

## Article

# Multirobot Formation with Sensor Fusion-Based Localization in Unknown Environment

Anh Vu Le <sup>1</sup>, Koppaka Ganesh Sai Apuroop <sup>1</sup>, Sriniketh Konduri <sup>1,2</sup>, Huy Do <sup>1</sup> , Mohan Rajesh Elara <sup>1</sup> ,  
Ray Cheng Chern Xi <sup>1</sup>, Raymond Yeong Wei Wen <sup>3</sup>, Minh Bui Vu <sup>4</sup> , Phan Van Duc <sup>5</sup> , Minh Tran <sup>6,\*</sup> 

- <sup>1</sup> ROAR Lab., Engineering Product Development, Singapore University of Technology and Design, Singapore 487372, Singapore; anhvu\_le@sutd.edu.sg (A.V.L.); ganesh\_koppaka@mymail.sutd.edu.sg (K.G.S.A.); sriniketh\_konduri@mymail.sutd.edu.sg (S.K.); huy\_do@sutd.edu.sg (H.D.); rajeshelara@sutd.edu.sg (M.R.E.); ray\_cheng@sutd.edu.sg (R.C.C.X.)
- <sup>2</sup> Electronics Engineering and Information Management, Chang Gung University, Taoyuan City 33302, Taiwan
- <sup>3</sup> Oceania Robotics, PTE, Ltd. #01-03, 3 Soon Lee Street, Singapore 627606, Singapore; raymond@oceaniarobotics.com
- <sup>4</sup> Faculty of Automotive, Mechanical, Electrical and Electronic Engineering, Nguyen Tat Thanh University, 300A-Nguyen Tat Thanh, Ward 13, District 4, Ho Chi Minh City 700000, Vietnam; bvminh@ntt.edu.vn
- <sup>5</sup> Faculty of Automobile Technology, Van Lang University, Ho Chi Minh City 700000, Vietnam; duc.pv@vlu.edu.vn
- <sup>6</sup> Optoelectronics Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam
- \* Correspondence: tranhoangquangminh@tdtu.edu.vn



**Citation:** Le, A.V.; Apuroop, K.G.S.; Konduri, S.; Do, H.; Elara, M.R.; Xi, R.C.C.; Wen, R.Y.W.; Vu, M.B.; Duc, P.V.; Tran, M. Multirobot Formation with Sensor Fusion-Based Localization in Unknown Environment. *Symmetry* **2021**, *13*, 1788. <https://doi.org/10.3390/sym13101788>

Academic Editor: Jan Awrejcewicz

Received: 29 July 2021

Accepted: 18 September 2021

Published: 26 September 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Abstract:** Multirobot cooperation enhancing the efficiency of numerous applications such as maintenance, rescue, inspection in cluttered unknown environments is the interesting topic recently. However, designing a formation strategy for multiple robots which enables the agents to follow the predefined master robot during navigation actions without a prebuilt map is challenging due to the uncertainties of self-localization and motion control. In this paper, we present a multirobot system to form the symmetrical patterns effectively within the unknown environment deployed randomly. To enable self-localization during group formatting, we propose the sensor fusion system leveraging sensor fusion from the ultrawideband-based positioning system, Inertial Measurement Unit orientation system, and wheel encoder to estimate robot locations precisely. Moreover, we propose a global path planning algorithm considering the kinematic of the robot's action inside the workspace as a metric space. Experiments are conducted on a set of robots called Falcon with a conventional four-wheel skid steering schematic as a case study to validate our proposed path planning technique. The outcome of our trials shows that the proposed approach produces exact robot locations after sensor fusion with the feasible formation tracking of multiple robots system on the simulated and real-world experiments.

**Keywords:** swarm robotics; sensor fusion; robot localization; path planning; robots symmetry formation

## 1. Introduction

Multirobot cooperation as the swarm system is one of the most discussed topics in the field of Robotics. Multirobot symmetry pattern formation helps robots to implement tasks together. The numerous applications can open up in terms of surveillance, inspection, maintenance, and rescue. Multirobot systems (MRS) include a group of multiple robots with the same characteristics operating in a shared, and symmetrically formed workspace [1]. This concept covers the examples of robotic arms, humanoid robots, ground and aerial mobile systems, and autonomous vehicles. The MRS term can be defined as multiagent systems, robotic swarms, or sensor networks in the literature.

Many robotic applications are focused on carrying out tasks in an unknown environment that would require different perception or sensing abilities. Even after the robot is

equipped with multiple arrays of sensors, in these environments, there can be sensory blind spots that might trigger a failure mode within the robot. If the robot does not have the contingency to identify this sensory blind spot in time, there might be an accident involving the robot causing injury to living things or structural damage to the surroundings and the robot itself. One of the swarm robot methodologies is primarily used in situations where the environment is unknown. Mapping an unknown environment is dangerous pertaining to the losses or damages that can cause the system's failure. To avoid this, the robots set out to map the unknown area with precision by still considering the dangers lurking in the forms of uneven terrains, glass walls, steep slopes, among others. The information gathered by all the swarm robots will help in learning collectively about the characteristics of the environment, thus preventing any damages to the robots or the environment. These swarm robots can be expanded to applications such as surveillance as mentioned by the authors in [2]. In the paper, a self-reconfigurable robot that can crawl, climb, and roll in any terrain was presented. This kind of robot assists in opening up new doors in the field of surveillance and inspection. Multiple robots of the same kind can be deployed into areas with a high degree of uncertainty.

These swarm robots usually are inspired by nature. Swarms of insects can be observed to understand swarming techniques, later adapted to different robotic scenarios. Some of such famous algorithms are ant colony optimization (ACO) [3], the bees algorithm (BA) [4], and glow-worm swarm optimization (GSO), among others. These swarm algorithms can be used for many applications, as mentioned earlier. One such interesting application is to search for a target object within a given unknown environment. The authors in the paper [5] proposed a decentralized control algorithm that enables the swarm robots to search for a given target. This method was inspired by bacteria chemotaxis. In order to search for a target, the robots have to visit every place in a given area. This can be rather termed as an area coverage problem. Numerous literature works addressed this area coverage issue using single robots [6,7], self-reconfigurable robots [8–10], and swarm robots [11]. Even with the area coverage problem, it is important to have a controller to enable the swarm robots to localize themselves relatively and move in a shape.

Research is being carried out in certain aspects, such as shape formation within the swarm structure. It will not be easy to enable all the swarming robots to form inside complex shapes without proper algorithms. The authors in [12] utilized a bioinspired morphology change approach that can be applied to the multirobot control algorithm to achieve the complex shape. In another example, the authors [13] proposed a control technique that enables a swarm of satellites to assemble in a hexagonal symmetrical lattice shape. There are several such examples [14–16] that showcases the control strategies that can be adapted to achieve different topology symmetrical shapes. Having different topology shapes is necessary in some situations. Let us consider the surveillance situation, where the intended application tracks a given object of interest in an unknown area. In this scenario, instead of sending the robots randomly in different directions, it will be safe to arrange the robots in a given defensive shape to perform the area coverage task. In this way, the risk to the robots from external threats would be reduced, and any knowledge gained through such collective learning can be easily applied to the whole group.

There is another important aspect of overcoming the challenge of control and coordination. In the cases of multirobot setups, the control and coordination strategies usually boil down to four categories: virtual structure algorithms, leader–follower algorithms, artificial potential field (APF) algorithms, and bioinspired algorithms. The virtual structure algorithms are the primitive algorithms that regard the robot as a single structure and establish all the algorithms that are originally developed for single robots [17–19]. In the second category, leader–follower algorithms consist of a leader, and the other robots follow the leader. Their locations are estimated relatively with each other. Here, most of the intelligent operations are performed on a single robot, the leader robot, and the other robots will follow it. There are several implementations done by various researchers for various applications [20–22]. The current paper utilizes the leader–follower approach

where the master robot runs the intelligent operations while the slave robots follow it. In the third category, APF algorithms are implemented for real-time path planning of the robots addressing the obstacle avoidance situations [23]. It is a distributed control model where the goal is considered to attract while the obstacle tends to repel. There are certain challenges with this as well, such as getting stuck in local minima. However, there are various literature works addressing this issue [24].

In the last set of algorithms, bioinspired algorithms are inspired by the flocking patterns of the insects. Many path planning strategies have been proposed based on this concept. The authors in [25] presented a cooperative and a distributive navigation strategy for a multirobot system. Even though with limited communication capabilities, the robots could find and decide the paths to reach the goal points efficiently. An artificial pheromone is utilized to mark the areas that are visited by the other robots. This is done to prevent revisiting the areas in order to find the path. Another group of authors in [26] mimicked biological neural systems to perform task assignments to swarm robots. This approach can handle dynamic path planning in 3D environments that are prone to uncertainty. The authors in [27] presented an algorithm that can produce an optimal path for multirobot groups. As proposed in the algorithm, the scout ants searched for the grid randomly without employing the pheromones. This algorithm utilizes two types of ants: one searching from the home position to the destination, while the other searching from the destination to the home position, thus, becoming into a bidirectional search. The grid places visited by these ants will be stored to avoid future repetition. Another group of authors in [28] proposed two extensions of the Particle Swarm Optimization (PSO) and Darwinian Particle Swarm Optimization (DPSO) as Robotic PSO (RPSO) and Robotic DPSO (RDPSO). These algorithms utilize bioinspired techniques while accounting for obstacle avoidance. They utilize social inclusion and exclusion concepts, analogous to reward–punish, to perform the estimation away from the local minima. The authors in the [29] presented an implementation of the previously mentioned Robotic Darwinian Particle Swarm Optimization (RDPSO) algorithm for a search and rescue application. They reiterated that this approach helps solve the local minimum problem by adopting a Darwinian approach suitable for search and rescue applications. There are also other applications where the evolutionary algorithms can be used.

The authors of [30] used the differential evolution to update the control parameters, without the need of a model, for multirotor applications. Their research was aimed to address the problems aroused in the flight performance due to the possible reasons of rotor hardware degradation and weight distribution due to extra payload among others. The evolution of the controllers based on the given requirement can replace the traditional approach of manually tuning the control parameters. The authors of another paper [31] proposed Hyb-CCEA. It is a cooperative coevolutionary algorithm that is meant to handle situations that involves heterogeneous multiagent systems. Their approach can also help to converge suitable team compositions, in the case of homogeneous, heterogeneous or hybrid teams. Their approach can be used in the domains where the possible outcomes are unknown. In another work [32], the authors analyzes the performance of the Brain Storm Optimization (BSO) and proposed an orthogonal learning framework to improve its learning mechanism. In regard to this, the authors proposed two Orthogonal Design (OD) engines that includes the exploration and exploitation. They are introduced to discover and utilize the relevant search experiences for performance improvements. The authors claim that the proposed approach can optimize complex functions.

With the increase in the complexity of the applications, the traditional algorithms might not be able to solve the optimization problem. The authors in [33] proposed an approach to solve the large-scale multiobjective and many-objective optimization problems. The approach involves the optimization of decision variables using the adaptive strategy by embedding the guidance of reference vectors in the control variable analysis. The authors claimed to have validated the efficiency of the proposed approach on large-scale multiobjectives and many-objective optimization problems. Reinforcement learning approaches

are also adopted to solve the optimization problems. The authors in the [34], took an approach that leverages on the multiagent reinforcement learning approaches. The authors proposed a dynamic correlation matrix based multiagent reinforcement learning approach. The approach results in updating the metaparameters using the evolutionary algorithms. They claim that this method not only enables the agents to learn from the local experiences but also from the experiences gained by the other agents. In similar lines, the authors from the paper [35] proposed an adaptive reference vector reinforcement learning approach to decomposition-based algorithms. The reinforcement learning handles the reference vector adaptation process, by leveraging on the environment feedback and choosing the optimal actions for convergence. Along with it, the reference point sampling utilizes the estimation-of-distribution learning models to sample new reference points.

The current paper utilizes a swarm of robots to address the problem of robot formation in unknown environments by low cost sensor fusion for robot self-localization and global path planning. Specifically, by following the master, multifollowers form symmetrical shapes as defined locations. The global path planning strategy enables the robots to reach the desired goal point while avoiding obstacles in their path. The master robot defines the path then leads the team of several robots to follow to desired locations. That framework reduces the complexity of the swarm system since the advance decision is processed at the master robot; the follower issues merely the middleware commands from motor driver to actuators. The system is also deployed as service bots where robots transport particular payloads from one place to another as instructed by the operator.

The main contributions of the paper are summarized as follows:

- The sensor fusion from the UWB position system, IMU, and wheel encoders for accurate localization in the unknown environment is validated on a low-cost and lightweight platform called the Falcon platform through a built-in GUI.
- A control algorithm as global path planning incorporating the skid steering kinematic based path tracking for multiple robots in the swarm system with a master and followers. To this end, the integrator-based dynamic linearization decouples the dynamics in the 2D coordinate, then given the designed controller for the leader robot, the formation control problem boils down to choosing the desired trajectory for the follower robots.
- Experiment results in both simulated and real-world experiments demonstrate that formation strategy for multiple robots by the proposed global path planning combined with the skid steering based control schematic.

The rest of the paper is organized as follows. Section 2 presents the context of application. Section 3 describes the system architecture. Section 4 presents a sensor system for autonomy navigation in the unknown environment. In Section 5, the proposed technique of mathematical model for multiagents control mechanism is given in detail. Section 6 presents the results and discussion on the proposed method's performance. Finally, Section 7 is conclusions.

## 2. System Requirement of Context of Application

The multirobot system needs to be deployed in outdoor conditions without the knowledge of workspace. The requirements of reliable and less expansive systems and easy deployment are desirable. In an emergency, these robotic systems can be deployed in no time with less intervention of humans at the target site to perform defined tasks such as inspections and carrying the load together. The system also requires these robots to move on uneven terrains while holding a payload of up to 1 kg and operate over  $30 \times 30 \text{ m}^2$  with precise localization up to 10 cm accuracy. Different symmetrical formations can be formed according to the user input to perform the task as desired. These kinds of systems can also be used as service bots where robots can transport a certain payload from one place to another as instructed by the operator.

### 3. System Architecture

As the requirement as described in the context of application, we have designed the swarm system. The swarm system consists of the operator control center installed in the workstation and two classes of robots named Falcon stationary and Falcon mobile platforms. The robot's mechanisms, electronics, and sensor components are shown in Figure 1. One is the stationary platform that carries a stationary beacon as shown in Figure 1a, and the other is a mobile platform that carries a mobile beacon as shown in Figure 1b. The system architecture with ROS [36] as the communication media for the main components, including an operator Console Unit, ROS master, the stationary platform, and the mobile platform, is shown in Figure 2. The stationery platforms provide the manual control mode for the operator to navigate them remotely to the desired location in the workspace to setup the workspace. The mobile platforms are set to form a group of up to ten units with one master and the others as followers inside the workspace to conduct the swarming tasks. ROS master is the central controller in which all information is received, processed, and sent. The operator's Console Unit is designed as a graphical user interface for the operator to command platforms.

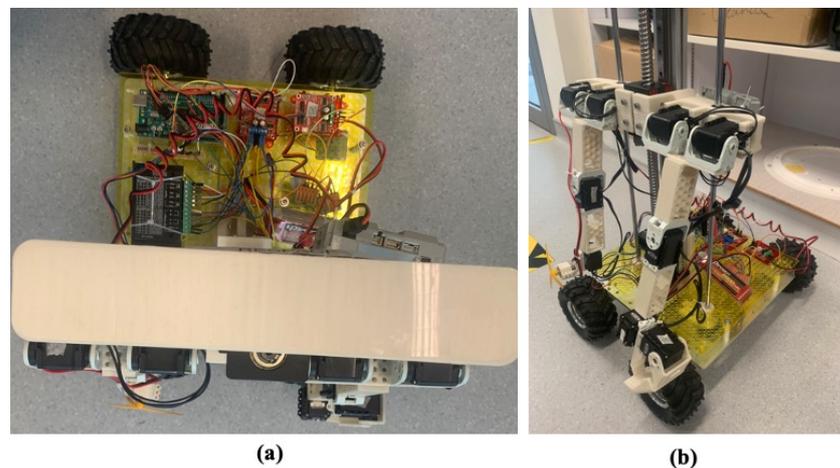


Figure 1. Falcon robots: (a) stationary platform, (b) mobile platform.

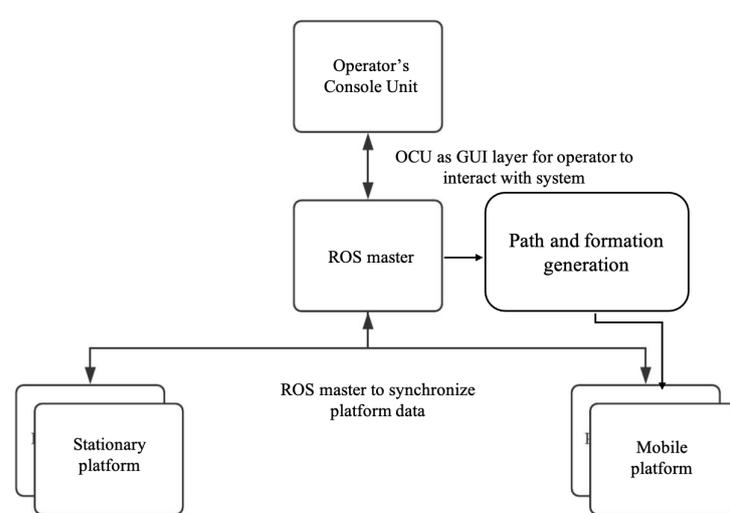


Figure 2. System architecture.

Falcon, a skid steering robot, carries lidar (only mobile beacon robot for obstacle avoidance) and IMU and UWB beacon to enable autonomous navigation with obstacle avoidance feature in the workspace. The dimensions of each robot are  $10 \times 40 \times 40$  cm, which weighs about 1.2 kg. The robots are 3D printed with a nylon material. They are

four-wheel mobile skid steering robots. All wheels are made of rubber of the same size with a diameter of 10 cm. A microprocessor is placed on each robot, which is responsible for the actions performed by the robot. The microprocessor used is the raspberry pi 4 model to which all the electronic sensors and motors are connected. 12 V DC motors are used to drive the wheels. RoboClaw 2 × 7 A motor controllers are used. The feedback loop is implemented in the control unit using motor encoders to robot wheel odometry. Pololu MinIMU-9 v5 9-degree-of-freedom absolute IMU is used for calculating the heading angle during its navigation. IMU is interfaced with raspberry pi over the I2C bus. A Low-cost RPLidar A1 sensor is mounted on the top of the front side of the robot. By using laser scan and project on a local costmap, the Lidar aids in the process of obstacle avoidance during the robot traversal toward the goal location. This lidar is efficient and has the capability to measure obstacles up to 12 m range. The robots see each other as an obstacle and try to avoid collision while reaching its goal. A Marvelmind UWB with stationary and mobile versions are shown in Figure 1 in which the beacons on top of the robots are connected to Raspberry Pi through a USB cable, which will enable the positioning of swarm robots in the 2D map. Among the mobile robots, the master robot is equipped with the Intel® Compute Stick, the advanced and lightweight CPU to program the planning tasks during the swarm actions.

#### 4. Sensor System for Autonomy Navigation in the Unknown Environment

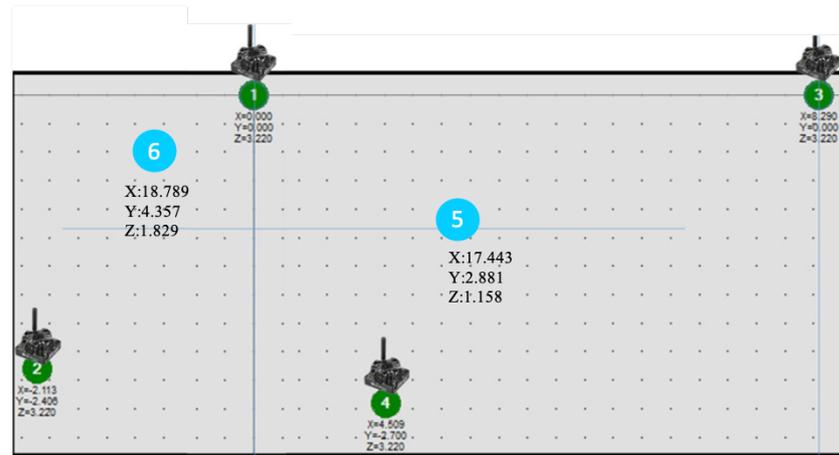
##### 4.1. Workspace Setup with the UWB Positioning System

This section illustrates the system behavior on how the robots are localized. Note that localization consists of the location and yaw orientation. Generally, without the built 2D or 3D map, the robots use special modules like GPS and Inertial measurement units to facilitate localization during navigation. However, GPS modules such as real-time kinematic (RTK)-based technology are expensive, bulky, and require an open space to get the accuracy. Therefore, precise localization is only possible with the high-performance GPS modules, which is a menial task to integrate on every robot in our proposed swarm system.

The robots of the swarm system get the 3D position  $(x, y, z)$  inside the deployed unknown-workspace using the UWB localization technique. UWB localization uses modules called static and mobile beacons to estimate the beacons mounted on the robots. The UWB beacon-based system provided a high precision of approximately  $\pm 2$  cm inside the workspace defined by stationary beacons within the diameter of 50 m. To guarantee precise localization, the mobile beacons are mounted on top of the robots to assume a line of sight condition with at least three stationary beacons.

A workspace for the robot to operate is formed by deploying manually stationary robots to corners of the workspace, which is defined according to the user's demands. At least three stationary beacons (generally with four beacons to ensure the redundancy backup in case of hardware failure as shown in the Figure 3) are needed in this method. They can be far away from each other up to 30 m to form a triangle with the line of sight condition. After setting up the workspace as the boundary of stationary beacons, each robot in this workspace carrying a mobile beacon facilitates finding the accurate position concerning the map formed by the previous beacons workspace setup.

Moreover, it is critical that the coordinate axis in which the robot is localized should have reference to the orientation system provided by the IMU coordinate axis so that both can be related in the robot operation. In the next section, we discuss a sensor fusion method to localize, orient, and navigate a robot in an unknown map formed by a precision positioning ultrawideband (UWB) beacon-based system incorporating orientation system IMU and wheel odometry encoders. All of the robots are operated using the ROS (robotic operating system) framework over WiFi or a 4G communication network. The position topics sent by the stationary robot are available for all the robots in the system.



**Figure 3.** Workspace setup within 4 stationary beacons. Green icons are the stationary beacons; the blue icons are the mobile beacons in the workspace.

#### 4.2. Initial Orientation Estimation

Since the UWB beacon system only provides the position  $(x, y, z)$  in the 3D coordinate and relative orientation by IMU, we developed the method to estimate the transformation between frames to find the robot's orientation in the beacon frame. The IMU data is not directly fused with the beacon data as their sensor values are in different reference frames. There is a need to find the transformation between frames to find the robot's orientation in the beacon frame. In order to localize the robot, a novel method called Initial Orientation Estimation is proposed. This method finds the initial orientation angle of the robot in the beacon frame, and then the real-time orientation in the global frame is updated depending on the robot's movement. This initial orientation estimation is the critical step for each Falcon in the robot workspace to localize itself properly.

The beacon coordinates axes' orientation with respect to the ground frame depends on how the stationary beacons are deployed in the open space. As the deployment of the stationary beacons is arbitrary, there is no reference to the ground frame formed by the absolute IMU cardinal directions. A relation between these coordinate axes should exist for localization and navigation. To find this, a method to find the initial docking orientation of robots in beacon coordinate axes had been developed purely based on the coordinates generated by the beacon system.

##### 4.2.1. Finding the Initial Orientation Angle

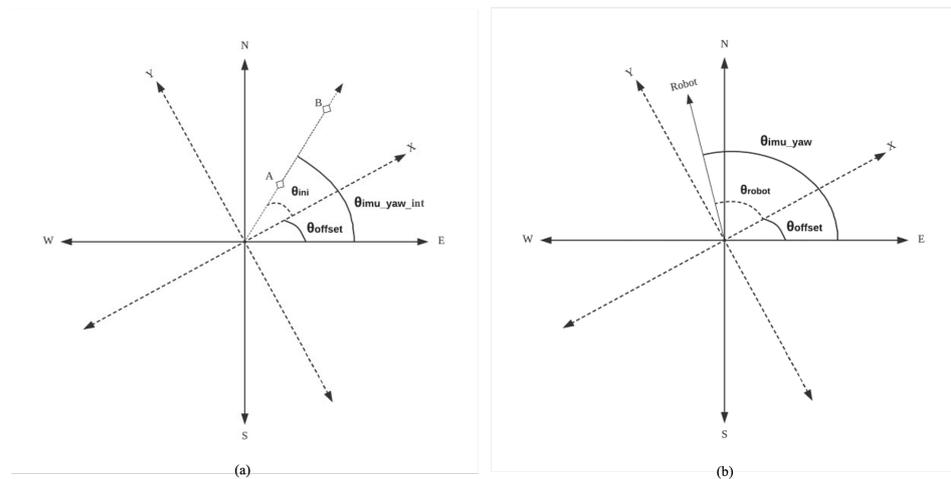
When deployed in the unknown environment, the stationary beacons are set randomly by the operator, i.e., Figure 3, the initial orientation angle is found solely using beacon generated coordinates with the help of primary slope and linear equations. After the system finishes booting up, the robot is initially moved forward for 3 s. During this process, the location as coordinate points, namely  $A$  and  $B$  generated by beacons, are recorded before and after robot forward motion. These two coordinate points are used in finding the robot heading in the beacon frame using the slope formed by two points  $A$  and  $B$ .

In Figure 4a the  $X, Y$  coordinate axes and cardinal directions represent the beacon frame and the IMU frame, respectively. However, these two frames may not be oriented in the same way, and let us consider that the beacon frame is  $\theta_{offset}$  times rotated about geographical axes.

Initially, considering the robot is moved forward from point  $A$  to  $B$ , let  $A$  is  $(x_1, y_1)$  and  $B$  is  $(x_2, y_2)$ , then slope of  $AB$  is given by Equation (1). The initial orientation angle  $\theta_{ini}$  of the robot with respect to the beacon frame can be derived as in Equation (2).

$$\tan(\theta_{ini}) = (y_2 - y_1) / (x_2 - x_1) \quad (1)$$

$$\theta_{ini} = \arctan((y_2 - y_1) / (x_2 - x_1)) \quad (2)$$



**Figure 4.** Orientation estimation by beacon positioning system and IMU system: (a) initial orientation angle, (b) heading angle of robot in beacon frame.

#### 4.2.2. Calculate the Offset Angle

Using the initial orientation angle and the yaw generated from the absolute IMU, the relation between the beacon frame and the IMU frame is calculated by finding the rotation angle between the frames. As denoted in the Figure 4a, from the the value of  $\theta_{imu\_yaw\_int}$  given by the IMU sensors real-time value,  $\theta_{offset}$  can be calculated by the Equation (3).

$$\theta_{offset} = \theta_{imu\_yaw\_int} - \theta_{ini} \quad (3)$$

#### 4.2.3. Heading Angle of Robot in Beacon Frame

After estimating the offset angle of the localization system, the robot's heading while the moving arbitrarily is calculated in the UWB beacon frame. Specifically, considering the arrow with robot index represents the orientation of the robot in the Figure 4b, the heading angle,  $\theta_{robot}$  is calculated as in Equation (4) by using the  $\theta_{offset}$  from the previous step. Where  $\theta_{robot}$  is the robot heading angle in the beacon frame,  $\theta_{imu\_yaw}$  is the robot heading angle given by the IMU sensor's real-time value.

$$\theta_{robot} = \theta_{imu\_yaw} - \theta_{offset} \quad (4)$$

#### 4.3. Sensor Fusion for Precise Localization

The IMU data is not directly fused with the beacon data as their sensor values are in different reference frames. There is a need to find the transformation between frames to find the robot's orientation in the beacon frame. To enhance the self-localization in the unknown environment we use the sensor fusion technique.

The swarm robots are randomly deployed in the unknown environment and do not use the sensor such as lidar to build the map. To enable autonomous navigation and swarm, the individual agent of this robot system should continuously be updated with the information of both its and the team members' locations consisting of the absolute orientation and position with respect to the reference global map.

The wheel odometry is fused with the yaw value of IMU to further refine the orientation of the robot odometry. The IMU data is not directly fused with the beacon data as their sensor values are in different reference frames. There is a need to find the transformation between frames to find the robot's orientation in the beacon frame. The raw IMU values as raw pitch yaw fields is added to robot localization package which uses EKF [37] to eliminate the sensor noise. To compensate for errors in the relative velocities between wheels which can affect robot trajectory, the feedback control is used to derive the left and right wheel velocities. The robot location, filtered by EKF, including the current position and heading in the global frame is the input for the control loop. The ROS based sensor

fusion flow is shown in Figure 5. This filtered odometry is then passed to the global and local path generators to execute the navigation in the UWB workspace.

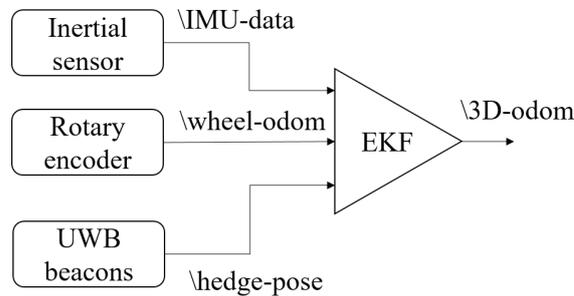


Figure 5. Robot localization by EKF sensor fusion.

### 5. Mathematical Model for Multiagents Control Mechanism

Using the proposed UWB and IMU-based self-localization system, the location of the developed robot in the global frame is well-defined. Based on the ROS network, the robot locations can be comprehended by the other members. To simplify the navigation task for multiple agents, we develop the master following in which all the slave robots will follow the one master path defined by the ROS move base package to maintain the required formation symmetrically. We consider a group of skid-steering four-wheel for our purpose to generate the appropriate motion command for robot.

Figure 6 represents the schematic of a conventional four wheel skid steering robot. The robot dynamics can be described by the robot coordinates  $p = [x_r, y_r, \beta]^T$  (planar position and steering). The variable  $\alpha$  denotes the angle between the COG ( $x_o, y_o$ ) and the wheels,  $L_1$  and  $L_2$  are the width and length of the robot. The relative position of the wheels are derived as in Equation (5):

$$\begin{aligned}
 W_{fi}(x, y) &= x_o + L\cos(\alpha + \beta), y_o + L\sin(\alpha + \beta) \\
 W_{fo}(x, y) &= x_o + L\cos(-\alpha + \beta), y_o + L\sin(-\alpha + \beta) \\
 W_{ri}(x, y) &= x_o + L\cos(\pi - \alpha + \beta), y_o + L\sin(\pi - \alpha + \beta) \\
 W_{ro}(x, y) &= x_o + L\cos(\pi + \alpha + \beta), y_o + L\sin(\pi + \alpha + \beta)
 \end{aligned} \tag{5}$$

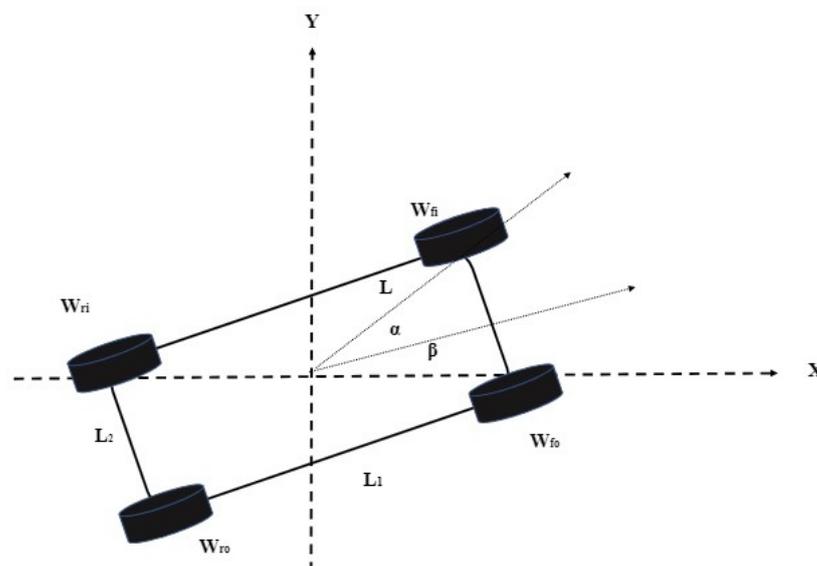


Figure 6. Schematic of the robot.

The robot is driven through reverse differential methodology, where the control inputs for the two inner and outer wheels are kept as same, i.e., as in Equation (6),

$$\begin{aligned}v_{in} &= v\left(1 - \frac{2L_2}{L_1}\tan(\beta)\right) \\v_{out} &= v\left(1 + \frac{2L_2}{L_1}\tan(\beta)\right) \\ \omega_{in} &= \omega\left(\frac{L_1}{L_1 - L_2\tan(\beta)}\right) \\ \omega_{out} &= \omega\left(\frac{L_1}{L_1 + L_2\tan(\beta)}\right)\end{aligned}\tag{6}$$

where  $v, \omega$  are the velocity and steering inputs for the COG. The kinematics for the COG can be expressed as the unicycle model, described as in Equation (7):

$$\dot{p} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \cos(\beta) & 0 \\ \sin(\beta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}\tag{7}$$

The kinematics for the  $x, y$  coordinate of the robot becomes as in Equation (8) :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos(\beta) & 0 \\ \sin(\beta) & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}\tag{8}$$

which is singular and not controllable. This singularity problem can be avoided by exploiting an integrator  $\zeta$  and modeled as in Equation (9):

$$\dot{\zeta} = f, \quad v = \zeta\tag{9}$$

where  $f$  is the linear acceleration. By differentiating again as in Equation (10), one obtains

$$\begin{bmatrix} \dot{\dot{x}} \\ \dot{\dot{y}} \end{bmatrix} = \begin{bmatrix} \cos(\beta) & -\zeta\sin(\beta) \\ \sin(\beta) & \zeta\cos(\beta) \end{bmatrix} \begin{bmatrix} f \\ \omega \end{bmatrix} = \Theta U\tag{10}$$

where the new input and the input matrix are given by  $U = [f, \omega] \in R^2$  and  $\Theta$  respectively. By using a control law as in Equation (11)

$$U = \Theta^{-1}(\zeta)\tag{11}$$

(10) is converted into a decoupled linear system as in Equation (12)

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \zeta\tag{12}$$

where  $\zeta = [\zeta_1, \zeta_2]^T$  can be chosen as the control input, and separate PID like feedback for  $\zeta_1, \zeta_2$  can be chosen for tracking any desired trajectory  $(x_d, y_d)$ .

The above control law assumes that the measurements for  $x, y, \beta$  are available and accurate. As this is not the case in many practical applications, the control law should deal with it.

### 5.1. Robust Control Law

Sometimes, different uncertainties such as inaccuracies in the measurement of  $\Theta$  or external disturbances creep into the mathematical model. The control law should be chosen to mitigate such issues. For (10), a control law can be chosen as in Equation (13):

$$U = \hat{\Theta}^{-1}(\zeta + \zeta_r)\tag{13}$$

where  $\zeta = [\zeta_1, \zeta_2]^T$  can be selected as two separate PD feedback, whereas  $\zeta_r = [\zeta_{r1}, \zeta_{r2}]^T$  represents the robust part. The uncertain linearized system can be expressed as in Equation (14):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \zeta_1 + d_x + \zeta_{r1} \\ \zeta_2 + d_y + \zeta_{r2} \end{bmatrix} \quad (14)$$

where the uncertainties are lumped in the terms  $d_x$  and  $d_y$ . For the design, assume the existence of two constants  $d_1, d_2 > 0$ , which satisfies conditions as in Equation (15)

$$\|d_x\| \leq d_1, \quad \|d_y\| \leq d_2. \quad (15)$$

we select two sliding surfaces designed as in Equation (16):

$$s_x = \dot{x} + k_1 e_x, \quad s_y = \dot{y} + k_2 e_y \quad (16)$$

where  $e_x = x - x_d$ ,  $e_y = y - y_d$ , and  $k_1, k_2 \in R^+$ . The velocity and acceleration are defined as in Equation (18), and Equation (17), respectively

$$V_x = \frac{1}{2} s_x^2 \quad (17)$$

$$\begin{aligned} \dot{V}_x &= s_x \dot{s}_x = s_x (\ddot{x} - \ddot{x}_d + k_1 \dot{e}_x) \\ &= s_x (\zeta_1 + \zeta_{r1} + d_x - \ddot{x}_d + k_1 \dot{e}_x). \end{aligned} \quad (18)$$

For  $\zeta_1 = \ddot{x}_d - k_1 \dot{e}_x - k_3 s_x$  ( $k_3$  is a positive scalar), the acceleration as in Equation (19),

$$\dot{V}_x = s_x (-k_2 s_x + \zeta_{r1} + d_x). \quad (19)$$

If  $\zeta_{r1} = -(1 + d_1) \tanh(s_x)$ , then the acceleration as as in Equation (20)

$$\begin{aligned} \dot{V}_x &\leq -k_3 s_x^2 - (1 + d_1) s_x \tanh(s_x) + d_1 \|s_x\| \\ \Rightarrow \dot{V}_x &\leq -k_3 s_x^2 - (1 + d_1) \|s_x\| + d_1 \|s_x\| < 0. \end{aligned} \quad (20)$$

From Lyapunov [38] direct method, it can be concluded that  $s_x \rightarrow 0$ , and can be presented as in Equation (21)

$$s_x \rightarrow 0 \Rightarrow \dot{e}_x = -k_1 e_x \Rightarrow x \rightarrow x_d. \quad (21)$$

The convergence of  $s_y$  can be inferred similarly. The overall control law for the robot is given by Equation (22)

$$\begin{aligned} \zeta_1 &= \ddot{x}_d - k_1 \dot{e}_x - k_3 s_x, \quad \zeta_{r1} = -(1 + d_1) \tanh(s_x) \\ \zeta_2 &= \ddot{y}_d - k_2 \dot{e}_y - k_4 s_y, \quad \zeta_{r2} = -(1 + d_2) \tanh(s_y) \\ \begin{bmatrix} f \\ \omega \end{bmatrix} &= \hat{\Theta}^{-1} \begin{bmatrix} \zeta_1 + \zeta_{r1} \\ \zeta_2 + \zeta_{r2} \end{bmatrix} \\ v &= \zeta. \end{aligned} \quad (22)$$

To avoid singularity, one can pick  $\zeta_1(o) = \sqrt{x(0)^2 + y(0)^2}$ , which prohibits  $v$  from crossing zero.

## 5.2. Leader Follower Formation Control

The integrator-based dynamic linearization decouples the dynamics in the  $x$ - and  $y$ -directions. Given that the controller for the leader robot is designed according to the method given in the previous section, and the formation control problem boils down to choosing the desired trajectory for the follower robots. Let us represent the leader position

at any instance as  $x_l(t), y_l(t)$ . For the follower robots, the desired trajectories  $(x_{dfi}, y_{dfi})$  can be chosen as in Equation (23)

$$x_{dfi} = x_l + c_{xfi}, y_{dfi} = y_l + c_{yfi} \quad (23)$$

where  $c_{xfi}, c_{yfi}$  are constant representing the desired separation in the respective coordinate. These constants can also be chosen as time varying quantities, which can assure time varying formation tracking. we define the error variables and the sliding surfaces for followers as in Equation (24):

$$e_{xfi} = x_{fi} - x_{dfi}, e_{yfi} = y_{fi} - y_{dfi}, s_{fxi} = \dot{e}_{xfi} + k_{f1i}e_{xfi}, s_{fyi} = \dot{e}_{yfi} + k_{f2i}e_{yfi} \quad (24)$$

The control law for the followers can be selected as as in Equation (25):

$$\begin{aligned} \zeta_{f1i} &= \ddot{x}_{dfi} - k_{f1i}\dot{e}_{xfi} - k_{f3i}s_{fxi}, \zeta_{r1fi} = -(1 + d_{1fi})\tanh(s_{fxi}) \\ \zeta_{f2i} &= \ddot{y}_{dfi} - k_{f2i}\dot{e}_{yfi} - k_{f4i}s_{fyi}, \zeta_{r2fi} = -(1 + d_{2fi})\tanh(s_{fyi}) \\ \begin{bmatrix} f_{fi} \\ \omega_{fi} \end{bmatrix} &= \Theta_{fi}^{-1} \begin{bmatrix} \zeta_{f1i} + \zeta_{r1fi} \\ \zeta_{f2i} + \zeta_{r2fi} \end{bmatrix} \\ v_{fi} &= \zeta_{fi}. \end{aligned} \quad (25)$$

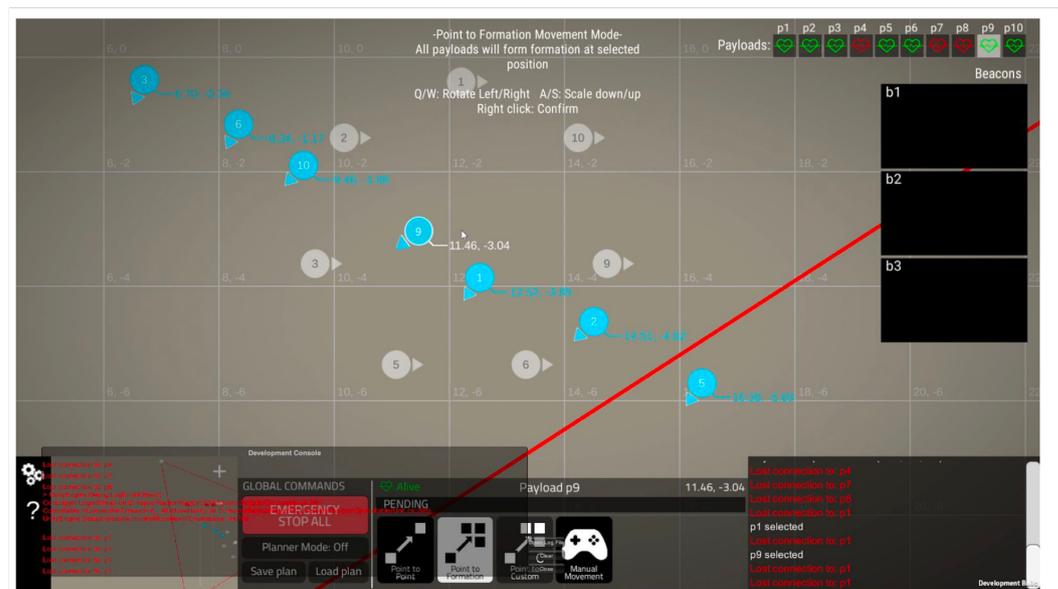
## 6. Experimental Results and Discussion

In this work, the sensor fusion for precise localization and the path tracking algorithms implemented to solve the navigation of multirobots path planning on the minimization of position tracking are validated. Furthermore, we evaluate the Falcons robot's performance in MATLAB and ROS simulation and validate the proposed method in real environment scenarios with group of Falcon robots as part of this research work.

### 6.1. Sensor Fusion for Localization Result

We design an operator's console unit (OCU) as a graphical interface as in Figure 7 to observe the robot location after fusing the sensor information the system, built using Unity 2019.4, a cross-platform engine for developing graphical programs. The OCU is required to be run on a workstation connected to the same network as the ROS master. On the first startup, the OCU will attempt to connect to the ROS master automatically. Upon connection failure, it will prompt the operator to enter the IP address of the ROS master and reconnect. Once connected, the system will enter the initialization phase and display red icons on the screen while connecting to stationary and mobile platforms is being established. Once connected, the system will enter the initialization phase where mobile platforms conduct the orientation estimation as described in Section 4.2 and present as blue icons with an arrow as in Figure 7. This allows the system to self-deploy payload platforms without further intervention from the operator from a single operator's command. As we can see, the robot icons show the stable location and orientations accuracy map to the setup of 8 robots with the straight line in the workspace.

Since the system relies on the accuracy of UWB outdoor localization, the localization can slightly be off in a short period. The location can jump back to the right location when the mobile robot carrying the mobile UWB beacon in the light of sight with at least 3 static UWB becomes fixed on the stationary robot. We have added this phenomenon in the discussion section. We have tested our system with current hardware configuration, the system can integrate 10 units and work simultaneously together. For stabilizing the working condition and preventing the lag of communication between robots, we realize that the system up to 8 robots is recommended.

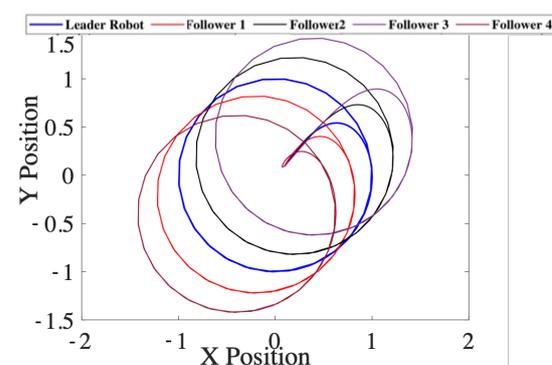


**Figure 7.** Robot location after sensor fusion in the OCU. Blue color icons are the real time robot locations. The gray color icons are the desired locations after formation.

The function of multirobot formation is achieved by a planner mode that allows the operator to preplan platform positions as shown in the gray icons and then execute the movement order when desired. The planner mode can be accessed by the button under Global Commands in the OCU or automatically from the load formation button if there is already a saved formation plan. When planner mode is active, planner units (shown in green) represent the objective of self-location as shown in Figure 7. The mean error of all robots after 5 deployments of the system is 0.114 m in location and 0.13 rad in orientation.

### 6.2. Simulation Results for Multirobot Path Tracking

To evaluate the controller in a multirobot model with one robot as master and other robots as followers, we deploy 5 Falcon robots to follow a predetermined circular and sinusoidal symmetry path. The master and follower controllers are programmed and embedded in the move-base function of the ROS operating system as a global path planner to generate velocity commands for the transmission systems added to the respective robot wheels. The simulation results are shown in Figure 8 for the circular symmetrical formation and Figure 9 for the sinusoidal symmetrical formation. We can see that the trajectories of the robots are displayed as expected.



**Figure 8.** Circular formation.

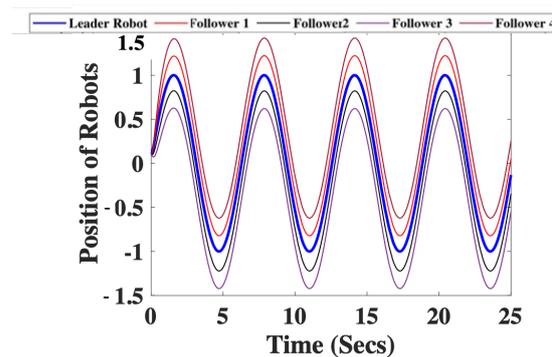


Figure 9. Sinusoidal formation.

### 6.3. Real Multi Robots Navigation in Real Environment

After setting up the numbers of a robot swarm in the system, based on the velocity value derived by the control mechanism for the multirobots, the control commands as the local planner of ROS move base are issued for motor drivers with the internal tuned PID values. Each robot advertises its location and identifies the other members inside the defined workspace of UWB stationary beacons by getting the locations of fusing multiple sensors with an initial orientation angle estimated when the system is booted up. The GUI assists the operator ensuring the Falcons understand the location of hardware uncertainty circumstances and sensor noise. The operator monitors the overall system by the GUI installed on the workstation. In the experiment, the best parameter values of path following are found using the experimental approach by trials. The robots are equipped with the main battery of 5 V 1000 mAh, and the maximum speed of the motor sets at 80 rpm with a gear ratio of 150. While robot navigating within the predefined time for completing the task, the real-time localization of robots are updated to the OCU, if the robot operation time exceeds this predefined time or the location offset of the robots are lower than the predefined value of 0.5 m, the system will declare robot in the complete task status. The robot in the complete task status will try to reduce the offset error if its navigation time does not exceed the time limit.

After setting up the workspace in an outdoor space with 4 stationary robot locations as shown in Figure 10, we deploy 5 robots marked as blue icons from 5 to 9. The robot locations in the straight line to circle in the symmetrical forms are observed in the GUI as presented in the testbed workspace of Figure 10a,b, respectively. We can observe that all the agents have arrived at the defined formation locations marked as yellow icons in the figure. There are slight misalignments of location for agents 7 and 8 since they are set near the boundary of UWB-workspace marked as the red line where the localization is not as stable as in the center of the center workspace.

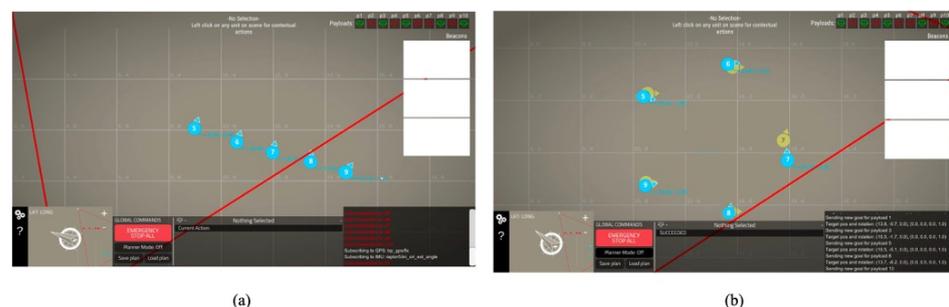
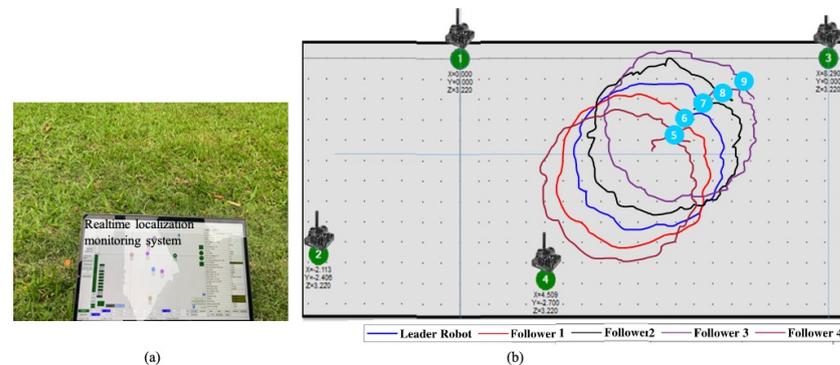


Figure 10. 5 units execute the line to circle formation: (a) before formation, (b) after formation.

The master robot is set to perform the navigation in the circular trajectory as represented as blue in the simulated result of Figure 8. As shown in the obtained results of Figure 11, the other robots perform path tracking of the master robot along the path to

form the symmetrical circles. We can see that the trajectories of the robots are displayed as expected. The multiple robot path tracking trials are validated to derive the possible error in 2D real environments.



**Figure 11.** Robots follow the master in a real testbed: (a) environment setup, (b) trajectories of robots.

The deviation and offset with goals in the  $x$  and  $y$  directions during 5 trials of the swarm robot system to follow the master robot with the symmetrical circular trajectory in the real workspace of Figure 11 is presented in Table 1. The average offset with the value of real tests and the value of simulation is about 10%. The error in the  $x$  direction is more significant than in the  $y$  direction. The derivation and the offset with the goal in the  $x$  direction are also more than the deviation and the offset with the goals in the  $y$  direction about 7% and 15%, respectively. The results are the causes of the skid steering control where each motion of the robot agent in the  $x$  direction is affected by the slippage. We can observe that the leader can track the circular trajectory in blue precisely. On the other hand, the other followers create minor vibrations while following the defined path to track the leader robot, but they still close the loop of circular trajectories represented in other colors. The errors are caused by the localization and the delay of instance linear velocity and angular velocity and motors PID driver of each robot. The result in Table 1 shows that the proposed solution can deliver comparable performance using only position feedback with error shown in Table 1. The results proved that the path follower controller with localization implemented in each robot in unknown workspaces is able to handle the sensor fusion errors.

**Table 1.** Numerical data of robots path following.

Trials	Deviation in X (m)	Deviation in Y (m)	Offset with Goal X (m)	Offset with Goal Y (m)
1	0.568	0.512	0.225	0.191
2	0.552	0.502	0.214	0.185
3	0.546	0.521	0.256	0.219
4	0.552	0.529	0.242	0.201
5	0.561	0.518	0.251	0.215
Average	0.554	0.516	0.238	0.202

The performance of the proposed output feedback controller is also compared with the performance of the same controller structure with full state measurement. We compare between our controller with the A\* global planner integrated with ROS for the skid steering multirobot to derive the efficiency in terms of time and energy the robot spent while robots follow the global path planner. To do it, the current sensor is attached serially to the robot battery and recorded during robot navigation. Table 2 presents the average energy and navigation time after 5 trials output for in real environment experiments. Despite better performance than following the route selected by random path, the A\* method consumes more time and energy than the proposed method about 11.12% and 5.31%, respectively.

Since more effort is needed to integrate the other advanced methods to our swarm systems, we added the discussion about validating the proposed method with these methods as a potential part of future work.

**Table 2.** Energy and time comparison.

Planners	Path Generation Time (s)	Execution Time (s)	Energy Spent (ws)
A*	0.58	181.38	38.20
Proposed method	0.51	161.21	36.17

## 7. Conclusions

A sensor fusion and output feedback controller were proposed for a swarm system with skid steering kinematic-based mobile robots. The proposed swarm system demonstrates the effective performance of the symmetrical pattern formations of path following within the unknown environment without a prebuilt map. The sensor fusion leveraging the UWB based positioning system, IMU, and wheel encoder enable the robots to navigate smoothly with the defined path tracking-based objectives. The experimental results with Falcon robots show the effectiveness of the proposed design in various tested environments with the feasible optimization path planning. In the present study, the proposed method considers only a swarm robot with unknown static environments.

**Author Contributions:** Conceptualization, A.V.L. and P.V.D.; data curation, S.K. and M.B.V.; formal analysis, A.V.L., K.G.S.A., R.C.C.X. and M.T.; project administration, M.R.E. and R.Y.W.W.; funding acquisition, M.R.E. and R.Y.W.W.; software, S.K., R.C.C.X., M.B.V., P.V.D. and M.T.; supervision, H.D. and M.R.E.; validation, H.D. and S.K.; visualization, K.G.S.A. and R.C.C.X.; writing—original draft, A.V.L., K.G.S.A. and H.D.; writing—review and editing, M.R.E., R.Y.W.W., M.B.V., P.V.D. and M.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is supported by the National Robotics Programme under its Robotics Enabling Capabilities and Technologies (Funding Agency Project No. 1922500051), National Robotics Programme under its Robot Domain Specific (Funding Agency Project No. 1922200058), National Robotics Programme under its Robotics Domain Specific (Funding Agency Project No. 1922200108), and administered by the Agency for Science, Technology and Research.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare that there is no conflict of interest.

## References

- Zakiev, A.; Tsoy, T.; Magid, E. Swarm robotics: Remarks on terminology and classification. In Proceedings of the International Conference on Interactive Collaborative Robotics, Leipzig, Germany, 18–22 September 2018; Springer: Berlin, Germany, 2018; pp. 291–300.
- Tan, N.; Mohan, R.E.; Elangovan, K. Scorpio: A biomimetic reconfigurable rolling–crawling robot. *Int. J. Adv. Robot. Syst.* **2016**, *13*, 1729881416658180.
- Dorigo, M.; Birattari, M.; Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39.
- Pham, D.T.; Ghanbarzadeh, A.; Koç, E.; Otri, S.; Rahim, S.; Zaidi, M. The bees algorithm—A novel tool for complex optimisation problems. In *Intelligent Production Machines and Systems*; Elsevier: Amsterdam, The Netherlands, 2006; pp. 454–459.
- Zheng, Z.; Tan, Y. Group explosion strategy for searching multiple targets using swarm robotic. In Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, 20–23 June 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 821–828.
- Le, A.V.; Veerajagadheswar, P.; Thiha Kyaw, P.; Elara, M.R. and Nhan, N.H.K. Coverage Path Planning Using Reinforcement Learning-Based TSP for hTetran—A Polyabolo-Inspired Self-Reconfigurable Tiling Robot *Sensors* **2021**, *21*, 2577.
- Le, A.V.; Parween, R.; Kyaw, P.T.; Mohan, R.E.; Minh, T.H.Q. and Borusu, C.S.C.S. Reinforcement learning-based energy-aware area coverage for reconfigurable hRombo tiling robot. *IEEE Access* **2020**, *8*, 209750–209761.

8. Le, A.V.; Parween, R.; Elara Mohan, R.; Khanh Nhan, N.H.; Enjikalayil, R. Optimization complete area coverage by reconfigurable hTrihex tiling robot. *Sensors* **2020**, *20*, 3170.
9. Le, A.V.; Prabakaran, V.; Sivanantham, V.; Mohan, R.E. Modified a-star algorithm for efficient coverage path planning in tetris inspired self-reconfigurable robot with integrated laser sensor. *Sensors* **2018**, *18*, 2585.
10. Apuroop, K.G.S.; Le, A.V.; Elara, M.R.; Sheu, B.J. Reinforcement Learning-Based Complete Area Coverage Path Planning for a Modified hTrihex Robot. *Sensors* **2021**, *21*, 1067.
11. Xiao, R.; Wu, H.; Hu, L.; Hu, J. A swarm intelligence labour division approach to solving complex area coverage problems of swarm robots. *Int. J. Bio-Inspired Comput.* **2020**, *15*, 224–238.
12. Meng, Y.; Guo, H.; Jin, Y. A morphogenetic approach to flexible and robust shape formation for swarm robotic systems. *Robot. Auton. Syst.* **2013**, *61*, 25–38.
13. Pinciroli, C.; Birattari, M.; Tuci, E.; Dorigo, M.; del Rey Zapatero, M.; Vinko, T.; Izzo, D. Self-organizing and scalable shape formation for a swarm of pico satellites. In Proceedings of the 2008 NASA/ESA Conference on Adaptive Hardware and Systems, Noordwijk, The Netherlands, 22–25 June 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 57–61.
14. Rubenstein, M.; Cornejo, A.; Nagpal, R. Programmable self-assembly in a thousand-robot swarm. *Science* **2014**, *345*, 795–799.
15. Sahin, E.; Labella, T.H.; Trianni, V.; Deneubourg, J.L.; Rasse, P.; Floreano, D.; Gambardella, L.; Mondada, F.; Nolfi, S.; Dorigo, M. SWARM-BOT: Pattern formation in a swarm of self-assembling mobile robots. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Hammamet, Tunisia, 6–9 October 2002; IEEE: Piscataway, NJ, USA, 2002; Volume 4, p. 6.
16. Jung, H.; Kim, D.H. Potential-function-based shape formation in swarm simulation. *Int. J. Control. Autom. Syst.* **2014**, *12*, 442–449.
17. Lewis, M.A.; Tan, K.H. High precision formation control of mobile robots using virtual structures. *Auton. Robot.* **1997**, *4*, 387–403.
18. Mehrjerdi, H.; Ghommam, J.; Saad, M. Nonlinear coordination control for a group of mobile robots using a virtual structure. *Mechatronics* **2011**, *21*, 1147–1155.
19. Li, X.R.; Li, B.; Zhang, P.; Zhang, J.Y.; Zhang, S. Large-scale unmanned aerial vehicle formation maintenance algorithm based on virtual structure. In Proceedings of the 2015 8th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 12–13 December 2015; IEEE: Piscataway, NJ, USA, 2015; Volume 1, pp. 417–421.
20. Edwards, D.; Bean, T.; Odell, D.; Anderson, M. A leader-follower algorithm for multiple AUV formations. In Proceedings of the 2004 IEEE/OES Autonomous Underwater Vehicles (IEEE Cat. No. 04CH37578), Sebasco, ME, USA, 17–18 June 2004; IEEE: Piscataway, NJ, USA, 2004; pp. 40–46.
21. Gu, D.; Wang, Z. Leader-follower flocking: Algorithms and experiments. *IEEE Trans. Control Syst. Technol.* **2009**, *17*, 1211–1219.
22. Yu, W.; Chen, G.; Cao, M. Distributed leader-follower flocking control for multi-agent dynamical systems with time-varying velocities. *Syst. Control Lett.* **2010**, *59*, 543–552.
23. Sutantyo, D.K.; Kernbach, S.; Levi, P.; Nepomnyashchikh, V.A. Multi-robot searching algorithm using Lévy flight and artificial potential field. In Proceedings of the 2010 IEEE Safety Security and Rescue Robotics, Bremen, Germany, 26–30 July 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 1–6.
24. Mabrouk, M.; McInnes, C. Solving the potential field local minimum problem using internal agent states. *Robot. Auton. Syst.* **2008**, *56*, 1050–1060.
25. de Almeida, J.P.L.S.; Nakashima, R.T.; Neves, F., Jr.; de Arruda, L.V.R. Bio-inspired on-line path planner for cooperative exploration of unknown environment by a Multi-Robot System. *Robot. Auton. Syst.* **2019**, *112*, 32–48.
26. Yi, X.; Zhu, A.; Yang, S.X.; Luo, C. A bio-inspired approach to task assignment of swarm robots in 3-D dynamic environments. *IEEE Trans. Cybern.* **2016**, *47*, 974–983.
27. Zhu, Q.; Hu, J.; Henschen, L. A new moving target interception algorithm for mobile robots based on sub-goal forecasting and an improved scout ant algorithm. *Appl. Soft Comput.* **2013**, *13*, 539–549.
28. Couceiro, M.S.; Rocha, R.P.; Ferreira, N.M. A novel multi-robot exploration approach based on particle swarm optimization algorithms. In Proceedings of the 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, Kyoto, Japan, 1–5 November 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 327–332.
29. Kumar, A.S.; Manikutty, G.; Bhavani, R.R.; Couceiro, M.S. Search and rescue operations using robotic darwinian particle swarm optimization. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udipi, India, 13–16 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1839–1843.
30. Howard, D. A platform that directly evolves multirotor controllers. *IEEE Trans. Evol. Comput.* **2017**, *21*, 943–955.
31. Gomes, J.; Mariano, P.; Christensen, A.L. Dynamic team heterogeneity in cooperative coevolutionary algorithms. *IEEE Trans. Evol. Comput.* **2017**, *22*, 934–948.
32. Ma, L.; Cheng, S.; Shi, Y. Enhancing learning efficiency of brain storm optimization via orthogonal learning design. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, doi:10.1109/TSMC.2020.2963943.
33. Ma, L.; Huang, M.; Yang, S.; Wang, R.; Wang, X. An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization. *IEEE Trans. Cybern.* **2021**, doi:10.1109/TCYB.2021.3086501.
34. Gan, X.; Guo, H.; Li, Z. A new multi-agent reinforcement learning method based on evolving dynamic correlation matrix. *IEEE Access* **2019**, *7*, 162127–162138.

35. Ma, L.; Li, N.; Guo, Y.; Wang, X.; Yang, S.; Huang, M.; Zhang, H. Learning to optimize: Reference vector reinforcement learning adaption to constrained many-objective optimization of industrial copper burdening system. *IEEE Trans. Cybern.* **2021**, doi:10.1109/TCYB.2021.3086501.
36. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009; Volume 3, p. 5.
37. Le, A.V., Veerajagadheswar, P., Kyaw, P.T., Muthugala, M.V.J., Elara, M.R., Kuma, M. and Nhan, N.H.K. Owards optimal hydro-blasting in reconfigurable climbing system for corroded ship hull cleaning and maintenance. *Expert Syst. Appl.* **2021**, *170*, 114519.
38. Li, Y.; Chen, Y.; Podlubny, I. Stability of fractional-order nonlinear dynamic systems: Lyapunov direct method and generalized Mittag–Leffler stability. *Comput. Math. Appl.* **2010**, *59*, 1810–1821.