SwarmBots : Autonomous Multi-Agent Task Execution with Centralized Ultrasonic Mapping

Bathina Sumanth, Sowmith and G.V.V. Sharma
Department of Electrical Engineering,
Indian Institute of Technology Hyderabad,
Kandi, India 502284
gadepall@ee.iith.ac.in

Abstract—**This paper presents a novel low-cost, centralized framework for mapping and multi-task execution using multiple Unmanned Ground Vehicles (UGVs). The system leverages ultrasonic sensors for environmental mapping, followed by autonomous task execution using a master-slave architecture. A user-interactive graphical interface allows manual selection of task points on the generated map. The master UGV maps the environment and broadcasts task locations to slave UGVs for autonomous execution. Experimental results demonstrate reliable mapping accuracy and robust coordination among UGVs in a controlled indoor environment.**

1 Introduction

With the increasing adoption of autonomous systems in logistics, surveillance, and factory automation, there is a growing demand for scalable and cost-efficient robotic platforms capable of operating collaboratively. Traditional solutions rely on expensive sensors like LiDAR and high-end computational platforms for SLAM and autonomous task delegation. These setups are cost-prohibitive for small-scale, indoor, or educational use cases.

This research proposes an ultrasonic sensor-based mapping framework integrated with a GUI for point-driven task assignment. The architecture consists of a central master UGV that maps the environment and a group of slave UGVs that execute assigned tasks based on user input. The motivation lies in building a system that blends human decision-making with autonomous robot execution in a simple, modular, and affordable way.

2 Related Work

Extensive research exists on multi-robot systems, focusing on cooperative exploration, SLAM, and task planning using high-end sensors. For instance, Jin et al. (2024) proposed a DDQN and Ant Colony Optimization (ACO)-based system for dynamic task assignment in UGV networks. MDPI Robotics (2022) discussed hierarchical planning techniques for cooperative UAV and UGV missions. Other studies focus on cloud-integrated robot planning or computer vision-based mapping. However, few works explore the use of low-cost ultrasonic mapping combined with manual task definition through GUI interfaces in a master-slave UGV architecture.

This work contributes to that niche by offering an accessible alternative using simple sensors, local microcontrollers, and embedded wireless communication (ESP-NOW or Wi-Fi Direct) to enable centralized planning and distributed task execution.

3 Proposed Navigation System

The proposed architecture for multi-UGV coordination is based on a deterministic master-slave hierarchy where a central master UGV conducts initial mapping, and slave UGVs autonomously execute tasks. Key components of the system include:

3.1 *Collaborative Mapping Using Ultrasonic Sensors*

The master UGV maps the unknown environment using ultrasonic sensors (HC-SR04) and a grid traversal strategy. Each measurement is synchronized with RPM-based dead reckoning for position estimation. A virtual occupancy grid map is constructed in real-time and visualized via the GUI.

3.2 *Dead Reckoning for Localization*

Each UGV computes its position based on wheel rotations (RPM), using:

$$\text{Distance} = \frac{\text{RPM} \times \pi \times \text{Diameter} \times \Delta t}{60} \tag{1}$$

This allows real-time navigation feedback without relying on GPS or encoders.

3.3 *Time-Aware Task Distribution Algorithm*

After mapping, users select task points via the web interface. The master UGV divides the drawn path into segments based on execution time estimations. Each slave UGV receives a sub-task proportional to its expected task completion duration, ensuring balanced workloads.

3.4 *Peer-to-Peer Communication with ESP-NOW*

ESP32 microcontrollers enable fast, reliable communication using ESP-NOW. This protocol supports low-latency, router-free communication by using MAC-address-based peer discovery, ideal for indoor or offline applications.

3.5 GUI-Based Human Interaction Layer

The GUI is hosted on the master UGV and serves as the primary human-machine interface. It enables the following functionalities:

- Canvas-based path drawing: Users can draw custom paths or regions directly onto the grid map for task planning.
- Click-based task selection: Individual grid points can be selected to assign discrete tasks to UGVs.
- Real-time feedback of UGV positions: Each UGV's estimated location is updated and visualized on the interface.
- Task status visualization: Task execution states (e.g., pending, in-progress, completed) are reflected dynamically on the GUI.

3.6 Embedded Software Architecture

Each UGV runs modular firmware developed using either the ESP-IDF or Arduino framework, depending on complexity. The software architecture is divided into:

- Master UGV: Responsible for path segmentation, task scheduling, and GUI synchronization. It acts as a bridge between the user and slave units.
- Slave UGVs: Execute motion control routines, update position through RPM-based dead reckoning, and report task status back to the master.

Firmware modules are structured to separate communication, control logic, sensor interfacing, and utility functions for maintainability and reusability.

The platform is engineered for deployment in resource-constrained environments, enabling scalable swarm behaviors through compact embedded logic and lightweight communication protocols.

4 Methodology

4.1 Mapping Phase

The master UGV performs 2D environmental mapping by traversing the environment in a structured grid pattern. It uses ultrasonic distance sensors to gather obstacle data, which is fused with positional estimates obtained via dead reckoning. A 2D occupancy grid is constructed dynamically to represent free and occupied spaces in the environment.

4.2 GUI-Based Task Assignment

The real-time occupancy grid is rendered on the GUI, allowing users to visually inspect the environment and select target task points. These click-based inputs are encoded into coordinate values and transmitted wirelessly to the master UGV for further processing.

4.3 Task Distribution Strategy

Once the master UGV receives the task coordinates, it allocates them to the slave UGVs based on either:

- Round-robin assignment: Tasks are distributed sequentially to each available slave UGV.
- Distance-based allocation: Each task is assigned to the nearest available slave UGV, minimizing traversal time.

This strategy ensures balanced workload distribution and reduces the likelihood of navigation conflicts or congestion.

4.4 Execution Phase

Upon receiving their assigned task coordinates, slave UGVs begin autonomous navigation. Navigation is achieved through time-based motor control or simple PID algorithms. On reaching the target, each UGV performs the associated operation, such as flashing an LED, taking sensor readings, or signaling task completion to the master.

5 Results

The SwarmBots system was tested in a structured indoor environment using a master-slave configuration of three UGVs. The key outcomes of the experiment are summarized below

Reliable Mapping Accuracy: The master UGV successfully generated a real-time 2D occupancy map using only ultrasonic data and dead reckoning. Spatial features such as walls and obstacles were distinguishable without requiring GPS, LIDAR, or encoders.

5.1 Interactive GUI Performance

The GUI enabled smooth human-robot interaction. Users could assign target points via simple clicks, and receive real-time visual feedback on robot positions and task progress.

5.2 Autonomous Multi-UGV Task Execution

Slave UGVs executed assigned tasks in parallel with high independence. Each UGV navigated to its target using internal logic, and task completion was communicated back to the master without human intervention.

5.3 Position Estimation Precision

Experimental trials demonstrated a positional accuracy within ±5 cm, sufficient for typical indoor navigation and task execution scenarios.

5.4 System Scalability

The framework scaled effectively with multiple UGVs. Task distribution strategies (round-robin and distance-based) helped avoid collisions and ensured balanced workloads.

5.5 Low-Cost, High-Impact

All experiments were conducted using off-the-shelf ESP32 boards and basic sensors, underscoring the potential of this architecture in resource-constrained deployments.

6 Conclusion and Future Work

In this work, we developed and demonstrated a low-cost, modular, and scalable system for autonomous task execution using a swarm of Unmanned Ground Vehicles (UGVs). Our approach integrated ultrasonic-based mapping, RPM-based dead reckoning, and a user-friendly GUI for task coordination. The system successfully enabled collaborative mapping and parallel task execution in an indoor setting without relying on GPS, LiDAR, or expensive sensors. The experiments confirmed the reliability of position estimation, real-time feedback, and decentralized task handling using ESP-NOW communication. The architecture proved to be robust, scalable, and efficient for indoor robotic coordination, even with hardware constraints and future work includes.

- Incorporating IMU and optical encoders to enhance localization accuracy
- Implementing advanced obstacle avoidance using IR or vision-based sensors
- Developing adaptive task reassignment algorithms based on real-time status
- Extending the system to semi-structured outdoor environments with longer range communication
- Introducing swarm-level intelligence to support dynamic decision-making and reconfiguration

References

1) Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: A review from the swarm engineering perspective. *Swarm Intelligence*, **7**(1), 1–41.
2) Kalra, N., Ferguson, D., & Stentz, A. (2005). Hoplite: A Markov Decision Process-Based Multirobot Coordination System. *International Journal of Robotics Research*, **24**(10), 971–1000.
3) Espressif Systems. ESP-NOW User Guide. *[Online]*.
4) Arduino. HC-SR04 Ultrasonic Distance Sensor Tutorial. *[Online]*.
5) Kumar, M., & Michael, N. (2008). Opportunistic Task Allocation in Multi-Robot Systems. *Robotics: Science and Systems IV*.
6) Random Nerd Tutorials. ESP32 Web Server using SPIFFS (Host HTML Files). *[Online]*.
7) Gadepalli, V. (2023). Future Wireless Communication: Embedded Systems and Robotics Tutorials. *GitHub Repository*.