Cooperative 3-D relative localization for UAV swarm by fusing UWB with IMU and GPS

Yang Qi, Yisheng Zhong and Zongying Shi
Department of Automation, Tsinghua University, Beijing 100084, P. R. China
E-mail: qiy17@mails.tsinghua.edu.cn; zys-dau,szy@mail.tsinghua.edu.cn

Abstract. A real-time 3-D relative position estimator is presented for fixed-wing UAV swarm system in formation flight, using measurements from IMU, compass, GPS, and a set of UWB ranging radios. Instead of using stationary UWB anchors to provide 3-D coordinate and fusing it with IMU, the estimator uses UWB measurements for the construction of a non-convex function, and calculates the global optimum of the non-convex function to get position estimation, while the GPS positions are used to provide bearing information. Actual flight experiments with seven fixed-wing UAVs are conducted outdoors, and the experimental results indicate that the proposed scheme achieves lower distance estimation error between neighbor UAVs compared to traditional GPS/IMU/UWB fusing algorithm using EKF.

1. Introduction

Relative positioning between multiple mobile agents with high accuracy is essential for many applications, such as formation flight, cooperative control and mapping for the swarm system consisting of multiple Unmanned Aerial Vehicles (UAVs). In formation flight, each UAV is controlled based on the relative position of its neighbors. For localization and mapping, the relative position between UAVs is necessary for information fusion to construct a map of the environment.

For typical UAV applications, the position estimation of vehicles accomplished by low-cost navigation system using inertial measurement unit (IMU) and GPS with traditional fusion algorithm generally has the localization precision with 1∼2m horizontal locating error and 2∼5m vertical locating error. However, for the UAV swarm, where the distances between vehicles are very small compared to normal flight mission, more precise localization is necessary. In outdoor, RTK GPS has been used for locating automobiles with very high accuracy, however, a base station which broadcasts the pseudo range difference information to the vehicle equipped with RTK GPS receiver is required, so the valid range of RTK GPS receiver is constrained within the coverage area of the base station signal. In cases such as fixed-wing UAV formation flight, the mission of the UAV swarm covers a large area where the signal of the base station is not available, so RTK GPS is not an option for such cases.

A relatively new and appealing method of localization utilizes low power, low cost Ultra-Wide Band (UWB) radio modules, which measures relative distances between modules by counting the time interval of UWB pulses transmission and reception [1], and has an accuracy of less than 10 cm. UWB signals are particularly suitable for localization systems due to their high accuracy as well as the ability to operate in Non-Line-of-Sight (NLOS) conditions. Furthermore,
it provides a channel for data communication and supports data transfer at a rate up to 6.8 Mbps.

The majority of work on UWB localization require several stationary landmarks (anchors) with known coordinates \([2][3][4][5][6]\). First, the moving vehicles carrying UWB (tags) initiate measurement to anchors and the time differences of arrival (TDOA) or time of flight (TOF) at the tags are determined. Using the TDOA or TOF, tags’ location relative to the anchors can be calculated using the Taylor-series method \([7]\) or least-squared method (or Chan method)\([8]\). For estimating the position relative to UWB anchors, several sensor fusion algorithms can be utilized. In \([2]\), a 2-D UWB positioning system combined with the IMU information that has an accuracy of less than 4 cm is described. In \([3]\), IMU/UWB sensor fusion is accomplished with Extended Kalman Filter (EKF), which estimates the IMU sensors’ systematic errors and corrects the positioning errors, and the laboratory test for estimating the motion of a 3 DOF platform is present. In \([4]\), a method estimating the state consisting of position, velocity, orientation, and angular velocity for a quadrocopter is presented, using measurements from an IMU and a set of UWB modules. In \([6]\), a cooperative localization method using only UWB ranging information is proposed, which require one landmark whose global position is known to a subset of agents.

It is possible to achieve accurate position estimation using UWB when anchors with known coordinates are deployed. Due to that UWB does not provide any bearing information when measuring the inter-module distance, thus the tag location cannot be determined by UWB alone if anchors are absence. In \([9]\), cooperative relative positioning of mobile users without UWB anchor is presented, where a probability-based framework is introduced, the IMU Inertial and UWB ranging information are fused using particle filter, and a 2-D motion experiment is conducted. In \([10]\), a scheme of relative localization in which all agents cooperatively estimate the relative positions of their neighbors in real-time using distance measurement and relative velocity between neighbors is present. The scheme is used in 2-D localization and only simulation result is given.

These problems provide the motivation for the work presented in this paper. Here we propose an approach to combine UWB ranging measurement and GPS coordinates of the UAV swarm for 3-D relative positioning in real-time without deployment of any stationary anchor. An illustration with four vehicles in an UAV swarm is given in figure 1, where the dot lines connecting two vehicles indicate that they can range and exchange sensor information with each other using UWB.

An optimization method is presented to solve the relative localization problem. First we incorporate the UWB and the IMU measurement to reject the ranging outlier and recover the distance between neighbors when ranging failure occurs, next we use these distance estimations to construct a rigid body composed of these moving nodes and edges of certain length by finding the global minimum of a non-convex function, finally the orientation of the rigid body system
is calculated using GPS coordinates of several nodes. To solve the non-convex optimization problem with constraints, a dedicated algorithm is designed to reduce the computational load while convergence is guaranteed in finite steps. We have conducted outdoor experiments with seven fixed-wing UAVs to demonstrate the effectiveness of this locating algorithm.

The paper is organized as follows: the problem statement and notation used in formulation are given in Sect.2. Sect.3 presents the sensor models used for localization. An optimization algorithm for relative position estimation is given in Sect.4. The approach is validated in experiment in Sect.5, and the paper concludes with Sect.6.

2. Problem Statement

The paper addresses the cooperative 3-D relative localization problem, for which the goal is to estimate the relative coordinates of all the other vehicles for each UAV in a common local NED coordinate frame (N frame) based on IMU/UWB/GPS measurements and limited information exchange between neighbors.

Assuming the swarm system consisting of N fast moving UAVs labeled 1, 2, ..., N, each vehicle i carries IMU/Compass/GPS sensors and has an onboard Altitude and Heading Reference System (AHRS) running on the flight control unit, which is able to access its own GPS position \( r_{i,GPS} = [x_{i,GPS} \ y_{i,GPS} \ z_{i,GPS}]^T \) and provide velocity estimation \( \hat{v}_i = [\hat{v}_{xi} \ \hat{v}_{yi} \ \hat{v}_{zi}]^T \) and acceleration estimation \( \hat{a}_i = [\hat{a}_{xi} \ \hat{a}_{yi} \ \hat{a}_{zi}]^T \) in N frame. Also, assume that each vehicle can get UWB ranging value about its neighbors. Let \( N_i \) represent the neighbor set of vehicle i, and for each vehicle \( j \in N_i \), the distance between vehicle i and vehicle j, denoted as \( d_{ij} \), can be measured mutually. Moreover, vehicle i and j can exchange sensor information, such as their own GPS coordinate \( r_{i,GPS} \), velocity \( \hat{v}_i \) and acceleration estimation \( \hat{a}_i \) as well as their UWB ranging data with their neighbors \( d_{ik}, k \in N_i \). It is important to note that \( N_i \) might be time-varying. We use an undirected graph \( G = (V, E) \) of N nodes to represent the sensing topology, where \( V = \{1, ..., N\} \) corresponds to the N vehicles, and there is an edge \((i, j) \in E \) if \( j \in N_i \) and \( i \in N_j \). The graph \( G \) might be time-varying due to possible ranging or communication failures between the neighbors.

We aim to determine accurate relative 3-D positions estimation \( \hat{r}_{ij} \) in N frame for every vehicle pair i and j, even when edge \((i, j) \notin E \), as long as \( G \) is a connected graph.

3. Sensor Modeling

3.1. UWB

A TOA-based ranging method known as Single-sided Two-way Ranging (SS-TWR) is used for UWB measurement. The operation of SS-TWR is shown in figure 2, where device i initiates the measurement and device j responds to complete the data exchange and each device precisely
stamps the transmission and reception times of the message frames, and so round times can be calculated by simple subtraction. And the distance $d_{ij}$ can be estimated by equation (1):

$$d_{ij} = \frac{1}{2}(T_{round} - T_{reply}) \times c$$

where $c$ represents the constant of light speed. The high resolution of timestamp enables accurate range measurement; however, $T_{round}$ and $T_{reply}$ are counted in different device, and due to clock frequency differences between UWB modules, the timing error are introduced into the measurement, $d_{ij}$ is assumed to be the true distance between node $i$ and $j$ corrupted by zero mean white noise $\eta_{UWB}$ as formula (2).

$$d_{ij} = \|r_i - r_j\| + \eta_{UWB}$$

where $r_i$ and $r_j$ are the true position of node $i$ and node $j$, $\|\|$ represents the Euclidean norm, and $\eta_{UWB} \sim N(0, \sigma^2_{UWB})$, here the standard deviation $\sigma_{UWB}$ is set to 0.1m.

The UWB module is set to operate at a rate of 100Hz. During each measuring interval, for every node $i \in V$, the distance $d_{ij}$ is measured ($j \in N_i$). After the measurement period, all the ranging data of node $i$ is transmitted to all its neighbors. In figure 1, the distance between vehicle 2 and vehicle 3 is not available due to long range or communication error, however, $d_{13}$ and $d_{34}$ are still available for vehicle 2 through UWB communication with vehicle 1 and vehicle 4 respectively.

3.2. GPS

A NEO-M8N GPS module produced by U-blox is used to provide global coordinates, and the global coordinates of all the vehicles are transformed into N frame. In this subsection, the GPS relative coordinates are modeled for filtering usage. For simplicity, we assume the relative errors in each axis are zero mean white noises with zero covariance as formula (3).

$$r_{ij,GPS} = r_{i,GPS} - r_{j,GPS} = r_{ij} + \eta_{GPS}$$

where $r_{ij}$ represents the true relative coordinate between node $i$ and node $j$, $\|\|$ represents the Euclidean norm, and $\eta_{GPS} \sim N(0, \sigma^2_{GPS}, \sigma^2_{GPS}, \sigma^2_{GPS})$, the standard deviation $\sigma_{x,GPS}, \sigma_{y,GPS}, \sigma_{z,GPS}$ are set to 0.6m, 0.6m, 0.8m respectively. The GPS update rate is configured to 5Hz.

3.3. IMU and Compass

The IMU measures the vehicle’s angular velocity and acceleration in the vehicle body frame (B frame), and compass measures local geomagnetic field in B frame, thus the relative pose between B frame and N frame can be calculated, and the vehicle’s acceleration $a$ in N frame and the accelerometer measurement $z_{acc}$ satisfy formula (4).

$$z_{acc} = C_B^N (a - g) + \eta_{acc}$$

where $C_B^N$ is the cosine matrix of B frame respect to N frame, $g$ is gravitational acceleration and $\eta_{acc}$ is zero-mean white noise. We can obtain the acceleration estimation $\hat{a}$ in N frame using formula (4), and acquire velocity estimation $\hat{v}$ by integrating and fusing with GPS velocity measurement.

The Pixhawk PX4 flight management unit is used, on which an EKF estimator runs and outputs the velocity and acceleration estimations in N frame through Mavlink protocol in 10Hz. The state estimations $\hat{a}_i$ and $\hat{v}_i$ as well as the covariance matrix are transmitted to all its neighbors to calculate the relative state as formula (5).

$$\begin{aligned}
\hat{a}_{ij} &= \hat{a}_i - \hat{a}_j = a_{ij} + \eta_{acc} \\
\hat{v}_{ij} &= \hat{v}_i - \hat{v}_j = v_{ij} + \eta_{v}
\end{aligned}$$
where $\eta_{acc}$ and $\eta_v$ are assumed to be zero-mean white noise. $\text{Cov}[[\eta_{acc}, \eta_v],[\eta_{acc}, \eta_v]]$ is set according to the value returned by PX4.

4. Algorithm for Position Estimation

4.1. UWB Ranging Data Preprocessing

Before relative position estimation, data preprocessing step is performed for the following purpose:

- **Ranging outlier rejection**: Outliers from the UWB ranging may be detected by computing likelihood of a given ranging measurement. If the difference between the distance estimation and the actual range measurement is larger than a given threshold, then it is rejected as an outlier.

- **Ranging failure data completion**: If a ranging failure occurs, the current distance can be estimated using the state estimator.

The data preprocessing step is accomplished through the use of an Extended Kalman Filter (EKF). For each node $i \in \mathcal{V}$ and each edge $(i, j) \in \mathcal{E}$, an EKF is established to give a distance estimation $\hat{d}_{ij}$. The state includes the relative position and relative velocity from $i$ to $j$, that is $x[t] = [r_{ij}[t] \quad v_{ij}[t]]^T$, and the system can be modeled with a linear state equation and a time-varying nonlinear measurement equation, expressed as (6).

$$
\begin{aligned}
\dot{x}[t+1] &= A \cdot x[t] + B \cdot u[t] + \Gamma \cdot \xi[t] \\
z[t] &= g(x[t], t) + \eta[t]
\end{aligned}
$$

The relative acceleration $\hat{a}_{ij}[t]$ is treated as system input $u[t]$, thus the process noise $\xi[t]$ is equal to $\eta_{acc}$, and the state transfer matrix $A$ and input matrix $B$ are given as follows:

$$
A = \begin{bmatrix}
I_3 & I_3 \cdot \Delta T \\
0_{3 \times 3} & I_3
\end{bmatrix},
B = \begin{bmatrix}
\frac{1}{2} \Delta T^2 \cdot I_3 \\
\Delta T \cdot I_3
\end{bmatrix}
$$

The prediction time step $\Delta T = 0.01s$ is determined by the sensor measurement frequency $f_{\text{UWB}}$. The measurement function $g(x[t], t)$ is a time varying nonlinear function, and $z[t] \in \mathbb{R}^1, \mathbb{R}^3, \mathbb{R}^4, \mathbb{R}^6$ or $\mathbb{R}^7$ depending on whether GPS position, velocity or UWB measurement is used.

![Figure 3. Data preprocessing work flow using EKF.](image-url)
is available during this period. For period during which all the sensor data are available, $z[t] = [r_{ij,GPS}; \hat{v}_{ij}; d_{ij}] \in \mathbb{R}^7$, and $g(x[t], t)$ are given as (7), and $\eta[t] = [\eta_{GPS}; \eta_{v}; \eta_{UWB}] \in \mathbb{R}^7$.

\[
g(x[t], t) = \left[ I_6 \cdot x[t] \left/ \|r_{ij}[t]\| \right. \right] (7)
\]

After each EKF state propagation step and before state correction step show in figure 3, $\hat{d}_{ij}[t|t-1]$ is calculated by $\hat{d}_{ij}[t|t-1] = \|r_{ij}[t-1]\|$, if $|\hat{d}_{ij}[t|t-1] - d_{ij}| > 5 \cdot \sigma_{UWB}$, $d_{ij}$ is regarded as outlier and rejected. Under any circumstances, $\hat{d}_{ij}[t|t] = \|r_{ij}[t]\|$ is calculated after the state correction step, and used for the later position estimation.

4.2. Relative Position Estimation

We treat the swarm system represented by graph $G$ as a spring system, where each vehicle is a mass point with identical mass, and there are constraints of elastic force between vehicles. For vehicle $i$ and its neighbor $j \in N_i$, an ideal spring with original length $\hat{d}_{ij}$ and stiffness coefficient $k_0$ connects the two nodes. When the spring system goes into stable state, the position of each node is regarded as the final vehicle position estimation. We regard the swarm system as a rigid body. When the node number in $G$ is greater than 4, the swarm system will have 3-DOF of translation and 3-DOF of rotation, so at least 4 GPS coordinates are needed to get a unique solution. For node $i$, an ideal spring with length 0 and stiffness coefficient $k_i$ is placed between $\hat{r}_i$ and $r_{i,GPS}$, as shown in figure 4. There are 4 nodes and the GPS coordinate of each node is available. $G$ is a fully connected graph, which means all the UWB distance measurements between every node pair are known.

![Figure 4. An illustration of the spring system.](image)

Assume $m$ GPS coordinates $r_{i_1,GPS}, r_{i_2,GPS}, r_{i_m,GPS}$ are used in the system, then the total potential energy $J$ of the swarm system is:

\[
J(\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_n) = \frac{1}{4} k_0 \sum_{i < j \in V} (\|\hat{r}_i - \hat{r}_j\| - \hat{d}_{ij})^2 + \frac{1}{2} \sum_{l=1}^{m} k_l \|\hat{r}_{i_l} - r_{i_l,GPS}\|^2 (8)
\]

**Theorem 1 (Minimum total potential energy principle).** *a spring structure or body shall deform or displace to a position that (locally) minimizes the total potential energy.*[11]

Theorem 1 tells us that when the spring system goes into stable state, position $\hat{r}_i$ will minimize $J$, thus we turn the estimation problem into a non-convex optimization problem. For objective
function (8), the Hessian matrix about \( \hat{r} \), that is \( \nabla^2_J = \frac{d^2 J}{d \hat{r}^2} \), is not positive definite for all possible \( \hat{r} \). Special numerical descent method (see algorithm 1 for details) is used to calculate the global minimum. Instead of computing the gradient and Hessian about \( \hat{r} \), the gradient and Hessian about \( \hat{r} \) is calculated each time, and \( \hat{r} \) is optimized in order using Gradient Descent method or Newton method, depending on whether the Hessian about \( \hat{r} \) is positive defined or not. Numerical test shows that this algorithm will converge to a stable point before exceed the maximum iteration bound in most cases, if a proper initial point is given.

Algorithm 1: Iterative Descend Algorithm

Require: \( J(\hat{r}(0)) \): objective function;
\( \hat{r}(0) = [\hat{r}_1^{(0)T}, \hat{r}_2^{(0)T}, \ldots, \hat{r}_n^{(0)T}]^T \): initial solution;
Ensure: global optimum \( \hat{r}^* \)
1: initial \( \Delta r = \infty \) and \( m = 0; /\Delta r: \text{Step size, m: Iteration} \); 2: while \((m < \text{Iteration Bound})\&\& (\Delta r > \text{Step Bound}) \) do 3: for \( i = 1; i \leq n; i++ \) do 4: Compute \( \frac{dJ}{dr_i} |_{\hat{r} = \hat{r}(m)}; \quad J_{ii} |_{\hat{r} = \hat{r}(m)} \); 5: if \( J_{ii} \leq 0 \) then 6: \( dx = -\frac{dJ}{dr_i}; \quad /\text{use Gradient Descent method} \); 7: else 8: \( dx = -J^{-1}_{ii} \cdot \frac{dJ}{dr_i}; \quad /\text{use Newton's method} \); 9: end if 10: Use Backtracking Line Search method to find an appropriate \( t \); 11: \( \hat{r}^{(m)} = t \cdot dx \); 12: Set \( \hat{r}^{(m)} = [\hat{r}_1^{(m)T}, \ldots, \hat{r}_i^{(m)T}, \hat{r}_{i+1}^{(m-1)T}, \ldots, \hat{r}_n^{(m-1)T}]^T \); 13: end for 14: end while 15: Set \( \hat{r}^* = \hat{r}^{(m)} \); 16: Check whether \( J_{ii} > 0, \forall i \in \{1, 2, \ldots, n\} \);

Set \( k_l = k_1, l = 1, \ldots, m \), let \( k = \frac{k_1}{k_0} \), numerical experiment indicates that lower \( k \) will decrease the influence of GPS error and increase confidence in UWB measurements thus increase the estimation precision. However, lower \( k \) will decrease the convergence speed of algorithm 1 when step bound \( \Delta r \) hold the same. In particular, when \( k \to 0 \), the algorithm above will not converge to the global minimum in finite time. An algorithm is designed to find the minimum of \( J \) when \( k \to 0 \): First, remove all GPS nodes and calculate the rigid body structure using only distance estimation \( \hat{d}_{ij} \), then determine the position and orientation of the rigid body with valid GPS coordinates of the nodes in \( G \), for \( \|V\| \geq 4 \), no less than 4 GPS coordinates are needed to eliminate DOF.

4.2.1. Calculate the rigid body structure. First calculate the rigid body structure by finding the minimum of equation (9) using algorithm 1.

\[
\text{minimize} \quad J(r'_1, r'_2, \ldots, r'_n) = \sum_{i \in V} \left( \|r'_i - r'_j\| - \hat{d}_{ij} \right)^2
\]  

4.2.2. Calculate the position and orientation of the rigid body. Let \( C \in \mathbb{R}^{3 \times 3} \) be the rotation matrix, \( r_{C1}, r_{C2} \) be the translation vectors, calculate the minimum of equation (10) by rotating
and translating the rigid body.

\[
\text{minimize } \quad J(C, r_{C1}, r_{C2}) = \sum_{l=1}^{m} k_l \|\hat{r}_i - r_{i,GPS}\|^2, \quad i \in \mathcal{V} \tag{10}
\]

subject to \( \hat{r}_i = C \cdot (r'_i - r_{C1}) + r_{C2}, \quad i \in \mathcal{V} \)

For the rigid body system, when the global minimum of potential \( J \) is reached, the system will be in a stable state. The necessary conditions for the system to enter a stable state are: the sum of all the external forces and the sum of all the external torque on the system are zero.

\[
\sum_{l=1}^{m} F_i = \sum_{l=1}^{m} -k_l(\hat{r}_i - r_{i,GPS}) = 0 \tag{11}
\]

\[
M = \sum_{l=1}^{m} (\hat{r}_i - r_{C2}) \times F_i = \sum_{l=1}^{m} (\hat{r}_i - r_{C2}) \times -k_l(\hat{r}_i - r_{i,GPS}) = 0 \tag{12}
\]

**Proposition 1** Set:

\[
\begin{align*}
    r_{C1} &= \sum_{l=1}^{m} k_l r'_i, \\
    r_{C2} &= \sum_{l=1}^{m} k_l r_{i,GPS} \\
    J &= \sum_{l=1}^{m} r_{i,GPS} \cdot r_{i,GPS}
\end{align*} \tag{13}
\]

Then for \( \forall C \in \mathbb{R}^{3 \times 3}, C \cdot C^T = I_3, \) equation (11) holds.

This proposition can be easily proved by substituting equation (13) into equation (11).

It’s difficult to find an analytic solution \( C \) which satisfies equation (12). Instead, a dynamic simulation method is used to calculate \( C \) iteratively. Denote \( u = [u_x, u_y, u_z]^T = \begin{bmatrix} M \\ 0 \end{bmatrix} \) as the rotation axis, \( \varphi \in \mathbb{R} \) as the rotation angle, \( \varphi = \varphi u \) as the rotation vector. If we fix \( u \), then \( J \) is a function about \( \varphi \). We rotate the rigid body around axis \( u \), and find \( \varphi \) to minimize \( J(\varphi) \), and update \( M \) according to current \( \hat{r}_i \) rotated by \( \varphi \) until convergence criterion is satisfied. Detailed description of the method is given as algorithm 2.

**Algorithm 2** Rotating Vector Algorithm

**Require:** \( r_{i,GPS}, i \in \mathcal{V} \): valid GPS coordinates;
\( \hat{r}_i = r'_i - r_{C1}, \quad i \in \mathcal{V} \): solution of equation (9) returned by algorithm 1 minus mass center;

**Ensure:** global optimum \( \hat{r}'_i 

1: Calculate \( M, u, \frac{dJ}{d\varphi} \)
2: **while** \(|\frac{dJ}{d\varphi}| > \text{Step Bound}\) **do**
3: Compute \( \Delta \varphi = -\frac{dJ}{d\varphi} \times \frac{d^2J}{d^2\varphi} / ||\Delta \varphi|| \)/Quadratic approximation of minimum
4: Set \( q = [\cos(\Delta \varphi), -\sin(\Delta \varphi) u, \ldots] \)/Calculate quaternion
5: \( \hat{r}_i = q^{-1} \hat{r}_i \otimes q \)/Use quaternion multiplication rules to update the coordinates
6: Calculate \( M, u, \frac{dJ}{d\varphi} \) using new \( \hat{r}_i \)
7: **end while**
8: Set \( \hat{r}_i = \hat{r}_i + r_{C2} \)

\[
dJ(\varphi) \quad \text{and} \quad \frac{d^2J(\varphi)}{d\varphi^2} \quad \text{can be analytically given as equation (14) and equation (15) respectively.}
\]

\[
-\frac{dJ}{d\varphi} = \sum_{l=1}^{m} F_i \cdot \hat{r}_i = \sum_{l=1}^{m} F_i \cdot \hat{r}_i \times (\hat{r}_i - r_{C2}) = \sum_{l=1}^{m} (\hat{r}_i - r_{C2}) \times F_i \cdot d\varphi \tag{14}
\]

\[
M \cdot d\varphi = M \cdot ud\varphi = ||M||d\varphi
\]
\[
\frac{d^2 J(\varphi)}{d\varphi^2} = - \frac{d\|M(\varphi)\|}{d\varphi} = -\mathbf{u} \cdot \frac{dM(\varphi)}{d\varphi} = -\mathbf{u} \cdot \sum_{l=1}^{m} \left( \frac{d\hat{r}_l}{d\varphi} \times \mathbf{F}_i + (\hat{r}_l - r_{C2}) \times \frac{d\mathbf{F}_i}{d\varphi} \right)
\]
\[
= -\mathbf{u} \cdot \sum_{l=1}^{m} \left( \frac{d\hat{r}_l}{d\varphi} \times -k_l (\hat{r}_l - r_{i_l,GPS}) + (\hat{r}_l - r_{C2}) \times -k_l \frac{d\hat{r}_l}{d\varphi} \right)
\]
\[
= \mathbf{u} \cdot \sum_{l=1}^{m} k_l (r_{i_l,GPS} - r_{C2}) \times \frac{d\hat{r}_l}{d\varphi} = \mathbf{u} \cdot \sum_{l=1}^{m} k_l (r_{i_l,GPS} - r_{C2}) \times \left[ \mathbf{u} \times (\hat{r}_l - r_{C2}) \right]
\]

5. Experiment

5.1. System Design

The proposed method has been experimentally tested using seven fixed-wing UAVs, where each UAV runs the estimation algorithm independently and five of them are equipped with RTK GPS to acquire accurate position as ground truth at rate of 5Hz. In the experiments, seven identical UAVs fly in a circle formation with 200 m radius for about 140 s, and the distances between each vehicle range from 20 m to 40 m. The system diagram is shown in figure 5. For each UAV a PX4 flight stack is used for altitude control and acceleration/velocity/position measurement, a UWB module named LinkTrack is used for ranging and inter-vehicle communication, flight information is transmitted to ground station by UBNT, and distributed position estimation and formation control algorithm as well as sensor measurements logging are done on RaspberryPi 3B+.

![Figure 5. UAV swarm system diagram for positioning error test.](image-url)
5.2. Experimental Results

Relative position estimation error $\hat{r}_{37} - r_{37}$ is given in figure 6, and the errors between other UAVs show similar magnitudes and characteristics. Errors are introduced by GPS when calculating the rigid body rotation $C$ and translation $r_{C1}$ and $r_{C2}$ using equation (10). As a comparison, RTK-GPS coordinates are used instead of GPS in equation (10) for estimation, the estimation error is plotted in figure 7. We can see that more accurate GPS coordinates will significantly reduce the norm of relative position error magnitude $\|\hat{r}_{ij} - r_{ij}\|$ while the estimated relative distance error $\bar{d}_{ij} - \|r_{ij}\|$ remain the same, as illustrate in figure 8.

![Figure 6. $\hat{r}_{37} - r_{37}$ using GPS coordinates for rotation.](image1)

![Figure 7. $\hat{r}_{37} - r_{37}$ using RTK-GPS coordinates for rotation.](image2)

![Figure 8. norm of relative position error $\|\hat{r}_{37} - r_{37}\|$ and estimated relative distance error $\|\hat{r}_{37}\| - \|r_{37}\|$.](image3)

![Figure 9. Estimated positions using different GPS coordinates and position ground truth at $T = 42$s.](image4)
However, the relative position estimation error $\hat{r}_{ij} - r_{ij}$ is only affected by the rotation matrix $C$ in equation (10), meanwhile, a common translation and rotation error for the whole rigid body in estimation is not vital when used in formation control, thus, we can estimate the error of this algorithm using $\hat{r}_{ij}$ estimated by RTK-GPS. A 3-D plot of the position estimation for the UAV swarm is shown in figure 9.

Comparison of the error $\|\hat{r}_{ij} - r_{ij}\|$ between the proposed method and the estimation using EKF described in section 4.1 is also given in table 1, it shows that the error RMS can reach 40cm level.

| Proposed method | 0.40m | 0.38m | 0.32m | 0.42m | 0.37m | 0.36m |
|-----------------|-------|-------|-------|-------|-------|-------|
| EKF             | 1.00m | 1.52m | 1.65m | 0.95m | 1.82m | 0.90m |

6. Conclusions
In this paper, a new method for relative localization of UAV swarm by fusing GPS, IMU inertial and UWB ranging measurements is proposed, the position estimation is obtained by calculating the global minimum of a non-convex function using specially designed iterative descent method. This method achieves a small distance estimation error (within 0.4 m) while maintaining low computational load. Due to the large GPS positioning error and inability of UWB module to provide bearing information, position offset and angle error are introduced into the estimation of rigid body structure, which is not vital for formation flight. In the future work, we intend to add new sensors which can measure accurate distance as well as angle and design dedicated algorithm to eliminate the positioning error introduced by angle measurements.

References
[1] Gezici S, Zhi Tian, Giannakis G B, Kobayashi H, Molisch A F, Poor H V and Sahinoglu Z 2005 IEEE Signal Processing Magazine 22 70–84
[2] Nilsson J, De Angelis A, Skog I, Carbone P and Händel P 2009 2009 17th European Signal Processing Conference pp 2161–2165
[3] Yao L, Wu Y A, Yao L and Liao Z Z 2017 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN) pp 1–8
[4] Mueller M W, Hamer M and D’Andrea R 2015 2015 IEEE International Conference on Robotics and Automation (ICRA) pp 1730–1736
[5] Olsson F, Ranta-Palokko J and Nygård J 2014 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN) pp 122–131
[6] Guo K, Han D and Xie L 2017 2017 13th IEEE International Conference on Control Automation (ICCA) pp 588–593
[7] FOY W H 1976 IEEE Transactions on Aerospace and Electronic Systems AES-12 187–194
[8] Chan Y T and Ho K C 1994 IEEE Transactions on Signal Processing 42 1905–1915
[9] Liu R, Yuen C, Do T, Jiao D, Liu X and Tan U 2017 2017 IEEE International Conference on Robotics and Automation (ICRA) pp 5623–5629
[10] Han Z, Guo K, Xie L and Lin Z 2019 IEEE Transactions on Automatic Control 64 20–34
[11] Timoshenko 2010 Theory of elasticity Engineering societies monographs (McGraw-Hill Education (India) Pvt Limited)