Battery-free Real-time Localization of Autonomous Robot Swarms Using UWB

Tilboon Elberier
2nd Year Ph.D. ECE (UID: 106303745)
University of California Los Angeles
tilboon@g.ucla.edu

Shengguang Cui
1st Year M.S. ECE (UID: 406530609)
University of California Los Angeles
lilil@g.ucla.edu

Colin Recker
1st Year M.S. ECE (UID: 206530280)
University of California Los Angeles
colinrec34@g.ucla.edu

Renish Israel
1st Year M.S. ECE (UID: 606530590)
University of California Los Angeles
renishisrael@g.ucla.edu

***Abstract*—This paper presents a novel approach to utilizing UWB for the localization of autonomous, battery-free swarms of centimeter-scale robots. In this work, we focus on algorithm development and simulation to evaluate the mapping capabilities of dynamic environments using simulated, parameterized autonomous robots in three key areas: power consumption, map accuracy, and map coverage.**

This document serves as the final paper for the Computational Robotics class in Fall 2024 at UCLA. It describes the motivation, implementation techniques, simulation results, and relevant references.

***Index Terms*—UWB, BLE, Low Power Mode, Mapping, Path Planning**

I. Introduction

A. Problem Statement

This work aims to establish a proof of concept for mapping a given environment with high-precision localization while minimizing total power consumption. Specifically, it focuses on modeling centimeter-scale sensor nodes and their interaction with two-dimensional environments. The evaluation criteria include power consumption, mapping accuracy, and map coverage of the swarm as a whole. These metrics are compared against a baseline scenario involving random navigation with no localization communication to assess the performance improvements achieved by the proposed system and technology improvements.

B. Background

As robotics technology advances, larger, battery-powered robots achieve more sophisticated capabilities but require significantly more energy for locomotion than fixed IoT nodes [1], resulting in shorter operational lifetimes due to the energy density of current batteries [2]. This also limits travel distances and necessitates human intervention for battery maintenance, raising environmental and health concerns from toxic materials in lithium batteries, such as lead, cobalt, and chromium [3]. In contrast, microrobots, with their small form factor, enable battery-free autonomous locomotion, allowing large-scale swarm deployments, reduced costs, minimized hazards, and potentially indefinite operation. However, small-scale robots have yet to match the transformative advancements of larger systems, particularly in sensing and navigation, due to power constraints and assumptions about energy requirements. Bridging this gap is key to realizing microrobots as networked mobile sensory nodes for long-term environmental intelligence in hard-to-reach areas. This research focuses on integrating networking, sensing, and mapping capabilities in battery-free microrobots to achieve functionality comparable to larger systems while sustainably operating on ambient energy.

C. Our Solution

High-efficiency solar cells, developed by Microlink Devices, generate 6 mW of power from a 6x6 mm, 6 mg cell, enabling power-harvesting microrobots like the MilliMobile and Origami Microfliers [4] [5]. These robots demonstrate autonomous operation with minimal power but lack intelligent navigation and networking capabilities necessary for large-scale swarms [4].

To address this, we developed localization algorithms optimized for solar-powered robots. Using duty-cycled UWB communication, our approach minimizes power consumption while maintaining high-precision localization. Leader nodes are dynamically selected based on power levels and proximity to unexplored regions, enabling efficient energy use and effective terrain mapping. Simulations evaluate power consumption, mapping accuracy, and coverage, incorporating realistic power and communication constraints to bridge the gap between simulation and real-world deployment. The comparison in Table 1 highlights how our work stands out by achieving battery-free and extremely low power operation for enhanced swarm efficiency, addressing key limitations in existing approaches.

The project motivation comes from the idea of using a swarm of centimeter sized robots that are battery-free to obtain a highly accurate mapping of an environment. Solar cells placed on a robot provide the required power for motion during exploration, and for communication to a “central hub” that is placed in the near vicinity of the environment to be explored. Each robot is capable of transmitting localization

Table.1. System and Performance Comparison

	Mapping Accuracy	Battery free	Weight	Total Power	Computing Power	Reference
This work	<10-20cm	Yes	<3g	<50mW	<1mW	
1	<30cm	No	46g	8.96W	960mW	[10]
2	8-12cm	No	35.68g	10W	200mW	[9]
3	10-15cm	No	34.8g	5-10W	240mW	[8]
4	8-10cm	No	44g	5-10W	350mW	[7]
5	2.14cm	No	>2kg	100W	30W	[6]

Fig. 1. System and Performance Comparison

data through Ultra-wideband (UWB) communication to the hub. UWB provides centimeter-scale accuracy in localization.

D. Related Work

Collaborative mapping for resource-constrained robot swarms has been explored through various approaches. PGO-LIOM [6] uses tightly coupled LiDAR-inertial odometry with parallel, gradient-free optimization for high-accuracy trajectory estimation on low-power platforms. NanoSLAM [7], designed for miniature robots, runs on ultra-low-power GAP9 SoCs, achieving real-time SLAM with minimal energy consumption. A distributed mapping system for nano-drones [8] employs graph-based SLAM and ICP algorithms on ARM Cortex-M microcontrollers, enabling consistent mapping without external infrastructure. Minimal navigation solutions for tiny flying robots [9] support efficient exploration in constrained environments, indirectly benefiting mapping tasks. Ultra-lightweight collaborative SLAM [10] leverages UWB technology for inter-robot communication, enabling accurate and scalable swarm mapping. Despite these advancements, intelligent navigation and energy-efficient networking in battery-free microrobots remain underdeveloped. Our solution addresses this gap by integrating solar-powered robots with localization techniques optimized for minimal energy consumption.

II. Problem and Mathematical Formulation

A. State Space

We define the state vector for a multi-agent system with n robots. We can define a substate $\hat{x}_n(t)$ that stores the substate for each individual robot. The system state will be the collection of these substates.

$$x_n(t) = \begin{bmatrix} x_n(t) & y_n(t) & \theta_n(t) & b_n(t) & (V_C)_n(t) & (V_p)_n(t) \end{bmatrix} \tag{1}$$

This substate of robot n has these six real number values: $x_n(t)$ for the x position, $y_n(t)$ for the y position, $\theta_n(t)$ for the heading, $b_n(t)$ for the button sensor state (0 or 1), $(V_C)_n(t)$ for the capacitor voltage, and $(V_p)_n(t)$ for the photoresistor voltage. These are continuous substates. So we form the system state as the collection of n substates.

$$x(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{bmatrix} \tag{2}$$

This state has a size of $x(t) \in \mathbf{X}, |\mathbf{X}| = \mathbb{R}^{6n}$ for the $6n$ real numbers represented by our state.

B. Action Space

The action space represents the movement actions all the robots can take. We have no system dynamics for our simple map and only need to define the control function $u(t)$. We also include a Gaussian noise term $w(t)$ associated with the process noise. This is drawn from a Gaussian with covariance Q. $w(t) \sim \mathcal{N}(0, Q)$

$$x(t+1) = Ax(t) + Bu(t) + w(t) \tag{3}$$

C. Output (Observation) Space

We define the output space based on the UWB communication data and the BLE data transmitted by robots. We have measurement noise $v(t) \sim \mathcal{N}(0, R)$

$$z(t) = Cx(t) + v(t) \tag{4}$$

The C matrix will map the state to a measured state based on the BLE and UWB communications.

D. State Estimation (Kalman Filter)

The Kalman filter is estimating the x-y position of an individual robot. Each robot will have their own Kalman filter as described here (indicated by subscript n). This is represented as the position and heading values in the state. We first have the state/covariance prediction step:

$$\hat{x}_{n,t+1|t} = \hat{x}_{n,t|t} + Bu_n(t) \tag{5}$$

$$P_{n,t+1|t} = P_{n,t|t} + Q_n \tag{6}$$

then the update step:

$$\hat{x}_{n,t|t} = \hat{x}_{n,t|t-1} + K_n(t)[z_n(t) - C\hat{x}_{n,t|t-1}] \tag{7}$$

with Kalman gain

$$K_n(t) = P_{n,t|t-1}C_n^T(C_nP(n,t|t-1)C_n^T + R_n)^{-1} \tag{8}$$

and covariance update

$$P_{n,t|t} = (I - K_n(t)C_n)P_{n,t|t-1} \tag{9}$$

E. UWB Localization

An ultra-wideband (UWB) localization system uses a combination of two components, anchors and tags, to localize robots. Each robot is fitted with a tag that emits UWB signals that are received by multiple anchors placed in the operational space of the robots. The time of flight of the signals between the tags and anchors can be used to precisely localize the robots.

Utilizing multiple anchors, the absolute position can be determined using these ToF measurements. This is a two-way communication scheme which requires the tags to receive signals from the anchors. Instead of the ToF two way communication scheme,a time difference of arrival (TDoA) method with multiple anchors can be applied so only one-way communication (from tag to anchors) is necessary. In the TDoA scheme, a tag will emit a "blink" signal that is

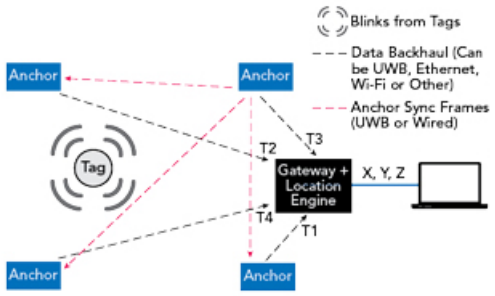

Fig. 2. Time Difference of Arrival (TDoA) UWB Localization [11]

measured by all anchors; these anchors are precisely time synchronized so the difference in blink signal arrivals between the anchors can uniquely determine the localization. This one-way communication cuts down on power requirements for the tags, which is advantageous given the power constraints of our robots.

With the TDoA measurements and three or more anchors, a multilateration algorithm [12] can determine the robot position with on the scale of ± 10 cm [13].

F. Bluetooth Low Energy Communication

The Bluetooth Low Energy (BLE) communication is only used for the hub-robot intercommunication. The robots send their measured quantities (sensor value, capacitor voltage, photoresistor voltage) to the hub, and the hub transmits back movement commands based on the estimated state of the robot. BLE communication is used for communication because of its low power consumption for the robots, on the order of 5 microwatts. This allows us to effectively communicate with our robots without significant power draw on the capacitors of the robots.

G. Frontier Selection Algorithm

A 'Frontier' is a boundary that separates the explored and unexplored regions of the map. The speed of exploration in this method is very high. Redundancy of visiting the same cells is greatly reduced. The objective of this algorithm is used to find an optimal 'goal' coordinate for a robot to move towards.

1) Frontier Representation: Let $\mathbf{M_t}$ be a matrix that mathematically describes the map of the environment the robot lives in at time t. The environment can be discretized into a grid divided into $m \times n$ cells. Hence, the dimensions of $\mathbf{M_t}$ would also be $m \times n$. It should be noted that the aspect ratio of the map should be equal to (m/n).

$$\mathbf{M_t} = \begin{bmatrix} M_{11} & M_{12} & \dots & M_{1n} \\ M_{21} & M_{22} & \dots & M_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ M_{m1} & M_{m2} & \dots & M_{mn} \end{bmatrix} \tag{10}$$

$$M_{ij}(t) = \begin{cases} 0 & \text{if the cell is still unexplored} \\ 1 & \text{if the cell is explored, and is free} \\ 2 & \text{if the cell is occupied by an obstacle} \end{cases}$$

Based on the map matrix $\mathbf{M}$, the set F which contains all the points on the frontier can be defined as:

$$F = \{(i,j) \mid M_{ij} = 1 \text{ and } \textit{atleast} \text{ one adjacent } M_{i'j'} = 0\} \tag{11}$$

In other words, all the frontier points must have been explored by the robot, free of obstacles, and must be adjacent to *atleast* one cell in the grid that has not been explored yet.

2) Choosing the Frontier: Selection of the optimal frontier involves defining and evaluating certain metrics:

- Information Gain: The expected information gain $I(f)$ of a frontier point $f \in F$, is a measure of how much new information could be gained by exploring it. The larger the gain, the better it is to explore the point because of higher number of surrounding unexplored cells.

$$I(f) = \sum_{(i,j) \in \mathcal{N}(f)} \mathbf{1}(M_{ij} = 0) \tag{12}$$

  $\mathcal{N}(f)$ is the set of all neighboring cells of f with respect to the map-matrix $\mathbf{M}$.
- Path Length: The path length $L(f)$ from the current robot position (x_r, y_r) to the frontier position (x_f, y_f) needs to be taken into account. The higher the length, the higher the energy consumption of the robot, and the longer it takes for the robot to reach the frontier. Let the path from the robot to the frontier be given as a sequence of $(n+1)$ coordinates:

$$P_f = \{(x_0, y_0), (x_1, y_1), (x_2, y_2), \dots (x_n, y_n)\}$$

  Here, $(x_0, y_0) \equiv (x_r, y_r)$ and $(x_n, y_n) \equiv (x_f, y_f)$. The sequence P_f can be obtained by using algorithms like Dijkstra or A^*.

$$L(f) = \sum_{i=0}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{13}$$

The overall cost function corresponding to a frontier f can then be given as:

$$C(f) = -\alpha \left(\frac{I(f)}{I_{\text{max}}} \right) + \beta \left(\frac{L(f)}{L_{\text{max}}} \right) \tag{14}$$

where, I_{max} and L_{max} are parameters of feature-normalization. α and β are positive real numbers that act as weights for the individual metrics. The optimal frontier f_{opt} is simply chosen as:

$$f_{\text{opt}} = \arg\min_{f \in F} C(f) \equiv (x_{\text{goal}}, y_{\text{goal}}) \tag{15}$$

H. Artificial Potential Field (APF) Algorithm

The main idea in this algorithm is to manually define "attractive" and "repulsive" potentials within the environment map. The robot navigates using attractive forces towards the goal, and repulsive forces away from the obstacles and boundaries. This is a very common and useful approach to avoid obstacles, by carefully assigning obstacle potentials.

The destination has the lowest potential, and the obstacles have the highest. Intuitively, the robot moves from higher to lower potentials (in the direction of decreasing gradient of the potential field). A potential field is a physical field that obeys Laplace's equation. In other words, the force field associated with such a potential field ϕ can be evaluated by taking the gradient of the potential field.

$$\vec{F} = -\nabla\phi$$

1) Attractive Force: The potential field corresponding to the goal can be given as:

$$\phi_{\text{goal}}(x, y) = c\sqrt{(x - x_{\text{goal}})^2 + (y - y_{\text{goal}})^2} \tag{16}$$

where, (x, y) is the coordinate of the robot (node) and $(x_{\text{goal}}, y_{\text{goal}})$ is the coordinate of the 'goal'. The goal coordinates are obtained at any point in time using the Frontier Selection algorithm. c is a positive constant. Also, the force field can then be obtained as:

$$\vec{F}_{\text{goal}} = \left(\frac{-c}{\sqrt{(x - x_{\text{goal}})^2 + (y - y_{\text{goal}})^2}}\right)\begin{bmatrix}(x - x_{\text{goal}})\\(y - y_{\text{goal}})\end{bmatrix} \tag{17}$$

The formulation is such that the potential field ϕ_{goal} has a minima at $(x_{\text{goal}}, y_{\text{goal}})$

2) Repulsive Forces: Two kinds of repulsive forces are considered here, produced by the following sources:

1) The boundaries of the considered environment
2) All the obstacles in the environment

Boundary Potential:
The repulsive potential field ϕ_b caused by the boundaries can be formulated as follows:

$$\phi_b(x, y) = \frac{1}{\delta + \sum_{i=1}^{s}(g_i + |g_i|)}$$

where, δ is a small positive number to avoid ϕ_b going to ∞, s is the total number of boundary segments, and g_i is the linear equation that represents the boundary i mathematically. Clearly, one can observe that there is a maxima for ϕ_b at each boundary.

Obstacle Potential:
Let the repulsive potential field generated by obstacle k be given by ϕ_k. If the obstacle k is a square of side l_k with center at (x_k, y_k), then ϕ_k is formulated as:

$$\phi_k(x, y) = \frac{\phi_{\max}}{1 + h(x, y)}$$

where, $h(x, y) = (x_k - l/2 - x) + |x_k - l/2 - x| + (y_k - l/2 - y) + |y_k - l/2 - y| + (x + 1 - x_k - l/2) + |x + 1 - x_k - l/2| + (y + 1 - y_k - l/2) + |y + 1 - y_k - l/2|$. Also, $\phi_{\max}$ is the maxima of the function ϕ_k for all values of k. The function h captures the side of the obstacle boundary the robot is in, and assigns potential accordingly. Finally, the obstacle potential ϕ_{obs} can be calculated as follows:

$$\phi_{\text{obs}} = \max\{\phi_k\}_{k=1}^{K}$$

where, K is the total number of obstacles in the environment.

3) Robot Trajectory: Now, the net force on the robot at time t can be calculated as follows:

$$\vec{F}_{\text{net}} = \vec{F}_{\text{goal}} - (\nabla\phi_b + \nabla\phi_{\text{obs}})$$

Hence, the robot's subsequent direction of motion should be given by the following direction vector: $\hat{d} = \cos\theta\,\hat{i} + \sin\theta\,\hat{j}$, where

$$\theta = \tan^{-1}\left(\frac{\vec{F}_{\text{net}} \cdot \hat{j}}{\vec{F}_{\text{net}} \cdot \hat{i}}\right)$$

III. System Design

We proposed a comprehensive framework for the coordination of swarm robots, with the overall objective of creating an accurate mapping of the environment. This system leverages Ultra-Wideband (UWB) localization, Bluetooth Low Energy (BLE) communication, and energy-aware operations to ensure sustainable energy usage, precise localization, and accurate mapping. The general architecture of the system is shown in Figure 3, providing a clear description of the components and their interactions. The system consists of two main components: the central hub and the sensor nodes, which work in a coordinated pipeline. After map creation, the central hub manages sector assignment, leader selection, data integration, state estimation, mapping, and swarm coordination. Processes information from sensor nodes to perform state estimation, mapping, and path planning and sends updated instructions to guide robot actions. Sensor nodes execute assigned tasks, gather data, and transmit it back to the hub while harvesting energy.

A. Sector Assignment and Leader Node Selection

Due to the high power consumption of UWB localization, we needed to identify the most efficient way to duty cycle our localization communications. To achieve this, we first divide the map into sectors. Depending on the map size, we split the generated map into equal-sized sectors and place the sensor nodes accordingly to begin the simulation. The sensor node swarm is initially dispersed evenly across all sectors to ensure uniform spacing. Once the map is divided into equal-sized sectors, leader nodes are selected for each sector.

The role of a leader node is to manage localization communication within its assigned sector. Our method for selecting leader nodes is currently based on the power levels of all robots within each sector. At each time step, the central hub evaluates the power levels of all robots in each sector. The robot with the highest power level is designated as the leader node. If the selected leader node's power level is above the threshold required for UWB communication, it performs localization. Otherwise, the robot will move to a power harvesting mode

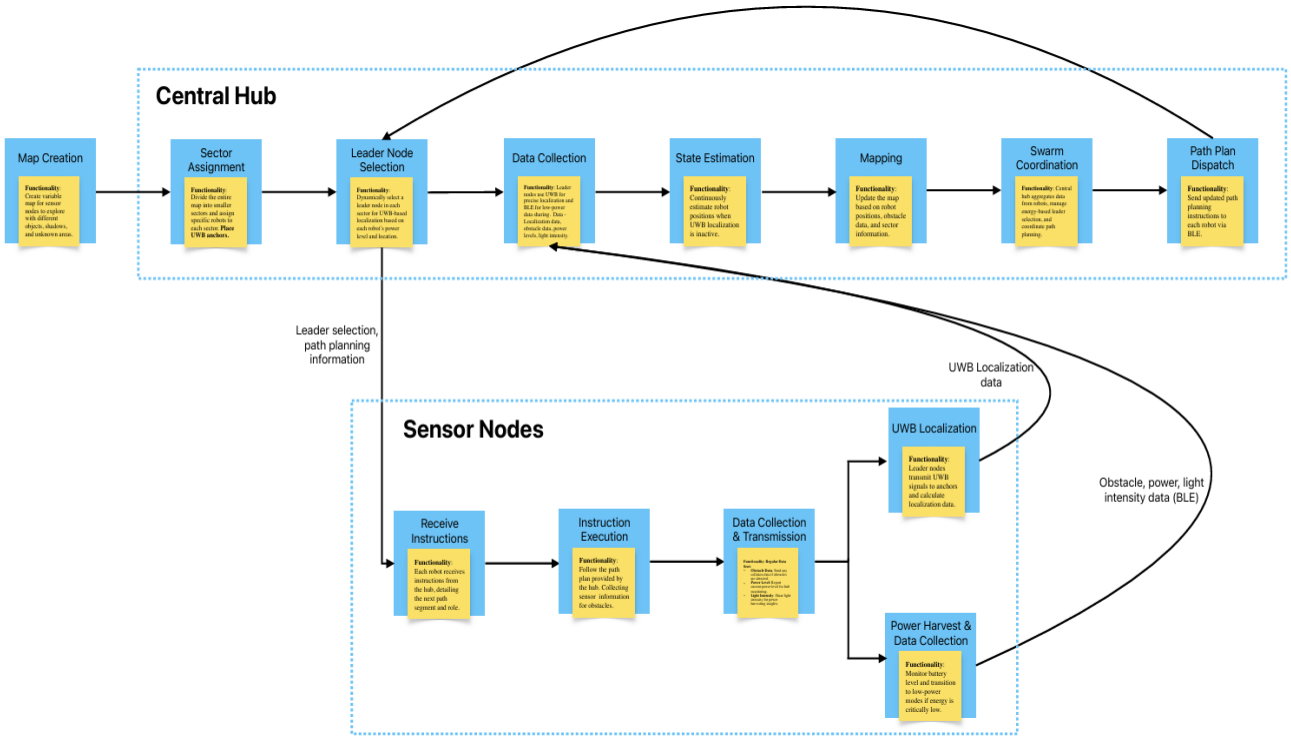

Fig. 3. General Block Diagram

until a new leader is selected or it accumulates enough power to perform localization using UWB.

B. State Estimation and Mapping

When UWB localization is temporarily unavailable, the central hub uses a Kalman Filter to estimate the positions of the robots. This estimation relies on the history of previous state estimates and collision data, which are combined to continuously update the map.

When UWB localization becomes available, the measured positions of the robots (z) are incorporated into the state estimator as observations. These measurements refine the Kalman Filter's predictions, updating the robot's current state, including position (x, y), orientation, and velocity. This process ensures the state estimates remain accurate and aligned with the robots' actual trajectories.

Despite sparse measurement updates, the state estimation remains effective because the robots move slowly and predictably. Their motion is linear, and the only control input is the orientation angle, making the system dynamics straightforward. This simplicity allows the Kalman Filter to maintain reliable estimates of the robots' states, even when direct localization data is infrequent.

C. Swarm Coordination

The central hub acts as the core of decision-making of the system. It aggregates data received from all robots, and performs path planning algorithms and develops an optimized path for each robot. The swarm coordination begins with collision detection. If a collision is detected, the system triggers path recalculations; otherwise, the robot continues on its current trajectory. For detected collisions, frontiers for exploration are identified. The system prioritizes unexplored locations while avoiding overlaps with other robots, ensuring optimal coverage of the environment. The system computes the potential field forces to guide the robot's path. Once computed, the central hub recalculates the robot's path, which ensures efficient navigation of the robot to its target while avoiding collisions. This process ensures efficient swarm coordination while maintaining optimal exploration and resource usage. After path calculation, the central hub then communicates updated path instructions as well as role information to individual robots via BLE.

D. Robot Instruction Reception and Execution

Figure 4 illustrates the detailed process of instruction reception and execution by each robot, ensuring seamless coordination and efficient navigation within the swarm.

- **Instruction reception:** Each robot receives instructions from the central hub, detailing the next path segment and its assigned role. These instructions are transmitted via BLE to maintain efficient communication. Upon receiving instructions, robots immediately begin their assigned tasks while monitoring for potential collisions.
- **Instruction execution:** Robots moves along the path provided by the central hub while concurrently detecting the collisions. If the attached button sensor is pressed, which indicates a collision, the robot transmits the collision data

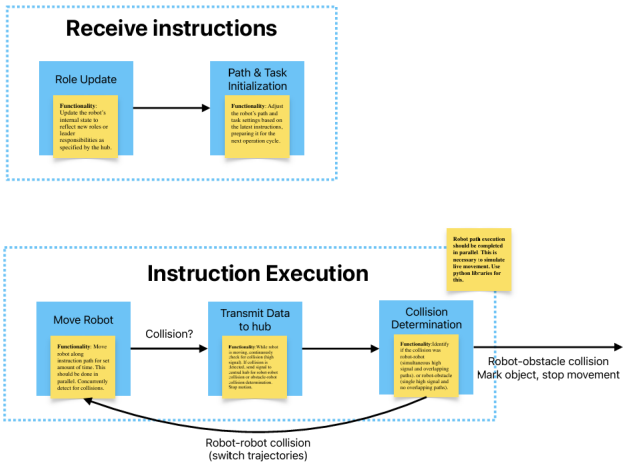

Fig. 4. Per Robot Instruction Execution

to the hub. Simultaneously, it moves one step backwards and stops motion to prevent energy wastage and further disruptions, waiting for the next instruction to execute.

E. UWB localization

If one robot has enough power for UWB localization, it may be assigned as a leader and transmit UWB signal to stationary anchors. The localization data will finally be collected and calculated in the central hub, enabling precise localization of that robot. This high-resolution positioning data is essential for accurate state estimation and map creation as well as effective swarm coordination.

F. Power Harvest and Lower Power Action

Robots are equipped with solar cells for power harvesting to realize battery-free operation. They continuously monitor their energy levels and transition to low-power modes if energy is critically low, informing the hub by sending a BLE signal and stopping all actions except power harvesting. The robot remains in low-power mode until its energy level exceeds a predefined threshold, at which point it resumes active mode and informs the hub of that.

G. Regular Data Collection and Transmission

Each robot regularly transmits data to the central hub using BLE for efficient communication. Data transmission operates on a duty cycle to conserve energy and maintain system efficiency. The transmitted data includes sensor readings of light intensity, button sensor states for collision detection, and the robot's current power level for leader selection. The duty cycle ensures that data updates are sent periodically to ensure efficient energy usage and avoid overwhelming the communication network.

IV. Simulation Setup

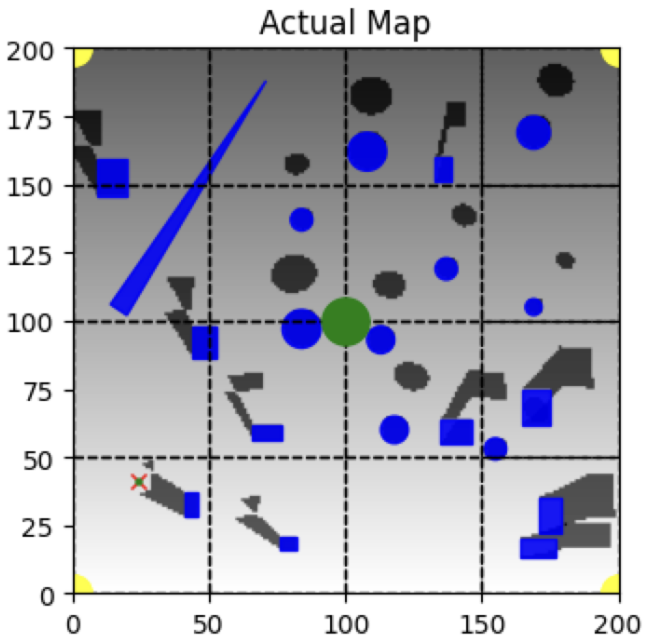

Fig. 5. Initial Mapping of the Simulation

For the purpose of testing our algorithms, we use a parameterized environment map with configurable dimensions, number of randomly generated obstacles, mode of lighting, and the number of sectors. In addition to this, several parameters including the number of nodes (robots), the time step between each state estimate, the noise in the estimation and observation, and the power consumption due to motor rotation, UWB & BLE communication have been included as variables that can be changed during the simulation. All of these variables are included in a graphical user interface that allows users to quickly run simulations with varying input parameters.

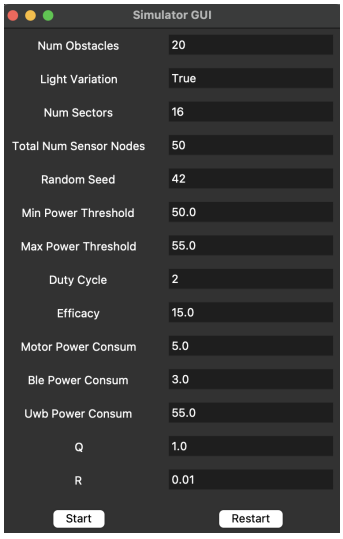

Fig. 6. Simulation Configuration Menu

We use 3 distinct metrics to judge the performance of a simulation.

1) Mapping Accuracy: At each time step, the list of collisions of nodes with obstacles or boundaries is updated. We compare these collision points with the actual map obstacles and calculate the ratio of the true collisions, which gives us the accuracy of the mapping till that time.
2) Swarm Low Power Ratio: Using a minimum threshold for the power level of the robots, we assign power modes to them. Robots go into low power mode if their power level is below the threshold. These particular robots cannot transmit UWB signals. Since we do not want a majority of the swarm to go into low power mode and potentially die, we come up with a metric that captures the total number of robots that are in low power mode compared to the overall number of robots in the environment.
3) Map Coverage: This metric calculates the total area explored by the swarm divided by the entire area of the map.

V. Results

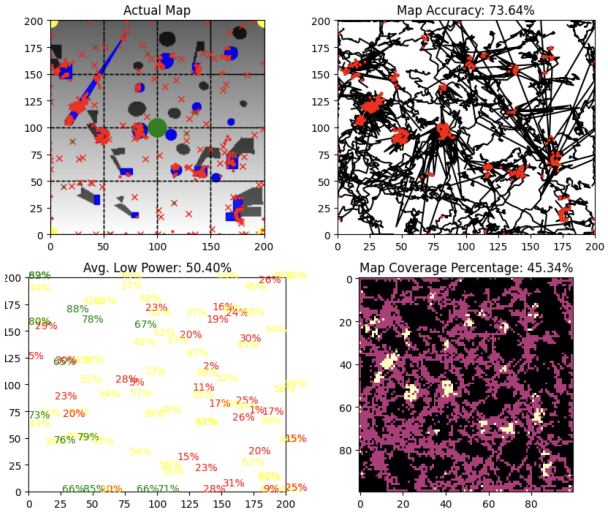

Fig. 7. Example Simulation Maps with 100 Robots

Figure 7 shows the four different maps that capture the simulation behavior and metrics at each time step. We have maps for the actual operational space (configured in the setup), the mapping as measured by the collisions (along with robot paths), a mapping of the power levels for each sensor node, and the map coverage. Figure 7 is an example with 100 sensor nodes.

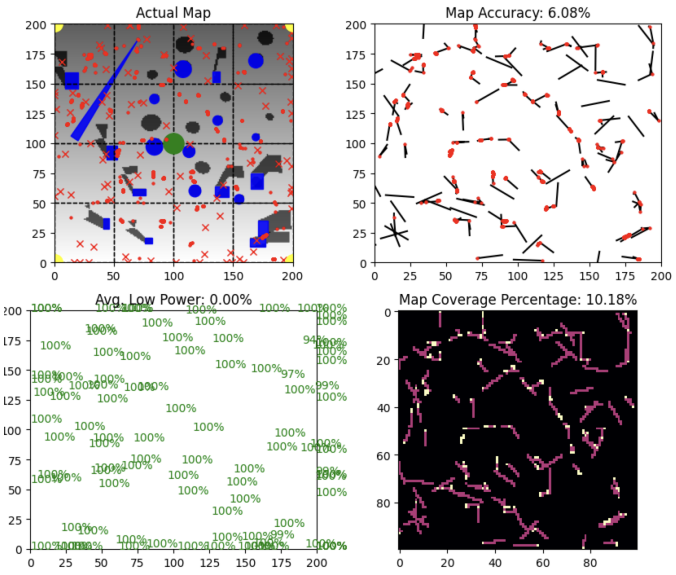

Fig. 8. Simulation under Random Walk and No UWB Localization

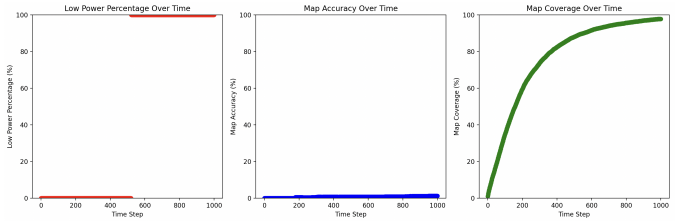

Fig. 9. Performance under random walk without UWB localization

The baseline performance of a 100 sensor node system under a random walk algorithm without UWB localization had very poor map accuracy, only reaching around 6% accuracy after 1000 time steps. The low power percentage metric shows that all robots went into low power mode around time step 550; since they were not UWB localizing this step to 100% indicates that all robots ran out of power from movement at nearly the same time step; this would be caused by insufficient power harvesting. The random walk is still able to achieve 100% map coverage over 1000 time steps.

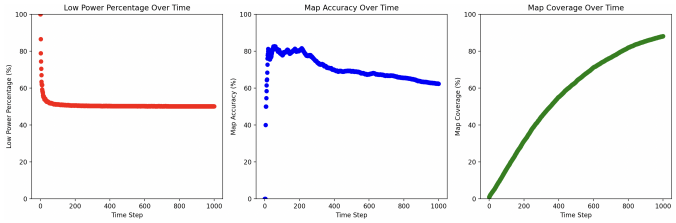

Fig. 10. Performance under frontier selection and APF

With the introduction of the frontier selection and APF algorithm and UWB localization, there are many differences in performance metrics. All robots start in low power mode, but the average slowly stabilizes to around 50%, indicating that

half of the robots are able to UWB localize on average. The map accuracy dramatically increased compared to the random walk, peaking at around 80% map accuracy. The decrease in map accuracy down to 70% is attributed to the number of localizations decreasing over time, leading to collision detection events being associated more heavily to the state estimates and less accurate collision positions.

VI. Conclusion

Our simulation has shown that localization of battery-free robot swarms is feasible under the power constraints that describe our robots. By leveraging UWB for precise localization and Bluetooth Low Energy (BLE) for efficient communication, we demonstrated the viability of localization and mapping under strict power constraints and created a viable testbed for future simulations of these systems.

VII. Future Work

The aim of this project is to integrate with hardware to demonstrate that battery-free, small-scale autonomous robots can localize and generate terrain maps using UWB localization. We aim to work in collaboration with Ankur Mehta through the LEMUR Lab and Professor Vikram Iyer from the University of Washington to enable robust control systems for autonomous battery-free robot sensor nodes.

Areas for Improvement: For the algorithm simulation component of this work, there are several areas for improvement to make the system more robust and better reflect real-world scenarios:

a) Realistic Parameter Matching.: One way to improve the simulation is to match our parameters with real-world values for UWB, BLE, and motor power consumption. By doing so, we can make our simulations more realistic and better understand how the hardware will interact with real-world environments.

b) Robust Anchor Node Placement.: UWB and BLE communication are constrained by strict distance limitations, which introduce challenges for efficient anchor node placement. To address this, we plan to implement a communication range model and dynamic anchor node placement. Simulating these constraints will help us optimize anchor node placement for improved communication efficiency.

c) Improved Leader Node Selection.: Currently, leader nodes are selected based only on the stored power levels of nodes within a sector. While effective for energy efficiency, this approach does not consider the nodes' locations. We aim to improve the leader node selection algorithm by incorporating the distance to unexplored regions, allowing us to select leaders that are better positioned for coverage expansion.

d) Shadow Mapping and Avoidance.: We also want to enable robots to autonomously map and avoid shadows or regions lacking sufficient light. This improvement will enhance the robots' power harvesting capabilities and reduce overall swarm energy consumption.

These improvements will make the simulation more robust and reflective of diverse environments and dynamic conditions. By enhancing the realism and functionality of the simulation, we can better inform hardware design decisions and component selection for future robotic systems.

APPENDIX

GitHub: https://github.com/kbaseba/UWB_low_power_localization

REFERENCES

[1] H. D. Escobar-Alvarez, N. Johnson, T. Hebble, K. Klingebiel, S. A. Quintero, J. Regenstein, and N. A. Browning, "R-advance: rapid adaptive prediction for vision-based autonomous navigation, control, and evasion," *Journal of Field Robotics*, vol. 35, no. 1, pp. 91–100, 2018.

[2] M. Thackeray, C. Wolverton, and E. Isaacs, "Energy environ sci 5: 7854 414 n," *Saxena et al*, 2012.

[3] D. H. P. Kang, M. Chen, and O. A. Ogunseitan, "Potential environmental and human health impacts of rechargeable lithium batteries in electronic waste," *Environmental science & technology*, vol. 47, no. 10, pp. 5495–5503, 2013.

[4] K. Johnson, Z. Englhardt, V. Arroyos, D. Yin, S. Patel, and V. Iyer, "Millimobile: An autonomous battery-free wireless microrobot," in *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 2023, pp. 1–16.

[5] K. Johnson, V. Arroyos, A. Ferran, R. Villanueva, D. Yin, T. Elberier, A. Aliseda, S. Fuller, V. Iyer, and S. Gollakota, "Solar-powered shape-changing origami microfliers," *Science Robotics*, vol. 8, no. 82, p. eadg4276, 2023.

[6] H. Shen, Q. Zong, B. Tian, X. Zhang, and H. Lu, "Pgo-liom: Tightly coupled lidar-inertial odometry and mapping via parallel and gradient-free optimization," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 11, pp. 11 453–11 463, 2022.

[7] V. Niculescu, T. Polonelli, M. Magno, and L. Benini, "Nanoslam: Enabling fully onboard slam for tiny robots," *IEEE Internet of Things Journal*, 2023.

[8] C. Friess, V. Niculescu, T. Polonelli, M. Magno, and L. Benini, "Fully onboard slam for distributed mapping with a swarm of nano-drones," *IEEE Internet of Things Journal*, 2024.

[9] K. McGuire, C. De Wagter, K. Tuyls, H. Kappen, and G. C. de Croon, "Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment," *Science Robotics*, vol. 4, no. 35, p. eaaw9710, 2019.

[10] V. Niculescu, T. Polonelli, M. Magno, and L. Benini, "Ultra-lightweight collaborative mapping for robot swarms," *arXiv preprint arXiv:2407.03136*, 2024.

[11] M. Viot, A. Bizalion, and J. Seegars, "Exploring ultra-wideband technology for micro-location-based services," *Microwave Journal*, June 2021, accessed 15 June 2021. [Online]. Available: https://www.microwavejournal.com/articles/36143-exploring-ultra-wideband-technology-for-micro-location-based-services

[12] "Multilateration," https://www.sciencedirect.com/topics/engineering/multilateration, accessed 19 Nov. 2024.

[13] Qorvo, "Product documentation for qm33120w," accessed 19 Nov. 2024. [Online]. Available: https://www.qorvo.com/products/p/QM33120Wdocuments