Cooperative PNT in Swarms

Dr. Maarten Uijt de Haag and Mats Martens, M.Sc.
Institute of Aeronautics and Astronautics
Flight Guidance and Air Transport
Marchstrasse 12, 10587 Berlin
GERMANY

maarten.uijtdehaag@tu-berlin.de

ABSTRACT

This paper focusses on one challenge of operating small Unmanned Aerial Vehicle (sUAV) swarms in urban environments, namely advanced absolute and relative navigation methods for swarms and swarm members. The paper discusses various absolute and relative pose and velocity estimation approaches that extend conventional integrated navigation methods to include principles of cognition (i.e., appropriate, and adapted actions based on perception and knowledge) and collaboration (i.e., improved ability to reason and interact based on information exchange and spatial distribution) between the swarm members. This paper outlines the underlying methodology, simulation results, and the platform hardware components. It furthermore analyses and discusses the navigation performance using sUAV flight test data collected in an open-sky and simulations in an urban environment.

1.0 MOTIVATION AND BACKGROUND

Over the last decade, the number of applications that use small Unmanned Aerial Vehicles or sUAVs have been increasing and the applications have become more and more complex in terms of operations and environments. Using multiple cooperative sUAVs (i.e., a swarm or group) might be beneficial or even necessary to perform tasks, such as infrastructure inspection, mapping, law enforcement, traffic monitoring either independently or collaboratively [1]. Using multiple dissimilarly equipped sUAVs to perform a task may not only significantly reduce the time required to complete a task, but also reduce the ground risk, i.e. risk of threats to population and property, due to the reduced complexity and weight, increased reliability, and longer endurance of the smaller platforms. Furthermore, swarms of sUAVs may be able to increase safety by exploiting the increase and reliability of knowledge through distributed cognition and swarm-wide collaboration.

Since many swarm applications take place at very low altitude levels (VLL) and over urban, often populated, areas, it is important to address and limit the air and ground risks associated with their operation [2]. To lower the probability of an accident, and, thus, make the operation safer and more acceptable to the public, the operation of the swarm members must be carefully planned, limited to pre-defined areas that are free of obstacles such as routes or geofenced zones, or achieved by equipping the swarm members with a conflict detection and resolution function that ensures that possible conflicts of a swarm member with other members or its environment are detected in time and alternative routes can be planned and executed. These strategies require reliable estimates of position, velocity, and attitude (PVA) of the swarm and its members both in the *absolute* sense (i.e., with respect to the geographic coordinate frame in which the routes and geofences are defined) and in the *relative* sense (i.e., with respect to other swarm members, other traffic, and objects in the environment).

For many commercial (s)UAVs, Global Navigation Satellite Systems (GNSS) have become the de facto source for absolute on-board position and velocity information enabling very accurate position control and navigation along predefined trajectories for an sUAV operating in autonomous or semi-autonomous modes. Examples of the use of GNSS as the positioning source for swarm operations include various research flight tests such as the evaluation of methods to mimic the behavior of a flock of birds with UAS [3], and the development of centralized [4] or ad-hoc communication networks [5][6] to relay information between the various members of the swarm. Even the large lightshows that use hundreds of UAVs by companies such as Intel [7] and Ehang [8] are heavily choreographed and mainly use GNSS or real-time kinematic or precise point positioning GNSS. These systems are typically operated in open areas where GNSS reception is optimal and, can, support the navigation performance (i.e., accuracy, integrity, availability, and continuity) required for the safe and conflict-free operation. Relative GNSS-based navigation methods, such as the one presented in previous work by the authors [9] work well for this task, given that all members have a GNSS capability. Because most errors are the same for receivers in proximity, the separation vectors can be easily calculated and remain accurate at the decimeter level. However, in the event of an GNSS outage or a compromised or malfunctioning receiver, the ability to calculate accurate separation vectors significantly degrades.

The reliance on GNSS forms a big challenge for operation of cooperative swarms in GNSS-denied or semi-denied environments such as, urban canyons for bridge or construction inspection, under-the-canopy operation during environmental monitoring or inventory tasks, city traffic surveillance, etc. If a swarm seeks to function in such environments, members must be able to perform relative (and absolute) navigation when some or all members are GNSS-denied. To enable operation of the swarm the PVA must be robust and not solely dependent on GNSS.

To improve availability and guarantee continuity of service in GNSS-challenged environments, GNSS can be integrated with an Inertial Measurement Unit (IMU) [10] or improved by increasing its sensitivity by using external data sources (i.e., assisted GPS). An alternative strategy is the integration or fusion of multiple sources of data which may not only improve the accuracy of the position and attitude estimate, but also add integrity, continuity, and availability to the solution. Alternative navigation technologies may include (a) the integration of inertial sensors with imagery and laser scanners [11], (b) beacon-based navigation (i.e., pseudolites, UWB), (c) or navigation using signals of opportunity [12].

2.0 COOPERATIVE AND COGNITIVE SWARM NAVIGATION

The basic swarm geometry is shown in Figure 2-1. One navigation-related objective for each swarm member (or agent) is the estimation of the absolute pose and velocity of ownship and the other agents in the swarm, and the relative poses between the swarm's agents. These estimates are essential to maintain a safe flight (e.g., free of collisions with other members and its environment) and to meet the swarm's mission goals.

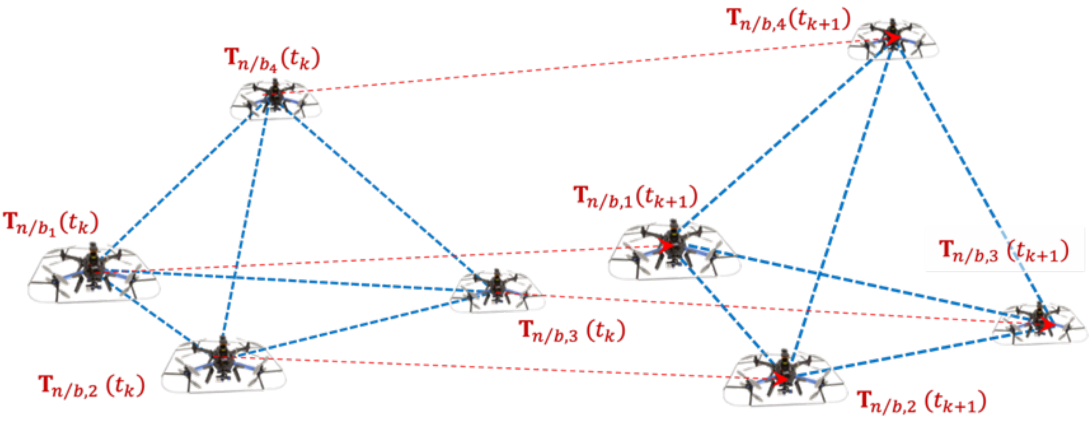

$\mathbf{T}_{n/b,4}(t_{k+1})$
$\mathbf{T}_{n/b_4}(t_k)$
$\mathbf{T}_{n/b,1}(t_{k+1})$
$\mathbf{T}_{n/b_1}(t_k)$
$\mathbf{T}_{n/b,3}\ (t_{k+1})$
$\mathbf{T}_{n/b,3}\ (t_k)$
$\mathbf{T}_{n/b,2}\ (t_k)$
$\mathbf{T}_{n/b,2}\ (t_{k+1})$

Figure 2-1: Swarm geometry of four sUAS as a function of time.

The absolute pose is defined by the orientation of agent 'i' with respect to the navigation frame, $\mathbf{R}_{n/b_i}$ and its position in the navigation frame, $\mathbf{r}_{n,i}$, given by the matrix.

$$\mathbf{T}_{n/b_i} = \begin{bmatrix} \mathbf{R}_{n/b_i} & \mathbf{r}_{n,i} \\ 0 & 1 \end{bmatrix} \qquad (1)$$

The relative pose is defined by the orientation of agent 'j' with respect to the body frame of agent 'i', $\mathbf{R}_{b_i/b_j}$ and the separation vector expressed in navigation frame, $\mathbf{s}_{n,ij}$:

$$\mathbf{T}_{b_i/b_j} = \begin{bmatrix} \mathbf{R}_{b_i/b_j} & \mathbf{r}_{b_i,j} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \qquad (2)$$

where $\mathbf{r}_{b_i,j} = \mathbf{R}_{b_i/n}\mathbf{s}_{n,ij} = \mathbf{R}_{b_i/n}\left(\mathbf{r}_{n,j} - \mathbf{r}_{n,i}\right)$.

The relative pose can be used to relate the poses of agents 'i' and 'j' in the navigation frame, or:

$$\mathbf{T}_{n/b_j} = \mathbf{T}_{n/b_i}\,\mathbf{T}_{b_i/b_j} \Longrightarrow \mathbf{T}_{b_i/b_j} = \mathbf{T}_{n/b_i}^{-1}\mathbf{T}_{n/b_j} \qquad (3)$$

It is important to note, that, in terms of exchange of information between the swarm members, the swarm should typically not be considered a fully connected network so that the performance (accuracy, integrity, availability, and continuity) of the absolute and relative pose estimates $\widehat{\mathbf{T}}_{n/b_i}$ and $\widehat{\mathbf{T}}_{b_i/b_j}$ will be a function of the information locally available and available through the agents (i.e., swarm members) with which it is connected (through the connection matrix). So, first, each agent must gather all information from its connected neighbors, determine for what navigation-related parameters sufficient information is available (i.e., what absolute and relative state variables are observable), perform the estimation of the observable states, and then assess the actual performance of these estimates. Since, this process may differ from agent to agent, estimates for the same quantity may differ from one sUAV to another, leading to an inconsistent view of the "world". The assessment of this global view is not the focus of this paper, but has been addressed by papers addressing the scheduling of tasks within a swarm, such as the work described in [13] and the work in [14] which also addresses the communications aspects of the problem.

In nature, swarms of animals have proven to be capable of solving a variety of different and complex navigation tasks using their cognitive and collaborative abilities (e.g., honeybees, ants, birds [15][16]). Following the success of these swarms, we identified cognition and collaboration as the central elements of a swarm navigation architecture and use existing knowledge from neuroscience, biology, and robotics to design methods to achieve the swarm's mission while meeting the stringent navigation requirements for urban operation. Here, cognition is defined as *the selection of an appropriate and adapted action based on perception and knowledge*, and collaboration as *the improved ability to reason and interact based on information exchange and spatial distribution* of a swarm. As an example of how cognition and collaboration is helping navigation in nature, [17] describes five mechanisms a swarm of animals may use to improve their navigation abilities and accuracy during migratory behavior:

A. **Many wrongs***:* in this case improvement of the navigation performance is achieved by using across platform averaging or filtering in the navigation estimation process. This principle is illustrated in Figure 2-2(a). The figure shows five sUAVs along with their individual position uncertainty ellipses. When combining the noisy position and velocity estimates to obtain a global estimate of the swarm's motion using some filter approach, a noise reduction can be expected due to the averaging effect.

B. **Leadership***:* in many cases some swarm members have better knowledge of their navigation solution (in terms of performance or observability) and use that knowledge to help the remaining members meet their navigation performance requirements. An example of this mechanism is illustrated in Figure 2-2(b). Here, two high flying sUAVs are equipped with high-accuracy and resilient GNSS/Inertial systems where the other three swarm members (3,4, and 5) are not and are operating in an environment where GNSS performance is significantly deteriorated resulting in an inability of sUAS 3,4, and 5 to determine its absolute position accurately or at all. Now, sUAVs 3,4, and 5 may be able to use the advanced knowledge of the leaders to improve their absolute and global position estimate. An example of this mechanism is discussed in [18].

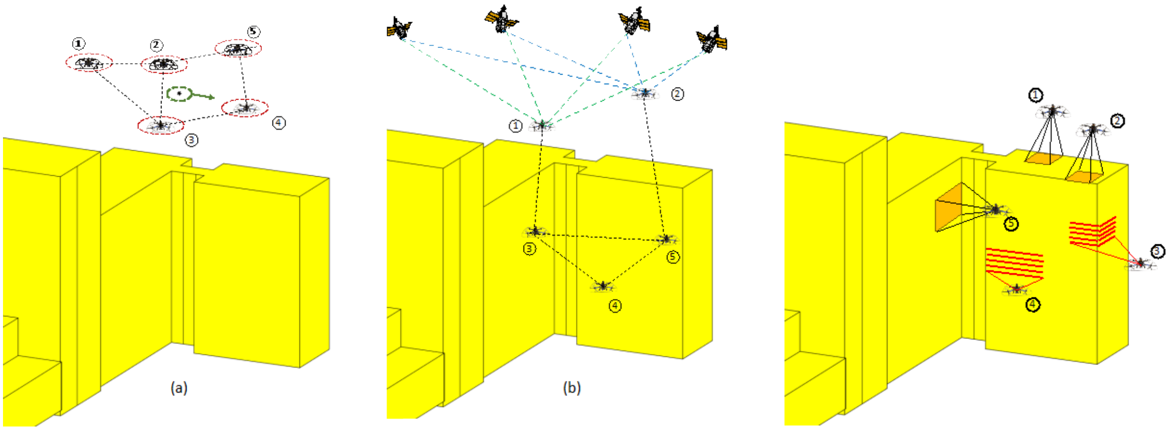


Figure 2-2: (a) many wrongs principle; (b) leadership, (c) Emergent sensing.

C. Emergent sensing*:* in this mechanism the whole swarm comes up with a set of measurements that can be used to build a model of the environment (e.g., situational awareness) and help in the current or future navigation efforts. An example scenario that highlights this mechanism is shown in Figure 2-2(c). In this example, swarm members with different sensor equipment perform an infrastructure inspection mission for the first time and use the different sensor modalities to generate a map of the environment, that then can be used for navigation purposes in a manner like simultaneous localization and mapping (SLAM). Over time, this model will not only help the swarm perform its infrastructure inspection mission more efficiently, but also come up with a mission path that guarantees that the required navigation performance criteria are met, and the mission is performed safely.

D. **Social learning***:* in this case information existing with the individual swarm members is exchanged so the whole group can benefit. This mechanism allows new swarm sUAVs to learn from the members that have been used for missions for a while. Furthermore, equally experienced sUAVs can exchange information from past missions when their experiences have been different due to, for example, different routes/trajectories that they have flown. Information could include (partial) maps of the environment or expected navigation performance at certain locations as e.g., under bridges or urban canyons.

E. **Collective learning***:* in this mechanism interactions within the group lead to better and more detailed knowledge of the environment and, therefore, so better collision avoidance decisions can be made, and routes can be identified that support the required navigation performance. An example would be to fly a configuration that optimizes map-building while at the same time using that map to maintain a good estimate of the relative location of the swarm members (in addition to GNSS).

As mentioned in the introduction, a swarm of UAVs may significantly reduce the time required to complete the application task. If one distributes the sensors necessary to perform a certain task across the UAS within the swarm, it will be possible to use smaller and lighter UAS that operate longer and cost less, thus, offsetting the costs associated with the increase in platforms required. More importantly, it is our thesis that, due to the possibility to spatially distribute the swarm members, the swarm sensors combined can observe a larger part of the environment when working together, while at the same time use different perspectives to estimate for e.g., building features. In nature, this mechanism is referred to as *emergent sensing* and *collective learning* and may lead to increased navigation accuracy in animal swarms. The result will be a better model of the environment that can be used to assess the collision risk by detecting possible loss-of-separation events of the swarm members for the near future. If so, appropriate action can be taken to avoid this collision. Furthermore, the navigation capability may also deteriorate for one of the swarm members if its sensors does not provide sufficient information to estimate both the absolute and relative PVA state. Collaboration through the exchange of information, may alleviate this problem.

3.0 METHODOLOGY

3.1 Perception, Comprehension and Projection and Decision Making

The high-level block diagram of the proposed method for cognitive and collaborative navigation of swarms is shown in Figure 3-1. This block diagrams resembles the model of situational awareness in dynamic decision systems as introduced by Endsley [19]. Even though Endsley's work focusses on Human Factors, the concept of situational awareness for a human can be easily translated to a situation awareness model for the swarm members and the swarm as a whole. The decision-making loop shown in Figure 3-1 also shows a lot of similarities with the Observe, Orient, Decide and Act (OODA) loop by Boyd and the perception-action cycle [20].

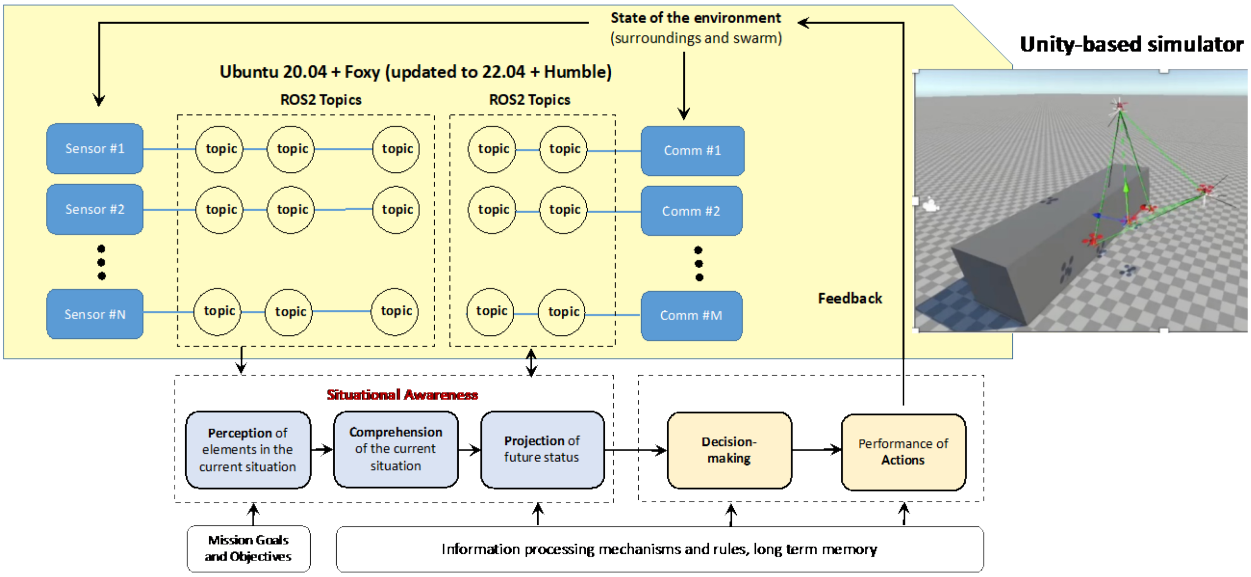

State of the environment
(surroundings and swarm)
Unity-based simulator
Ubuntu 20.04 + Foxy (updated to 22.04 + Humble)
ROS2 Topics
ROS2 Topics
Sensor #1
Sensor #2
Sensor #N
topic
Comm #1
Comm #2
Comm #M
Feedback
Situational Awareness
Perception of elements in the current situation
Comprehension of the current situation
Projection of future status
Decision-making
Performance of Actions
Mission Goals and Objectives
Information processing mechanisms and rules, long term memory

Figure 3-1: Swarm Architecture – Dynamic Decision System.

In the decision-making loop, the swarm members improve their awareness of their surroundings both to achieve their mission goals and objectives, but also to do so safely by meeting the required (stringent) absolute and relative navigation performance requirements imposed by the operational use and mission of the swarm. Like in human factors, three levels of situational awareness can be defined. In the *perception* level (level 1), the swarm member assesses what relevant information is available. This information includes local navigation sensor information, information in onboard long-term memory (e.g., models of the environments, maps, etc.), and sensory information from other agents in the swarm available through the communication network. Note that the latter information may not be complete, i.e., include sensory information from all other swarm agents, due to the lack of full network connectivity.

Within our framework, the Robotic Operating System 2 (ROS2) [21] is used to interface with the local sensors and communication links (the interface program is referred to as a ROS2 node). The messages output by these nodes are clearly defined and contain the actual raw or processed data of the sensor (whatever is applicable for the sensor in question) as well as meta data describing some of the sensor characteristics, settings, and limitations. These topics are then used by the perception module.

In the current framework implementation, all components with the yellow box in Figure 3-1, can also be simulated. Although various options are available and have been used such as Gazebo or AirSim, an in-house Unity-based simulator has been developed and is continued to be improved to support this simulation feature. The assessment results output by the *perception* module, are then used by the *comprehension* module to get a clear understanding of the current situation, and, finally, to predict what the future situation will look like in the *projection* module. In terms of navigation performance this could refer to the current and future absolute and relative navigation state and associated performance, respectively (see Figure 3-2). Based on this knowledge and the mission goals and objectives, the swarm members will make decisions regarding their planned actions (e.g., motion changes, new trajectories, new swarm configurations to optimize the navigation performance or network configuration). Example behaviors that would require absolute and relative navigation capability is the earlier discussed flocking behavior that implements, for example, flocking based on Reynold's principles of separation (try to stay well clear of the other swarm members), alignment (steer towards the average track of local swarm members), and cohesion (steer towards the average position of the local swarm members, try to stay together) [22].

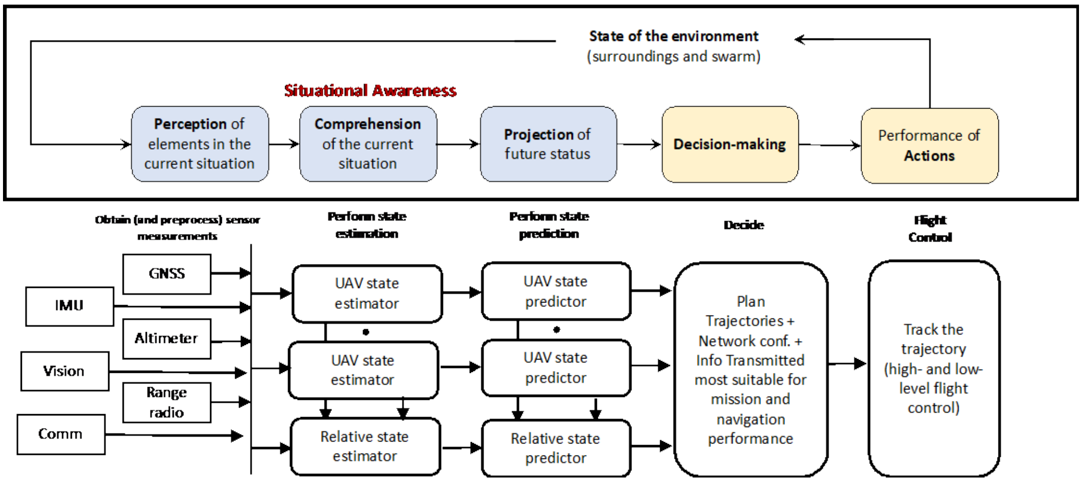


Figure 3-2: Swarm Architecture – Dynamic Decision System – Navigation-specific implementation.

Table 3-1 shows some of the navigation-related sensors that may be available on the swarm members. Note that the swarm may not necessarily have to consist of identical sUAS resulting in a lower cost and weight of the platforms.

Table 3-1: Sensor information: measurement examples.

	Raw	Processed
GNSS	Pseudorange, carrier-phase	Position, position change
Laser range scanners	Range, scan angle, point cloud	Position change, orientation change
3D Imagers	Range, azimuth, elevation (θ_i), point cloud	Position change, orientation change
Camera (mono)	Unit vector pointing to feature/pixel + intensity	Scaled position change, orientation change
Camera (stereo)	Pairs of unit vector pointing to feature/pixel	Position change ($\Delta\mathbf{r}_i$), orientation change
Beacons	Range	Position
Range radios	Relative range between 'i' and 'j'	-

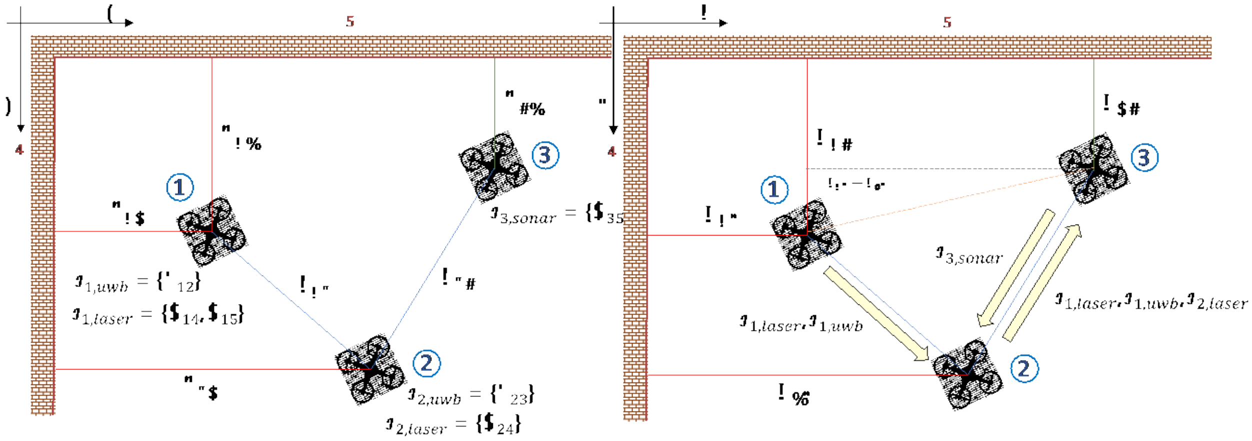
IMU	Acceleration/specific force, angular rate	Position, velocity, attitude
Optical flow	-	Scaled velocity
Radio/laser altimeter	Height above ground	-
Baro altimeter	Height w.r.t. pressure reference height	-
Magnetometer	Orientation w.r.t. magnetic field	-

An example of this approach is illustrated in the simple two-dimensional (2D) 3-member swarm example given in Figure 3-3 and Figure 3-4. Based on the information available in each of the UAVs in Figure 3-3 (left), only the 2D position of UAV 1 can be calculated as it is completely constraint by the two distance measurements to walls ‘4’ and ‘5’ obtained from a laser range scanner. However, both UAV 2 and 3 do not have sufficient measurements to estimate their 2D position.

$$\mathbf{r}_1 = \begin{bmatrix} d_{14} \\ d_{15} \end{bmatrix}, \mathbf{r}_2 = \begin{bmatrix} d_{24} \\ ? \end{bmatrix}, \mathbf{r}_3 = \begin{bmatrix} ? \\ d_{35} \end{bmatrix} \tag{4}$$

However, by exchanging information, $\mathcal{I}$, in the form of measurements between the swarm members, enough information will be available in each UAV to locally compute not only the UAV’s location but also its teammates in a consistent manner. In this example, ρ_{ij} is the range between UAS ‘i’ and ‘j’ from a range radio sensor, and d_{ij} is the shortest distance between UAS ‘i’ and surface ‘j’ from a laser range scanner.

$$\mathbf{r}_1 = \begin{bmatrix} d_{14} \\ d_{15} \end{bmatrix}, \mathbf{r}_2 = \begin{bmatrix} d_{24} \\ d_{15} + \sqrt{\rho_{12}^2 - (d_{24} - d_{14})^2} \end{bmatrix}, \mathbf{r}_3 = \begin{bmatrix} d_{24} + \sqrt{\rho_{23}^2 - \left(r_{2,x} - d_{35}\right)^2} \\ d_{35} \end{bmatrix} \tag{5}$$



5
4
5
4
1
2
3
1
2
3
$\mathcal{I}_{1,uwb} = \{' _{12}\}$
$\mathcal{I}_{1,laser} = \{\$_{14}, \$_{15}\}$
$\mathcal{I}_{2,uwb} = \{' _{23}\}$
$\mathcal{I}_{2,laser} = \{\$_{24}\}$
$\mathcal{I}_{3,sonar} = \{\$_{35}$
$\mathcal{I}_{3,sonar}$
$\mathcal{I}_{1,laser}, \mathcal{I}_{1,uwb}$
$\mathcal{I}_{1,laser}, \mathcal{I}_{1,uwb}, \mathcal{I}_{2,laser}$

Figure 3-3: 2D swarm navigation without information exchange (left), and with information exchange (right)

If, in addition to the information exchange, UAV actions are also employed, then, a movement of UAV 3 in Figure 3-4 (right) within radio range of the ultra-wideband (UWB) range radio of UAV 1 results in two ways to estimate the UAV positions, allowing the user to perform a consistency check and possibly detect sensor faults that may exist and lead to hazardous situations. A consensus-based method that uses concepts from information theory such as the approach proposed in [23] could be used to obtain the coordination actions needed to automate the above concept with our dynamic decision system.

Another consideration in the decision on how to move the swarm members (i.e., how to configure them), is the effect of the geometry on the navigation performance (like the Dilution of Precision or DOP in GNSS). This may mean that to achieve a required navigation performance imposed by the operational scenario, the swarm configuration must be chosen such that the DOP and nominal measurement accuracy support the accuracy requirements. In [24] an approach is described to perform this task on a swarm whose members have knowledge of the ranges between another.

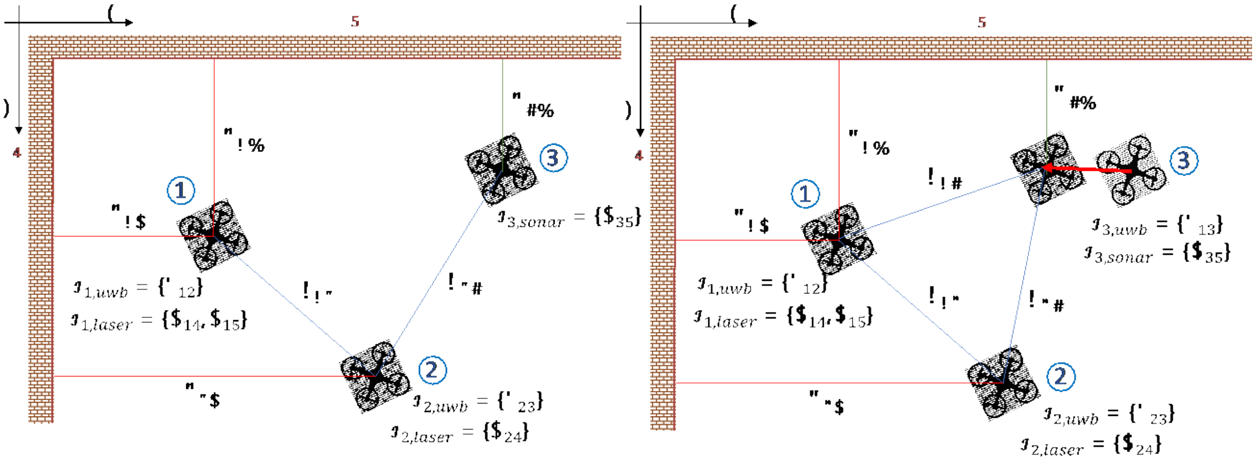


Figure 3-4: 2D swarm navigation without information exchange (left), and with information exchange and action reasoning (right).

3.2 Swarm Constraints and Rules

When selecting the (minimum) set of sensor measurements required to estimate the position of each swarm member, it is important to identify the constraints introduced by the available measurements. The measurements must be selected such that the defined state vector determined by PVA, can be observed [25] and that a minimum navigation performance level (i.e., accuracy, integrity, availability, continuity) can be achieved. In the following examples, it is assumed that the whole swarm configuration is computed by each swarm member's processing unit. Note that the choice between central and decentralized swarm architecture is not the focus of this paper.

Figure 3-5 (a), (b) and (c) show 2D examples of a small swarm with 3 sUAVs where the ranges between all members are known due to the availability of range radio measurements. Without any further constraints, the swarm has 3 degrees of freedom. It could be freely translated and rotated as a rigid body within the 2D navigation frame without being inconsistent with any measurement.

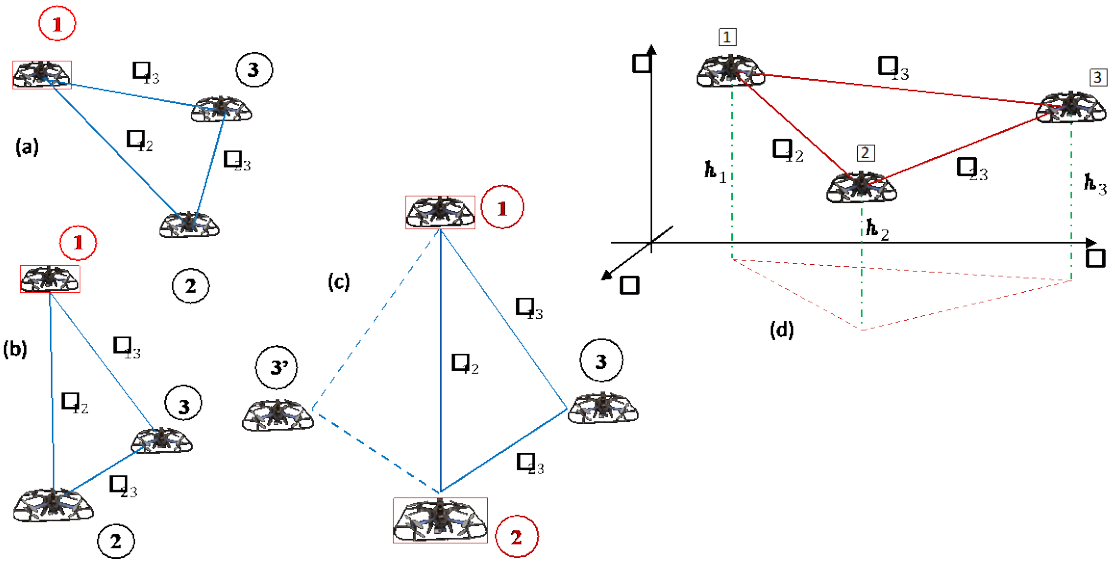


Figure 3-5: Constraints –(a) and (b) 2D position constraints with known ranges and a single global constraint, (c) 2D position constraints with known ranges and two global constraints, (d) 3D swarm configuration with range and altitude constraints.

When two of the members (1 and 2) have knowledge of their global position estimate, but member 3 does not, the ambiguity is reduced to 1-degree of freedom as shown in Figure 3-5(c). With known ranges ρ_{12}, ρ_{13}, and ρ_{23}, member 3's position is ambiguous: it could either be in at position 3 or position 3' in 2D. In three dimensions (3D), the location of sUAV 3 could be on a circle. In 3D the ambiguity could again be reduced to two degrees (horizontal translation, rotation around a vertical axis) when the altitude is known from baro-altimeters, for example, as illustrated in Figure 3-5 (d).

4.0 MECHANIZATIONS – BAYESIAN ESTIMATORS

4.1 Overview

To integrate the information from a swarm member's onboard sensors and data received from other swarm members, various integration approaches may be selected including snapshot methods such as ordinary least squares (OLS) and weighted least squares (WLS) estimators, parametric sequential estimators such as Kalman Filters (KF), Extended Kalman filters (EKF), or non-parametric estimators such Particle Filters (PF). For details on all these estimators many good reference texts are available including [25] and [26].

The filter mechanizations must be selected based on available sensor information and care must be taken that potential transitions from one mechanization to another do not affect the navigation performance adversely (e.g., jumps or divergence). For each of the sensors available within the swarm, a basic measurement model must be derived that relates the sensor measurements to the state vector, $\mathbf{x}$ (i.e., navigation- or error state) through function $\mathbf{h}$.

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \tag{6}$$

Component $\mathbf{v}_k$ represents the error introduced by the sensor. In many cases, this error is assumed to be a zero-mean normally distributed noise with a covariance matrix equal to $\mathbf{\Sigma}_v$. However, this assumption must always be tested before designing the filter mechanization. In the following sections we will shortly describe example swarm mechanizations that are used in the results section.

4.2 Example: Inertial/Range-radio/Baro Integration

The example discussed in this section shows an example of the leadership and social learning principles mentioned in Section 2.0. It is assumed here that all measurement information is shared among the sUAVs. Of course, in a real scenario this could be too much of a burden on the data links of the swarm members. To estimate the position of all sUAVs, a centralized complementary extended Kalman filter (CEKF) has been used. Note that this could be implemented decentralized as well with partial information for each swarm member, however, in that case the consistency of the situation awareness by all members must be considered and assessed.

The state vector consists of the errors states of all swarm members involved in the position calculation (here: 6). Error states include 3D position and velocity errors, the tilt error, and the accelerometer and gyro bias vectors. Expressions for these as well as their corresponding state transition matrices can be found in reference texts such as [10] and [33]. For a swarm with 'N' Inertial Navigation Systems (INS), the state vector would be size $15N$.

The range between any two sUAVs can be computed using their respective INS position estimates $\tilde{\mathbf{r}}_{k,ins}$, and expressed in terms of the true relative position and an additional error term that combines the INS errors from 'i' and 'k', or:

$$\rho_{ik_{ins}} = \left\|\tilde{\mathbf{r}}_{i,ins} - \tilde{\mathbf{r}}_{k,ins}\right\| = \left\|\mathbf{s}_{ik} - \delta\mathbf{r}_{ik,ins}\right\| \tag{7}$$

Equation (7) can be linearized with respect to the involved sUAS INS position estimates.

$$\tilde{\rho}_{ik_{ins}} \approx \rho_{ik_{true}} + \frac{\mathbf{s}_{ik}^T}{\rho_{ik}}\delta\mathbf{r}_{ik,ins} = \rho_{ik_{true}} + \underbrace{\mathbf{u}_{ik}^T\,\delta\mathbf{r}_{ik,ins}}_{\delta\rho_{ik_{ins}}} \tag{8}$$

The row elements of the measurement vector, **z**, are given by the various available range differences in the swarm:

$$z_{ik} = \tilde{\rho}_{ik_{ins}} - \tilde{\rho}_{ik_{rr}} = \delta\rho_{ik_{ins}} + v_{rr} \tag{9}$$

where $\delta\rho_{ik_{ins}}$is the error due to the contributions of two INSs involved, and $v_{rr}$is the range noise error of the range-radio. Equation (9) can be expressed in the inertial position error terms as follows:

$$z_{ik} = \delta\rho_{ik_{ins}} + v_{rr} \approx -\mathbf{u}_{ik}^T\delta\mathbf{r}_{ik,ins} + v_{rr} \tag{10}$$

The baro-altimeter measurement is much simpler, as it is already linear:

$$z_{alt,k} = h_{k,ins} - h_{k,alt} = \delta r_{k_{z,ins}} + v_{baro} \tag{11}$$

where $\delta r_{k_{z,ins}}$ is the z-component of the position vector computed by the INS of swarm member k'. The results of equations (10) and (11) can be used to setup the **H**-matrix. A block diagram that summarizes this method is shown in Figure 4-1.

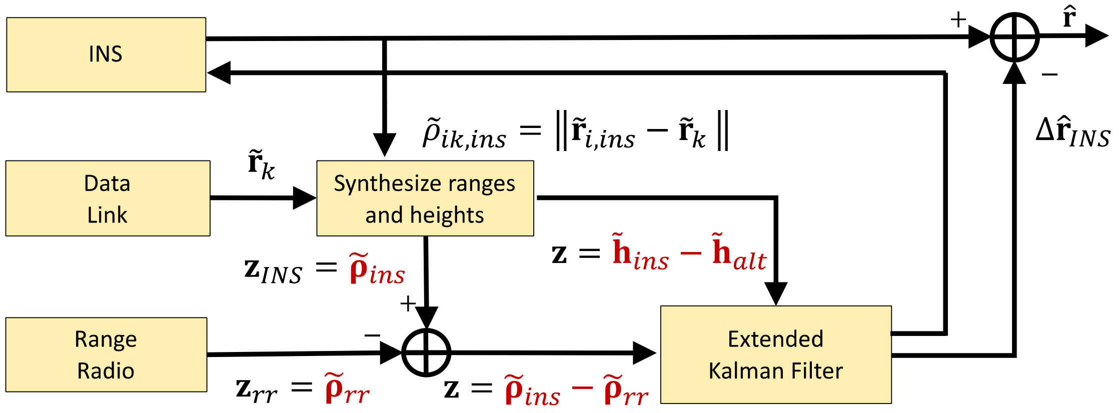

INS
Data Link
Range Radio
Synthesize ranges and heights
Extended Kalman Filter
$\tilde{\mathbf{r}}_k$
$\tilde{\rho}_{ik,ins} = \|\tilde{\mathbf{r}}_{i,ins} - \tilde{\mathbf{r}}_k\|$
$\mathbf{z}_{INS} = \tilde{\boldsymbol{\rho}}_{ins}$
$\mathbf{z} = \tilde{\mathbf{h}}_{ins} - \tilde{\mathbf{h}}_{alt}$
$\mathbf{z}_{rr} = \tilde{\boldsymbol{\rho}}_{rr}$
$\mathbf{z} = \tilde{\boldsymbol{\rho}}_{ins} - \tilde{\boldsymbol{\rho}}_{rr}$
$\hat{\mathbf{r}}$
$\Delta\hat{\mathbf{r}}_{INS}$
+
−
+
−

Figure 4-1: Integration of range radio, INS, altimeter, and leader sUAS GNSS positions.

For the sUAVs that have valid and reliable GNSS available, the measurement vector **z** will be augmented with the difference between its INS and GNSS position and velocity.

5.0 MECHANIZATIONS – FACTOR GRAPH-BASED ESTIMATORS

Alternatively to sequential estimation, one can also utilize a batch of data (across sensors and across time) and obtain a maximum likelihood estimate using non-linear least squares solver tools such as g2o [27], Ceres [28], SymForce [29] and the various tools available in Matlab. These non-linear least squares solvers (NLS-Solver) have been used extensively in Simultaneous Localization and Mapping (SLAM) methods such as GraphSLAM [30].

Factors are based on the measurement equation for each of the available measurements (see Equation (4)), where $\mathbf{h}$ is a linear or non-linear function. The probabilistic representation of the measurement equation is the likelihood function; the probability density of a certain measurement vector given a certain state vector. Assuming zero-mean normally distributed noise with a covariance matrix equal to $\boldsymbol{\Sigma}_v$, this likelihood function looks like:

$$p(\mathbf{z}_k|\mathbf{x}_k) = \eta \, exp\{[\mathbf{z}_k - \mathbf{h}(\mathbf{x}_k)]^T \boldsymbol{\Sigma}_v^{-1} [\mathbf{z}_k - \mathbf{h}(\mathbf{x}_k)]\} \tag{12}$$

where η is the probability density function's normalizing constant and $\mathbf{z}_k - \mathbf{h}(\mathbf{x}_k)$ is referred to as the residual vector.

In a maximum likelihood estimator, the likelihood function can also be replaced by the log-likelihood function, as the maximum will be for identical $\mathbf{x}$:

$$\begin{gathered} \log\{p(\mathbf{z}|\mathbf{x})\} = \log(\eta) + [\mathbf{z} - \mathbf{h}(\mathbf{x})]^T \boldsymbol{\Sigma}_v^{-1} [\mathbf{z} - \mathbf{h}(\mathbf{x})] \\ \log\{p(\mathbf{z}|\mathbf{x})\} \propto [\mathbf{z} - \mathbf{h}(\mathbf{x})]^T \boldsymbol{\Sigma}_v^{-1} [\mathbf{z} - \mathbf{h}(\mathbf{x})] \end{gathered} \tag{13}$$

For measurement set $\mathcal{S} = \{\mathbf{z}_l | l = 1, \ldots, N\}$ of independent measurements, one can obtain an estimate for state vector $\mathbf{x}$ by minimizing the log likelihood sum of all observations:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \mathbf{F}(\mathbf{x}) \tag{14}$$

where $\mathbf{F}(\mathbf{x})$ is the sum of the factors associated with the available measurements $\mathcal{S}$:

$$\mathbf{F}(\mathbf{x}) = \sum_{z_l \in \mathcal{S}} [\mathbf{z}_l - \mathbf{h}_l(\mathbf{x})]^T \boldsymbol{\Sigma}_{v_l}^{-1} [\mathbf{z}_l - \mathbf{h}_l(\mathbf{x})] \tag{15}$$

Additionally, motion models can also be considered by adding factors that relate the state at various time epochs:

$$\mathbf{F}(\mathbf{x}) = \sum_{\forall k} [\mathbf{x}_k - \mathbf{g}(\mathbf{x}_{k-1})]^T \mathbf{Q}^{-1} [\mathbf{x}_k - \mathbf{g}(\mathbf{x}_{k-1})] \tag{16}$$

In a simple case, this optimization method could be replaced by a carefully formulated OLS, WLS or EKF as well. However, in general, one can obtain a maximum likelihood estimate using non-linear least squares solver tools such as g2o [21], Ceres [22], SymForce [23] and the various tools available in Matlab or Python. In our case, the factors have been described in Python using SymForce and the symbolically prepared for the optimization process.

Some relevant measurements for our test and simulation setup are shown in Figure 5-1. For each of these measurements a factor can be described and related to the absolute and relative poses defined in Section 2.0. Each of these measurements can be seen as a constraint with uncertainty. For example, the range radio measurements form a constraint between two swarm members, the GNSS pseudorange measurements a constraint between an agent and a satellite, etc.

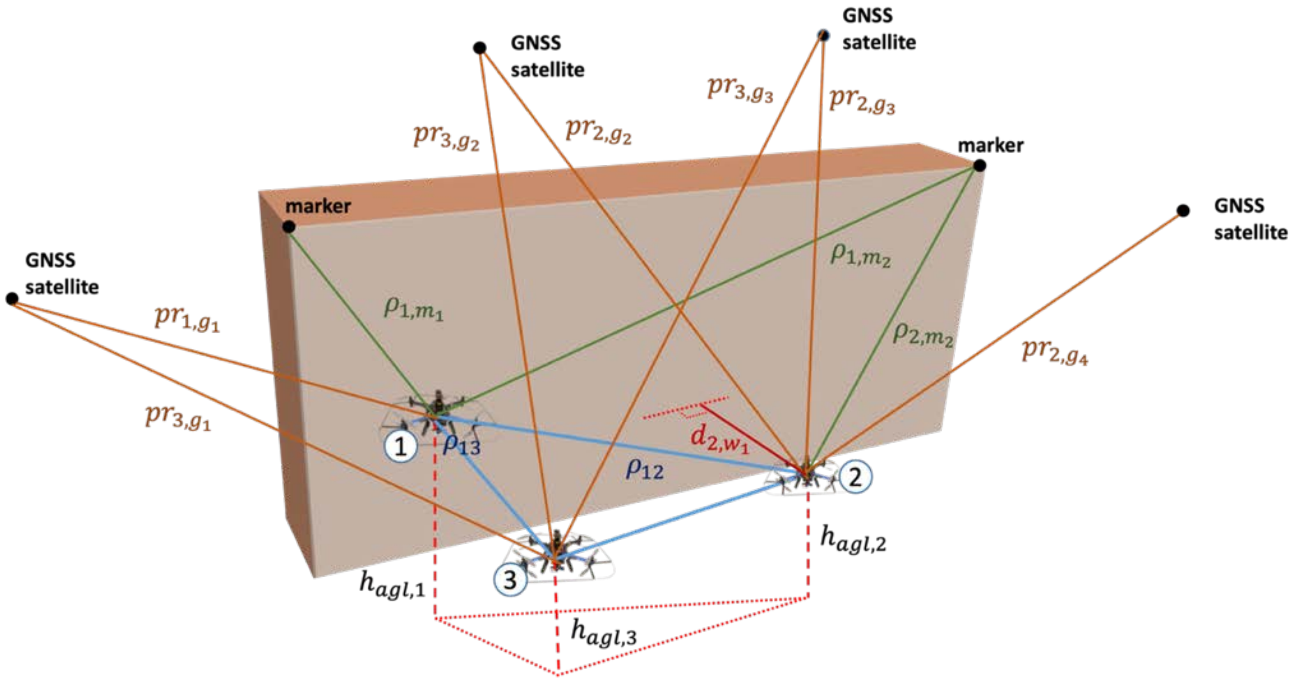


Figure 5-1: Example measurements by a small swarm of sUAVs in a relevant environment.

Note that the residuals are weighted by the inverse of the standard deviation to obtain residuals that are unit-variance.

Table 5-1: Subset of example factors.

Name	Factor	
Range radio	$F_{rr,ij} = \{\rho_{ij} - \lVert \mathbf{s}_{ij} \rVert\}/\sigma_{rho,j}$	(17)
Ranging beacons	$F_{i,m_l} = \{\rho_{i,m_l} - \lVert \mathbf{r}_{m_l} - \mathbf{r}_{n,i} \rVert\}/\sigma_{m\cancel{,j}}$	(18)
Marker beacons (direction)	$F_{i,m_l} = \{\tilde{\mathbf{e}}_{i,m_l} - (\mathbf{r}_{m_l} - \mathbf{r}_{n,i})\lVert \mathbf{r}_{m_l} - \mathbf{r}_{n,i} \rVert\}\mathbf{\Sigma}^{-1/2}$	(19)
Laser altimeter	$F_{alt,i} = \{z - h_{agl,i}\}/\sigma_{alt}$	(20)
GNSS pseudoranges	$F_{i,g_l} = \{pr_{i,g_l} - \lVert \mathbf{r}_{g_l} - \mathbf{r}_{n,i} \rVert - \delta t_{clk}\}/\sigma_{pr,g_l}$	(21)
Agent position report	$F_{i,p_j} = \{\tilde{\mathbf{r}}_{n,j} - \mathbf{r}_{n,j}\}\mathbf{\Sigma}_{p_j}^{-1/2}$	(22)
Agent position change report	$F_{i,p_j} = \{\Delta\tilde{\mathbf{r}}_{n,j} - \Delta\mathbf{r}_{n,j}\}\mathbf{\Sigma}_{dp_j}^{-1/2}$	(23)

IMU measurements are integrated using the pre-integrated IMU actor as described in detail in [34][35]. Additionally, factors for single and sequential differences can be included but their treatise is outside the scope of this paper. In addition, position and change in position reports can be considered for when the absolute position of other swarm agents is also part of the parameters to be estimated. Like mentioned earlier, it is important to perform an observability analysis to check if agent 'i' has sufficient information to also estimate agent 'j's position.

6.0 EVALUATION TOOLS

6.1 Flight Test Evaluation – Ohio University Open-Sky Test Setup

To evaluate the swarm's absolute and relative positioning capability using the method explained in Section 3.3, data from real sUAV platforms was used (see Figure 6-1 on the left). The platforms were equipped with an Odroid XU4 processor (running Ubuntu and the Robotics Operating System) to perform the data collection, with multiple inertial units of varying costs and qualities and different GNSS receivers: Platform A with a Sensonor STIM300 and a Novatel OEM-615; Platform B with a VectorNav VN-100 and Xsens Mti-1 inertial and a Novatel OEM-615 GNSS receiver, and Platform C with an Xsens Mti-1 and a U-blox M8T. In addition, the platforms were equipped with a point-to-multipoint data radio for platform-to-platform and platform-to-ground station communication. The latter was just present for monitoring purposes.

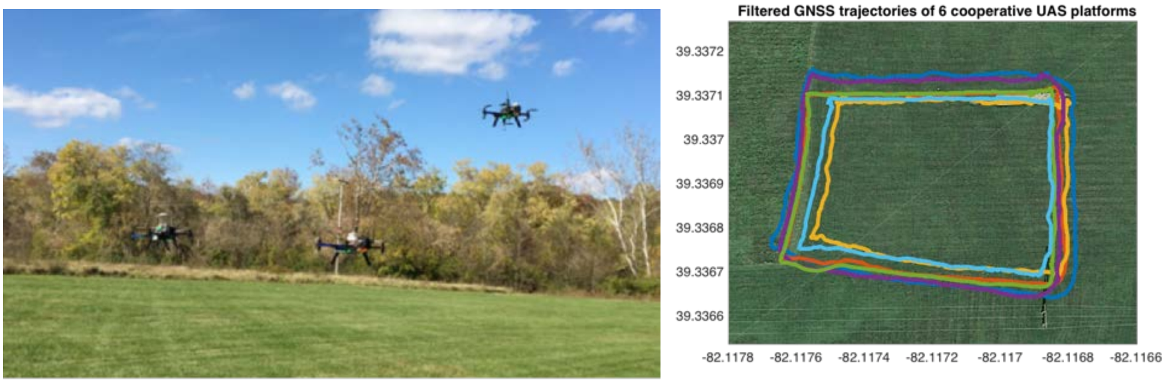


Figure 6-1: Formation flight at test field at Ohio University and overlaid flight trajectories.

During two flights, sensor data was collected, and time tagged. The two flight-data set times were adjusted and then overlaid in post-processing to generate a single data set equivalent to a six-member flight. The 6 overlaid flight trajectories are shown in (see Figure 6-1 on the right) on the right side. Range radio measurements were simulated from the GNSS position solutions and noise was added based on the ranging performance of the range radios developed by Ohio University and presented earlier in [31].

6.2 Hybrid Simulation Setup

To evaluate the various swarm navigation in a more flexible environment, a simulation environment has been used. This simulation includes 3D models of the urban environment based on CityGML LOD2 files for Berlin. In past work, this model has been used to perform extensive urban Dilution of Precision analyses for urban GNSS performance assessments.

This simulation environment allows for use of actual flight test data and simulated data (i.e., a hybrid simulation). For the example provided here, the 6-sUAV trajectories from the flight test described in Section 6.1, were geographically moved from their benign environment in Ohio to the urban environment in Berlin. The advantage of this hybrid approach is that it allows for partial reuse of the actual sensor data with their actual errors. For example, in shown case the real inertial data (accelerometers and gyro outputs), baro-altitudes and GPS data was "translated" to Berlin. For the GPS data, this means expressing all reference data (i.e., satellite positions) in a local frame. Of course, this means that the GPS satellite constellation does no longer represent the actual one. An example of what these simulation trajectories look like is shown in Figure 6-2.

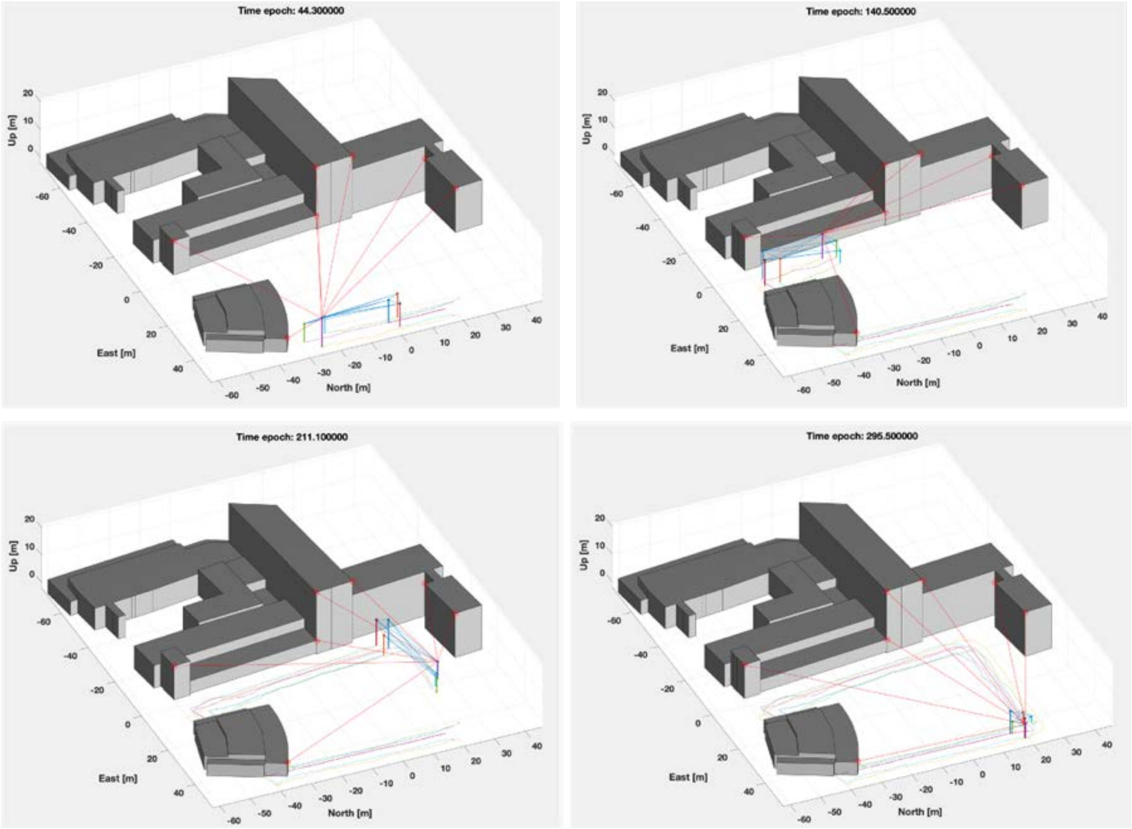


Figure 6-2: Snapshots of simulation of a swarm of 6 UAS in courtyard behind the Institute of Aeronautics and Astronautics (ILR) using GPS, range-radios, altimeters and simulated UWB beacons.

To illustrate some of the concepts, discussed in the previous sections, two cases have been implemented in the simulator and evaluated. Case I and II simulate a swarm of 6 and 8 sUAS, respectively. Both the Case I and II swarms operate in an urban environment at low altitudes and the absolute position is estimated through fixed, position referenced beacons and GNSS, respectively. Rather than being dependent on reference beacons, both cases can also be implemented using visual features whose locations are known a priori from urban databases following a method derived from the tight optical integration method described in [32]. Results of that integration are presented in Section 7. The equipment list for all swarm members is provided in Table 6-1. In Case I, a network of local beacons is installed and two of the six sUAS (i.e., leaders) are equipped with receivers that are capable of making range measurements, ρ_{i,b_k}, to these beacons $\{b_k | k = 1, \dots, M\}$. Instead, in Case II two high flying sUAS are equipped with high-end GNSS/INS platforms and range-radios. Regarding communications, the swarm is fully connected.

Table 6-1: sUAS equipment list

Case	Equipment	1	2	3	4	5	6	7	8
I	IMU	√	√	√	√	√	√	n/a	n/a
	Baro	√	√	√	√	√	√	n/a	n/a
	Beacon	√	-	-	√	-	-	n/a	n/a
	Range-radio	√	√	√	√	√	√	n/a	n/a
	GNSS	-	-	-	-	-	-	n/a	n/a
II	IMU	√	√	√	√	√	√	√	√
	Baro	√	√	√	√	√	√	√	√

	Beacon	-	-	-	-	-	-	-	-
	Range-radio	√	√	√	√	√	√	√	√
	GNSS	-	-	-	-	-	-	√	√

In terms of filters, the swarm of fleet has multiple variants available. For the sUAV equipped with the beacon system, these filters include (i) a standard OLS that uses the range measurements to the beacons and baro-altitudes (see Figure 7), (ii) and a loosely coupled linear Complementary Kalman Filter (CKF) that integrates the INS position output with the position output of the beacon/baro OLS solution (see Figure 8). In addition, the filter sets also included a tightly integrated beacons/baro/inertial mechanization, but they are not included in the Case I result section.

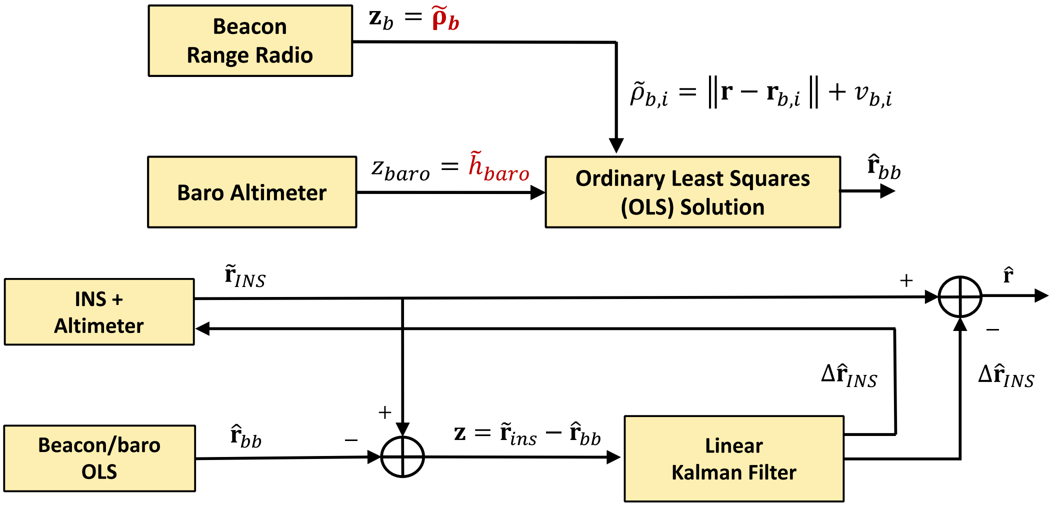

Beacon Range Radio
$\mathbf{z}_b = \tilde{\boldsymbol{\rho}}_b$
$\tilde{\rho}_{b,i} = \|\mathbf{r} - \mathbf{r}_{b,i}\| + v_{b,i}$
Baro Altimeter
$z_{baro} = \tilde{h}_{baro}$
Ordinary Least Squares (OLS) Solution
$\hat{\mathbf{r}}_{bb}$
INS + Altimeter
$\tilde{\mathbf{r}}_{INS}$
+
$\hat{\mathbf{r}}$
−
$\Delta\hat{\mathbf{r}}_{INS}$
$\Delta\hat{\mathbf{r}}_{INS}$
Beacon/baro OLS
$\hat{\mathbf{r}}_{bb}$
−
+
$\mathbf{z} = \tilde{\mathbf{r}}_{ins} - \hat{\mathbf{r}}_{bb}$
Linear Kalman Filter

Figure 6-3: Beacon/baro ordinary least squares solution (top), Loosely coupled linear Complementary Kalman Filter (CKF) (bottom).

The other Case I sUAS must rely on their INS, baro, range-radio and information received from the beacon-system equipped leaders to obtain an estimate of their position. Note that, in this case, the sUAS rely on the installed beacons and the knowledge of their positions in a global frame. Exchange of knowledge of the relative location of the beacons by some of the swarm members and exchange of this information among all members, makes this an example of principle C. Since the overall position of the individual members becomes smaller, Case I is also an example of Section 2.0 mechanism A. For their position solution, they depend on a non-linear least squares solver like the ones mentioned in Section 5.2.

6.3 Flight Test Setup – Berlin "Urban" Test Setup

To enable the evaluation of the various operational swarm concepts, sUAV platforms and associated payloads were developed. The basic sUAV platform was the Holybro S500. The sensor payload onboard the sUAV consists of common navigation-related hardware including a custom printed circuit board (PCB), i.e., the SwarmEx board, with a GNSS receiver, an IMU and a range radio. The common hardware, furthermore, includes a TF03 laser altimeter, onboard processors (RaspberryPi 4, NVIDIA nano or TX2, Odroid XU4), communication systems/data links (WiFi, LTE, Xbee or SRD890), and surveillance equipment like ADS-B and FLARM (Aerobits TR-1/F).

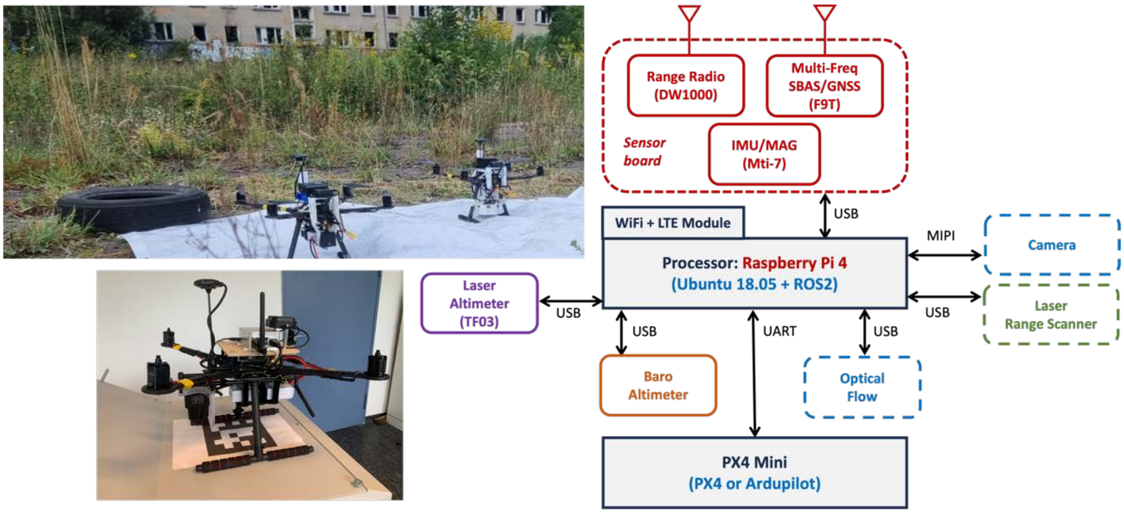


Figure 6-4: S500 Quadcopter Swarm UAVs.

The processor is connected to a PixHawk PX4 flight controller to allow for manual, semi-autonomous and autonomous operation with external position and velocity capability and the option of the processor to send it commands.

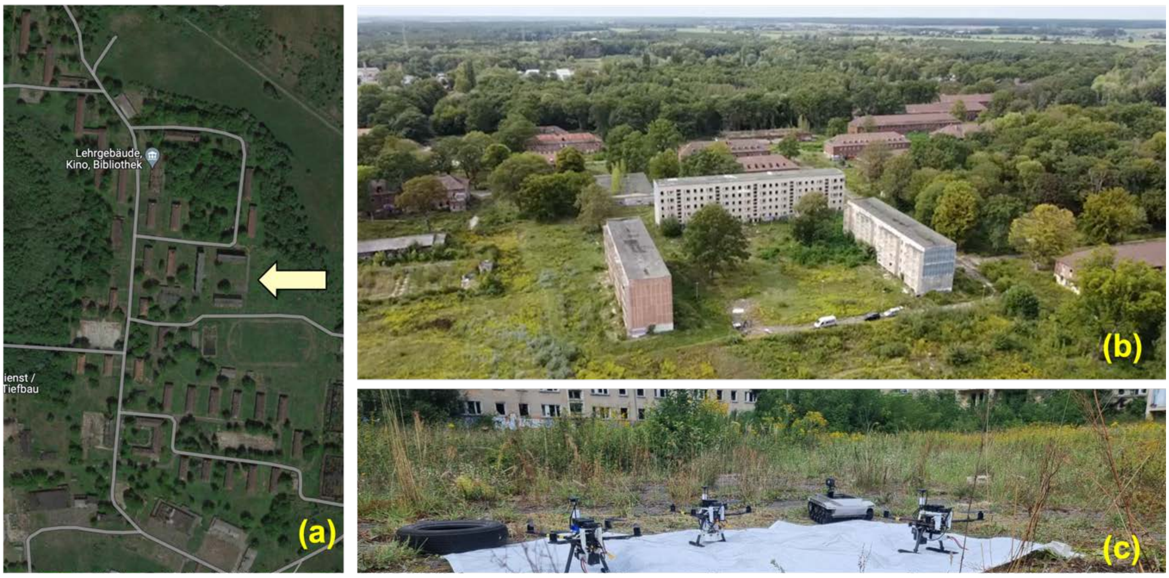


Figure 6-5: Swarm flight test facility in Schönwalde.

7.0 RESULTS

7.1 Flight Test Results

The results for the inertial-only and inertial-range-radio integration mechanizations discussed in Section 5.1 for the flight test data of Section 6.1 are shown here. The position estimate results are shown in Figure 7-1.

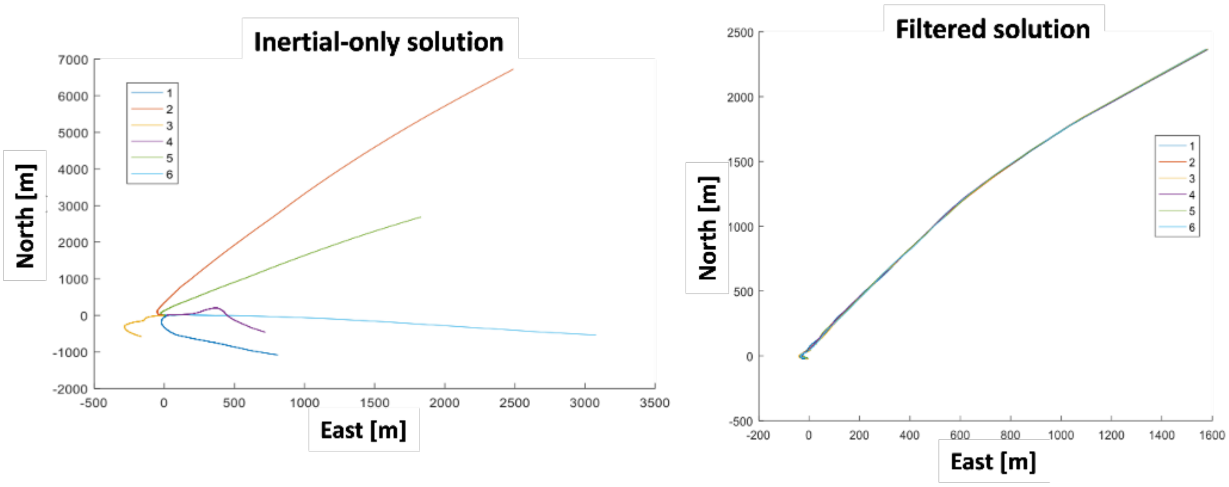


Figure 7-1: (left) positioning estimates based on inertial only; (right) inertial/range radio integration.

Whereas the scenario in which each sUAS determines its position based on its own inertial and baro-measurements shows positions estimates that drift apart over time, the filter results show that the position estimate still drifts but that now the swarm drifts due to a significant performance improvement relative navigation performance. This is even more evident from the error plots shown in Figure 7-2 where the relative navigation error plots are shown using log(y) axes to better capture both the inertial-only and the filtered results. In case one or more members are equipped with GNSS equipment and in an area where GNSS would provide fault free performance (as established by integrity-monitoring functions) or, in case of interference or denial-of-service, have resilient GNSS receivers, their knowledge of the global position can be used by the swarm.

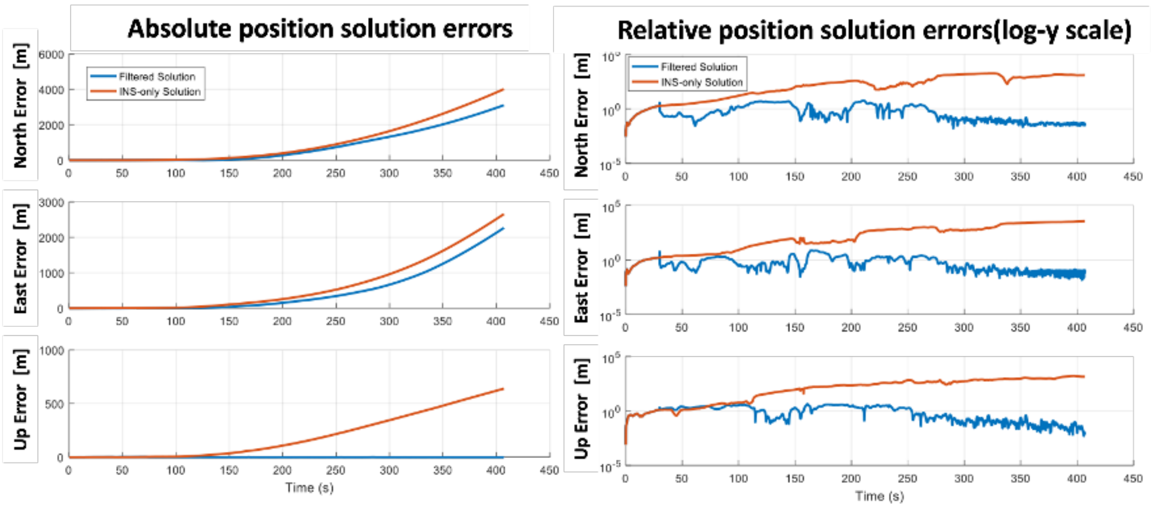


Figure 7-2: Absolute and relative positioning errors with and without filtering.

Figure 7-3 and 7-4 shows the trajectories and error plots for a varying number of leader sUAVs. The effect on the estimated trajectories when one, two or three of the six swarm members can determine their "global" position, can clearly be observed.

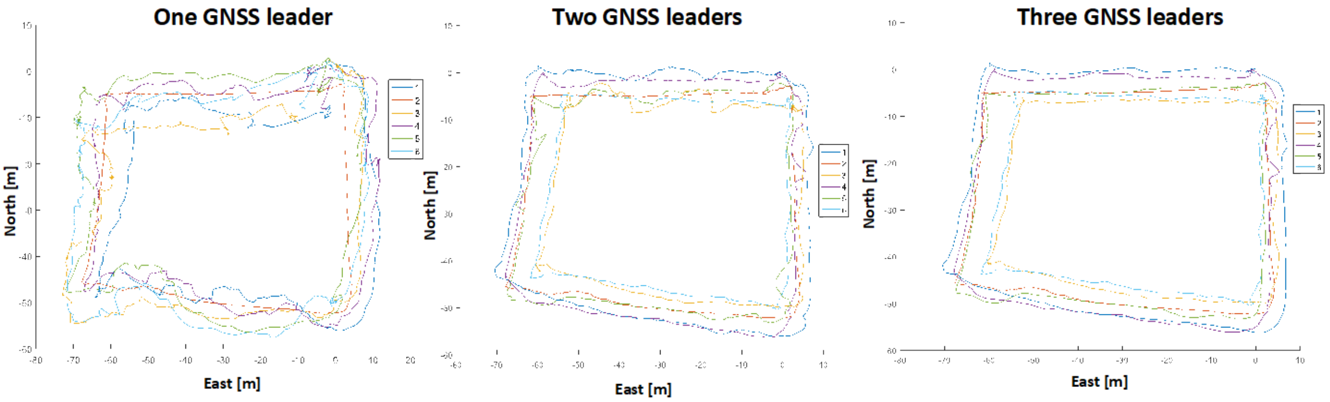


Figure 7-3: Filter results (trajectories) when one or more leaders have knowledge of the global (absolute) position due to their advanced equipment or advantageous location.

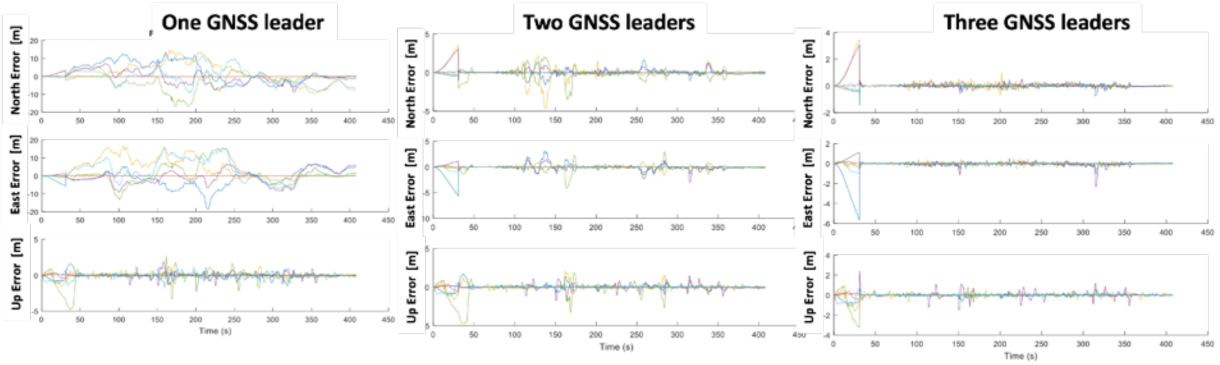


Figure 7-4: Filter results (errors) when one or more leaders have knowledge of the global (absolute) position due to their advanced equipment or advantageous location.

With a single GNSS-aided member in the swarm (sUAV 2), every member's solution becomes more accurate, both in an absolute and relative sense. The swarm is globally tied down by the GNSS-aided member and will rotate around it as the members' INSs drift while keeping relative positions intact. When the filter settles after about 30 seconds, the solution with two GNSS-aided members (sUAV 2 and 6) becomes even more accurate than with one GNSS-enabled member – the absolute error drops to near-zero from a ±20m fluctuation in the one-GNSS-enabled configuration. At this point, all three dimensions are tied down – x and y from the two GNSS-aided members, and z from the baro-altimeter.

With three points of constraint, the solution should improve, but diminishing returns are expected as the solution becomes is over-constrained.

7.2 Simulation Results

Next, both sequential and factor-graph based methods were evaluated using the simulation described in Section 6.2. Figure 7-5 shows the results for Case I and the left side depicts one of the output screens of the simulation environment. This figure is a snapshot as, typically, this figure will show an animation of the trajectories followed by the swarm. The figure also shows the filter performance on the right side for sUAV 1, 3 and 4, where sUAVs 1 and 4 are the leaders equipped with beacon system receivers. The beacon system measurements were given a noise error of 0.7m 1-σ. One would expect some multipath errors as well, but these were not modelled in this simulation. As can be seen in Figure 7-5, both the OLS (UAS #4) and the integrated beacon/inertial, show good results with an expected noise reduction in the latter.

The factor-based solution (NLS solver) plot for sUAV #3 shows good results as well. The large uncertainty at the beginning can be explained by a bad geometry formed by the two leaders and sUAV #3. With some of the others that is not the case. It would therefore have to be part of the *comprehension* and *projection* module to predict that this will occur, and define appropriate actions (trajectory changes) to avoid such a bad geometry. To evaluate that, the simulation tool will have to be adjusted as the trajectories can no longer be based on playback data.

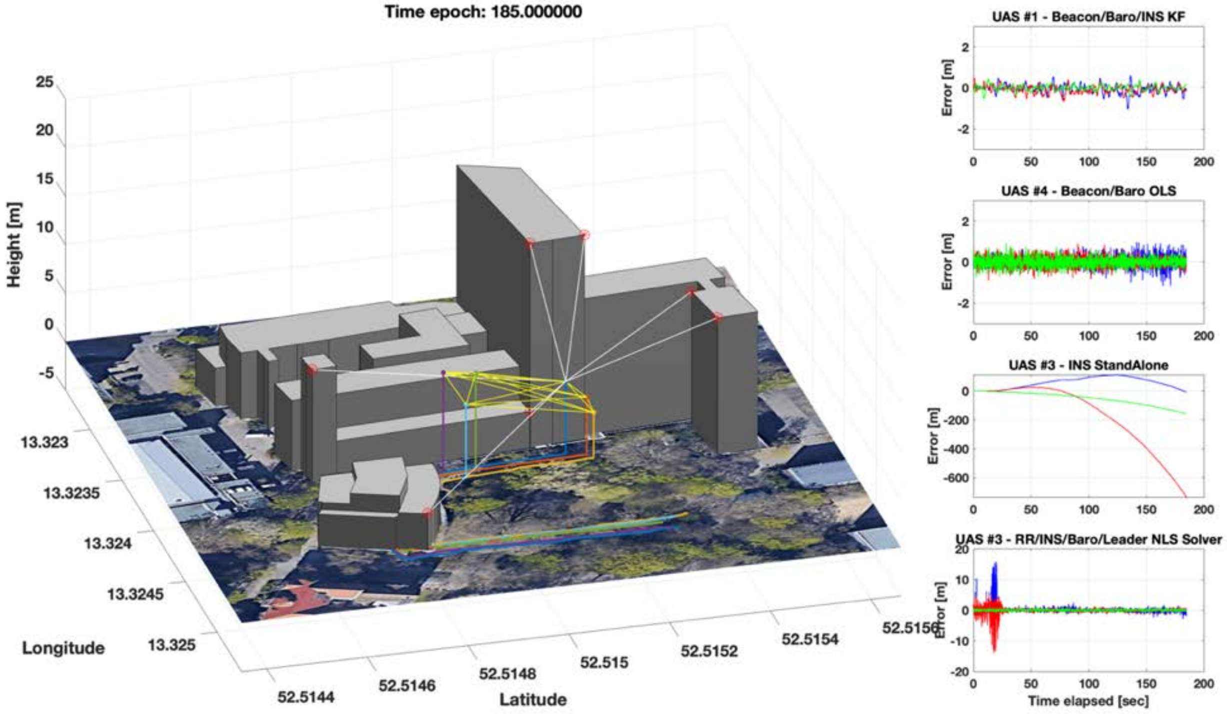


Figure 7-5: Simulation Case I: Swarm of 6 sUAS performing a collaborative mission in an urban environment using a locally installed beacon system.

The results for the Case II scenario are shown in Figure 7-6. As expected, the OLS results are noisier due to the noise contributions of the leaders' position estimates. This is less visible in the KF results. Again, the NLS-Solver results of sUAV #3 are quite similar including the larger error during the time that sUAV #3 forms a bad geometry with sUAV #1 and #4. The NLS-Solver results are still noisy, and it is expected that the inclusion of IMU measurements as well as optimizing using a time-window of data (like in GraphSLAM) will smooth the results. Alternatively, a EKF implementation may smooth the trajectories. Of course, no accuracy (and other navigation performance) targets were defined for this paper, so it is hard to say if the results meet the target levels of safety.

7.0 SUMMARY AND CONCLUSIONS

This paper introduced the initial foundations of a cognition and collaboration approach to absolute and relative navigation of swarms of sUAV based, in part, one some basic principles of swarm navigation in nature. Results from a flight test and a simulation demonstrated that the leadership and social learning principles do apply nicely and that by evaluating the necessary constraints filters can be defined that allow some of the swarm members to operate in GNSS-challenging environments.

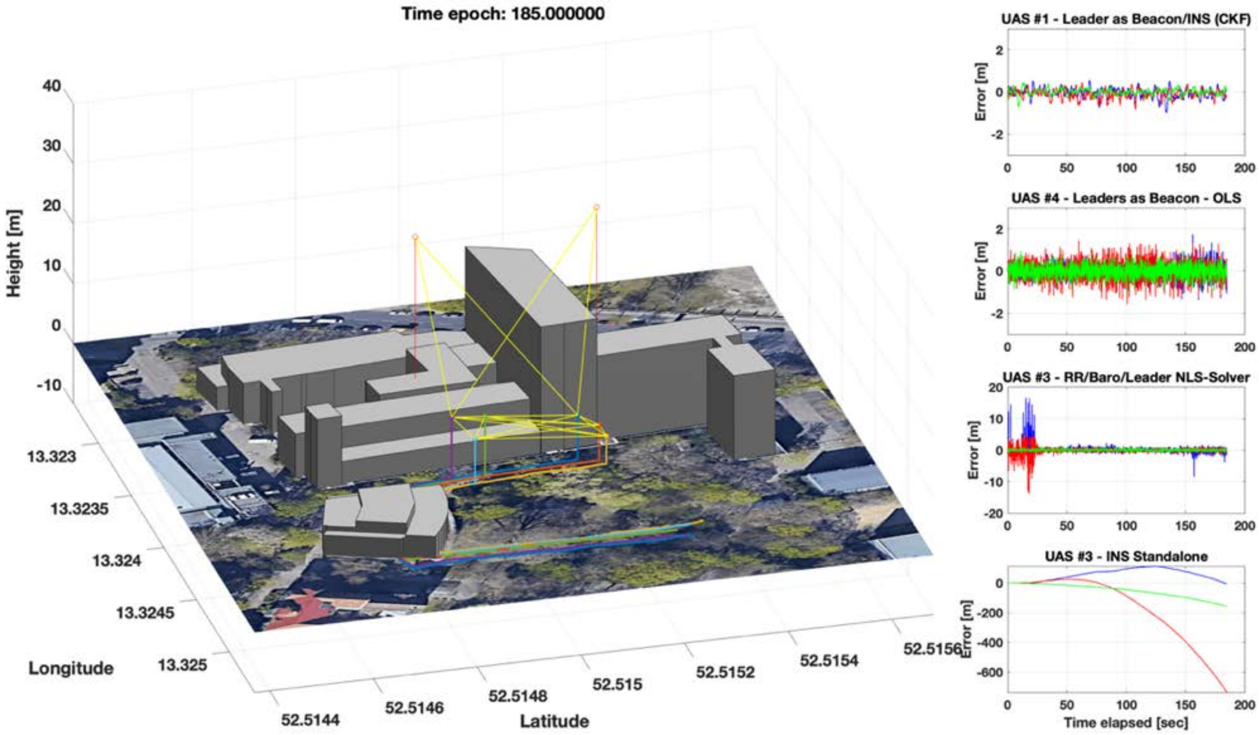


Figure 7-6: Simulation Case II: Swarm of 8 sUAS performing a collaborative mission in an urban environment using two leaders with high-performance navigation equipment at higher altitudes.

So far, the methods have mainly addressed the accuracy performance of the navigation solution. The next steps will address the inclusion of other navigation performance parameters (i.e., integrity, continuity, and availability). Once those have been established, the situational awareness module can be completely implemented by including methods to predict what actions (i.e., short- and mid-term trajectory changes) of the swarm and its members will lead to a continued safe operation. To evaluate these new methods the simulation tool must be updated and improved flight test in a relevant operation prepared.

Finally, the proposed method must be aligned with the large volume of communication literature to minimize the utilized communication bandwidth, while maintaining safe operation.

8.0 REFERENCES

[1] Y. Tan, Z. Zheng, “Research Advance in Swarm Robotics,” in *Defence Technology*, ISSN: 2214-9147, Vol: 9, Issue: 1, 2013, pp. 18-39.

[2] JARUS, “JARUS Guidelines on Specific Operations Risk Assessment (SORA),” JAR-DEL-WG6-D.04, January 2019.

[3] G. Vásárhelyi, Cs. Virágh, G. Somorjai, T. Szörényi, T. Nepusz, A. E. Eiben, T. Vicsek, “Optimized flocking of autonomous drones in confined environments,” *Sci. Robot.* 3, 2018.

[4] T. Indriyanto, et al., “Centralized swarming UAV using ROS for collaborative missions,” *AIP Conference Proceedings* 2226, 030012 (2020); https://doi.org/10.1063/5.0002616, June 2020.

[5] A. Pospischil, “Mapping ad hoc communications network of a large number fixed-wing uav swarm,” Thesis, Naval Postgraduate School, Monterey, California, 2017.

[6] M. Campion, P. Ranganathan, and S. Faruque “UAV swarm communication and control architectures: a review,” *J. Unmanned Veh. Syst.*, Vol. 7, 2019.

[7] Intel, https://www.intel.com/content/www/us/en/technology-innovation/aerial-technology-light-show.html, Accessed on 12 August 2019.

[8] Ehang, https://www.ehang.com/news/249.html, Accessed on 12 August 2019.

[9] J. E. Huff and M. Uijt de Haag, "Assured relative and absolute navigation of a swarm of small UAS," in *Proceedings of the IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, St. Petersburg, FL, USA, 2017.

[10] Farrell, J. L., *GNSS Aided Navigation & Tracking – Inertially Augmented or Autonomous*, American Literary Press, 2007.

[11] M. M. Miller, M. Uijt de Haag, A. Soloviev, M. Veth, “Navigating in Difficult Environments: Alternatives to GPS – 1,” Proceedings of the NATO RTO Lecture Series on “Low Cost Navigation Sensors and Integration Technology,” SET-116, November 2008.

[12] M. M. Miller, J. Raquet, M. Uijt de Haag, “Navigating in Difficult Environments: Alternatives to GPS – 2,” Proceedings of the NATO RTO Lecture Series on “Low Cost Navigation Sensors and Integration Technology,” SET-116, November 2008.

[13] H.-L. Choi, L. Brunet, and J. P. How, ‘‘Consensus-based decentralized auctions for robust task allocation,’’ *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912–926, Aug. 2009.

[14] S. Raja, G. Habibi, J. P. How, “Communication-Aware Consensus-Based Decentralized Task Allocation in Communication Constrained Environments,” *IEEE Access*, 2022.

[15] C. A. Freas, P. Schultheiss, “How to Navigate in Different Environments and Situations: Lessons from Ants,” *Front. Psychol.*, 29 May 2018.

[16] R. Menzel, et al., “Honeybees navigate according to a map-like spatial memory,” *Proc. Natl. Acad. Sci.* U.S.A., February 2005.

[17] A. M. Berdahl, et al., “Collective Animal Navigation and Migratory Culture: from Theoretical Models to Empirical Evidence,” Phil. Trans. R. Soc. B37320170009, March 2018.

[18] F. Causa, “Planning Guidance and Navigation for Autonomous Distributed Aerospace Platforms,” Ph.D. Dissertation, Università degli Studi di Napoli “Federico II”, March 2020.

[19] M. R. Endsley, “Toward a Theory of Situation Awareness in Dynamic Systems,” *Human Factors*, 1995, 37(1).

[20] S. Haykin, J. M. Fuster, “On Cognitive Dynamic Systems: Cognitive Neuroscience and Engineering Learning From Each Other,” *Proceedings of the IEEE,* Vol. 102, No. 4, April 2014.

[21] Robotic Operating System 2, Accessed on 5 August 2023, https://docs.ros.org/en/rolling/Releases/Release-Humble-Hawksbill.html.

[22] C. W. Reynolds, “Flocks, Herds, and Schools: A Distributed Behavioral Model,” *Computer Graphics*, 21(4), July 1987, pp. 25-34.

[23] B. J. Julian, M. Angermann, M. Schwager, D. Rus, „Distributed Robotic Sensor Networks: An Information Theoretic Approach,” *The International Journal of Robotics Research*, Vol. 31 (10).

[24] M. Martens, M. Uijt de Haag, „Cooperative Swarm Geometry Optimization for Assured Navigation with Range Radios in GNSS Denied Environments,*” Proceedings of the European Navigation Conference (ENC)*, 2023.

[25] P. S. Maybeck, *Stochastic Models, Estimation and Control – Vol. I*, Navtech Books and Software Store, 1994.

[26] B. Ristic, et al., *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, 2004.

[27] R. Kuemmerle, et al., “g2o: A General Framework for Graph Optimization,” Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2011.

[28] S. Agarwal, K. Mierle, et al., “Ceres Solver”, http://ceres-solver.org.

[29] H. Martiros, et al., “SymForce: Symbolic Computation and Code Generation for Robotics,” Proceedings of Robotics: Science and Systems, 2022.

[30] S. Thrun and M. Montemerlo, “The graph SLAM algorithm with applications to large-scale mapping of urban structures,” *Int. Journal of Robotics Research*, 25(5-6):403, 2006.

[31] J. Huff, Absolute and Relative Navigation of an sUAS Swarm Using Integrated GNSS, Inertial and Range Radios, *M.S.E.E. Thesis*, Ohio University, 2018.

[32] M. Uijt de Haag, Z. Zhu, T. Arthur, “Performance Analysis and Integrity Aspects of Tight Optical Integration (TOI) with GPS,” *Proceedings of the IEEE/ION Position Location and Navigation Symposium (PLANS)* 2008, Monterey, CA, May 5-8, 2008, pp. 846-850.

[33] P. D. Groves, Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Artech House, 2008.

[34] T. Lupton and S. Sukkarieh, “Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions”, TRO, 28(1):61-76, 2012.

[35] C. Forster, et al., “IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation”, *Robotics: Science and Systems (*RSS), 2015.