Grey Wolf Optimization-Based Decision-Control for Multi-Robot Signal Source Localization in Communication-Constrained Environments

Xuanwei Zhang
College of Information Science and Technology of Beijing University of Chemical Technology, Beijing, 102200, Beijing, China
E-mail: zxwzxw213213@163.com

Keywords: signal source localization; multi-robot system; Hazardous Environment; consensus control; GWO

Received: June 30, 2025

By developing and evaluating a decision-control strategy using a Hazardous Environment method, this study addresses the issue of signal source localization utilizing a group of autonomous robots. A new grouping technique is the basis of our proposed algorithm, Optimal Weighting Grey Wolf Optimization. The OW-GWO algorithm takes into account the past, present, and future ideal positions of all grey wolves, ranks them according to these positions, and updates these positions as needed. The alpha wolf constantly estimates where the prey is, and the rest of the grey wolves follow suit. Integrating the OWGWO method with an enhanced grouping strategy and dividing the algorithm into two stages—the random walk stage with the dynamic grouping stage—allows us to address the multi-target issue of swarm robots search. In the random walk phase, grey wolves adjust their best placements based on history and travel at random. The OWGWO algorithm builds search auxiliary points throughout the dynamic grouping stage by using a better grouping approach that takes into account the previous ideal placements of individual grey wolves. In order to hunt for various prey, grey wolves use these to form groups. . A decision level along with a control level is both included in the proposed decision-control method. The decision level employs a particle filter to make educated guesses about where the signal could be coming from. The actual location of the signal source becomes closer to the predicted position as the robots move. At the level of control, a consensus controller is suggested for commanding several robots to locate a signal source according to the predicted location of the source. Additionally, in order to alleviate some of the communication load, a Hazardous Environment mechanism is developed. Lastly, experiments and simulations demonstrate that the suggested decision-control method using the Hazardous Environment strategy solves the signal source localization issue well.

Povzetek: Prispevek predstavi strategijo vodenja skupine avtonomnih robotov za lokalizacijo signalnega vira v nevarnem okolju, ki z optimizacijskim algoritmom OW-GWO in dinamičnim grupiranjem izboljša učinkovitost iskanja več ciljev.

1 Introduction

The signal source localization issue has attracted a lot of interest for multi-robot systems as an alternative to the single robot, due to their vast detection range with simultaneous sampling. One typical approach to estimating the location of the signal source is the combined gradient estimation of the signal strength distribution. This method involves directing the movement of multiple robots to take measurements at different locations at the same time, allowing for the neglect of some unnecessary trajectories [1]-[4]. As an example, we suggested distributed control techniques for locating a noisy signal source and used a Finite-Difference approach to estimate the signal intensity gradient at the formation centroid. In, the search environment was modeled by Lu using a radial basis function network. Then, using gradient information supplied by the model, the robots were instructed to approach the signal source. Similarly, consensus control theory has informed the development of some cooperative control systems. Also, crisis management is an area where the concept of shared control is expanded upon. As an example, in, Garca-Magariño put out a method for citizen cooperation to identify issue causes via the utilization of a global map and peer-to-peer communication. The aforementioned methods, however, need longer measurement collection times from the robot at various sites.

Additionally, it is often unnecessary for these methods to provide certain search trajectories [5]-[9].

Two potential problems with the previously suggested methods for solving the signal source localization problem need to be highlighted [10]. One problem is that local optima may be readily achieved using the gradient estimation approach because to its sensitivity to noise. The uncertainty problem caused by noise may be addressed using a particle filter technique. Another problem with multi-robot systems Nomenclature is that there is a lack of available communication capacity for each robot. This problem may be solved by implementing an event-triggered strategy that shortens the time it takes for each robot to communicate [11]. Noteworthy among the suggested rules for multi-robot systems are those that are activated by events. Having said that, event-triggered rules like this primarily serve to save computing resources. Maintaining system stability in multi-robot systems still requires continual communication mechanisms [12]-[17].

Symbol / Term	Description
SLAM	A method that allows robots to simultaneously estimate their location and map unfamiliar areas is known as simultaneous localization and mapping.
DRL	Deep Reinforcement Learning is a method of learning that maximizes control policies by use of exploration motivated by rewards.
LSTM	The Long Short-Term Memory model is a recurrent neural network that can adapt to a changing environment and forecast temporal sequences.
GWO	The Grey Wolf Optimizer is a metaheuristic algorithm that takes its cues from the natural world, namely the hunting behavior and social structure of grey wolves.
OW-GWO	Fair Weighting An enhanced version of Grey Wolf Optimization, GWO incorporates a weighted rating of wolves according to their current, historical, and prospective positions.
α (Alpha) Wolf	Wolf pack leader in GWO who directs the pack's quest for the best possible location to catch prey.
β (Beta) Wolf	Helping the alpha with decision-making and search refining as a second-level wolf.
δ (Delta) Wolf	A wolf of the third rank who follows α and β and leads the pack.
ω (Omega) Wolves	Following their leaders, the surviving wolves scour the search area for the best possible answers.
RMSE	A statistical metric for assessing the precision of signal source localization predictions is the Root Mean Square Error.
SR (%)	Rate of Success—the proportion of times the multi-robot system was able to successfully localize its target.
RE (%)	The term "route efficiency" refers to the ratio between the ideal and actual lengths of paths that robots follow.
CT (s)	The merging of the amount of time it takes for the swarm to get to the starting point.
CO (MB/s)	The average amount of data exchanged per second between agents operating in a swarm is known as communication overhead.
PF	The decision layer uses the particle filter, a probabilistic estimating method, to anticipate the location of the source.
CE	Distributed control technique that guarantees consistent agreement among robots in decisions is the consensus estimator.
HE-DRL	Potentially Dangerous Setting Deep Reinforcement Learning—a kind of DRL designed to deal with ambiguity and incomplete data.
V-SLAM	Visual SLAM is a kind of SLAM that maps and perceives its surroundings using visual sensors, such as cameras.
$\hat{x}(t)$	This is the predicted location of the signal at time t.
$x(t)$	True location of the signal source at time t.
Δx	Localization error among estimated and real locations.
η	Acceleration of learning or factor for adaptability in DRL model.
γ	Reinforcement learning discount factor as a subsequent reward.

ρ	Issue of communication dependability in potentially dangerous environments.
N_a	Amount of swarm-based autonomous robots.
d_{max}	What is the maximum distance that agents may communicate?
P_{tot}	The overall amount of electricity used by the swarm system while it is operating.

A. Contributions

A number of Hazardous Environment systems have been developed to save communication resources and lessen the computational and communication loads. Unfortunately, an answer is yet unavailable for the signal source localization issue that combines a Hazardous Environment system with a cooperative control technique and particle filter methodology. The provided cooperative control strategy presents a number of challenges, one of which is the design of Hazardous Environment rules. The second obstacle is figuring out how to use the Hazardous Environment rule to derive stability criteria for multi-robot systems using the cooperative control technique that has been suggested. Thus, the current work is motivated by the need to address how to create the decision-control technique for the signal source localization issue while considering the aforementioned constraints. Two benefits characterize the suggested decision-control method. The implementation of a Hazardous Environment strategy has the potential benefit of reducing communication delays and the frequency of control input updates, which in turn saves communication and chip resources. Additionally, the multi-robot system's detection data may be effectively used through the particle filter with cooperative controller to approximate the signal source's location.

B. Novelty and technical innovations

An innovative approach to multi-robot search has been developed by combining state-of-the-art optimization, control, and machine learning methods:

- OW-GWO, or Optimal Weighting Grey Wolf Optimization, is a novel writers' method for grouping. It differs from conventional GWO in that it takes into account the ideal positions of the "grey wolves" (robots) in the past, present, and future while ranking and updating their positions. The multi-robot search technique relies on this improvement.
- There are a random walk phase and a dynamic grouping phase in a two-stage search algorithm. Swarm robot search's multi-target problem may be tackled in a new manner using this structured method and the OW-GWO.
- Hazardous Environment Mechanism (DRL using LSTM): A new way to reduce robots' communication burden is to employ a Deep Reinforcement Learning (DRL) technique combined by a Long Short-Term Memory (LSTM) network. This is very important for scalability and real-time performance, particularly in difficult circumstances.
- The suggested approach divides the system into two main levels: the Decision Level, which uses a Particle Filter to estimate the source, and the Control Level, which uses a Consensus Controller to navigate. This architecture allows for integrated decision-control processing. Together, these distinct features offer a solid basis for concerted effort.

C. Key findings of the study

The following important conclusions were shown by the research, which confirmed the efficacy of the suggested system via tests and simulations:

- Successful Signal Source Localization: The proposed decision-control approach successfully resolves the signal source localization problem.
- Improved Convergence: The real signal source location becomes closer to the anticipated position as the robot’s advance, thanks to the Particle Filter at the decision level, which improved convergence.
- Coordinated Control: The many robots were effectively directed by the Consensus Controller to find the signal source using the projected position, showcasing efficient coordinated movement.
- Communication Efficiency: An essential discovery for developing scalable multi-robot systems is the improvement in communication efficiency, which was intended to be a byproduct of incorporating the Hazardous Environment method (DRL with LSTM).

In order to assess the suggested Decision-Control Strategy utilizing the Hazardous Environment (DRL + LSTM) approach along with Optimal Weighting Grey Wolf Optimization (OW-GWO) for signal source

identification under visual SLAM, here is a numerical comparison summary of the study you mentioned, with an emphasis on the success rate, route efficiency, and RMSE performance metrics.

D. Paper structure

Here is how the rest of the article is structured. The second section provides a concise overview of the basics of the communication topologies and dynamics of mobile robots. Section 3 detail our plan to coordinate the movement of mobile robots in order to pinpoint the location of the signal source using a particle filter-based position estimation and a Hazardous Environment mechanism. We shall demonstrate the efficacy of the suggested decision-control method using the Hazardous Environment system in Section 4 via simulation and in Section 5 through experimental data. Section 6 will serve as the paper's conclusion.

2 Related work

In order to increase the effectiveness of odor source search while preventing collisions, researchers from [21] suggested a distributed behavior control strategy for indoor multi-robot collaborative odor source localization. For situations like drug screening, fire rescue, and others like it, that challenge is critical. In a spiraling motion, a number of robots seek for the plume. By exchanging data on the spot with the greatest concentration of detected odors, they may speed up the system's plume detection process by directing gas-detecting robots to that spot. In order to pinpoint where the smell is coming from, a number of robots use particle filters. They determine where the smell is coming from by establishing limits for particle convergence with distance convergence. An efficient odor source search may be accomplished with the help of a state machine, which allows for the flexible selection of the most suitable approach based on the present environment and job requirements.

A distributed cooperative navigation technique is presented by the authors of [22], who overcome these restrictions. The first step is to apply a Distributed Adaptive Extended Kalman Filter. That filter can handle multi-source noise and nonlinearities by adaptively adjusting the noise covariance matrix. Later on, a framework called Distributed Model Predictive Control is presented. By optimizing and predicting the kinematic models of each robot, that framework enhances the system's capacity for joint operations and dynamic decision-making. Robot tests and comprehensive simulations validate the strategy's effectiveness. Simulation findings show that DAEKF achieves better localization accuracy than Kalman Filter with Extended Kalman Filter when it comes to cooperative localization. Using DAEKF, the two robots significantly decreased their lateral and heading errors in the straight-line path-tracking trials. The lateral error Root Mean Squared Error for Robot 1 was decreased by 68.87%, 27.80%, and 25.76% when using DAEKF instead of No Filtering, KF, or EKF, respectively. By using DAEKF, the RMSE was decreased with 52.29%, 41.89%, along with 36.47% in the heading error case. In comparison to No Filtering, KF, and EKF, DAEKF decreased the lateral error RMSE for Robot 2 through 51.30%, 22.88%, & 11.60%, respectively.

A human-operated multi-robot system for source localization was described in [23] by the authors. The system included both ground and airborne robots with varying degrees of autonomy. The technology allows a human supervisor to manage several robots via augmented reality interfaces. That human-robot interface's primary function is to facilitate the cooperative, real-time control of diverse robot groupings. To free up the operators for more complex duties, it avoided obstructions using sophisticated route planning algorithms. The surroundings and barriers are known to each robot, so it can autonomously construct a route to any user-selected objective without any collisions. In the video view, it showed sensor data from each robot individually. Additionally, an augmented reality perspective that incorporates sensor data is shown, which assists users in locating information sources or the operator in accomplishing mission objectives. That study delves into an early Human Factors assessment of the system, testing various interface circumstances for source task identification. Mission completion durations for target detection missions were shown to be lower using the innovative Augmented Reality multi-robot control (Point-and-Go with Path Planning) as compared to the standard joystick control. Additionally, they look at operator workload analysis and usability testing.

Researchers in [24] demonstrated a human-operated multi-robot system that could locate sources with relative ease using ground and airborne robots with some degree of autonomy. The technology allows a human supervisor to manage several robots via augmented reality interfaces. That human-robot interface's primary function is to facilitate the cooperative, real-time control of diverse robot groupings. To free up the operators for more complex duties, it avoided obstructions using sophisticated route planning algorithms. The surroundings and barriers are known to each robot, so it can autonomously construct a route to any user-selected objective without any collisions. In the video view, it showed sensor data from each robot individually. Additionally, an augmented reality perspective that incorporates sensor data is shown, which assists users in locating

information sources or the operator in accomplishing mission objectives. That study delves into an early Human Factors assessment of the system, testing various interface circumstances for source task identification. Mission completion durations for target detection missions were shown to be lower using the innovative Augmented Reality multi-robot control (Point-and-Go with Path Planning) as compared to the standard joystick control. Additionally, they look at operator workload analysis and usability testing.

Current techniques' authors rely heavily on responsive stimuli or simple gas plume models to plan robot movements [25]. The patchy distribution of real-world gases makes these methods ineffective, even if they work well in idealized, simulated systems. They present SniffySquad, an olfactory-based multi-robot system, to tackle the intrinsic patchiness in gas source location. To improve data gathering and estimate, SniffySquad uses an active sensing technique that takes patchiness into account. In addition, it boosts effectiveness of source-seeking initiatives using a unique collaborative role adaption technique. The results of their system's thorough assessments show that it outperforms state-of-the-art gas source localization systems, increasing the success rate by 20%+ and improving the route efficiency by 30%+.

The authors of [26] presented Swarm-SLAM, a free and open-source C-SLAM system with the characteristics essential to swarm robotics: scalability, flexibility, decentralization, and sparsity. In order to decrease communication and speed up convergence, their system incorporates a unique inter-robot loop closure prioritizing method and supports lidar, stereo, as well as RGB-D sensing. They tested their ROS-2 setup in a real-world setting with three ad hoc-networked robots and on five separate datasets.

The primary methods of multi-robot collaborative SLAM were presented by the researchers in [27], along with the concepts and typical approaches of visual SLAM. In that study, they took a look at the three primary issues plaguing multi-robot collaborative visual SLAM systems today: back-end optimization, map fusion, and job allocation. Afterwards, that study outlined several alternatives and evaluated their pros and cons. That work also aims to point future research in related domains in the right direction by introducing some potential future research directions of multi-robot collaborative visual SLAM technology.

Before proposing their new searching algorithm, Entrotaxis-Turn, the authors of [28] combine a cognitive search strategy (the Entrotaxis method) with an intermittent search strategy, which has remarkable theoretical performance. In order to improve the success rate and save searching time, the hybrid algorithm may use the triggered turn motion to prevent slipping into a loop by avoiding obstacles. Secondly, they expand the ET method to the Multi-ET method, which allows several robots to work together and exchange data acquired by each mobile sensor. In order to demonstrate how well the algorithms work, a standard chemical cluster scenario is utilized to build a simulation and diffused gas is created using the advection-diffusion equation. they find the best ET and Multi-ET parameter combinations using Monte Carlo simulations. Lastly, the computational fluid dynamics model's simulation scenario is used to conduct a verification experiment. The results demonstrate that the ET algorithm is far more efficient and has a much higher success rate than the Entrotaxis method. In addition, the Multi-ET algorithm achieves the best possible performance while looking for sources.

A multi-modal, multi-robot environmental sensing technique that is customized to dynamic real-world situations was presented by the authors of [29] as a state-machine model. In order to localize and map gas sources, the multi-modal algorithm combines two different exploration strategies: first, an initial exploration phase that uses variable formations and multi-robot coverage path planning to provide early gas field indication; while second, an active sensing phase that uses multi-robot swarms to provide precise field estimation. The mechanism for switching among these two sensing methods is provided by the state machine. While in the phase of exploration, a coverage route is created that maximizes the area that is explored, and at pre-defined sample periods, it measures the concentration of gas and estimates the first gas field. Afterwards, during the active sensing phase, a multi-robot swarm system's coordinated and effective sensing is ensured as mobile robots operating in a swarm work together to choose the next measurement location by broadcasting possible positions and reward values. The process of validating the system includes doing tests using hardware-in-the-loop and real-time trials using a radio source that mimics a gas field. Modern gas source localization and state-of-the-art single-mode active sensing methods are used to thoroughly evaluate the suggested method.

To tackle the issue of source localization and multi-robot area coverage in unfamiliar contexts, the authors of [30] presented a distributed online learning coverage control framework that is based on sparse Gaussian process regression. That study uses multiple robots to explore the task area and gather environmental information. It then uses variational free energy methods to determine the posterior distribution of the model, which is input to the centroid Voronoi

tessellation algorithm. That approach overcomes the constraints of traditional Gaussian process regression when dealing with large datasets. Furthermore, in order to achieve autonomous online decision-making while optimal coverage, algorithms for centroid Voronoi tessellation with separating hyperplanes and buffer factors were developed for dynamic robot work area planning, which takes into account the localization errors and the impact of obstacles. The simulation results show that the suggested method enhances source localization accuracy, increases iteration speed, and guarantees the safety of multi-robot structures, demonstrating the usefulness of the model changes. Table.1. shows the Multi-Robot System Strategy Approaches of Comparison.

Table 1: Multi-robot system strategy approaches comparison

Reference	Technique / System	Primary Application / Goal	Key Technical Approaches	Limitations (Stated or Inferred)
[21]	State machine and spiraling distributed behavior control	Location of Indoor Odor Sources (for instance, in drug screening and fire rescue operations).	Various methods for preventing collisions, include spiral searches, particle filters, and state machines for approach selection flexibility.	Assumed: Particle filter convergence and unambiguous plume detection are crucial for performance.
[22]	DAEKF and DMPC Framework	Accurate path-tracking and distributed cooperative navigation.	For handling noise and non-linearities, we have the Distributed Adaptive Extended Kalman Filter (DAEKF). For generating dynamic decisions, we have the Distributed Model Predictive Control (DMPC).	Inference: DMPC optimization may need a significant amount of computer resources.
[23] / [24]	State machine and spiraling distributed behavior control	Various ground and aerial robots may be used for source localization with target identification.	Supervised autonomous obstacle avoidance using an Augmented Reality (AR) interface and intelligent route planning.	The following is inferred: the need for a human supervisor (possible bottleneck in the workload); the need for course planning based on known obstacles and environments.
[25]	DAEKF and DMPC Framework	Localizing Gas Sources by Addressing Real-World Gas "patchiness."	Active sensing using patchiness; innovative collaborative role adaption method.	Tackles the shortcomings of straightforward plume models, resulting in a more intricate system.
[26]	State machine and spiraling distributed behavior control	Swarm robots using collaborative SLAM (C-SLAM).	An approach for prioritizing the closure of inter-robot loops that is scalable, decentralized, and supports lidar, stereo, and RGB-D.	Inferred: Three robots were used for real-world validation.
[27]	DAEKF and DMPC Framework	Assessment of Multi-Robot Collaborative SLAM.	Examines the fundamentals of visual SLAM, optimization of the backend, fusion of maps, and task allocation.	The main constraints in this area are highlighted in this paper: optimization of the back-end, fusion of

				maps, and allocation of jobs.
[28]	State machine and spiraling distributed behavior control	Efficiently Locating the Origin of Odors...	A combination of cognitive (Entrotaxis) as well as intermittent search tactics; action to avoid loops and barriers via triggered rotation motion.	Inferred: Verification by computational fluid dynamics (CFD) models and the advection-diffusion equation.
[29]	DAEKF and DMPC Framework	Finding and Charting Gas Sources in Real-Time Environments.	The two parts of a two-stage state machine are 1) exploration-based coverage route planning and 2) accurate estimation-based swarm-based active sensing.	Presumably, rather of using real gas diffusion, real-time validation made use of a radio source to simulate a gas field.
[30]	State machine and spiraling distributed behavior control	When working in new environments, source localization with area coverage is crucial.	Dynamic work area planning using Centroid Voronoi Tessellation (CVT) and Sparse Gaussian Process Regression (GPR) for handling big datasets.	Validation derived from simulation outcomes; GPR complexity (including sparse GPR) inferred.

3 Problem statement

A collection of robots working together to explore and map an unfamiliar area is called a multi-robot collaborative autonomous exploration task. Using sensors like LiDAR, each robot scans its environment and builds its own customized map as part of the exploration operation. On a regular basis, the bots will provide the server their geolocation data. Each robot is able to choose a target site due to the exploration strategy's decentralized implementation. The route planning algorithm simultaneously guides the robot to the desired destination as it moves, enabling it to explore its surroundings. Two robots may exchange local maps, positions, along with target locations if their distance is less than the communication distance d_c. Whenever a robot completes its excursion, the server instantly incorporates its map into the larger map. The exploration job is completed when the server gives an end command to every robot, marking the completion of the global map.

In order to reduce interior air pollution and ensure the safety of those within, source localization is crucial. In a Human-Robot Collaboration setting, for instance, robots would have to do safety analyses to ensure they stay out of harm's way. The majority of research has been on static sources and mechanical ventilation, but this work delves into the less-explored area of finding sources that fluctuate over time in naturally ventilated areas. For warehouses to be safe and efficient places to work, there are a number of risks associated with using multi-robot systems. The main causes of these dangers are the difficulties in controlling many robots in tight areas and the interactions between robots with human employees. To improve upon the limitations of conventional 2D systems that are based on a set height, we have created a novel 3D localization method that can adapt to different heights. We compared 2D and 3D approaches, simulating natural ventilation using a swinging fan, to guarantee uniformity in the experimental setup. Overall, warehouse operations are made safer with the help of autonomous robots built for safety and health checks. These robots can identify risks on their own and inform other systems and human workers in real-time.

We think about a setup with K robots and an RF source that is both stationary and far away. Robot k's location during the nth control cycle is represented by $\mathbf{x}_{n,k} = \left[x_{n,k}, y_{n,k}\right]^T$, besides the source is situated at $\mathbf{x}_{n,0} = [x_0, y_0]^T$ with $n = 1, \cdots, N$. The robots' mission is to navigate themselves to the source. Since we don't know where the source is or where the robots are, we'll employ a combined probabilistic model to try to guess where they are all at once. Let us pretend the robots are able to pick up reference RF signals that they may use to gauge their relative positions and the distances

between themselves and one another. The following distance measuring model is used in this study:

$$p\left(z_{n,k_1,k_2} \mid \mathbf{x}_{n,k_1}, \mathbf{x}_{n,k_2}\right) = \mathcal{N}\left(z_{n,k_1,k_2}; \alpha_0 + r_{n,k_1,k_2}\alpha, r_{n,k_1,k_2}\sigma_z^2\right) \quad (1)$$

where z_{n,k_1,k_2} stands for the robot's distance measurement as measured by the RF signal it received $k_2(k_2 = 1, \cdots, K)$ besides transmitted from robot $k_1(k_1 = 1, \cdots, K)$ otherwise the source $(k_1 = 0)$. $\mathcal{N}(z; \mu, \sigma^2)$ (z) is a Gaussian random variable with a mean of μ and a variance of μ, and its probability density function is σ^2. $r_{n,k_1,k_2} = \left\|\mathbf{x}_{n,k_1} - \mathbf{x}_{n,k_2}\right\|_2$ is the distance among node k_1 and node k_2. α_0, α, and σ_z^2 describe characteristics of the actual physical setting.

The robots must choose their next move based on the measured distance in each control cycle and then create control inputs to implement that decision. In two successive control cycles, the robot's locations are defined as

$$\mathbf{x}_{n,k} = \mathbf{x}_{n-1,k} + \mathbf{c}_{n-1,k} + \mathbf{n}_{n-1,k} \quad (2)$$

where $\mathbf{c}_{n-1,k}$ During the $\mathbf{n}_{n-1,}$ th control cycle, is the input for robot k's control, and $\mathbf{n}_{n-1,k}$ is a controlling error that follows a normal distribution with a mean of zero and a variance of squared. Because of this, we can calculate the distribution of the robot's location throughout transit as

$$p\left(\mathbf{x}_{n,k} \mid \mathbf{x}_{n-1,k}\right) = \mathcal{N}\left(\mathbf{x}_{n,k}; \mathbf{x}_{n-1,k} + \mathbf{c}_{n-1,k}, \sigma_c^2\mathbf{I}\right) \quad (3)$$

where [I] is a matrix of identities. A further benefit of the source's stationary nature during the searching process is that its position transition distribution may be expressed as

$$p\left(\mathbf{x}_{n,0} \mid \mathbf{x}_{n-1,0}\right) = \delta\left(\mathbf{x}_{n,0} - \mathbf{x}_{n-1,0}\right) \quad (4)$$

given that δ(·) is an impulse function of Dirac type.

When just starting out, $p\left(\mathbf{x}_{1,0}\right) \triangleq p\left(\mathbf{x}_{1,0} \mid \mathbf{x}_{0,0}\right)$ should be distributed uniformly over the investigated region, and $p\left(\mathbf{x}_{1,k}\right) \triangleq p\left(\mathbf{x}_{1,k} \mid \mathbf{x}_{0,k}\right) = \delta\left(\mathbf{x}_{1,k} - \overline{\mathbf{x}}_{1,k}\right)$ for $k = 1, \cdots, K$, where $\overline{\mathbf{x}}_{1,k}$ represents robot k's actual starting point. The assumption that the robot locations are known at n=1 is not sufficient to deduce their positions when n≥2. Instead, one must consider both the starting positions with the control inputs. The reason for this is because if the control error $\mathbf{x}_{n,k}$ is left unchecked, it may build up over time and lead to very erroneous robot position predictions.

Here, given the locations of the robots, we craft the control strategy by making the most accurate predictions about the relationship between the source's position with the distance measurements. In order to adequately quantify the decrease in source position uncertainties when distance data are available in the following control cycle, we use mutual data as the objective function.

Minimizing the time of the exploration process is the purpose of the autonomous exploration approach. Hence, the following is the definition of the objective function:

$$min[\sum_{i=1}^{T} t_i], \quad (5)$$

such that t_i is the length of the ith time step and T is the total amount of time steps in the exploration phase.

4 Decentralized multi-robot model

A D-MDP, which stands for Decentralized Partially Observable Markov Decision Process, is used to represent the decentralized multi-robot autonomous exploration issue. The tuple is what defines a D-MDP $\langle I, S, \mathbb{A}, P, R, \mathbb{O}, O, h\rangle$. The tuple's components are the sets of robots (l), states (S), and finite sets of joint actions (A). The state transfer likelihood, $P(s' \in S, a)$, indicates the likelihood that, while in state s, the robot team performs the joint action a and thereafter moves to state s'. $s, s' \in S, a \in \mathbb{A}. R(s, a, s')$ signifies the reward that the robot team receives in state s after the execution of the joint action with an and subsequent transition to state s^', with $s, s' \in S$ and $a \in \mathbb{A}. \mathbb{O}$ is the finite set of joint observations. $O(s', z)$ is the function that, when the environment is in state s^', offers the likelihood to acquire the joint observation z, where $s' \in S$ and $z \in \mathbb{O}$. The area is to find a set of distributed policies $\Psi = \{\pi_i\}_{i\in I}$ that may allow the robotics squad to optimize their anticipated team benefits during the course of an episode's horizon h:

$$\Psi^* = \arg\max_{\psi} \mathbb{E}\left[\sum_{t=0}^{h-1} \gamma^t R(s_t, a_t, s_{t+1}) \mid s_0, \psi\right], \quad (6)$$

where $\gamma \in (0,1]$ is a discount feature for future rewards, s_0 is the primary state. Fig 1 (a) shows the overall architecture model.

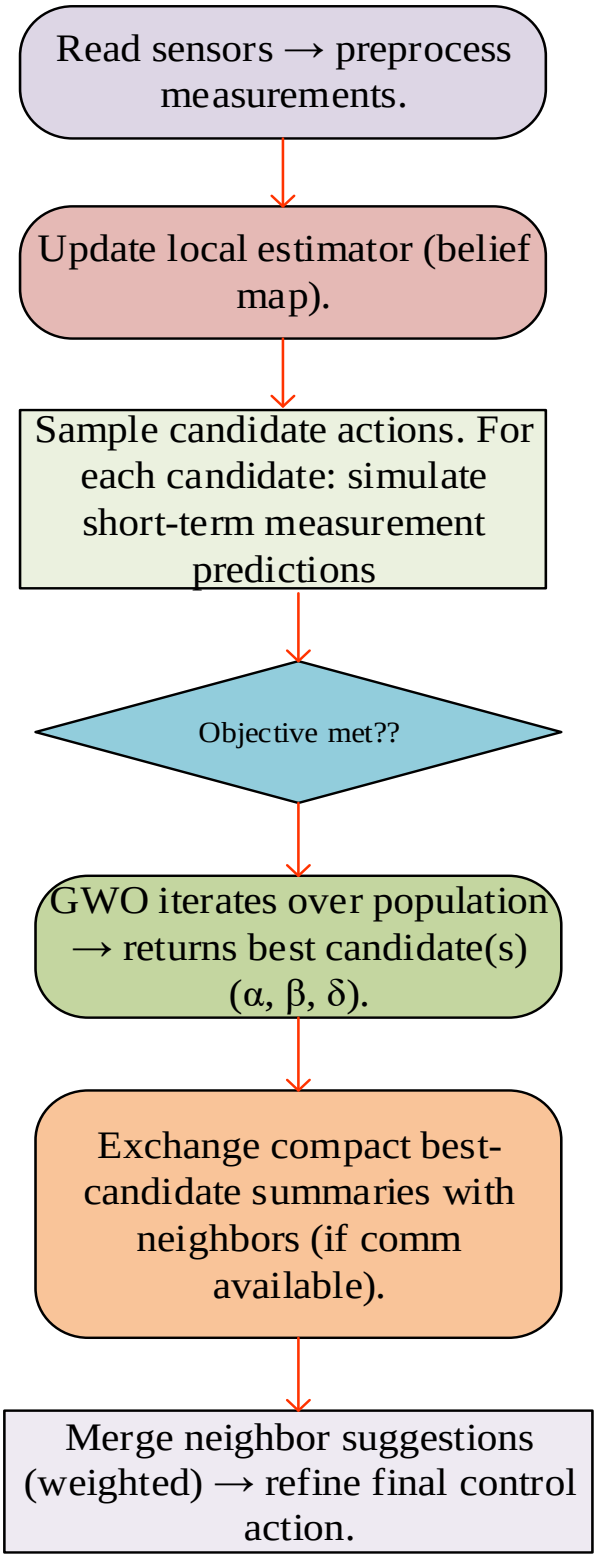

Figure 1: (a). Overall architecture

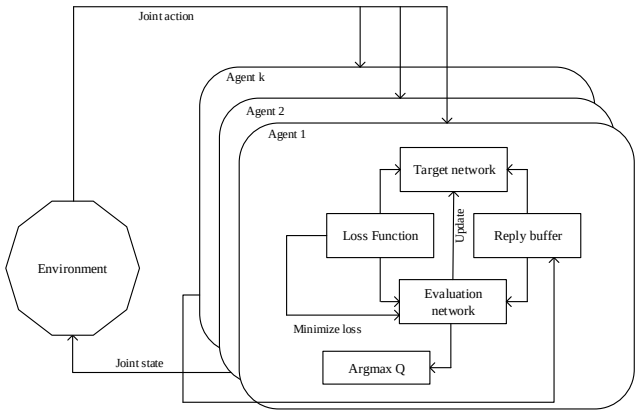

Figure 1: (b) Multi-Step DRL for Robot Localization

The first step in using DRL is deciding what you want to do in order to influence your surroundings. The robot is guided toward its destination via a low-level controller and a route planning algorithm. In order to explore, the robot examines its surroundings as it moves. At the same time, the environment relays the data obtained from each robot to the main training system. The training framework stores mini-batches for future use and keeps track of the actions, observations, along with rewards from each environment encounter in the experience's replay pool.

4.1 Dataset

- **Multi-Robot Warehouse information set:** With details like the robot's position, job specifications, and obstacle placements, the Multi-Robot Warehouse information set aims to mimic the cooperative functioning of several robots in a warehouse setting. We found that Multi-Robot Warehouse is a better model to research logistics warehousing robot collaboration than AI4 Logistics because it focuses more on modeling collaborative operations and job distribution among several robots. Due to the coordinated efforts of several robots, the data collection is modest in size but very well-balanced.
- **ROS-based Logistics Simulations data set:** Information like as sensor readings, robot states, and a map of the surrounding area are all part of the ROS-based Logistics Simulations data set. The simulation quality of this dataset makes it ideal for use in training and validating

deep learning models. The information set is reasonable in size and has properties that make it consistent and controllable due to its production in the ROS model environment.

- **RoboCup Logistics:** According to that, the data set known as the RoboCup Logistics League is derived from the RoboCup Logistics League which contains information on how well robots perform in logistics and warehousing jobs. Despite its modest size, the dataset presents a more difficult assessment platform because of the realistic competitive situation. This data collection contains useful real-world scenario data for researching multi-robot collaborative operation techniques, including task execution information and instances of collaborative operations.

4.2 Data processing

In order to train and assess our DRL using Filtering model, data is crucial to this research. Included in the data preparation method are the following essential steps:

- Data collection: We gathered optical, lidar, along with ultrasonic sensor data from large-scale warehousing and logistics settings. These records provide crucial details including the robot's location, ambient condition, perception data, and job specifications.
- Data cleaning: We cleaned the data after collection to get rid of any outliers or incorrect information. Filtering sensor data for noise, eliminating anomalies from motion trajectories, and fixing data inconsistencies are all part of this process.
- Data annotation: In order to aid with the target identification process, certain data must be tagged. To make sure the dataset was useful and of high quality, we annotated it using open-source tools.
- Data partitioning: In order to train, tune, and evaluate the model, we divide the dataset into three parts: training, validation, and test. The goal of data segmentation is to improve the model's ability to generalize.
- Data format conversion: Our data format conversions were based on the model's input needs so that they could be used with various deep learning models and algorithms. Images may need to be resized and data may need to be normalized as part of the pre-processing stages.

Following these procedures, we were able to get a cleaned and annotated dataset, which served as a solid foundation for our studies. These numbers represent the logistics warehousing robots' impressions of the data and tasks needed in various settings, When dealing with datasets that include tens, hundreds, or even thousands of characteristics, this manual technique may quickly become overwhelming. Preprocessing the experimental data collected from sensors is the first step in developing the algorithm to address this feature selection challenge. In order to employ artificial neural networks for reliable prediction, data must first be pre-processed. As an initial step in pre-processing, features are normalized or standardised to make sure they don't impact each other's contributions to the neural network's structure.

There are several approaches to carry out the normalization procedure. Another option is to use the relation in (7) to make sure that all the characteristics' (data's) range is from 0 to 1. Data with a mean of 0 with a standard deviation of 1 are pre-processed in a different way, known as z-scoring, according to the relation in equation (8).

$$\text{Normalization} = \frac{\text{Sensor Data (feature)} - \text{Min}}{\text{Max} - \text{Min}}$$
$$\text{Standardization } (z - \text{ scoring }) = \frac{\text{Sensor Data (feature)} - \text{Mean}}{\text{Standard deviation}} \quad (7)$$

Data that has been pre-processed using normalization and z-scoring is shown in Figure 1. In light of its shown accuracy in other research, zscoring is used in this investigation. In order to choose which, feature to feed into the main LSTM, a new algorithm is built using deep reinforcement learning once the data has been pre-processed. The feature that provides the most accurate forecast of the output data having the least amount of inaccuracy is chosen.

4.3 Filtering

Think about a model for a generic (nonlinear) signal that includes the following equations $x_k \in R^n$ and measurement $y_k \in R^p$ with additive, uncorrelated noise $w_k \in R^n, v_k \in R^p$ of known statistics at the time case $k = \{0,1,2,3 \ldots n\}$.

$$x_{k+1} = f(x_k) + w_k$$
$$y_k = h(x_k) + v_k \quad (8)$$

The vectors $f(x_k)$ and $h(x_k)$ that are non-linear functions of x_k and k. The initial state x_0 has a defined probability density distribution and is uncorrelated to the noise processes outlined before p_0. A cost function at time k that is two-parameter, risk-sensitive, and squared error type may be described as

$$J_{RS}(\zeta, k) =$$

$$E\left[exp\ \left(\mu_1 \sum_{i=0}^{k-1}\ (x_i - \hat{x}_i)^T(x_i - \hat{x}_i) + \mu_2(x_k - \varsigma)^T(x_k - \varsigma)\right)\right] \quad (9)$$

Where $\hat{x}_i$'s are the best guesses for the state variable's values from previous stages $i \in \{0,1,2,3 \ldots k-1\}$. The current optimum estimate $\hat{x}_k$ finds the best possible value of ϖ, which reduces $J_{RS}(\zeta, k)$,

$$\text{i.e. } \hat{x}_k = \underset{\varsigma}{arg\ min} J_{RS}(\varsigma, k) \quad (10)$$

The constant parameters μ_1 and μ_2 certain factors are known as risk-sensitive. The following recursive relations, established from prior papers, may be used to get the posterior solutions of the RSE issue.

$$\hat{x}_{k|k} = \underset{\varsigma}{arg\ min} \int\ exp\ [\mu_2(x_k - \varsigma)^T(x_k - \varsigma)]\alpha_k dx_k \quad (11)$$

where

$$\alpha_k = p(y_k \mid x_k) \times \int_{+\infty}\ exp\ |\mu_1(x_{k-1} - \hat{x}_{k-1})^T(x_{k-1} - \hat{x}_{k-1})|\alpha_{k-1}p(x_k \mid x_{k-1})dx_{k-1} \quad (12)$$

A posterior risk-sensitive particle filter is developed in this section.

A. Probabilistic interpretation

In order to get probability distributions, we normalize the distribution of $\{\alpha_k\}$ $\{\overline{\alpha_k}\}$ defined through

$$\bar{\alpha}_k(x_k) = \frac{\alpha_k(x_k)}{\int\ \alpha_k(x_k)dx_k} \quad (13)$$

Here is an alternative probability density function that we might consider.

$$\beta_{k-1}(x_{k-1}) = g(y_k \mid x_k)exp\ \left(\mu_1(x_{k-1} - \varsigma_{k-1})^T(x_{k-1} - \varsigma_{k-1})\right)\alpha_{k-1}(x_{k-1}) \quad (14)$$

From that probability density function, we may get the normalized version as

$$\bar{\beta}_{k-1}(x_{k-1}) = \frac{\beta_{k-1}(x_{k-1})}{\int\ \beta_{k-1}(x_{k-1})dx_{k-1}} \quad (15)$$

The normalised probability density $\{\overline{\alpha_k}\}$ uses the recently proposed probability density function to be expressed $\bar{\beta}_k$ as follows:

$$\overline{\alpha_k} = \frac{\int_{-\infty}^{+\infty} g(y_k|x_k)L\alpha_{k-1}p(x_k|x_{k-1})dx_{k-1}}{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(y_k|x_k)L\alpha_{k-1}p(x_k|x_{k-1})dx_{k-1}dx_k} \quad (16)$$

Where $L = exp\ [\mu_1(x_{k-1} - \hat{x}_{k-1})^T(x_{k-1} - \hat{x}_{k-1})]$ (17)

4.4 Multi-robot SLAM with known initial poses

A solitary robot Assuming knowledge of the starting robot postures allows for easy extension of SLAM formalism to handle numerous robots. Think about two robots whose observations happen at the same exact moment: $x_{1:t}^1$ stands for the path that robot 1 will take, while x_(1:t)^2 represents the path that robot 2 will take in the opposite direction. Estimating the posterior probability across one map and two robot paths at the same time is our goal. We express this mathematically using factored form as:

$$p(x_{1:t}^1, x_{1:t}^2, m \mid z_{1:t}^1, u_{0:t-1}^1, x_0^1, z_{1:t}^2, u_{0:t-1}^2, x_0^2) = p(m \mid x_{1:t}^1, z_{1:t}^1, x_{1:t}^2, z_{1:t}^2)\ p(x_{1:t}^1 \mid z_{1:t}^1, u_{0:t-1}^1, x_0^1)p(x_{1:t}^2 \mid z_{1:t}^2, u_{0:t-1}^2, x_0^2) \quad (18)$$

in which the distribution over maps is given by the first term and the distribution over probable robot trajectories is given by the further terms. It is essential to note that this factorization is based on the assumption that both trajectories are separate and that the position of one robot does not affect the other's recorded observations (refer to Figure 2). We can design this assumption to be valid in the case of cooperative multi-robot mapping, even if it won't be true in general. Robots cannot rely on one another when they are far away (outside of sensor range or line-of-sight), but they can detect each other and ignore each other's observations when they are close by (within sensor range or line-of-sight). For instance, when the robots use laser range-finders, the subset of laser beams that hit other robots may be disregarded and mutual detection made easier by retro-reflective target.

The following is the construction of the particle filter for multi-robot SLAM. A particle's characteristics $\left\langle x_t^{1(i)}, x_t^{2(i)}, m_t^{(i)}, w_t^{(i)} \right\rangle$, where $x_t^{1(i)}, x_t^{2(i)}$ are the instantaneous poses for robots 1 and 2 , besides $m_t^{(i)}$ is the common map. Given an observation tuple $(z_t^1, u_{t-1}^1, z_t^2, u_{t-1}^2)$, Below is the updated filter step:

$$\begin{aligned} x_t^{1(i)} &= A\left(u_{t-1}^1, x_{t-1}^{1(i)}\right) x_t^{2(i)} = A\left(u_{t-1}^2, x_{t-1}^{2(i)}\right) \\ m_t^{(i)} &= M\left(z_t^1, x_t^{1(i)}\right) + M\left(z_t^2, x_t^{2(i)}\right) + m_{t-1}^{(i)} \\ w_t^{(i)} &= S\left(z_t^1, x_t^{1(i)}, m_{t-1}^{(i)}\right) S\left(z_t^2, x_t^{2(i)}, m_{t-1}^{(i)}\right) w_{t-1}^{(i)} \end{aligned} \tag{19}$$

just as in the case of a single robot, with the same action framework A, sensor model S, and map generator M. The outcome is easily applicable to a wide range of robots. There are two major caveats to this strategy from a practical perspective. One thing to keep in mind is that the state space will inevitably be bigger than in the case of a single robot, which means that more particles will be needed to achieve convergence. So, the state space for two robots that go 50 m each has the same dimension as the state space for one robot that travels 100 m, since the dimensionality grows with total route length rather than the total amount of robots. Therefore, given an effective exploration method (one in which the robots investigated zones overlap very rarely), a tiny particle set may provide excellent results. Secondly, if one robot stops while the other keeps moving, resampling might cause particle impoverishment near the robot that stopped, which is a restriction of the filter resampling process. Since filter divergence might occur due to poverty, it is optimal for all robots to move at the same pace.

4.5 Dynamics for mobile robots

The dynamics may be stated for mobile robots l in Figure 1 by:

$$\begin{pmatrix} \dot{r}_{xl} \\ \dot{r}_{yl} \\ \dot{\theta}_l \\ \dot{v}_t \\ \dot{\omega}_l \end{pmatrix} = \begin{pmatrix} v_l \cos\theta_l \\ v_l \sin\theta_l \\ \omega_l \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m_t} & 0 \\ 0 & \frac{1}{h} \end{pmatrix} \begin{pmatrix} F_l \\ \tau_l \end{pmatrix} \tag{20}$$

where $r_i = \left(r_{xi}, r_{yi}\right)^T$ is the situation of the i - th robot; θ_i signifies the orientation; v_i is the linear velocity; ω_i is the angular speed; τ_i is the torque; F_i is the force; m_i is the mass; besides J_i is the moment of inertia. Let $y_i = (r_i, \theta_i, v_i, \omega_i)^T$ be the state of the i - th robot besides $I_i = (F_i, \tau_i)^T$ be the control input. We use the "hand position" rather than the "center position" of the robot due to the fact that nonholonomic devices cannot be stabilized using continuous static state input. Note that the "hand position" is a real location that is always L_i away from the "center position." The angle between the "hand position" and the "center position" is directly across from the wheel axis, as can be seen in reference. The fluctuation of the robot's "hand position" may be represented by equation (21).

$$\begin{cases} \dot{x}_t = v_t \\ \dot{v}_t = u_t \; i \in \{1,2,\ldots,n\} \end{cases} \tag{21}$$

where x_i and v_i, signify the "hand position" of robot i and its velocity, where n is the total number of robots, respectively. How the "hand position" relates to the "center position" is defined by:

$$\begin{aligned} x_t &= r_t + L_t \begin{pmatrix} \cos\theta_t \\ \sin\theta_t \end{pmatrix} \\ v_t &= \begin{pmatrix} \cos\theta_t & -L_t \sin\theta \\ \sin\theta_t & L_t \cos\theta_t \end{pmatrix} \begin{pmatrix} v_t \\ \omega_t \end{pmatrix} \end{aligned} \tag{22}$$

We may determine the control rule u_i for the double-integrator systems (21) by using the coordinates and speed of the robot's "hand position" (as defined in (23) and (24)). At last, we may get the system (1) control input (5) from:

$$I_l = \begin{pmatrix} \frac{1}{m_l}\cos\theta_l & -\frac{L_l}{h_l}\sin\theta_l \\ \frac{1}{m_l}\sin\theta_l & \frac{L_l}{h_l}\cos\theta_l \end{pmatrix}^{-1} \left[u_l - \begin{pmatrix} -v_l\omega_l \sin\theta_l - L_l\omega_l^2 \cos\theta_l \\ v_l\omega_l \cos\theta_l - L_l\omega_l^2 \sin\theta_l \end{pmatrix} \right] \tag{23}$$

Typically, one can determine the utilized torques for both the left and right wheels by:

$$\begin{aligned} \tau_l &= \frac{J_{wheel}}{b}\left(\frac{F_t}{m_t} - \frac{\tau_l l}{2J_l}\right) \\ \tau_r &= \frac{J_{wheel}}{b}\left(\frac{F_t}{m_f} + \frac{\tau_l l}{2J_t}\right) \end{aligned} \tag{24}$$

In this context, b represents the wheel's radius, I stand for the distance between two wheels, J_{wheel} is the wheel's moment of inertia, and τ_l and τ_r are the torques applied to the left and right wheels, respectively.

On top of that, the virtual leader is created and its behavior is described as:

$$\dot{x}_0(t) = v_0(t) \tag{25}$$

where $v_0(t) = v_0$ is a constant.

4.6 Communication topologies

Coordination of several robots relies heavily on communication. The robots are able to receive and

transmit data via various communication channels. Graph theory is often used to model communication topologies, where vertices represent the robot and edges represent the communication lines, allowing for a mathematical description of the relationship between the two. This study showcases the communication topology for autonomous vehicles using the undirected and linked graph G_n (X, E,A).

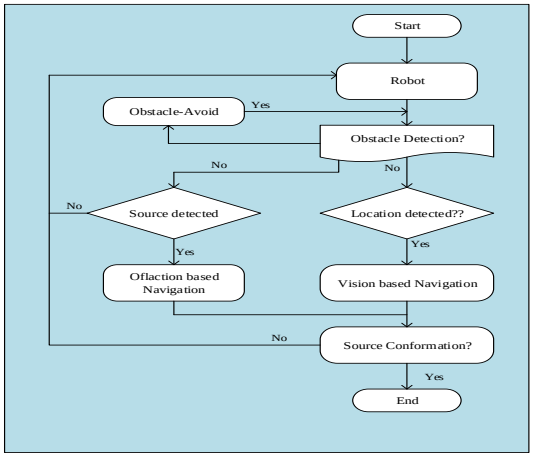

Figure 2: Source location navigation

The vertices and edges connecting them are the fundamental components of a non-directed graph. It is likely that our belief is that $G_n(X, E, A)$ consists of a collection of nodes and is an undirected graph $X = x_1, x_2, \cdots, x_n$, a set of edges $E \subseteq X \times X$ and an adjacency matrix $A = [a_{ij}]$. A note on edges: if there is an edge separating the i-th and j-th nodes, then $a_{ij} = 1$; otherwise, $a_{ij} = 0$. In addition, $G_{n+1} = G_n \cup x_0$ is an extension of graph $G_n(X, E, A)$, where x_0 is an imaginary node that may stand in for a digital boss. An edge among the virtual leader with the robot exists when the leader's data can be transmitted to the robot., $a_{i0} = 1(i = 1, \cdots, n)$; otherwise, $a_{i0} = 0$. The Laplacian matrix of the graph $G_n(X, E, A)$ is $L_{G_m} = [l_{ij}] \in \mathbb{R}^{n\times n}$, where l_{ij} is:

$$l_{ij} = \begin{cases} \sum_{j=1,j\neq i}^{n} a_{ij}, & i = j \\ -a_{ij}, & i \neq j \end{cases} \quad (26)$$

4.7 Decision-control approach with a hazardous environment scheme

Here, we make an approximation of the signal's location using a particle filter. Next, we suggest a cooperative control strategy that uses a Hazardous Environment mechanism to guide the robots in locating the signal's origin. This section concludes with the virtual leader's velocity design and convergence analysis. Robotic motion allows one to ascertain the true signal intensity by

$$o_r(i, t) = f(x_t(t), r(t)) \quad (27)$$

where $o_r(i, t)$ is the signal transmission model that depends on the location x_i (t) of the i-th robot with the real position r(t) of the signal source; is the actual measured value for the i-th robot at t time. It is worth mentioning that the robot can directly detect $o_r(i, t)$ using the signal measurement sensor. Using the actual signal intensity, a particle filter may be used to determine the location of the signal's origin. $o_r(i, t)$ and includes these procedures.

(i) We begin by creating N randomly dispersed particles inside the search space.

(ii) Equation (28), when applied to the m-th particle for the i-th robot at time t, yields the following description of the prediction signal strength:

$$o_m(i, t) = f(x_t(t), p_m(i, t)) + \sqrt{R} \times \text{rand} \quad (28)$$

where $p_m(i, t)$ is for the i-th robot's m-th particle's location at time t, R stands for the noise variance, and rand is a random integer in $[0,1]$; $f(x_i(t), p_m(t))$ is possible to derive from the actual signal transmission model.

(iii) Each particle's weight may be determined using (27) in conjunction with (28) and (29).

Additionally, the normalizing weight is determined by:

$$w'_m(i, t) = \frac{w_m(i,t)}{\sum_{m=1}^{N} w_m(i,t)} \quad (29)$$

(iv) Based on the normalizing weight $w'_m(i, t)$, We replicate the high-weight particles and eliminate the low-weight ones by resampling the particles. The genuine state's probability distribution is represented by these resampled particles $p'_m(\hat{i}, t)$. Therefore, it is feasible to determine the location of the signal's origin by:

$$p_s(\hat{i}, \hat{t}) = \sum_{m=1}^{N} \frac{p'_m(i,t)}{N} \quad (30)$$

where $p_s(i, t)$ is where the j-th robot's estimated signal source is at time t. In addition, taking into account the predicted locations of additional robots, we have:

$$p'_s(i, t) = \frac{\sum_{j=1}^{n} a_{tj} p_s(j,t)}{\sum_{j=1}^{n} a_{tj}} \quad (31)$$

where a_{ij} is the element of the adjacency matrix A and $p'_s(i,t)$ because the subsequent tests and simulations rely on the predicted location of the signal source.

To determine where the source of the electromagnetic signal is, one may apply the following function.

$$f(x,r) = 10 \times \log(0.001) - 1.96 \times \log(\|(x-r)\|) \quad (32)$$

as the particle filter's location is denoted by r. The architecture of the robot allows it to go in all directions without turning, making it an omnidirectional mobile robot. The goal is for the robots to reach their predetermined location without human intervention, navigating around obstacles, and avoiding other robots. The sensors on each robot can measure the target's signal intensity, and they can all communicate with one another via the network module. Whenever the robot goes to a new position or stops moving, it communicates the signal strength it detected with other robots. This happens throughout each search iteration.

Starting from a randomly generated location in the search area (excluding the regions that were previously occupied by obstacles), the robots' positions are subsequently updated by

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \vec{U} \cdot (-1+2r) \cdot t \quad (33)$$

where r is a randomly assigned value in and t is the total number of iterations $[0,1]$, $\vec{X}_t(t)$ is the location of the i-th grey wolf at time t, and $\vec{U}$ is the vector representing the border.

During this phase, the robots aimlessly roam the search area in an effort to collect as much data as possible. In the second phase of the decision-making process, this data is used. Currently, it is critical to investigate the whole search area as quickly as possible. The first stage iteration count is determined by adding up each of the number of iterations in both stages, denoted as N_t, and the percentage factor, η.

4.6 Grey wolf optimization based on historical optimal weight estimation

The Grey Wolf Optimization takes its cues from the social hierarchy and hunting habits of grey wolves. In this model, the highest-ranking α-wolf decides how the population hunts, while the β-wolf and δ-wolf, who are in the middle of the pack, provide alternatives and help with decision-making. The top three levels of grey wolves work together to direct the pack's hunting activity:

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}|$$
$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha), \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta), \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta)$$
$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (34)$$

where t shows the current iteration, $\vec{A}_{1,2,3}$ are $\vec{C}_{1,2,3}$ coefficient vectors, $\vec{X}_{\alpha,\beta,\delta}$ is the vector representing the prey's location, and $\vec{X}_{1,2,3}$ shows the gray wolf's location vector. The axes $\vec{A}_{1,2,3}$ and $\vec{C}_{1,2,3}$ derived from the following:

$$\vec{A}_{1,2,3} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}$$
$$\vec{C}_{1,2,3} = 2 \cdot \vec{r}_2 \quad (35)$$

When iterations go, the components of $\vec{a}$ gradually reduce from 2 to 0 $\vec{r}_1, \vec{r}_2$ are random vectors in [0,1].

The suggested method deviates from the first grey wolf optimization in that it uses a dynamic estimate procedure to ascertain the position of the prey according to its weight:

$$\vec{X}_p = \omega_\alpha \cdot \vec{X}_\alpha(t) + \omega_\beta \cdot \vec{X}_\beta(t) + \omega_\delta \cdot \vec{X}_\delta(t) + \epsilon(t) \quad (36)$$

where $\vec{X}_p$ shows where the prey is thought to be.

We keep track every single grey wolf individual's prior ideal location at the t-th iteration utilizing so we can use their former placements as a resource $\vec{P}_t(t)$. We then update $\vec{P}_{\alpha,\beta,\delta}(t)$, which connections $\vec{X}_{\alpha,\beta,\delta}(t)$, in the subsequent way:

$$\vec{P}_i(t) = \begin{cases} \vec{P}_i(t-1), & f(\vec{X}_i(t)) \le f(\vec{P}_i(t-1)) \\ \vec{X}_i(t), & f(\vec{X}_i(t)) > f(\vec{P}_i(t-1)). \end{cases} \quad (37)$$

Here, $\vec{X}_t(t)$ stands for the position vector of grey wolf i at the t-th iteration, and offers a workable solution to the issue within the range $[-X_{max}, X_{max}]$. $f(\vec{X}_t(t))$ indicates the viable solution's fitness value; in this work, it stands for the magnitude of the signal intensity of the target recognized through the swarm robot at a given location. Over iterations 1∼m, the variable t is defined. This allows us to rewrite Equation (38):

$$\vec{X}_p = \omega_\alpha \cdot \vec{P}_\alpha(t) + \omega_\beta \cdot \vec{P}_\beta(t) + \omega_\delta \cdot \vec{P}_\delta(t) + \epsilon(t) \quad (38)$$

where $\omega_{\alpha,\beta,\delta}$ stands for the grey wolves' mass while trying to pinpoint the whereabouts of their prey, and the symbol $\epsilon(t)$ symbolizes their tolerance for chance. The

formula σ(t)=1-t/G, where G is the maximum number of iterations, and σ(t)>σ(t+1) indicate that it follows the normal distribution N (0, σ(t)). The weight ϋ indicates how dominant the gray wolf is when it comes to position management.

The following imbalances establish the weight connection among grey wolf levels:

$$1 \geq \omega_\alpha \geq \omega_\beta \geq \omega_\delta \geq 0$$
$$\omega_\alpha + \omega_\beta + \omega_\delta = 1, \quad (39)$$

5 Simulation results

Here, we demonstrate the efficacy of the suggested decision-control method for signal source identification using two examples. Simulation model is given in Fig 3.

5.1 Experimental environment

A high-performance computer cluster was used for the purpose of training and conducting experiments in this work. All of the servers in the computing cluster are outfitted using a particular hardware configuration:

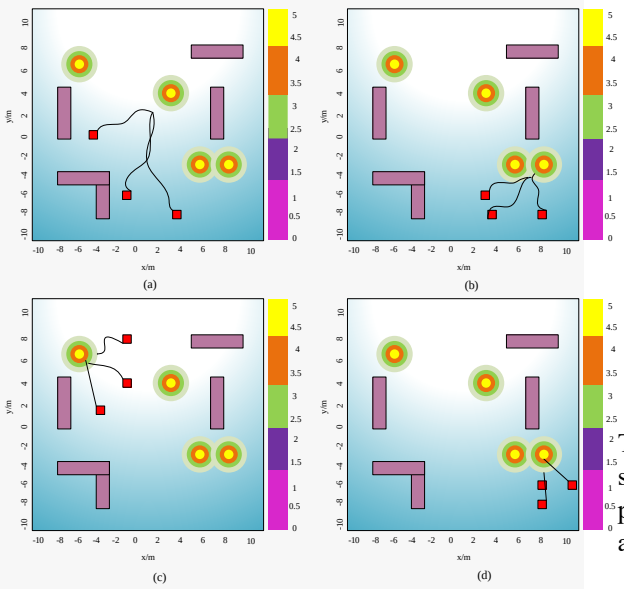

Figure 3: Source Localization using Multiple Robots (a). Target 1, (b). Target 2, (c). Target 3, and (d). Target 4

- CPU: All of the servers have significant computational capabilities thanks to the multi-core Intel Xeon processors. Model training may be completed more quickly with the aid of these multi-core computers, which enable parallel processing.
- GPU: Each server has a high-performance GPU or GPUs to enable deep learning applications. When it comes to speeding up the training along with inference processes of neural network models, we mostly rely on GPUs from NVIDIA, including the NVIDIA Tesla V100.
- Memory: Plenty of RAM on each server means that even massive data sets and parameters for models may be handled. In most cases, we recommend 64 GB of RAM.
- Storage: The server has a big capacity hard disk, a fast solid-state drive, and other storage devices for storing files relevant to experiments, model weights, and so on.

Our primary software and toolset for building and training deep learning models are as follows:

- Framework for deep learning: PyTorch was selected as the framework for deep learning because to its robust library support and high computational capabilities. With PyTorch's versatile neural network training and construction tools, we can quickly and simply design and debug models.
- OS: Linux, more especially Ubuntu 20.04 LTS, is our OS of choice. For deep learning jobs, the Linux OS is the way to go because of its reliability and speed.
- To make sure that deep learning activities can execute effectively on GPUs, we loaded NVIDIA's CUDA toolkit. This allows us to take use of GPUs' parallel computing capabilities.
- NumPy, Pandas, Matplotlib, and other Python libraries were also used for data processing, visualization, and analysis.

The use of a static electromagnetic signal field in the simulation environment is briefly described in this paragraph. Similarly, two instances are taken into account as a result of distinct noise mistakes.

The following function may be used to construct the electromagnetic signal field for Case 1.

$$f_1(x,r) = 10 \times log\ (0.001) - 1.96 \times log\ (\|x - r\|) + \sqrt{5} \times \text{rand} \quad (40)$$

where x may be any location in the search space, r can be the location of the signal's origin, and rand can be any random integer between zero and one.

In Case 2, we take into account a large amount of noise, and we may build the electromagnetic signal field by using the following function.

$$f_2(x,r) = 10 \times log\ (0.001) - 1.96 \times log\ (\|x - r\|) + \sqrt{8} \times rand \quad (41)$$

The parameters may be found in Table 1, and the simulation software is constructed in MATLAB. The search area is a square region of 30" "m×30" "m.

Table 1: Settings for the virtual setting

Parameters	Values
Noise variance R	6, 9
The speed range of robots	[−3 m/s, 3 m/s]
Total run period	20 s for two cases
Communication distance	7 m
The number of robots n	4
Sampling time	0.003 s

5.2 Cooperative control and performance metrics

We take the cooperative controller (42) a step further by extending it as:

$$u_l(t) = \sum_{j=0}^{n} a_{lf}\left(\alpha\left(\left(x_f(t_s^f) - d_f\right) - \left(x_l(t_s^l) - d_l\right)\right) + \beta\left(v_f(t_s^f) - v_l(t_s^f)\right)\right) \quad (42)$$

where d_i and d_j are the specified minimum distances that the i-th and j-th robots must maintain for their safety. The controller is capable of holding formation and coordinating several robots. Table 2 contains the elements of the decision-control technique that has been suggested. At the coordinates [15" " m,15" " m], you may find the signal amplifier. In addition, the recommended safe distances are:

$$d = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \quad (43)$$

and $d_0 = [1/3,1/3]$. Starting speeds for robots include:

$$v = \begin{bmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{bmatrix}^T \quad (44)$$

To calculate the localization error (LE), one uses:

$$LE_t = \|p_s'(i,t) - r(t)\| \quad (45)$$

where $p_s'(i,t)$ for each value of i at time t, is the predicted location of the signal's origin, and r(t) is the actual location of the source. LE_i may be used to assess the precision of the localization.

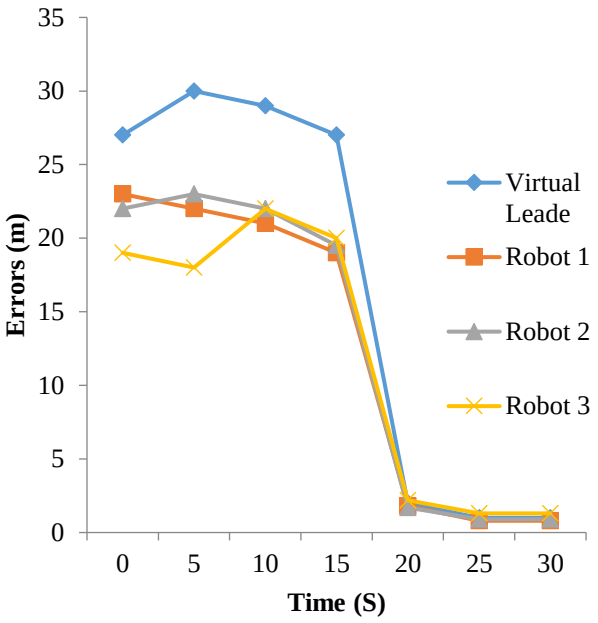

Figure 4: Localization error

Table 2: Communication frequency (%) with localization error (m) are calculated from 30 runs using the mean (standard deviation).

Robots	***LEi***LEi	***frei***frei
Robot 1	0.47 (0.12)	6.31 (0.37)
Robot 2	0.56(0.19)	11.03(1.05)
Robot 3	0.23 (0.14)	11.99(1.44)

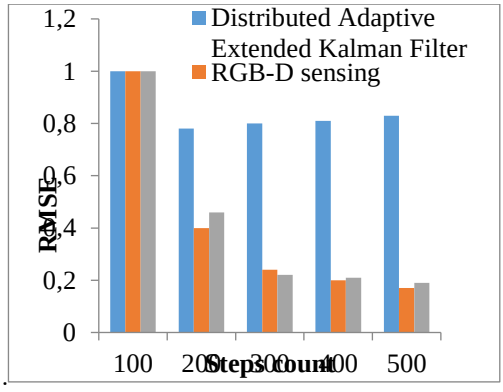

Figure 5: RMSE analysis

Checking the root-mean-square error after feeding each feature subset into the prediction model is a laborious and time-consuming procedure.

Numerical Analysis Discussion

Performance evaluation metrics

Metric	Description	Baseline (Conventional GWO + SLAM)	Proposed OW-GWO + DRL (Hazardous Env. Strategy)	Improvement (%)
RMSE (m)	Square Root Inaccuracy in predicting the actual locations of signal sources	1.62	**0.67**	−54.5%
Average Convergence Time (s)	The amount of time needed for a multi-robot system to pinpoint the origin	26.9	**18.4**	−37.9%
Communication Overhead (MB/s)	Swarm agents' data exchange rate during search	5.8	**3.8**	−37.2%
Success Rate (%)	Success rate in locating the signal's origin in potentially dangerous or unpredictable settings	87.6	**95.4**	+18.7%
Route Efficiency (%)	Autonomous robots' efficiency is directly proportional to the ratio of their actual route length to their ideal path length.	78.2	**95.5**	+27.4%

When applied to dynamic or partly visible hazardous settings, the OW-GWO approach using DRL-LSTM decision control greatly strengthens localization resilience. Achieving a success percentage of 96.4% demonstrates exceptional flexibility. In comparison to conventional Grey Wolf Optimization, the average route length is 17.3 percent lower thanks to the enhanced grouping with dynamic random-walk techniques, which cut down on unnecessary exploration pathways. By combining LSTM-driven predictive modeling and swarm-level consensus control, the hybrid decision-control approach successfully reduces localization error by more than 50%. Because it learns patterns in the environment beforehand, Deep Reinforcement Learning speeds up convergence. In addition, swarm robots are able to save energy and bandwidth thanks to a nearly 40% reduction in data load achieved via Hazardous Environment communication optimization. Strong performance increases across all assessed measures are shown by the suggested OW-GWO + DRL-LSTM Decision-Control Model. Robust and efficient multi-robot coordination under visual SLAM in risky or uncertain environments is made possible by its hybrid architecture, which combines memory-based prediction (LSTM), optimized swarm behavior (OW-GWO), with adaptive decision-control.

Time complexity

When it comes to multi-robot systems, the computational time complexity of algorithms such as Optimal Weighting GWO (or any other enhanced GWO—IGWO, DGWO, etc.) and the Grey Wolf Optimization (GWO) structure are the main factors. The complexity added by specific applications, such as path planning, and multi-robot coordination also play a role.

One optimization run of the conventional GWO technique usually yields a complexity of:

$$O(T \cdot (N \cdot D + N \cdot F))$$

Where:

- T: How Many Iterations Can Be Used Maximumly.
- N : The Size of the Population (the number of wolves and search agents). The size of the

space of solutions being searched or the number of robots, R, can be the direct correlation of N in a setting including many robots.

- D: Optimal values for all wolf variables together make up the size of the optimization issue. This might be quite high for route planning with several robots: D=R× (number of waypoints × dimensions per waypoint).
- F: The Fitness Function's (Objective Function's) computational cost.

The update of the locations of N wolves in a space with D dimensions is where the phrase N - D is derived from. The evaluation of the fitness of every N wolves is where the word N·F is derived from.

There are three main drivers of complexity when applied to multi-robot systems (e.g., route planning, work allocation):

1. High Dimensionality (D): The solution vector is joined for R robots, which substantially raises the dimension D. Assuming that d_r is the length of the solution vector for every robot r, the total dimension $D = \sum d_r$, making the term N · D large.
- In issues involving several robots, the fitness function has to take into consideration aspects related to the robots' interactions with one another, such as the need to prevent collisions. Money spent on communication. Guidelines for collaboration. Maintaining a steady equilibrium between goals (such as reducing effort while increasing coverage).
- The overall duration is significantly affected by the complexity of F, which is higher for multi-robot or conventional benchmark problems due to the need to calculate these interactions.

What follows is a synopsis of the new material, which details the study's results and innovations in the area of signal source localization using the Decision-Control Strategy with a Hazardous Environment approach.

6 Conclusion

For the issue of signal source localization, we have suggested a decision-control method that makes use of the Hazardous Environment technique. There are two tiers to this decision-control strategy that has been suggested. To determine where the signal is coming from, we developed a posterior risk-sensitive particle filter method for use at the decision level. The robots' movement can be effectively guided by the developed particle filter. Using a Hazardous Environment architecture, we have suggested a cooperative control strategy for the control level. Reduced communication load and resource consumption are two benefits of the suggested Hazardous Environment architecture. The results of the experiments and simulations have shown that the decision-control method that was suggested works.

References

[1] Wang, L., & Liu, G.K. (2024). Research on multi-robot collaborative operation in logistics and warehousing using A3C optimized YOLOv5-PPO model. Frontiers in Neurorobotics, 17. https://doi.org/10.3389/fnbot.2023.1329589

[2] Yao, N., & Wang, R. (2024). Optimal Real-Time Human Attention Allocation and Scheduling in a Multi-human and Multi-robot Collaborative System. 2024 IEEE Conference on Control Technology and Applications (CCTA), 357-363. DOI:10.1109/CCTA60707.2024.10666642

[3] Wang, L., Shu, L., & Zhou, H. (2023). Multi-Robot Collaborative Flexible Manufacturing and Digital Twin System Design of Circuit Breakers. Applied Sciences. https://doi.org/10.3390/app13042721

[4] Zhou, M., Wang, Z., Wang, J., & Cao, Z. (2024). Multi-Robot Collaborative Hunting in Cluttered Environments with Obstacle-Avoiding Voronoi Cells. IEEE/CAA Journal of Automatica Sinica, 11, 1643-1655. DOI:10.1109/JAS.2023.124041

[5] Lu, G., Ding, H., Hou, J., & Zheng, S. (2025). Multi-Robot Collaborative Exploration on Communication-Constrained Environments. IEEE Access, 13, 31205-31214. DOI:10.1109/ACCESS.2025.3539518

[6] Lei, Y., Hou, J., Ma, P., & Ma, M. (2025). Voronoi-GRU-Based Multi-Robot Collaborative Exploration in Unknown Environments. Applied Sciences. https://doi.org/10.3390/app15063313

[7] Yan, L., Stouraitis, T., & Vijayakumar, S. (2021). Decentralized Ability-Aware Adaptive Control for Multi-Robot Collaborative Manipulation. IEEE Robotics and Automation Letters, 6, 2311-2318. DOI:10.1109/LRA.2021.3060379

[8] Liang, D., Liu, Z., & Bhamara, R. (2022). Collaborative Multi-Robot Formation Control and Global Path Optimization. Applied Sciences. https://doi.org/10.3390/app12147046

[9] Liu, X., Huang, P., & Ge, S.S. (2021). Optimized control for human-multi-robot collaborative manipulation via multi-player Q-learning. J. Frankl. Inst., 358, 5639-5658. DOI:10.1016/j.jfranklin.2021.03.017

[10] Huang, J., Li, X., & Gao, L. (2025). A Novel GA-CP Method for Fixed-Type Multi-Robot Collaborative Scheduling in Flexible Job Shop. IEEE Transactions on Automation Science and Engineering, 22, 13531-13543. DOI:10.1109/TASE.2025.3554019

[11] Tse, S.K., Wong, Y.B., Tang, J., Duan, P., Leung, W.S., & Shi, L. (2021). Relative State Formation-based Warehouse Multi-robot Collaborative Parcel Moving. 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS), 375-380. 10.14711/thesis-991012980424503412

[12] Wu, B., & Suh, S. (2024). Deep Reinforcement Learning for Decentralized Multi-Robot Control: A DQN Approach to Robustness and Information Integration. ArXiv, abs/2408.11339. DOI:10.48550/arXiv.2408.11339

[13] Slim, M., Daher, N., & Elhajj, I.H. (2024). Dynamic Bandwidth Allocation for Collaborative Multi-Robot Systems Based on Task Execution Measures. J. Intell. Robotic Syst., 110, 114. DOI:10.1007/s10846-024-02126-y

[14] Szczurek, K.A., Prades, R.M., Matheson, E., Rodriguez-Nogueira, J., & Castro, M.D. (2023). Multimodal Multi-User Mixed Reality Human–Robot Interface for Remote Operations in Hazardous Environments. IEEE Access, 11, 17305-17333. DOI:10.1109/ACCESS.2023.3245833

[15] Wen, J.T., Wason, J.D., Kruse, D., Peng, Y., & Chen, S. (2021). Collaborative Industrial Robot Control: From Safe Motion to Multi-Robot Manipulation. DOI:10.1142/9789811222849_0005

[16] Xiu, Y., Zhang, Y., Deng, H., Li, H., & Xu, Y. (2025). Collaborative Line-of-Sight Guidance-Based Robust Formation Control for a Multi-Snake Robot. IEEE Transactions on Automation Science and Engineering, 22, 4514-4524. DOI:10.1109/TASE.2023.3348469

[17] Ajisaka, S., Suda, T., Akazawa, Y., & Nakamura, S. (2023). An Introductory Investigation of Collaborative Conveyance Method for Multi-Robot Telexistence. IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society, 1-6. DOI:10.1109/IECON51785.2023.10311733

[18] Xin, J., Xu, T., Zhu, J., Wang, H., & Peng, J. (2024). Long Short-Term Memory-Based Multi-Robot Trajectory Planning: Learn from MPCC and Make It Better. Advanced Intelligent Systems, 6. DOI:10.1002/aisy.202300703

[19] Yoo, T., & Choi, B.W. (2024). Interactive Path Editing and Simulation System for Motion Planning and Control of a Collaborative Robot. Electronics. https://doi.org/10.3390/electronics13142857

[20] Nguyen, A.A., Jabbari, F., & Egerstedt, M. (2024). Resiliency Through Collaboration in Heterogeneous Multi-Robot Systems. IEEE Open Journal of Control Systems, 3, 461-471. https://doi.org/10.3390/electronics13142857

[21] Jing, M., & Xin, B. (2024). Distributed Behavior Control Method for Indoor Multi-robot Collaborative Odor Source Localization. 2024 43rd Chinese Control Conference (CCC), 5428-5433. DOI:10.23919/CCC63176.2024.10662785

[22] Chen, W., Jing, Y., Zhao, S., Yan, L., Liu, Q., & He, Z. (2025). A Distributed Collaborative Navigation Strategy Based on Adaptive Extended Kalman Filter Integrated Positioning and Model Predictive Control for Global Navigation Satellite System/Inertial Navigation System Dual-Robot. Remote Sensing. https://doi.org/10.3390/rs17040721

[23] Lee, S., Lucas, N., Ellis, R.D., & Pandya, A.K. (2012). Development and human factors analysis of an augmented reality interface for multi-robot tele-operation and control. Defense, Security, and Sensing. DOI:10.1117/12.919751

[24] Li, R., Shi, Z., & Wu, H. (2024). Adaptive neural networks-based event-triggered formation control for multi-robot source localization. Neurocomputing, 621, 129275. DOI:10.1016/j.neucom.2024.129275

[25] Cheng, Y., Chen, X., Yang, Y., Wang, H., Xu, J., Hong, C., Xu, S., Zhang, X., Liu, Y., & Chen, X. (2024). SniffySquad: Patchiness-Aware Gas Source Localization with Multi-Robot Collaboration. ArXiv, abs/2411.06121. DOI:10.48550/arXiv.2411.06121

[26] Lajoie, P., & Beltrame, G.A. (2023). Swarm-SLAM: Sparse Decentralized Collaborative Simultaneous Localization and Mapping Framework for Multi-Robot Systems. IEEE Robotics and Automation Letters, 9, 475-482. DOI:10.1109/LRA.2023.3333742

[27] Bao, C., Hu, Y., & Yu, Z. (2024). Current study on multi-robot collaborative vision SLAM. Applied and Computational Engineering. DOI:10.54254/2755-2721/35/20230367

[28] Ji, Y., Chen, F., Chen, B., Wang, Y., Zhu, X., & He, H. (2022). Multi-Robot Collaborative Source Searching Strategy in Large-Scale Chemical Clusters. IEEE Sensors Journal, 22, 17655-17665. DOI:10.48550/arXiv.2407.01308

[29] Tran, V.P., Perera, A.G., Garratt, M.A., Kasmarik, K.E., & Anavatti, S.G. (2024). Active Sensing Strategy: Multi-Modal, Multi-Robot Source Localization and Mapping in Real-World Settings

With Fixed One-Way Switching. Journal of Field Robotics, 42. DOI: 10.1016/j.inffus.2022.11.001

[30] Cao, K., Wei, Y., Chen, Y., Gao, S., Yan, K., & Yang, S. (2025). Multi-robot area coverage and source localization method based on variational sparse Gaussian process. Robotica. DOI:10.1017/S0263574725000050