sampled waypoints randomly; and the greedy method choosing the sampled waypoints with the maximum estimated information.

Methodology

Simulation is performed in Gazebo using the ballast tank model in Fig. 4.2. A rusty texture is applied to the entire tank structure and all the light sources are removed. The robot is a

---

order dynamics, where the potential field does not produce sufficient acceleration for collision avoidance.

Figure 4.12: Example picture of the Gazebo simulation with the purple TurtleBot3 Burger running in the ballast tank in Fig. 4.2, with rusty texture and no external light source. A few scattered AprilTags are used as known features in the tank.

TurtleBot3 Burger with the original camera replaced by a RealSense D435. AprilTags are scattered around the tank as partially known features and the robot wheel encoders provide odometry information for the hierarchical SLAM framework described in Section 4.A.1. A representative image of the simulated environment is shown in Fig. 4.12. The same occupancy grid map from Section 4.5.2 is used as the static global cost map for the ROS navigation planner and a LiDAR is used to generate the local cost map for the local planner. The number of points in the reference point cloud, $\mathcal{P}_{\text{ref}}$, is empirically selected as $100,000$ to provide sufficient density for the detection of FODs, given the size of the tank and the FODs. The buffer zone threshold, d_{buffer}, in Fig. 4.20 is selected to be 3.5 cm. The smoothing factor, δ, in Algorithm 1 is selected to be 0.001.

Each method is allowed to travel on the graph for 35 steps, with two waypoints in the same region per step. 1,000 waypoints are sampled for waypoint placement. The hierarchical SLAM has an estimation horizon of 5 pose nodes, including the inspection nodes. FOD detection is performed when the inspection nodes are pruned from the estimation horizon. The anomaly detection update rule (4.55) is initialized with a prior null hypothesis of $\mathbb{P}(H_0) = 0.8$ given that proving the null hypothesis is more challenging than proving the alternative hypothesis. In other words, the robot assumes the tank is FOD-less until proven otherwise. A set of six common hand tools is chosen as the FOD, comprising 1) power drill × 2; 2) spirit level; 3) sander; 4) hammer; 5) measuring tape. 4 to 6 FODs are sampled for each trial, and each FOD is placed by uniform sampling from the edges of the occupancy map of the tank. If the FODs are hidden behind a structure or blocking the doorways, they are moved to the nearest available location.

The robot is placed in region 1, near the access hole shown in Fig. 4.2, at the beginning of each trial. After an inspection trial is complete, all the points with $\mathbb{P}(H_1) \geq 0.5$ from the reference point clouds, shown in Fig. 4.13, are extracted and clustered using hierarchical clustering [2], with a distance criterion of $0.5m$. The centroids of the clusters are calculated, and photos of the tank sections taken from the inspection waypoints with the centroids in view are saved as candidate FODs. During actual deployment, these photos can be

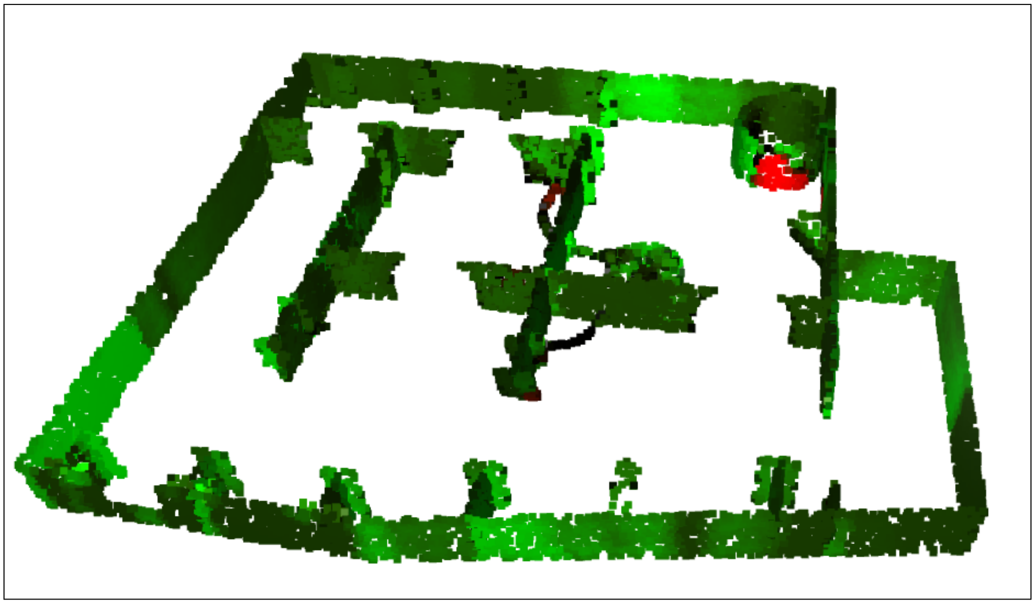


Figure 4.13: Example of a typical anomaly belief of the reference point cloud for HEMaP inspection with the hue (from red to green) representing the null hypothesis, $\mathbb{P}(H_0)$, (from 0 to 1); and the value (from color to black) representing the entropy of the hypotheses (from 0 to 0.69, corresponding to the minimum and maximum entropy of the Bernoulli distribution).

(a) FOD Image

(b) False Positive

Figure 4.14: Example images collected by the robot during the inspection process with (a) showing a power drill behind a cable hub, and (b) showing a false positive with a pillar in the tank.

presented to a human operator, who can then decide whether the candidates are actually FODs. Figure 4.14 shows two example photos saved by the robot. A total of 15 trials are performed and the FODs captured in the photos are analyzed.

	Detected FODs	Missed FODs	False Positives
Random	45	31	19
Greedy	39	37	**7**
HEMaP	**64**	**12**	21

Table 4.1: Total FOD detection counts (15 trials).

Results

Table 4.1 shows the results for the FODs detected in the photos by the three inspection methods over 15 trials. A total of 76 FODs are present, of which HEMaP only misses 12 FODs. However, the random and greedy methods miss a much larger number of FODs. This occurs due to an inherent bias toward visiting highly connected regions and high-information regions by the random and greedy methods, respectively. As a result, some of the other tank regions are not inspected adequately when the total number of time steps is limited. The visitation frequency for each region is discussed in more detail in our subsequent analysis. HEMaP and random methods share a similar number of false positive photos, while the greedy has the lowest number. This happens because both HEMaP and random methods distribute the inspection points, whereas, the greedy method focuses on refining several high information points.

A paired t test is also performed between the HEMaP and the other two FOD detection methods (across all the simulation trials) to investigate any difference in detection perfor-

	Mean	Standard Dev.	p-value (w/ HEMaP)
Random	0.59	0.24	1.1×10^{-3}
Greedy	0.51	0.18	4.8×10^{-9}
HEMaP	**0.84**	**0.13**	N/A

Table 4.2: FOD detection rate statistics for 15 simulation trials with hypothesis tests done using a paired t-test.

mance. The results are shown in Table 4.2. Consistent with the total count results, HEMaP has the highest mean detection rate along with the lowest standard deviation, indicating that it is more consistent in detecting FODs. The HEMaP also shows statistically significant improvements over the random and greedy methods, with corresponding p-values of 1.1×10^{-3} and 4.8×10^{-9}, respectively.

Regions	1	2	3	4	5	6	7
Random	0.24	0.21	0.15	0.23	0.04	0.10	0.02
Greedy	0.30	0.26	0.00	0.04	0.36	0.03	0.00
HEMaP	0.23	0.16	0.10	0.13	0.22	0.10	0.05

Table 4.3: Average distributions of waypoints in different regions in the ballast tank with the three traversal methods.

Comparisons among the three traversal methods in terms of the average distribution of waypoints in the regions are reported in Table 4.3. Using the region graph shown in

Fig. 4.3, it is seen that the distribution of waypoints for HEMaP is roughly correlated to the region size, with regions 1 and 5 being highly visited and regions 3 and 7 visited less frequently. As expected, the random method is highly susceptible to the region graph structure. Accordingly, the highly connected regions of 1, 2, 3, and 4 are visited most frequently, whereas region 5 with the directed edge is significantly under-visited. This results in a large number of missed FODs when they are spawned in these under-visited regions. The greedy method, on the other hand, is susceptible to an overreliance on the highest estimated information. The robot spends a large amount of time in the two locally largest regions of 1 and 2 until they are completely covered. The robot then quickly moves on to the most information-rich region 5 through regions 4 and 6. The robot never leaves region 5 during the allowed 35 steps, resulting in zero inspection for regions 3 and 7.

Furthermore, 15 trials with the same setup are performed for 15 and 25 steps, and the detection rates are shown in Fig. 4.15. As expected, all the methods have a decreased detection rate with a smaller number of steps. Nevertheless, even with a small number of steps, HEMaP is more effective in detecting FODs than the other two methods. This experiment showcases the adaptability of REMC to arbitrary target distributions and how it is combined effectively with the anomaly detector in the HEMaP framework. The random and greedy exploration methods are highly dependent on the graph structure, where the visitation frequencies are determined by the connectivity of the graph, causing the highly connected regions to be visited more frequently. Additionally, the convergence rates of the time averages to the corresponding target distributions are not optimized for the random and greedy methods. Therefore, the short-term visitation frequency is biased toward the regions closer to the starting region, while the far away regions are less likely to be visited.

4.5.4 Physical Deployment

The inspection framework using the REMC method from simulation is applied to a physical mock-up ballast tank to verify the simulation results and showcase its capability for real-world deployment.

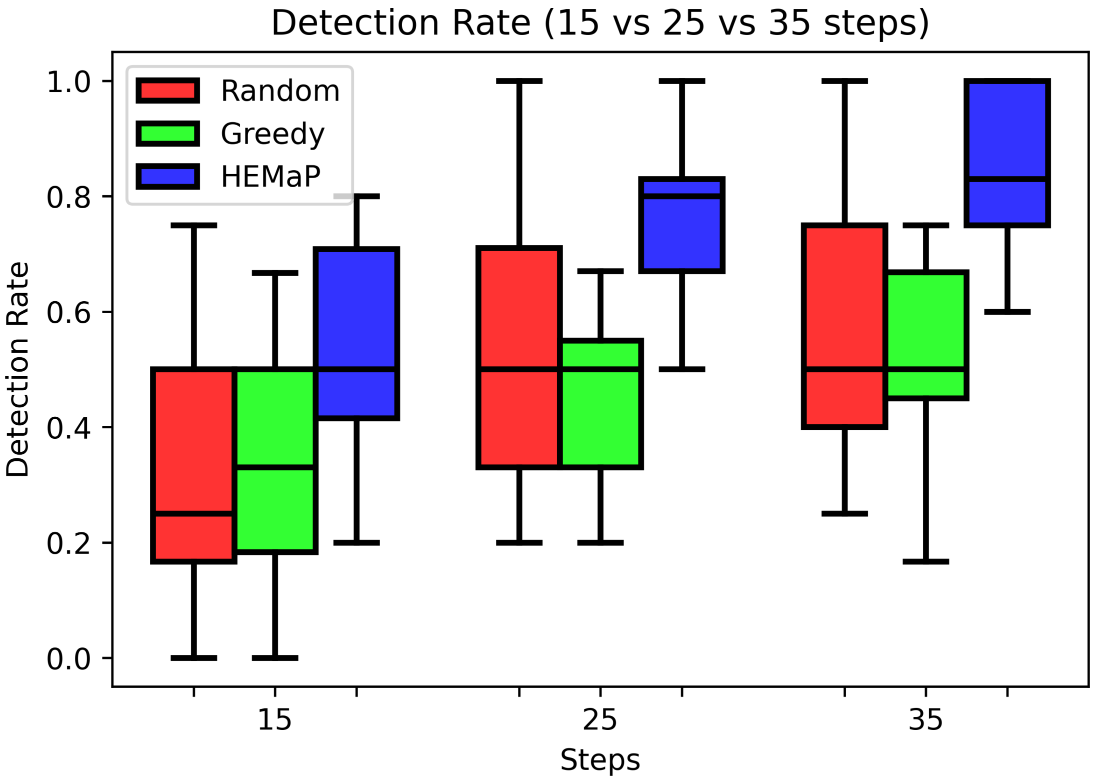


Figure 4.15: Comparison of FOD detection rates among HEMaP, random, and greedy methods for 15, 25 and 35 steps. HEMaP is found to be more efficient at detecting FODs even with a limited number of steps.

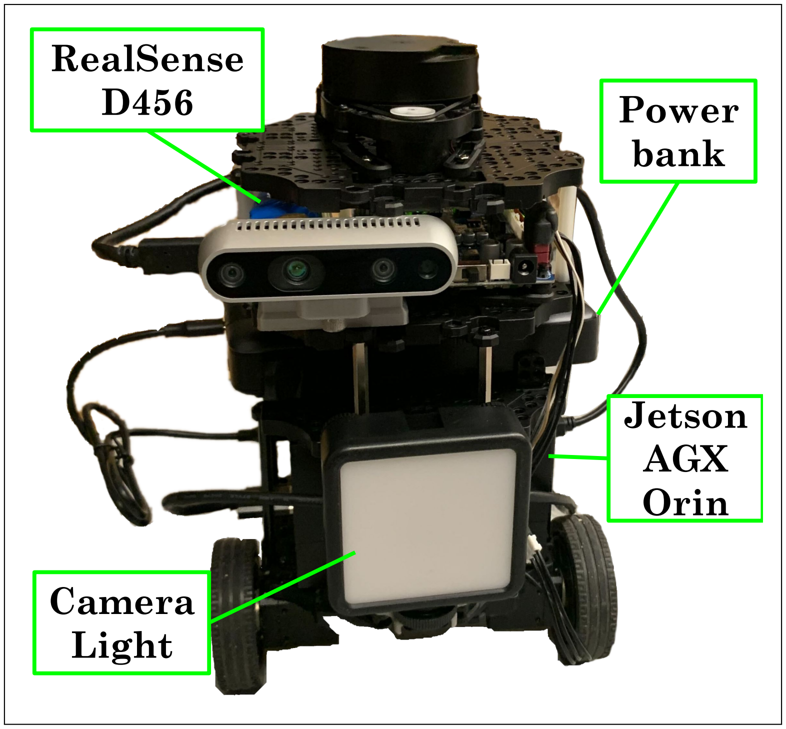
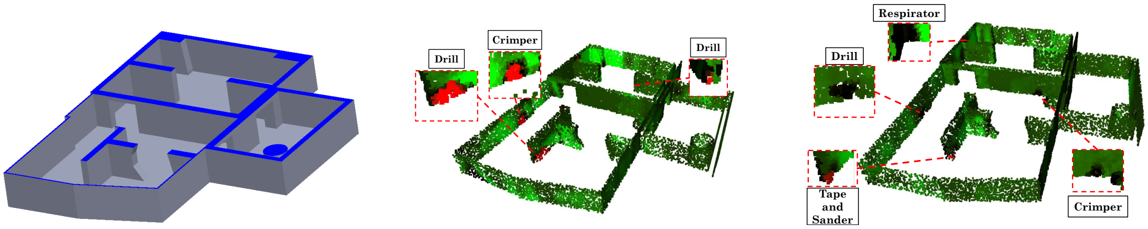


Figure 4.16: TurtleBot3 Burger used in the physical experiment, customized with camera light, Jetson AGX Orin, RealSense D435 and a power bank.

Methodology

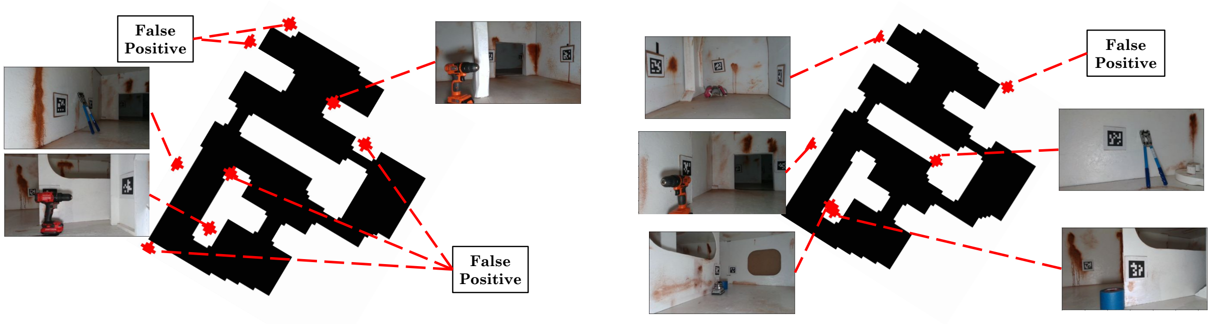
The mock ballast tank is constructed using a combination of plywood and ramboard for the walls. The walls are painted white to mimic a typical tank color and an additional brown paint is used to mimic natural corrosion and rust caused by seawater. AprilTags from the family 36h11 of size 0.1 m are scattered around the tank as partially known features for the SLAM. A TurtleBot3 Burger is chosen as the robot platform. The robot is customized by replacing the Raspberry Pi with an Nvidia Jetson AGX Orin running Ubuntu 22.04 and ROS Noetic. The Pi camera is replaced by a RealSense D435 for depth perception. The ball bearing caster wheel is replaced with a 1.27 cm swivel caster for better locomotion on uneven ground. An LED camera light is mounted on the robot to illuminate the tank. The TurtleBot LiDAR is used to generate odometry using the RF2O ROS package [59].

The hierarchical SLAM is run once in teleoperation mode to map the AprilTags locations before the autonomous inspection trials. The local point clouds are also collected and assembled into a global map to generate the reference CAD model. Similar to Section 4.5.3, the occupancy grid map is generated by projecting the reference CAD model within a height range of 0.05 m to 0.2 m. Similarly, the occupancy grid map is used for the global planner and the LiDAR is used for the local planner. The number of points in the reference point cloud, $\mathcal{P}_{\text{ref}}$, is empirically selected as $100,000$. The buffer zone threshold, d_{buffer} and the smoothing factor, δ, are chosen to be 1 cm and 0.001, respectively. A set of six common hand tools is chosen for the FODs, specifically: 1) power drill × 2; 2) masking tape; 3) sander; 4) crimper; 5) sander. A total of 5 trials are performed with 35 steps each. 3 to 5 FODs were sampled for each trial with a total of 21 FODs for the five trials. Each FOD is placed in the tank by uniformly sampling from the edges of the occupancy map. If the FODs are hidden behind a structure or blocking the doorways, they are moved to the nearest available location.



(a) Reference Mesh (b) Example trial 1 (c) Example trial 2

Figure 4.17: The reference mesh of the mock ballast tank and two example reference point clouds after the inspection showing HEMaP has a good coverage of the workspace in small steps. The true FOD locations are denoted by the text box. The hue (from red to green) representing the null hypothesis, $\mathbb{P}(H_0)$, (from 0 to 1), i.e. red is considered to be FODs by the robot; and the saturation (from color to black) representing the entropy of the hypotheses (from 0 to 0.69, corresponding to the lowest and highest entropy of Bernoulli distribution, respectively). Figure (b) shows a trial of a set of three FODs and Figure (c) shows a trial of a set of 5 FODs. The corresponding detection of these FODs are shown in Fig. 4.18.



(a) Example trial 1 (b) Example trial 2

Figure 4.18: Occupancy map of the ballast tank showing the high detection rate of HEMaP. Red points shows the candidate FOD locations, where the reference points from the corresponding trial in Fig. 4.17 with $\mathbb{P}(H_1) > 0.5$ are segmented and clustered. The photos collected from the robot inspections are shown for the true positives.

Results

The overall FOD detection rate is 0.853 ± 0.2, which is similar to the detection rate in simulation. The average number of false positives is 3.4, which is slightly higher than in simulation. This is expected given that sensor readings are noisier in real life, together with modeling inaccuracy in the reference point cloud. Figure 4.17 shows examples of the reference point cloud from two trials after inspection. More black patches (close to 0.5 posterior confidence) are observed compared to the simulation trials, meaning that there is more confusion from the robot. The locations of the true FODs are often in bright red, representing a high confidence. Figure 4.18 shows the associated points of interest, where the points with an anomaly posterior greater than 0.5 are segmented and clustered. In both the trials, all the FODs are found, with trial 1 having more false positives. It can be seen that most of the false positives arise in the corners of the structures. These corners are sensitive to misalignment in the reference model. The depth camera is also noisier around the corners with sharp changes in depth.

4.6 Discussion

The high variance in performance arising from the stochasticity of the Markov chain can be reduced through Monte Carlo tree search (MCTS) [12]. In MCTS, multiple sequences with a predefined time horizon are sampled from the Markov chain, and the sequence with the highest increase in ergodicity is selected instead of sampling the next state at every step. Here, we use the single-step sampling option to investigate the fundamental capability of an ergodic Markov chain and devise an exploration framework that can be extended to communication-limited multi-robot systems. This extended framework can be potentially applied to a wider array of exploration problems beyond inspection, such as wildlife monitoring in a biome and lost object retrieval in a building environment.

An interesting potential of the hierarchical planning framework is that graph ergodicity can be used in conjunction with the various continuous space exploration methods. In this case, the workspace can be partitioned into spaces with simpler topology, and instead of waypoint placement, the robot executes a continuous space ergodic trajectory within each region. This alleviates the need for calculating the basis functions globally for the complex topology and instead focuses on the fine details only in the local region, while the region transition is handled by the Markov chain.

An ergodic Markov chain can also be potentially used in continuous space. The Metropolis-Hasting algorithm allows one to construct a continuous-space Markov chain with any target distribution by using the detailed balance condition. Since the detailed balance condition is completely local, the robot can execute an ergodic transition without knowing the global distribution. However, as discussed earlier, the detailed balance condition impacts the convergence rate to time-discounted ergodicity and requires a reversible motion model of the robot. To improve the convergence rate and remove the reversibility requirement, it would be interesting to study the potential of REMC in continuous space.

It is also possible to consider an ergodic Markov chain in the state space instead of the physical space. The state space can be either graph-based or topological, with inherently stochastic dynamics analogous to a Markov decision process. A classical strategy for state space exploration in reinforcement learning (RL) is to uniformly sample an available action [18, p.32]. This is analogous to the random method in Section 4.5.3, where the visitation frequency is highly dependent on the connectivity of the space. Therefore, we are interested in investigating whether an on-policy RL method can be developed using an ergodic Markov chain-based stochastic policy.

In this work, the transition times between the regions are not considered. The transition model of [11] can potentially be applied to REMC to reduce extensive region switching. This also opens another application of rapid ergodicity to Markov processes, which is the continuous-time version of Markov chains. In the context of confined space work, modeling the transition time, or equivalently, region residence time, opens the method to more types of tasks that require long wait times, such as removal of the detected FODs or on-the-spot maintenance of damaged structures. A time sensitive ergodic Markov chain can explicitly balance the visitation frequency according to how long the robot would reside in a region.

A final direction would be to extend the anomaly detection method to a broader class of anomalies that include rust patches and structural damages, in addition to FODs. Conceptually, a machine learning-based object detection model with a confidence score, such as [87, 100], can be used to replace the likelihood calculation in (4.53)-(4.54), while the Bayesian update and information metric can remain the same.

Appendix 4.A Estimation Framework

We now present the estimation module in Fig. 4.1 that comprises SLAM and anomaly detection. In addition to the occupancy grid map and pose estimates for navigation, the estimator also provides the measure μ (defined in (4.3)) to the REMC method. Specifically, this measure is selected as information entropy using a Bayesian anomaly (FOD) detection method. The core idea of detecting FODs is to evaluate the hypothesis that the difference between the local (spatial) point cloud given by the robot's depth camera and the reference model of the corresponding section of the confined space is significant. Consequently, the FOD detection method uses the marginal posterior of our SLAM formulation described in the following Section.

In short, the SLAM module (Section 4.A.1) provides optimal estimates of the robot's poses and map features along with their uncertainties. These estimates are then used to transform the local spatial point cloud, sampled by the RGB-D camera, into the global coordinate system, which subsequently helps with anomaly detection (Section 4.A.2). Finally, the information metric μ_p is generated from the confidence level of anomaly detection. A flow chart outlining the steps is shown in Fig. 4.19.

4.A.1 Hierarchical SLAM

The SLAM module serves two purposes in our inspection framework. First, it provides a probabilistic mapping of the environment in the form of a 3D point cloud of the structures that is used for FOD detection via Bayesian inference. Second, it estimates the robot pose

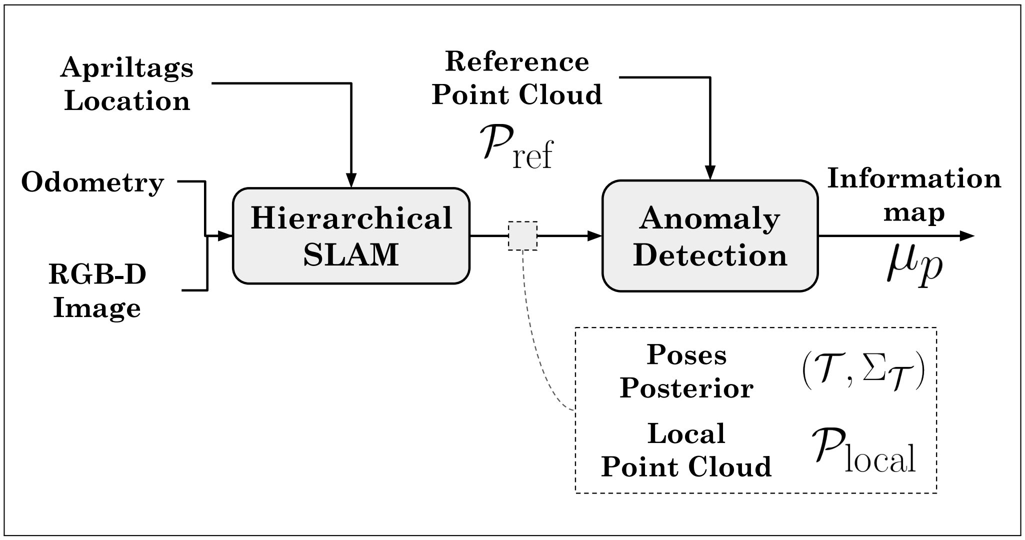


Figure 4.19: Flow chart of the estimation module. Hierarchical SLAM processes the sensor information to generate a Gaussian posterior of the pose, $(\mathcal{T}, \Sigma_{\mathcal{T}})$, detailed implementation of which is discussed in Section 4.A.1. The anomaly detector compares the local point cloud from the RGB-D camera $\mathcal{P}_{\text{local}}$ with the reference point cloud $\mathcal{P}_{\text{ref}}$ using the posterior to generate the information map μ_p.

(location and orientation with respect to the map origin), enabling navigation within the space. To accommodate non-linear robot motion and sensor models, we use the Graph-SLAM algorithm. In this method, a factor graph is constructed during run-time based on the collected observations and odometry, with nodes representing the robot's poses and map features.

Specifically, the poses are calculated on the SE(3) manifold, i.e., a set of 3D rigid transformation matrices, according to the formulation in [107], and are optimized by the Gauss–Newton algorithm described in [45]. We modify the full graph-SLAM, optimized on the graph of all historical poses, as a hierarchical SLAM, similar to [15] and [46]. Instead of maintaining the full resolution factor graph and performing hierarchical optimization, marginalization is done as an intermediate EKF-SLAM, and only the sparse factor graph is maintained with the posterior from the EKF-SLAM as the factor between the nodes. In particular, EKF-SLAM is re-initialized to the initial condition (zero mean and infinite covariance), and a new pose node is created based on the distance traveled from the most recent pose node. Special nodes, called *inspection nodes*, are created once the robot reaches the waypoint specified by the waypoint placement method. The corresponding depth point clouds, i.e., the point clouds generated by the depth images taken by the robot at the waypoints, are also stored at these inspection nodes. Furthermore, nodes that fall outside the current *estimation horizon* are pruned and replaced by a factor node through marginalization. Once the Graph-SLAM is optimized, the estimated poses of all the unpruned nodes, $\mathcal{T} \triangleq \{T_1, T_2, \cdots T_i \cdots\}$, and the corresponding covariance matrices, $\Sigma_{\mathcal{T}} \triangleq \mathrm{Var}(\mathcal{T})$, are used for recursive anomaly detection described in the next Section.

4.A.2 Bayesian Anomaly Detection

As indicated earlier, we perform Bayesian inference on the likelihoods of multiple competing hypotheses for anomaly detection. In particular, we use the following null and alternate hypotheses, denoted by H_0 and H_1, respectively.

$$\begin{aligned} H_0&\text{: The local point cloud is sampled from the reference model of the confined space.}\\ H_1&\text{: The local point cloud is sampled from a space outside a buffer zone around the reference model.}\end{aligned} \tag{4.40}$$

The buffer zone indicates an area around a confined space structure that is neither considered a FOD nor a part of the structure. This buffer zone allows for deviation in the modeling of the structure or objects that are too small to be considered FODs. This buffer zone is specified by a distance threshold, d_{buffer}, offset in the direction of the normal vector of the walls of the reference model, as illustrated in Fig. 4.20.

The probabilistic observation model requires the posterior estimation of the pose of the inspection nodes from the hierarchical SLAM, which is in the form of a multivariate Gaussian, $\mathcal{N}(\mathbf{x}, \Sigma_{\mathbf{x}})$, with $\mathbf{x}$ being the mean vector and $\Sigma_{\mathbf{x}}$ being the covariance matrix. In particular,$\mathbf{x}$ is a concatenation of all the estimated poses and the blocks of $\Sigma_{\mathbf{x}}$ are the covariance matrices associated with the poses,

$$\mathbf{x} = \begin{bmatrix} \vdots \\ \mathbf{x}_i \\ \mathbf{x}_j \\ \vdots \end{bmatrix} \Sigma_{\mathbf{x}} = \begin{bmatrix} \ddots & & & \\ & \Sigma_{i,i} & \Sigma_{i,j} & \\ & \Sigma_{i,j}^{\mathsf{T}} & \Sigma_{j,j} & \\ & & & \ddots \end{bmatrix}. \tag{4.41}$$

The marginal posterior of a particular inspection node can be obtained from the graph posterior by marginalization. For a Gaussian distribution, it is done by selecting the corresponding entries in $\mathbf{x}$ and blocks from the diagonal of $\Sigma_{\mathbf{x}}$ [30]. For instance, the marginal posterior of node i in (4.41) is

$$\mathbb{P}(\mathbf{x}_i) = \mathcal{N}(\mathbf{x}_i, \Sigma_{i,i}). \tag{4.42}$$

In the SE(3) formulation, $\mathbf{x}_i$ is represented by a 4×4 rigid transformation matrix T_i. The corresponding covariance is a 6×6 matrix with respect to the 6 degrees of freedom of T_i

on the manifold as defined in [107, eq.(52)]. The local point cloud, $\mathcal{P}_{\text{local},i}$, with a set of points, $p_{\text{local}} \in \mathbb{R}^3$, is stored in the local coordinate of its associated pose node, with some independent observation noise $\Sigma_{p_{j,\text{local}}}$ for each point $p_{j,\text{local}}$. An example is illustrated in Fig. 4.20(a).

Once the pose graph is optimized, the local point cloud, $\mathcal{P}_{\text{local},i}$, is transformed into the global coordinate with T_i

$$p_j = T_i p_{j,\text{local}}, \quad \forall p_{j,\text{local}} \in \mathcal{P}_{\text{local},i} \tag{4.43}$$

with the noise propagated as

$$\begin{aligned}\Sigma_{p_j} =& J_{T_i}^{T_i p_{j,\text{local}}} \Sigma_{i,i} \left(J_{T_i}^{T_i p_{j,\text{local}}} \right)^{\mathsf{T}} + \\ & J_{p_{j,\text{local}}}^{T_i p_{j,\text{local}}} \Sigma_{p_{j,\text{local}}} \left(J_{p_{j,\text{local}}}^{T_i p_{j,\text{local}}} \right)^{\mathsf{T}}.\end{aligned} \tag{4.44}$$

Here, $J_{T_i}^{T_i p_{j,\text{local}}}$ and $J_{p_{j,\text{local}}}^{T_i p_{j,\text{local}}}$ are the Jacobians of the 3D rigid transformation with respect to T_i and $p_{j,\text{local}}$, respectively. The exact forms of the Jacobians are available from [107].

After transformation, the point cloud is compared with the reference model $(p_{\text{ref}}, n(p_{\text{ref}}))$, where p_{ref} is the representation of the point cloud and $n(p_{\text{ref}})$ are the normal vectors of the points that are inherited from the plane the points are sampled from. As illustrated in Fig. 4.20(b), for each point, p_j, in the observed point cloud, we find the corresponding reference point as

$$p^*_{\text{ref},j} = \arg\min_{p_{\text{ref}}} (p_j - p_{\text{ref}})^{\mathsf{T}} \Sigma_{p_j}^{-1} (p_j - p_{\text{ref}}). \tag{4.45}$$

Consequently, the likelihood of p_j being sampled from the plane formed by $(p^*_{\text{ref},j}, n(p^*_{\text{ref},j}))$ or behind the plane (i.e., on the negative side of the normal vector $n(p^*_{\text{ref},j})$), is calculated by the half-space integral as

$$\begin{gathered}\mathbb{P}(p_j|H_0) = \int_{\Omega_0} \mathcal{N}(p_j, \Sigma_{p_j}) dp \\ \Omega_0 \triangleq \{p | n(p^*_{\text{ref}})^{\mathsf{T}} p - n(p^*_{\text{ref}})^{\mathsf{T}} p^*_{\text{ref}} \leq 0\}.\end{gathered} \tag{4.46}$$

This is calculated by the normal cumulative distribution function (normcdf) as

$$\mathbb{P}(p_j|H_0) = 1 - \text{normcdf}(c_0(p_j)) \tag{4.47}$$

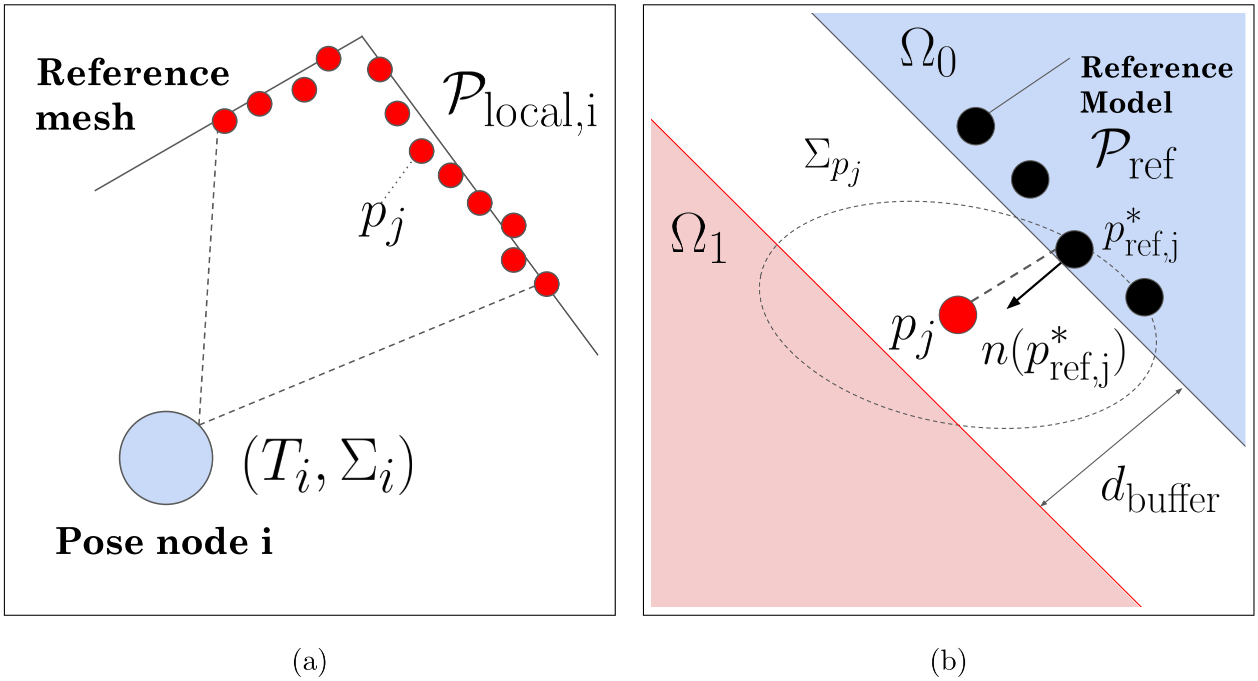


Figure 4.20: SLAM-based anomaly detection. (a) shows an illustration of a local point cloud $\mathcal{P}_{\text{local}}$ of an inspection pose node. T_i is the 4×4 rigid transformation of the pose node i, Σ_i is the associated covariance matrix, and the small circles, p_j, are the point cloud samples collected by the depth camera. (b) illustrates the associated FOD detection on a single point $p_j \in \mathcal{P}_{\text{local}}$ with the uncertainty Σ_{p_j} propagated from (a). The blue area, Ω_0, is the half space wherein the point would be considered to be inside the reference mesh; and the red area, Ω_1 is the half space wherein the point would be considered a foreign object. The black points are from the reference model, p_{ref} is statistically closest point to p_j with normal vector $n(p_{\text{ref}})$, and d_{buffer} is the buffer zone threshold. The likelihood of the point p_j being sampled from the blue and red areas are obtained by integrating the Gaussian $\mathcal{N}(p_j, \Sigma_{p_j})$ over the corresponding halfspaces.

where $c_0(p_j)$ is the signed Mahalanobis distance

$$c_0(p_j) \triangleq \frac{(n(p^*_{\text{ref},j}))^\mathsf{T}(p^*_{\text{ref},j} - p_j)}{\sqrt{(n(p^*_{\text{ref},j}))^\mathsf{T}\Sigma_{p_j} n(p^*_{\text{ref},j})}}. \tag{4.48}$$

$\mathbb{P}(p_j|H_1)$ is calculated similarly by offsetting p^*_{ref} by the buffer zone threshold d_{buffer}, and integrating from the opposite halfspace by $\mathbb{P}(p_j|H_1) = \text{normcdf}(c_1(p_j))$, with

$$c_1(p_j) \triangleq c_0(p_j) - \frac{d_{\text{buffer}}}{\sqrt{(n(p^*_{\text{ref},j}))^\mathsf{T}\Sigma_{p_j} n(p^*_{\text{ref},j})}}. \tag{4.49}$$

Since the likelihood of the observed signed Mahalanobis distance is a standard normal distribution, assuming independence, the average of $c_0(p_j)$ (and c_1) over different observed points p_j is also a normal distribution with

$$\bar{c}_0 = \frac{1}{\nu}\sum_{j=1}^{\nu} c_0(p_j), \quad \mathbb{E}(\bar{c}_0) = 0, \quad \sigma(\bar{c}_0) = \frac{1}{\sqrt{\nu}} \tag{4.50}$$

where $\sigma(\cdot)$ denotes the standard deviation and ν is the number of observations. This forms a new z-score

$$z_0 = \frac{\bar{c}_0 - \mathbb{E}(\bar{c}_0)}{\sigma(\bar{c}_0)} = \frac{1}{\sqrt{\nu}}\sum_{j=1}^{\nu} c_0(p_j). \tag{4.51}$$

z_1 can be defined identically using c_1. This means that instead of performing the integral one point at a time, the points can be batch processed as an average observation for each reference point. For each point $p_{\text{ref},i}$ in the reference model $\mathcal{P}_{\text{ref}}$, find the set of observed points, $\mathcal{P}_{\text{sample},i}$, which corresponds to itself. Additionally, for smoothing, consider the point set of the neighboring reference points given by

$$\mathcal{P}_{\text{sample},i} = \{p_j | p^*_{\text{ref},j} \in \mathcal{P}_{\text{NN},i}\} \tag{4.52}$$

where $\mathcal{P}_{\text{NN},i}$ is the set of k-nearest neighbor points of $p_{\text{ref,i}}$ in $\mathcal{P}_{\text{ref}}$.

Putting everything together, the likelihood for all the points in the reference model is calculated as

$$\begin{aligned} \mathbb{P}_i(\mathcal{P}_{\text{sample},i}|H_0) &= 1 - \text{normcdf}(z_{0,i}) \\ z_{0,i} &= \frac{1}{\sqrt{\nu_i}} \sum_{p_j \in \mathcal{P}_{\text{sample},i}} c_0(p_j) \end{aligned} \tag{4.53}$$

$$\begin{aligned}\mathbb{P}_i(\mathcal{P}_{\text{sample},i}|H_1) &= \text{normcdf}(z_{1,i})\\ z_{1,i} &= \frac{1}{\sqrt{\nu_i}}\sum_{p_j\in\mathcal{P}_{\text{sample},i}} c_1(p_j).\end{aligned} \tag{4.54}$$

Finally, the posterior of the hypothesis for each reference point is updated using Bayes law as

$$\mathbb{P}_i(H_0|\mathcal{P}_{\text{sample},i}) = \zeta\mathbb{P}(\mathcal{P}_{\text{sample},i}|H_0)\mathbb{P}_i(H_0) \tag{4.55}$$

$$\mathbb{P}_i(H_1|\mathcal{P}_{\text{sample},i}) = \zeta\mathbb{P}(\mathcal{P}_{\text{sample},i}|H_1)\mathbb{P}_i(H_1) \tag{4.56}$$

$$\zeta \triangleq \frac{1}{\mathbb{P}_i(H_0|\mathcal{P}_{\text{sample},i}) + \mathbb{P}_i(H_1|\mathcal{P}_{\text{sample},i})} \tag{4.57}$$

where $\mathbb{P}(H_0)$ and $\mathbb{P}(H_1)$ are the prior probability of the null and alternative hypothesis, respectively.

4.A.3 Information Measure

The region measure, μ_r, for generating the space average, $\bar{\rho}$, in (4.7) is defined using the posterior of the hypothesis. In particular, we define information measure using the entropy of the posterior for each reference point as

$$\mu_p(p_{\text{ref},i}) = -\mathbb{P}_i(H_0)\ln(\mathbb{P}_i(H_0)) - \mathbb{P}_i(H_1)\ln(\mathbb{P}_i(H_1)). \tag{4.58}$$

The region measures are calculated by adding up the entropies of the reference points belonging to the region,

$$\mu_r(i) = \sum_{p_{\text{ref},j}\in r_i} \mu_p(p_{\text{ref},j}). \tag{4.59}$$

Conceptually, the reference point entropy measures the expected surprise of sampling from the reference point $p_{ref,i}$, and the region measure is the maximum amount of surprise the robot can get by sampling from the region. Combining this with the REMC, the robot travels to the region with a high probability of being frequently surprised.

Appendix 4.B Proof of Theorem 2

Proof. Since P is a symmetric stochastic matrix, the eigenvalues of P, $\lambda(P)$, are real and lies within the domain $[-1, 1]$, and it can be diagonalized as $P = V\Lambda U^\mathsf{T}$, where Λ is a diagonal matrix of all the eigenvalues of P, V is a matrix with its column being the right eigenvectors of P, and U is a matrix with its column being the left eigenvectors of P ordered according to Λ. Moreover, f_w is a power series, and the set of eigenvalues $\lambda(P)$ lies within its domain, $f_w(P)$, obtained by replacing the scalar λ in (4.14) with the matrix P,

$$f_w(P) \triangleq \lim_{K\to\infty} \frac{1}{\sum_{k=0}^{K-1} w_k} \left(\sum_{k=0}^{K-1} w_k P^k \right) \tag{4.60}$$

exists, has the same eigenvectors as P, and the eigenvalues of $f_w(P)$ are the scalar mapping $f_w(\lambda(P))$ of the eigenvalues $\lambda(P)$ of the matrix P [50, Definition 1.2. on p.3], i.e.,

$$f_w(P) = V f_w(\Lambda) U^\mathsf{T}. \tag{4.61}$$

With (4.61), the numerator in (4.17) can be written as

$$\begin{aligned} &\|\mathbb{E}[\hat{\rho}_w] - \bar{\rho}\|_2 \\ &\quad = \|f_w(P)\rho_0 - \bar{\rho}\|_2 = \quad \left\|V f_w(\Lambda) U^\mathsf{T} \rho_0 - \bar{\rho}\right\|_2 \\ &\quad = \left\| f_w(\lambda_1) v_1 u_1^\mathsf{T} \rho_0 + \left(\sum_{i=2}^{n} f_w(\lambda_i) v_i u_i^\mathsf{T} \rho_0 \right) - \bar{\rho} \right\|_2 , \end{aligned} \tag{4.62}$$

where $\lambda_1 = 1$ is the largest eigenvalue of the stochastic matrix P, $u_1 = \mathbf{1}$ is the associated left eigenvector and $v_1 = \bar{\rho}$. This results in

$$\begin{aligned} &\|\mathbb{E}[\hat{\rho}_w] - \bar{\rho}\|_2 \\ &\quad = \left\| f_w(1) \bar{\rho} \mathbf{1}^\mathsf{T} \rho_0 + \left(\sum_{i=2}^{n} f_w(\lambda_i) v_i u_i^\mathsf{T} \rho_0 \right) - \bar{\rho} \right\|_2 . \end{aligned} \tag{4.63}$$

Using $\mathbf{1}^\mathsf{T}\rho_0 = 1$ for any probability-distribution vector, and $f_w(1) = 1$ by construction in (4.14), the previous expression can be rewritten as

$$\begin{aligned}\|\mathbb{E}[\hat{\rho}_w]-\bar{\rho}\|_2 &= \left\|\bar{\rho}+\left(\sum_{i=2}^{n} f_w(\lambda_i) v_i u_i^{\mathsf{T}}\rho_0\right)-\bar{\rho}\right\|_2 \\ &= \left\|\sum_{i=2}^{n} f_w(\lambda_i) v_i u_i^{\mathsf{T}}\rho_0\right\|_2 = \left\|V_\perp f_w(\Lambda_\perp) U_\perp^{\mathsf{T}}\rho_{0\perp}\right\|_2 \\ &= \left\|f_w\left(P|_{u_1^\perp}\right)\rho_{0\perp}\right\|_2 .\end{aligned} \tag{4.64}$$

Here, $P|_{u_1^\perp}$ is the restriction of P in the subspace orthogonal to u_1, $\rho_{0\perp}$ is the component of ρ_0 that is orthogonal to u_1, and $V_\perp$, $U_\perp$, $\Lambda_\perp$ are the associated restrictions of V, U, Λ in (4.62), respectively. Then, the NED objective in (4.17) can be written using (4.64) as

$$J_{\mathbb{E}} = \max_{\rho_0}\frac{\|\mathbb{E}[\hat{\rho}_w]-\bar{\rho}\|_2}{\|\rho_0-\bar{\rho}\|_2} = \max_{\rho_{0\perp}}\frac{\left\|f_w\left(P|_{u_1^\perp}\right)\rho_{0\perp}\right\|_2}{\|\rho_{0\perp}\|_2}. \tag{4.65}$$

Since the set of $\rho_{0\perp}=\epsilon x$, where ϵ is a sufficiently small scaling factor, spans the orthogonal space (see Lemma 5 in Appendix 4.C), the right hand side of (4.65) is equivalent to the induced matrix norm as

$$\begin{aligned}\max_{\rho_{0\perp}}\frac{\left\|f_w\left(P|_{u_1^\perp}\right)\rho_{0\perp}\right\|_2}{\|\rho_{0\perp}\|_2} &= \max_{\|x\|_2=1}\frac{\left\|f_w\left(P|_{u_1^\perp}\right)\epsilon x\right\|_2}{\|\epsilon x\|_2} \\ &= \max_{\|x\|_2=1}\left\|f_w\left(P|_{u_1^\perp}\right)x\right\|_2 = \left\|f_w\left(P|_{u_1^\perp}\right)\right\|_2 .\end{aligned} \tag{4.66}$$

Moreover, $f_w(P|_{u_1^\perp})$ is positive semi-definite since, from symmetry of P, $P|_{u_1^\perp}$ is symmetric. Therefore, its 2-norm is equal to the maximum eigenvalue, i.e.,

$$\min_P\left\|f_w\left(P|_{u_1^\perp}\right)\right\|_2 = \min_P \lambda_{\max}\left(f_w\left(P|_{u_1^\perp}\right)\right) \tag{4.67}$$

and since $f_w(\lambda)$ is strictly monotonic in λ, the eigenvalue of $f_w\left(P|_{u_1^\perp}\right)$ is maximized if and only if the eigenvalue of $P|_{u_1^\perp}$ is maximized, i.e.,

$$\min_P \lambda_{\max}\left(f_w\left(P|_{u_1^\perp}\right)\right) = f_w\left(\min_P \lambda_{\max}\left(P|_{u_1^\perp}\right)\right) \tag{4.68}$$

where $\lambda_{\max}\left(P|_{u_1^\perp}\right)=\max_{i\neq 1}\lambda_i(P)$. From (4.65)-(4.68),

$$\arg\min_P\left(\left\|f_w\left(P|_{u_1^\perp}\right)\right\|_2\right) = \arg\min_P\left(\max_{i\neq 1}\lambda_i(P)\right) \tag{4.69}$$

leading to the theorem's result in (4.18). □

Appendix 4.C Span of initial distributions

Remark 5. *The set of all n-dimensional probability vector forms an $(n-1)$ dimensional unit simplex, denoted by Δ^{n-1}.*

$$\Delta^{n-1} \triangleq \{\rho \in \mathbb{R}^n | \boldsymbol{1}^\mathsf{T} \rho = 1, \quad \rho \succeq 0\} \tag{4.70}$$

The hyperplane formed by the difference between any two members of Δ^{n-1} is orthogonal to the first left eigenvector $u_1 = \boldsymbol{1}$ of any stochastic matrix P. This hyperplane is denoted $u_1^\perp$. A 3-D graphical example is illustrated in Fig. 4.21. Additionally, if P is symmetric, then $v_1 \sim u_1$ and $u_1^\perp = v_1^\perp$

The following lemma shows that for a uniform stationary distribution $\frac{1}{n}\mathbf{1}$, there always exist a distribution ρ such that the orthogonal components on the simplex $\rho_\perp$ spans the hyperplane $u_1^\perp$.

Lemma 5. *Any nonzero vector x in the orthogonal space of $\boldsymbol{1}$ is the scaling of the orthogonal component of a distribution ρ.*

Proof. Consider an arbitrary non-zero vector x orthogonal to $\mathbf{1}$

$$x \in \{x | \mathbf{1}^\mathsf{T} x = 0\} \tag{4.71}$$

which implies that x contains both positive and negative values to sum to zero. Therefore, the minimum term of x is less than zero,

$$\min_i(x_i) < 0. \tag{4.72}$$

Consider x_ϵ, which is pointing in the same direction of x given by

$$x_\epsilon = \epsilon x \tag{4.73}$$

where

$$0 < \epsilon \leq \frac{1}{n|\min(x)|}. \tag{4.74}$$

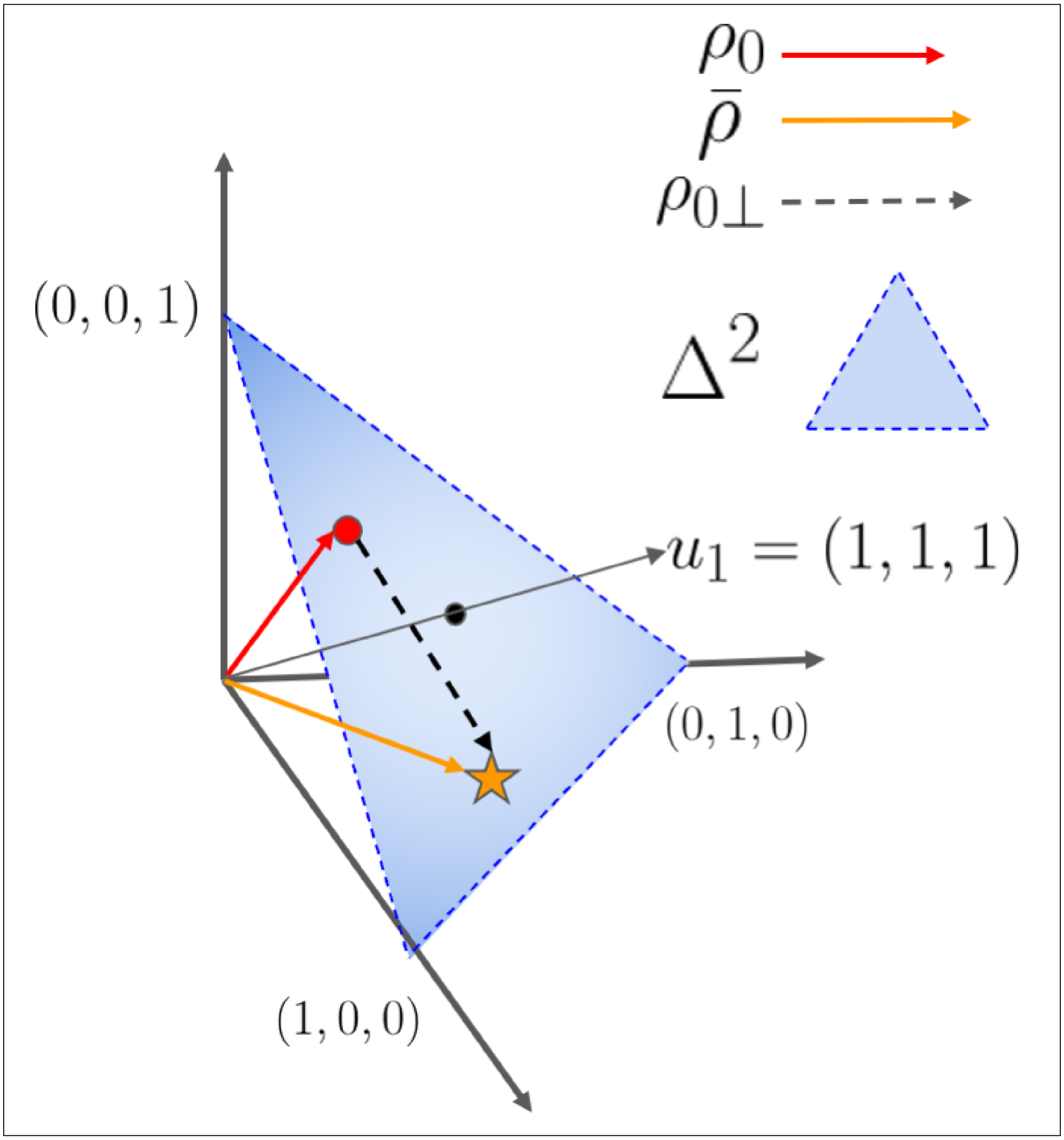


Figure 4.21: Graphical illustration of a simplex for a 3-node graph, such as the one in Fig. 4.4, which contains all valid distributions, ρ, and closed under transformation by any stochastic matrix, P. u_1 is the first left eigenvector of the stochastic matrix P; the blue triangle is the 2-D unit simplex, Δ^2; red arrow is the arbitrary initial distribution, ρ_0; orange arrow is the target distribution, $\bar{\rho}$; and the dashed arrow is the difference between the target distribution and the initial distribution, which is always parallel to the simplex.

Then, terms of $\rho = \epsilon x + \frac{1}{n}\mathbf{1}$ are non-negative, i.e.,

$$\rho = \epsilon x + \frac{1}{n}\mathbf{1} \succeq 0. \tag{4.75}$$

Moreover, ρ sums to one from (4.71)

$$\mathbf{1}^{\mathsf{T}}\rho = \mathbf{1}^{\mathsf{T}}\epsilon x + \mathbf{1}^{\mathsf{T}}\frac{1}{n}\mathbf{1} = \mathbf{1}^{\mathsf{T}}\frac{1}{n}\mathbf{1} = 1. \tag{4.76}$$

and is therefore a valid distribution, and its orthogonal component $\rho_\perp$ is a scaling of x i.e.,

$$\rho_\perp = \rho - \frac{1}{n}\mathbf{1} = \epsilon x. \tag{4.77}$$

□

Appendix 4.D Proof of Corollary 2.2

The normalized ergodic deviation in (4.17) can be rewritten as

$$\begin{aligned}
&\max_{\rho_0} \frac{\|Q^{1/2}(\mathbb{E}[\hat{\rho}_w] - \bar{\rho})\|_2}{\|Q^{1/2}\left(\rho_0 - \bar{\rho}\right)\|_2} \\
&\quad = \max_{\rho_0} \frac{\left\| Q^{1/2} f_w\left(P|_{u_1^\perp}\right)\rho_{0\perp}\right\|_2}{\|Q^{1/2}\rho_{0\perp}\|_2} \\
&\quad = \max_{\rho_0} \frac{\left\| Q^{1/2} f_w\left(P|_{u_1^\perp}\right)Q^{-1/2}Q^{1/2}\rho_{0\perp}\right\|_2}{\|Q^{1/2}\rho_{0\perp}\|_2} \\
&\quad = \max_{\rho_0} \frac{\left\| f_w\left(Q^{1/2}P|_{u_1^\perp}Q^{-1/2}\right)Q^{1/2}\rho_{0\perp}\right\|_2}{\|Q^{1/2}\rho_{0\perp}\|_2} \\
&\quad = \max_{\rho_0} \frac{\left\| f_w\left(\tilde{P}|_{u_1^\perp}\right)\tilde{\rho}_{0\perp}\right\|_2}{\|\tilde{\rho}_{0\perp}\|_2}
\end{aligned} \tag{4.78}$$

with

$$\begin{aligned}
\tilde{\rho} &= Q^{1/2}\rho = \Pi^{-1/2}\rho, \\
\tilde{P} &= Q^{1/2}PQ^{-1/2} = \Pi^{-1/2}P\Pi^{1/2}
\end{aligned} \tag{4.79}$$

where $\Pi = \operatorname{diag}(\bar{\rho})$ is invertible and its square root exists because $\bar{\rho}$ is strictly positive. Note that $\tilde{P}$ is symmetric since the detailed balance condition in (4.22), results in

$$\Pi^{-1/2}P\Pi^{1/2} = \Pi^{1/2}P^{\mathsf{T}}\Pi^{-1/2} \tag{4.80}$$

i.e., $\tilde{P} = \tilde{P}^{\mathsf{T}}$, which also implies that the restriction of $\tilde{P}$ to the subspace orthogonal to u_1, i.e., $\tilde{P}|_{u_1^\perp}$ is symmetric, since all the eigen-components, $\lambda_i v_i u_i^{\mathsf{T}}$, of $\tilde{P}$ are symmetric. Then, following arguments as in Eqs. (4.65)- (4.69) in the proof of Theorem 2,

$$\begin{aligned}\arg\min_{P}\max_{\rho_0} & \frac{\left\| f_w\left(\tilde{P}|_{u_1^\perp}\right)\tilde{\rho}_{0\perp}\right\|_2}{\|\tilde{\rho}_{0\perp}\|_2} \\ &= \arg\min_{P}\lambda_{\max}(\tilde{P}|_{u_1^\perp}).\end{aligned} \tag{4.81}$$

The corollary follows since a coordinate transform preserves eigenvalues and, therefore,

$$\lambda_{\max}(\tilde{P}|_{u_1^\perp}) = \max_{i\neq 1}\lambda_i(\tilde{P}) = \max_{i\neq 1}\lambda_i(P). \tag{4.82}$$

Appendix 4.E Proof of Lemma 3

Proof of Lemma 3: The optimality of even finite horizon ergodicity can be established by showing that the even-finite horizon time average can be written as a special case of the discounted-time average, and that the corresponding discounted-time average operator $f_w(x)$ is (Step i) positive semi-definite and (Step ii) monotonically increasing with respect to x within the domain $x \in [-1, 1]$. Then the requirements for f_w in Theorem 2 (and Corollary 2.2) are satisfied and optimality is guaranteed.

Consider the weighting function corresponding to the even-term finite horizon ergodicity

$$w_k = \begin{cases} 1 & \text{if } k < K \\ 0 & \text{if } k \geq K \end{cases} \tag{4.83}$$

where K is an even number. Then, the discounted time-average operator in (4.14) is

$$f_w(x) = \frac{1}{K}(1 + x + x^2 + \cdots x^{K-1}). \tag{4.84}$$

(Step i) Positive semi-definiteness: From the sum of a finite geometric series, (4.84) can be rewritten as

$$\begin{aligned} f_w(x) &= \frac{1}{K}(1 + x + x^2 + \cdots x^{K-1}) \\ &= \frac{1}{K}\frac{1 - x^K}{1 - x}. \end{aligned} \tag{4.85}$$

Since $0 \leq x^K < 1$, $0 < 1 - x^K$ the above expression is always positive, i.e., $f_w(x) > 0,\ \forall x \in [-1, 1)$. Also, for $x = 1$, from (4.84)

$$f_w(1) = 1 > 0. \tag{4.86}$$

Thus, the discounted time-average operator $f_w(x)$ satisfies the positive semi-definite condition in the domain $x \in [-1, 1]$.

(Step ii) Monotonicity: $f_w(x)$ can be shown to be monotonically increasing with respect to x by noting that its derivative is positive within the domain. This follows from the facts that (i) the derivative is positive at the boundaries and (ii) the minima of the derivatives in the domain are also positive ensuring that the continuous function cannot be negative anywhere in the domain. From (4.85),

$$\begin{aligned} \frac{d}{dx} f_w(x) &= \frac{d}{dx} \frac{1}{K} \frac{1 - x^K}{1 - x} \\ &= \frac{1}{K} \frac{(K-1)x^K - Kx^{K-1} + 1}{(1-x)^2} \end{aligned} \tag{4.87}$$

Note that the derivative $\frac{d}{dx} f_w$ is positive at the boundaries, since at $x = -1$, from (4.87),

$$\begin{aligned} \frac{d}{dx} f_w(x)\bigg|_{x=-1} &= \frac{1}{K} \frac{(K-1) + K + 1}{(2)^2} \\ &= \frac{1}{2} > 0. \end{aligned} \tag{4.88}$$

and, from (4.84) at $x = 1$

$$\begin{aligned} \frac{d}{dx} f_w(x)\bigg|_{x=1} &= \frac{1 + 2x^2 + \cdots (K-1)x^{K-2}}{K}\bigg|_{x=1} \\ &= \frac{1 + 2 + \cdots (K-1)}{K} \\ &= \frac{K-1}{2} > 0. \end{aligned} \tag{4.89}$$

Next, it is shown that the mimima of the derivative $\frac{d}{dx} f_w$ in (4.87) is positive inside the domain $x \in [-1, 1)$. Since the denominator $(1-x)^2$ is positive the derivative $\frac{d}{dx} f_w$ is positive if the numerator

$$N_{f_w}(x) = (K-1)x^K - Kx^{K-1} + 1 \tag{4.90}$$

is positive.

Note that at $x = -1$ the numerator is positive $N_{f_w}(-1) = 2K > 0$. Although at $x = 1$ the numerator is zero $N_{f_w}(1) = 0$, it is shown below that the numerator cannot be zero anywhere else since its value at the only other potential extrema within $x \in [-1, 1)$ is positive and the the values at the boundaries $x = 1, x = -1$ are non-negative.

Potential local extrema locations can be found by setting the derivative of the numerator to zero,

$$\begin{aligned} & \frac{d}{dx}\big((K-1)x^K - Kx^{K-1} + 1\big) \\ & \quad = K(K-1)x^{K-1} - K(K-1)x^{K-2} \\ & \quad = 0 \\ \implies & \quad x^{K-1} - x^{K-2} = x^{K-2}(x-1) = 0 \end{aligned} \tag{4.91}$$

resulting in a minima of the numerator $N_{f_w}(x)$ potentially occurring at two distinct points, $x = 0$ within $x \in [-1, 1)$ and another potential minima, outside, at $x = 1$. Specifically, at the only potential local minima location $x = 0$ within $x \in [-1, 1)$, the derivative of the numerator $N_{f_w}(x)$ is positive,

$$(K-1)0^K - K0^{K-1} + 1 = 1. \tag{4.92}$$

Thus, the numerator $N_{f_w}(x)$ remains positive within $x \in [-1, 1)$ and becomes zero only at $x = 1$. As a result, the derivative of $f_w(x)$ is positive in the domain $x \in [-1, 1]$ resulting in $f_w(x)$ strictly monotonically increasing in $x \in [-1, 1)$.

Since the discounted-time average operator $f_w(x)$ associated with the even-K finite horizon time average satisfies the condition of Corollary 2.2, Corollary 2.2 can be applied to the even-K finite time ergodicity and the Markov chain obtained from (4.23) is optimally ergodic.

Appendix 4.F Proof of Theorem 4

Proof. With the factorial weighting $w_k = \frac{1}{k!}$ in (4.15), the associated discounted-time-average operator f_w in (4.16) is the exponential function,

$$f_w(P) = \frac{1}{e}\exp(P) \tag{4.93}$$

and therefore, $f_w(P)$ exists for any square matrix P. Using the steps in (4.78), the left hand side of (4.26) can be rewritten as

$$\begin{aligned} J_{\mathbb{E}} &= \max_{\rho_0} \frac{\|\Pi^{-1/2}\left(\mathbb{E}[\hat{\rho}_w] - \bar{\rho}\right)\|_2}{\|\Pi^{-1/2}\left(\rho_0 - \bar{\rho}\right)\|_2} \\ &= \left\| f_w(\tilde{P}|_{u_1^\perp}) \right\|_2 = \left\| e^{-1}\exp(\tilde{P}|_{u_1^\perp}) \right\|_2 . \end{aligned} \tag{4.94}$$

This can be bounded by the following expression [31, p.34 Theorem 27]

$$\begin{aligned} &\left\| e^{-1}\exp(\tilde{P}|_{u_1^\perp}) \right\|_2 \\ &\quad \le \lambda_{\max}\left(e^{-1}\exp\left(\frac{1}{2}(\tilde{P}|_{u_1^\perp} + (\tilde{P}|_{u_1^\perp})^\mathsf{T})\right)\right) \\ &\quad = \overline{J}_{\mathbb{E}}(P). \end{aligned} \tag{4.95}$$

Since $(\tilde{P}|_{u_1^\perp} + (\tilde{P}|_{u_1^\perp})^\mathsf{T})$ is symmetric, the eigenvalues are real and the exponential of the eigenvalues is monotonic. Thus, the bound $\overline{J}_{\mathbb{E}}(P)$ can be minimized by selecting P as

$$\begin{aligned} &\arg\min_P \overline{J}_{\mathbb{E}}(P) \\ &\quad = \arg\min_P \lambda_{\max}\left(\exp\left(\frac{1}{2}(\tilde{P}|_{u_1^\perp} + (\tilde{P}|_{u_1^\perp})^\mathsf{T})\right)\right) \\ &\quad = \arg\min_P \exp\left(\lambda_{\max}\left(\frac{1}{2}(\tilde{P}|_{u_1^\perp} + (\tilde{P}|_{u_1^\perp})^\mathsf{T})\right)\right) \\ &\quad = \arg\min_P \lambda_{\max}\left(\frac{1}{2}(\tilde{P}|_{u_1^\perp} + (\tilde{P}|_{u_1^\perp})^\mathsf{T})\right). \end{aligned} \tag{4.96}$$

The symmetric part of $\tilde{P}|_{u_1^\perp}$ is related to the symmetric part of $\tilde{P}$ by

$$\begin{aligned}
&\frac{1}{2}\left(\tilde{P}+\tilde{P}^{\mathsf{T}}\right)\\
&=\frac{1}{2}\left(\Pi^{-1/2}P\Pi^{1/2}+(\Pi^{-1/2}P\Pi^{1/2})^{\mathsf{T}}\right)\\
&=\frac{1}{2}\Big(\Pi^{-1/2}(\bar{\rho}\mathbf{1}^{\mathsf{T}}+P|_{u_1^\perp})\Pi^{1/2}\\
&\qquad\qquad+(\Pi^{-1/2}(\bar{\rho}\mathbf{1}^{\mathsf{T}}+P|_{u_1^\perp})\Pi^{1/2})^{\mathsf{T}}\Big)\\
&=\frac{1}{2}\left((\bar{\rho}^{1/2}\bar{\rho}^{\mathsf{T}/2}+\tilde{P}|_{u_1^\perp})+(\bar{\rho}^{1/2}\bar{\rho}^{\mathsf{T}/2}+\tilde{P}|_{u_1^\perp})^{\mathsf{T}}\right)\\
&=\bar{\rho}^{1/2}\bar{\rho}^{\mathsf{T}/2}+\frac{1}{2}\left(\tilde{P}|_{u_1^\perp}+(\tilde{P}|_{u_1^\perp})^{\mathsf{T}}\right).
\end{aligned} \tag{4.97}$$

Furthermore, the symmetric part of $\tilde{P}$ has a eigenpair of $\lambda_1 = 1$ and $u_1 = v_1 = \bar{\rho}^{1/2}$,

$$\begin{aligned}
&\frac{1}{2}(\tilde{P}+\tilde{P}^{\mathsf{T}})\bar{\rho}^{1/2}\\
&\quad=\frac{1}{2}(\Pi^{-1/2}P\Pi^{1/2}+\Pi^{1/2}P^{\mathsf{T}}\Pi^{-1/2})\bar{\rho}^{1/2}\\
&\quad=\frac{1}{2}(\Pi^{-1/2}P\bar{\rho}+\Pi^{1/2}P^{\mathsf{T}}\mathbf{1})\\
&\quad=\frac{1}{2}(\Pi^{-1/2}\bar{\rho}+\Pi^{1/2}\mathbf{1})\\
&\quad=\frac{1}{2}(\bar{\rho}^{1/2}+\bar{\rho}^{1/2})\ =\bar{\rho}^{1/2}.
\end{aligned} \tag{4.98}$$

Then, $\bar{\rho}^{1/2}\bar{\rho}^{\mathsf{T}/2}$ in (4.97) is an eigen-component of the symmetric part of $\tilde{P}$, thus, the maximum eigenvalue of $\left(\frac{1}{2}(\tilde{P}|_{u_1^\perp}+(\tilde{P}|_{u_1^\perp})^{\mathsf{T}})\right)$ is the largest eigenvalue of the symmetric part of $\tilde{P}$ other than $\lambda_1 = 1$

$$\begin{aligned}
&\lambda_{\max}\left(\frac{1}{2}\left(\tilde{P}|_{u_1^\perp}+(\tilde{P}|_{u_1^\perp})^{\mathsf{T}}\right)\right)\\
&\qquad=\max_{i\neq 1}\lambda_i\left(\frac{1}{2}\left(\tilde{P}+\tilde{P}^{\mathsf{T}}\right)\right).
\end{aligned} \tag{4.99}$$

□

Appendix 4.G Proof of Corollary 4.1

Proof. By Perron-Forbenius theorem, for any irreducible non-negative matrix, there can only be one non-negative eigenvector, and such eigenvector is associated with the largest eigenvalue λ_1 [18]. Since $\bar{\rho}^{1/2}$ is a non-negative eigenvector of $\frac{1}{2}\left(\tilde{P} + \tilde{P}^{\mathsf{T}}\right)$ with associated eigenvalue of 1, then 1 is the largest eigenvalue of $\frac{1}{2}\left(\tilde{P} + \tilde{P}^{\mathsf{T}}\right)$. Therefore similar to previous theorems, the second largest eigenvalue of $\left(\frac{1}{2}(\tilde{P}|_{\tilde{v}_1^{\perp}} + (\tilde{P}|_{\tilde{v}_1^{\perp}})^{\mathsf{T}})\right)$ can be access by the deflation

$$\begin{aligned}
&\lambda_{\max}\left(\frac{1}{2}(\tilde{P}|_{\tilde{v}_1^{\perp}} + (\tilde{P}|_{\tilde{v}_1^{\perp}})^{\mathsf{T}})\right) \\
&\quad = \max_{i\neq 1} \lambda_i \left(\frac{1}{2}(\tilde{P} + (\tilde{P})^{\mathsf{T}})\right) \\
&\quad = \lambda_{\max}\left(\frac{1}{2}(\tilde{P} + (\tilde{P})^{\mathsf{T}}) - 2\bar{\rho}^{1/2}\bar{\rho}^{\mathsf{T}/2}\right).
\end{aligned} \tag{4.100}$$

□

Appendix 4.H Variance of REMC

As REMC is a stochastic method, one might be interested in the robustness of the coverage generated by such a method. The variance of the (finite) time average of a Markov chain with a general stochastic matrix P is still an open problem, often found in the Markov chain Monte Carlo literature.

More precisely, we are interested in the variance at each time step k with respect to the expected $\hat{\rho}_k$, where

$$\begin{aligned}
\hat{\rho}_k &= \frac{1}{K}\sum_{k=0}^{K-1} I(r[k]) \\
\mathbb{E}[\hat{\rho}_k] &= \frac{1}{K}\sum_{k=0}^{K-1} P^k \rho_0 \\
\mathrm{Var}[\hat{\rho}_k] &= \mathbb{E}[(\hat{\rho}_k - \mathbb{E}[\hat{\rho}_k])(\hat{\rho}_k - \mathbb{E}[\hat{\rho}_k])^{\mathsf{T}}].
\end{aligned} \tag{4.101}$$

The Markov chain central limit theorem (CLT) states that the variance of the time average is proportional to $\frac{1}{k}$, i.e.,

$$\mathrm{Var}(\hat{\rho}_k) \propto \frac{1}{k} \tag{4.102}$$

as k becomes large [61].

To verify the approximation, a simulation is conducted using the graph from Fig. 4.3 with target distribution $\bar{\rho} = \frac{1}{n}\mathbf{1}$ and initial condition $r[0] = 1$. $m = 1,000$ trials are conducted

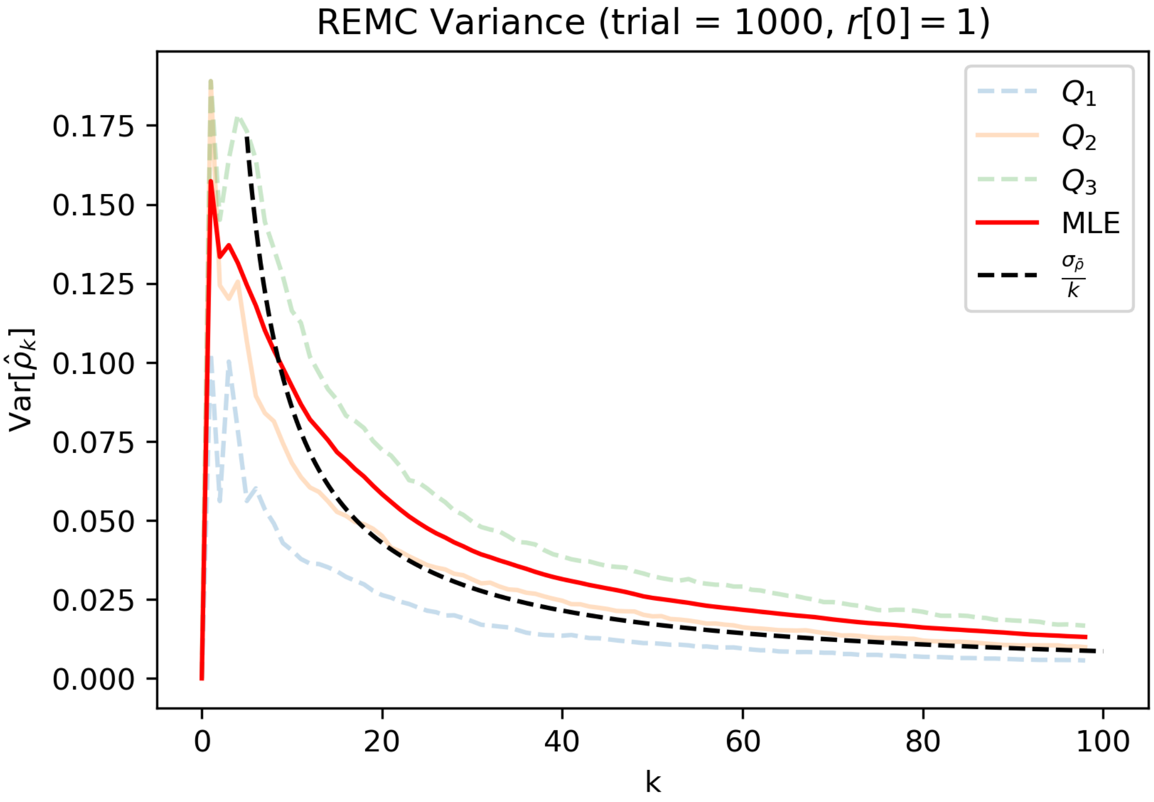


Figure 4.22: Variance of the time average of REMC at each time step k over $1,000$ trials with initial condition $r[0] = 1$ and a uniform target distribution $\bar{\rho} = \frac{1}{n}\mathbf{1}$. Red curve is the maximum likelihood estimation (MLE) of the variance and black dashed line is the variance estimated by Markov chain central limit theorem (CLT).

and variance is computed using maximum likelihood estimation (MLE) as

$$\operatorname{tr}(\operatorname{Var}[\hat{\rho}_k]) \approx \frac{1}{m}\sum_{j=1}^{m}(\hat{\rho}_{k,j} - \mathbb{E}[\hat{\rho}_k])^{\mathsf{T}}(\hat{\rho}_{k,j} - \mathbb{E}[\hat{\rho}_k]). \tag{4.103}$$

The CLT variance is calculated assuming i.i.d. samples to be $\operatorname{tr}(\operatorname{Var}(\hat{\rho}_k)) \approx \frac{1}{k}\sum_i^n \bar{\rho}_i(1-\bar{\rho}_i)$. The simulation results are shown in Fig. 4.22.

It is observed that the MLE variance has the general shape of $\frac{1}{k}$ roughly after 20 steps, which shows a rapid decrease in the variance. The CLT prediction has a faster decrease, as expected from [61], with the MLE converging to it as k grows. Interestingly, the median (Q_2) values are closely approximated by the CLT predictions. Therefore, we can conclude that REMC variance decreases rapidly as the number of steps increases, and the results are closely predicted by the expected value.

Appendix 4.I 3 Nodes Simplex Example

Example 4.I.1. *For any stochastic matrix P, the dynamics is closed in the simplex Δ^{n-1}, i.e. the distribution vector stays as distribution vector when multiplied by P. From that, the dynamic of the Markov chain can be studied by restricting the P in the subspace of the simplex, resulting in $P|_{u_1^\perp}$, and the dynamic is operating on the orthogonal component $\rho_\perp = \rho - \bar{\rho}$. Figure 4.23 shows the difference of the trajectory generated by FMMC and REMC, with the simplex transformed to a 2-D space. The figure shows the trajectory of the distribution (red) and the trajectory of the time average (blue) for four time steps. As shown in the plots, the time average, $\hat{\rho}$, of REMC converges faster than that of FMMC, even though the FMMC distribution ρ converges to the target distribution, $\bar{\rho}$ faster than the REMC case.*

Appendix 4.J FMMC vs REMC on Erdős–Rényi Graphs

The following experiment extends the experiment in Section 4.5.1 to cover a more diverse set of directed graphs. In particular, we consider graphs that are generated by the Erdős–Rényi-Gilbert model [37]. Each directed graph has 2 random components: the number of nodes and the existence of edges between the nodes. The number of nodes n is generated by uniformly

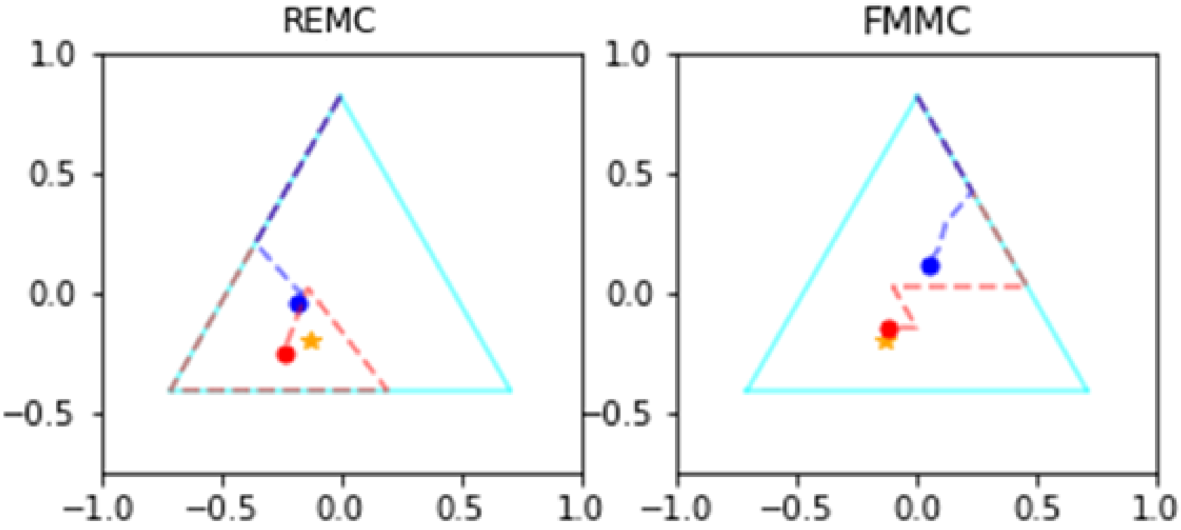


Figure 4.23: Comparison of REMC and FMMC. REMC has a faster convergence for the time average, $\hat{\rho}$ of the distribution ρ_k, compared to FMMC even though the distribution ρ_k itself converges to the target distribution $\bar{\rho}_k$ faster with FMMC. Here, the cyan line is the boundary of the simplex; the red marker is the distribution ρ_k at current step $k = 4$; the red dashed line is the trajectory, $\rho_{0:k}$; the blue marker is the time average $\hat{\rho}_k$ at current step, k; and the blue dashed line is the time-averaged trajectory, $\hat{\rho}_{0:k}$.

sampling from 5 to 50. Edges between nodes are randomly generated with an independent probability of $p = 4\ln(n)/n$.[3] All self-loops are generated with a probability of 1. The rest of the setup is identical to section 4.5.1.

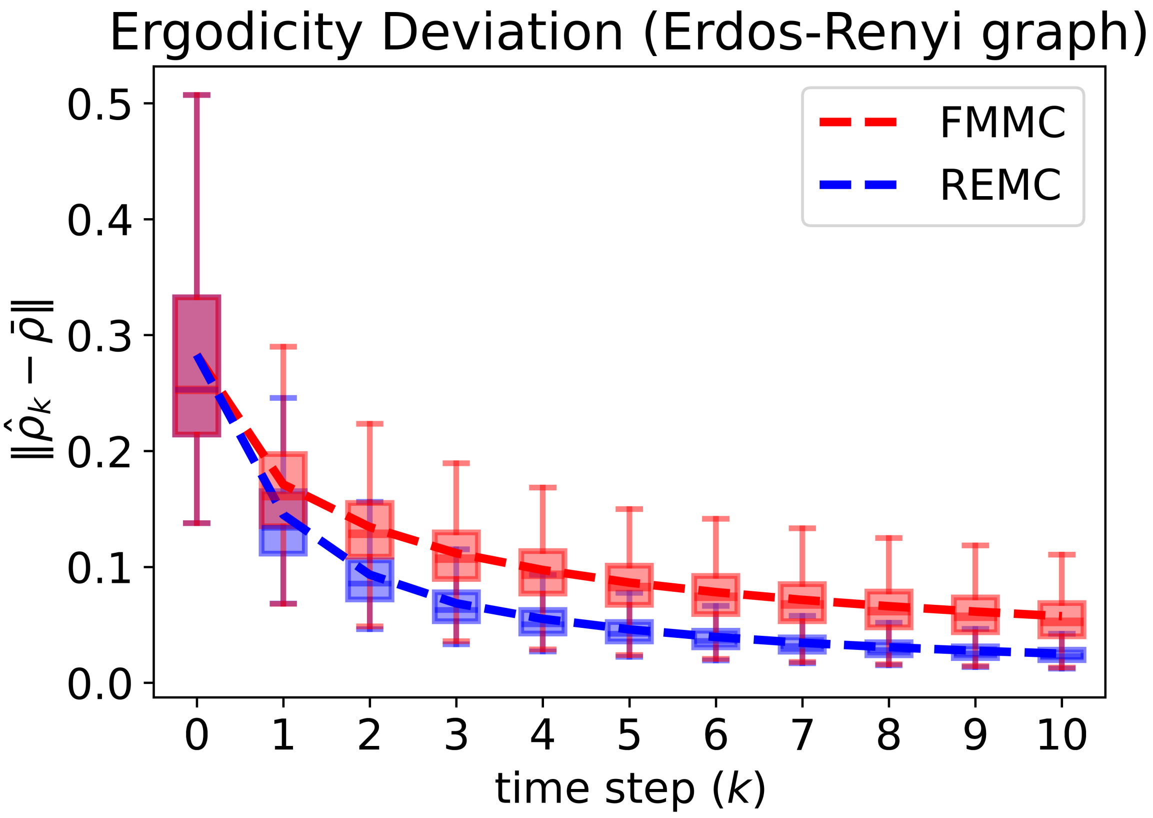


Figure 4.24: Ergodicity comparison between fastest mixing Markov chain (FMMC) and REMC on graphs generated from Erdos-Renyi model, showing that REMC has a significantly lower ergodicity error.

Figure 4.24 shows the expected ergodicity deviations in 10 time steps for FMMC and REMC separately for 1,000 trials. The results shows even stronger difference between FMMC and REMC, with REMC having significantly faster convergence rate in most trials. This

---

[3] $p = 2\ln(n)/n$ is the empirical critical probability for a graph to be almost surely strongly connected. It is inflated to $4\ln(n)/n$ for FMMC to have sufficient connectivity.

is a result of the tank-graph having a relatively low connectivity compared to the random-graphs, which constrained the minimum achievable λ_2. Whereas the λ_2 is able to achieve lower values in the generic random graphs (this also allows FMMC is minimize the SLEM to closer to 0).

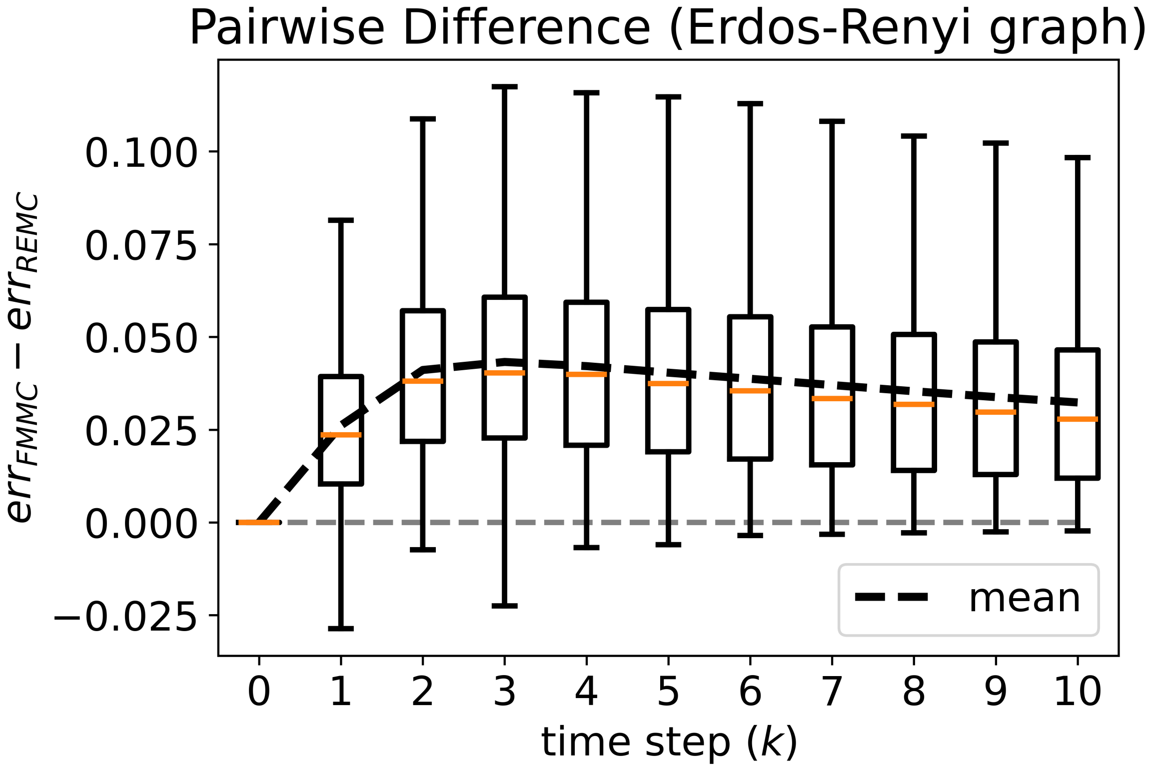


Figure 4.25: Boxplot of the pair-wise error differences for all the trials in Fig. 4.24. For the same initial condition and target distribution, REMC has significantly less error compared to FMMC.

Figure 4.25 shows the pairwise difference (FMMC - REMC) between the two methods for the same trials. The result shows that REMC is consistently more ergodic than FMMC as the first quartiles are positive at all time steps.

Chapter 5

SIMULATED ANNEALING FOR MULTI-ROBOT ERGODIC INFORMATION ACQUISITION USING GRAPH-BASED DISCRETIZATION

To fully utilize the power of ergodic search,the robots need to know the quality of information across the region graph. In this chapter we extended the ergodic traversal to a swarm robotic system in a Graph with unknown distribution of information. The main contribution is to achieve efficient information gathering in this scenario by applying simulated annealing to the estimated target distribution. This chapter is based on our published work in [116].

5.1 Problem Statement

5.1.1 Problem Formulation

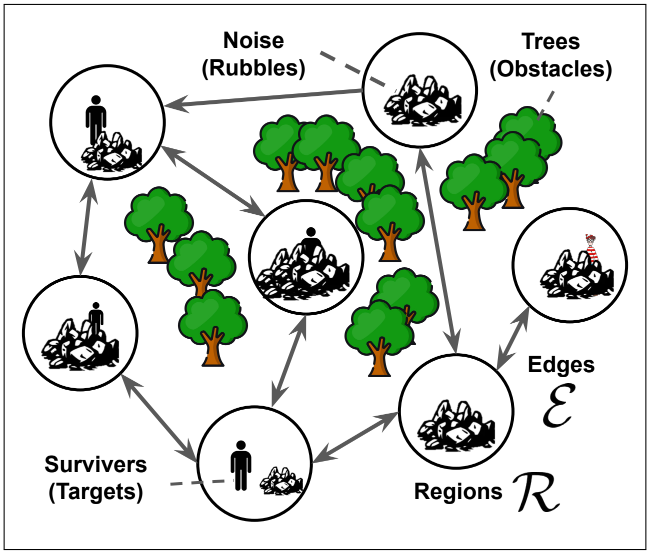


Figure 5.1: Example information gathering task of locating survivors in a map $\mathcal{G}$ with regions, modeled as set of nodes $\mathcal{R}$, and edges $\mathcal{E}$ caused by blockage of trees. The regions containing various amount of rubble, which causes differences in information quality i.e., noise levels σ_i^2, across the regions.

This article considers information gathering tasks that are scattered in a large area, with regions of interest separated by obstacles. A team of N robots has to estimate the states, x_i, internal to each region. An example task of finding survivors in disaster relief is shown in Fig. 5.1, where the states to be estimated are the survivors' locations in all the regions (in the case of scalar x_i, we assume there is exactly one survivor per region[1]).

The regions are modeled as a set of nodes $\mathcal{R}$. The connections between pairs of regions are represented by an edge, and $\mathcal{E}$ is the set of all the edges. Then, the tuple of nodes and edges forms the graph $\mathcal{G} = (\mathcal{R}, \mathcal{E})$. Information gathering is considered an estimate of the scalar state x_i for each region $r_i \in \mathcal{R}$. Each robot can collect a single observation from the region at every time step, with the noisy observation model

$$z_i = x_i + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_i^2). \tag{5.1}$$

The variance σ_i^2 varies between the regions but not between the robots, i.e., some regions have lower information quality per observation than others. The varying variance can be an effect from the lack of landmark for the robot to localize or high visual noise in the region that conceals the target. [2] The state x_i can be estimated from the observations by

$$\bar{x}_i = \frac{1}{\nu_i} \sum_{j=1}^{\nu_i} z_j \approx x_i, \quad \mathrm{Var}[\bar{x}_i] = \frac{\sigma_i^2}{\nu_i} \tag{5.2}$$

where ν_i is the number of observations acquired by the robots collectively. With the standard error, i.e. the variance of the estimation $\mathrm{Var}(\bar{x}_i)$, being a function of the number of observations ν_i and the regional noise σ_i^2 [3]

5.1.2 Optimal Distribution

To ensure no one region has more standard error than the others, the goal is to minimize the maximum posterior variance after any time horizon K, i.e. all regions are equally confident

---

[1] Multiple survivors can be modeled as a multivariate Gaussian distribution, which is outside of the scope for this document. Alternatively, it can be done by considering x_i as the number of survivors instead of the location of one survivor.

[2] The observation model can be considered as an abstraction of a robot collecting observations continuously within one region for a fix amount of time before transitioning, where z_i, σ_i^2 are the result of integrating all observation collected within the time step.

[3] By the central limit theorem, this is also extendable to a non-Gaussian distribution with a sufficiently large ν.

at all time,

$$\begin{aligned} \min_{\bar{\rho}} \quad & \max_{i} \left[\frac{\sigma_i^2}{KN\bar{\rho}_i} \right] \\ \text{s.t.} \quad & \mathbf{1}^T \bar{\rho} = 1 \\ & \bar{\rho} \succeq 0 \end{aligned} \tag{5.3}$$

where $\bar{\rho}_i$ is the relative visitation frequency of region i, and $\bar{\rho}$ is the target distribution, which is a vector with the i-th entry being $\bar{\rho}_i$; N is the number of robots. For a known sample variance σ^2, the optimal target distribution is

$$\bar{\rho}_i^* = \frac{1}{\sum_i \sigma_i^2} \sigma_i^2. \tag{5.4}$$

Then, by substituting $\bar{\rho}^*$ to $\bar{\rho}$ in (5.3), the variance of the estimation at any time step K is[4]

$$\mathrm{Var}[\bar{x}_i] = \frac{\sigma_i^2}{KN\bar{\rho}_i} = \frac{\sum_i \sigma_i^2}{KN} \tag{5.5}$$

which is the same for all regions. Hence, there exists a need for a planning strategy such that the target $\bar{\rho}$ can be reached under the constraint of the graph connectivity.

Moreover, the true sampling variance σ_i^2 are initially unknown to all the agents. The variance can also be estimated along side x_i as the robots start collecting samples, which is done by

$$\bar{\sigma}_i^2 = \frac{1}{\nu_i - 1} \sum_{j=1}^{\nu_i} (z_j - x_i)^2 \approx \sigma_i^2. \tag{5.6}$$

Similar to the mean $\bar{x}$, the estimated variance $\bar{\sigma}$ is unreliable at the beginning and are improved with increasing number of samples. As a result, $\bar{\rho}^*$ in (5.4) is unknown to the robots and the $\bar{\rho}$ generated by the estimated variance $\bar{\sigma}^2$ is unreliable and causes inefficient allocation. This motivates the research problem to develop a planning method that can utilize the estimated variance while accounting for the unreliable initial estimations. This can be considered as two subproblems in developing the planning method: (subproblem 1) to

---

[4] It can be observed that uniform values minimize the maximum value since to maintain a constant sum, any decrease of value in one region will cause an increase of equal amount in other regions, which increases the maximum value.

reach a given target distribution on a graph with a multi-robot system and (subproblem 2) to account for the unknown variance and unreliable estimates in the beginning.

5.2 *Methodology*

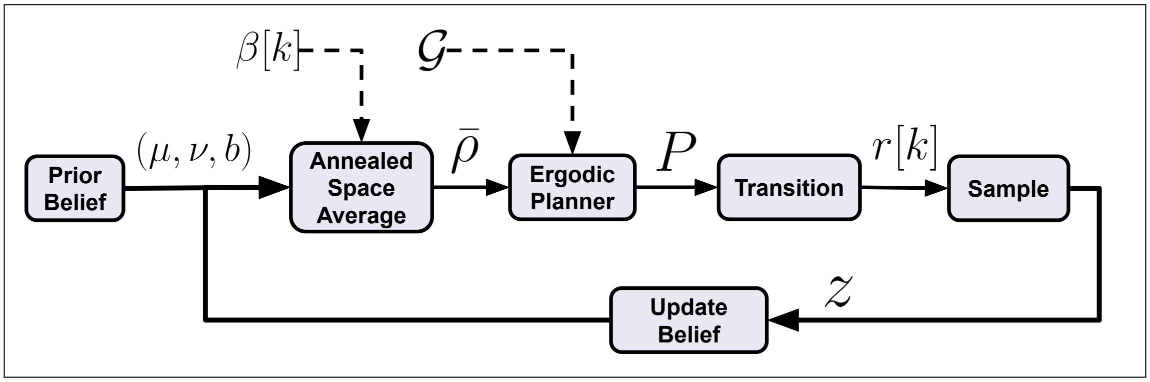


Figure 5.2: Flowchart of the annealed ergodic information gathering algorithm.

The problem of achieving the target distribution is solved by using *Rapidly Ergodic Markov Chain* (REMC), and the problem of unreliable initial estimated variance is solved by using *simulated annealing*. The solution methods are explained in Sections 5.2.1 and 5.2.2, respectively. The overall pipeline of the planning method is shown in Fig. 5.2.

5.2.1 Subproblem 1 Solution: Multi-Robot Ergodic Control

Ergodic control can solve the subproblem 1 to achieve the target distribution $\bar{\rho}$ for a given σ^2. In general, the goal for ergodic control is to synthesize a control law such that the dynamic system has a time average equal to the space average for almost all initial conditions. This is formulated in a graph space as

$$\lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} (F(r[k])) = \frac{1}{\mu(\mathcal{R})} \sum_{r_i \in \mathcal{R}} F(r_i)\mu(r_i) \tag{5.7}$$

where $r[\cdot]$ is the region trajectory; $\mathcal{R}$ is the set of regions; μ is a measure on the region set; and F is an arbitrary μ-measurable function. To measure visitation, F can be defined as the indicator function

$$I(r_i) \triangleq \begin{bmatrix} \delta_{1,i} & \delta_{2,i} & \cdots & \delta_{n,i} \end{bmatrix}^T \tag{5.8}$$

where $\delta_{i,j}$ is the Kronecker delta, and

$$\hat{\rho} \triangleq \lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} \left(I(r[k]) \right), \quad \bar{\rho} \triangleq \frac{1}{\mu(\mathcal{R})} \sum_{r_i \in \mathcal{R}} I(r_i)\mu(r_i). \tag{5.9}$$

The ergodic objective for a single agent has been shown to be achieved by generating the trajectory with *Markov chains* [115]. The Markov chain is generated by randomly sampling the next region based only on the current region, i.e. $r[k+1] \sim \mathbb{P}(R \mid r[k])$, with R being the random variable of the possible region. In a finite graph space, the transition probability can be represented by a stochastic matrix P. The time average can then be expressed in terms of expected value as

$$\begin{aligned} \mathbb{E}\left[\hat{\rho}\right] &= \lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} \left(I(\mathbb{E}[r[k]]) \right) \\ &= \lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} \left(P^k \rho[0] \right) \end{aligned} \tag{5.10}$$

with $\rho[0]$ as the initial distribution.

The transition matrix that guarantees ergodicity and also optimizes the convergence rate can be found by the following convex program:

$$\begin{aligned} \arg\min_{P} \quad & \lambda_{\max}\left(\frac{1}{2}\left(\tilde{P} + \tilde{P}^T \right) - 2\bar{\rho}^{1/2}\bar{\rho}^{T/2} \right) \\ \text{s.t.} \quad & \mathbf{1}^T P = \mathbf{1}^T \quad \text{(Stochastic } P\text{)} \\ & P\bar{\rho} = \bar{\rho} \quad \text{(Target Distribution)} \\ & P_{i,j} \geq 0 \quad \text{(Stochastic } P\text{)} \\ & P_{i,j} = 0 \quad \text{if} \quad (j,i) \notin \mathcal{E} \quad \text{(Transitions in } \mathcal{G}\text{)} \\ & \tilde{P} = \text{diag}(\bar{\rho}^{-1/2}) P \text{diag}(\bar{\rho}^{1/2}). \end{aligned} \tag{5.11}$$

As ergodicity is a property derived from ensemble systems, (5.7) is extensible to multi-robot system. When all the robots follow the same Markov chain, (5.9) is modified to account for the trajectories of all the robots as

$$\begin{aligned}\mathbb{E}[\hat{\rho}] &= \frac{1}{N}\sum_{a=1}^{N}\mathbb{E}[\hat{\rho}_a] \\ &= \frac{1}{N}\sum_{a=1}^{N}\left(\lim_{K\to\infty}\frac{1}{K}\sum_{k=0}^{K-1}P^k\rho_a[0]\right) \\ &= \lim_{K\to\infty}\frac{1}{K}\sum_{k=0}^{K-1}\left(P^k\frac{1}{N}\sum_{a=1}^{N}\rho_a[0]\right) \\ &= \lim_{K\to\infty}\frac{1}{K}\sum_{k=0}^{K-1}\left(P^k\rho[0]\right).\end{aligned} \tag{5.12}$$

Here, ρ_a denotes the distribution for agent a, and ρ is averaged over all the robots. Consequently, the more robots are in the team, the less variance the Markov chain would have. That is, the true trajectory follows more closely to the expected value

$$\frac{1}{N}\sum_{a=1}^{N} I(r_a[k]) \to P^k\rho[0] \quad \text{as} \quad N\to\infty. \tag{5.13}$$

5.2.2 Subproblem 2 Solution: Annealing for Space Average

To account for the second subproblem of unreliable initial variance, the target distribution is designed to focus on collecting information on the variance at the beginning and gradually shift toward the oracle solution as the variance becomes reliable. This is achieved by simulated annealing, where the robots starts at the most random configuration, i.e. uniform random, and shift toward the optimal solution by varying a temperature parameter.

To achieve this, the *Gibbs measure*[5] is chosen as the measure μ in (5.7), with

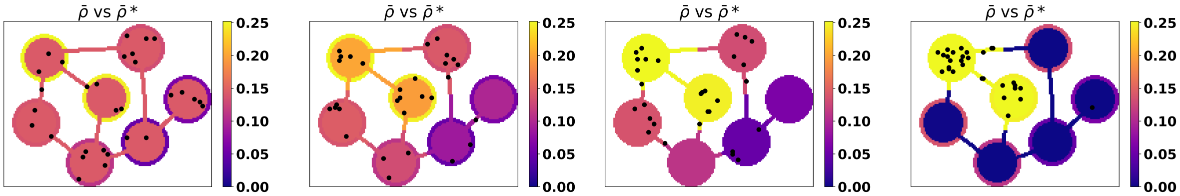
$$\mu(r_i) = \exp(-\beta E(r_i)) \tag{5.14}$$

where $E(r_i)$ is the energy associated with the state r_i, and β is the *coldness* (or *inverse temperature*) parameter. Then, since the region set is disjointed, the normalizing factor is

$$\mu(\mathcal{R}) = \sum_{i=1}^{n}\exp(-\beta E(r_i)). \tag{5.15}$$

---

[5] Also known as *Boltzmann distribution in finite space*



(a) $\beta = 0$ (b) $\beta = 0.5$ (c) $\beta = 1$ (d) $\beta = 10$

Figure 5.3: Example distribution of 30 robots (black markers) with various constant coldness β (after a sufficient amount of steps for reaching equilibrium ($K = 100$)

. The color of the border of the region represents the relative variance σ^2, the inner color represents the target distribution $\bar{\rho}(\beta)$, generated using the ground truth variance. (a) the robots are equally spread out regardless of the variance; (b) the robots are somewhat between the uniform distribution and the optimal distribution; (c) the distribution of robots are proportional to the variance, which is the optimal for information gathering; (d) the robots are concentrated at the two regions with the highest variance, this causes severe under-sampling in the rest of the regions.

In particular, we define the energy to be the negative (differential) entropy of the sample distribution as

$$E(r_i) = -\ln(2\pi e \bar{\sigma}_i^2). \tag{5.16}$$

The space average in (5.9) is then

$$\begin{aligned} \bar{\rho}(\beta) &= \frac{1}{Z(\beta)} \sum_{i=1}^{n} I(r_i) \exp(\beta \ln(\bar{\sigma}_i^2)) \\ Z(\beta) &= \sum_{i} \exp(\beta \ln(\bar{\sigma}_i^2)). \end{aligned} \tag{5.17}$$

An additional advantage of using the Gibbs measure is that the target distribution $\bar{\rho}$ is guaranteed to be reachable by an ergodic Markov chain, as all the entries are strictly positive for any real-valued energy level $E(r_i)$. More importantly, the coldness parameter β provides a smooth control on the uniformness of the target distribution, with $\beta = 0$ generating a uniform distribution regardless of the energy level; and $\beta \to \infty$ generating a delta function at the minimum energy state.

In our setup, the target distribution will assign more samples to regions with higher sampling entropy as β increases. Specifically, when $\beta = 1$, the target distribution is

$$\begin{aligned} \bar{\rho}(1) &= \frac{1}{Z(1)} \sum_{i=1}^{n} I(r_i) \exp(\ln(\bar{\sigma}_i^2)) \\ &= \frac{1}{Z(1)} \sum_{i=1}^{n} I(r_i) \bar{\sigma}_i^2. \end{aligned} \tag{5.18}$$

If $\bar{\sigma}^2 = \sigma^2$, then this is the estimation of the optimal sample size in (5.4). An example of the effect of the value of β is shown in Fig. 5.3. As a result, if β varies gradually from 0 to 1, i.e., annealed, the goal of transitioning from uniform to optimal is achieved. In this document, annealing is done by the first-order step response as[6]

$$\beta(k) = 1 - \exp(-\alpha k), \tag{5.19}$$

with k being the time step and α the cooling rate. The choice of α is discussed in Section 5.5.

---

[6] Different annealing schedule can be used, such as tanh, with similar result. Here first-order step response is chosen with its similarity to Newton's law of cooling[72, eq.(1.22)].

5.3 *Annealed Ergodic Algorithm*

The complete algorithm for applying the annealed ergodic information gathering is shown in Algorithm 3. Since ergodic control can run persistently, no time horizon has to be specified, and the only parameter required is the cooling rate α. The main algorithm is composed of two stages: a) sampling (lines 4 - 10), b) planning (lines 11 - 19). Using the equations in (5.2) and (5.6) to estimate the mean and variance requires the entire set of observations to be stored. Instead, it is updated sequentially in a Bayesian manner using the normal-inverse gamma parameterization (ν, μ, b) [86, Section 6], where μ is the mean, ν is the number of samples, and b is an auxiliary variable to recover the variance, which is done using the equation in line 12. This framework can also be used to assign prior knowledge for the mean and variance. We use an uninformed prior and except for a well-defined initial variance, the parameter ν is initialized at 1. Once the most recent estimated variance σ^2 is recovered from the parameters, the target distribution $\bar{\rho}$ is calculated using the current temperature $\beta[k]$. The transition matrix P is then obtained using the REMC algorithm. The robots randomly transition to the next region individually according to the transition matrix.

5.4 *Experiments*

In the section, the annealing algorithm is compared against uniform coverage

$$\bar{\rho}_{\text{uniform}} = [1/n, 1/n, \cdots] \tag{5.20}$$

and direct ergodic coverage

$$\bar{\rho}_{\text{direct}}(k) = \frac{1}{Z(1)} \sum_{i=1}^{n} I(r_i)\bar{\sigma}_i^2[k] \tag{5.21}$$

with different numbers of robots (5, 30) and ergodic Markov chain planning methods (REMC and FMMC).

Algorithm 3 Annealed Ergodic Information Gathering

```
Parameter: Cooling Rate: $\alpha$
Initialize $\nu \leftarrow \text{ones}(n), b \leftarrow \text{ones}(n),$
        $\mu \leftarrow \text{zeros}(n), k = 0$
while inspecting do
    for each robot $a$ do
        $z = \text{Sample}(r(a))$                                    ▷ Sample from region of $a$
        ▷ Update law for normal inverse-gamma dist.
        $b[r(a)] \leftarrow b[r(a)] + \frac{1}{2}\frac{n[r(a)]}{n[r(a)]+1}(z - \mu[r(a)])$
        $\mu[r(a)] \leftarrow \frac{n[r(a)]\mu[r(a)]+z}{n[r(a)]+1}$
        $\nu[r(a)] \leftarrow \nu[r(a)] + 1$
    end for
    for $r_i \in \mathcal{R}$ do
        $\sigma^2[i] \leftarrow 2b[i]\frac{\nu[i]+1}{(\nu[i])^2}$
    end for
    $\beta = 1 - \exp(-\alpha k)$
    $\bar{\rho} \leftarrow \exp(\beta \ln(\sigma^2))$                    ▷ Entry-wise exponential
    $\bar{\rho} \leftarrow \bar{\rho} / \sum_i \bar{\rho}_i$
    $P \leftarrow \text{REMC}(\bar{\rho})$
    for each robot $a$ do
        $r[a] \leftarrow P(r[a])$
    end for
    $k \leftarrow k + 1$
end while
```

5.4.1 REMC

Methodology

The simulation is performed on a graph structure identical to the graph in Fig. 5.1. The observation noises σ^2 are uniformly randomly generated from the range $(0, 200]$ and the mean are uniformly randomly generated from the range $(-10, 10]$. Both are sampled once at the beginning and used throughout the trials. Additionally, the noise is multiplied by the number of robots N, this is emulating for a fix cost, one can get a large team of robot with bad sensors or a small team of robot with good sensors. All the robots start from the same region (region 1) to emulate the condition that the team is activated at the same time from a base station. The annealing rate α is chosen as 0.025. 100 trials are conducted for each method. The robots are modeled as collision-free point masses that synchronously transition to their planned regions at each time step k. Parameter estimation (of μ, ν, b) and Markov chain optimization are performed in a centralized manner. The resultant stochastic matrix P is published to the robots and stochastic planning is carried out by the robots independently.

Results

Fig. 5.4 shows that the annealing method outperforms the direct method during the transient phase, and the uniform method asymptotically with respect to the true posterior entropy. The true posterior entropy is obtained from the true sample variance σ^2 and is maximized over the region, i.e. the region with the worst entropy is chosen for each time step k, where

$$h(k) = \max_i \left[\ln \left(\frac{\sigma_i^2}{\nu_i(k)} \right) \right]. \tag{5.22}$$

It is seen that the direct method has a true posterior entropy higher than both uniform and annealing methods from $k = 0$ to $k = 200$. This is caused by the previously mentioned problem of variance estimation error, which leads the robots to initially misallocate the samples until the variance is more refined. Consequently, the direct method also has more variability over the trials (shown in the shaded region) because of its high dependence on the quality of initial observations. Conversely, while the uniform method shows a better initial performance, its long-term performance is plateaued by the regions with high observation noise, since all regions are always assigned the same number of samples with the maximum entropy dominated by the noisiest region. The annealing method shows advantages over both the uniform and direct methods. Initially, when the coldness parameter $\beta = 0$, it closely follows the performance of the uniform method. As more samples are collected and $\beta \to 1$, the robots shift toward the optimal distribution.

More insight can be gained from the bottom row of Fig. 5.4, which plots the posterior entropy estimated by the robots. Here, the oracle σ^2 is replaced by the estimated variance at each time step $\bar{\sigma}^2(k)$ to yield

$$\bar{h}(k) = \max_i \left[\ln \left(\frac{\bar{\sigma}_i^2(k)}{\nu_i(k)} \right) \right]. \tag{5.23}$$

Uniform and annealed methods show similar estimated posterior entropy to the true posterior entropy, except they are nosier at the beginning when the sample size is small. However, direct entropy shows a significantly smaller estimated entropy than the true entropy. In other words, with the direct ergodic method, the robots believe they have more information than they really have, which directly causes the aforementioned misallocation of resources. Therefore, this shows that starting with uniform search is more advantageous as it avoids the problem of overconfidence.

The region-wise behavior of the robots distribution can be seen in Fig. 5.5, which shows the space average and time average of one example trial from the annealed and direct methods. It can be seen that the annealed method generates smooth trajectories from uniform distribution to the optimal distribution for the time average to follow, while the direct method has a fluctuating space average caused by the initial noisy variance estimation with a sudden shift to the optimal distribution as the sample size increases. This causes a highly time-varying distribution that is difficult for the ergodic control to track and leads to the extreme values seen in the time average.

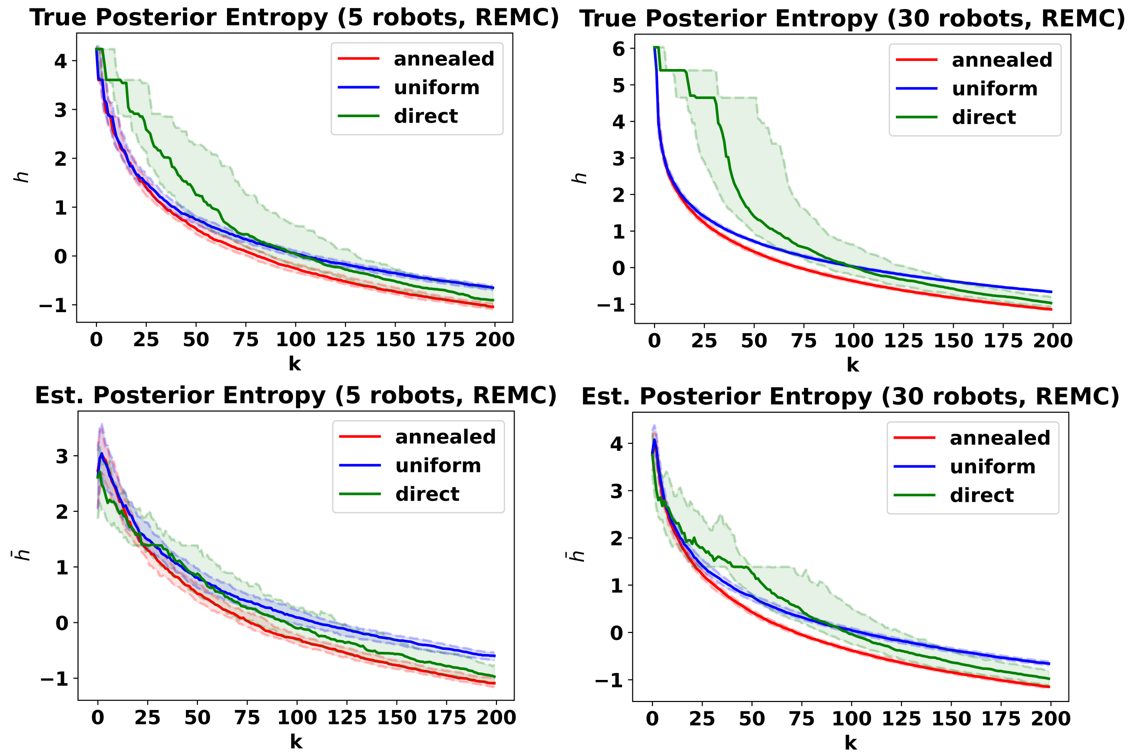


Figure 5.4: Comparison of maximum posterior entropy between uniform, direct ergodic, and annealed ergodic over 100 trials. With the solid line representing the median and the shaded region bounded by the dash line representing the first to third quartile region. top row shows the true entropy obtained from external oracle. This shows that annealing is consistently performing better both transiently and asymptotically. bottom row shows the estimated entropy from the internal believe of the robots. This shows that when compared to the true entropy, direct ergodic method has a problem of overestimating the information it has.

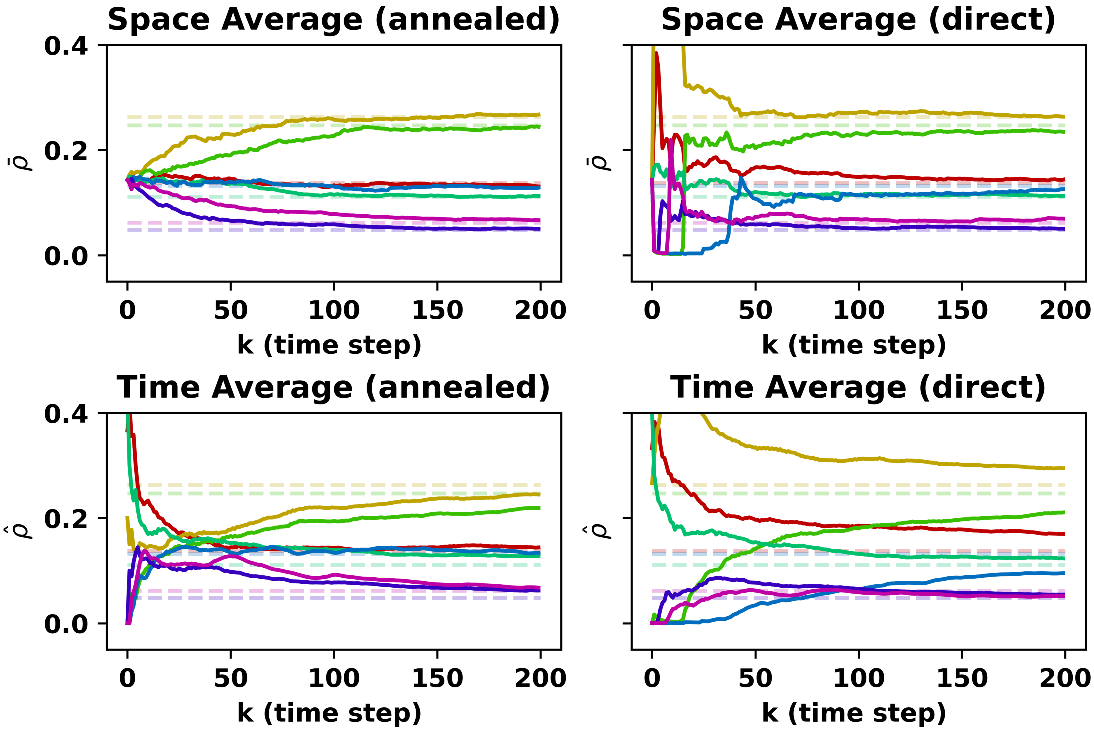


Figure 5.5: Example space average $\bar{\rho}$ and time average $\hat{\rho}$ of each region shown in different color, with the optimal distribution shown in dashed-line. The left column shows the result using annealing and right column shows the result of direct ergodic. Annealing shows a smooth transition from uniform to the optimal solution which is trackable for the time average; while the direct ergodic method produced a fluctuating space average that causes an extreme time average.

5.4.2 FMMC and Metropolis-Hasting

In this section, to explore the benefit of annealing to other graph traversal methods, the simulation is run with the ergodic graph planner switched from REMC to other Markov chain algorithms capable of producing ergodic traversal. Specifically, FMMC [17, Section 6], which is optimized for convergence of $\rho[k]$, instead of $\hat{\rho}$, toward $\bar{\rho}$; and Metropolis-Hasting[17, Section 1.2.2], which has the benefit of being ergodic without knowing the graph structure. Both the methods require a *reversible Markov chain*, which has a negative impact on the convergence rate.

Results

Fig. 5.6 shows similar improvements over uniform method and direct method when annealing is applied with the previous section in the true posterior entropy. Since uniform is a static distribution, the convergence rate has less impact on performance. Similarly, the annealed method provides a smoothly varying target distribution that is easy for the Markov chain to track. Conversely, the direct method shows even worse performance compared to the previous section, with high variance across the trials, and an overall worse performance where the median performance fails to reach the performance of the annealed solution. This shows that the concept of annealed ergodic coverage provides an advantage to a wide spectrum of ergodic methods.

5.4.3 Erdős–Rényi Random Graph

To show more generalized results, additional experiments were done on Erdős–Rényi graphs, where (directed) edges are independently generated with probability p. A total of 10 region nodes were considered in this part; with an edge probability of $p = \frac{2\ln(n)}{n}$; the rest of the setup is identical to Section 5.4.1. The results for 30 robots shows similar behavior as in the original graph, where direct method performs the worst initially and uniform method performs the worst near the end; while annealed method performs the best through out. For

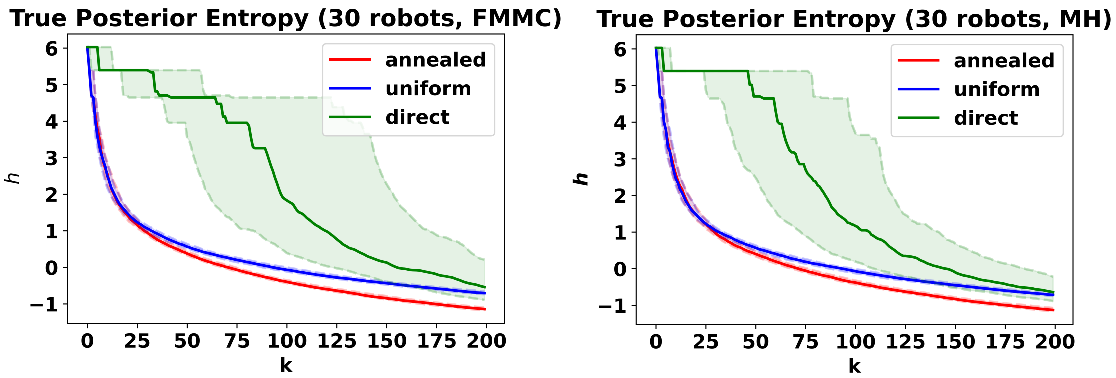


Figure 5.6: Posterior entropies similar to Fig. 5.4 with REMC ergodic planner replaced with FMMC on the left and Metropolis-Hasting (M-H) on the right. This shows more significant advantage of using annealing method under slower time-average convergence rate when compared to REMC.

the 5 robots case, direct method shows even more extreme result where the true entropy remains high through out. This is potentially caused by the higher number of regions, where there are less chance that the robots will explore regions they wrongly perceived as low variance and thus requires more time before correcting the error.

5.4.4 Physical Demonstration

The simulation result is applied to a swarm system with an OptiTrack motion capture rig to validate the feasibility of executing the exploration on physical platforms. A total of five TurtleBot Burger robots are used with twelve Flex13 tracking cameras on the motion capture system. A 4-region graph is projected to the ground. The cooling rate α is adjusted to 0.1. Snapshots of the system at 3 different time steps are shown in Fig. 5.8. The result shows that the planning pipeline can be executed in a physical platform with a suitable low-level collision avoidance technique.

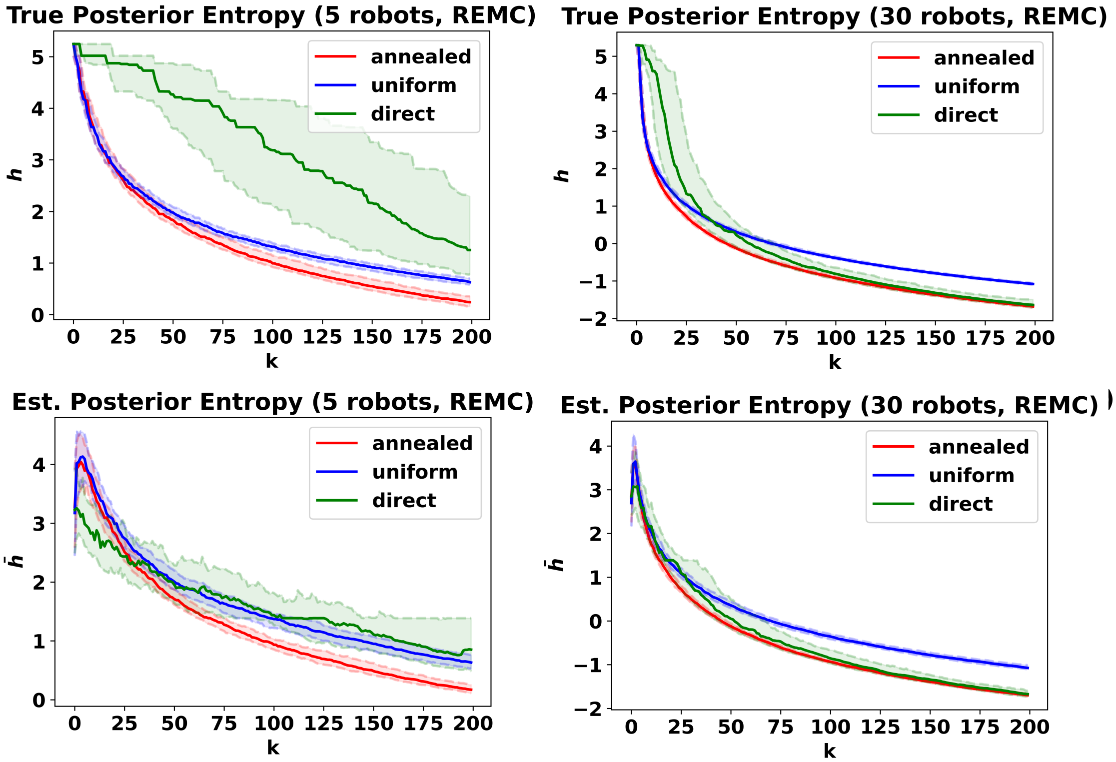
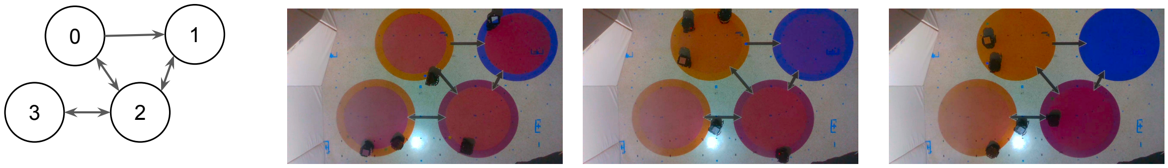


Figure 5.7: Simulation identical to Fig. 5.4 done on directed Erdős–Rényi random graphs with 10 regions instead of the original graph. The direct method shows an even worse performance potentially because of the increased number of regions and the increased number of directed edges.



(a) Graph structure (b) $k = 3$ ($\beta = 0.26$) (c) $k = 13$ ($\beta = 0.73$) (d) $k = 29$ ($\beta = 0.94$)

Figure 5.8: Experimental validation (see accompanying video). (a) shows the graph structure of the system used for physical demonstration. (b)-(d) shows the distribution of the TurtleBot Burger robots at different representative timestamps during the annealing process. The fill colors in the circular regions match the border colors as β increases, showing that the system has correctly identified the optimal target distribution.

5.5 *Discussion*

An open question for the annealing process is the optimality of the cooling rate α. In this document, we choose α empirically according to the simulation time horizon. Conceptually, the optimality of the cooling rate is correlated to the convergence rate of the ergodic planner, e.g. the ergodic controller should be able to reach uniform distribution before the annealing completely cools down. It is also correlated to the convergence of the noise estimation. Some observation models may require more samples to reach accurate noise estimation. The cooling rate then has to be slower so that the robots spend more time in the information collection phase.

In this document, the target distribution is time-varying with the application of annealing and the updating variance σ^2. This is not included in the proof for the optimality of REMC. Conceptually, since the variance estimation in (5.6) converges as ν converges, and the annealing converges to the estimated variance, a convergence proof can be established. This is potentially related to the notion of *weak ergodicity*, which is defined for a time-varying

Markov chain and is used to develop the convergence proof in simulated annealing [19].

One main challenge of the practicality of this framework is the assumption that all agents can communicate with the command/control center at all time steps. The Markov chain-based ergodic planner can be used in a fully decentralized manner without any communication for a static target distribution. However, due to the information gathering task, the target distribution is constantly being updated, which requires a new Markov chain to be generated and broadcast to the robots. In our future work, we will investigate a fully decentralized framework using percolation theory [65]. With the controller guaranteeing (weak) ergodicity, percolation theory can be applied [41] and, ideally, a critical transition point can be established for information to be fully shared in the robot network without requiring a command center.

For simplicity, the target information in this document is scalar and static. However, in many robotics applications, such as target tracking, the information will be multi-dimensional and dynamic. The core idea of “lower the quality of information, more robots should be allocated” should still be valid. Some extra considerations have to be accounted for, such as choosing a proper scalarization of variance (entropy, a-optimality, etc.) and establishing asymptotic confidence so that weak ergodicity holds true.

Chapter 6

FUTURE WORK

6.1 Decentralized Ergodic Swarm

While the distributing the robot is fully decentralized given a known Markov chain,the current method for generating this Markov chain requires a centralized system to integrate observations collected by the robots. To fully utilize the decentralized nature of an ergodic system, future work aims to decentralize observation integration through inter-robot communication. Using ergodicity, one can establish the probability that a pair of robots is within communication range and thus analytically derive the time required for information to be spread through the robot network. k. This will enable deployment of a self-organizing ergodic swarm without any central coordinator.

6.2 Statistical Congestion Avoidance

With the current method, the robots do not account for other's presence. This can cause many robots to concentrate in the same area, creating safety concerns (higher collision risk) and degrading performance (congestion and reduced observation quality). To avoid this, transition probabilities can be modified using an inter-robot potential energy term. In particular, a weakly interacting gas model can be used, where a region's total potential energy is defined by the ratio of number of robots and the region's area. Combined with the Boltzmann formulation, high potential energy reduces the probability that robots will transition into that region. Thus mitigating congestion.

BIBLIOGRAPHY

[1] Das iterationsverfahren bei natürlicher eigenwertproblem. *Annales Polonici mathematici.*

[2] Hierarchical clustering. URL:https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hier (version: 2024-11-07).

[3] *Confined space entry : an AIHA protocol guide.* AIHA publications. American Industrial Hygiene Association, Fairfax, Va, 1995.

[4] Ian Abraham and Todd D. Murphey. Decentralized ergodic control: Distribution-driven sensing and exploration for multiagent systems. *IEEE robotics and automation letters*, 3(4):2987–2994, 2018.

[5] Ian Abraham, Ahalya Prabhakar, and Todd D. Murphey. An ergodic measure for active learning from equilibrium. *IEEE Trans. Autom. Sci. Eng.*, 18(3):917–931, 2021.

[6] Aliakbar Akbari, Puneet S Chhabra, Ujjar Bhandari, and Sara Bernardini. Intelligent exploration and autonomous navigation in confined spaces. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 2157–2164, 2020.

[7] W Alt. Biased random walk models for chemotaxis and related diffusion approximations. *Journal of mathematical biology*, 9(2):147–177, 1980.

[8] Anonymous. Human-assisted robotic detection of foreign object debris inside confined spaces of marine vessels using probabilistic mapping. *Robot. Auton. Syst.*, 161:104349, 2023.

[9] Elif Ayvali, Hadi Salman, and Howie Choset. Ergodic coverage in constrained environments using stochastic trajectory optimization. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pages 5204–5210, 2017.

[10] H. Azpúrua, A. Rezende, G. Potje, et al. Towards semi-autonomous robotic inspection and mapping in confined spaces with the EspeleoRobô. *J. Intell. Robot. Syst.*, 101(4):69, 2021.

[11] S. Berman, A. Halasz, M.A. Hsieh, and V. Kumar. Optimized stochastic policies for task allocation in swarms of robots. *IEEE transactions on robotics*, 25(4):927–937, 2009.

[12] Graeme Best, Oliver M Cliff, Timothy Patten, Ramgopal R Mettu, and Robert Fitch. Dec-MCTS: Decentralized planning for multi-robot active perception. *Int. J. Robot. Res.*, 38(2-3):316–337, 2019.

[13] Cem Bilaloglu, Tobias Löw, and Sylvain Calinon. Tactile ergodic control using diffusion and geometric algebra. 2024.

[14] G. D. Birkhoff. What is the ergodic theorem? *The American Mathematical Monthly*, 49(4):222–226, 1942.

[15] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An Atlas framework for scalable mapping. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 2, pages 1899–1906 vol.2. IEEE, 2003.

[16] Stephen Boyd, Persi Diaconis, and Lin Xiao. Fastest mixing Markov chain on a graph. *SIAM Review*, 46(4):667–689, 2004.

[17] Stephen P. Boyd and Lieven. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[18] Pierre Brémaud. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*, volume 31. Springer Science & Business Media, Cham, Switzerland, 2013.

[19] Pierre Bremaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*, volume 31 of *Texts in Applied Mathematics*. Springer Nature, Netherlands, 2013.

[20] Rune Y. Brogaard, Marcin Zajaczkowski, Luka Kovac, Ole Ravn, and Evangelos Boukas. Towards UAV-based absolute hierarchical localization in confined spaces. In *IEEE Int. Symp. Safety Security Rescue Robot.*, pages 182–188, 2020.

[21] Gary Chambers. 2.4 confined space, 2001.

[22] Gary Chambers. 5.5 potential confined space hazards, 2001.

[23] Chun-Hung Chung, Kuan-Chi Wang, Kuan-Ting Liu, Yu-Ting Wu, Chien-Chou Lin, and Chuan-Yu Chang. Path planning algorithm for robotic lawnmower using RTK-GPS localization. In *Int. Symp. Community-centric Syst.*, pages 1–4, 2020.

[24] Laurène Claussmann, Marc Revilloud, and Sébastien Glaser. Simulated annealing-optimized trajectory planning within non-collision nominal intervals for highway autonomous driving. In *Int. Conf. Robot. Autom.*, pages 5922–5928, 2019.

[25] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Trans. Robot. Autom.*, 20(2):243–255, 2004.

[26] Bojan Crnković, Stefan Ivić, and Mila Zovko. Fast algorithm for centralized multi-agent maze exploration. *arXiv:2310.02121*, 2023.

[27] Inc. CVX Research. CVX: Matlab software for disciplined convex programming, version 2.0. `https://cvxr.com/cvx`, August 2012.

[28] Xu-Yang Dai, Jia-Ying Wang, and Qing-Hao Meng. An infotaxis-based odor source searching strategy for a mobile robot equipped with a TDLAS gas sensor. In *Chinese Cont. Conf.*, pages 4492–4497, 2019.

[29] Paolo De Petris, Huan Nguyen, Tung Dang, Frank Mascarich, and Kostas Alexis. Collision-tolerant autonomous navigation through manhole-sized confined environments. In *IEEE Int. Sym. Safety Security Rescue Robot.*, pages 84–89, 2020.

[30] Frank Dellaert and Michael Kaess. *Factor Graphs for Robot Perception*, volume 6 of *Foundations and trends in robotics.* Now Publishers, Boston - Delft, 2017.

[31] C.A. Desoer. *Feedback Systems: Input-output Properties.* Elsevier Science, 2012.

[32] Nikos Dimitropoulos, Niki Tzirtzilaki, George Michalos, and Sotiris Makris. Bio-inspired robots for confined space industrial applications: a novel cable-driven snake robot design. *Procedia CIRP*, 125:107–112, 2024.

[33] Saeid Doodman, Mir Abolfazl Mostafavi, Raja Sengupta, and Ali Afghantoloee. A novel Voronoi-driven optimization approach for point-based sensor network deployment. *IEEE Access*, 13, 2025.

[34] Hailong Duan, Yongde Zhang, and Han Liu. Continuous body type prostate biopsy robot for confined space operation. *IEEE access*, 11:113667–113677, 2023.

[35] Ayan Dutta, Swapnoneel Roy, O. Patrick Kreidl, and Ladislau Bölöni. Multi-robot information gathering for precision agriculture: Current state, scope, and challenges. *IEEE Access*, 9:161416–161430, 2021.

[36] EDF. CloudCompare local statistical test, 2020.

[37] Stephen E. Fienberg. A brief history of statistical models for network analysis and open challenges. *Journal of computational and graphical statistics*, 21(4):825–839, 2012.

[38] André R. Fioravanti, Jakub Mareček, Robert N. Shorten, Matheus Souza, and Fabian R. Wirth. On the ergodic control of ensembles. *Automatica (Oxford)*, 108:108483–, 2019.

[39] Sumit Gajjar, Jaydeep Bhadani, Pramit Dutta, and Naveen Rastogi. Complete coverage path planning algorithm for known 2D environment. In *IEEE Int. Conf. Recent Trends Electron. Inform. & Commun. Tech.*, pages 963–967, 2017.

[40] Mark Galassi, Jim Davies, James Theiler, Brian Gough, Gerard Jungman, Michael Booth, and Fabrice. Rossi. Gnu scientific library- reference manual, 2021.

[41] Ramen Ghosh. Ergodic control of multi-agent systems under communication constraints, 2025. DOI: 10.13140/RG.2.2.33017.38245.

[42] Dimitris Gkouletsos, Andrea Iannelli, Mathias Hudoba de Badyn, and John Lygeros. Decentralized trajectory optimization for multi-agent ergodic exploration. *IEEE robotics and automation letters*, 6(4):6329–6336, 2021.

[43] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, London, UK, 2008.

[44] Charles M Grinstead and J. Laurie Snell. *Introduction to Probability.* American Mathematical Society, Place of publication not identified, 1997.

[45] G Grisetti, R Kummerle, C Stachniss, and W Burgard. A tutorial on graph-based SLAM, 2010.

[46] Giorgio Grisetti, Rainer Kummerle, Cyrill Stachniss, Udo Frese, and Christoph Hertzberg. Hierarchical optimization on manifolds for online 2d and 3d mapping. In *2010 IEEE International Conference on Robotics and Automation*, pages 273–278. IEEE, 2010.

[47] Zhao Han, Jordan Allspaw, Gregory LeMasurier, Jenna Parrillo, Daniel Giger, S. Reza Ahmadzadeh, and Holly A. Yanco. Towards mobile multi-task manipulation in a confined and integrated environment with irregular objects. In *IEEE Int. Conf. Robot. Autom.*, pages 11025–11031, 2020.

[48] Sui Haotian, Li Tong, Wang Pu, Xu Liang, and Zhao Hongwei. Foreign object detection of electric transmission line with dynamic federated learning. *IOP Conf. Ser.: Earth Environ. Sci.*, 791(1):012159, 2021.

[49] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[50] Nicholas J. Higham. *Functions of Matrices : Theory and Computation.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.

[51] F. J. Humphreys, Gregory S. Rohrer, and A. D. Rollett. *Recrystallization and related annealing phenomena.* Elsevier, Amsterdam, third edition edition, 2017.

[52] Global Drone Inspection. Inspection of ballast tanks: let's keep it safe. URL:https://www.drone-inspection.global/maritime/ballast-tanks/ (version: 2024-11-07).

[53] S M Towhidul Islam and Xiaolin Hu. Towards decentralized importance-based multi-UAS path planning for wildfire monitoring. In *Annu. Syst. Syst. Eng. Conf.*, pages 67–72, 2022.

[54] Stefan Ivic. Motion control for autonomous heterogeneous multiagent area search in uncertain conditions. *IEEE transactions on cybernetics*, 52(5):3123–3135, 2022.

[55] Stefan Ivic, Bojan Crnkovic, and Igor Mezic. Ergodicity-based cooperative multiagent area coverage via a potential field. *IEEE Trans. Cybern.*, 47(8):1983–1993, 2017.

[56] Stefan Ivić, Bojan Crnković, Hassan Arbabi, Sophie Loire, Patrick Clary, and Igor Mezić. Search strategy in a complex and dynamic environment: The MH370 case. *Sci. Rep.*, 10(1):19640–19640, 2020.

[57] Stefan Ivić, Bojan Crnković, Luka Grbčić, and Lea Matleković. Multi-uav trajectory planning for 3d visual inspection of complex structures. *Automation in construction*, 147:104709–, 2023.

[58] Stefan Ivić, Ante Sikirica, and Bojan Crnković. Constrained multi-agent ergodic area surveying control based on finite element approximation of the potential field. *Eng. Appl. Artif. Intell.*, 116:105441, 2022.

[59] Mariano Jaimez, Javier G. Monroy, and Javier Gonzalez-Jimenez. Planar odometry from a radial laser scanner: A range flow-based approach. In *IEEE Int. Conf. Robot. and Autom.*, pages 4479–4485, 2016.

[60] Katharin R. Jensen-Nau, Tucker Hermans, and Kam K. Leang. Near-optimal area-coverage path planning of energy-constrained aerial robots with application in autonomous environmental monitoring. *IEEE Trans. Autom. Sci. Eng.*, 18(3):1453–1468, 2021.

[61] Galin L. Jones. On the Markov chain central limit theorem. *Probability Surveys*, 1:299–320, 2004.

[62] Phoebe J M Jones, Aaron Sim, Harriet B Taylor, Laurence Bugeon, Magaret J Dallman, Bernard Pereira, Michael P H Stumpf, and Juliane Liepe. Inference of random walk models to describe leukocyte migration. *Physical biology*, 12(6):066001–066001, 2015.

[63] Svenja Kahn, Ulrich Bockholt, Arjan Kuijper, and Dieter W. Fellner. Towards precise real-time 3D difference detection for industrial applications. *Comput. Ind.*, 64(9):1115–1128, 2013.

[64] Fidaa Karkori. *Means of Access, Safe Movement, and Enclosed Spaces*, pages 35–49. Springer Nature Switzerland, Cham, 2024.

[65] Christensen Kim. Percolation theory.

[66] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Sci.*, 220(4598):671–680, 1983.

[67] R. J. Kuo and Faisal Fuad Nursyahid. Foreign objects detection using deep learning techniques for graphic card assembly line. *J. Intell. Manuf.*, 2022.

[68] Mathieu Labbé and François Michaud. RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *J. Field Robot.*, 36(2):416–446, 2019.

[69] Ken Latimer. Remote visualization and detection of foreign object debris in aerospace manufacturing using a low-cost depth camera. Master's thesis, University of Washington, 2019.

[70] Cameron Lerch, Dayi Dong, and Ian Abraham. Safety-critical ergodic exploration in cluttered environments via control barrier functions. In *IEEE Int. Conf. Robot. Autom.*, pages 10205–10211, 2023.

[71] Jing Li, Ru-Chuan Wang, Hai-Ping Huang, and Li-Juan Sun. Voronoi based area coverage optimization for directional sensor networks. In *Int. Symp. Electronic Commerce Security*, volume 1, pages 488–493, 2009.

[72] John H Lienhard. *A Heat Transfer Textbook: Fourth Edition.* Dover Publications, 2013.

[73] Tobias Low, Jeremy Maceiras, and Sylvain Calinon. drozbot: Using ergodic control to draw portraits. *IEEE robotics and automation letters*, 7(4):11728–11734, 2022.

[74] Albert Madansky. Weighted standard error and its impact on significance testing (wincross vs. quantum spss). Accessed: 2022-07-28.

[75] Prithwish Manna, Sudipta Majumder, N. Hemarjit Singh, and Rajesh Bose. A context-aware framework for sensor placement using PSO and Voronoi in dynamic scenarios. In *Int. Conf. Intell. Syst. Advanced Comput. Commun.*, pages 1002–1006, 2025.

[76] Eitan Marder-Eppstein. nav_core, 2020. URL:https://wiki.ros.org/nav_core (version: 2024-11-07).

[77] Eitan Marder-Eppstein, David V. Lu, and Dave Hershberger. costmap_2d. `http://wiki.ros.org/costmap_2d`. Accessed: 2022-07-28.

[78] George Mathew and Igor Mezić. Metrics for ergodicity and design of ergodic dynamics for multi-agent systems. *Physica. D*, 240(4):432–442, 2011.

[79] Anastasia Mavrommati, Emmanouil Tzorakoleftherakis, Ian Abraham, and Todd D. Murphey. Real-time area coverage and target localization using receding-horizon ergodic exploration. *IEEE Trans. Robot.*, 34(1):62–80, 2018.

[80] C. D. (Carl Dean) Meyer. *Matrix analysis and applied linear algebra.* Society for Industrial and Applied Mathematics SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104, Philadelphia, Pa, 2000.

[81] Sean Meyn and Richard L. Tweedie. *Markov Chains and Stochastic Stability.* Communications and Control Engineering. Cambridge University Press, Cambridge, 2nd edition edition, 2009.

[82] Sean Meyn, Richard L. Tweedie, and Peter W. Glynn. *Ergodicity*, page 313–335. Cambridge Mathematical Library. Cambridge University Press, 2 edition, 2009.

[83] Lauren M. Miller, Yonatan Silverman, Malcolm A. MacIver, and Todd D. Murphey. Ergodic exploration of distributed information. *IEEE transactions on robotics*, 32(1):36–52, 2016.

[84] Lauren M. Miller, Yonatan Silverman, Malcolm A. MacIver, and Todd D. Murphey. Ergodic exploration of distributed information. *IEEE Trans. Robot.*, 32(1):36–52, 2016.

[85] Robb J. (Robb John) Muirhead. Aspects of multivariate statistical theory, 1982.

[86] Kevin P Murphy. Conjugate Bayesian analysis of the Gaussian distribution, 2007.

[87] Will Nash, Liang Zheng, and Nick Birbilis. Deep learning corrosion detection with confidence. *Npj Mater. Degrad.*, 6(1):1–13, 2022.

[88] Parker Owan, Joseph Garbini, and Santosh Devasia. Addressing agent disagreement in mixed-initiative traded control for confined-space manufacturing. In *IEEE Int. Conf. Adv. Intell. Mechatronics*, pages 227–234, 2017.

[89] Parker Owan, Joseph Garbini, and Santosh Devasia. Faster confined space manufacturing teleoperation through dynamic autonomy with task dynamics imitation learning. *IEEE Robot. Autom. Lett.*, 5(2):2357–2364, 2020.

[90] Emanuel Parzen. *Stochastic processes.* Holden-Day, 1962, California, 1962.

[91] Shivang Patel, Senthil Hariharan, Pranav Dhulipala, Ming C Lin, Dinesh Manocha, Huan Xu, and Michael Otte. Multi-agent ergodic coverage in urban environments. In *IEEE Int. Conf. Robot. Autom.*, pages 8764–8771, 2021.

[92] Emmanuel Pignat, João Silvério, and Sylvain Calinon. Learning from demonstration using products of experts: Applications to manipulation and task prioritization. *Int. J. Robot. Res.*, 41(2):163–188, 2022.

[93] Ahalya Prabhakar, Ian Abraham, Annalisa Taylor, Millicent Schlafly, Katarina Popovic, Giovani Diniz, Brendan Teich, Borislava Simidchieva, Shane Clark, and Todd Murphey. Ergodic specifications for flexible swarm control: From user commands to persistent adaptation. 2020.

[94] Victoria Preston, Taavi Salumäe, and Maarja Kruusmaa. Underwater confined space mapping by resource-constrained autonomous vehicle. *J. Field Robot.*, 35(7):1122–1148, 2018.

[95] Herbert Robbins. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.*, 58(5):527–535, 1952.

[96] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: An open-source library for real-time metric-semantic localization and mapping. In *IEEE Int. Conf. Robot. Autom.*, pages 1899–1906, 2020.

[97] Occupational Safety and Health Administration. Confined spaces. URL:https://www.osha.gov/confined-spaces (version: 2024-11-07).

[98] Hadi Salman, Elif Ayvali, and Howie Choset. Multi-agent ergodic coverage with obstacle avoidance. In *Int. Conf. Autom. Plan. Schedul.*, volume 27, pages 242–249, 2017.

[99] Ekta U Samani and Ashis G Banerjee. Persistent homology meets object unity: Object recognition in clutter. *IEEE Trans. Robot.*, 2023. Conditionally accepted for publication, arXiv preprint arXiv:2305.03815.

[100] Ekta U. Samani and Ashis G. Banerjee. THOR2: Topological analysis for 3D shape and color-based human-inspired object recognition in unseen environments. *Adv. Intell. Syst.*, page 2400539, 2025.

[101] Richard. Serfozo. Basics of applied stochastic processes, 2009.

[102] Nikunj Shah, Utsav Dey, and Kenji Nishimiya. End-to-end framework for robot lawnmower coverage path planning using cellular decomposition. *arXiv:2506.06028*, 2025.

[103] Suhan Shetty, Joao Silverio, and Sylvain Calinon. Ergodic exploration using tensor train: Applications in insertion tasks. *IEEE Trans. Robot.*, 38(2):906–921, 2022.

[104] Weijie Shi, Zhou He, Wei Tang, Weifeng Liu, and Ziyue Ma. Path planning of multi-robot systems with Boolean specifications based on simulated annealing. *IEEE Robot. Autom. Lett.*, 7(3):6091–6098, 2022.

[105] Burhanuddin Shirose, Adam Johnson, Bhaskar Vundurthy, Howie Choset, and Matthew Travers. GESCE: Graph-based ergodic search in cluttered environments. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 7611–7616, 2024.

[106] Aleksandrs Slivkins. Introduction to multi-armed bandits. *Found. Trends Mach. Learn.*, 12(1–2):1–286, 2019.

[107] Joan Solà, Jeremie Deray, and Dinesh Atchuthan. A micro Lie theory for state estimation in robotics. *arXiv.org*, 2021.

[108] Tami. Water ballast tank inspection. URL:https://www.madfly.fr/news/post/water-ballast-tank-inspection (version: 2023-02-22).

[109] Eddyfi Technologies. Versatrax pipe inspection crawler, 2021.

[110] Meera Towler. Designing a confined space robot for complex tasks with high force processes. URL:https://www.swri.org/industry/industrial-robotics-automation/blog/designing-confined-space-robot-complex-tasks-high-force (version: 2022-02-04).

[111] Paolo Tripicchio, Massimo Satler, Matteo Unetti, and Carlo A Avizzano. Confined spaces industrial inspection with micro aerial vehicles and laser range finder localization. *Int. J. Micro Air Veh.*, 10(2):207–224, 2018.

[112] user8675309 (https://math.stackexchange.com/users/735806/user8675309). Reference request: convergence for periodic markov chains. Mathematics Stack Exchange. URL:https://math.stackexchange.com/q/3704687 (version: 2020-06-08).

[113] Peter Walters. An introduction to ergodic theory, 1982.

[114] Zhi Wen, Xiaotao Liu, Gaojie Lu, and Jing Liu. Rapid autonomous exploration of large-scale environments for ground robots based on region partitioning. In *Int. Conf. Robot. Autom.*, pages 13067–13073, 2025.

[115] Benjamin Wong, Ryan H Lee, Tyler M Paine, Santosh Devasia, and Ashis G Banerjee. Rapidly converging time-discounted ergodicity on graphs for active inspection of confined spaces. 2025.

[116] Benjamin Wong, Aaron Weber, Mohamed M Safwat, Santosh Devasia, and Ashis G Banerjee. Simulated annealing for multi-robot ergodic information acquisition using graph-based discretization. 2025.

[117] Haitao Xiong, Jiaqing Wu, Qing Liu, and Yuanyuan Cai. Research on abnormal object detection in specific region based on Mask R-CNN. *Int. J. Adv. Robot. Syst.*, 17(3):1729881420925287, 2020.

[118] Albert Xu, Bhaskar Vundurthy, Geordan Gutow, Ian Abraham, Jeff Schneider, and Howie Choset. Measure preserving flows for ergodic search in convoluted environments. *arXiv preprint arXiv:2409.09164*, 2024.

[119] Haoyu Xu, Zhenqi Han, Songlin Feng, Han Zhou, and Yuchun Fang. Foreign object debris material recognition based on convolutional neural networks. *EURASIP J. Image Video Proc.*, 21:1–10, 2018.

[120] Yan Yongjie and Zhang Yan. Collision avoidance planning in multi-robot based on improved artificial potential field and rules. In *IEEE Int. Conf. Robot. Biomimetics*, pages 1026–1031, 2009.

[121] María Inés Zamudio. Worker death in louisiana confined space showcases dangerous trend. *Center for Public Integrity.*

[122] Tianpeng Zhang, Kasper Johansson, and Na Li. Multi-armed bandit learning on a graph. In *Annu. Conf. Inf. Sci. Syst.*, pages 1–6, 2023.

[123] Weishan Zhang, Xia Liu, Jiangru Yuan, Liang Xu, Haoyun Sun, Jiehan Zhou, and Xin Liu. RCNN-based foreign object detection for securing power transmission lines (RCNN4SPTL). *Procedia Comput. Sci.*, 147:331–337, 2019.

[124] Xiaoming Zheng, Sonal Jain, S. Koenig, and D. Kempe. Multi-robot forest coverage. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pages 3852–3857, 2005.

[125] Qidan Zhu, Yongjie Yan, and Zhuoyi Xing. Robot path planning based on artificial potential field approach with simulated annealing. In *Int. Conf. Intell. Syst. Des.Appl.*, volume 2, pages 622–627, 2006.