Article

Experimental Investigation of Relative Localization Estimation in a Coordinated Formation Control of Low-Cost Underwater Drones

Thierry Soriano [1,*], Hoang Anh Pham [1,2] and Valentin Gies [2]

1 COSMER Laboratory, University of Toulon, 83041 Toulon, Francer
2 IM2NP Laboratory, University of Toulon, 83041 Toulon, France
* Correspondence: thierry.soriano@univ-tln.fr

Abstract: This study presents a relative localization estimation method for a group of low-cost underwater drones (l-UD), which only uses visual feedback provided by an on-board camera and IMU data. It aims to design a distributed controller for a group of robots to reach a specific shape. This controller is based on a leader–follower architecture. The main contribution is to determine the relative position between the l-UD without using digital communication and sonar positioning methods. In addition, the proposed implementation of the EKF to fuse the vision data and the IMU data improves the prediction capability in cases where the robot is out of view of the camera. This approach allows the study and testing of distributed control algorithms for low-cost underwater drones. Finally, three robot operating system (ROS) platform-based BlueROVs are used in an experiment in a near-realistic environment. The experimental validation of the approach has been obtained by investigating different scenarios.

Keywords: low-cost underwater drones; coordinated formation control; robot operating system; vision-based navigation; relative localization estimation; extended Kalman filter (EKF)

1. Introduction

In recent years, the systems and control community has been actively researching distributed coordination of various vehicles, such as unmanned aerial vehicles (UAV), unmanned ground vehicles (UGV), and unmanned underwater vehicles (UUV). In order to achieve a cooperative group performance, the distributed approach has several benefits, particularly with regard to low operational costs, fewer system requirements, high robustness, strong adaptivity, and flexible scalability. In general, distributed coordination research has three main problems: (i) distributed tracking control, (ii) navigation and localization, and (iii) the ability to deploy and test on real robots.

(i) Distributed tracking control: The objective of designing a distributed controller for a group of autonomous vehicles is to help them make decisions by themselves in a coordinated group task. Here, coordination refers to the close relationship between all vehicles in the group, where information sharing plays a central role. This past decade has seen intense research on distributed control of multi-robots under different aspects, such as consensus over switching network topology [1,2], consensus with delays [1,3,4], optimal consensus [5,6] sampled-data consensus, adaptive consensus [7], quantized consensus, second-order consensus [8], the consensus of generic linear agents [9,10], and consensus with multiple leaders [11]. Furthermore, under conditions of constrained radio waves, leading to limited information sharing (especially in underwater conditions), the design of distributed control algorithms has many challenges. Interested readers are referred to survey papers [12,13] for excellent reviews of the progress made before 2017 in the multi-agent coordination problem. In our previous research [14], we also demonstrated and simulated on ROS/Gazebo a complete control architecture for a group of AUVs that

can maintain formation, as well as avoid collisions and change the distance among them when crossing narrow areas.

(ii) Navigation and Localization: This is a challenging problem in the development of autonomous vehicles. Navigation is to guide the robot from one point to another. Localization is how well the drone localizes itself within a map or with another robot. In particular, the problem of underwater navigation and localization is even more challenging. In [15], a review of an AUV navigation and localization method was presented. These methods can be divided into three categories: inertial/dead reckoning, acoustic transponders and modems, and geophysical. Most recently, in [16], a robot swarm relative localization method was studied, and in [17], a review of localization, navigation, and communication of UUVs for collaborative missions is presented. In [18], the fusion of an inertial sensor and a vision is introduced. The inertial sensor has six degrees of freedom (6-DoF): 3-axis of an accelerometer and the 3-axis of a gyroscope. The goal is to determine a low-cost and accurate position for an autonomous mobile robot. In [19], the research is focused on the tracking of an AUV with three light beacons to transmit their IDs using a wide FOV camera on a leader AUV. If three light beacons are captured on the image, it allows for estimating the target AUV's pose with respect to the camera. The distance between the AUVs was estimated by using the geometry of the relative positions of the beacons on the image and the beacon's location on the target AUV. In [20], the relative localization of mobile robots based on multiple ultra-wideband ranging measurements was studied and applied to a group of ground mobile robots. In addition, there is a technique being researched and developed called SLAM (simultaneous localization and mapping). This technique is the process by which an autonomous robot constructs a map of its environment and simultaneously locates itself in this environment. Some of the most popular categories of SLAM approaches are EKF-SLAM [21,22], FastSLAM [23], and GraphSLAM [24]. However, these methods also require a high number of sensors with high costs as well as the high computing power of embedded systems.

(iii) Experimental and numerical investigation: Actual experiments still have many parameters that are not predefined by the simulation. In addition, there are limitations in both technology and cost-effectiveness, especially for autonomous underwater vehicles. In [25], the problem of pipeline following for AUVs was studied by using a monocular camera. In [26], a robust multi-robot convoying approach that relies on visual detection of the leading agent was presented, thus enabling target following in unstructured 3D environments. Moreover, an approach that uses vision and a convolutional neural network for an autonomous underwater vehicle to avoid collisions while observing objects was studied in [27]. In addition, the EC MORPH project [28,29] also studied formation control and tested it on real robot models which were equipped with an ultra-short baseline (USBL) and an underwater acoustic communication system.

Motivated by using low-cost underwater drones with low-cost sensors in the condition of limitation of communication and localization, this study:

- Proposed an effective prototype using four LEDs for the low-cost underwater drones to be able to determine the relative position between robots and an experimental evaluation of the algorithms for the low-cost underwater drones, which has not yet been investigated to the authors' knowledge.
- Proposed to use an EKF to estimate the position of the underwater vehicle follower-leader in case the robot is out of view of the camera. Compared with [30], the dynamic adaptive Kalman filter was used to navigate a group of AUVs in cases with and without GNSS. We implemented EKF for underwater vehicles and used only vision and IMU data. The use of cameras is low cost compared to positioning methods using sonar technology, which according to article [15], costs thousands to hundreds of thousands of dollars.
- Tested coordination control algorithms (i.e., formation control) in real environment conditions.

Figure 1 introduces the test scenario in this paper. In this case, we used three low-cost robots, in which it is assumed that a leader robot can be controlled to follow a given trajectory, and two follower robots automatically move to form a formation and follow the leader robot.
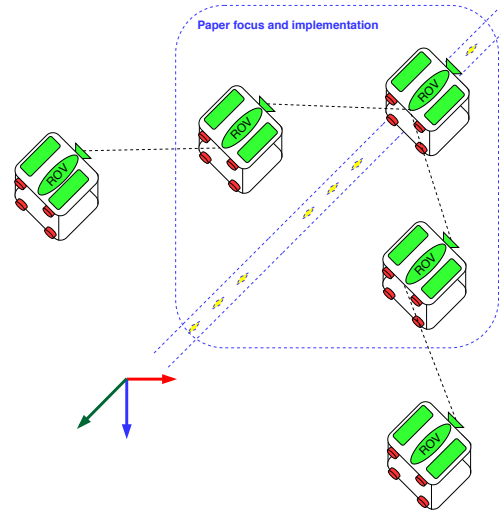
Figure 1. Proposal of the concept of a group of l-UDs.

2. Related Work

2.1. Low-Cost Underwater Drone Modeling

The equation of the dynamics of the low-cost underwater drone i can be remodeled into a state space model of the form:

$$\dot{\mathbf{x}}_i = A_i\mathbf{x}_i + B_i[\tau_{hi} + \mathbf{f}_i(\mathbf{x}_i) + \mathbf{w}_i(t)] \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^6$ is the state vector of the robot i;
$\mathbf{f}_i(\mathbf{x}_i) \in \mathbb{R}^3$ is the unknown uncertainty of the robot;
$\mathbf{w}_i(t) \in \mathbb{R}^3$ represents the corresponding unknown disturbance;
$A_i \in \mathbb{R}^{6\times6}$ and $B_i \in \mathbb{R}^{6\times3}$ are known matrices and (A_i, B_i) is assumed to be stabilizable.

According to [31], a local feedback controller is chosen:

$$\tau_{hi} = K_r\mathbf{x}_i + \mathbf{u}_i \tag{2}$$

so that $A + BK_r$ is to *Hurwitz*. $\mathbf{u}_i \in \mathbb{R}^3$ is the control input.

2.2. Formation Tracking Control

We have chosen the leader and follower architecture to implement the formation tracking control for underwater vehicles with low-cost criteria. In this case, the model of the leader robot is as follows:

$$\dot{\mathbf{x}}_L = A_L\mathbf{x}_L + B_L r(\mathbf{x}_L, t) \tag{3}$$

where $\mathbf{x}_L \in \mathbb{R}^6$ represents the state of the reference robot and $r(\mathbf{x}_L, t) \in \mathbb{R}^3$ represents a reference input. $A_L \in \mathbb{R}^{6\times6}$ and $B_L \in \mathbb{R}^{6\times3}$ are known matrices of the reference robot. The reference model can be considered as a virtual leader in the group, and it generates a desired trajectory for the whole group to follow.

To control the formation, we defined an auxiliary variable vector, which is a relative position tracking error $\mathbf{e}_i \in \mathbb{R}^6$ for l-ROV i, as

$$\mathbf{e}_i = \sum_{j\in N_i} a_{ij}\big(\mathbf{x}_i - \delta_i - (\mathbf{x}_j - \delta_j)\big) + a_{iL}(\mathbf{x}_i - \delta_i - \mathbf{x}_L) \tag{4}$$

where $\delta_i - \delta_j \forall i \neq j$, indicates the desired formation between robots (see Figure 2), a_{ij} and a_{iL} demonstrate connectivity between robots, and $\mathbf{x}_L \in \mathbb{R}^6$ represents the state of the reference robot, which is defined in Equation (3).
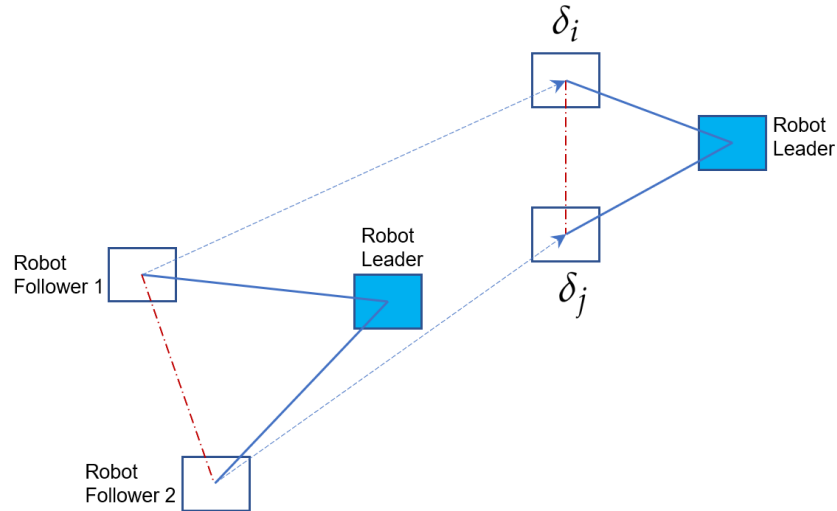
Figure 2. Illustration of a robot moving to make a formation. The dashed red line indicates that there is no information sharing between the two follower robots.

A distributed adaptive training tracking control was then designed based on the consensus tracking protocol and the feedback linearization technique, which is given by:

$$u_{iFC} = c_i K \mathbf{e}_i \tag{5}$$

where $K \in \mathbb{R}^{3 \times 6}$ is the return control gain and $c_i \in \mathbb{R}$ is the updated weight. Readers can find the details about formation tracking control in our previous research [14,32].

In the following experiment, we are concerned with determining the relative position between the follower and the leader robot because each robot can only be equipped with one camera. It is therefore able to detect the objects in front of it. This also leads to a limitation in that the developed control algorithms can guarantee only the constant distance between the follower robots and the leader robot but cannot guarantee the distance between the follower robots. However, this might also be improved if the robots are equipped with four cameras or a 360-degree camera.

2.3. Vision-Based Pose Estimation

In this section, we assume that a 3D model of the scene is available and can be estimated online. The position must be estimated by knowing the correspondences between the 2D measurements in the images and the 3D features of the model (see Figure 3).
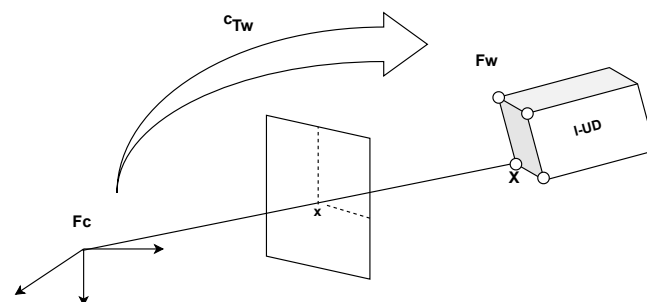
Figure 3. Rigid transformation $^c\mathbf{T}_w$ between the world image, F_w, and the camera image, F_c, and the projection in perspective.

Let us denote F_c as the camera frame and $^c\mathbf{T}_w$ as the transformation that defines the position of F_w with respect to F_c.

$$^c\mathbf{T}_w = \begin{pmatrix} ^c\mathbf{R}_w & ^c\mathbf{t}_w \\ \mathbf{0}_{3\times1} & 1 \end{pmatrix} \tag{6}$$

where $^c\mathbf{R}_w$ and $^c\mathbf{t}_w$ are the rotation matrix and the translation vector defined by the camera position in a global reference frame, respectively.

The projection in perspective, $\bar{\mathbf{x}}_v = (u, v, 1)^\top$, from a point $^w\mathbf{X} = (^wX, {}^wY, {}^wZ, 1)^\top$ is given by

$$\bar{\mathbf{x}}_v = \mathbf{K}\Pi^c\mathbf{T}_w{}^w\mathbf{X} \tag{7}$$

where $\bar{\mathbf{x}}_v$ is the perspective projection of a point $\mathbf{X}$ and $\mathbf{K}$ is the matrix of intrinsic parameters of the camera.

These parameters can be obtained by an offline calibration step. In addition,

$$\Pi = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{8}$$

We consider the image coordinates expressed in the normalized metric space

$$\mathbf{x}_v = \mathbf{K}^{-1}\bar{\mathbf{x}}_v \tag{9}$$

If we have N points $^w\mathbf{X}_i, i = 1..N$ whose coordinates are expressed in F_w are given by $^w\mathbf{X}_i = (^wX_i, {}^wY_i, {}^wZ_i, 1)^\top$, the projection $\bar{\mathbf{x}}_{vi} = (x_{vi}, y_{vi}, 1)^\top$ of these points in the image plane is then given by:

$$\mathbf{x}_{vi} = \Pi^c\mathbf{T}_w{}^w\mathbf{X}_i \tag{10}$$

If we know the 2D–3D point correspondences, $\mathbf{x}_{vi}$, and $^w\mathbf{X}_i$, the pose estimation $^c\mathbf{T}_w$ is the solution of Equation (10). This inverse problem is known as the N-point perspective problem or PnP problem (*Perspective-n-point*). The reader can find details in [33,34].

2.4. Theory of Extended KALMAN Filter

A non-linear dynamic system can be described as:

$$\mathbf{x}_n = \mathbf{f}_{kf}(\mathbf{x}_{n-1}) + \mathbf{w}_{n-1} \tag{11}$$

where $\mathbf{x}_n$ is the robot's system state (i.e., the 3D pose) at time n, $\mathbf{f}_{kf}$ is a non-linear state transition function, and $\mathbf{w}_{n-1}$ is the noise of the process, which is assumed to be normally distributed. In addition, the form of the measures is

$$\mathbf{z}_n = h(\mathbf{x}_n) + \mathbf{v}_n \tag{12}$$

The Kalman filter works in a predictive loop. Once initialized, the Kalman filter must predict the state of the system at the next time step. In addition, the Kalman filter provides the uncertainty of the prediction.

$$\hat{\mathbf{x}}_{n+1,n} = F\hat{\mathbf{x}}_{n,n} + G\hat{\mathbf{u}}_{n,n} \tag{13}$$

$$P_{n+1,n} = FP_{n,n}F^T + Q \tag{14}$$

Once the measurement is obtained, the Kalman filter updates (or corrects the prediction) the uncertainty of the current state. In addition, the Kalman filter predicts the next states, and so on.

$$K_n = P_{n,n-1}H^\top \left(HP_{n,n-1}H^\top + R_n\right)^{-1} \tag{15}$$

$$\hat{\mathbf{x}}_{n,n} = \hat{\mathbf{x}}_{n,n-1} + K_n(\mathbf{z}_n - H\hat{\mathbf{x}}_{n,n-1}) \tag{16}$$

$$P_{n,n} = (I - K_n H)P_{n,n-1}(I - K_n H)^\top + K_n R_n K_n^\top \tag{17}$$

Details of the Kalman filter variables can be found in Table 1.

Table 1. The standard notation used for the extended Kalman filter.

Notation	Name	Dimensions
$\mathbf{x}$	State vector	$n_x \times 1$
$\mathbf{z}$	Output vector	$n_z \times 1$
F	State transition matrix	$n_x \times n_x$
$\mathbf{u}$	Input variable	$n_u \times 1$
G	Matrix control	$n_x \times n_u$
P	Estimation of uncertainty	$n_x \times n_x$
Q	Uncertainty on process noise	$n_x \times n_x$
R	Uncertainty of measurements	$n_z \times n_z$
$\mathbf{w}$	Process noise vector	$n_x \times 1$
$\mathbf{v}$	Measurement noise vector	$n_z \times 1$
H	Observation matrix	$n_z \times n_x$
K	Kalman gain	$n_x \times n_z$
n	Discrete time index	–

Using EKF theory and the vision-based estimation presented above, combined with our previous theoretical studies [14], we introduce the implementation of relative location estimation for real robots in the next section.

3. Implement Relative Positioning for l-UD Followers

In previous work, we introduced a control architecture [14]. It consists of four components, which are the formation tracking control input, u_{iFC}, the robust control input, u_{iR}, the neural network control input, u_{iNN}, and the collision avoidance, u_{iCA} (see Figure 4). We evaluated and validated these proposals on the Gazebo simulator, assuming that the underwater robots can localize and share information with other underwater robots. However, this is challenging due to the limitation of underwater radio transmission in the real environment. Therefore, we introduce an approach that uses a camera to determine the relative position between the follower and the leader robot to experiment with the complete process in this section.
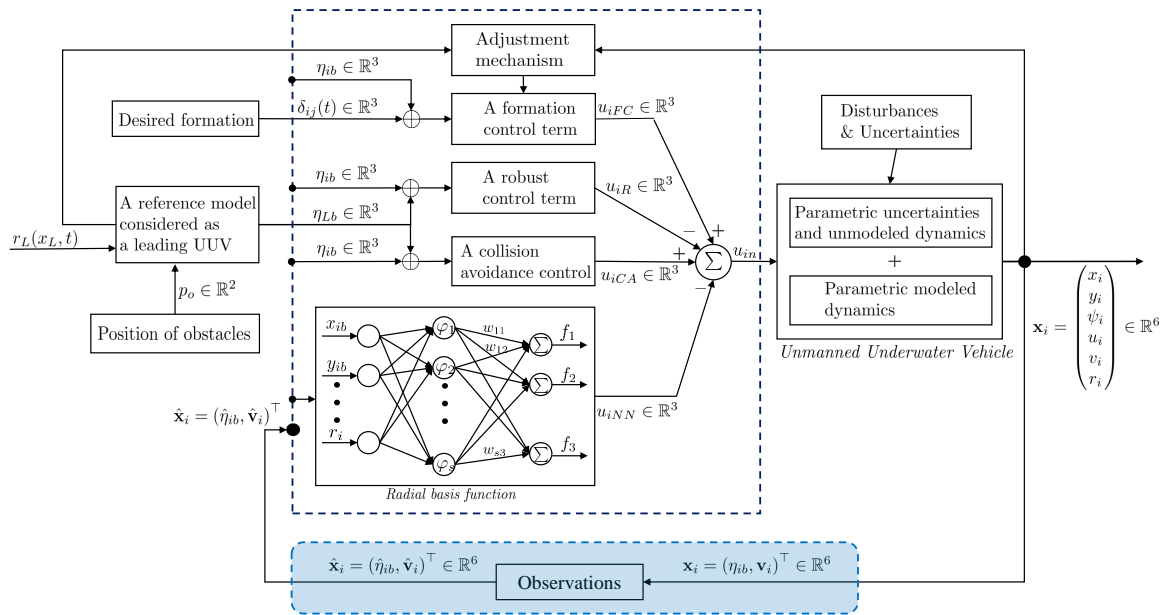
Figure 4. A complete diagram of the control and observation components of the low-cost underwater drone *i*.

This study is equivalent to the third round of the spiral model-based research methodology (see Figure 5). The first circle studies the simple simulation model and the second circle studies the complete simulation model, which was introduced in the previous study [32]. This third circle focuses on researching and testing control algorithms for low-cost underwater UAV prototypes.
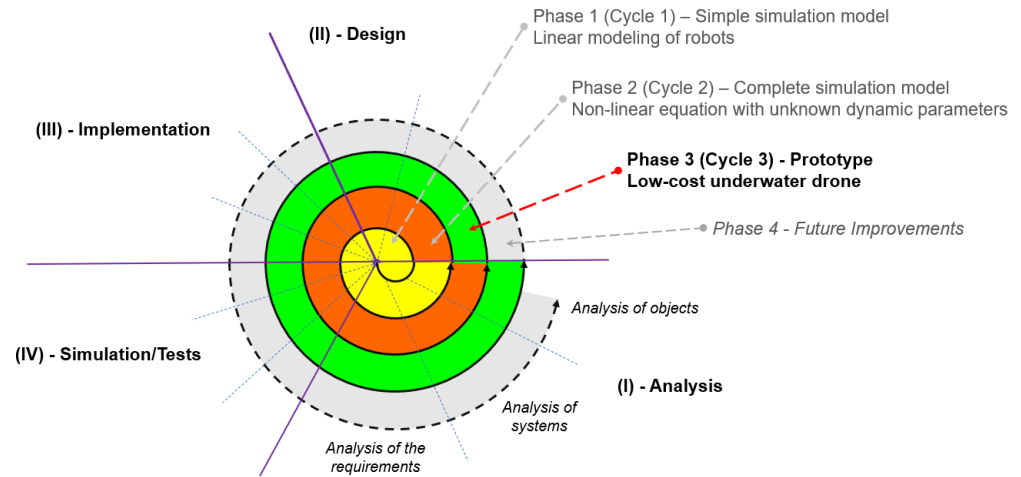

Figure 5. Iterative research on low-cost underwater robot systems.

Inspired by previous works [33,35] for pose estimation in computer vision, we propose the prototype of l-UD, which already has an HD USB camera with low-light performance and low-cost sensors (IMU). We then equipped the leader l-UD with four high intensity LEDs (see Figure 6). This equipment is built on the low-cost robot platform BlueROV [36], which is open source to allow for extensive customization.
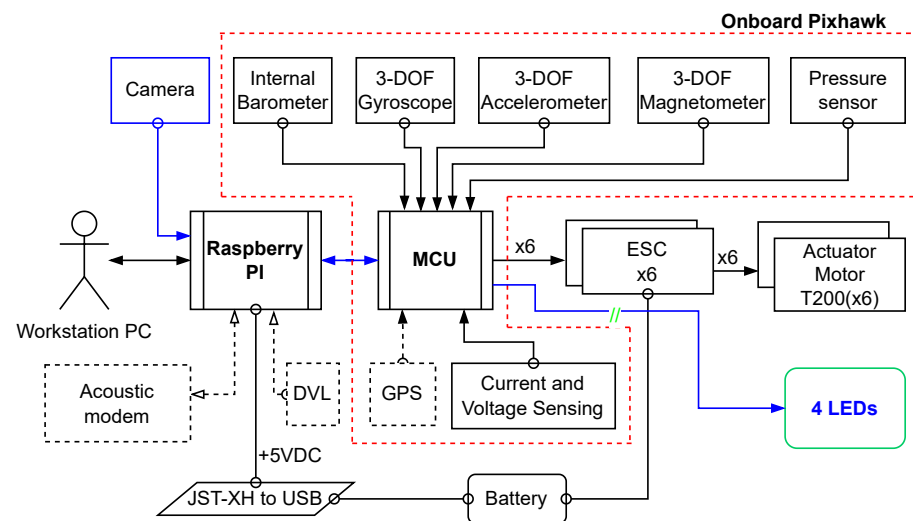
Figure 6. Architecture of the electronic components of the BlueROV underwater robot with the addition of four LEDs.

Thanks to this equipment, the camera mounted on the follower l-UD is able to determine the leader l-UD in the underwater environment under low light conditions. In addition, we also placed these LEDs at a 10 cm depth inside plastic tubes to avoid light dispersion in the underwater environment (see Figure 7).
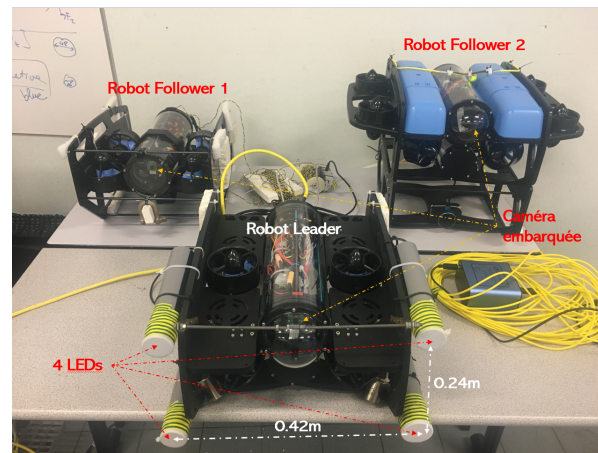

Figure 7. Three l-UD BlueROVs for experimentation.

More precisely, the camera of the following robot detects four LEDs placed on the robot leader and its position. Then, it determines the relative position of the camera with respect to the center of gravity of these four LEDs. This also means determining the relative position between the follower robot and the leader robot (see Figure 8).

Assumption 1. *The assumption is that the speed of the robot leader is predefined and is within a given range.*

Assumption 2. *The loss of position between the robot follower and the robot leader only occurs for a limited time, $\triangle T$.*

By combining the relative position obtained by the camera with the signal measured by the sensor via the EKF filter, we obtain the relative position between the l-UD follower and the leader. The details on the use of the EKF are presented in Figure 9.
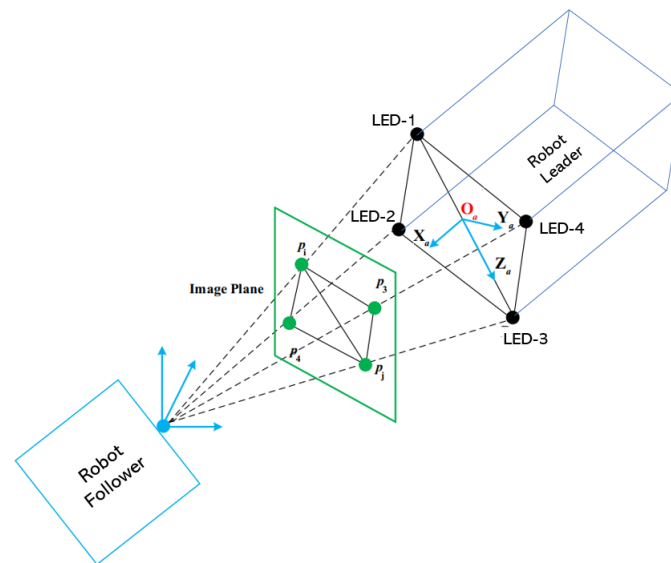
**Figure 8.** Determination of the relative position of four points using the camera.
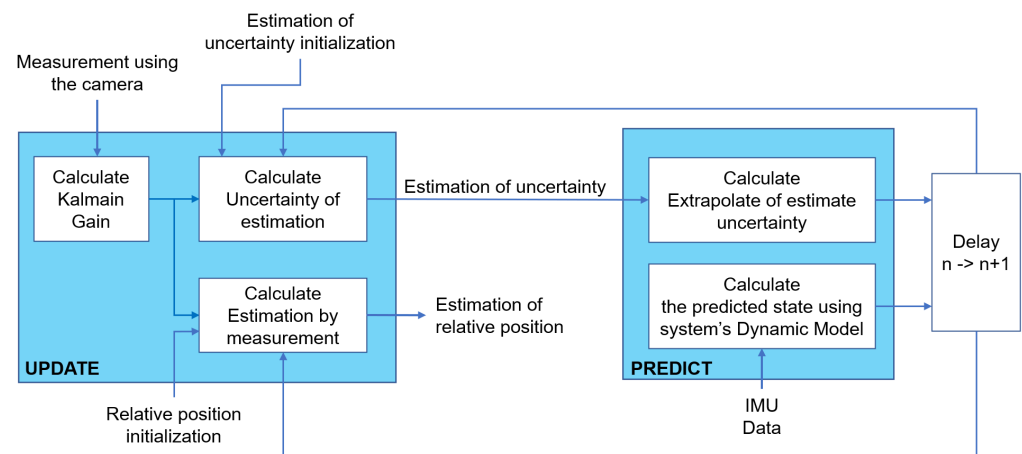
Figure 9. Architectural diagram of the process of fusing camera and IMU data using EKF.

The limitation of using low-cost cameras and underwater testing conditions may result in the camera being unable to detect all four LEDs continuously. As a consequence, the EKF can be used to predict the relative position between the follower robot and the leader for a short period of time. In addition, compared to machine learning algorithms, implementing EKF also allows us to take advantage of using an embedded system with limited computing power.

4. Experiment Setup

4.1. Low-Cost Underwater Drone BlueROV with LED

To perform the experiment, we used the Lumen Subsea Light, which is a blindingly bright LED light for use on ROVs and AUVs. The light outputs over 1500 lumens at 15 Watts and has a 135 degree beam angle for wide illumination in front of an ROV. Moreover, this LED has a fully dimmable control using a PWM servo signal and simple on–off control with no signal needed. In addition, while testing, we found that underwater light scattering can reduce the detection efficiency of the camera. Therefore, we also placed these LEDs in plastic tubes (with a length of 30 cm) to help the lights concentrate on one point (see Figure 10).
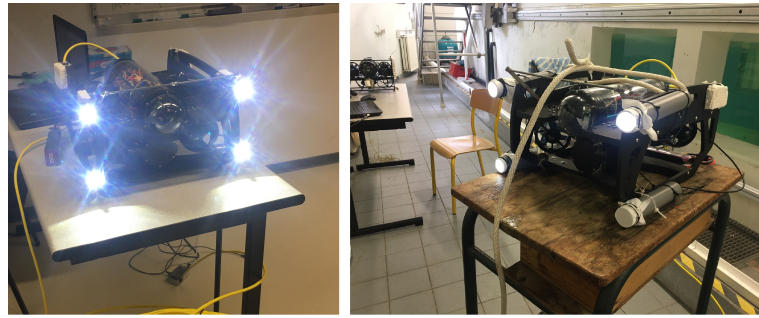
Figure 10. Four LEDs equipped on the follower robot.

We carried out the tests in the experimental pool of the University of Toulon. The dimensions are length × width × depth = 10 m × 3 m × 1.5 m (see Figure 11). The objective of these experiments was to validate the algorithms that have been developed and simulated on the Gazebo simulator, which was presented in our previous study [14]. The successful testing of these algorithms on the Gazebo simulator assumes that the relative distances between the robots can be obtained through different positioning methods. However, in the actual test conditions in the laboratory, the use of the camera positioning method was the most feasible. These experiments also demonstrated that the algorithms can operate in a near-realistic environment (i.e., a fully open marine environment). The methods limitations are the camera's ability to detect objects and the camera detecting the wrong object when the robot is very close to the water surface due to the phenomenon of light reflection.
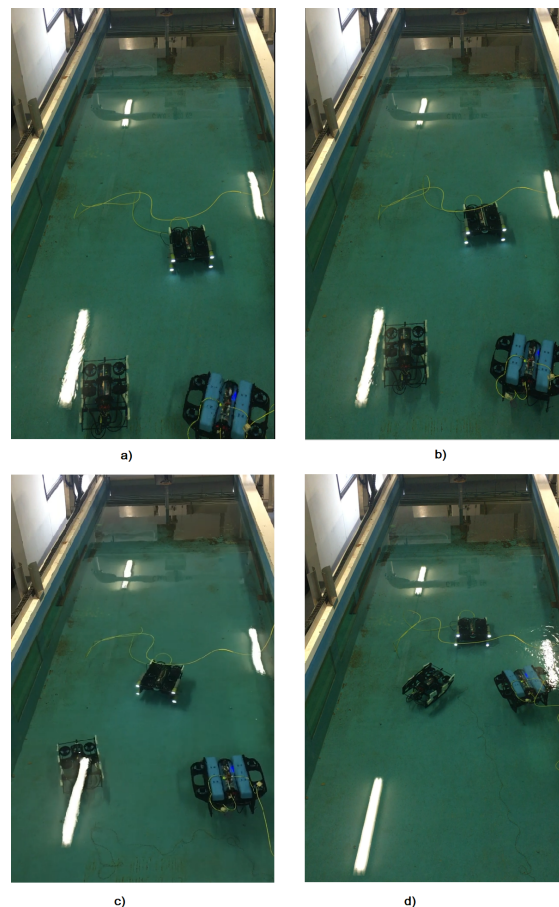
Figure 11. Testing of three robots forming a formation and moving according to the leading robot. (**a**) description of the initial positions of the robots. (**b**,**c**) illustration of robot group movement. (**d**) indicating the formation of the robots.

Figure 7 shows three l-UD BlueROVs, including one leader robot equipped with four LEDs and two follower robots equipped with cameras. The cable visible in the image is only for monitoring purposes, but there is no communication between robots. The size of the four LEDs equipped on the leader of the robot is length × width = 0.42 m × 0.24 m.
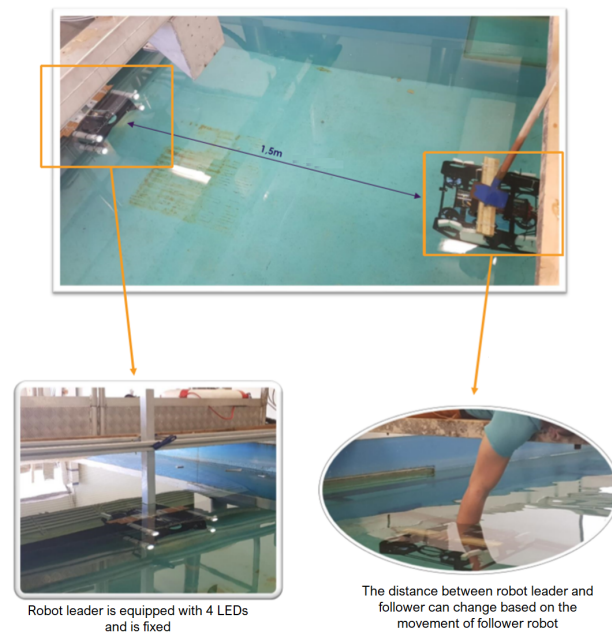
4.2. Determination of the Function of the Real Distance and the Distance Measured by the Camera

Firstly, we would like to determine the actual distance and the distance measured with the camera between the two robots. For this purpose, we placed two robots at different fixed distances from 1.5 m to 8.5 m, respectively (see Figure 12). The distance of 8.5 m was the maximum distance at which the camera is capable of detecting the four LEDs in our test environment. Simultaneously, we also determined the distance measured by the camera.

By linearizing the obtained values (see Figure 13), we obtained a first-order function between the actual distance and the distance obtained from the camera:

$$y = 20.207x + 0.2249 \tag{18}$$

where x is the distance measured by the camera and y is the actual distance. The objective of function (18) is to help correct between the real and measured distances.

Figure 12. Determination of the real distance between two robots by using a camera.
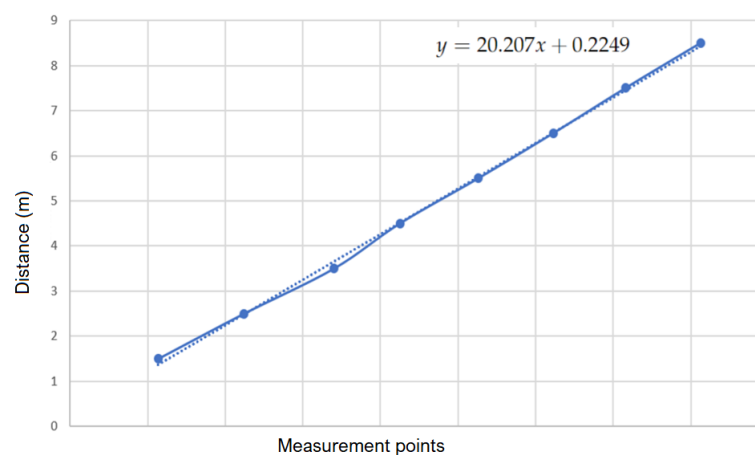
Figure 13. Determination of the real distance function two robots by using a camera.

4.3. The Experiment with Three Robots

Secondly, the three robots were placed in an arbitrary position in the experimental pool in the realized experimentation. However, they must fulfil the initial condition that the robot leader must be in the visible area of the two cameras on the robot followers. The robot follower detects four LEDs placed on the robot leader and then calculates the relative position between the camera and the four LEDs. Without loss of generality, we assume that the relative position between the robot follower and the robot leader is also the relative position between the robot follower camera and the four LEDs mounted on the robot leader.

The initial relative position between the BlueROV-1 tracking robot and the BlueROV leader robot is $(x, y) = (1.6$ m, 0.5 m$)$. The initial relative position between the BlueROV-2 follower robot and the BlueROV leader robot is $(x, y) = (1.6$ m, -0.5 m$)$. The goal is that the follower robots 1 and 2 should move automatically to ensure that the relative position between them is equal to 1.2 m.

Then, the leader robot moves along a given trajectory. In this case, it is controlled via a joystick. However, it could also be possible to adopt a vision-based approach for the l-UD leader, for example, a pipeline tracking application [25].

Simultaneously, the follower robots 1 and 2 must move and always ensure their distances from the leader robot are equal to 1.2 m. During movement, the robot depths were controlled by an appropriate PD control and built-in pressure sensor.

The objective is to maintain the l-UD BlueROV follower at a distance of 1.2 m between it and the l-UD BlueROV leader, and while the leader robot moves, the follower robots need to follow the leader robot. Moreover, they still have to maintain a distance of 1.2 m from the leader robot. Figure 11a shows the initial position of the l-UD BlueROV. Figure 11b shows the l-UD BlueROV tracking underwater robots moving in formation. Figure 11c,d show the l-UD leader moving; simultaneously, the two follower robots follow the leader robot and maintain a distance of 1.2 m.

It is also to be noted that it can guarantee only the distance between the follower robot and the leader robot, but it can not ensure a constant distance between the follower robots. Figures 14 and 15 show the distance between the follower robots 1 and 2 with the leader robot. We see that these distances all converge to 1.2 m. This may prove that the follower robots can keep a constant distance from the leader robot. The red line shows the distance between the robot follower and the robot leader after using the EKF filter to combine the vision and IMU values. The blue line shows the distance between the two robots when the EKF filter is not used.

Figures 14 and 15 also indicate two peaks (blue lines) that indicate that the cameras of the follower robot failed to detect the four LEDs placed on the leader robot. The objective of using the EKF is to predict the relative position between the follower and leader robots in this case. The numerical values of the EKF matrices can be found in [32,37]. The architecture of software components developed on the ROS platform can be seen in Figure A1 of Appendix A.1.
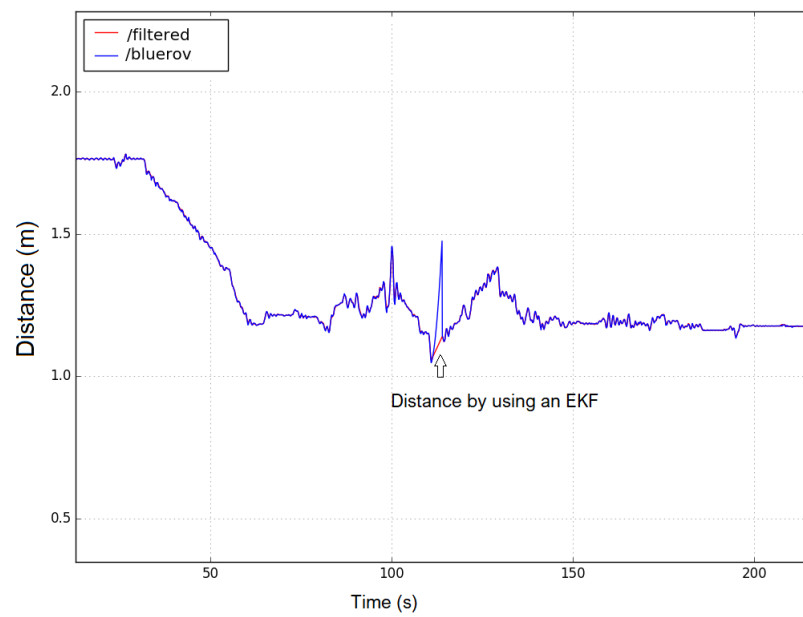
Figure 14. Distance between BlueROV robot follower 1 and the BlueROV robot leader.
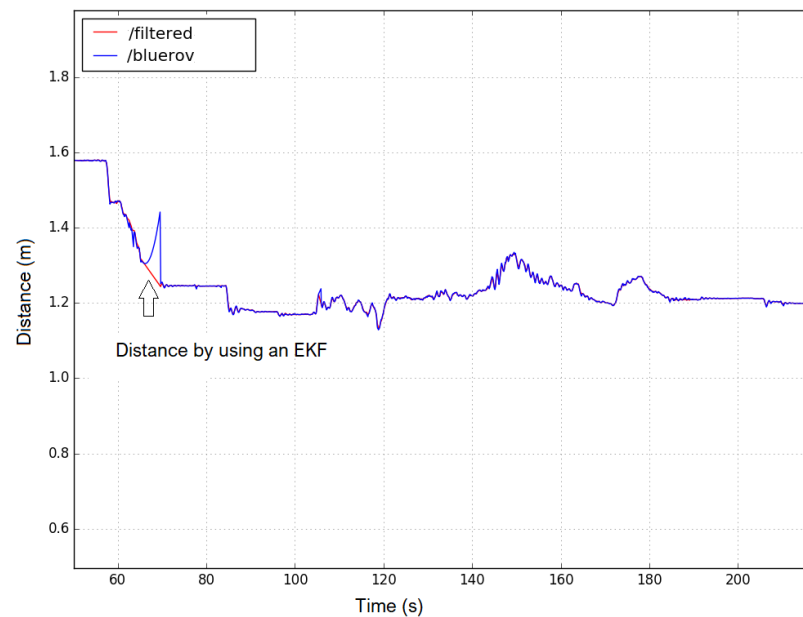

Figure 15. Distance between BlueROV robot follower 2 and the BlueROV robot leader.

Figures 16 and 17 show the control signals of the robot followers 1 and 2. The robots are controlled in the x (a blue line $/ux/data$) and y (a red line $/uy/data$) axes, respectively, to change the distance between the follower and the leader robot. We can observe that the rotation control signal $/up/data$ of the robots (the light blue line) has a large change because of the limitation of the accuracy of the sensors. However, these control signals can also converge when the distance between the robots reaches the desired value.
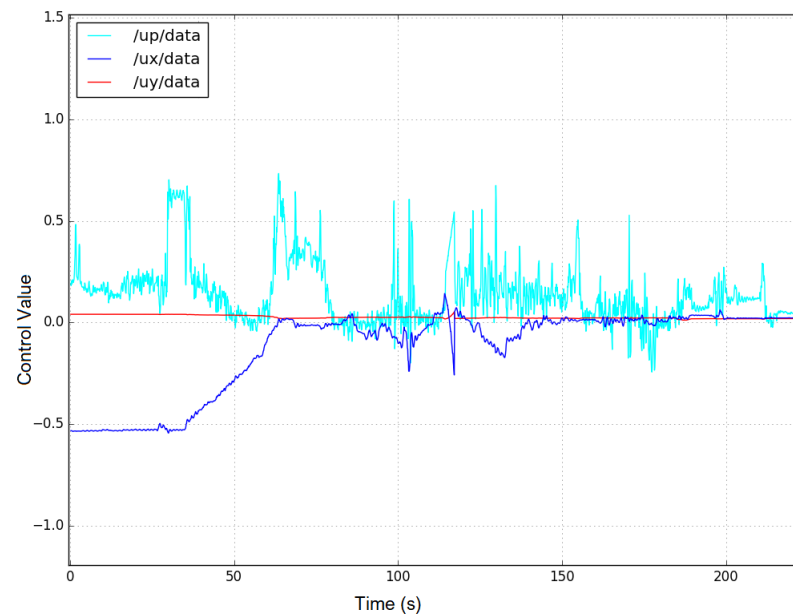
Figure 16. 3-axis control signals for the BlueROV follower-1 robot. (The blue, red, and light blue line represent the control signal on the x-axis and y-axis and the corresponding angle control signal, respectively).
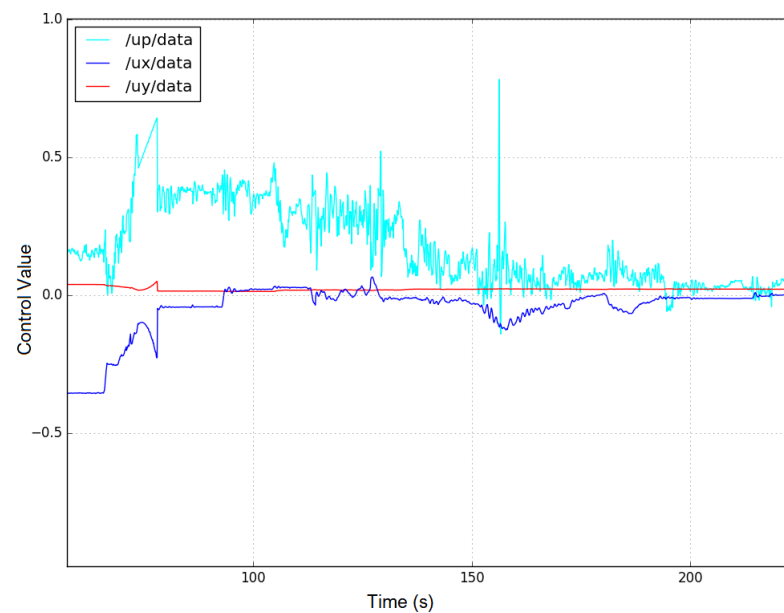

Figure 17. 3-axis control signals for the BlueROV follower-2 robot. (The blue, red, and light blue line represent the control signal on the x-axis and y-axis and the corresponding angle control signal, respectively).

Figure 18 shows images of the two BlueROV follower underwater robots, which were taken from the camera mounted on the leader robot.

Figure 19 shows images of the four LEDs detected on the leader robot from the BlueROV follower 1 and 2.
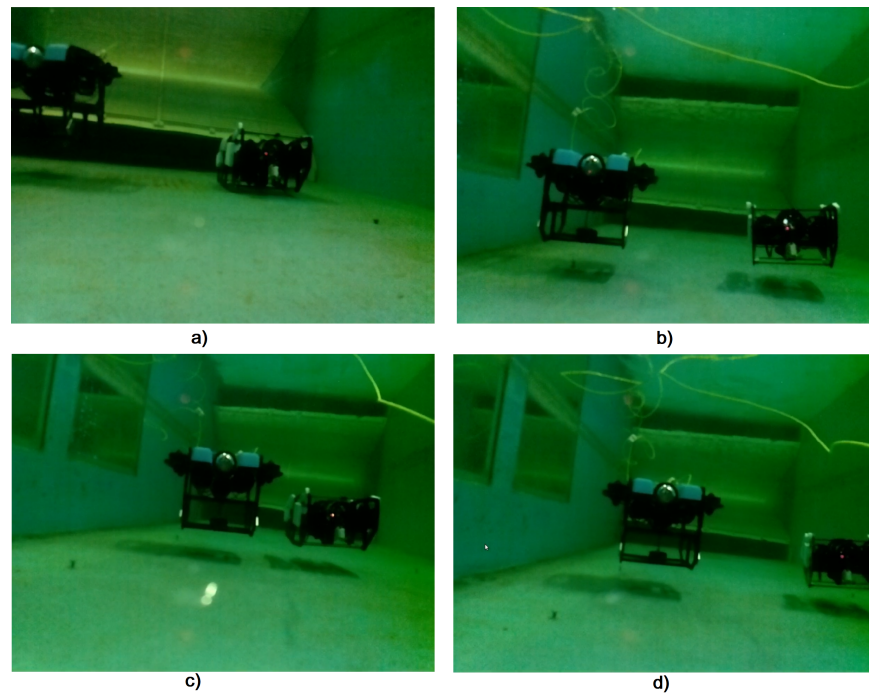
Figure 18. Image of the two following robots as seen from the camera of the leading robot. (**a**) description of the initial position of the robots follower. (**b**–**d**) illustration of the movement of the follower robots to make the formation.
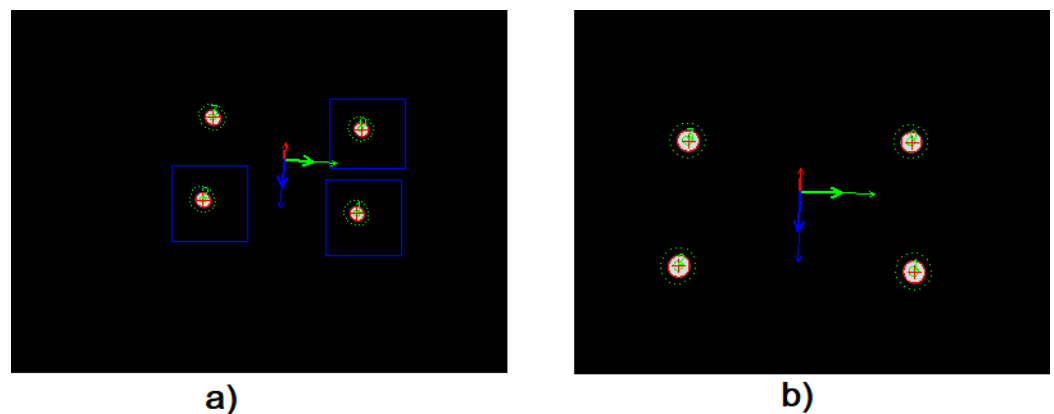
Figure 19. The two following robots look at the leading robot. (**a**,**b**) illustration of 4 LED detection by robot follower 1 and 2, respectively.

5. Conclusions

In order to complete a general approach to the distributed control of a swarm of low-cost underwater robots, which requires position information, we proposed a relative localization approach based on the fusion of vision and IMU data. Our experiments show that with three low-cost UAVs moving in the underwater environment, we are able to achieve a formation. In addition, our approach presents a solution for localizing a team of drones without digital communication and navigation sensors. The use of the EKF filter fulfils the purpose of predicting the relative position between the follower robots and the leader robot in cases where the follower robot's camera fails to detect the four LEDs placed on the leader robot (due to the environment or the robot being out of view of the camera).

In future work, robustness problems require further study to increase the accuracy. The relative position obtained between the robots is completely due to the combination of the camera and an IMU using an EKF filter. However, the use of the cameras to localize

underwater robots still presents many challenges in terms of accuracy, object detection (in this case, four LEDs), and changes in light intensity of the underwater environment.

Abbreviations

The following abbreviations are used in this manuscript:

AUV	Autonomous underwater vehicle
EKF	Extended Kalman filter
FOV	Field of view
IMU	Inertial measurement unit
LED	Light-emitting diode
l-UD	Low-cost underwater drone
PD	Proportional derivative
ROV	Remotely operated vehicle
UAV	Unmanned aerial vehicle
UGV	Unmanned ground vehicles
UUV	Unmanned underwater vehicles
PWM	Pulse-width modulation
USBL	Ultra-short baseline
GNSS	Global navigation satellite systems

Appendix A

Appendix A.1

Figure A1 shows a structure of the ROS nodes for the follower robot *i*, where we have developed additional nodes to be able to detect four LEDs of the l-UD leader, as well as using an additional EKF. Thanks to the advantages of ROS, these components can be fully reused or switched between the simulation model and the real model.
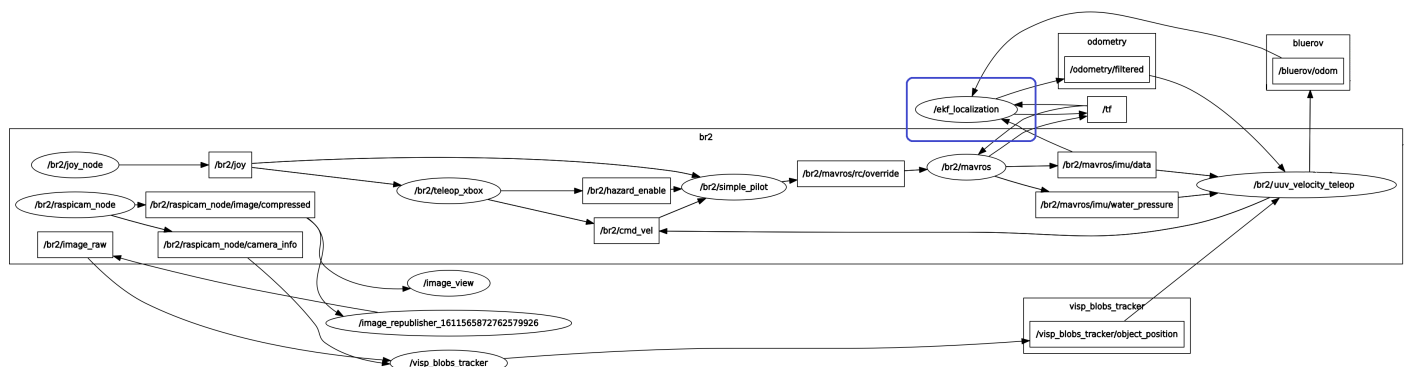

Figure A1. ROS node structure for a follower robot *i* with EKF nodes.

The reader can find details on the structure of the ROS nodes in our previous research [32].

References

1. Qin, J.; Gao, H.; Zheng, W.X. Second-order consensus for multi-agent systems with switching topology and communication delay. *Syst. Control. Lett.* **2011**, *60*, 390–397. https://doi.org/https://doi.org/10.1016/j.sysconle.2011.03.004.
2. Qin, J.; Yu, C.; Gao, H. Coordination for Linear Multiagent Systems With Dynamic Interaction Topology in the Leader-Following Framework. *IEEE Trans. Ind. Electron.* **2014**, *61*, 2412–2422. https://doi.org/10.1109/TIE.2013.2273480.
3. Qin, J.; Gao, H. A Sufficient Condition for Convergence of Sampled-Data Consensus for Double-Integrator Dynamics With Nonuniform and Time-Varying Communication Delays. *IEEE Trans. Autom. Control* **2012**, *57*, 2417–2422. https://doi.org/10.1109/TAC.2012.2188425.
4. Xiao, F.; Wang, L. Asynchronous Consensus in Continuous-Time Multi-Agent Systems With Switching Topology and Time-Varying Delays. *IEEE Trans. Autom. Control* **2008**, *53*, 1804–1816. https://doi.org/10.1109/TAC.2008.929381.
5. Li, H.; Shi, Y. *Robust Receding Horizon Control for Networked and Distributed Nonlinear Systems*; Springer: Cham, Switzerland, 2017. https://doi.org/https://doi.org/10.1007/978-3-319-48290-3.
6. Shi, G.; Johansson, K.H. Randomized optimal consensus of multi-agent systems. *Automatica* **2012**, *48*, 3018–3030. https://doi.org/https://doi.org/10.1016/j.automatica.2012.08.018.
7. Yu, W.; DeLellis, P.; Chen, G.; di Bernardo, M.; Kurths, J. Distributed Adaptive Control of Synchronization in Complex Networks. *IEEE Trans. Autom. Control* **2012**, *57*, 2153–2158. https://doi.org/10.1109/TAC.2012.2183190.
8. Qin, J.; Zheng, W.X.; Gao, H. Coordination of Multiple Agents With Double-Integrator Dynamics Under Generalized Interaction Topologies. *IEEE Trans. Syst. Man, Cybern. Part B Cybern.* **2012**, *42*, 44–57. https://doi.org/10.1109/TSMCB.2011.2164523.
9. Qin, J.; Gao, H.; Zheng, W.X. Exponential Synchronization of Complex Networks of Linear Systems and Nonlinear Oscillators: A Unified Analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 510–521. https://doi.org/10.1109/TNNLS.2014.2316245.
10. Yang, T.; Roy, S.; Wan, Y.; Saberi, A. Constructing consensus controllers for networks with identical general linear agents. *Int. J. Robust Nonlinear Control* **2011**, *21*, 1237–1256. https://doi.org/https://doi.org/10.1002/rnc.1641.
11. Peng, Z.; Wang, D.; Shi, Y.; Wang, H.; Wang, W. Containment control of networked autonomous underwater vehicles with model uncertainty and ocean disturbances guided by multiple leaders. *Inf. Sci.* **2015**, *316*, 163–179. https://doi.org/https://doi.org/10.1016/j.ins.2015.04.025.
12. Qin, J.; Ma, Q.; Shi, Y.; Wang, L. Recent Advances in Consensus of Multi-Agent Systems: A Brief Survey. *IEEE Trans. Ind. Electron.* **2017**, *64*, 4972–4983. https://doi.org/10.1109/TIE.2016.2636810.
13. Cao, Y.; Yu, W.; Ren, W.; Chen, G. An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination. *IEEE Trans. Ind. Informatics* **2013**, *9*, 427–438. https://doi.org/10.1109/tii.2012.2219061.
14. Pham, H.A.; Soriano, T.; Ngo, V.H.; Gies, V. Distributed Adaptive Neural Network Control Applied to a Formation Tracking of a Group of Low-Cost Underwater Drones in Hazardous Environments. *Appl. Sci.* **2020**, *10*, 1732. https://doi.org/10.3390/app10051732.
15. Paull, L.; Saeedi, S.; Seto, M.; Li, H. AUV Navigation and Localization: A Review. *IEEE J. Ocean. Eng.* **2014**, *39*, 131–149. https://doi.org/10.1109/joe.2013.2278891.
16. Chen, S.; Yin, D.; Niu, Y. A Survey of Robot Swarms' Relative Localization Method. *Sensors* **2022**, *22*, 4424. https://doi.org/10.3390/s22124424.
17. Garcia, J.G.; Gomez-Espinosa, A.; Cuan-Urquizo, E.; Valdovinos, L.G.G.; Salgado-Jimenez, T.; Cabello, J.A.E. Autonomous Underwater Vehicles: Localization, Navigation, and Communication for Collaborative Missions. *Appl. Sci.* **2020**, *10*, 1256. https://doi.org/10.3390/app10041256.
18. Alatise, M.; Hancke, G. Pose Estimation of a Mobile Robot Based on Fusion of IMU Data and Vision Data Using an Extended Kalman Filter. *Sensors* **2017**, *17*, 2164. https://doi.org/10.3390/s17102164.
19. Bosch, J.; Gracias, N.; Ridao, P.; Istenic, K.; Ribas, D. Close-Range Tracking of Underwater Vehicles Using Light Beacons. *Sensors* **2016**, *16*, 429.
20. Cao, Z.; Liu, R.; Yuen, C.; Athukorala, A.; Ng, B.K.K.; Mathanraj, M.; Tan, U.X. Relative Localization of Mobile Robots with Multiple Ultra-WideBand Ranging Measurements. *arXiv* **2017**. https://doi.org/10.48550/arxiv.2107.08842.
21. Bosse, M.; Zlot, R. Map Matching and Data Association for Large-Scale Two-dimensional Laser Scan-based SLAM. *Int. J. Robot. Res.* **2008**, *27*, 667–691. https://doi.org/10.1177/0278364908091366.
22. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: part I. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110. https://doi.org/10.1109/MRA.2006.1638022.
23. Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In Proceedings of the Eighteenth National Conference on Artificial Intelligence, Edmonton, Alberta, Canada, 28 July 2002, American Association for Artificial Intelligence; p. 593–598.
24. Thrun, S.; Montemerlo, M. The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. *Int. J. Robot. Res.* **2006**, *25*, 403–429. https://doi.org/10.1177/0278364906065387.
25. Allibert, G.; Hua, M.D.; Krupínski, S.; Hamel, T. Pipeline following by visual servoing for Autonomous Underwater Vehicles. *Control Eng. Pract.* **2019**, *82*, 151 – 160. https://doi.org/https://doi.org/10.1016/j.conengprac.2018.10.004.

26. Shkurti, F.; Chang, W.D.; Henderson, P.; Islam, M.J.; Higuera, J.C.G.; Li, J.; Manderson, T.; Xu, A.; Dudek, G.; Sattar, J. Underwater Multi-Robot Convoying using Visual Tracking by Detection. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017.

27. Manderson, T.; Higuera, J.C.G.; Cheng, R.; Dudek, G. Vision-based Autonomous Underwater Swimming in Dense Coral for Combined Collision Avoidance and Target Selection. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018.

28. Abreu, P.C.; Pascoal, A.M. Formation Control in the scope of the MORPH project. Part I: Theoretical Foundations. *IFAC-PapersOnLine* **2015**, *48*, 244–249. https://doi.org/10.1016/j.ifacol.2015.06.040.

29. Abreu, P.C.; Bayat, M.; Pascoal, A.M.; Botelho, J.; Góis, P.; Ribeiro, J.; Ribeiro, M.; Rufino, M.; Sebastião, L.; Silva, H. Formation Control in the scope of the MORPH project. Part II: Implementation and Results. *IFAC-PapersOnLine* **2015**, *48*, 250–255. https://doi.org/10.1016/j.ifacol.2015.06.041.

30. Zhang, J.; Zhou, W.; Wang, X. UAV Swarm Navigation Using Dynamic Adaptive Kalman Filter and Network Navigation. *Sensors* **2021**, *21*, 5374. https://doi.org/10.3390/s21165374.

31. Peng, Z.; Wang, D.; Chen, Z.; Hu, X.; Lan, W. Adaptive Dynamic Surface Control for Formations of Autonomous Surface Vehicles With Uncertain Dynamics. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 513–520. https://doi.org/10.1109/tcst.2011.2181513.

32. Pham, H.A. Coordination de Systèmes Sous-Marins Autonomes Basée sur une Méthodologie Intégrée dans un Environnement Open-Source. Ph.D. Thesis, University of Toulon, La Valette-du-Var, France, 2021.

33. Marchand, E.; Uchiyama, H.; Spindler, F. Pose estimation for augmented reality and a hands-on survey. *IEEE Trans. Vis. Comput. Graph.* **2016**, *22*, 2633–2651.

34. Wang, P.; Xu, G.; Cheng, Y.; Yu, Q. A simple, robust and fast method for the perspective-n-point Problem. *Pattern Recognit. Lett.* **2018**, *108*, 31–37. https://doi.org/10.1016/j.patrec.2018.02.028.

35. Comport, A.I.; Marchand, E.; Pressigout, M.; Chaumette, F. Real-Time Markerless Tracking for AugmentedReality: The Virtual Visual Servoing Framework. *IEEE Trans. Vis. Comput. Graph.* **2006**, *12*, 615–628.

36. BlueRobotics. BlueROV2 Assembly. **2016,** California St, Torrance, CA 90503, USA. Available online: https://bluerobotics.com/store/rov/bluerov2/.

37. Moore, T.; Stouch, D. A Generalized Extended Kalman Filter Implementation for the Robot Operating System. In Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13), Padova, Italy, 15–18 July 2014.