IAES International Journal of Robotics and Automation (IJRA)
Vol. 13, No. 2, June 2024, pp. 205~219
ISSN: 2722-2586, DOI: 10.11591/ijra.v13i2.pp205-219

Distributed and autonomous multi-robot for task allocation and collaboration using a greedy algorithm and robot operating system platform

Abderrahmane Tamali[1], Nourredine Amardjia[1], Mohammed Tamali[2]
[1]LIS Laboratory, Department of Electronics, Ferhat Abbas University Setif 1, Setif, Algeria
[2]ENERGARID Laboratory, Department of Electrical Engineering, Tahri Mohammed Bechar University, Bechar, Algeria

Article Info

Article history:

Received Jan 24, 2024
Revised Feb 28, 2024
Accepted Apr 23, 2024

Keywords:

3D digitization
Collaboration
Light detection and ranging
Multi-robot
Multi-robot task allocation
Robot operating system
Simultaneous localization and mapping

ABSTRACT

Research investigations in the realm of micro-robotics often center around strategies addressing the multi-robot task allocation (MRTA) problem. Our contribution delves into the collaborative dynamics of micro-robots deployed in targeted hostile environments. Employing advanced algorithms, these robots play a crucial role in enhancing and streamlining operations within sensitive areas. We adopt a tailored GREEDY approach, strategically adjusting weight parameters in a multi-objective function that serves as a cost metric. The objective function, designed for optimization purposes, aggregates the cost functions of all agents involved. Our evaluation meticulously examines the MRTA efficiency for each micro-robot, considering dependencies on factors such as radio connectivity, available energy, and the absolute and relative availability of agents. The central focus is on validating the positive trend associated with an increasing number of agents constituting the cluster. Our methodology introduces a trio of micro-robots, unveiling a flexible strategy aimed at detecting individuals at risk in demanding environments. Each micro-robot within the cluster is equipped with logic that ensures compatibility and cooperation, enabling them to effectively execute assigned missions. The implementation of MRTA-based collaboration algorithms serves as an adaptive strategy, optimizing agents' mobility based on specific criteria related to the characteristics of the target site.

Corresponding Author:

Abderrahmane Tamali
LIS Laboratory, Department of Electronics, Setif 1 University-Ferhat Abbas
Setif, Algeria
Email: abdou.t10@gmail.com

1. INTRODUCTION

Robotic technology has the potential to replace human workers in high-risk scenarios, reducing the potential for human and financial loss. Recent research has led to progress in developing autonomous techniques and devices capable of operating without direct human intervention, including working online and collaborating with humans in hazardous environments. A major focus of our research is the study of the optimal number of robots per task of a cluster (multi-robot teamwork) under specifications and particular potentials. The primary objective is to refine control methods for multi-robot in hostile areas. The importance of this research is underlined by its significant professional benefits and potential applications. Micro-robots are expected to play a key role in improving task distribution and execution, as well as facilitating the transmission of relevant information and collected data [1]–[4]. The challenging transmission process makes use of affordable means and logic strategies, such as radio communication, within the cluster.

To facilitate simulation, we have chosen to utilize the open-source robotics operating system (ROS) solution along with its associated tools and modules [5], [6]. ROS will streamline the process of translating both conceptual and physical models into reality. An illustrative instance within this field has been developed by [7].

Multi-robot task allocation (MRTA) problems are intricate case studies. In operational research (OR) and for NP-class problems characterized by numerous constraints and a single objective function, special approaches are used, such as differential methods and AI-based algorithms. Our decision-making process adheres to specific criteria to adopt a proactive and constructive approach: i) form and structure (dimensions, degrees of freedom), ii) data transmission (means, persistence), iii) economy and energy autonomy (energy security), iv) equipment carried out as payload (sensors), and v) embedded logic.

The collected data is vital for the cluster. Each agent undergoes an evolution process through a collaborative treatment approach. Our goal is to ensure efficient completion of the task in the shortest time, while maintaining the importance of the collected data and utilizing dedicated resources.

The main objective addressed by Rekleitis *et al.* [8] is the optimization of the generated map by minimizing errors at the estimation level, contingent on the robot's capabilities and the attached payload. Task coordination for optimal multi-robot evolution through a task-based multi-robot task allocation (MRTA) optimal assignment problem (OAP) is emphasized by Gerkey and Matarić [9]. Additionally, Lee *et al.* [10] focus on developing a metric for estimating fault levels within a swarm of robots, while Zhang *et al.* [11] introduce a model based on data correlation, specifically the correlated random walk model, to efficiently approximate task searching time for distributions of multi-robot systems in large arenas.

Recent literature, including [12] and [13], explores bio-inspired techniques for collaboration and sharing state information between pursuing agents and fast evaders. Collaboration necessitates meeting specific criteria, such as sharing each agent's state information and utilizing onboard resources to complete tasks. The context of a hostile site adds complexity, demanding strategies with mathematical intricacy [14], [15], influenced by ecosystem characteristics (multi-robot setups, tasks, region of interest, and resources). In the case of a hostile site with intricate morphology, the challenge intensifies, relying on capabilities like finding a direct path. The objective is to derive optimal solutions [16] for the routes to be followed, minimizing time to reach the rescue target [17].

In the context of industrial sites, the focus shifts dramatically, with an emphasis on completing surveillance in the most relevant manner for incident localization [18]. Other research concentrates on the capabilities that micro-robots must possess to acquire cognitive abilities, enabling them to navigate and evolve on the study site using reinforced learning methods [19]. Regarding morphology, concrete examples like MIT's Cheetah 3 [20] or ANYBOTICS' ANYmal from ETH Zurich [21] meet the requirements of degrees of freedom in challenging sites with heightened aggressiveness.

2. METHOD

Various tools, including proprietary OnShape assembly and open-source Phobos add-on in blender, are utilized for preparatory tasks before simulating under ROS. XACRO, simulation description format (SDF), or unified robotics description format (URDF) scripts are essential for developing 3D robot models with joints for Gazebo/RVIZ simulation. The JETBOT prototype by NVIDIA serves as the modeled micro-robot with updated URDF (SDF) to fit study needs. Elements at this stage are crucial for setting up a compatible simulation scene in line with our approach:

- The actual space is defined by a specific location (the dimensions of an apartment with spatial limitations) in 3D [5], [10], [18].
- The proposed cluster of multi-robot is a formation of three similar agents [22].
- Possibility of the heterogeneous case [23]–[25].
- Target, an injured human (3D model of a human, target behavior complexity).

The standard packaging used in this research are as follows. i) Economy and energy autonomy for a given payload (implementation of ROS battery plug-in as shown in Figure 1 within the URDF joints with its compatible compiled library). ii) Ensure stability in communication links using the ROS radio connectivity plug-in with the cluster. iii) Enhance collaborative capabilities for improved performance by implementing necessary plug-ins, rules, and algorithms, as illustrated in Algorithm 1. iv) Algorithm enhancements-based AI (AI Logic code integrated). v) Minimum agent's size and volume (for swarm extension purposes).

The scene requirements in this research are: i) The cluster incorporates three micro-robots (agents) denoted as $\mu R_i|_{i\epsilon[1,R]}$, each equipped with motorized wheels for locomotion (differential, driver). ii) Every agent unit is furnished with transmission capabilities for data transmission and maintaining communication links. iii) The autonomy of every agent is supported by an onboard battery, providing sufficient power for its activities within the cluster. iv) Every agent possesses an integrated AI algorithmic foundation, empowering it to make decisions for both collaborative and individual purposes. v) Every agent unit within the cluster is outfitted with essential equipment, including a Camera, LiDAR (light detection and ranging), a motor driver, actuators, and sensors as shown in Figure 2. vi) The micro-robot group operates in two modes: Supervised mode and collaboration mode.

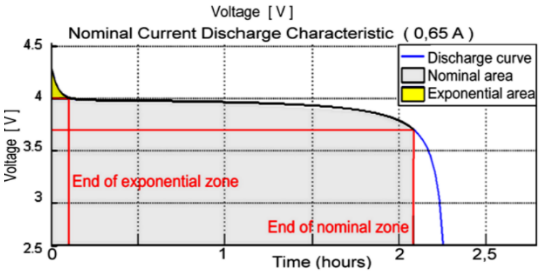


Figure 1. Nominal current discharge characteristic (0.65A)

Algorithm 1. MRTA, Greedy oriented search algorithm

```
Begin
   Initializations:
      Let $S^k \leftarrow \emptyset$
      Let $K_0 \leftarrow k'$                    {$\forall$ k' $\in$ [1;R] , R: number of robots}
      Let $T^k \leftarrow \emptyset$
      Let $P_i^k = \begin{cases} (P_i^k)^* \text{ get all previous tasks if they exist} \\ \emptyset \text{ otherwise} \end{cases}$
   for (k $\in$ [1; R[) do
      $S^k$= area(k)
      if (k $\neq$ $K_0$) then
         Let $S^{K_0} \leftarrow (S^{K_0} - S^{K_0} \cap S^k)$
      end if
   end for
   $T \leftarrow task(K_0)|_{S^{K_0}}$
   $i \leftarrow 1$
   while ($i \leq |T|$) do
      $t_i^{K_0} \leftarrow Eval(f_i^{K_0}|_{S^{K_0}})$          {ith objective function}
      if ($T_i^{K_0}(t_i^{K_0}) \subset P_i^{K_0}$) then
         $t_i^{K_0} \leftarrow None$
      end if
      $i \leftarrow i+1$
   end while
   $T^{K_0} \leftarrow T_j^{K_0}|_{S^{K_0},Best(t_j^{K_0})}$          {for j $\in$ [1;m], m is set of specific
   if ($T^{K_0} \neq \emptyset$) then
   tasks}
      $P_i^{K_0} \leftarrow P_i^{K_0} + T^{K_0}$
   else
      $T \leftarrow \emptyset$
      for (k $\in$ [1; R[) do
         if (k $\neq$ $K_0$)do
            $T \leftarrow T + task(k)|_{S^k,k \neq K_0}$
            $i \leftarrow 1$
            while ($i \leq |T|$) do
               $t_i^k \leftarrow Eval(f_i^{K_0}|_{S^k,k \neq K_0})$
               if ($T_i^k(t_i^k) \subset P_i^k$) then
                  $t_i^k \leftarrow None$
               end if
               $i \leftarrow i+1$
            end while
            $T^k \leftarrow T_j^k|_{S^k,Best(t_j^k),k \neq K_0}$          {for j $\in$ [1;m] and m
         end if                                  is set of specific tasks}
      end for
      $T^{K_0} \leftarrow Best(T^k)|_{k \in \{1,2,\dots,R\}}$
      $P_i^k \leftarrow P_i^k + T^{K_0}|_{k \in \{1,2,\dots,R\}}$
   end if
End.
```

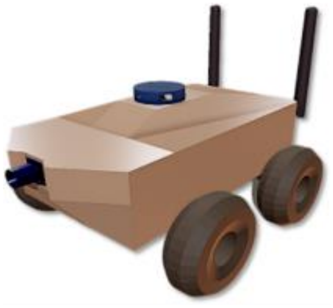


Figure 2. AL Mustaksheef3D, wheeled robot developed

We aim to effectively locate and aid victims in challenging situations through Collaborative MRTA. Our approach involves the implementation of 3D digitization utilizing LiDAR or a depth camera, ensuring optimal outcomes for Search and simultaneous localization and mapping (SLAM) purposes. The ROS environment serves to manage the context, and, in addition to interoperability, ROS can engage with various platforms dedicated to simulating MRTA problems [26]–[30]. The ROS.MSG embedded module facilitates message exchange among micro-robots, with inertial measurement unit (IMU) data conversion to ODOM (odometry) enabling the validation and voting of agent moves in areas that are still unoccupied as shown in Figure 3.

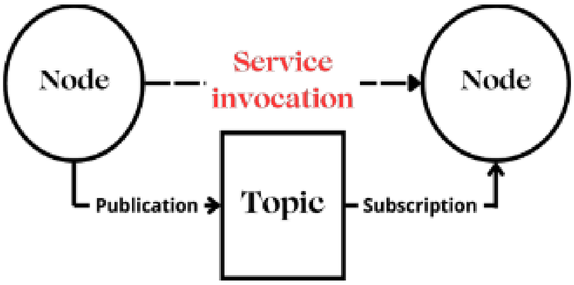


Figure 3. ROS basics and concepts

2.1. Distributed MRTA use case and problem statement

Achieving coordinated teamwork among robots with real-time task distribution necessitates a decentralized framework [31] that covers resilient robot awareness, low-level motion control, and high-level task scheduling [32], [33]. Effective location management in multi-robot networks is critical for collaboration, where decision-making and communication play pivotal roles in mission distribution, presenting substantial challenges in data exchange among robots and the operating station [34]. In approaches that are self-organized and decentralized, individual robot nodes independently make decisions with limited regard for other agents [35]. Decisions in these approaches are frequently influenced by natural or real-world phenomena, as highlighted in [3], including bees swarm, market strategy, swarm intelligence, and ant colony. These inspirations give rise to complex collective behavior arising from local interactions among numerous agents with straightforward behaviors. In such methodologies, sensors play a pivotal role by actively collecting local knowledge for sharing within the cluster, as emphasized by Stolzle *et al*. [7] and Ball *et al*. [36]. The collaborative use of sensors facilitates the accumulation of necessary knowledge pertaining to an overarching goal in these procedures as shown in Figure 4.

Robots need to be able to understand the tasks they have to perform by collecting data through sensors and using specific code to make decisions. Broadband, range, power, and data rates are critical to system performance. Point-to-point communication is the most basic form, and the choice of transmission medium is a function of the type of information being exchanged. Robot evolution on hostile sites can, in some manner, be assumed to be a progression of the vehicle forming a path on a surface contoured by a set of N points in a space defined in a plane delimited by a closed polygon, where $P = P_1, P_2, \dots, P_N$ define a poly-point or a set of N points as shown in Figure 5.

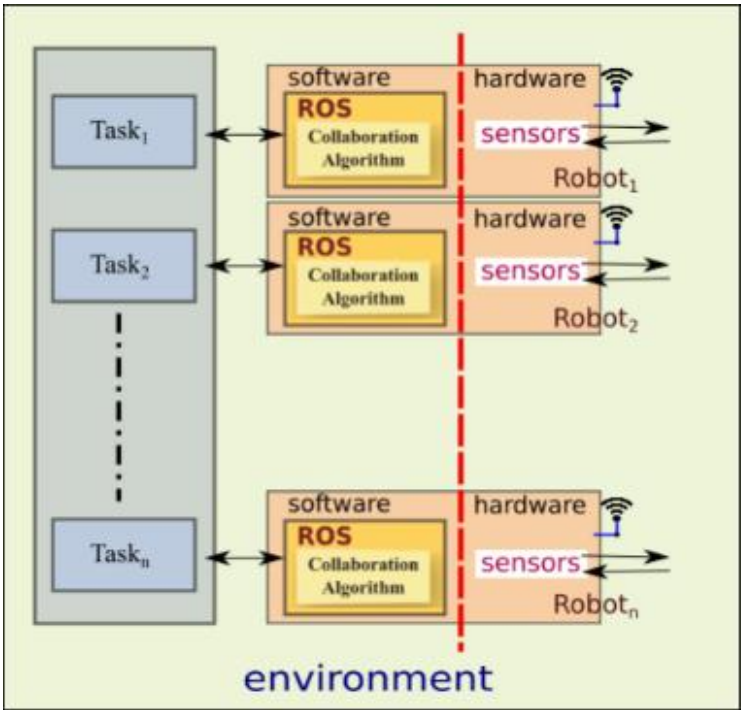


Figure 4. Cluster architecture

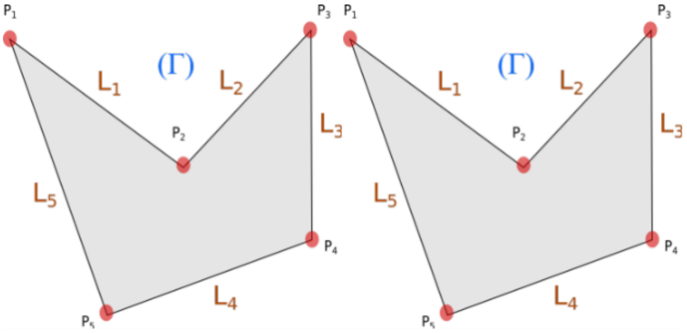


Figure 5. N-gon of a scanned area

Normally each point is located on the plan (Γ) by its Cartesian coordinates x_i and y_i. A one-line equation can be written in the form of (1).

$$y = a.x + b \tag{1}$$

a and b are two parameters related to the jth line (slope and y-intercept) in the Polyline defined by P components. By using each couple of points coordinates, the related a and b parameters are obtained according to (using Cramer's rule).

$$a = \frac{y_1 - y_2}{x_1 - x_2} \tag{2}$$

$$b = \frac{x_1 y_2 - x_2 y_1}{x_1 - x_2} \tag{3}$$

So, (1) can be like follows for a line equivalent (L$_1$) between two points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$. y is given by (4).

$$y = \begin{cases} \dfrac{y_1 - y_2}{x_1 - x_2} \cdot x + \dfrac{x_1 y_2 - x_2 y_1}{x_1 - x_2} \Bigg|_{\substack{x_1 \leq x < x_2, \forall x_1 - x_2 < 0 \\ x_2 \leq x < x_1, \forall x_1 - x_2 > 0 \\ \\ y_1 \leq y < y_2, \forall y_1 - y_2 < 0 \\ y_2 \leq y < y_1, \forall y_1 - y_2 > 0}} \\ 0 \;\; otherwise \end{cases} \tag{4}$$

Now, we need to construct a Polygon equation using a combination of multiple-line equations. We have N points, which implies that the number of line equations is $N(L_1, L_2, \ldots., L_N)$. The n-gon's (polygon) formula is given by (5).

$$Y = Y_1 + Y_2 + \cdots\cdots + Y_N = \sum_{i=1}^{N-1} Y_i + Y_N \tag{5}$$

Witch is $Y_1, Y_2, \ldots., Y_N$ or simply Y_I, Y_N where $i \in \mathbb{N}$ natural strictly positive number and, the set Y_I with $i \in \mathbb{N}$ represent each i^{th} line's equation:

$$Y_i = y = \begin{cases} \dfrac{y_i - y_{i+1}}{x_i - x_{i+1}} \cdot x + \dfrac{x_i y_{i+1} - x_{i+1} y_i}{x_i - x_{i+1}} \Bigg|_{\substack{x_i \leq x < x_{i+1}, \forall x_i - x_{i+1} < 0 \\ x_{i+1} \leq x < x_i, \forall x_i - x_{i+1} > 0 \\ \\ y_i \leq y < y_{i+1}, \forall y_i - y_{i+1} < 0 \\ y_{i+1} \leq y < y_i, \forall y_i - y_{i+1} > 0}} \\ 0 \;\; otherwise \end{cases} \tag{6}$$

Equation valid for $1 \geq i \geq N - 1$, and the last line (L_N):

$$Y_N = y = \begin{cases} \dfrac{y_1 - y_N}{x_1 - x_N} \cdot x + \dfrac{x_1 y_N - x_N y_1}{x_1 - x_N} \Bigg|_{\substack{x_1 \leq x < x_N, \forall x_1 - x_N < 0 \\ x_N \leq x < x_1, \forall x_1 - x_N > 0 \\ \\ y_1 \leq y < y_N, \forall y_1 - y_N < 0 \\ y_N \leq y < y_1, \forall y_1 - y_N > 0}} \\ 0 \;\; otherwise \end{cases} \tag{7}$$

For $x, y, \alpha \in \mathbb{R}$ and $f(x) = y = \alpha$, The area inside the irregular polygon can be defined as the result of:

$$\mathrm{Area_Y} = \int_{\mathbb{R}} \alpha \big|_{\substack{\mathrm{f(x)=\alpha=Y=\{x';x'';\ldots;x^n;x^{n+1}\}} \\ \mathrm{where}\ \mathrm{x^n, x^{n+1} \in \gamma(x)}}} \mathrm{d}\alpha \tag{8}$$

Consider expressions like $n = 2m + 1|_{m \in N}$, where $x_n < x_n + 1$, and $\gamma(x)$ representing the variation domain of the variable x. Consequently, the explored area by the kth robot can be denoted as *Area(k)*, where $k=1,2, \ldots, R$, and R is the total number of cluster agents. We introduce the relation $g_k(\alpha,x)$, which satisfies the condition $f(x)=\alpha=Y=\{x';x'';x''';.;.;x^n;x^{n+1}\}$. This relation, denoted as g_k, aids in determining whether the robot k is situated within the designated *Area(k)* or not. T_0 assess whether a robot is outside or inside an area, we can analyze four zone-shaped situations illustrating the most probable cases, as depicted in Figures 6(a), 6(b), 6(c), and 6(d).

In the first scenario as shown in Figure 6(a), $robot_1$ is situated in zone Z, while the other robots ($robot_2$ and $robot_3$) are not. Formulating this situation involves expressions such as $g_1(\alpha_1,x)=g_2(\alpha_2,x)=Z\{x',x''\}$ and $g_3(\alpha,x)= ø$. Therefore, by comparing x_{R_k} with x' and x'', we can draw the following conclusions:

- If gk(α,x)=ø ⇒ the robotk is outside zone Z.
- If x'≤x_{R_k}≤x" and gk(α,x)≠ø ⇒ the robotk is within zone Z.
- If x_{R_k}<x' or x"<x_{R_k} and gk(α,x)≠ø ⇒ the robotk is outside zone Z.

In the second scenario as shown in Figure 6(b), we have $g_1(\alpha_1,x)=g_2(\alpha_2,x)=g_3(\alpha_3,x)=Z\{x',x'',x''',x''''\}$. $robot_2$ is within the area where $x' \leq x_{R_2} \leq x''$, while the others are not. Specifically, $x'' < x_{R_2} < x'''$ and $x_{R_2} < x'$, leading to the following determinations:

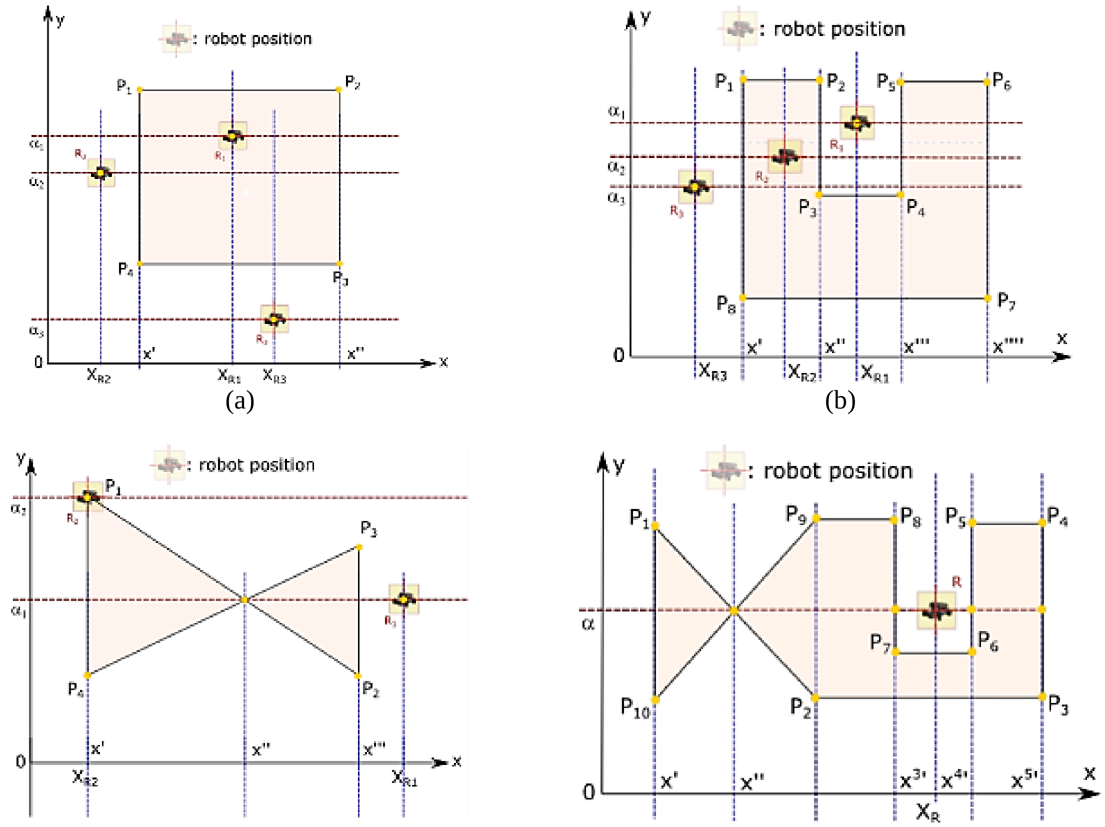
- If $x^n \leq x_{R_k} \leq x^{n+1}\big|_{n=2m+1, \forall m \in N}$ and $g_k(\alpha,x) \neq ø$ ⇒ the $robot_k$ is within the area.
- If $x^n < x_{R_k} < x^{n+1}\big|_{n=2m+2, \forall m \in N}$ and $g_k(\alpha,x) \neq ø$ ⇒the $robot_k$ is outside the area.

The third situation as shown in Figure 6(c) encompasses two singular cases, where $g_1(\alpha_1,x)=\{x',x'',x'''\}$ and $g_2(\alpha_2,x)=\{x'\}$. The robot's presence in the area can only be determined in these cases where $|g_k|=3$ for $robot_1$ and $|g_k|=1$ for $robot_2$:

- If $x' \leq x_{R_k} \leq x'''$ and $g_k(\alpha,x) \neq ø$ where $|g_k|=3$ ⇒ the $robot_k$ is in the area.

- If $x' = x_{R_k}$ and $g_k(\alpha,x)\neq\emptyset$ where $|g_k|=1 \Rightarrow$ the $robot_k$ is in the area.

In the last case as shown in Figure 6(d), it is impossible to ascertain whether the robot is in the zone or not using the relation g_k, where $g_k(\alpha,x)\neq\emptyset$, and $|g_k|=n|_{n=2m+5,\forall m \in\mathbb{N}}$.



(a) (b)

(c) (d)

Figure 6. Potential scenarios for the robot's posture (a) simple case with 3 robots, (b) complex case with 3 robots, (c) singular case, and (d) insoluble case

2.2. Simulation cases and success factors

Before initiating our simulations, we define our context and our environment. To simulate collaboration and explore to locate a victim, a workstation is employed. This workstation features an HP ProBook x360 435 G7 with an AMD Ryzen 7 PRO 4750U processor, AMD Radeon RX Vega 7 graphics, 32 GB DDR4 Kingston RAM, and a 1TB Samsung SSD Drive. The simulation utilizes ROS [37] on Linux Ubuntu 20.04, enhancing node communication and facilitating message passing between robots and computers. Additionally, JetBot, powered by NVIDIA Jetson Nano and accessible at NVIDIA, supports sensors and implements RNN for object detection and collision avoidance. Furthermore, JetBot is capable of connecting through various radio links.

2.2.1. Optimization based algorithm

The next flowchart illustrates the overall process as shown in Figure 7. The cluster collects data related to the ROI's context, investigates frontiers around each micro-robot, generates map fragments, and shares them within the cluster. If tasks are completed, processes can be disposed of, otherwise, the preparation step is repeated. In a practical scenario, each micro-robot needs to meet the optimal condition defined by the cost function f_{cost}^{R} in (9).

- Radio connectivity to the sink or to all cluster $F_{con/AP}$ which guarantees data exchange and/or maintain a link (AP for access point).
- Economic and energy autonomy F_{auton} quantifies the battery lifespan for a particular task.
- Absolute availability $Disp^{/all}$, Denotes the *'OK state'* of the unit within the cluster, signifying its capability to proficiently carry out the task.

- The relative availability, denoted as $Disp^{/Res}$, signifies the accessibility of the payload, which is a specific resource crucial for a particular task.

$$f_{cost}^{\mu R} = k_1 * F_{con/AP} + k_2 * F_{auton} + k_3 * Disp^{All} + k_4 * Disp^{/Res} \quad \text{where } k_1+k_2+k_3+k_4=1 \tag{9}$$

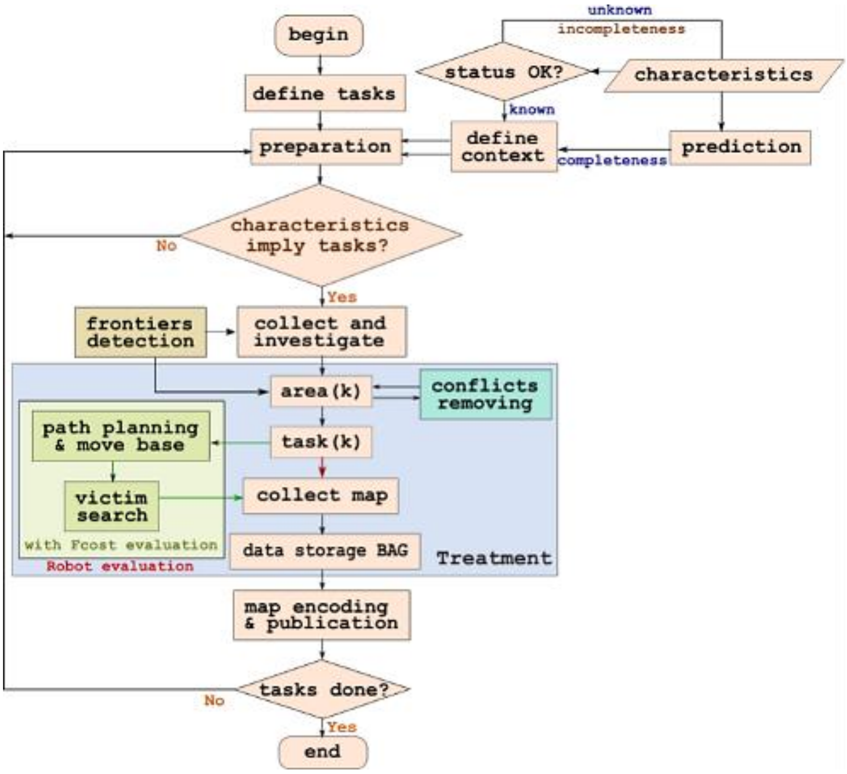


Figure 7. Flowchart of our approach

The weights k_i for $i\in[1,4]$ symbolize the contributions of each specific part to the agent's functional cost, as defined earlier. Our goal is to discover the optimal outcome (10) for the function f_{cost} for every agent in the cluster, representing the best solution. Subsequently,

$$F_{cost}^{opt} = Max|_{j\in[1,R[} \left\{ f_{cost}^{\mu R_j}, constraints \right\} \tag{10}$$

The cluster's optimal estimate is indicated by its capability to declare an 'OK' status for availability, surpassing the minimum required condition for executing assigned tasks. We introduce $F_{con/AP}$ as the reduced effective availability of at least one link with a predefined access point. If $\mathrm{F}_{\mathrm{con}}^{\mathrm{ref}}$represents the reference threshold for a Wi-Fi connection to an access point and $F_{con}(t)$ signifies the current connectivity level of an agent to the access point, then $F_{con/AP}$ is defined as the ratio of the instant connectivity $F_{con}(t)$ to the reference threshold $\mathrm{F}_{\mathrm{con}}^{\mathrm{ref}}$ This is expressed as (11):

$$\mathrm{F}_{\mathrm{con/AP}} = 100 * (\mathrm{Fcon(t)}/\mathrm{F}_{\mathrm{con}}^{\mathrm{ref}}) \tag{11}$$

Therefore, F_{auton} signifies the battery autonomy, indicating the remaining energy in Ah needed by the battery to provide adequate power for the agent to successfully accomplish the assigned task as a singular unit within the cluster. Here, we represent τ as the estimated time required for the assigned task, I as the actual discharged current of the battery in Ampere (A), and C as the battery capacity, acting as a current source for a specified duration in Ampere-hour (Ah). Hence, the expressions are as (12).

$$F_{auton} = C/\tau * (1/I - 10/P_{charge}) \tag{12}$$

In estimating $Disp^{/all}$ and $Disp^{/Res}$, the availability of a cluster member is assessed based on its effective suitability for any given task. Relative availability for a resource indicates the agent's capacity to have the specified resource (payload) ready at the designated time when a task is assigned. On the other hand, absolute availability encompasses the sum of all relative availabilities (of N_{res} resources), signifying that all of the agent's resources are ready for use. The relative availability is set to 1 when there is positive feedback upon querying the resource and 0 otherwise. Therefore:

$$D_{Rel/Res} = \begin{cases} 1 \; if \; the \; targeted \; resource \; is \; available \\ \qquad 0 \; otherwise \end{cases} \tag{13}$$

Absolute availability is given by (14).

$$Disp^{All} = \bigcap_{i=1}^{N_{res}} ({D_{Rel/Res}}^{i}) \tag{14}$$

The agent's ability to fully intervene in a task depends on the logic AND connection of all relevant availability functions. The weighting coefficients k_i are selected using various methods to optimize the objective function's final result. We adopt a bio-inspired approach to determine these coefficients.

2.3. Collaboration by greedy algorithm for distributed MRTA

We introduce a greedy algorithm in Algorithm 1 [3] designed for the MRTA problem with broadcast messaging. This algorithm is configured to allocate agents based on the optimal assignment to a task that maximizes the performance-to-cost ratio (V_i^k/d_i^k). Here, V_i^k represents the performance of the k^{th} agent for the accomplishment of the i^{th} task, and d_i^k denotes the Euclidean distance between the agent and the target/assigned location (8) and (9).

$$\mathrm{d_i^k} = \sqrt{(\mathrm{x_i} - \mathrm{x_k})^2 + (\mathrm{y_i} - \mathrm{y_k})^2 + (\mathrm{z_i} - \mathrm{z_k})^2} \tag{15}$$

$$T_i^k = \max_{i \in T}\left(V_i^k/d_i^k\right) = \max_{i \in T}\left(V_i^k \eta_i^k\right) \text{ with } \eta_i^k = 1/d_i^k \tag{16}$$

Here, task i is the chosen task for the k^{th} agent from the entire set of potential tasks, and T represents the set of viable tasks within the k^{th} agent's scope out of the total M available tasks. The algorithm, identified as Algorithm 1, operates as follows: Initially, we define surface variables explored by each $robot_k$, where $S_k|_{k \in R}$ denotes the set of boundary points of the presently scanned surface. Additionally, $f_i^k|_k$ represents an objective function (cost function) for each $robot_k$, with $task_i$ being the sub-mission stored in T_i^k Furthermore, T_i^k is the collection of tasks that have not been assigned yet and are included in T, representing the total available tasks $(T_i \in T)$. Concurrently, T^k for k ∈ R is the selected task for the robot, and P_i^k denotes the locations that have already been assembled. Here, k indicates the robot that needs to update its bid at the current stage.

At the start, no tasks are assigned, so $T^k=\emptyset$ for all robots $k \in \mathbb{R}$. In each step, one task is allocated to a single robot independently as shown in Figure 8, following decentralization. Thus, we need $|T|$ steps, the number of tasks for the robot, to complete its state. At each iteration i, after removing conflicting parts with areas explored by other robots, all robots $k \in R$ submit an offer (t^k_i, T^k_i). Each robot k selects task T^k from the list of non-located tasks T_i^k to maximize its objective function f_i^k. Upon gathering all the bids, we discover a superior optimal gain for the collective objective: the multiplicative success of group F (17). Through bidding, we efficiently choose the optimal task pair-robot combination for the greatest overall benefit [38], [39].

$$F = Best_{\{T^k\}_{k \in R}} \prod_{k \in R} [f_k\,(T^k)] \tag{17}$$

When a robot is surrounded by other robots, it can be misled about its capabilities, hindering its development and exploration. This situation can impede the robot's ability to effectively navigate and fulfill its tasks. To address this issue and maintain high performance even when surrounded ($T^k=\emptyset$), our objective is to enhance the robot's cost function by utilizing the neighboring robots' available spaces until it can operate independently. By incorporating information from surrounding robots, the robot can make more informed decisions and adapt its behavior, accordingly, ultimately improving its overall performance and autonomy.

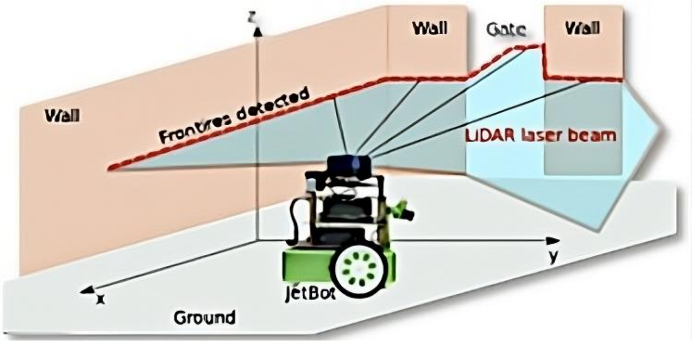


Figure 8. JetBot during domain exploration

3. RESULTS AND DISCUSSION

The simulation scenario is illustrated in Figure 9. It is characterized by various constraints that evaluate the efficiency and reliability of our approach in navigating a cluster of micro-robots, both with and without collaboration. The simulation scene consists of rooms (isolated spaces) containing furniture (obstacles). Micro-robots work together to rescue multiple victims as shown in Figure 10, to minimize time and enhance reliability. The simulation illustrates three severity levels involving 1-3 micro-robots as shown in Figure 11.

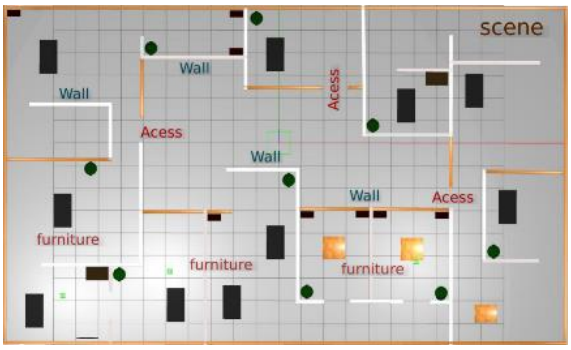


Figure 9. Model area of the simulation

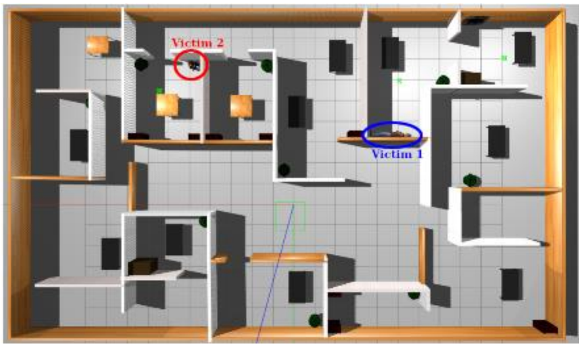


Figure 10. Scene with target location

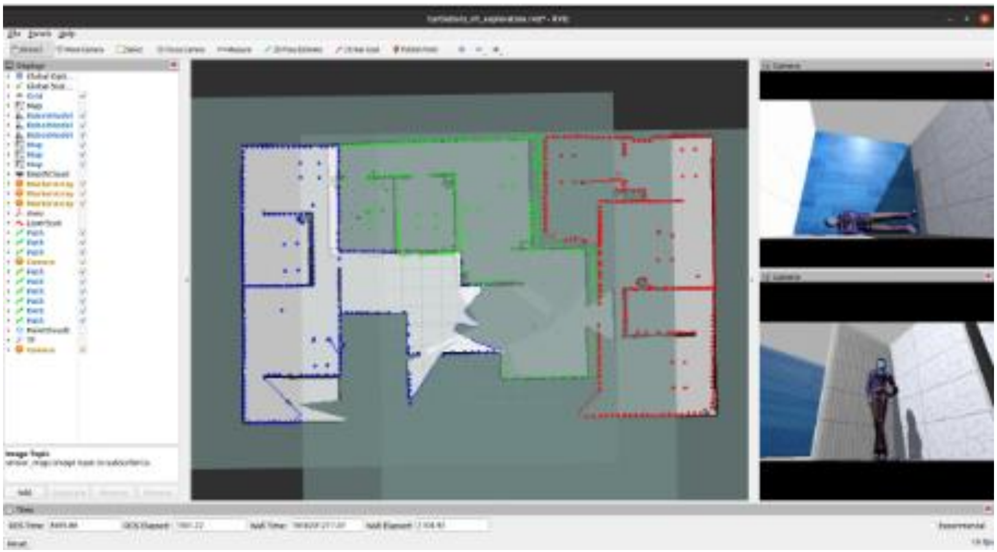


Figure 11. Space segmentation

We make the following assumptions for the simulation:

- The scene's domain and dimensions remain constant throughout the investigation.
- The target position within the scene does not significantly affect the simulation's duration.
- The agents (micro-robots) are assumed to be homogeneous, possessing identical characteristics.
- Connectivity, Payload, and Autonomy conditions are satisfactory for all units within the cluster.

The simulation's outcomes are presented in the subsequent Figures 11 to 15 and Tables 1 to 8. The plan involves finding the time to reach each victim, pinpointing the longest time to reach two victims, and averaging the maximum values. We focus on the maximum and minimum average durations. The regions in Figure 15 show designated areas in RGB colors within the All Area. These areas come from segmenting the total ROI using the greedy algorithm. The process is repeated three times. The first iteration involves a single robot over ten attempts, and the results are in Table 1. Where time distribution by number of attempts is given (with trend function) in the next chart as shown in Figure 16.

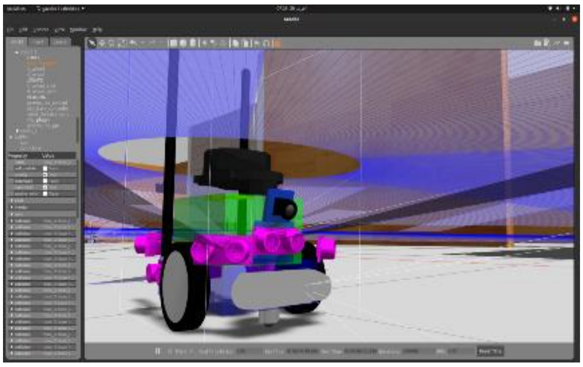


Figure 12. The used robot model

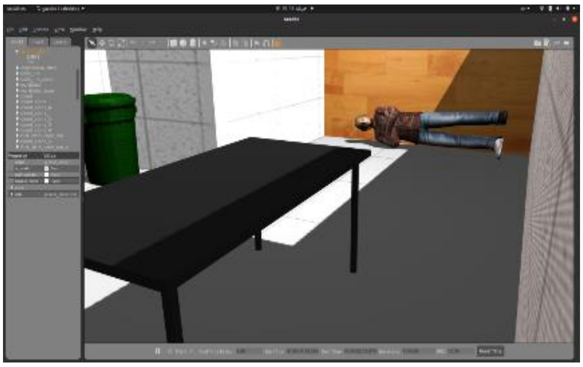


Figure 13. GAZEBO Scene Simulator

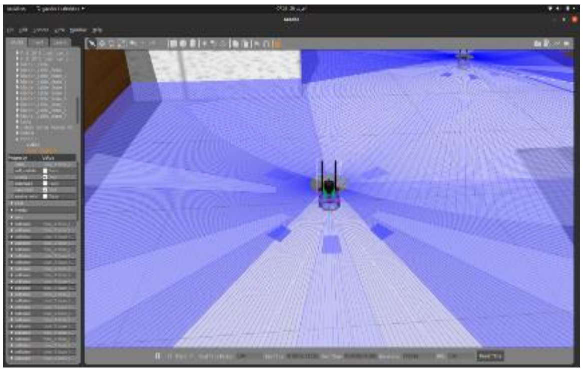


Figure 14. Robots while performing a mission

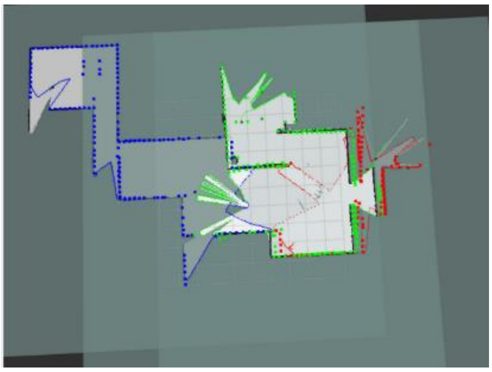


Figure 15. Agents ROI updates

Table 1. Simulation time to find the two victims (in seconds)

Attempts	Target 1	Target 2	To mission's end
1st	55	102	102
2nd	70	30	70
3rd	20	150	150
4th	93	35	93
5th	64	86	86
6th	46	90	90
7th	77	112	112
8th	95	150	150
9th	9	76	76
10th	110	123	123
Mean	63.9	95.4	105.2

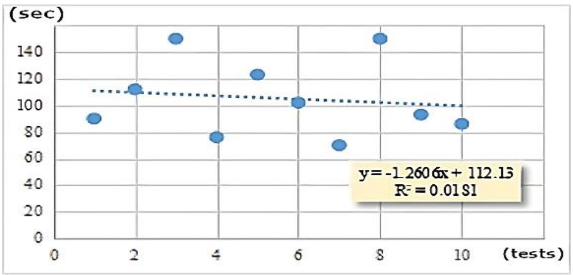


Figure 16. Duration taken by a single robot to reach the target on the initial attempt

The process involves ten simulations, each done three times. Results are assessed across five different scenarios (cases I to V). In each test, we note the time to meet all goals until finding two targets (victims), with and without considering cooperation.

- Scenario I: single robot (refer to Table 2).
- Scenario II: two robots without collaboration (refer to Table 3).
- Scenario III: two robots with collaboration (refer to Table 4).
- Scenario IV: three robots without collaboration (refer to Table 5).
- Scenario V: three robots with collaboration (refer to Table 6).

Table 2. Simulation time of one robot to find the two victims (in seconds)

	target1	target2	duration
Exp.1	80.2	37.5	90.7
Exp.2	67.5	99.5	120.6
Exp.3	63.9	95.4	105.2

Table 3. Two robots without collaboration algorithm average time (sec)

	target1	target2	duration
Exp.1	72	38.2	85.3
Exp.2	72	61.6	95.2
Exp.3	62.4	40.3	66.9

Table 4. Two robots with collaboration algorithm average time (sec)

	target1	target2	duration
Exp.1	27.2	21.7	34.2
Exp.2	33.8	30.4	47.5
Exp.3	60.1	38.9	41.7

Table 5. Three robots without collaboration algorithm average time (sec)

	target1	target2	duration
Exp.1	72.6	32.2	76.5
Exp.2	57.8	39.3	69.1
Exp.3	71.2	29	73.9

Table 6. Three robots with collaboration algorithm average time (sec)

	target1	target2	duration
Exp.1	22.7	7.9	22.7
Exp.2	18.3	21.5	24
Exp.3	18.4	14.1	20.7

Aggregate outcomes are compiled as the min and max values for potential consolidation as shown in Tables 7 and 8. Table 7 presents the average time durations across various experimental conditions, offering insights into the comparative efficiency of different strategies. In contrast, Table 8 delineates the minimum and maximum simulation times required to locate two victims, providing a comprehensive view of performance variability under different scenarios.

Table 7. Time duration average by experiment

	Exp.1	Exp.2	Exp.3
One Robot	90.7	120.6	105.2
2NoCollab	85.3	95.2	66.9
2WCollab	34.2	47.5	41.7
3NoCollab	76.5	69.1	73.9
3WCollab	22.7	24	20.7

Table 8. Simulation time to find the two victims (in seconds)

	One	2NoCol	2WCol	3NoCol	3WCol
Min	90.7	66	34.2	69.1	20.7
Max	120.6	95.2	62.1	76.5	24

Collaboration increasingly impacts the duration of time as shown in Figures 17 and 18. These figures vividly illustrate the significant influence of collaboration on time duration, underscoring its growing importance in various contexts. Additionally, as the number of agents increases, the duration decreases, particularly when a cooperative strategy is enabled, highlighting the efficiency gains achieved through collaborative efforts, especially in scenarios involving larger groups.

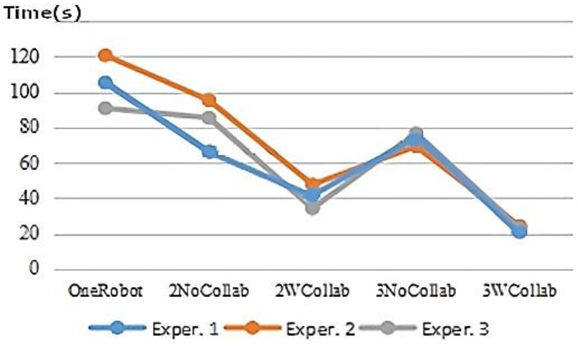


Figure 17. Progression of time duration through experimentation

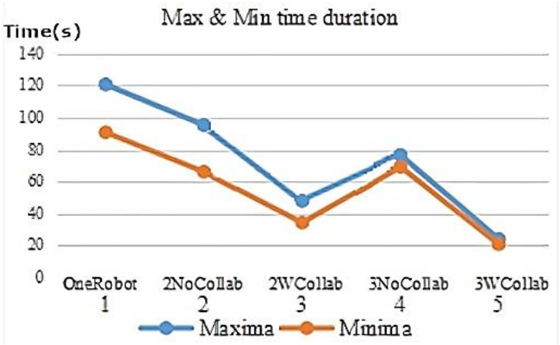


Figure 18. Max and min time duration by experiment until the victim is found

The ROI was evaluated in two ways: by pinpointing the targets and by 3D scanning the area for potential reconnaissance. This was done using an enhanced version of a horizontal LiDAR (RPLiDAR A1M8) as part of the new features on our AL Moustaksheef3D platform, a robotic payload for 3D SLAM currently in development. The enhanced LiDAR as shown in Figures 19 and 20 was tested on a wheeled machine and a Drone. The collected data as shown in Figure 21 is provided in point cloud (PCL) of ".pcd" format. This model integrates two LiDARs, with the first used for geolocation and the second for 3D vision. Scene reproduction yields relative data for estimating facts in the study area. Point-cloud data is used for scene reconstruction. Software like CloudCompare, MeshLab, Blender, and Gimp enable background processing on point cloud data. The micro-cobot agents integrate APIs into their embedded logic, employing robust algorithms for identifying specific targets in hostile environments. Using an MRTA approach with the enhanced Greedy algorithm, our strategy optimizes target identification through cluster agent cooperation, significantly reducing task time. This collaborative approach requires substantial processor capabilities, with basic calculations centralized for collaboration and decentralized for independent agent decision-making post-task assignment.

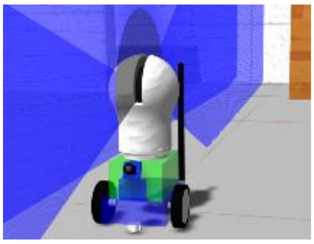


Figure 19. JetBot/AlMustaksheef3D

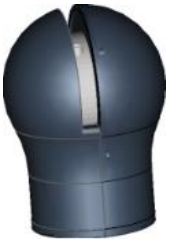


Figure 20. The 3D model of the new LiDAR

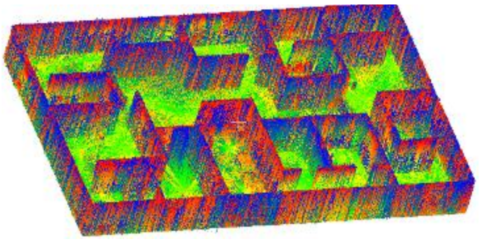


Figure 21. ROI SLAM Segmentation, on RVIZ and Gazebo

4. CONCLUSION

The project aims to design collaborative robots with physical capability and intelligence to work effectively in challenging environments. These cobots will help reduce the workload of human response or rescue teams by identifying targets. They will also help in areas where the density of robots and technicians poses a risk to human safety.

Our research focuses on micro-robots collaborating with humans, leading to the concept of micro-cobots. This aims to reduce payload and minimize bottlenecks in tasks. Increasing micro-robot agents enhances investigation speed and target search. The results prove our initial consideration and confirm the predefined hypotheses. The results obtained from the simulation explain that the cooperation significantly improves the cluster's progress in the search mission. Collaboration prevents erratic micro-robot behavior, reduces duration, and prevents system collapse. Fewer agents lead to more dispersion and chaos. A moderate number of investigators ($1<N_R\leq3$) is preferable. Swarm techniques with collaboration are used to manage the situation.

REFERENCES

[1] F. Montori, L. Bedogni, and L. Bononi, “A Collaborative Internet of Things Architecture for Smart Cities and Environmental Monitoring,” *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 592–605, Apr. 2018, doi: 10.1109/JIOT.2017.2720855.

[2] Y. Dong, L. Zhang, and Y. Liu, “Experimental study on control strategy of cobot micro operation force hoisting system,” *Gaojishu Tongxin/Chinese High Technology Letters*, vol. 17, no. 5, pp. 493–497, 2007.

[3] A. El shenawy, K. Mohamed, and H. M. Harb, “Exploration Strategies of Coordinated Multi-Robot System: A Comparative Study,” *IAES International Journal of Robotics and Automation (IJRA)*, vol. 7, no. 1, p. 48, Mar. 2018, doi: 10.11591/ijra.v7i1.pp48-58.

[4] I. El Mallahi, J. Riffi, H. Tairi, A. Ez-Zahout, and M. A. Mahraz, “A Distributed Big Data Analytics Model for Traffic Accidents Classification and Recognition based on SparkMlLib Cores,” *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 16, no. 4, pp. 62–71, Oct. 2022, doi: 10.14313/jamris/4-2022/34.

[5] X. Wang, X. Ma, X. Li, X. Ma, and C. Li, “Target-biased informed trees: sampling-based method for optimal motion planning in complex environments,” *Journal of Computational Design and Engineering*, vol. 9, no. 2, pp. 755–771, Apr. 2022, doi: 10.1093/jcde/qwac025.

[6] S. Michieletto, S. Ghidoni, E. Pagello, M. Moro, and E. Menegatti, “Why teach robotics using ROS,” *Journal of Automation, Mobile Robotics and Intelligent Systems*, pp. 60–68, Jan. 2014, doi: 10.14313/jamris_1-2014/8.

[7] M. Stolzle, T. Miki, L. Gerdes, M. Azkarate, and M. Hutter, “Reconstructing Occluded Elevation Information in Terrain Maps with Self-Supervised Learning,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1697–1704, Apr. 2022, doi: 10.1109/LRA.2022.3141662.

[8] I. M. Rekleitis, G. Dudek, and E. E. Milios, “Multi-robot collaboration for robust exploration,” in *Proceedings-IEEE International Conference on Robotics and Automation*, 2000, vol. 4, pp. 3164–3169, doi: 10.1109/ROBOT.2000.845150.

[9] B. P. Gerkey and M. J. Matarić, “Multi-robot task allocation: Analyzing the complexity and optimality of key architectures,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2003, vol. 3, pp. 3862–3868, doi: 10.1109/robot.2003.1242189.

[10] S. Lee, E. Milner, and S. Hauert, “A Data-Driven Method for Metric Extraction to Detect Faults in Robot Swarms,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10746–10753, Oct. 2022, doi: 10.1109/LRA.2022.3189789.

[11] Y. Zhang, D. Boley, J. Harwell, and M. Gini, “A Correlated Random Walk Model to Rapidly Approximate Hitting Time Distributions in Multi-robot Systems,” in *Lecture Notes in Networks and Systems*, vol. 577 LNNS, Springer Nature Switzerland, 2023, pp. 724–736.

[12] X. Fu, Y. Zhang, J. Zhu, and Q. Wang, “Bioinspired cooperative control method of a pursuer group vs. a faster evader in a limited area,” *Applied Intelligence*, vol. 53, no. 6, pp. 6736–6752, Jul. 2023, doi: 10.1007/s10489-022-03892-8.

[13] G. Merkulov, M. Weiss, and T. Shima, “Virtual Target Approach for Multi-Evader Intercept,” in *2022 European Control Conference, ECC 2022*, Jul. 2022, pp. 1491–1496, doi: 10.23919/ECC55457.2022.9838427.

[14] A. Giusti, G. C. Maffettone, D. Fiore, M. Coraggio, and M. di Bernardo, “Distributed control for geometric pattern formation of large-scale multirobot systems,” *Frontiers in Robotics and AI*, vol. 10, Sep. 2023, doi: 10.3389/frobt.2023.1219931.

[15] S. Wang, Z. Li, H. Gao, K. Shan, J. Li, and H. Yu, “Agile Running Control for Bipedal Robot Based on 3D-SLIP Model Regulation in Task-Space,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13455 LNAI, Springer International Publishing, 2022, pp. 505–516.

[16] P. Fong, C. Fang, and J. He, “Optimal Attack Against Coverage Path Planning in Multi-robot System,” in *Lecture Notes in Electrical Engineering*, vol. 934 LNEE, Springer Nature Singapore, 2023, pp. 1760–1772.

[17] L. E. Parker, “Multiple Mobile Robot Teams, Path Planning and Motion Coordination in,” in *Encyclopedia of Complexity and Systems Science*, Springer New York, 2009, pp. 5783–5800.

[18] S. H. Kim, S. G. Yoon, S. H. Chae, and S. Park, “Economic and environmental optimization of a multi-site utility network for an industrial complex,” *Journal of Environmental Management*, vol. 91, no. 3, pp. 690–705, Jan. 2010, doi: 10.1016/j.jenvman.2009.09.033.

[19] A. Yahya, A. Li, M. Kalakrishnan, Y. Chebotar, and S. Levine, “Collective robot reinforcement learning with distributed asynchronous guided policy search,” in *IEEE International Conference on Intelligent Robots and Systems*, Sep. 2017, vol. 2017-September, pp. 79–86, doi: 10.1109/IROS.2017.8202141.

[20] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, “Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control,” in *IEEE International Conference on Intelligent Robots and Systems*, Oct. 2018, pp. 7440–7447, doi: 10.1109/IROS.2018.8594448.

[21] M. Bjelonic *et al.*, “Offline motion libraries and online MPC for advanced mobility skills,” *International Journal of Robotics Research*, vol. 41, no. 9–10, pp. 903–924, Jun. 2022, doi: 10.1177/02783649221102473.

[22] K. H. Lee, O. Nachum, T. Zhang, S. Guadarrama, J. Tan, and W. Yu, “PI-ARS: Accelerating Evolution-Learned Visual-Locomotion with Predictive Information Representations,” in *IEEE International Conference on Intelligent Robots and Systems*, Oct. 2022, vol. 2022-October, pp. 1447–1454, doi: 10.1109/IROS47612.2022.9981952.

[23] A. Das, O. Naroditsky, Z. Zhu, S. Samarasekera, and R. Kumar, “Robust visual path following for heterogeneous mobile platforms,” in *Proceedings - IEEE International Conference on Robotics and Automation*, May 2010, pp. 2431–2437, doi: 10.1109/ROBOT.2010.5509699.

[24] M. Debord, W. Honig, and N. Ayanian, “Trajectory Planning for Heterogeneous Robot Teams,” in *IEEE International Conference on Intelligent Robots and Systems*, Oct. 2018, pp. 7924–7931, doi: 10.1109/IROS.2018.8593876.

[25] Z. G. Saribatur, V. Patoglu, and E. Erdem, “Finding optimal feasible global plans for multiple teams of heterogeneous robots using hybrid reasoning: an application to cognitive factories,” *Autonomous Robots*, vol. 43, no. 1, pp. 213–238, Mar. 2019, doi: 10.1007/s10514-018-9721-x.

[26] J. Collins, S. Chand, A. Vanderkop, and D. Howard, “A review of physics simulators for robotic applications,” *IEEE Access*, vol. 9, pp. 51416–51431, 2021, doi: 10.1109/ACCESS.2021.3068769.

[27] Y. Yu, Z. Miao, X. Wang, and L. Shen, “Cooperative circumnavigation control of multiple unicycle-type robots with non-identical input constraints,” *IET Control Theory and Applications*, vol. 16, no. 9, pp. 889–901, Mar. 2022, doi: 10.1049/cth2.12275.

[28] C. Calderón-Arce, J. C. Brenes-Torres, and R. Solis-Ortega, “Swarm Robotics: Simulators, Platforms and Applications Review,” *Computation*, vol. 10, no. 6, p. 80, May 2022, doi: 10.3390/computation10060080.

[29] M. Garzón *et al.*, “Using ROS in multi-robot systems: Experiences and lessons learned from real-world field tests,” in *Studies in Computational Intelligence*, vol. 707, Springer International Publishing, 2017, pp. 449–483.

[30] C. Pinciroli *et al.*, “ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems,” *Swarm Intelligence*, vol. 6, no. 4, pp. 271–295, Nov. 2012, doi: 10.1007/s11721-012-0072-5.

[31] M. U. Arif and S. Haider, “A Flexible Framework for Diverse Multi-Robot Task Allocation Scenarios Including Multi-Tasking,” *ACM Transactions on Autonomous and Adaptive Systems*, vol. 16, no. 1, pp. 1–23, Mar. 2022, doi: 10.1145/3502200.

[32] P. Fraisse, D. Andreu, R. Zapata, J. P. Richard, and T. Divoux, “Remote decentralized control strategy for cooperative mobile robots,” in *2004 8th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2004, vol. 2, pp. 1011–1016, doi: 10.1109/icarcv.2004.1468982.

[33] M. Dihya, M. Moufid, B. Chemseddine, and B. Moussaab, “Switched time delay control based on neural network for fault detection and compensation in robot,” *IAES International Journal of Robotics and Automation (IJRA)*, vol. 10, no. 2, p. 91, Jun. 2021, doi: 10.11591/ijra.v10i2.pp91-103.

[34] W. Łabuński and A. Burghardt, “Software for the Control and Monitoring of Work of a Collaborative Robot,” *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 2021, no. 3, pp. 29–36, May 2021, doi: 10.14313/JAMRIS/3-2021/16.

[35] Y.-C. Wu, J.-W. Lee, and H.-C. Wang, “Robots for search site monitoring, suspect guarding, and evidence identification,” *IAES International Journal of Robotics and Automation (IJRA)*, vol. 9, no. 2, p. 84, Jun. 2020, doi: 10.11591/ijra.v9i2.pp84-93.

[36] M. G. Ball, B. Qela, and S. Wesolkowski, “A review of the use of computational intelligence in the design of military surveillance networks,” in *Studies in Computational Intelligence*, vol. 621, Springer International Publishing, 2016, pp. 663–693.

[37] L. Joseph and J. Cacace, “Mastering ROS for robotics programming : design, build, and simulate complex robots using Robot Operating System,” 2015.

[38] M. Otte, M. J. Kuhlman, and D. Sofge, “Auctions for multi-robot task allocation in communication limited environments,” *Autonomous Robots*, vol. 44, no. 3–4, pp. 547–584, Jan. 2020, doi: 10.1007/s10514-019-09828-5.

[39] D. Tihanyi, Y. Lu, O. Karaca, and M. Kamgarpour, “Multi-robot task allocation for safe planning against stochastic hazard dynamics,” Jun. 2023, doi: 10.23919/ECC57647.2023.10178126.

BIOGRAPHIES OF AUTHORS

Abderrahmane Tamali received a master’s degree in Telecommunications Systems from Tahri Mohammed Bechar University. Currently pursuing a Ph.D. in Networks and Telecommunications at Ferhat Abbas Setif1 University, he excels in both substantive and group work environments. He possesses a diverse skill set, including programming, electronic application development, manipulation of Linux environments, development of robotic machines, utilization of ROS framework, and IoT applications. In addition to his academic achievements, Abderrahmene holds a senior IT technician diploma with a specialization in computer network maintenance. He is proficient in computer networks and is adept at troubleshooting and maintaining network systems. He can be contacted via email at abdou.t10@gmail.com or abderrahmane.tamali@univ-setif.dz.

Nourredine Amardjia received an engineering degree in electronics in 1982 from ENPA (Ecole Nationale Polytechnique d'Alger), Algeria, his Master of Science degree in electrical engineering in 1985, from Fairleigh Dickinson University, New Jersey, USA, and his State Doctorate (Ph.D.) in communications in 2007 from University of Setif, Algeria. He joined the Electronics Department, University of Setif as an assistant professor in 1986. He was granted the level of associate professor in 2007. He is a member of the LIS laboratory, the University of Setif. His research interests include discrete transforms, image processing, filter design techniques, systolic architectures, and fast algorithms for signal processing applications. He can be contacted at amardjianour@univ-setif.dz.

Mohammed Tamali graduated from USTO-MB as a state engineer in Electrical engineering, he received his M.Sc. in 1996 in Energetical physics from Bechar University and the Ph.D. degree from UST-MB of Oran, Algeria in 2007. In 2013, he became a professor in Electrical Engineering. He is head of the SimulIA research team at the ENERGARID Lab. His fields of interest included power electrical systems, scientific computing tools, sustainable development, environmental studies applied to distributed electrical network optimization, and system theory application on power systems. He worked as a research professor at the University of Bechar from 1986 until today. He can be contacted at tamali.mohammed@univ-bechar.dz.