

# Peer-to-Peer Localization via On-board Sensing for Aerial Flocking

Fat-Hy Omar Rajab<sup>1</sup>, Samet Güler<sup>2</sup> and Jeff S. Shamma<sup>1</sup>

**Abstract**—The performance of mobile multi-robot systems dramatically depends on the mutual awareness of individual robots, particularly the positions of other robots. GPS and motion capture cameras are commonly used to acquire and ultimately communicate positions of robots. Such sensing schemes depend on infrastructure and restrict the capabilities of a multi-robot system, e.g., the robots cannot operate in both indoor and outdoor environments. Conversely, peer-to-peer localization algorithms can be used to free the robots from such infrastructures. In such systems, robots use on-board sensing to infer the positions of nearby robots. In this approach, it is essential to have a model of the motion of other robots. We introduce a flocking localization scheme that takes into account motion behavior exhibited by the other robots. The proposed scheme depends only on the robots' on-board sensors and computational capabilities and yields a more accurate localization solution than the peer-to-peer localization algorithms that do not take into account the flocking behavior. We verify the performance of our scheme in simulations and demonstrate experiments on two unmanned aerial vehicles.

## I. INTRODUCTION

Multi-robot systems (MRS) can outperform single robot systems in a variety of metrics. To effectively interact with the environment and carry out a given task efficiently, robots in MRS must have situational awareness, which requires accurate localization of the other robots in the system [1], [2]. In MRS, a typical localization solution is to equip every robot with GPS receivers to acquire absolute positions. However, this method requires continuous communication within the MRS, which may not be desirable for large networks due to the bandwidth constraints in the communication channels [3], [4]. Also, GPS shows low performance in indoor applications and occluded areas [5].

Recent developments in on-board sensing and computational units have enabled the real-time implementation of simultaneous localization and mapping (SLAM) applications [6], [7]. For instance, references [8] and [9] introduce an algorithm that produces real-time full dense visual maps, but their methods impose a high computational burden that restricts its applicability in mobile robotic systems with limited processing power. For mobile robotic systems, reference [10] proposes the non-iterative SLAM (NI-SLAM) algorithm that is computationally light, but the algorithm has a setback of accumulating visual drifts in the long run. The visual drifts can be reduced by using loop-closure

algorithms that necessitate more computational power [11], [12]. Reference [13] uses on-board ultra-wideband (UWB) distance measurements and depth camera data in an extended Kalman filter (EKF) to improve the performance of NI-SLAM.

Cooperative localization makes use of relative measurement of the robots as feedback to refine the joint pose estimate of the MRS which can be done in a centralized or decentralized manner [1], [5], [14]. Although centralized cooperative localization (CCL) is accurate and easily implementable, it has a single point of failure and imposes high communication costs, making the decentralized cooperative localization (DCL) more desirable [14]. However, it is challenging to design an algorithm that maintains consistent communication and computational costs while maintaining an excellent joint pose estimate [5], [14], [15].

UWB sensors provide omnidirectional distance measurements with high precision [5], [13], [16]–[18]. A conventional approach is to use a set of UWB antennas as anchors with known locations and estimate the location of a UWB sensor onboard a robot by optimization or geometric methods, as in motion capture camera systems [16], [17]. Recently, references [19]–[21] installed UWB anchors onboard the robots to localize other robots in the MRS, i.e an *onboard anchor configuration*. The resulting configuration possesses several drawbacks such as (i) the increased uncertainty due to moving anchors, (ii) short inter-anchor distances as a result of onboard configuration, and (iii) lack of a global reference entity. Nevertheless, satisfactory estimation results are demonstrated with filtering methods in [21], [22].

Most localization algorithms do not make use of the behavioral characteristics exhibited by the robots in an MRS. If robots work together collaboratively, they tend to move together in close vicinity and, in some settings, follow each other [1], [14]. This characteristic, called flocking, occurs in natural phenomena like in flocks of birds or a school of fish. This work aims to develop a localization scheme called the flocking localization algorithm that explores the flocking behavior of robots in MRS. To quantify the improvement in performance, the flocking localization algorithm is compared to the drift localization scheme which is another localization scheme that does not explore the flocking behavior of robots. Both the flocking and drift localization algorithms use three anchor UWB sensors and on-board computational units mounted on both the anchor (localizing robot) and the tag (localized robot). It is assumed that the tag employs a flocking control system that makes it continuously follow the anchor drone. In this setup, the flocking localization algorithm captures the behavior of the tag robot more than the

\*The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST).

<sup>1</sup>Fat-Hy Omar Rajab and Jeff S. Shamma are with Robotics, Intelligent Systems, and Control (RISC) Lab, Computer, Electrical and Mathematical Science and Engineering Division (CEMSE), KAUST, Thuwal 23955–6900, Saudi Arabia. fathyomar.rajab@kaust.edu.sa.

<sup>2</sup>Samet Güler is with Abdullah Gül University, Kayseri, Turkey.

drift localization algorithm and thus, makes better estimates of the states of the tag robot even in the presence of noise in the measurements, as shown in later sections.

Also, there is no explicit communication between the anchor and the tag, making the algorithm infrastructure-independent and ideal for usage in both indoors or outdoors scenarios [21], [22].

The remainder of the paper is organized as follows. Section II will explain the problem setup and some theoretical background related to the problem. The proposed localization algorithm is introduced in Section III. Simulation setup and results are shown in Section IV. The experimental implementation and results are presented in Section V. Finally, we discuss and analyze the results in Section VI and present concluding remarks in Section VII.

## II. PROBLEM SETUP

We consider a system of two robots as shown in the Figure 1 below: The anchor robot  $R_a$  and the tag robot  $R_t$ . The robots operate on a 2D Euclidean plane whose fixed inertial coordinate frame is denoted by  $\mathcal{F}_0$ . The body frames of the robots originated at the centers of  $R_a$  and  $R_t$  are denoted by  $\mathcal{F}_a$  and  $\mathcal{F}_t$ , respectively. The positions of  $R_a$  and  $R_t$  with respect to the inertial coordinate frame  $\mathcal{F}_0$  are  $\mathbf{r}_{a|0}^0 = (x_a, y_a)$  and  $\mathbf{r}_{t|0}^0 = (x_t, y_t)$ , respectively. The velocity of the anchor and the tag with respect to the inertial coordinate frame expressed in inertial coordinate frame are  $\mathbf{v}_{a|0}^0 = (v_{ax}, v_{ay})$  and  $\mathbf{v}_{t|0}^0 = (v_{tx}, v_{ty})$ .

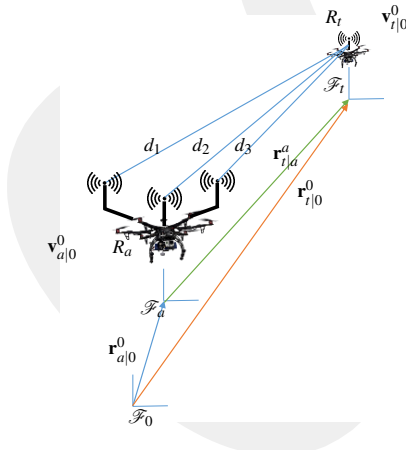


Fig. 1. Kinematic model of system of two robots

The discrete-time holonomic kinematic models of the anchor robot  $R_a$  and the tag robot  $R_t$  is expressed as follows:

$$\mathbf{r}_{a|0}^0[k+1] = \mathbf{r}_{a|0}^0[k] + \mathbf{v}_{a|0}^0[k]\Delta k + \bar{\mathbf{w}}_{a,0}, \quad (1)$$

$$\mathbf{v}_{a|0}^0[k+1] = \mathbf{v}_{a|0}^0[k] + \bar{\mathbf{w}}_{a,1}, \quad (2)$$

where  $\bar{\mathbf{w}}_{a,0} \in \mathfrak{R}^2$  and  $\bar{\mathbf{w}}_{a,1} \in \mathfrak{R}^2$  are noise vectors, and  $\Delta k$  is the time step. The motion of the tag robot  $R_t$  is given by the Equations (1) and (2) with the subscript  $a$  replaced by  $t$ .

Three UWB sensors are rigidly attached on the anchor robot  $R_a$  at known positions  $\mathbf{r}_{i|a}^a = (x_i^a, y_i^a)$  in frame  $\mathcal{F}_a$ , where  $i = \{1, 2, 3\}$  [21]. The position of the UWB anchors can be expressed in the inertial frame as follows:

$$\begin{bmatrix} \mathbf{r}_{i|0}^0 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{C}_a^0 & \mathbf{r}_{i|a}^a \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_{i|a}^a \\ 1 \end{bmatrix}, \quad (3)$$

where  $\mathbf{C}_a^0$  is the rotation matrix from  $\mathcal{F}_a$  to  $\mathcal{F}_0$ .

The tag drone  $R_t$  has a UWB sensor at the center of the body frame  $\mathcal{F}_t$ . Let the distance measurements between UWB anchor  $i$  on  $R_a$  and the UWB sensor on  $R_t$  be denoted by

$$d_i = \left\| \mathbf{r}_{i|0}^0 - \mathbf{r}_{t|0}^0 \right\|_2 + \mathcal{N}(0, \sigma^2), \quad (4)$$

where  $\mathcal{N}(0, \sigma^2)$  denotes Gaussian noise with zero mean and variance of  $\sigma^2$ .

We assume that the tag robot moves based on two modes  $\mathbf{M} = \{\text{Flocking}, \text{Drift}\}$ . When the tag robot moves based on the **Flocking** mode, it follows the anchor robot by maintaining a constant relative position with respect to the anchor robot. On the other hand, when the tag is moving in the **Drift** mode, it moves with random velocity.

In our design process, we assume that the tag robot is in **Flocking** mode. We will develop a peer-to-peer localization scheme called the flocking localization scheme which takes into account the flocking behavior of the tag robot in estimating the states of the tag drone. This work compares the performance of the flocking localization scheme to another localization scheme called drift localization scheme that does not take into account the flocking behavior of the tag robot in estimating the states of the tag robot. The state of the tag robot is a vector having the position of the tag robot and the velocity of the tag, which can be expressed as follows:

$$\mathbf{x}_t^0 = \begin{bmatrix} \mathbf{r}_{t|0}^0 & \mathbf{v}_{t|0}^0 \end{bmatrix}^\top.$$

The estimate of the state of the tag,  $\mathbf{x}_t^0$ , with respect to the inertial frame,  $\mathcal{F}_0$ , requires the knowledge of the position of the anchor robot,  $\mathbf{r}_{a|0}^0$ . Most of the applications require only relative position of the tag with respect to the anchor drone, in which case the frame  $\mathcal{F}_0$  is merged with frame  $\mathcal{F}_a$ . Consequently,  $\mathbf{r}_{a|0}^0 = \mathbf{0}$  and  $\mathbf{C}_a^0 = \mathbf{I}_4$ . Therefore, from Equation (5) below:

$$\mathbf{r}_{t|0}^0 = \mathbf{C}_a^0 \mathbf{r}_{t|a}^a + \mathbf{r}_{a|0}^0. \quad (5)$$

Substituting  $\mathbf{r}_{a|0}^0 = \mathbf{0}$  and  $\mathbf{C}_a^0 = \mathbf{I}_4$  in Equation (5), we will have:

$$\mathbf{r}_{t|0}^0 = \mathbf{r}_{t|a}^a.$$

Taking the derivative of Equation (5) with respect to time and doing the same manipulations will give the following results:

$$\mathbf{v}_{t|0}^0 = \mathbf{v}_{t|a}^a.$$

The state of the tag robot with respect to the anchor robot,  $\mathbf{x}_t^a$ , or simply  $\mathbf{x}_t$  will be a vector containing the relative

position and velocity of the tag robot with respect to the anchor drone:

$$\mathbf{x}_t = \mathbf{x}_t^a = \begin{bmatrix} \mathbf{r}_{t|a}^a & \mathbf{v}_{t|a}^a \end{bmatrix}^\top.$$

Our goal is to estimate the state  $\mathbf{x}_t$  of the tag robot  $R_t$  with respect to the anchor robot by using the on-board capabilities of robot  $R_a$ . Particularly, we define the estimation problem as follows: Given the motion models of robots  $R_a$  and  $R_t$ , the three distance measurements  $d_1, d_2, d_3$ , the on-board IMU measurements, and the motion mode of the robot, where  $\mathbf{M}$  is *Flocking* mode, generate the estimate of the state  $\hat{\mathbf{x}}_t$  of the tag robot  $R_t$  that includes relative position and velocity of the tag robot.

The best tag state estimate,  $\hat{\mathbf{x}}_t$ , is the one with the minimum error that is defined by  $\|e\| = \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|$ .

### III. ON-BOARD PEER-TO-PEER LOCALIZATION ALGORITHMS IN 2D

Robot  $R_a$  estimates the states of robot  $R_t$  by utilizing the UWB anchors and a computational unit attached on-board robot  $R_a$ , without any explicit communication between  $R_a$  and  $R_t$ . We will present two localization algorithms, each of which corresponds to the *Flocking* mode and the *Drift* mode. The two models differ only on the prediction model in the Kalman filter. The estimation of the flocking localization model takes into account the flocking behavior of the tag drone, while the estimation of the drift localization model assumes a random motion model of the tag robot,  $R_t$ .

We present the block diagram of our localization algorithms in Figure 2. First, the three distance measurements,  $d_1, d_2, d_3$ , between the three UWB anchors on robot  $R_a$  and the UWB tag sensor on robot  $R_t$ , are acquired. Using the three distance measurements, the Trilateration block generates a rough estimation of the position of the tag sensor,  $\mathbf{z}_k$ , which is then fed to the Kalman filter in the Correction block as the measurement model. Then, the Correction block produces the estimate  $\mathbf{x}_{k|k}$  for the tag robot's state. Remarkably, the *Flocking* mode and the *Drift* mode use different Prediction models in the Kalman filter, which we explain in the following sections in more detail.

#### A. Trilateration

In this phase, we estimate the tag robot's position by using the noisy raw distance measurements,  $d_i$ , acquired from the UWB sensors. Substituting the positions of the three UWB anchors  $\mathbf{r}_{i|0}^a = (x_i, y_i)$  in (4) yields

$$\mathbf{A}_k \mathbf{p}_k = \mathbf{b}_k,$$

where

$$\mathbf{A}_k = \begin{bmatrix} x_1 & y_1 & -\frac{1}{2} \\ x_2 & y_2 & -\frac{1}{2} \\ x_3 & y_3 & -\frac{1}{2} \end{bmatrix}, \quad \mathbf{b}_k = \begin{bmatrix} \frac{1}{2}(x_1^2 + y_1^2 - d_1^2) \\ \frac{1}{2}(x_2^2 + y_2^2 - d_2^2) \\ \frac{1}{2}(x_3^2 + y_3^2 - d_3^2) \end{bmatrix},$$

$$\mathbf{p}_k = \begin{bmatrix} x_t \\ y_t \\ x_t^2 + y_t^2 \end{bmatrix}.$$

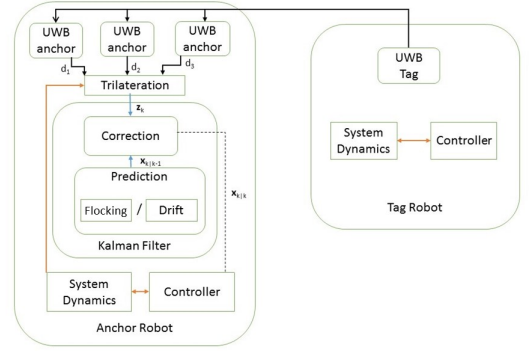


Fig. 2. Block diagram of the on-board peer-to-peer localization algorithms

Notably, since the distance measurements are noisy, this system of equations might not have a unique solution  $\hat{\mathbf{p}}_k$ . We find the estimate of  $\hat{\mathbf{p}}_k$  that minimizes the following objective function:

$$\hat{\mathbf{p}}_k = \arg \min_{\mathbf{p}_k} \|\mathbf{A}_k \mathbf{p}_k - \mathbf{b}_k\|_2^2,$$

which is a linear least square problem with the closed form solution [19], [20]:

$$\hat{\mathbf{p}}_k = \left( \mathbf{A}_k^\top \mathbf{A}_k \right)^{-1} \mathbf{A}_k^\top \mathbf{b}_k. \quad (6)$$

Linear least squares technique is known to be susceptible to outlier measurements. An exponential smoothing technique is applied to avoid abrupt jumps between consecutive data points in a signal [20]. Consider the following exponential smoothing function:

$$\mathbf{s}_k = \begin{cases} \hat{\mathbf{p}}_k, & k = 0 \\ \gamma \cdot \hat{\mathbf{p}}_k + (1 - \gamma) \cdot \mathbf{s}_{k-1}, & k > 0 \end{cases} \quad (7)$$

where  $\mathbf{s}_k$  is the smooth position estimate value at time step  $k$ , and  $\gamma$  is a scalar value between 0 and 1.

We extract the measured position of the tag robot as the first two elements of the vector  $\mathbf{s}_k$  as follows:

$$\mathbf{z}_k = [\mathbf{s}_k(1), \mathbf{s}_k(2)]^\top.$$

#### B. Estimation Algorithm

The estimation of the states of the tag robot is done using the Kalman filter [23]. The Kalman filter consists of two main phases, the Prediction phase, and the Correction phase.

1) *Prediction Phase*: We use two discrete linear models to describe the motion of the tag robot in this phase [20], [23].

a) *Drift Localization Model*: In this model, robot  $R_a$  assumes that robot  $R_t$  does not try to form a flocking behavior with itself. To represent this framework, we use the following discrete difference equation as the motion model for the tag drone:

$$\mathbf{v}_{t,k|k-1} = \mathbf{v}_{t,k-1|k-1} + \boldsymbol{\eta}_d, \quad (8)$$

where the tag drone is assumed to meander around the anchor drone with a bounded, random acceleration input,  $\eta_d$ , sampled from the Gaussian Distribution  $\mathcal{N}(0, \sigma_d^2)$  [19].

The position of the tag robot is described as follows [20]:

$$\mathbf{r}_{t,k|k-1} = \mathbf{r}_{t,k-1|k-1} + \Delta t \cdot \mathbf{v}_{t,k-1|k-1}. \quad (9)$$

Equations (8) and (9) are used as the discrete linear models in the Kalman filter. Using the drift model and the trilateration algorithm together, the Kalman filter formulates the drift localization algorithm.

2) *Flocking Localization Model*: In this model, we assume that the tag robot  $R_t$  tries to follow the anchor drone at a constant relative position, which can be described using the following differential equation:

$$\frac{d\mathbf{v}_t}{dt} = \alpha(\mathbf{v}_a - \mathbf{v}_t + \mathbf{w}), \quad (10)$$

where  $\alpha > 0$  is a scalar multiplier and  $\mathbf{w}$  is the uncertainty in the dynamics of the tag robot which is assumed to be Gaussian noise  $\mathcal{N}(0, \sigma_f^2)$ , with mean zero and standard deviation  $\sigma_f$ . Equation (10) is discretized to obtain equation (11) below:

$$\mathbf{v}_{t,k|k-1} = (1 - \alpha \cdot \Delta t) \mathbf{v}_{t,k-1|k-1} + \alpha \cdot \Delta t \cdot \mathbf{v}_{a,k-1|k-1} + \alpha \cdot \Delta t \cdot \mathbf{w}_{k-1}. \quad (11)$$

Refer to Equation (9), which describes the position of the tag robot.

Equations (9) and (11) are used as the discrete linear models in the Kalman filter. Using the flocking model and the trilateration algorithm together, the Kalman filter formulates the flocking localization algorithm.

The following equation [23] gives the prior prediction of the error covariance matrix:

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k|k-1} \mathbf{F}_k^\top + \mathbf{Q}_k, \quad (12)$$

where  $\mathbf{P}_{k|k} \in \mathfrak{R}^{4 \times 4}$  is a positive definite matrix that is initialized as  $\sigma_p^2 \mathbf{I}_4$  [19], [23], and the matrix  $\mathbf{Q}_k \in \mathfrak{R}^{4 \times 4}$  describes the process noise covariance matrix. For simplicity, we set  $\mathbf{Q}_k = \sigma_q^2 \mathbf{I}_4$  [23]. Finally,  $\mathbf{F}_k$  is obtained from the discrete linear models. For the drift localization model,  $\mathbf{F}_k$  is set as follows:

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (13)$$

On the contrary, for the flocking localization model,  $\mathbf{F}_k$  is set as follows:

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 - \alpha \cdot \Delta t & 0 \\ 0 & 0 & 0 & 1 - \alpha \cdot \Delta t \end{bmatrix}. \quad (14)$$

3) *Correction*: In this phase, the data obtained from the noisy measurements is used to refine the state estimation obtained in the prediction phase.

First, the optimal Kalman gain is obtained using the following equation [23]:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^\top (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_k)^{-1}, \quad (15)$$

where  $\mathbf{H}_k \in \mathfrak{R}^{2 \times 4}$  is an observation matrix written as

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Also,  $\mathbf{R}_k \in \mathfrak{R}^{2 \times 2}$  is an observation noise covariance matrix which, for simplicity, is set as  $\mathbf{R}_k = \sigma_r^2 \mathbf{I}_2$ .

The error covariance matrix is corrected using the optimal Kalman gain, as shown in the equation below:

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}. \quad (16)$$

The prior state estimate is refined using the optimal Kalman gain to obtain the posterior state estimate as shown in the equation below [19], [20]:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}). \quad (17)$$

The observability matrix of the linear systems is defined as:

$$\mathcal{O} = [\mathbf{H}_k, \mathbf{H}_k \mathbf{F}_k, \mathbf{H}_k \mathbf{F}_k^2, \mathbf{H}_k \mathbf{F}_k^3]^\top.$$

The rank of the observability matrix  $\mathcal{O}$  is 4 for both flocking and drift localization algorithms. Therefore, the full state of the system can be estimated from the measurement obtained from the Trilateration Stage III-A.

#### IV. SIMULATION RESULTS

We simulated the drift and flocking localization algorithms with the Robot Operating System (ROS) framework and Gazebo. In the simulations and experiments, we used two drones. The localizing drone,  $R_a$ , had three UWB anchors set at known positions on the drone:  $[0.34m, 0]^T$ ,  $[0, 0]^T$  and  $[0, 0.34m]^T$ . The UWB sensors were simulated to be providing noisy distance measurements in which the noise was assumed to be Gaussian distribution,  $\mathcal{N}(0, 0.05^2)$ . In the experiment, the noise characteristics of UWB sensors were obtained using a method described in [21].

Figure 3 shows the simulation and experimental setup where the anchor drone was controlled using the joystick, which sends velocity commands to the anchor drone. We assume that the fast dynamics of the drones are controlled using the flight controller, which is the PX4 through the mavros. This anchor drone,  $R_a$ , was localizing another drone called the tag drone,  $R_t$ . The tag drone is following the anchor drone by executing the following control law described in Equations (18)-(22). This control law aims at maintaining a constant distance between the tag and the anchor drone [24], [25]. First, the error between the desired distance,  $d_x$ ,  $d_y$ , and the actual distance between the drones in both x and y directions is calculated as shown in Equation (18) and (19). The actual distance between the anchor,  $R_a$ , and tag,  $R_t$ , is

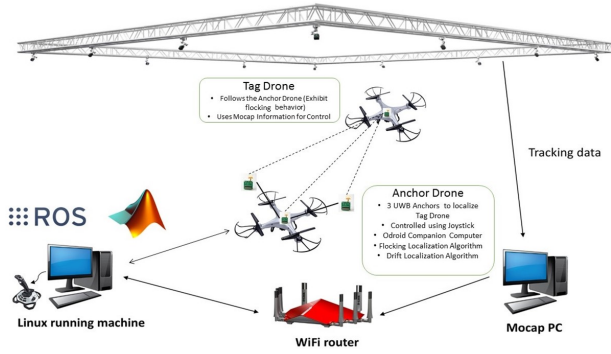


Fig. 3. Simulation and experimental setup

obtained from the ground truth data in the simulation and the motion capture (Mocoap) system in the experiment.

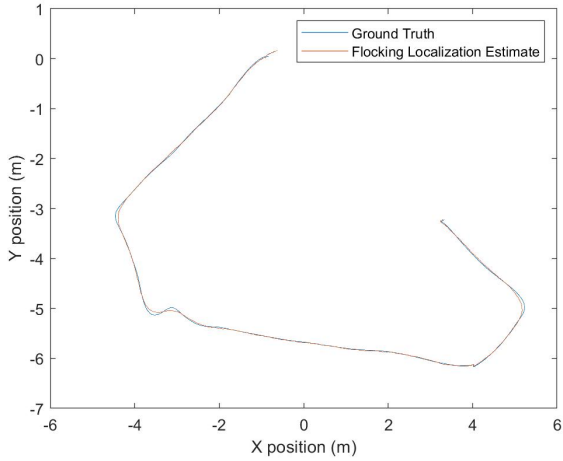


Fig. 4. Position estimate using flocking localization algorithm in simulations

$$e_{dx} = d_x - \|x_t - x_a\|_2, \quad (18)$$

$$e_{dy} = d_y - \|y_t - y_a\|_2. \quad (19)$$

Then, PI control law is used to calculate the velocity commands given to the tag drone [24]:

$$v_{tx,c}^0 = -\text{sgn}(x_a - x_t) \times \left( K_p e_{dx} + K_i \int_0^t e_{dx} dt \right), \quad (20)$$

$$v_{ty,c}^0 = -\text{sgn}(y_a - y_t) \times \left( K_p e_{dy} + K_i \int_0^t e_{dy} dt \right), \quad (21)$$

where the values of the proportionality constant  $K_p$  and integral constant  $K_i$  were manually tuned to obtain the best performance. In the simulations, setting  $K_p$  and  $K_i$  as 1.3 and 0.06 respectively, resulted to a settling time of 2 seconds and Percentage overshoot of 10%.

The above control law gave velocity commands in the inertial coordinate frame which have to be transformed to

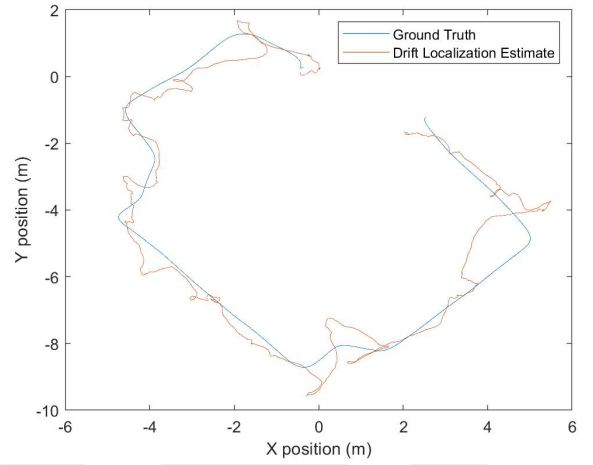


Fig. 5. Position estimate using drift localization algorithm in simulations

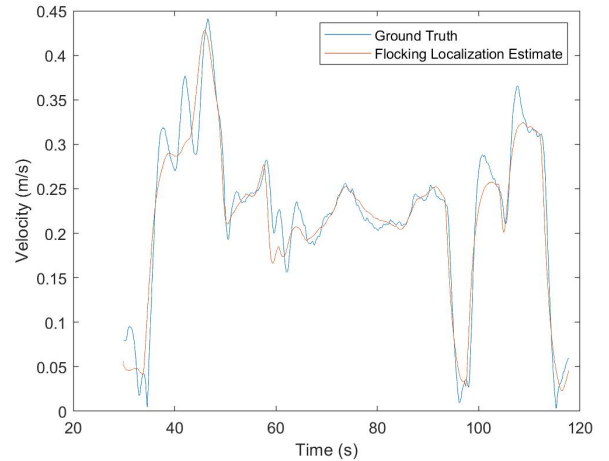


Fig. 6. Velocity estimate using flocking localization algorithm in simulations

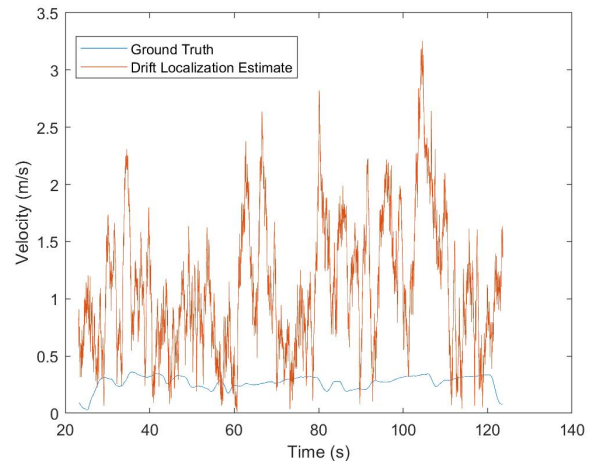


Fig. 7. Velocity estimate using drift localization algorithm in simulations

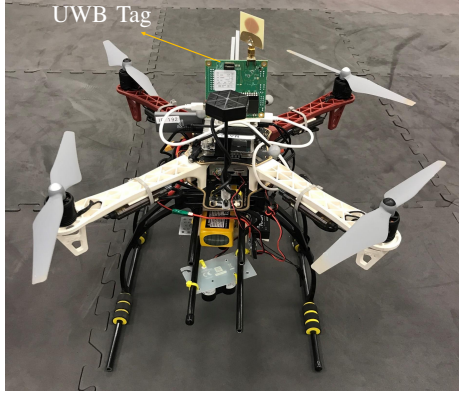


Fig. 8. Tag drone

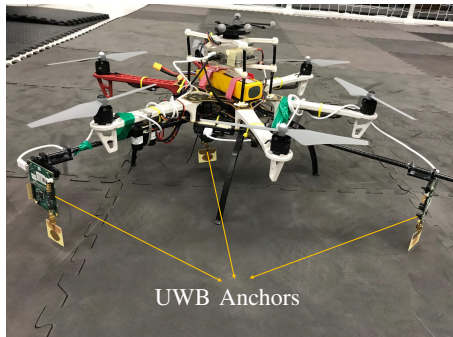


Fig. 9. Anchor drone

the coordinate frame of the tag drone as follows [26]:

$$\begin{bmatrix} v_{tx,c}^t \\ v_{ty,c}^t \end{bmatrix} = \mathbf{C}_0^t \begin{bmatrix} v_{tx,c}^0 \\ v_{ty,c}^0 \end{bmatrix}, \quad (22)$$

where  $\mathbf{C}_0^t$  is the rotation matrix from  $\mathcal{F}_0$  to  $\mathcal{F}_t$

The velocity commands calculated above are given to the flight controller as the high-level commands in the tag drone,  $R_t$ .

The drift and flocking localization algorithms in the simulations were subjected to several experiments in which different parameters of the Kalman filter (the scalar multipliers of the measurement noise, the process noise, and the error covariance matrices) were changed. The anchor drone was made to follow some random path using the joystick while localizing the tag drone, and the tag drone was following the anchor. The results of tag state estimations using the flocking and drift localization algorithms in simulations are seen in Figures 4, 5, 6 and 7.

## V. EXPERIMENTAL IMPLEMENTATION

The same setup made in the simulations shown in Figure 3 was built using real drones. The drones shown in Figures 8, and 9 used Odroid computers as the companion computers, PX4 as a flight controller, and LIDAR was used for altitude control. The motion capture system was used to provide ground truth data, which will be used for the comparison of the drift and flocking localization algorithms. Also, the motion capture system was used by the tag drone to provide the

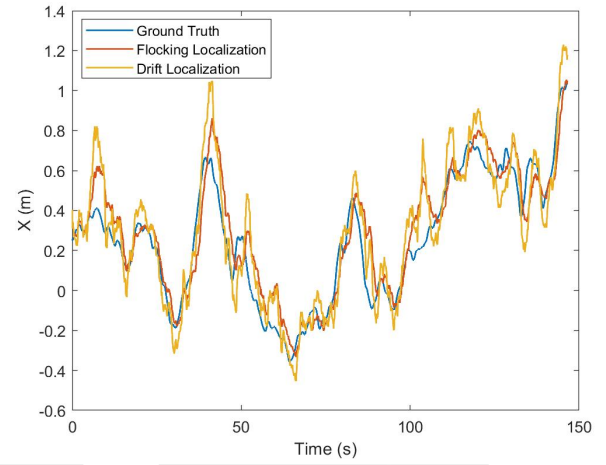


Fig. 10. X position estimation using the drift and flocking localization algorithms.

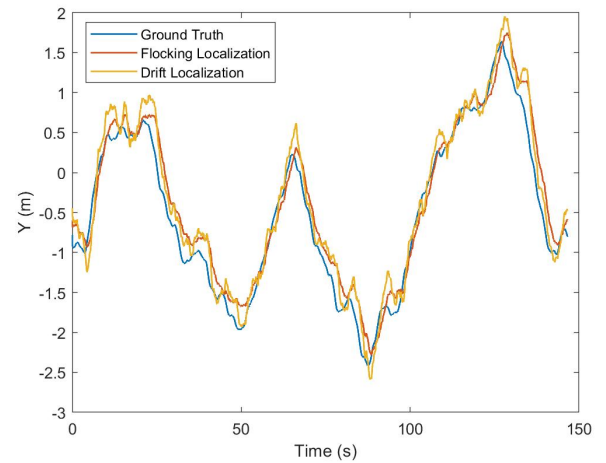


Fig. 11. Y position estimation using the drift and flocking localization algorithms.

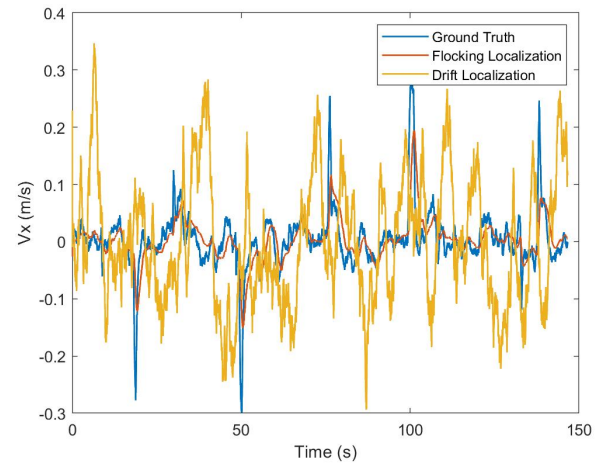


Fig. 12. X component of tag drone velocity estimated using the drift and flocking localization algorithms.

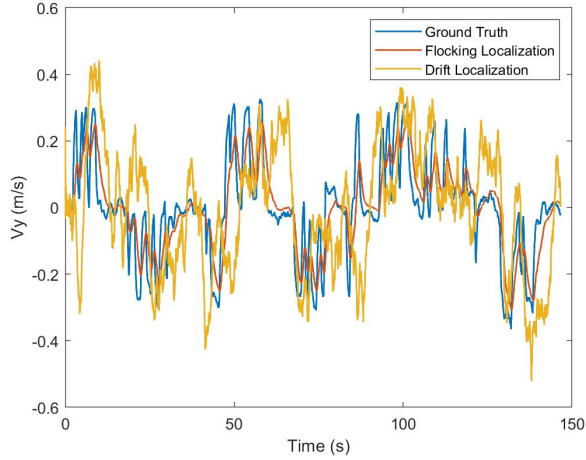


Fig. 13. Y component of tag drone velocity estimated using the drift and flocking Localization algorithms.

TABLE I

EFFECT OF CHANGING THE STANDARD DEVIATION  $\sigma_f$  IN THE MODEL OF FLOCKING LOCALIZATION ALGORITHM

$\sigma_f$	RMSE in Position Estimate (m)		RMSE in velocity Estimate (m/s)	
	$x_t$	$y_t$	$v_{tx}$	$v_{ty}$
0.001	0.0456	0.0646	1.3981	1.4355
0.01	0.0579	0.0709	1.4026	1.4474
0.1	0.1097	0.1306	1.4585	1.5680
1	0.1136	0.1352	3.5964	4.1857
10	0.1136	0.1353	32.4112	36.3719

position feedback to ensure that it exhibits flocking behavior by following the anchor drone. The 3 Decawave 1000 UWB sensors set as anchors were attached on the anchor drone at positions  $[0.35m, 0]^T$ ,  $[0, 0]^T$ , and  $[0, 0.35m]^T$ ; these sensors were used to localize another Decawave 1000 UWB sensor set as a tag mounted on the center of the tag drone.

The parameters of the Kalman filter were changed in both the drift and flocking localization algorithms to see how they affect the accuracy of the tag drone state estimations. The state estimations of the tag drone produced by the drift and flocking localization algorithms were recorded along with other relevant data (to be used) for analysis using the rosbags. Figures 10, 11, 12, and 13 show the state estimation obtained using the drift and flocking localization algorithm. Also, the figures show the plots of the real states of the drone obtained from the motion capture system and the flight controller.

## VI. DISCUSSION

First, we will discuss the effect of changing the standard deviation,  $\sigma_f$ , that is used to parameterize the uncertainty of the dynamics of the tag drone in Equation (10). From Table I, it was observed that: (i) high  $\sigma_f$  results in high root mean squared error (RMSE) values in state estimation, (ii) setting  $\sigma_f$  value low is equivalent to assuming that the tag drone is in Flocking mode with high confidence, and (iii) since the

TABLE II

EFFECT OF CHANGING THE SCALAR MULTIPLE OF THE ERROR COVARIANCE MATRIX IN THE TAG DRONE STATE ESTIMATIONS

$\sigma_p$	RMSE in Position Estimate (m)		RMSE in velocity Estimate (m/s)	
	Flocking	Drift	Flocking	Drift
3.1622	0.2299	0.2253	0.1069	0.2324
1	0.2131	0.2449	0.0953	0.2196
0.3162	0.2307	0.2310	0.0882	0.2254

TABLE III

EFFECT OF CHANGING THE SCALAR MULTIPLE OF THE PROCESS NOISE COVARIANCE MATRIX IN THE TAG DRONE STATE ESTIMATIONS

$\sigma_q$	RMSE in Position Estimate (m)		RMSE in velocity Estimate (m/s)	
	Flocking	Drift	Flocking	Drift
0.001	0.2131	0.2448	0.0953	0.2196
0.01	0.2197	0.2458	0.1125	0.2883
0.1	0.3228	0.3244	0.1127	0.3439

tag drone is actually in Flocking mode in the experiments, setting the value of  $\sigma_f$  low, gives the better state estimations.

Now, we will discuss the effect of changing the different parameters of the Kalman filter in turn while keeping other parameters constant.

By observing Table II, it was found that changing the multiplier of the error covariance matrix,  $\sigma_p$ , did not cause much difference in the value of the RMSE of the tag drone estimations when the other parameters of the Kalman filter were kept constant. However, it was observed that the flocking localization scheme produced estimates with lower RMSE compared to the drift localization scheme.

As seen in Table III, changing the values of the multiplier of the process noise covariance matrix,  $\sigma_q$  had the following effects on the localization performance: (i) when the values of  $\sigma_q$  are set low, the flocking localization scheme produced very accurate state estimations compared to the drift localization scheme, and (ii) high values of  $\sigma_q$  resulted in reliance on noisy measurements for state estimation, and thus reducing accuracy for both of the localization algorithms.

In Table IV, modifying the value of the multiplier of the observation covariance matrix,  $\sigma_r$ , also had effects on the localization performance: (i) low  $\sigma_r$  values result in high RMSE values in the state estimates, (ii) the flocking localization algorithm produced better state estimates compared to the drift localization scheme.

TABLE IV

EFFECT OF CHANGING THE SCALAR MULTIPLE OF THE OBSERVATION NOISE COVARIANCE MATRIX IN THE TAG DRONE STATE ESTIMATIONS

$\sigma_r$	RMSE in Position Estimate (m)		RMSE in velocity Estimate (m/s)	
	Flocking	Drift	Flocking	Drift
0.005	0.2235	0.2359	0.1152	0.2719
0.05	0.2131	0.2448	0.0953	0.2196

to the drift localization when  $\sigma_r$  was set high, and (iii) the best performance was obtained when  $\sigma_r$  was the same as the actual value of noise used to simulate the UWB sensors in the simulation.

## VII. CONCLUSION

We have proposed a real-time, distributed localization algorithm that takes into account the flocking behavior of robots present in MRS. The flocking localization algorithm was shown to produce more accurate state estimations of the tag robot compared to the drift localization algorithm which does not take into account the flocking behavior of the robots in the MRS. The localization algorithms were shown to depend only on the UWB sensors mounted on the robots and on-board computational units. There was no explicit communication between the robots. Also, the algorithm does not depend on the fixed infrastructure like the Mocap system and GPS, and thus makes it perfect for both indoor and outdoor applications. The flocking localization algorithm produced very accurate state estimations, which can be reliably used as feedback for different systems, especially when the system does not involve very aggressive maneuvers like in collaborative lifting. The possible extensions are to generalize the algorithm to 3D position estimation and to incorporate visual systems in the estimation of the states of the tag drone.

## REFERENCES

- [1] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427–450, 2009.
- [2] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal processing magazine*, vol. 22, no. 4, pp. 41–53, 2005.
- [3] C. J. Hegarty, "The global positioning system (GPS)," in *Springer Handbook of Global Navigation Satellite Systems*. Springer, 2017, pp. 197–218.
- [4] —, "Gnss signals—an overview," in *2012 IEEE International Frequency Control Symposium Proceedings*. IEEE, 2012, pp. 1–7.
- [5] K. Guo, Z. Qiu, W. Meng, L. Xie, and R. Teo, "Ultra-wideband based cooperative relative localization algorithm and experiments for multiple unmanned aerial vehicles in gps denied environments," *International Journal of Micro Air Vehicles*, vol. 9, no. 3, pp. 169–186, 2017.
- [6] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [7] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.
- [8] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *ISMAR*, vol. 11, no. 2011, 2011, pp. 127–136.
- [9] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-d mapping with rgb-d sensor," *Robotics, IEEE Transactions on*, vol. 23, no. 1, 2014.
- [10] C. Wang, J. Yuan, and L. Xie, "Non-iterative slam," in *2017 18th International Conference on Advanced Robotics (ICAR)*. IEEE, 2017, pp. 83–90.
- [11] S. Lowry and H. Andreasson, "Visual place recognition techniques for pose estimation in changing environments," in *Visual Place Recognition: What is it Good For? workshop, Robotics: Science and Systems (RSS) 2016, Ann Arbor, MI, USA, 19 June, 2016*, 2016.
- [12] A. Glover, W. Maddern, M. Warren, S. Reid, M. Milford, and G. Wyeth, "Openfabmap: An open source toolbox for appearance-based loop closure detection," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 4730–4735.
- [13] C. Wang, H. Zhang, T.-M. Nguyen, and L. Xie, "Ultra-wideband aided fast localization and mapping system," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1602–1609.
- [14] S. S. Kia, S. Rounds, and S. Martinez, "Cooperative localization for mobile agents: A recursive decentralized algorithm based on Kalman-filter decoupling," *IEEE Control Systems Magazine*, vol. 36, no. 2, pp. 86–101, 2016.
- [15] S. S. K. S. Martinez, "A partially decentralized EKF scheme for cooperative localization over unreliable communication links," *arXiv preprint arXiv:1608.00609*, 2016.
- [16] Z. Sahinoglu, S. Gezici, and I. Güvenc, *Ultra-wideband Positioning Systems: Theoretical Limits, Ranging Algorithms, and Protocols*. Cambridge University Press, 2008. [Online]. Available: <https://books.google.com.tr/books?id=ygXzSEzQy4oC>
- [17] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 70–84, 2005.
- [18] M. Z. Win and R. A. Scholtz, "On the robustness of ultra-wide bandwidth signals in dense multipath environments," *IEEE Communications letters*, vol. 2, no. 2, pp. 51–53, 1998.
- [19] M. Yavari, "Indoor real-time positioning using ultra-wideband technology," *Master's thesis, University of New Brunswick*, 2015.
- [20] G. Ortiz, F. Treven, L. Svensson, P. Larsson-Edefors, and S. Johansson-Mauricio, "A framework for a relative real-time tracking system based on ultra-wideband technology," in *2017 14th Workshop on Positioning, Navigation and Communications (WPNC)*. IEEE, 2017, pp. 1–6.
- [21] S. Güler, J. Jiang, A. A. Alghamdi, R. I. Masoud, and J. S. Shamma, "Real time onboard ultrawideband localization scheme for an autonomous two-robot system," in *2018 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2018, pp. 1151–1158.
- [22] S. Güler, M. Abdelkader, and J. S. Shamma, "Infrastructure-free multi-robot localization with ultrawideband sensors," in *2019 American Control Conference (ACC)*, 2019, pp. 13–18.
- [23] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.
- [24] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Ieee, 2007, pp. 153–158.
- [25] A. Zulu and S. John, "A review of control algorithms for autonomous quadrotors," *arXiv preprint arXiv:1602.02622*, 2016.
- [26] G. Strang, G. Strang, G. Strang, and G. Strang, *Introduction to linear algebra*. Wellesley-Cambridge Press Wellesley, MA, 1993, vol. 3.