Three-dimensional relative localization and synchronized movement with wireless ranging

Three-dimensional relative localization and synchronized movement with wireless ranging

Sven Pfeiffer[1] **Veronica Munaro**[2] · **Shushuai Li**[1] · **Alessandro Rizzo**[3] · **Guido C. H. E. de Croon**[1]

Received: 29 December 2021 / Accepted:

Abstract
Relative localization is a key capability for autonomous robot swarms, and it is a substantial challenge, especially for small flying robots, as they are extremely restricted in terms of sensors and processing while other robots may be located anywhere around them in three-dimensional space. In this article, we generalize wireless ranging-based relative localization to three dimensions. In particular, we show that robots can localize others in three dimensions by ranging to each other and only exchanging body velocities and yaw rates. We perform a nonlinear observability analysis, investigating the observability of relative locations for different cases. Furthermore, we show both in simulation and with real-world experiments that the proposed method can be used for successfully achieving various swarm behaviours. In order to demonstrate the method's generality, we demonstrate it both on tiny quadrotors and lightweight flapping wing robots.

Keywords Relative localization · Swarm robotics · Autonomous flight · Observability analysis

✉ Sven Pfeiffer
s.u.pfeiffer@tudelft.nl

Veronica Munaro
veronica.munaro22@gmail.com

Shushuai Li
s.li-6@tudelft.nl

Alessandro Rizzo
alessandro.rizzo@polito.it

Guido C. H. E. de Croon
G.C.H.E.deCroon@tudelft.nl

[1] Faculty of Aerospace Engineering, Technische Universiteit Delft, Kluyverweg 1, 2629HS Delft, The Netherlands

[2] Department of Control and Computer Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

[3] Department of Electronics and Telecommunications, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

1 Introduction

Following recent technological developments, autonomous micro-air vehicles (MAVs) have been getting increasing attention for a wide variety of tasks. These range from monitoring crops (Maes & Steppe, 2019) and inspecting power plants (Jordan et al., 2018) to acting as communication relays in areas with damaged infrastructure (Kwon & Hailes, 2014). For many of these tasks, using multiple collaborating agents has advantages in terms of performance, robustness and flexibility. These robotic swarms are inspired by self-organizing behaviour in animals and have the potential to complete complex tasks by exploiting cooperation among team members (Brambilla et al., 2013; Şahin, 2005). It is important here to emphasize that we do not equate 'swarming' with having thousands of robots in a tight space. For example, in exploration tasks, swarm members should spread out as much as possible over the area of interest; hence, there may well be only small groups in the order of five or ten robots on a floor of a building.

Relative localization of other agents is a key component of MAVs operating in swarms (Zhu & Kia, 2019). It is required for a variety of tasks, from avoiding collisions between swarm members to enabling complex swarming behaviours such as flocking, and a multitude of options are available when it comes to sensors being used (Coppola et al., 2020). In outdoor areas and in adequately equipped indoor environments, agents can communicate absolute positioning information, for example, from GNSS (Global Navigation Satellite System) (Vásárhelyi et al., 2018), motion-tracking cameras (Preiss et al., 2017) or beacon based systems (Ledergerber et al., 2015; Vedder et al., 2015). In GNSS-denied environments that cannot be prepared in advance, cameras (Faigl et al., 2013; Carrio et al., 2018), infrared sensors (Roberts et al., 2012) and microphones (Tijs et al., 2010) have been used for relative localization with varied success.

Based on knowledge of the distances between all swarm members, the morphology of a rigid swarm can be determined with the exception of rotation and translation. If the position of 4 nodes (in 3D) is known in a common reference frame, the network localization problem can be solved completely, yielding the position of each node in the same reference frame (Eren et al., 2004; Shang & Ruml, 2004).

In our work, we are interested in building swarms based on small and lightweight agents. This is not only advantageous for monetary and logistic reasons, but it also makes the system inherently safe. Simply put, a tiny drone can't build enough momentum to cause serious damage to people, infrastructure or other assets in the operational area. The drones we are working with in this research weigh ≤ 100 g (including battery) and do most of their processing on a STM32F4 series microcontroller (192 kB memory, 168 MHz processor). As a result, most of the previously mentioned strategies are difficult to implement due to the weight of the sensors or the computational requirements of the processing. A relative localization approach based on odometry and ranging between agents, while being less accurate, can be implemented with lightweight and energy-efficient hardware such as Ultra-Wideband (UWB) communication chips.

Some relevant work in this direction has been carried out by Trawny et al. (2010) and Trawny et al. (2007), where the authors aim at finding a solution for the 6-degree of freedom relative pose estimation problem in 3D based on odometry and inter-agent ranging. In their work, they also include a nonlinear local weak observability analysis to find sufficient conditions for the 3D relative pose to become locally weakly observable. However, their theoretical approach relies on a large set of distance measurements (10), which results in a computationally heavy polynomial system of equations.

More recently, several authors have worked on performing relative localization more efficiently, so that it can be used on smaller MAVs. For example, Coppola et al. (2018) implemented a relative localization algorithm for collision avoidance based on signal strength measurements with Bluetooth. The use of received signal strength (RSSI) and Bluetooth does, however, impose significant limitations on the range and accuracy of the system. Guo et al. (2017) present a 2D localization system based on UWB that does not suffer from these limitations, but requires the agents to determine their displacement in a common frame of reference. Li et al. (2021) on the other hand, use an Extended Kalman Filter (EKF) and UWB ranging to estimate relative position and heading in 2D, removing the need for a common orientation reference. While the above-mentioned, real-world studies on flying robots have focused on 2D localization, where the robots measured and controlled their height to a flat ground surface, some recent efforts have been made to perform relative localization in 3D as well. Cossette et al. (2021) present a ranging-based 3D relative localization scheme, which does, however, require a common orientation reference. The same goes for Shalaby et al. (2021), which additionally makes use of multiple UWB tags per agent. Using a common orientation reference requires exteroceptive sensors which often require proper initialization and can be subject to external disturbances. Specifically, the magnetometers used in the aforementioned papers are affected when operating in proximity of hard iron objects or electrical transmission lines. For higher robustness in GPS-denied, unprepared environments, the 3D relative localization without common orientation reference has been studied, using multiple consecutive distance measurements and computationally expensive optimization algorithms (Jiang et al., 2020; Ziegler et al., 2021; Nguyen & Xie, 2022).

In this work, we perform wireless-ranging-based relative localization for drones in three dimensions, without the need for a common orientation reference or extensive computational resources. With Lie derivatives (Williams & Sukhatme, 2015; Heintzman & Williams, 2020), we study the system observability and find favourable trajectories with high localization observability. Our contributions are thus:

- A formulation for a wireless ranging-based 3D relative localization system which does not require a common orientation reference and can be solved efficiently on computationally limited devices.
- An observability analysis of the system
- Experimental validation and comparison of open- and closed-loop performance
- The first experimental results of relative localization on flapping wing drones

This paper is structured as follows: In Sect. 2, we introduce the relative localization problem, the sensory information available to the drones, the continuous-time model and the EKF that performs the estimation. Section 3 presents the observability analysis carried out to evaluate the quality of the information provided by the filter and to detect conditions that may cause a degradation of the degree of observability of the system. The theoretical analysis from Sect. 3 is verified by means of simulation in Sect. 4 and by means of real-world robotic experiments in Sect. 5 and 6. Finally, we draw conclusions in Sect. 7.

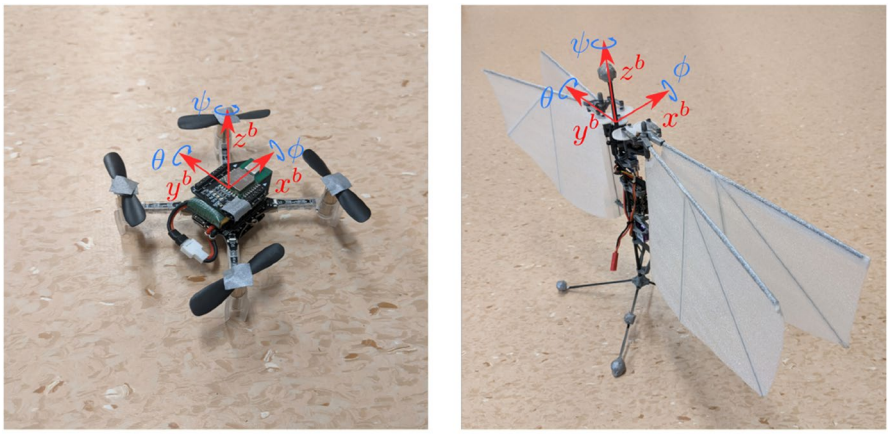


Fig. 1 Illustration of the coordinate system conventions shown on the drones used in this paper

2 Ranging-based three-dimensional relative localization

2.1 Notation and conventions

Let us first introduce some notation and conventions. Throughout this paper, we use right-handed coordinate systems with the x-axis pointing forward, the z-axis pointing up and the y-axis completing the right-handed system as shown in Fig. 1. When expressing the orientation of the MAV in Tait-Bryan angles (roll ϕ, pitch θ, yaw ψ), we follow the sequence ZYX from global to body frame, leading to the following expression for the rotation matrix:

$$\boldsymbol{u}^G = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi)\boldsymbol{u}^B = \mathbf{R}_{gb}\boldsymbol{u}^B \tag{1}$$

$$\mathbf{R}_{gb} = \begin{bmatrix} \mathrm{c}\psi\mathrm{c}\theta & \mathrm{c}\psi\mathrm{s}\theta\mathrm{s}\phi - \mathrm{s}\psi\mathrm{c}\phi & \mathrm{c}\psi\mathrm{s}\theta\mathrm{c}\phi + \mathrm{s}\psi\mathrm{s}\phi \\ \mathrm{s}\psi\mathrm{c}\theta & \mathrm{s}\psi\mathrm{s}\theta\mathrm{s}\phi + \mathrm{c}\psi\mathrm{c}\phi & \mathrm{s}\psi\mathrm{s}\theta\mathrm{c}\phi - \mathrm{c}\psi\mathrm{s}\phi \\ -\mathrm{s}\theta & \mathrm{c}\theta\mathrm{s}\phi & \mathrm{c}\theta\mathrm{c}\phi \end{bmatrix} \tag{2}$$

$\mathbf{R}_x(\phi)$, $\mathbf{R}_y(\theta)$ and $\mathbf{R}_z(\psi)$ are the rotation matrices for the elemental rotations around the local x, y and z axis, respectively, and we use shorthands for sines and cosines ($\mathrm{s}x = \sin x, \mathrm{c}x = \cos x$).

Since there is no common inertial reference frame, each agent i must be able to determine the relative position of the other robots in a body centred frame. However, instead of working in the drone's body frame B_i, we mainly work in a drone-centred horizontal frame H_i. In this horizontal frame, the xy-plane is parallel to the ground and the z-axis points upward. Since roll and pitch can be accurately estimated from IMU data thanks to the gravitational force (Mahony et al., 2008), any vector $\mathbf{u}^B$ expressed in B_i can be expressed in H_i by multiplication with the following rotation matrix:

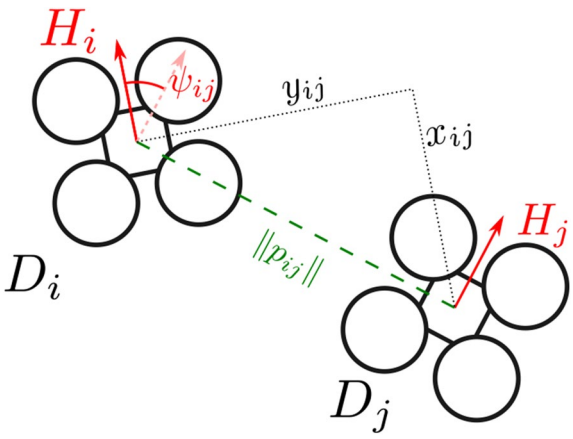


Fig. 2 Relative localization problem, seen from above

$$\mathbf{R}_{hb} = \begin{bmatrix} c\theta & s\theta s\phi & s\theta c\phi \\ 0 & c\phi & -s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{3}$$

2.2 System model

The 3D relative localization problem can be described as determining the 3D relative position of a tracked drone D_j in the body frame of a tracking drone D_i, $\{j | j \in \mathbb{N}, j \neq i\}$ where $\mathbb{N} = \{1, 2, ..., N\}$ and N is the number of robots in the flock within sight of D_i. We define the relative state vector for the tracking drone as $\mathbf{X}_{ij} = \left[x_{ij}, y_{ij}, z_{ij}, \psi_{ij}\right]^T$, where $\mathbf{p}_{ij} = \left[x_{ij}, y_{ij}, z_{ij}\right]^T$ is the relative position along the three axes of the tracking drone's horizontal frame H_i and ψ_{ij} is the relative yaw. The relative localization problem and its associated variables are illustrated in Fig. 2.

The continuous-time kinematic model $f(\mathbf{X}_{ij})$ describing the relative motion of the drones in the three dimensions can be derived using the Newton formula (Li et al., 2021). The full model, including the measurement equation, is then:

$$\dot{\mathbf{X}}_{ij} = f(\mathbf{X}_{ij}) = \begin{bmatrix} \mathbf{R}_z(\psi_{ij}) \boldsymbol{v}_j^{H_j} - \boldsymbol{v}_i^{H_i} - \mathbf{S}\dot{\psi}_i^G \boldsymbol{p}_{ij} \\ \dot{\psi}_j^G - \dot{\psi}_i^G \end{bmatrix} \tag{4}$$

$$h(\mathbf{X}_{ij}) = \|\boldsymbol{p}_{ij}\| = \sqrt{x_{ij}^2 + y_{ij}^2 + z_{ij}^2} \tag{5}$$

$$\mathbf{S} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{6}$$

The inputs to this model are the horizontal velocities $\boldsymbol{v}_i^{H_i}$ and $\boldsymbol{v}_j^{H_i}$ (expressed in H_i) as well as the yaw rates $\dot{\psi}_i^G$ and $\dot{\psi}_j^G$ in the global frame. Each drone can estimate its own horizontal

velocity and yaw rate from on-board measurements and communicates them to the other drones by including them in the same UWB messages that are used for ranging. Furthermore, the ranging messages are used to calculate a measurement of the distance between the drones, d_{ij}. Estimation of the horizontal velocity can vary depending on the drone platform and available on-board sensors. Two examples are presented in Sect. 5.2. The flow of data on-board and between the drones is shown in Fig. 3.

Note that the yaw rate $\dot{\psi}$ is defined around an intermediate axis of rotation and does therefore not directly correspond to the angular velocity around the z axis, ω_z, measured on-board the drone. Specifically, the relation between the angular rates around the body frame axes, $\boldsymbol{\omega}$, and the Euler angle rates (ZYX sequence) are given by:

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_x^{-1}(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{R}_x^{-1}(\phi)\mathbf{R}_y^{-1}(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \tag{7}$$

This relation can be expressed using a single matrix, which is easily inverted to find the

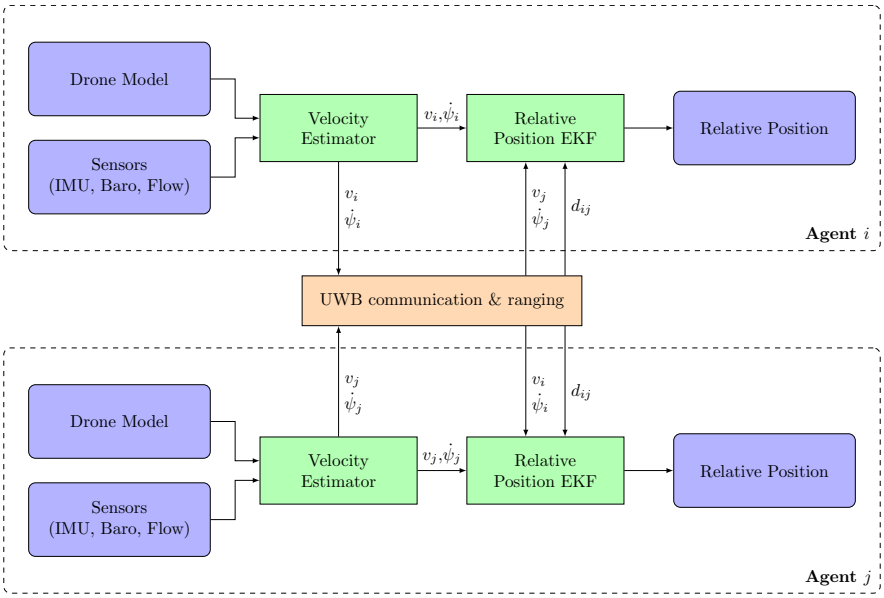
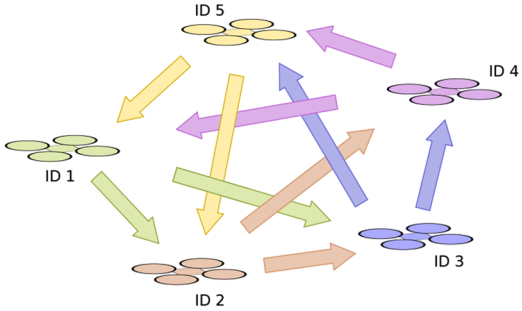


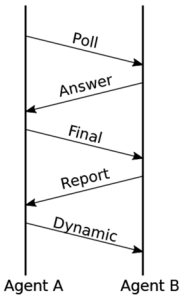
Fig. 3 Data flow on-board and between drones

expression for the Euler angle rates in terms of the gyro measurements:

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\mathrm{s}\theta \\ 0 & \mathrm{c}\phi & \mathrm{s}\phi\mathrm{c}\theta \\ 0 & -\mathrm{s}\phi & \mathrm{c}\phi\mathrm{c}\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \leftrightarrow \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \frac{\mathrm{s}\theta\mathrm{s}\phi}{\mathrm{c}\theta} & \frac{\mathrm{s}\theta\mathrm{c}\phi}{\mathrm{c}\theta} \\ 0 & \mathrm{c}\phi & -\mathrm{s}\phi \\ 0 & \frac{\mathrm{s}\phi}{\mathrm{c}\theta} & \frac{\mathrm{c}\phi}{\mathrm{c}\theta} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{8}$$



(a) High-Level Communication Control



(b) Low-Level Ranging

Fig. 4 Ultra-wideband communication protocol

2.3 EKF for relative localization

To estimate the relative state vector $\boldsymbol{X}_{ij}$ using the system model (4) and the measurement Eq. (5), an EKF is used (Simon, 2006). For the prediction step, we discretize (4) using the forward Euler method and linearize the system for the covariance update. In what follows, T_s is the sampling time, $\mathbf{P}$ is the state covariance matrix, and $\mathbf{Q}$ is the process noise covariance matrix. The input vector $\mathbf{U}$ is defined as $\mathbf{U} = [\boldsymbol{v}_i^T, \boldsymbol{v}_j^T, \dot{\psi}_i, \dot{\psi}_j]^T$. To simplify notation, we drop the indices ij for the state vector $\boldsymbol{X}$.

$$\begin{aligned} \hat{\mathbf{X}}_{k+1|k} &= F(\hat{\mathbf{X}}_k, \mathbf{U}_k) = \hat{\mathbf{X}}_k + \dot{\mathbf{X}}_k T_s \\ \mathbf{P}_{k+1|k} &= \mathbf{A}_k \mathbf{P}_{k|k} \mathbf{A}_k^T + \mathbf{B}_k \mathbf{Q} \mathbf{B}_k^T \end{aligned} \tag{9}$$

with

$$\mathbf{A} = \frac{\partial F}{\partial \mathbf{X}} = \begin{bmatrix} 1 & \dot{\psi}_i T_s & 0 & T_s(-\mathrm{s}\psi_{ij}\, v_j^x - \mathrm{c}\psi_{ij}\, v_j^y) \\ -\dot{\psi}_i T_s & 1 & 0 & T_s(\mathrm{c}\psi_{ij}\, v_j^x - \mathrm{s}\psi_{ij}\, v_j^y) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

$$\mathbf{B} = \frac{\partial F}{\partial \mathbf{U}} = \begin{bmatrix} -T_s & 0 & 0 & T_s \mathrm{c}\psi_{ij} & -T_s \mathrm{s}\psi_{ij} & 0 & T_s y_{ij} & 0 \\ 0 & -T_s & 0 & T_s \mathrm{s}\psi_{ij} & T_s \mathrm{c}\psi_{ij} & 0 & -T_s x_{ij} & 0 \\ 0 & 0 & -T_s & 0 & 0 & T_s & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -T_s & T_s \end{bmatrix} \tag{11}$$

The UWB measurement of the distance between the drones is the only measurement considered for the correction step of the EKF. The Jacobian matrix of the corresponding measurement Eq. (5) is given by

$$\mathbf{H} = \frac{\partial h}{\partial \mathbf{X}} = \frac{\boldsymbol{p}_{ij}^T}{\|\boldsymbol{p}_{ij}\|} = \frac{1}{\sqrt{x_{ij}^2 + y_{ij}^2 + z_{ij}^2}} \cdot [x_{ij}, y_{ij}, z_{ij}] \tag{12}$$

Given a noisy distance measurement d_{ij} with variance σ_d^2, it is then possible to calculate the Kalman gain $\boldsymbol{K}$ and apply the measurement update to the state vector and the state covariance matrix. Note that since we work with a scalar measurement, no matrix inversion is necessary to compute the Kalman gain.

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \sigma_d^2\right)^{-1} \tag{13}$$

$$\hat{\mathbf{X}}_k = \hat{\mathbf{X}}_{k|k-1} + \mathbf{K}_k\left(d_{ij} - h\left(\hat{\mathbf{X}}_{k|k-1}\right)\right) \tag{14}$$

To use this approach with multiple drones, a filter bank of several EKFs that run in parallel can be used. This approach was demonstrated successfully for a swarm of five drones in 2D (Li et al., 2021). While this approach does not leverage additional information from distances between other drones, it is highly flexible and thus suited for dynamic swarms. Furthermore, this approach is computationally efficient as there is no need to track covariances between the relative states, preventing large matrix operations.

2.4 Ultra-wideband communication protocol

In large part, the UWB communication protocol used is the same as described in Li et al. (2021) and shown in Fig. 4.

On a high level, every agent has its own ID and ranges to a set of predetermined other agents. After ranging and exchanging information with those agents, control of the communication channel is passed on to the next agent, which then performs its own ranging and communication. By sharing ranging information with the agent that is being ranged to, each link only needs to be initiated from one side and the control can be passed around in a circle.

The lower-level ranging protocol is an extension to the standard symmetrical double-sided two-way ranging (SDS-TWR). The "Report" message now also includes the reporting drone's velocity and yaw rate estimates. Furthermore, an additional "dynamic" message is returned by the agent that initiated the ranging which also transmits velocity estimates and yaw rate alongside the calculated range. Finally, the "Dynamic" message is also used to pass control of the communication channel to the next agent.

3 Observability analysis

It is important to be aware of the potential dangers resulting from rapidly degrading relative position estimates in a robotic swarm. This can result from a lack of observability of the system, that is, when the combination of inputs and measurements is not sufficient to infer the full state of the system. To identify these situations, we proceed with an observability analysis of the system.

Based on Arrichiello et al. (2013) and Hermann and Krener (1977), we study the *local weak observability* of the system described by (4) and (5) in Lie derivatives, by computing the observability matrix $\mathbf{O}$. The system is termed observable if $\mathbf{O}$ is full rank. To simplify the expressions in this chapter, we consider the measurement Eq. (5) in power form, i.e.

$$h(\mathbf{X}) = \frac{\boldsymbol{p}_{ij}^T \boldsymbol{p}_{ij}}{2} = \frac{1}{2}\left(x_{ij}^2 + y_{ij}^2 + z_{ij}^2\right) \tag{15}$$

hence,

$$\mathbf{O} = \begin{bmatrix} \nabla \mathcal{L}_f^0 h \\ \nabla \mathcal{L}_f^1 h \\ \nabla \mathcal{L}_f^2 h \\ \nabla \mathcal{L}_f^3 h \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_{ij} & 0 \\ (\mathbf{R}_z \boldsymbol{v}_j - \boldsymbol{v}_i)^T & \mathbf{p}_{ij}^T \mathbf{R}_z \mathbf{S} \mathbf{v}_j \\ (\dot{\psi}_j \mathbf{R}_z \mathbf{S} \boldsymbol{v}_j - \dot{\psi}_i \mathbf{S} \boldsymbol{v}_i)^T & -2\boldsymbol{v}_i^T \mathbf{R}_z \mathbf{S} \boldsymbol{v}_j + \dot{\psi}_j \boldsymbol{p}_{ij}^T \mathbf{R}_z \mathbf{S}^2 \boldsymbol{v}_j \\ (\dot{\psi}_j^2 \mathbf{R}_z \mathbf{S}^2 \boldsymbol{v}_j - \dot{\psi}_i^2 \mathbf{S}^2 \boldsymbol{v}_i)^T & -3(\dot{\psi}_j - \dot{\psi}_i)\boldsymbol{v}_i^T \mathbf{R}_z \mathbf{S}^2 \boldsymbol{v}_j - \dot{\psi}_j^2 \boldsymbol{p}_{ij}^T \mathbf{R}_z \mathbf{S} \boldsymbol{v}_j \end{bmatrix} \tag{16}$$

Without delving into an exhaustive analysis of all the states that would cause matrix [16] to lose its rank, some unobservable conditions can be highlighted by observing its structure and computing its determinant $|\mathbf{O}|$. While more conditions exist, they involve more complicated combinations of states and inputs that are unlikely to occur for longer time spans anyway. Specifically, the following conditions will result in a full row or column of zeros which in turn will result in $|\mathbf{O}| = 0$. This can easily be seen by performing a cofactor expansion following a particular row or column:

- *Identical velocities,* $\boldsymbol{v}_i = \mathbf{R}_z \boldsymbol{v}_j$
  This condition could potentially be problematic for missions that require leader–follower or flocking behaviour, where the drones are expected to move at identical velocities.
- *Same altitude and no relative motion along z,* $z_{ij} = 0$ and $v_i^z = v_j^z$
  Similar to the condition of identical velocities, this situation could occur in multiple scenarios.
- *Tracked drone not moving,* $\mathbf{v}_j = 0$
  This is a very relevant condition, as it means that the agents should be constantly moving in order to be able to determine their relative positions.
- *Identical positions,* $\mathbf{p}_{ij} = \mathbf{0}$
  While this condition will in theory cause the system to become unobservable, it is not relevant in practical applications, where it is impossible for two agents to be located in the same position in space.

It is also interesting to evaluate not only *if*, but *how well* the system is observable. To this end, it is possible to employ the *local estimation condition number* C, defined as the ratio between the largest singular value $\sigma_{\max}$ and the smallest singular value $\sigma_{\min}$ of $\mathbf{O}$ (Krener & Ide, 2009). To avoid division by zero if $\mathbf{O}$ loses rank, we will actually consider the inverse of the local estimation condition number, C^{-1}:

$$C^{-1} = \frac{\sigma_{\min}(\mathbf{O})}{\sigma_{\max}(\mathbf{O})} \tag{17}$$

Favourable conditions on the system will result in a small C^{-1}, while an ill-conditioned estimation problem will entail a rather large C^{-1}. The inverse local estimation condition number can be used to understand how the addition of the third dimension affects the observability of the system. We do this by varying one parameter of matrix $\mathbf{O}$ at a time and comparing the values of C^{-1} when estimating relative position in x and y, as well as the relative yaw. Figure 5 shows an example of the study. It can be noticed that the maximum value of C^{-1} is significantly lower for the 3D case compared to the 2D one, which means

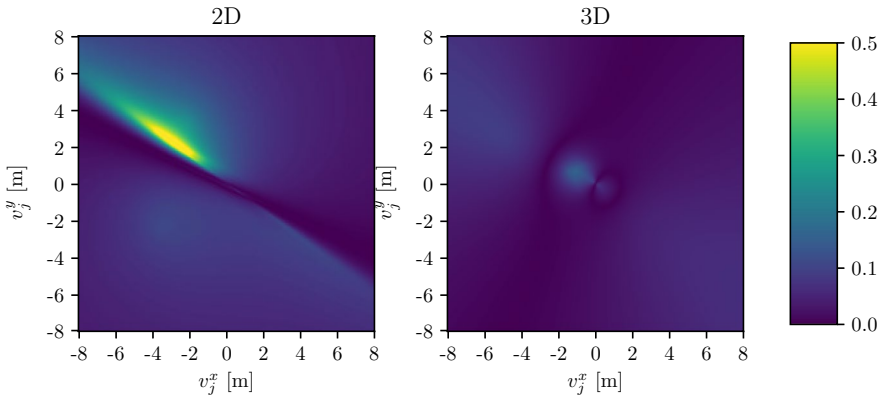


Fig. 5 Effect of v_j on the observability index. Analysis parameters $\mathbf{p}_{ij} = [2, 2, 3]^T$ m, $\mathbf{v}_i = [2, 2, 5]^T$ m/s, $v_{ij}^z = 5$ m/s $\psi_{ij} = 160°$, $\dot{\psi}_i = 10°$/s and $\dot{\psi}_j = -20°$/s

that adding the third dimension decreases the observability of the system in general. This can be explained by the higher ambiguity when determining the true states by relying on the same measurement: In 3D, any relative position on a sphere around the agent can result in a given measurement, while in 2D, this uncertainty is limited to a circle.

4 Simulation results

In this section, we discuss results obtained by simulating the multi-agent system in Python to validate the relative localization method. The EKF performance is assessed in terms of quality of estimation and consistency. For comparison, we also implemented the sliding-window estimator (SWE) presented in Cossette et al. (2021). The parameters for the EKF are computed as follows:

$$\mathbf{Q} = \text{diag}\left([\sigma_v^2, \sigma_v^2, \sigma_v^2, \sigma_v^2, \sigma_v^2, \sigma_v^2, \sigma_w^2, \sigma_w^2]\right) \tag{18}$$

$$\mathbf{R} = \sigma_r^2 \tag{19}$$

where the input noise deviation of velocity and yaw rate are, respectively, $\sigma_v = 0.15$ m/s and $\sigma_w = 0.1$ rad/s, the measurement noise deviation is $\sigma_r = 0.1$ m and the sampling time is $T_s = 0.01$ s, which is similar to the values observed in logs of the real systems.

The state vector $\mathbf{X}$ is initialized to a random initial relative position $(x_{ij}, y_{ij}, z_{ij}, \psi_{ij}$ in the vicinity of the true relative position. The covariance matrix $\mathbf{P}$ is set according to the random noise added to the initial position value. The horizontal velocity of the drones which serves as input to the EKF can stem from different sources that can affect the consistency of the estimator. The most general approach is to use a velocity that is provided from an on-board estimator. While this approach has the advantage of being applicable to almost any drone imaginable, it results in a cascaded setup of filters, which is known to lead to problems regarding the estimator's consistency (Shalaby et al., 2021). To avoid this problem, we compare the performance of the naive (simple) approach with two variations. The Sigma Point Covariance Intersection (SPCI) variant uses covariance intersection with a

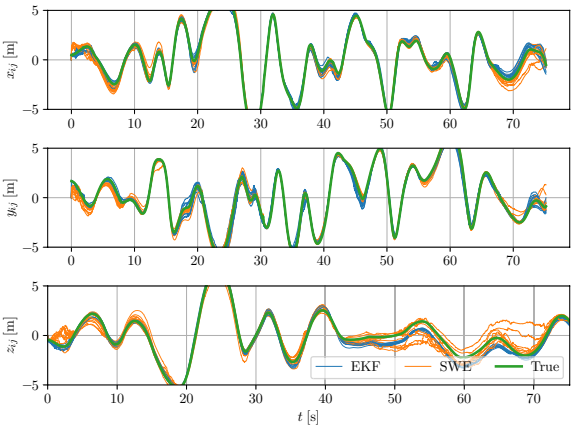


Fig. 6 Simulated performance of simple EKF and SWE for the three states x_{ij}, y_{ij}, z_{ij} for 10 runs on a random trajectory

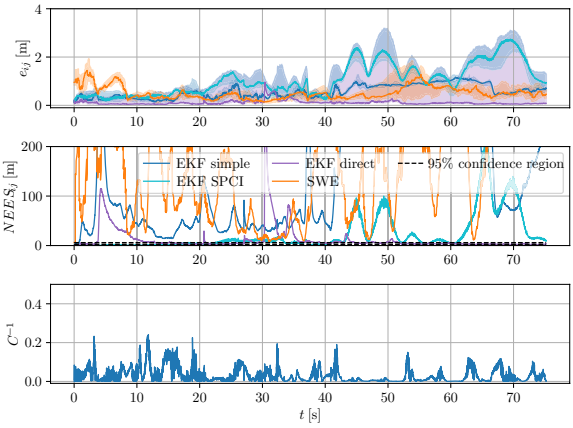


Fig. 7 Median absolute position error (e_{ij}) and Normalized-Estimation-Error-Squared ($NEES_{ij}$) over 10 runs, alongside the inverse local estimation condition number C^{-1} for random trajectories. Shaded regions indicate the lower and upper quartiles

sigma point approach to avoid having to estimate cross-covariances (Shalaby et al., 2021; Zhu & Kia, 2019). Furthermore, a direct approach can be used on platforms where direct velocity measurements are available, e.g. in the form of optic flow measurements.

For the first test, the robots are assigned a random trajectory. The test is repeated ten times with different seeds for the random number generator that is used to add noise to the inputs and measurements. Figure 6 shows the estimated relative positions of both estimators compared to the groundtruth. Furthermore, we compare the absolute relative position error e_{ij} and the normalized estimation error squared $NEES_{ij}$ to assess the performance and consistency of the estimators, as well as the inverse of the local estimation condition number C^{-1} (Fig. 7). Note that only the error in position is considered to allow a comparison between different estimators.

As can be seen, the EKF getting direct velocity measurements performs best, both in terms of error and consistency. The simple EKF using estimated velocities is highly inconsistent and does not provide the same quality of estimation. Using the SPCI approach improves the consistency in a cascaded setup, but does increase the error on this trajectory. We therefore conclude that where possible, direct velocities should be used.

It is interesting to verify the link between observability index and the filter performance. In the first test, we can already see that all filters perform better in the first half of the

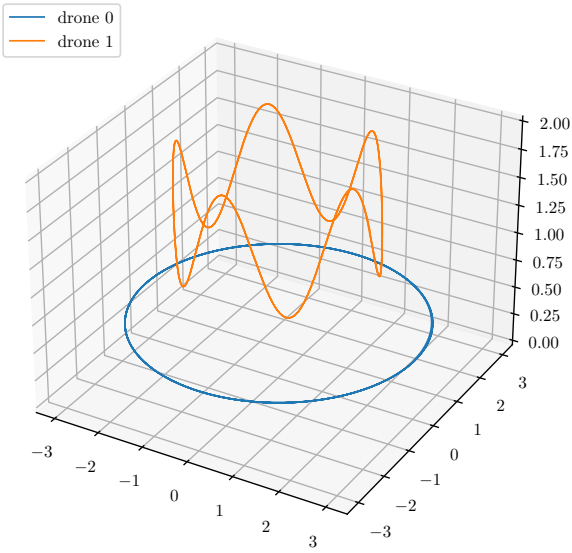


Fig. 8 Trajectories designed to maximize observability

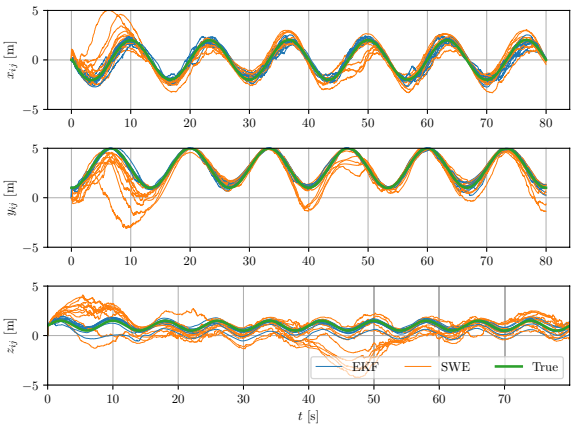


Fig. 9 Simulated performance of simple EKF and SWE for the three states x_{ij}, y_{ij}, z_{ij} for 10 runs on favourable trajectories

simulation, where the observability is better (larger values of C^{-1}). In a second test, the system is simulated with trajectories designed to maximize the observability of the system. The favourable trajectories are realized by making MAV i fly along a clockwise circular trajectory of radius ρ_i at an angular velocity ω at a height z_i, which oscillates in a sinusoidal way. MAV j, on the other hand, follows a circular path concentric to MAV i, counterclockwise, with a radius $\rho_j > \rho_i$, an angular velocity 2ω and a height $z_j \neq z_i$. With this setup, parallel velocities are avoided, as well as possible hits between the MAVs. Moreover, the difference in the velocity magnitude allows to excite the system with sufficient dynamics. The favourable trajectories are shown in Fig. 8.

The results of the second test are shown in Figs. 9 and 10. As can be seen, avoiding unobservable conditions leads to frequent high values of the inverse local estimation condition number, due to the better conditioning of matrix $\mathbf{O}$. We continue, however, to observe worse performance during times of lower observability, especially for the SPCI EKF

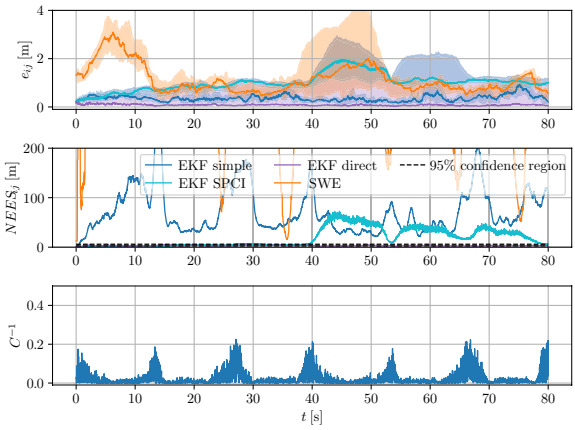


Fig. 10 Median absolute position error (e_{ij}) and Normalized-Estimation-Error-Squared ($NEES_{ij}$) over 10 runs, alongside the inverse local estimation condition number C^{-1} for favourable trajectories. Shaded regions indicate the lower and upper quartiles

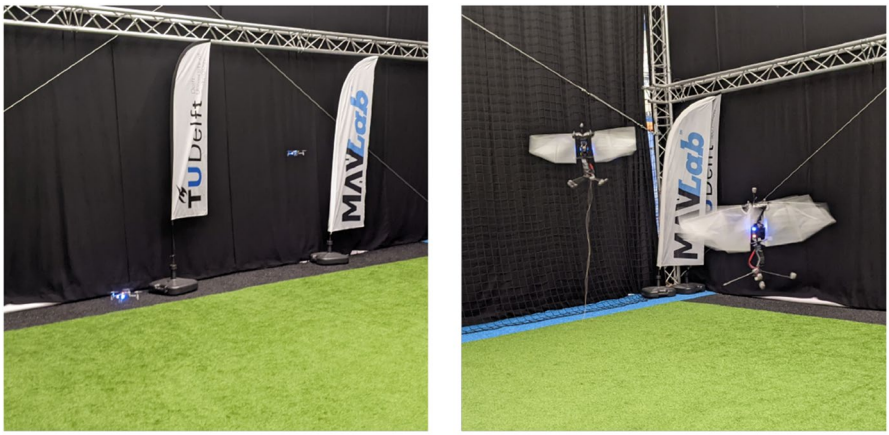


Fig. 11 Drones used during experimental testing

between 40 and 65 s, with a drop around 55 s. In addition to the improved performance, we also observe a higher consistency for all EKF variants.

The mean NEES after convergence is used as a measure of consistency. For a consistent, approximately linear Gaussian filter, the mean NEES is expected to tend towards the number of states of the filter, which in our case is three (Bailey et al., 2006). In our simulation, all estimators turn out to be overconfident to various degrees, with exception for the direct EKF on the favourable trajectory. For the simple and SPCI EKF, there is also an improved consistency on the favourable trajectory, but the NEES still stays outside of the 95% confidence interval most of the time. The advantage that the SPCI EKF offers over the simple EKF in terms of consistency does unfortunately not lead to an improved performance. Combined with the larger computational cost and the need to exchange covariance matrices between drones, this means that in its current form, the SPCI EKF is not appropriate for this problem.

5 Experimental setup

To test the proposed algorithm, we implemented our 3D relative localization on two different types of MAVs: Quadrotors (Crazyflie 2.1[1]) and flapping wing drones (Flapper Drones[2]), shown in Fig. 11. While flapping wing drones offer many potential advantages such as energy efficiency, agility and robustness to collisions, they have so far not received the same level of attention (de Croon, 2020). This might be due to additional challenges arising from the use of a flapping wing platform. Specifically, the vibrations created from the flapping of the wings results in more noise on IMU measurements. Furthermore, a large horizontal drag leads to increased pitch and roll angles. By performing tests on these vastly different platforms, we want to demonstrate the versatility and robustness of our proposed method.

5.1 Hardware

The Crazyflie is a versatile, lightweight platform with open-source hardware and software, whose capabilities can be extended by making use of expansion decks. In addition to the STM32F405 MCU which runs the firmware, it also employs a nRF51822 MCU for radio and power management. Since the Flapper Drones are based on the same Crazyflie hardware, they run almost the same firmware and are compatible with the Crazyflie expansion decks. Both types of drones are equipped with an Inertial Measurement Unit (IMU), containing a three-axis accelerometer and gyroscope (BMI088), and a barometric pressure sensor (BMP388). The Loco-Positioning Deck, which is used for UWB ranging between the drones, is based on the Decawave DWM1000 module. On the original Crazyflie, the Flow Deck V2 is used for optic flow measurements (PMW3901 optical flow sensor) and height measurements with a time-of-flight distance sensor (VL53L1x). Unfortunately, due to the orientation of the autopilot on the Flapper Drones and the large pitch and roll angles present during its flight, the Flow Deck V2 can't be employed on the Flapper Drones. As a result, it is more difficult to estimate the Flapper Drone's body velocity and altitude. To the best of our knowledge, this study is the first attempt to have flapping wing robots perform relative localization with onboard means.

5.2 Relative localization algorithm

To estimate the relative position of two drones, the EKF presented in Sect. 2 requires the yaw rate and horizontal velocities of both drones as input. Using (8), the yaw rate can be easily obtained from gyro measurements. Especially on the flapping wing drones, however, these measurements are extremely noisy and cause significant noise on the relative position estimates. We therefore filter yaw rate measurements using a second-order Butterworth filter with a cut-off frequency of 5 Hz (lower than the $\sim$ 12 Hz flapping frequency).

On the Crazyflie 2.1, we make use of the built-in EKF for on-board state estimation. The EKF relies on measurements from the IMU and the Flow Deck to estimate the drones attitude and velocity, which can then be used as inputs for the relative localization EKF after rotating them into the horizontal frame.

[1] https://www.bitcraze.io/

[2] https://flapper-drones.com/wp/

On the Flapper Drones on the other hand, using the Flow Deck is not possible due to the large vibrations and the drones architecture. Without optic flow and laser altimeter measurements, the on-board EKF cannot be used to calculate the inputs required for relative localization. Instead, we use a complementary filter for attitude estimation and predict the horizontal velocity as a function of attitude alone. For this model, we consider the drone as a rigid body subject to three principal forces acting on its centre of mass: thrust, drag and gravity. We further assume small pitch and roll angles and neglect changes in altitude to arrive at a rather simple expression for the evolution of the drone's horizontal velocity.

$$v_{x,k+1} = v_{x,k} + \mathrm{d}t\left(\tan\theta_k \cdot |g| - \cos^2\theta_k \cdot b_x v_{x,k}\right) \tag{20}$$

$$v_{y,k+1} = v_{y,k} + \mathrm{d}t\left(-\frac{\tan\phi_k}{\cos\theta_k}|g| - \cos^2\phi_k \cdot b_y v_{y,k}\right) \tag{21}$$

The drag coefficients b_x and b_y are found empirically by solving a least-squares minimization on a data set with ground truth. More details on the derivation of the model are found in Appendix A.

Since this model assumes no changes in altitude during horizontal motion, it cannot be used to estimate the vertical velocity. Instead, we fuse barometer and accelerometer data in a complementary filter following the approach of Wei et al. (2016). The resulting filter equation is given by (22), where h_{acc} is the accelerometer and h_{baro} the barometer measurement. For the implementation on the drone, we discretize the filter using the bilinear transform and a sampling period of $T_s = 0.01$ s. Note that the accelerometer measurement must of course be rotated first to correct for the pitch and roll of the drone.

$$\hat{V}_z(s) = \frac{s^2}{s^2 + k_1 s + k_2} \cdot \frac{h_{\text{acc}}(s)}{s} + \frac{k_1 s + k_2}{s^2 + k_1 s + k_2} \cdot s \cdot h_{\text{baro}}(s) \tag{22}$$

Since the estimation of v_z relies on different sensor inputs, it makes sense to assume a different standard deviation of the error, when compared to v_x and v_y. Additionally, due to different dynamics of the flapper in x and y direction, it makes sense to use individual process noise parameters for the relative localization EKF. Specifically, the process noise covariance matrix (18) is rewritten as (23) and is chosen independently for the quadrotor and for the flapping wing drone, respectively.

$$\mathbf{Q} = \mathrm{diag}\left([\sigma_{vx}^2, \sigma_{vy}^2, \sigma_{vz}^2, \sigma_{vx}^2, \sigma_{vy}^2, \sigma_{vz}^2, \sigma_w^2, \sigma_w^2]\right) \tag{23}$$

To measure the distance between two MAVs and exchange their horizontal velocity and yaw rate, we use UWB signals, employing the Crazyflie Loco-Positioning deck. The ranging and communication protocol used is the same as in Li et al. (2021), with the sole addition of also exchanging vertical velocity estimates.

The complete code used in the experiments is available on github for the crazyflies[3] and the Flapper Drones.[4]

[3] https://github.com/tudelft/crazyflie-firmware/tree/cf_swarm3d

[4] https://github.com/tudelft/crazyflie-firmware/tree/flapper_swarm3d

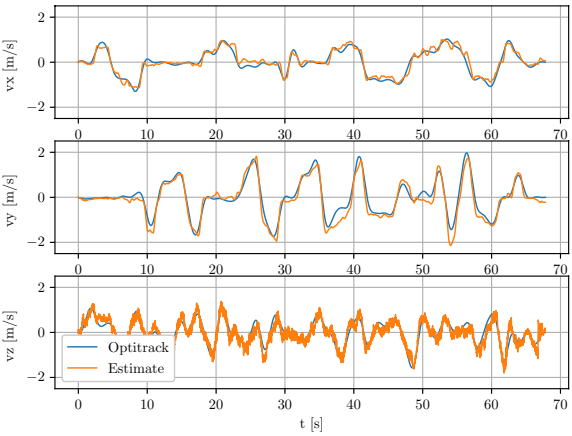


Fig. 12 Offline performance of the horizontal velocity model for the Flapper Drone when flying with varying altitude

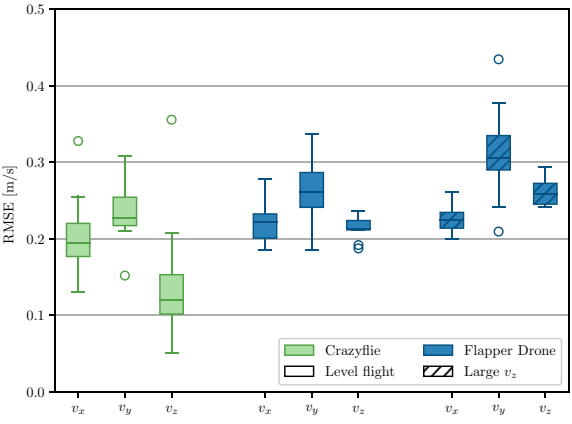


Fig. 13 RMSE of the horizontal and vertical velocity estimation on the Crazyflie 2.1 and on the Flapper Drones aggregated over 10 trajectories

5.3 Test environment

The flight tests were performed in the "Cyberzoo" test arena at the TU Delft Faculty of Aerospace Engineering. The "Cyberzoo" is a 10 m×10 m×5 m indoor arena equipped with an optitrack motion capture system that provides groundtruth with millimetre level accuracy. Data were logged from the optitrack system and from the drone using the Crazyflie Suite.[5]

[5] https://github.com/tudelft/crazyflie-suite

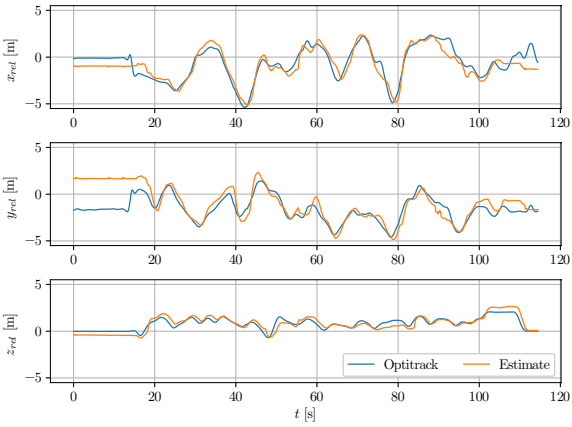


Fig. 14 Relative Localization during free flight with two Crazyflies

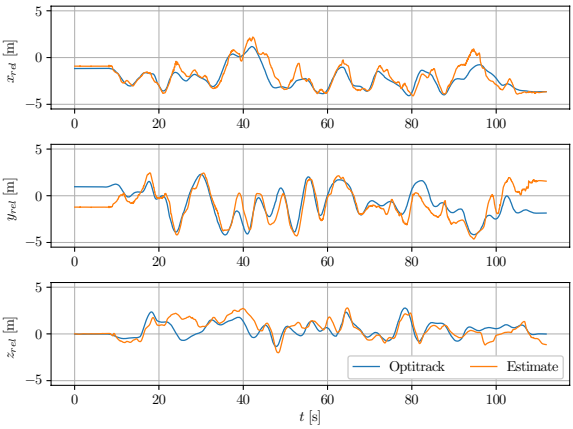


Fig. 15 Relative Localization during free flight with two Flapper Drones

6 Results

In this section, we will present the results of our flight tests. The complete data underlying the plots and analysis have been published on 4TU Research Data.[6] Videos of the experiments can be found on Youtube.[7]

6.1 Horizontal velocity estimation

Since the horizontal velocity of the drones is an important input to the relative localization EKF, it makes sense to analyse the performance of the simplified velocity model developed for the flapper. The flapper model was tuned offline on a dataset consisting of 3 trajectories, flown manually at almost constant altitude, resulting in drag coefficients $b_x = 4.2$ and $b_y = 1.8$. Using these parameters, the root mean squared error (RMSE) was 0.13 m/s for v_x and 0.22 m/s for v_y on the training set. The RMSE for

[6] https://doi.org/10.4121/17372348

[7] https://www.youtube.com/playlist?list=PL_KSX9GOn2P-dzSwBvYYRZuKOiYvx0pIS

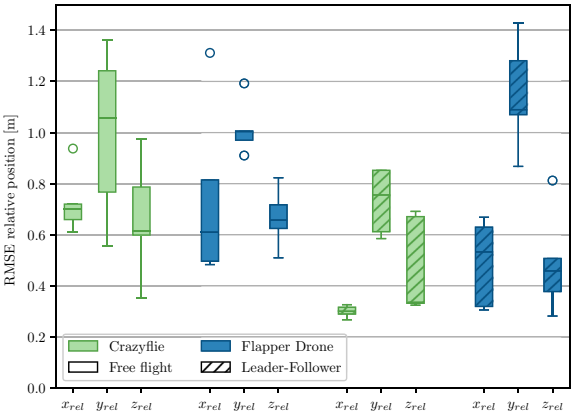


Fig. 16 Comparison of relative localization on Crazyflie and Flapper Drones for free flight and leader–follower experiment aggregated over 5 flights. RMSE calculation starts 30 s into the flight and stops 5 s before landing

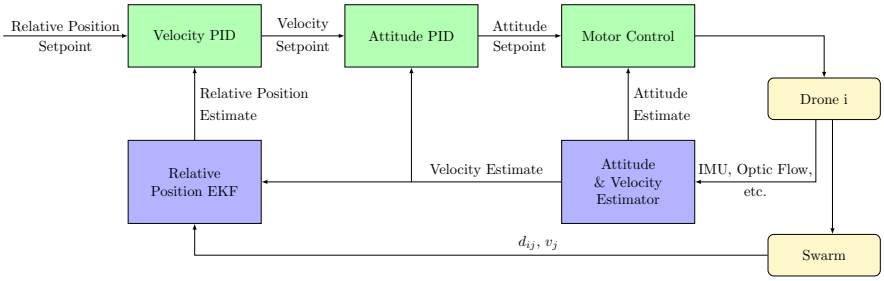


Fig. 17 Leader–follower control layout. Blocks in yellow represent the physical systems

the vertical velocity estimate using the manually tuned complementary filter was 0.20 m/s on the same dataset. During the derivation of the model, a constant altitude was assumed, which is often not the case. We therefore also validated the model on more aggressive trajectories that included major variations in altitude. An example for the performance in such a flight is shown in Fig. 12. While the performance is slightly reduced in x (RMSE=0.18 m/s), y (RMSE=0.31 m/s) and z (RMSE=0.26 m/s), the results still look good enough to justify the use of the model on-board the drones.

We can now also compare the on-board performance of the flapper model with the EKF of the Crazyflie 2.1. Figure 13 shows the aggregated performance data over 10 manual flights for the Crazyflie, Flapper Drones flying at almost constant altitude and Flapper Drones flying with large changes in altitude. Due to range limitations of the Flow Deck, it is not possible to fly large altitude variations with the Crazyflie while still estimating its velocities.

The on-board performance of the Flapper Drone's velocity estimation matches the observation from the offline testing. When flying aggressive trajectories with frequent altitude changes, the performance of the model degrades particularly along the y and z axis. It is, however, striking that, for similar conditions (level flight), the model performs only slightly worse in x and y than the EKF on the Crazyflie, despite having no access to optic flow.

Table 1 PID gains for relative position control

	k_P	k_D	k_I
Crazyflie	2.0	0.01	0.0001
Flapper drone (xy)	3.5	0.06	0
Flapper drone (z)	3.5	0.06	0.001

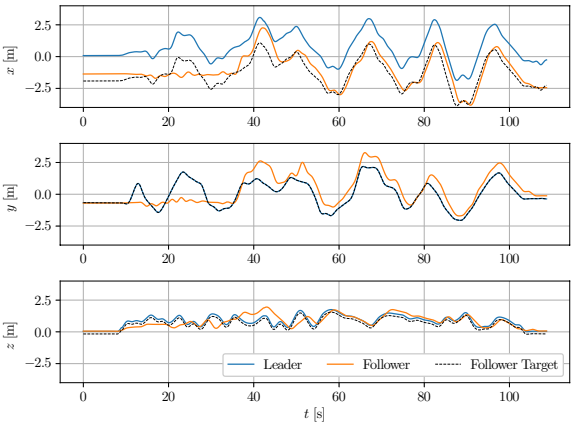


Fig. 18 Optitrack leader and follower position during leader–follower experiment with Crazyflies (target: Δx=-2 m, Δy=0 m, Δz =−0.2 m)

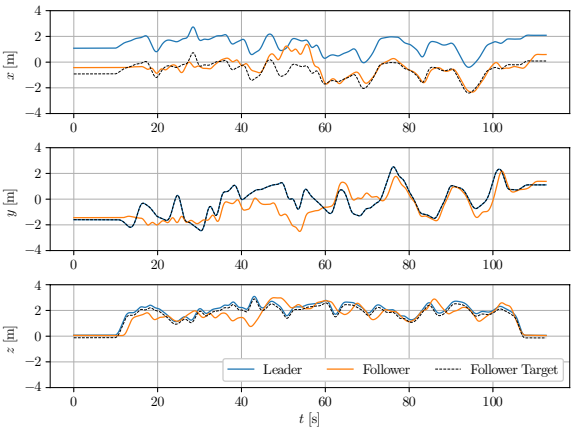


Fig. 19 Optitrack leader and follower position during leader–follower experiment with Flapper Drones (target: Δx=-2 m, Δy =0 m, Δz=−0.2 m)

6.2 Relative localization

To test the 3D relative localization, we performed manual flights with two MAVs without following any particular patterns. Looking at the results for two individual flights using Crazyflies (Fig. 14) and Flapper Drones (Fig. 15), we can observe that the relative position estimate stabilizes within several seconds, which is similar to what we saw in simulation.

Looking at the aggregated data over 5 flights in Fig. 16, it can be seen that the performance along different axes varies in a pattern similar to what could be observed for the velocity estimation (Fig. 13). This makes sense, given that the velocity estimates serve as the main input to determine the relative motion between the MAVs.

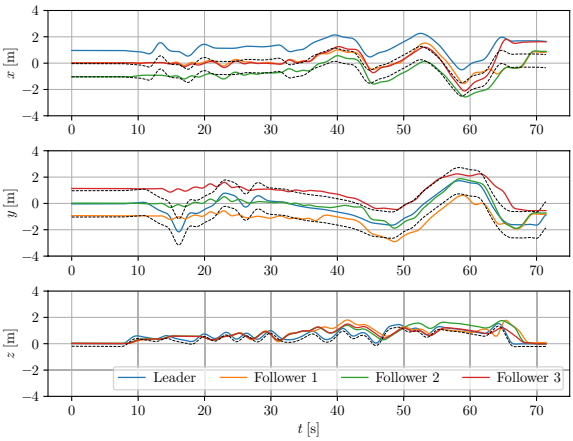


Fig. 20 Absolute position of the leader and the 3 followers. The dashed lines represent the target trajectory for the followers

6.3 Leader–follower

Finally, to show the applicability of the proposed relative localization method to real swarming scenarios, we performed leader-follower flights with two MAVs. In this scenario, the first MAV is controlled manually while the second MAV tries to keep a constant relative position. When the first MAV takes off manually, the second MAV receives a command to start its flight via UWB. The second MAV then takes off as well and performs a 20 s initialization sequence, during which it flies in random patterns based on odometry alone. After the initialization sequence is complete, the second MAV switches to follow mode, where it tries to stay in its desired relative position. This is done by calculating velocity commands directly from the relative position error using a simple PID controller along each individual axis. These velocity commands are then passed to the inner control loop of the drones that control attitude as shown in Fig. 17. Different gains were used for the two types of drones as reported in Table 1. Note that for the Flapper drone only a PD controller was used along the xy axes.

During the experiments, it was found that large deviations from the target position at the end of the initialization sequence could cause unstable behaviour, which is why during the initialization sequence, the leader MAV was flown close to the desired relative position. As can be seen in Fig. 18, the Crazyflie follower can track the leader very well after initialization, with errors rarely exceeding 1.0 in y and 0.5 m in x and z directions. The Flapper Drone on the other hand performs a bit worse in the sense that while the general shape of the trajectory is tracked, the offset from the desired position is larger (Fig. 19). This could be due to the differences in relative localization accuracy, but is also likely to result from sub-optimal controller tuning for this vehicle with more complex dynamics. The large difference between the error in x and y direction, also seen in Fig. 16, can be explained by the desired position offset of the follower, causing movement in x direction to be more observable than in y (movement perpendicular to the UWB ranging).

Surprisingly, Fig. 16 shows that the relative localization mostly performs better (lower RMSE) in the leader–follower scenario than in free flight. Particularly for the experiments on the Crazyflie, improvements of around 0.3 m can be observed along all directions. For the flapper drone, the difference is not as clear and is most pronounced along the z axis. This improvement is likely due to a self-stabilizing effect in the closed loop system. In

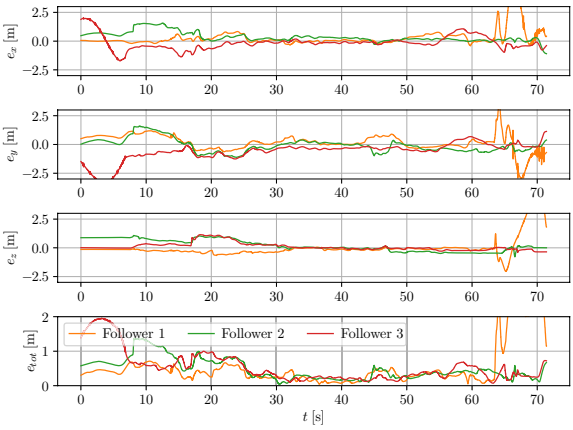


Fig. 21 Error of the relative position estimate on the leader drone

fact, if the relative position along a given axis becomes less observable, its estimate will degrade. This results in a control input that excites that particular axis, rendering it observable again. A more detailed look at this behaviour is given by Li et al. (2021). In free flight on the other hand, the MAVs might fly in patterns in which an axis remains unobservable for longer periods of times.

6.4 Multiple followers

One major concern in the development of algorithms for groups of robots is the scalability. For this relative localization method, the communication frequency is considered the main bottleneck. Specifically, the more drones are part of the swarm, the more messages need to be exchanged. This will result in a reduced frequency of ranging between two given drones. A naive, fully connected communication graph will drop below 5 Hz ranging frequency for more than 12 drones (Li et al., 2021). Furthermore, small drones don't carry a lot of computational power, adding another limitation to the number of drones to which any single drone is localizing. Advances in communication setup and flexible localization in local neighbourhoods will be necessary to scale up the UWB-based relative localization to much larger groups of robots. Here, to reduce the load on both communication and computation, we tested the implementation of a filter bank with a group of 4 drones. This limits the information that has to be exchanged through UWB and keeps the filter matrices small, resulting in a better computational efficiency. For our experiments, we let the followers fly behind the leader in a diamond formation with the following offsets: $(-1.0, -1.0)$ m, $(0.0, -2.0)$ m and $(-1.0, 1.0)$ m. As can be seen in Figs. 20 and 21, the group is still capable of performing the leader-follower behaviour with a similar accuracy as for the two drone case. Note that towards the end of the experiment, Follower 1 is running out of battery.

7 Conclusion

Relative localization is a key aspect of agents in robotic swarms. On a lower level, it helps the robots to avoid collisions with their peers, while on a higher level, it is a prerequisite for most complex swarming behaviours. In this paper, we presented a 3D relative localization algorithm based only on inter-agent ranging and body velocity estimates. Due to

its computational efficiency and the use of lightweight sensors, this algorithm is particularly promising for swarms of MAVs, where individual agents have a very limited payload capacity of only several grams and perform most of their computations on a single core microcontroller.

To understand the limitations of the algorithm, we performed an observability analysis of the system and determined conditions in which observability is no longer guaranteed. We then validated the algorithm in simulation and investigated the effect of choosing trajectories that maximize observability.

Finally, the algorithm was tested on two different types of drones: quadrotors and flapping wing drones. These experiments not only demonstrated the platform independence of the proposed algorithm, but also showed that purely IMU based velocity estimation is sufficiently accurate for relative localization on flapping wing drones. It is, however, important to note that the reliance on velocity estimates is a limitation of our algorithm. In situations, where the body velocities of the drones are difficult to estimate or are very noisy, the performance of the algorithm will start to deteriorate. While an analysis of the consistency of our method shows overconfidence of the estimator, its performance in simulation and practical experiments is in line with other algorithms developed for relative localization. It is, however, an aspect that would profit from additional improvements.

A noteworthy observation is the improved performance of the relative localization in a leader–follower scenario. While one would intuitively expect the algorithm to perform worse in an unobservable configuration (identical velocities), the closed-loop system exhibits a self-stabilizing behaviour that actually improves the localization accuracy. In fact, the unobservability of the relative position leads to small excitations along the unobservable axes. This in turn quickly makes the axes observable again, while for uncoordinated flights, these unobservable configurations might persist for longer.

Since this research mainly focused on relative localization between two agents, it makes sense to place this work in the larger swarming context. While our method can be easily extended to small swarms by running multiple EKFs in parallel on each drone, this would quickly strain memory and computation capacity if too many drones are added. Furthermore, this method does not take advantage of additional measurements between other drones like rigidity based methods do. However, it does allow drones to continue tracking each other's relative position, even in cases where communication links are interrupted and the swarm loses its rigidity. As a result, real-world robotic swarms would benefit from a combination of these two methods to improve the robustness of relative localization in a wide variety of scenarios.

In addition to investigating mixed relative localization methods, the scalability of these methods should further be improved. A critical bottleneck in wireless ranging-based relative localization is the communication channel which can quickly saturate as more agents are added to the swarm. To resolve this issue, a reduction in the number of required communication links within the swarm and in the amount of data exchanged is required.

Appendix A. Derivation of a horizontal velocity model for the flapping wing drone

In this derivation, we consider three main forces acting on the flapping wing drone. Since we are mainly interested in accelerations, we already divide out the mass in the following equations, i.e. $\mathbf{f} = \frac{\mathbf{F}}{m}$. First, the thrust along the drones central axis is given in the drones body frame as

$$\mathbf{f}_T^b = |T| \cdot \mathbf{e}_3 \tag{A1}$$

Second, we consider a simple drag model with drag forces proportional and opposed to the drone's body velocity:

$$\mathbf{f}_d^b = \begin{bmatrix} -b_x v_x^b \\ -b_y v_y^b \\ -b_z v_z^b \end{bmatrix} \tag{A2}$$

Finally, the force of gravity is easily expressed directly in the horizontal frame

$$\mathbf{f}_g^h = -|g| \cdot \mathbf{e}_3 \tag{A3}$$

The sum of acceleration in the horizontal frame experienced by the flapper is then given by A4.

$$\mathbf{a}^h = \sum \mathbf{f}^h = \mathbf{R}_{hb}\mathbf{f}_T^b + \mathbf{R}_{hb}\mathbf{f}_d^b + \mathbf{f}_g^h \tag{A4}$$

We can now put the drag coefficients into a diagonal matrix $\mathbf{B} = \mathrm{diag}(\mathbf{b})$ and expand the equation. Recall that rotation matrices are orthogonal, and therefore, $\mathbf{R}_{bh} = \mathbf{R}_{hb}^{-1} = \mathbf{R}_{hb}^T$. To simplify notation, we will at this point drop the superscript h for vectors in the horizontal frame.

$$\mathbf{a} = |T| \cdot \mathbf{R}_{hb}\mathbf{e}_3 - \mathbf{R}_{hb}\mathbf{B}\mathbf{R}_{hb}^T \cdot \mathbf{v} - |g| \cdot \mathbf{e}_3 \tag{A5}$$

We now write out the equation for each individual axis in the horizontal frame.

$$\begin{aligned} a_x = {} & |T|\mathrm{s}\theta\mathrm{c}\phi - (b_x\mathrm{c}^2\theta + b_y\mathrm{s}^2\theta\mathrm{s}^2\phi + b_z\mathrm{s}^2\theta\mathrm{c}^2\phi)\, v_x \\ & - (b_y - b_z)\mathrm{s}\theta\mathrm{s}\phi\mathrm{c}\phi\, v_y + (b_x - b_y\mathrm{s}^2\phi - b_z\mathrm{c}^2\phi)\mathrm{s}\theta\mathrm{c}\theta\, v_z \end{aligned} \tag{A6}$$

$$\begin{aligned} a_y = {} & -|T|\mathrm{s}\phi - (b_y - b_z)\mathrm{s}\theta\mathrm{s}\phi\mathrm{c}\phi v_x \\ & - (b_y\mathrm{c}^2\phi + b_z\mathrm{s}^2\phi)v_y - (b_y - b_z)\mathrm{c}\theta\mathrm{s}\phi\mathrm{c}\phi v_z \end{aligned} \tag{A7}$$

$$\begin{aligned} a_z = {} & |T|\mathrm{c}\theta\mathrm{c}\phi + (b_x - b_y\mathrm{s}^2\phi - b_z\mathrm{c}^2\phi)\mathrm{s}\theta\mathrm{c}\theta\, v_x - (b_y - b_z)\mathrm{c}\theta\mathrm{s}\phi\mathrm{c}\phi\, v_y \\ & - (b_x\mathrm{s}^2\theta + b_y\mathrm{c}^2\theta\mathrm{s}^2\phi + b_z\mathrm{c}^2\theta\mathrm{c}^2\phi)\, v_z - |g| \end{aligned} \tag{A8}$$

Assuming small roll and pitch angles, we can neglect any terms that include the multiplication of two sines (i.e. $\mathrm{s}^2\theta \approx \mathrm{s}^2\phi \approx \mathrm{s}\theta\mathrm{s}\phi \approx 0$)

$$a_x = |T|\mathrm{s}\theta\mathrm{c}\phi - b_x\mathrm{c}^2\theta\, v_x + (b_x - b_z\mathrm{c}^2\phi)\mathrm{s}\theta\mathrm{c}\theta\, v_z \tag{A9}$$

$$a_y = -|T|\mathrm{s}\phi - b_y\mathrm{c}^2\phi\, v_y - (b_y - b_z)\mathrm{c}\theta\mathrm{s}\phi\mathrm{c}\phi\, v_z \tag{A10}$$

$$a_z = |T|\mathrm{c}\theta\mathrm{c}\phi + (b_x - b_z\mathrm{c}^2\phi)\mathrm{s}\theta\mathrm{c}\theta\, v_x - (b_y - b_z)\mathrm{c}\theta\mathrm{s}\phi\mathrm{c}\phi\, v_y - b_z\mathrm{c}^2\theta\mathrm{c}^2\phi\, v_z - |g| \tag{A11}$$

Due to the difficulty of estimating the generated thrust of the flapper, we further simplify the model by decoupling horizontal and vertical movement of the flapper, i.e. we consider the vertical velocity to be negligible during horizontal motion. This allows us to find an expression for the thrust in terms of $|g|$ and the horizontal velocities.

$$a_z = |T|\mathrm{c}\theta\mathrm{c}\phi + (b_x - b_z\mathrm{c}^2\phi)\mathrm{s}\theta\mathrm{c}\theta\, v_x - (b_y - b_z)\mathrm{c}\theta\mathrm{s}\phi\mathrm{c}\phi\, v_y - |g| = 0 \tag{A12}$$

$$|T| = \frac{|g|}{\mathrm{c}\theta\mathrm{c}\phi} - (b_x - b_z\mathrm{c}^2\phi)\mathrm{s}\theta\, v_x + (b_y - b_z)\mathrm{s}\phi\, v_y \tag{A13}$$

This expression for the thrust can now be inserted into A9 and A10, while setting $v_z = 0$. In both cases, the thrust is multiplied with a sine, which again allows us to neglect some terms due to our small angle approximation. As a result, only the gravity term remains in the final expressions for the acceleration alongside the deceleration caused by drag.

$$a_x = \frac{\mathrm{s}\theta}{\mathrm{c}\theta}|g| - b_x\mathrm{c}^2\theta v_x \tag{A14}$$

$$a_y = -\frac{\mathrm{s}\phi}{\mathrm{c}\theta\mathrm{c}\phi}|g| - b_y\mathrm{c}^2\phi v_y \tag{A15}$$

We can now integrate these accelerations using the forward Euler method to estimate our horizontal velocities.

$$v_{x,k+1} = v_{x,k} + \mathrm{d}t\left(\frac{\mathrm{s}\theta_k}{\mathrm{c}\theta_k}|g| - b_x\mathrm{c}^2\theta_k v_{x,k}\right) \tag{A16}$$

$$v_{y,k+1} = v_{y,k} + \mathrm{d}t\left(-\frac{\mathrm{s}\phi_k}{\mathrm{c}\theta_k\mathrm{c}\phi_k}|g| - b_y\mathrm{c}^2\phi_k v_{y,k}\right) \tag{A17}$$

The drag coefficients b_x and b_y can be estimated from a data set with ground truth using a least squares approach. Specifically, for an overdetermined system of equations $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}$ the least-squares estimate for the parameters $\boldsymbol{\beta}$ is given by $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. To determine the drag coefficients in our horizontal velocity model, we solve two independent least-squares problems with for b_x

$$\boldsymbol{\beta} = b_x \tag{A18}$$

$$y_k = -\frac{v_{x,k+1} - v_{x,k}}{\mathrm{d}t} + \frac{\mathrm{s}\theta_k}{\mathrm{c}\theta_k}|g| \tag{A19}$$

$$X_k = \mathrm{c}^2\theta_k\, v_{x,k} \tag{A20}$$

and for b_y

$$\boldsymbol{\beta} = b_y \tag{A21}$$

$$y_k = -\frac{v_{y,k+1} - v_{y,k}}{\mathrm{d}t} - \frac{\mathrm{s}\phi_k}{\mathrm{c}\theta_k \mathrm{c}\phi_k}|g| \tag{A22}$$

$$X_k = \mathrm{c}^2\phi_k \, v_{y,k} \tag{A23}$$

From our data, we find $b_x = 4.2$ and $b_y = 1.8$ for the Flapper Drones.

References

Arrichiello, F., Antonelli, G., Aguiar, A. P., & Pascoal, A. (2013). An observability metric for underwater vehicle localization using range measurements. *Sensors, 13*(12), 16191–16215. https://doi.org/10.3390/s131216191.

Bailey, T., Nieto, J., Guivant, J., Stevens, M., & Nebot, E. (2006). Consistency of the EKF-slam algorithm. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3562–3568. https://doi.org/10.1109/IROS.2006.281644

Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: A review from the swarm engineering perspective. *Swarm Intelligence, 7*(1), 1–41. https://doi.org/10.1007/s11721-012-0075-2.

Carrio, A., Bavle, H., & Campoy, P. (2018). Attitude estimation using horizon detection in thermal images. *International Journal of Micro Air Vehicles, 10*(4), 352–361. https://doi.org/10.1177/1756829318804761.

Coppola, M., McGuire, K. N., Scheper, K. Y. W., & de Croon, G. C. H. E. (2018). On-board communication-based relative localization for collision avoidance in micro air vehicle teams. *Autonomous Robots, 42*(8), 1787–1805. https://doi.org/10.1007/s10514-018-9760-3.

Coppola, M., McGuire, K. N., De Wagter, C., & de Croon, G. C. H. E. (2020). A survey on swarming with micro air vehicles: Fundamental challenges and constraints. *Frontiers in Robotics and AI, 7,* 18. https://doi.org/10.3389/frobt.2020.00018.

Cossette, C. C., Shalaby, M., Saussié, D., Forbes, J. R., & Le Ny, J. (2021). Relative position estimation between two UWB devices with IMUS. *IEEE Robotics and Automation Letters, 6*(3), 4313–4320. https://doi.org/10.1109/LRA.2021.3067640.

de Croon, G. (2020). Flapping wing drones show off their skills. *Science Robotics, 5*(44), 0233. https://doi.org/10.1126/scirobotics.abd0233.

Eren, T., Goldenberg, O.K., Whiteley, W., Yang, Y.R., Morse, A.S., Anderson, B.D.O., Belhumeur, P.N. (2004). Rigidity, computation, and randomization in network localization. In: IEEE INFOCOM 2004, vol. 4, pp. 2673–26844. https://doi.org/10.1109/INFCOM.2004.1354686

Faigl, J., Krajník, T., Chudoba, J., Přeučil, L., Saska, M.: Low-cost embedded system for relative localization in robotic swarms. In: 2013 IEEE International Conference on Robotics and Automation, pp. 993–998 (2013). https://doi.org/10.1109/ICRA.2013.6630694

Guo, K., Qiu, Z., Meng, W., Xie, L., & Teo, R. (2017). Ultra-wideband based cooperative relative localization algorithm and experiments for multiple unmanned aerial vehicles in gps denied environments. *International Journal of Micro Air Vehicles, 9*(3), 169–186. https://doi.org/10.1177/1756829317695564.

Heintzman, L., & Williams, R. K. (2020). Nonlinear observability of unicycle multi-robot teams subject to non-uniform environmental disturbances. *Autonomous Robots, 44*(7), 1149–1166.

Hermann, R., & Krener, A. (1977). Nonlinear controllability and observability. *IEEE Transactions on Automatic Control, 22*(5), 728–740. https://doi.org/10.1109/TAC.1977.1101601.

Jiang, B., Anderson, B. D. O., & Hmam, H. (2020). 3-d relative localization of mobile systems using distance-only measurements via semidefinite optimization. *IEEE Transactions on Aerospace and Electronic Systems, 56*(3), 1903–1916. https://doi.org/10.1109/TAES.2019.2935926.

Jordan, S., Moore, J., Hovet, S., Box, J., Perry, J., Kirsche, K., et al. (2018). State-of-the-art technologies for UAV inspections. *IET Radar, Sonar and Navigation, 12*(2), 151–164. https://doi.org/10.1049/iet-rsn.2017.0251.

Krener, A.J., & Ide, K. (2009). Measures of unobservability. In: Proceedings of the 48h IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference, pp. 6401–6406. https://doi.org/10.1109/CDC.2009.5400067

Kwon, J., & Hailes, S. (2014). Scheduling uavs to bridge communications in delay-tolerant networks using real-time scheduling analysis techniques. In: 2014 IEEE/SICE International Symposium on System Integration, pp. 363–369. https://doi.org/10.1109/SII.2014.7028065

Ledergerber, A., Hamer, M., D'Andrea, R. (2015) A robot self-localization system using one-way ultra-wideband communication. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3131–3137. https://doi.org/10.1109/IROS.2015.7353810

Li, S., Coppola, M., Wagter, C.D., & de Croon, G.C.H.E. (2021). An autonomous swarm of micro flying robots with range-based relative localization. https://arxiv.org/abs/2003.05853

Maes, W. H., & Steppe, K. (2019). Perspectives for Remote Sensing with Unmanned Aerial Vehicles in Precision Agriculture. *Trends in Plant Science, 24*(2), 152–164. https://doi.org/10.1016/j.tplants.2018.11.007.

Mahony, R., Hamel, T., & Pflimlin, J.-M. (2008). Nonlinear Complementary Filters on the Special Orthogonal Group. *IEEE Transactions on Automatic Control, 53*(5), 1203–1218. https://doi.org/10.1109/TAC.2008.923738.

Nguyen, T.H., Xie, L. (2022). Relative transformation estimation based on fusion of odometry and UWB ranging data.

Preiss, J.A., Hönig, W., Sukhatme, G.S., Ayanian, N. (2017). Crazyswarm : A large nano-quadcopter swarm. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 3299–3304. https://doi.org/10.1109/ICRA.2017.7989376

Roberts, J. F., Stirling, T., Zufferey, J.-C., & Floreano, D. (2012). 3-D relative positioning sensor for indoor flying robots. *Autonomous Robots, 33*(1), 5–20. https://doi.org/10.1007/s10514-012-9277-0.

Şahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. In E. Şahin & W. M. Spears (Eds.), *Swarm Robotics* (pp. 10–20). Berlin, Heidelberg: Springer.

Shalaby, M., Cossette, C. C., Forbes, J. R., & Le Ny, J. (2021). Relative position estimation in multi-agent systems using attitude-coupled range measurements. *IEEE Robotics and Automation Letters, 6*(3), 4955–4961. https://doi.org/10.1109/LRA.2021.3067253.

Shang, Y., & Ruml, W. (2004). Improved mds-based localization. In: IEEE INFOCOM 2004, vol. 4, pp. 2640–26514. https://doi.org/10.1109/INFCOM.2004.1354683

Simon, D. (2006). *Optimal State Estimation*. Hoboken NJ: John Wiley & Sons.

Tijs, E., de Croon, G., Wind, J., Remes, B., De Wagter, C., de Bree, H., & Ruijsink, R. (2010). Hear-and-avoid for micro air vehicles. In: Proceedings of the International Micro Air Vehicle Conference and Competitions (IMAV), Braunschweig, Germany, vol. 69

Trawny, N., Zhou, X.S., Zhou, K.X., & Roumeliotis, S.I. (2007). 3d relative pose estimation from distance-only measurements. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1071–1078. https://doi.org/10.1109/IROS.2007.4399075

Trawny, N., Zhou, X. S., Zhou, K., & Roumeliotis, S. I. (2010). Interrobot transformations in 3-d. *IEEE Transactions on Robotics, 26*(2), 226–243. https://doi.org/10.1109/TRO.2010.2042539.

Vásárhelyi, G., Virágh, C., Somorjai, G., Nepusz, T., Eiben, A. E., & Vicsek, T. (2018). Optimized flocking of autonomous drones in confined environments. *Science Robotics, 3*(20), 1–14. https://doi.org/10.1126/scirobotics.aat3536.

Vedder, B., Eriksson, H., Skarin, D., Vinter, J., Jonsson, M.: Towards collision avoidance for commodity hardware quadcopters with ultrasound localization. In: 2015 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 193–203. https://doi.org/10.1109/ICUAS.2015.7152291

Wei, S., Dan, G., & Chen, H. (2016). Altitude data fusion Utilising differential measurement and complementary filter. *IET Science, Measurement and Technology, 10*(8), 874–879. https://doi.org/10.1049/iet-smt.2016.0118.

Williams, R.K., & Sukhatme, G.S. (2015). Observability in topology-constrained multi-robot target tracking. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 1795–1801. IEEE

Zhu, J., & Kia, S. S. (2019). Cooperative localization under limited connectivity. *IEEE Transactions on Robotics, 35*(6), 1523–1530.

Ziegler, T., Karrer, M., Schmuck, P., & Chli, M. (2021). Distributed formation estimation via pairwise distance measurements. *IEEE Robotics and Automation Letters, 6*(2), 3017–3024. https://doi.org/10.1109/LRA.2021.3062347.