Implementation of Algorithms for Multi-Robot Coordination in Robot Operating System (ROS)

Student Name: Sachin Maurya
Roll Number: 2022424

Student Name: Sankalp Raj
Roll Number: 2022445

Student Name: Sanyam Barwar
Roll Number: 2022447

BTP report submitted in partial fulfilment of the requirements for the Degree of B.Tech. in Computer Sciences & Biosciences as well as B.Tech. in Computer Science & Social Science

on July 21, 2025

BTP Track: Research

BTP Advisor
Dr. Tanmoy Kundu

Indraprastha Institute of Information Technology
New Delhi

Student Declaration

We hereby declare that the work presented in the report entitled **"Implementation of Algorithms for Multi-Robot Coordination in Robot Operating System (ROS)"** submitted by us for the partial fulfilment of the requirements for the degree of *B.Tech. in Computer Sciences & Biosciences* and *B.Tech. in Computer Science & Social Science* at Indraprastha Institute of Information Technology, Delhi, is an authentic record of our work carried out under the guidance of **Dr. Tanmoy Kundu** . Due acknowledgements have been given in the report for all material used. This work has not been submitted elsewhere for the reward of any other degree.

Sachin Maurya, Sankalp Raj & Sanyam Barwar <u>**Place & Date**</u>**:** July 21, 2025

Certificate

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Tanmoy Kundu <u>**Place & Date**</u>**:** July 21, 2025

Abstract

This project explores the design and implementation of a decentralized multi-robot coordination system using the Robot Operating System (ROS) Noetic and the Turtlesim simulator. The core objective is to demonstrate how multiple robots can cooperatively reduce uncertainty in a shared environment using belief space planning, without relying on centralized control. Each robot maintains a local belief grid representing probabilistic information about the workspace. Belief updates are performed using a Bayesian filter with both prediction and correction steps, allowing the robots to reason under uncertainty and incorporate past observations. Communication between robots is triggered adaptively when predicted actions disagree, ensuring efficient information exchange via custom ROS messages and topics. The system integrates elements of probabilistic robotics, distributed algorithms, and basic graph connectivity, and highlights key aspects of decentralized decision making. This work serves as a proof-of-concept for scalable multi-robot exploration strategies, with potential extensions to real-world sensor networks and autonomous systems operating in dynamic or partially observable environments.

Keywords: Decentralized coordination, belief space planning, Bayesian filter, multi-robot systems, ROS, probabilistic robotics, graph theory, Turtlesim.

Acknowledgment

We would like to express our sincere gratitude to our project supervisor, **Dr. Tanmoy Kundu**, for his continuous guidance, valuable feedback, and encouragement throughout this Bachelor's Thesis Project. His insights into decentralized algorithms, multi-robot systems, and belief space planning greatly shaped the direction and depth of this work.
We are also thankful to the Department of Computer Science at IIIT Delhi for providing the necessary infrastructure and resources for experimentation with ROS and simulation tools. Finally, we acknowledge the support and cooperation of our peers and labmates, who contributed ideas and discussions that helped refine the project.

Contents

Chapter 1

Introduction

1.1 Background

Modern multi-robot systems are an important field of research in robotics, providing scalable solutions for tasks that would be difficult, time-consuming, or infeasible for a single robot. Coordinating multiple robots brings unique challenges in planning, uncertainty handling, and decentralized control.

A key approach to managing uncertainty is *belief space planning*, where each robot maintains a probabilistic belief of its environment and updates it using a combination of motion models and sensor observations. This project takes inspiration from classic probabilistic robotics theory [2] and recent decentralized planning work by Kundu et al. [1].

1.2 Motivation

The main motivation for this BTP project, proposed by Prof. Tanmoy Kundu, is to practically demonstrate decentralized belief space planning in a simulated environment using ROS Noetic. The focus is on how robots can collaboratively reduce uncertainty through local decisions, minimal communication, and prediction.

1.3 Objectives

The objectives of this work are:

- Develop a decentralized belief space planner in ROS.
- Implement recursive Bayesian belief updates with both prediction and correction.
- Use conditional communication to share belief grids when disagreement is detected.
- Visualize the multi-robot system using Turtlesim with clear path differentiation.

1.4 System Overview

The system consists of two Turtlesim robots, each maintaining a local 5×5 belief grid. Robots choose target cells based on uncertainty, update beliefs using a simple Bayesian filter, and communicate when needed. The robots draw colored paths to make their movement distinguishable.

Figure 1.1 shows the overall system architecture, including the belief grids, prediction, correction, and communication loop.

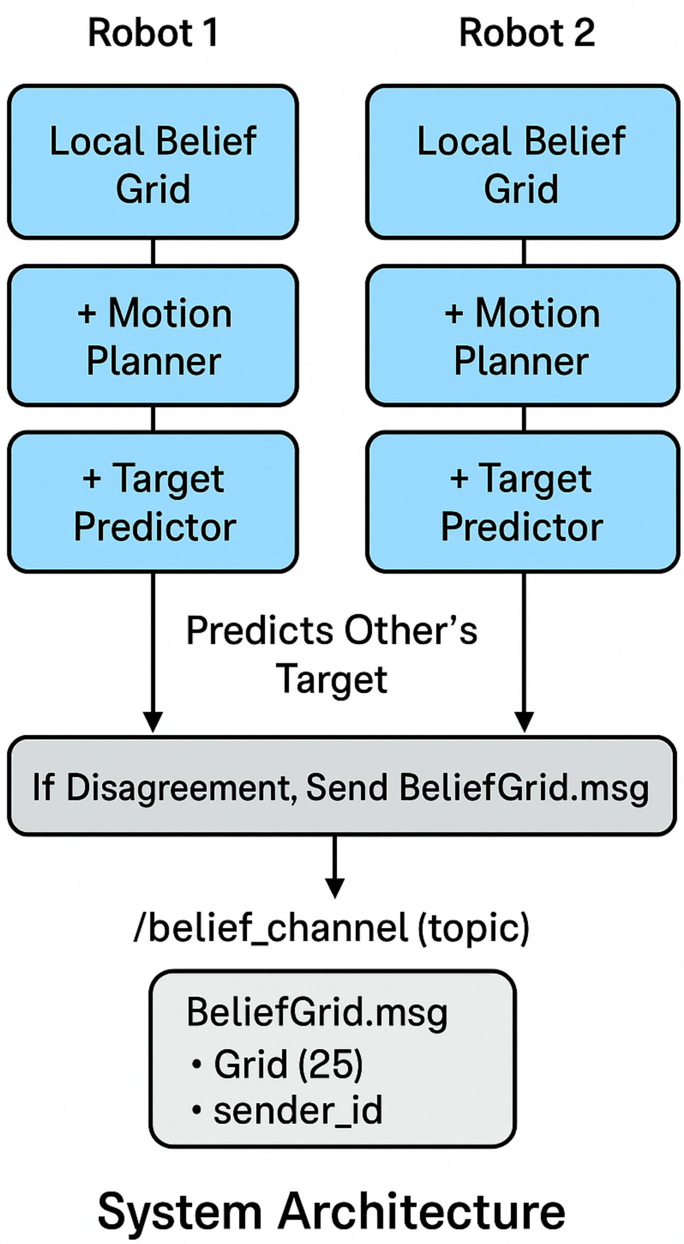

Figure 1.1: System architecture of the decentralized multi-robot belief space planner.

1.5 Placeholders for Additional Visuals

To support this introduction, we plan to include:

- **Turtlesim simulation screenshot** showing two robots with distinct pen colors. (See Figure 1.2)
- **Sample belief grid heatmaps** at different stages (initial, mid-run, final). (See Figure 1.3)
- **ROS communication graph** from `rqt_graph` visualizing the '/belief_channel' topic. (See Figure 1.4)

`figs/turtlesim_screenshot_placeholder.png`

Figure 1.2: Placeholder: Turtlesim simulation screenshot.

This report explains the design, mathematics, and results of our decentralized multi-robot belief space system.

figs/belief_grid_heatmap_placeholder.png

Figure 1.3: Placeholder: Example belief grid heatmap.

figs/ros_graph_placeholder.png

Figure 1.4: Placeholder: ROS communication graph (‘rqt_graph‘).

Chapter 2

Literature Review

2.1 Decentralized Belief Space Planning

The core of our Bachelor's Thesis Project (BTP) is based on the research paper by Dr. Tanmoy Kundu et al. [1], which investigates **Decentralized Multi-Robot Belief Space Planning** under communication constraints. Their work proposes a robust framework where multiple robots maintain local belief models of their environment and share information selectively to minimize redundant work and conserve bandwidth.

In particular, the paper introduces mechanisms such as **VERIFYAC**, where robots predict each other's plans to verify alignment, and **ENFORCEAC**, where belief synchronization occurs when disagreements arise. We implemented these ideas by:

- Maintaining local 5×5 belief grids for each robot.
- Using a simple target prediction mechanism to compare the plan of each robot.
- Trigger communication to share belief grids when predictions disagree.

Although Dr. Kundu's work targets more complex real-world scenarios with advanced path planning, our implementation demonstrates the same decentralized principles using ROS and Turtlesim.

2.2 Probabilistic Robotics and Bayesian Filters

To build the belief update mechanism, we refer to the foundational text *Probabilistic Robotics* by Thrun, Burgard, and Fox [2]. This book explains the **Bayes filter algorithm**, which underpins probabilistic state estimation in robotics. The classic **door state estimation** example in Chapter 2 illustrates how a robot predicts an uncertain state and corrects its estimate using noisy sensor data.

This concept shaped our belief grid update logic:

- A **prediction step** that diffuses uncertainty across the grid.
- A **correction step** that applies the Bayes rule when a robot senses a cell.
- Normalization and spillover to simulate realistic imperfect sensing.

2.3 Summary

In summary, Dr. Kundu et al. provided the **decentralized planning and coordination framework**, while Thrun's work guided the ** update of probabilistic beliefs**. Together, these sources form the theoretical backbone of our ROS-based multi-robot coordination system.

Chapter 3

Methodology

3.1 System Setup

Our decentralized multi-robot belief space planner is implemented using the Robot Operating System (ROS) Noetic and the Turtlesim simulator. Two separate turtles represent two robots operating in the same virtual environment but with distinct namespaces and visual pen colors.

Each robot runs a dedicated Python ROS node:

- `robot1_node.py`
- `robot2_node.py`

The system is launched using `multi_robot.launch`, which:

- Starts two identical Turtlesim windows.
- Spawns each turtle at the center of the window.
- Assigns red and green pen colors for Robot 1 and Robot 2 respectively.
- Launches both robot control nodes.

All nodes communicate using standard ROS publisher/subscriber patterns:

- Each robot subscribes to its pose topic `/robotX/turtle1/pose`.
- Each publishes velocity commands to `/robotX/turtle1/cmd_vel`.
- Robots use a custom message `BeliefGrid.msg` to broadcast belief grids.
- A simple `/comm_channel` logs when disagreements are detected.

3.2 Why This Approach

The motivation is to demonstrate how multi-robot systems can handle uncertainty collaboratively without centralized control. By combining local Bayesian belief updates with conditional communication, robots can coordinate efficiently to cover an environment, inspired by real-world decentralized planning problems.

3.3 Belief Representation

Each robot maintains a 5×5 grid:

$$belief[i][j] \in [0, 1].$$

Initially, all cells are:

$$belief[i][j] = 0.5.$$

Higher values mean higher uncertainty. The goal is to reduce the entire grid's uncertainty through movement and sensing.

3.4 Prediction and Correction

At each loop iteration, each robot performs:

- **Prediction (Motion Update)**:

$$belief'_{i,j} = 0.8 \times belief_{i,j} + 0.2 \times \left(\sum_{n \in N(i,j)} \frac{belief_{i,j}}{|N(i,j)|} \right).$$

  This step spreads uncertainty to neighbors, simulating environment change when unobserved.

- **Correction (Sensor Update)**: Using the current pose, the robot locates its grid cell and updates it using Bayes' rule:

$$P(H|Z) \propto P(Z|H)P(H). \quad \text{Practically:} \quad belief[i][j] = prior \times \big(P(Z|H) \times prior + P(Z|\neg H) \times (1 - prior) \big).$$

- **Spillover**: 20% of the update spills to 4-connected neighbors:

$$belief[i][j] = posterior \times 0.8, \quad belief[n] += \frac{posterior \times 0.2}{|N|}.$$

- **Normalization**: After updates:

$$belief[i][j] = \frac{belief[i][j]}{\sum_{i,j} belief[i][j]}.$$

3.5 Target Selection and Motion Control

Robots plan motion based on their local belief:

- **Robot 1** targets the cell with maximum uncertainty.
- **Robot 2** targets the cell with minimum uncertainty.

For each robot:

$$\theta = \operatorname{atan2}(y_{target} - y,\, x_{target} - x). \quad speed = 0.5 + 1.5 \times belief[i][j] + \text{noise}.$$

The local belief scales the speed: more uncertain → faster movement. Random noise breaks symmetry and prevents deadlocks.

3.6 Communication and Merging

To avoid redundant work:

- Each robot predicts the other's likely target.
- If predictions disagree, it broadcasts its belief grid.
- The receiver averages the received grid:

$$belief[i][j] = \frac{belief[i][j] + received[i][j]}{2}.$$

This conditional sharing implements VERIFYAC and ENFORCEAC in a simple form [1].

3.7 Build System

The project uses standard ROS build tools:

- `package.xml` declares dependencies: `geometry_msgs`, `std_msgs`, `turtlesim`, and custom messages.
- `CMakeLists.txt` builds the custom message and registers Python nodes.

3.8 Termination Condition

When:

$$belief[i][j] < 0.01 \quad \forall (i, j),$$

the robot stops moving, indicating the grid is fully explored.

3.9 Summary

This methodology combines:

- Decentralized Bayesian prediction and correction.
- ROS publisher/subscriber messaging.
- Conditional belief fusion to synchronize robots.
- Motion control directly tied to uncertainty.

It demonstrates key ideas from belief space planning in a simple, extensible ROS simulation inspired by [1] and the Bayesian filter framework of Thrun et al. [2].

Chapter 4

Results

Bibliography

[1] Tanmoy Kundu et al. "Decentralized Multi-Robot Belief Space Planning Under Communication Constraints". In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 5850–5857. DOI: 10.1109/IROS56455.2024.10801372. URL: https://ieeexplore.ieee.org/document/10801372.

[2] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005. ISBN: 978-0-262-20162-9. URL: https://mitpress.mit.edu/9780262201629/probabilistic-robotics/.