

Article

# Constrained ESKF for UAV Positioning in Indoor Corridor Environment Based on IMU and WiFi

Zhonghan Li <sup>1</sup> and Yongbo Zhang <sup>1,2,\*</sup> 

<sup>1</sup> School of Aeronautic Science and Engineering, Beihang University, Beijing 100191, China; Jason\_lzh@buaa.edu.cn

<sup>2</sup> Aircraft and Propulsion Laboratory, Ningbo Institute of Technology, Beihang University, Ningbo 315100, China

\* Correspondence: zhangyongbo@buaa.edu.cn

**Abstract:** The indoor autonomous navigation of unmanned aerial vehicles (UAVs) is the current research hotspot. Unlike the outdoor broad environment, the indoor environment is unknown and complicated. Global Navigation Satellite System (GNSS) signals are easily blocked and reflected because of complex indoor spatial features, which make it impossible to achieve positioning and navigation indoors relying on GNSS. This article proposes a set of indoor corridor environment positioning methods based on the integration of WiFi and IMU. The zone partition-based Weighted K Nearest Neighbors (WKNN) algorithm is used to achieve higher WiFi-based positioning accuracy. On the basis of the Error-State Kalman Filter (ESKF) algorithm, WiFi-based and IMU-based methods are fused together and realize higher positioning accuracy. The probability-based optimization method is used for further accuracy improvement. After data fusion, the positioning accuracy increased by 51.09% compared to the IMU-based algorithm and by 66.16% compared to the WiFi-based algorithm. After optimization, the positioning accuracy increased by 20.9% compared to the ESKF-based data fusion algorithm. All of the above results prove that methods based on WiFi and IMU (low-cost sensors) are very capable of obtaining high indoor positioning accuracy.



**Citation:** Li, Z.; Zhang, Y. Constrained ESKF for UAV Positioning in Indoor Corridor Environment Based on IMU and WiFi. *Sensors* **2022**, *22*, 391. <https://doi.org/10.3390/s22010391>

Academic Editor: Arturo Sanchez-Azofeifa

Received: 4 November 2021

Accepted: 31 December 2021

Published: 5 January 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** UAV; ESKF; WiFi; data fusion; indoor positioning

## 1. Introduction

An unmanned aerial vehicle (UAV) is a drone that integrates various sensors, flight control systems, data processing systems, power systems, and other modules, which can autonomously complete specified tasks without human intervention. Since the application of UAVs from their initial military purpose, such as Predator, Global Hawk, and Black Hornet, to later civilian and commercialization purposes, such as DJI and Zero-Zero technology, the drone market has entered a fierce development state. Rotor-wing UAVs, especially quadcopters, have good development prospects in indoor environments, due to their compact structure, easy hovering, and convenient side-flight. However, GNSS signals cannot be received indoors. Because of topological structures and spatial features, the indoor environment is complex, and signals are easily blocked and reflected. Therefore, quadcopters cannot rely on GNSS to realize positioning and navigation indoors. However, pseudolite-based methods can also solve GNSS-based method problems in an indoor environment [1,2]. Considering that current indoor environments can easily receive external source signals, such as WiFi and Bluetooth, deploying pseudolite transmitters also costs a lot, and method-based pseudolites may face near-far problems and time synchronization problems [1], this article mainly discusses other indoor positioning methods.

At present, the navigation and positioning methods used in indoor quadcopters mainly include the following two categories: vision or LiDAR-based methods and external source-based methods, in which the external source includes Bluetooth, WiFi, and UWB. Among them, vision or LiDAR-based methods are the current research hotspot [3–10], and the

sensors used in vision-based methods can be categorized into visual odometer [3,4], stereo camera [3,4], monocular camera [5–7], binocular camera [8], and optical flow sensor [9,10]. Reference [3] is mainly based on visual odometers and integrates a stereo camera and LiDAR to build an aircraft for indoor search and rescue. Based on the stereo camera and visual odometer, reference [4] additionally uses landmarks to achieve autonomous indoor navigation. A downward monocular camera and LiDAR are used to estimate the speed and attitude of a quadcopter in reference [5]. Reference [7] is based on the monocular camera and mainly improves the Oriented FAST and Rotated BRIEF (ORB) algorithm and obtains higher accuracy. The ArUco marks are used in [6] to realize indoor positioning control. The binocular camera is used to combine with IMU in [8], which realizes position and orientation control with respect to the specific target. Combining inertial sensors, optical flow, and airspeed data, reference [9] achieves indoor navigation and attitude control, while [10] realizing the tracking of a specific red line based on an optical flow sensor.

Although vision-based and LiDAR-based technology can provide high-precision indoor positioning results, in practical applications, high-precision LiDAR is expensive and obtains large amount of data, which require high calculation capabilities for drone equipment. Visual sensors are sensitive to light and dark, and processing image data also requires a large amount of calculation. Recently, with the development of mobile smart terminals, WiFi signals have become cheaper and easier to acquire. The positioning methods based on WiFi signals are less difficult to employ, and less is spent on construction. In addition, WiFi-based positioning methods are widely used in the indoor pedestrian positioning field. The common WiFi positioning approaches include fingerprinting [11–18], trilateration [19], and their combination [20]. Fingerprinting is more widely used, because location information can be provided without knowing access point locations and propagation models. Reference [11] proposes a Weighted K-Nearest Neighbor (WKNN) algorithm based on spatial feature division, which improves the smoothness of estimation results. Reference [12] uses the Local Feature-based Deep Long Short-Term Memory approach (LF-DLSTM) and achieved in local positioning. References [13,14] are specific applications for indoor intelligent vehicles and pedestrians, respectively; the errors of both are meter level. Series of studies on fingerprint optimization are in [15–18]. Reference [17] focuses on increasing the interference robustness of WiFi fingerprints, and the least square method is used to reduce location error in [18]. Reference [21] optimizes the AdaBoost algorithm to increase the accuracy of indoor/outdoor detection. The WiFi signature map and continuous perceptual model are established in [22] to realize indoor mobile robot navigation, and the localization error is a mean of 1.2 m. The series of references indicate that the WiFi-based positioning algorithm can meet the requirements of indoor positioning for mobile robots. In this article, a set of WiFi positioning methods is proposed which fuses IMU and WiFi by the constrained Error-State Kalman Filter (ESKF) method, and the accuracy of positioning becomes greatly improved.

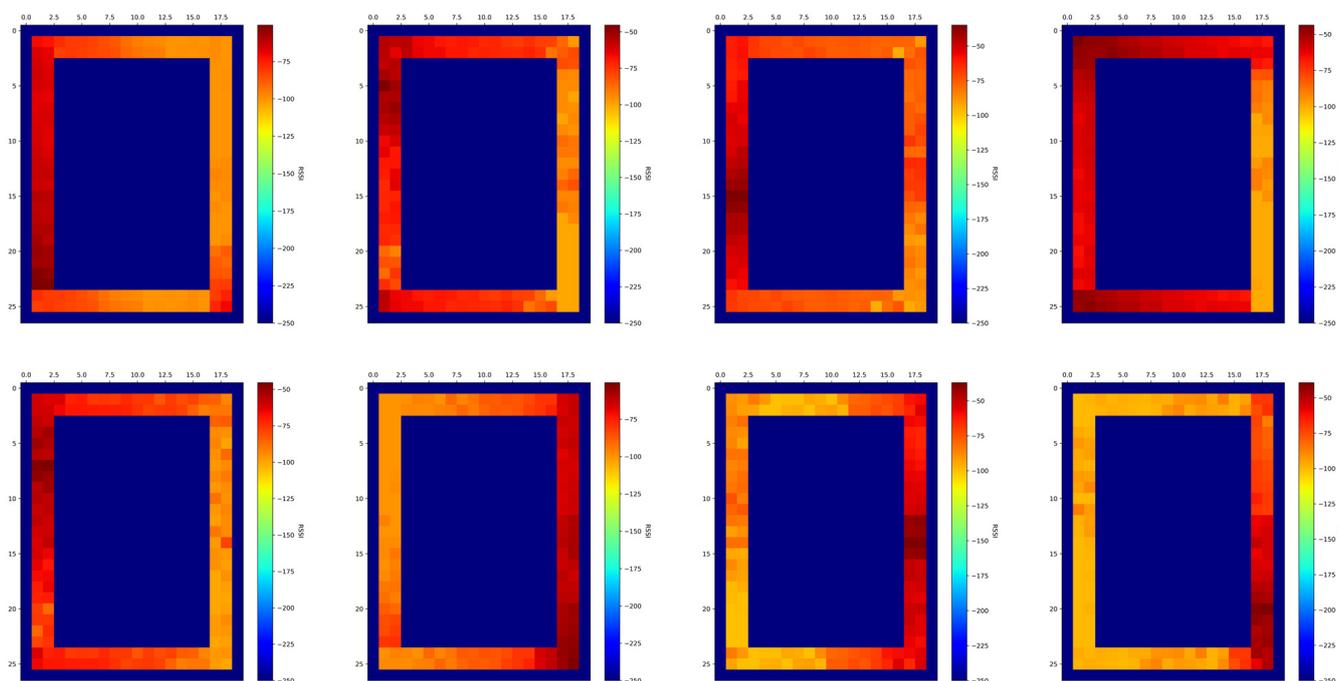
The rest of this article is organized as follows. In Section 2, the WiFi-based and IMU-based positioning methods are presented. In Section 3, the ESKF-based data fusion method and probability-based optimization method are presented. In Section 4, we provide experimental results of WiFi-based positioning, IMU-based positioning, ESKF-based data fusion method, and constrained optimization results. Finally, in Section 5, we draw conclusions.

## 2. Positioning Methods Based on IMU and WiFi

### 2.1. WiFi-Based Positioning Method

WiFi-based indoor positioning algorithms, especially the WKNN algorithm, have some inherent problems. On the one hand, before realizing positioning, data preprocessing is required, which needs to collect WiFi fingerprint data, and the number and the Media Access Control (MAC) address of reference points (RPs) are different. However, in order to facilitate the matching of fingerprint vectors and the calculation of the Euclidean distance, we need to extract the same access point (AP) information shared by all RPs into one database. During this process, part of the original data is lost. When the target area is small,

the loss of data does not have a significant impact on the location accuracy. However, when the target area is large or the spatial occlusion becomes serious, the APs in the database are not enough to support location, and there are even no fingerprint data in some places. On the other hand, the WKNN algorithm ignores the influence of the  $k$  points closest to the current position. When in the location process, it matches from the global area and only considers the constraints brought by the Euclidean distance. In addition, in the actual corridor environment, the signal strength in same position will be affected by different APs on the same or different floors. There are eight different AP signal strength heatmaps of one corridor area in Figure 1. The data were received in the true corridor environment. From the heatmaps we find that it is difficult to establish the WiFi propagation model to realize positioning. As a result, the algorithm we used is based on spatial feature partition and the WKNN algorithm and improved in order to achieve better location results.



**Figure 1.** WiFi signal strength heatmap of 8 different APs.

### 2.1.1. Zone Partition Algorithm by Spatial Feature

Research was conducted on the WKNN algorithm based on zone partition [11], and this method references the research. The WiFi positioning method based on spatial features can reduce the loss of original data and obtain higher accuracy. In the partitioning process, it is necessary to ensure that two adjacent areas share the same boundary points to achieve seamless partitioning. In the indoor environment, there are mainly four types in the corridor environment, which are straight line, corner, T-type, and crossroad as shown in Figure 2.

After establishing the WiFi fingerprint database for each partition, it is particularly important to accurately identify to which zone the UAV belongs. The zone discrimination algorithm used in this article contains two criteria, namely, Diagnostic Sequence Matching and the Euclidean Distance Comparison. Firstly, the diagnostic sequence needs to be generated, which is compared with the real-time sequence obtained by UAV, in order to obtain a preliminary discrimination result. The algorithm to obtain the diagnostic sequence is as follows.

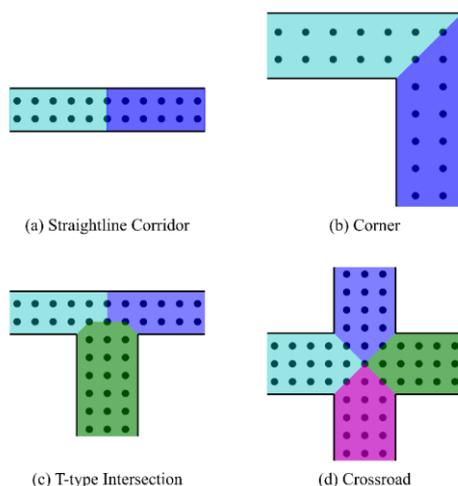


Figure 2. Some partition methods.

For the  $i$ -th zone, sum the received signal strength indication (RSSI) of the same MAC address in the database and sort results from large to small. The result is as follows:

$$[{}_i sum^1, {}_i sum^2, {}_i sum^3, \dots, {}_i sum^{n_i}]^T \tag{1}$$

where

$${}_i sum^j = \sum_{k=1}^m {}_i RSSI_j^k, (j = 1, 2, \dots, n_i) \tag{2}$$

$n_i$  is the number of the MAC address in the database of the  $i$ -th zone.  ${}_i sum^j$  means the sum of the RSSI of the  $j$ -th MAC address in the  $i$ -th zone, and  $m$  is the number of received RSSI of the  $j$ -th MAC address in the  $i$ -th zone.

Extract the former  $q$  elements of Formula (1), and their corresponding MAC address constitute the diagnostic sequence

$$[{}_i MAC^1, {}_i MAC^2, {}_i MAC^3, \dots, {}_i MAC^q]^T \tag{3}$$

However, at the junction of two adjacent zones, only relying on Diagnostic Sequence Matching may not be able to accurately determine the current zone of UAV. Thus, we need the Euclidean Distance Comparison to obtain precise results. The real-time RSSI vector obtained by UAV in real flight is

$$[rssi_1, rssi_2, rssi_3, \dots, rssi_p]^T \tag{4}$$

and the vectors in  $i$ -th zone and  $j$ -th zone are

$$[{}_i RSSI_{k_i}^1, {}_i RSSI_{k_i}^2, \dots, {}_i RSSI_{k_i}^{n_{k_i}}]^T, [{}_j RSSI_{k_j}^1, {}_j RSSI_{k_j}^2, \dots, {}_j RSSI_{k_j}^{n_{k_j}}]^T \tag{5}$$

The Euclidean distance between Formulae (4) and (5) should be calculated one by one, and the distance set can be obtained as

$$\left\{ \begin{matrix} L_i^1 & L_j^1 \\ L_i^2 & L_j^2 \\ \vdots & \vdots \\ L_i^{m_i} & L_j^{m_j} \end{matrix} \right\} \tag{6}$$

Then calculate the average value of the Euclidean distance, and the zone with the smaller value is the target.

### 2.1.2. WKNN Algorithm Based on Zone Partition

In the zone, the WKNN algorithm is used to realize UAV positioning. The algorithm contains two stages: the offline stage and online stage. In the offline stage, the WiFi fingerprint database is established; while in the online stage, the UAV acquires the RSSI of the current position and calculates the Euclidean distance of the real-time RSSI and database one. The position with the minimum distance is the target.

The data format in the database is

$$\left( \begin{array}{cc|cccc} x_1 & y_1 & RSSI_1^1 & RSSI_1^2 & \cdots & RSSI_1^m \\ x_2 & y_2 & RSSI_2^1 & RSSI_2^2 & \cdots & RSSI_2^m \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n & y_n & RSSI_n^1 & RSSI_n^2 & \cdots & RSSI_n^m \end{array} \right) \quad (7)$$

where  $(x_i, y_i)$  is the position coordinates of the  $i$ -th RP.  $RSSI_i^j$  is the received WiFi signal strength indication of the  $j$ -th AP of the  $i$ -th RP.  $n$  is the number of RPs, and  $m$  is the number of APs.

During the movement, the RSSI of the current position, which is obtained in real time, is

$$T = (rssi_1, rssi_2, rssi_3, \cdots rssi_m) \quad (8)$$

The Euclidean distance is used to characterize the distance between two vectors, which are the real-time measured RSSI and that in the database, that is

$$d_i = \|T_i - R_i^j\| = \sqrt{\sum_{j=1}^m |T_j - R_i^j|^2} \quad (9)$$

where  $d_i$  is the distance between two vectors.  $T_j$  is the signal strength vector of the  $j$ -th AP and is measured in realtime, and  $R_i^j$  is the signal strength vector of the  $j$ -th AP in  $i$ -th RP in database.

Then utilize the WKNN algorithm, rank  $d_i$  from small to large, select the smallest  $k$  points as the reference points for position estimation, and record it as the reference point set  $D$ , as shown in Formula (10).

$$D = \{ (x_i, y_i, R_i) \mid i = 1, 2, 3 \cdots K \} \quad (10)$$

The reciprocal of the Euclidean distance  $d_i$  determines the weight, and the current position  $(\hat{x}, \hat{y})$  in zone is located by Formula (11).

$$(\hat{x}, \hat{y}) = \frac{\sum_{i=1}^K \frac{(x_i, y_i)}{D_i}}{\sum_{i=1}^K \frac{1}{D_i}} \quad (11)$$

### 2.2. IMU-Based Positioning Algorithm

The inertial navigation system (INS), which is based on IMU sensors, measures the acceleration of vehicles and performs integration operations to obtain the current speed and position of the UAV. The INS equipment is installed in the vehicle, does not rely on external information or radiate energy to the outside world and is not easily interfered. However, due to the double integration operations, the system has obvious accumulated errors.

The basis of inertial navigation is a high-precision measuring element, generally the accelerometer and gyroscope, and the accelerometer measurement model can be expressed as

$$a_m^b = a^b + b_a + n_a \quad (12)$$

where  $a^b \in \mathbb{R}^3$  is the true value of the specific force in the body coordinate system.  $b_a$  is the drift error, and  $n_a$  is the accelerometer noise vector, which is regarded as Gaussian white noise. Further, the drift error  $b_a$  can be modeled as

$$\dot{b}_a = n_{b_a} \quad (13)$$

where  $n_{b_a}$  is regarded as Gaussian white noise.

The gyroscope is a sensor based on Coriolis force, and its measurement model can be expressed as

$$w_m^b = w^b + b_g + n_g \quad (14)$$

where  $w^b \in \mathbb{R}^3$  is the true value of angular velocity in the body coordinate system.  $b_g$  is the drift error, and  $n_g$  is the gyroscope noise vector, also regarded as Gaussian white noise. Further, the drift error can be modeled as

$$\dot{b}_g = n_{b_g} \quad (15)$$

where  $n_{b_g}$  is regarded as Gaussian white noise.

The drone position vector is composed of plane coordinates and height. In this article, we controlled the drone height as a constant, focusing on the 2D position, so the UAV position vector  $p^e$  is simplified as

$$p^e \triangleq [p_{x_e}, p_{y_e}]^T \in \mathbb{R}^3 \quad (16)$$

combined with the accelerometer measurement model, considering the integral relationship between position and velocity, the INS estimation formula can be expressed as

$$\begin{cases} \dot{p}^e = v^e \\ \dot{v}^e = R(a_m^b - b_a - n_a) + ge_3 \\ \dot{b}_a = n_{b_a} \end{cases} \quad (17)$$

where  $b_a$  is accelerometer drift error, and  $R \triangleq R_b^e$  is the rotation matrix characterizing body coordinate to global coordinate, which can be calculated as

$$R_b^e = \begin{bmatrix} c\theta c\psi & c\psi s\theta \sin \varphi - s\psi c\varphi & c\psi s\theta c\varphi + s\psi s\varphi \\ c\theta s\psi & s\psi s\theta \sin \varphi + c\psi c\varphi & s\psi s\theta c\varphi - c\psi s\varphi \\ -s\theta & s\varphi c\theta & c\varphi c\theta \end{bmatrix} \quad (18)$$

where  $c$  is cos,  $s$  is sin.  $\theta$  is pitch angle.  $\varphi$  is roll angle, and  $\psi$  is yaw angle.

### 3. Data Fusion Method and Optimization

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn.

#### 3.1. ESKF Algorithm for Combined Positioning

The ESKF method, which used an error-state representation, has several benefits compared to a nominal state representation [21]. First of all, the orientation error-state is represented as a 3D vector, which makes the state parameters equal to the degrees of freedom. Secondly, the value of error-state parameters is always around zero, which makes

the computation of the Jacobians easy and fast. Finally, singularities, gimbal lock issues, or similar problems can be avoided because of the small value of the orientation error-state.

### 3.1.1. System States

In the ESKF method, there are three states to describe the system, which are true-state  $x_t$ , nominal state  $x$ , and error-state  $\delta x$  and satisfy

$$x_t = x \oplus \delta x \quad (19)$$

here  $\oplus$  indicates a generic composition. In all three states, the state vector contains

- $p$ : position of UAV in global frame;
- $v$ : velocity of UAV in global frame;
- $q$ : quaternion of UAV;
- $R$ : rotation matrix of UAV;
- $a_b$ : accelerometer bias;
- $w_b$ : gyroscope bias.

For instance, the variable  $p$  in the true, nominal, and error-states should be expressed as  $p_t, p, \delta p$ . In addition, note that in the error-state, the angular error  $\delta\theta$  is used to describe the rotation error-state, such as quaternions or rotation matrix is used to describe rotation. The relationships between them are  $\delta q = e^{\delta\theta/2}, \delta R = e^{[\delta\theta]_\times}$ .

### 3.1.2. System Kinematics Models

The system true-state kinematics model is as follows.

$$\dot{p}_t = v_t \quad (20)$$

$$\dot{v}_t = a_t \quad (21)$$

$$\dot{q}_t = \frac{1}{2} q_t \otimes w_t \quad (22)$$

$$\dot{a}_{bt} = a_w \quad (23)$$

$$\dot{w}_{bt} = w_w \quad (24)$$

where the noisy accelerometer's reading in the body frame makes up the true acceleration  $a_t$ . Similarly, the true angular rate  $w_t$  is obtained from a gyroscope with noise in the body frame. So they satisfy

$$a_t = R_t(a_m - a_{bt} - a_n) + g_t \quad (25)$$

$$w_t = w_m - w_{bt} - w_n \quad (26)$$

where  $a_n, w_n$  denote the accelerometer and gyroscope noise, and  $a_w, w_w$  are the Gaussian random walk noise of biases.

Similar to the true-state kinematics model, the nominal-state kinematics model is without noises or perturbations:

$$\dot{p} = v \quad (27)$$

$$\dot{v} = R(a_m - a_b) + g \quad (28)$$

$$\dot{q} = \frac{1}{2} q \otimes (w_m - w_b) \quad (29)$$

$$\dot{a}_b = 0 \quad (30)$$

$$\dot{w}_b = 0 \quad (31)$$

The error-state kinematics model is derived from the nominal-state kinematics model,

$$\delta \dot{p} = \delta v \quad (32)$$

$$\delta \dot{v} = -R[a_m - a_b]_{\times} \delta \theta - R \delta a_b + \delta g - R a_n \tag{33}$$

$$\delta \dot{\theta} = -[w_m - w_b]_{\times} \delta \theta - \delta w_b - w_n \tag{34}$$

$$\delta \dot{a}_b = a_w \tag{35}$$

$$\delta \dot{w}_b = w_w \tag{36}$$

where the accelerometer noise is assumed to be white, uncorrelated, and isotropic. However, the assumption cannot be made in cases where XYZ accelerometers are not identical.

### 3.1.3. Propagation

The propagation step contains estimate error-state propagation and error covariance propagation. The estimate error-state is propagated through the error-state kinematics model in discrete time, which is

$$\delta p_{k+1} = \delta p_k + \delta v_k \Delta t \tag{37}$$

$$\delta v_{k+1} = \delta v_k + (-R_k[a_{m,k} - a_{b,k}]_{\times} \delta \theta_k - R_k \delta a_{b,k} + \delta g) \Delta t + v_i \tag{38}$$

$$\delta \theta_{k+1} = R_k^T \{ (w_{m,k} - w_{b,k}) \Delta t \} \delta \theta_k - \delta w_{b,k} \Delta t + \theta_i \tag{39}$$

$$\delta a_{b,k+1} = \delta a_{b,k} + a_i \tag{40}$$

$$\delta w_{b,k+1} = \delta w_{b,k} + w_i \tag{41}$$

where

$$v_i = \sigma_{a_n}^2 \Delta t^2 I \tag{42}$$

$$\theta_i = \sigma_{w_n}^2 \Delta t^2 I \tag{43}$$

$$a_i = \sigma_{a_w}^2 \Delta t I \tag{44}$$

$$w_i = \sigma_{w_w}^2 \Delta t I \tag{45}$$

Now the error-state system can be summarized as

$$\delta \hat{x}_{k+1} = f(x, \delta x, u_m, i) = F_x(x, u_m) \delta \hat{x} \tag{46}$$

$$F_x = \begin{bmatrix} I & I \Delta t & 0 & 0 & 0 & 0 \\ 0 & I & -R_k[a_{m,k} - a_{b,k}]_{\times} \Delta t & -R \Delta t & 0 & I \Delta t \\ 0 & 0 & R_k^T \{ (w_{m,k} - w_{b,k}) \Delta t \} & 0 & -I \Delta t & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \end{bmatrix} \tag{47}$$

The error covariance propagation is

$$\hat{P}_{k+1} = F_{x,k} P_k F_{x,k}^T + F_i Q_i F_i^T \tag{48}$$

$$F_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_i = \begin{bmatrix} v_i & 0 & 0 & 0 \\ 0 & \theta_i & 0 & 0 \\ 0 & 0 & a_i & 0 \\ 0 & 0 & 0 & w_i \end{bmatrix} \tag{49}$$

$F_x$  and  $F_i$  are the Jacobians of  $f()$  with respect to the error and perturbation vectors, and  $Q_i$  is the covariance matrix of the perturbation impulses.

### 3.1.4. Measurement Update

A WiFi position error measurement  $\delta y$  is used to update the error-state vector  $\delta x$ .

(1) Observation Model: with error-state representation, the observation model is simply linear:

$$\delta y = H\delta x = \begin{bmatrix} I_2 & 0 \\ 0 & 0 \end{bmatrix} \delta x \quad (50)$$

(2) Correction: after the WiFi measurements are computed, the filter corrections are implemented the same as the KF filter.

$$K = \hat{P}_{k+1} H_k^T (H \hat{P}_{k+1} H^T + V)^{-1} \quad (51)$$

$$\delta x_{k+1} = K(\delta y_{k+1} - H\delta \hat{x}_{k+1}) \quad (52)$$

$$P_{k+1} = (I - KH)\hat{P}_{k+1} \quad (53)$$

where  $V$  is the covariance of the measurements' white Gaussian noise, and the Jacobian matrix  $H$  is defined as

$$H = \left. \frac{\partial h}{\partial \delta x} \right|_x \quad (54)$$

### 3.1.5. Nominal State Update

After the error-state update, the nominal-state is updated with the observed error-state using the appropriate compositions,

$$x_{k+1} = x_k \oplus \delta \hat{x}_k \quad (55)$$

that is,

$$p_{k+1} = p_k + \delta \hat{p}_k \quad (56)$$

$$v_{k+1} = v_k + \delta \hat{v}_k \quad (57)$$

$$q_{k+1} = q_k \otimes q\{\delta \hat{\theta}_k\} \quad (58)$$

$$a_{b,k+1} = a_{b,k} + \delta \hat{a}_{b,k} \quad (59)$$

$$w_{b,k+1} = w_{b,k} + \delta \hat{w}_{b,k} \quad (60)$$

### 3.1.6. ESKF Reset

After error integrating into the nominal-state, the error-state variables need to be reset. This can be written as follows:

$$\delta x_{k+1} = \delta x_k \ominus \delta \hat{x}_k \quad (61)$$

where  $\ominus$  stands for the compositive inverse of  $\oplus$ , and the estimate error-state variables are set to 0. Finally, to make the ESKF update complete, the covariance of the error needs to be updated as follows:

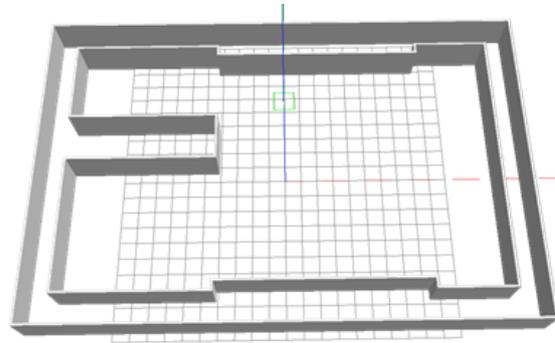
$$P_{k+1} = G_k P_k G_k^T \quad (62)$$

## 3.2. Optimization by Constrained ESKF

Computed by the ESKF algorithm, the IMU-based state and the WiFi-based state are fused together. However, the fusion data are not accurate enough for UAV indoor navigation. Due to specific features of the corridor environment, there are position constraints that can be used to improve positioning accuracy. The constraints can be divided into equality constraints and inequality constraints in the filtering field. For equality constraints, we can simplify the model variables utilizing the equations or implement state extensions to settle filtering problems. For inequality constraints, there are several methods mainly based on projection and probability. By projecting the filter results onto the constraint surface, the projection-based method can solve the inequality constraints problem. By assuming the filter results obey Gaussian distribution and cutting the distribution function at the boundary of constraints, the problem can also be solved.

### 3.2.1. Position Constraints

Unlike the cluttered indoor environment, the spatial feature of a corridor is simple, and specific physical constraints exist because of the corridor walls. The indoor corridor environment is shown in Figure 3 in the form of gazebo simulation model.

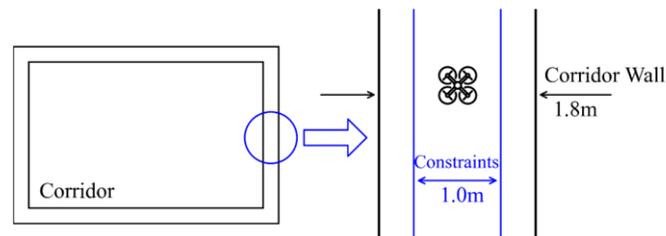


**Figure 3.** Gazebo simulation model of indoor corridor environment.

Only considering 2D position constraints, the corridor environment can be simplified as Figure 4, and the constraints are shown in the right of the figure. We implemented a 1 m constraint based on the center line in the 1.8 m corridor, and the constraints can be expressed by (46).

$$a \leq x \leq b \quad (63)$$

where vector  $x$  represents the UAV position in the corridor environment, and variables  $a, b$  represent the lower and upper bounds of the position constraints.



**Figure 4.** Simplified diagram of indoor corridor environment and its constraints.

### 3.2.2. Optimization Algorithm Based on Probability

With the physical constraints in indoor corridor environment, we assumed that at time interval  $k$  we can obtain  $s$  state constraints,

$$a_{ki} \leq \Phi_{ki}^T x_k \leq b_{ki} \quad i = 1, 2, \dots, s \quad (64)$$

and this is a bilateral constraint of the state linear function  $\Phi_{ki}^T x_k$ . Then, based on the filter's state estimation  $\hat{x}_k$  and its covariance  $P_k$ , we performed a transformation as follows:

$$z_{ki} = \rho W^{-1/2} T^T (x_k - \tilde{x}_{ki}) \quad (65)$$

where  $\rho$  is a  $n \times n$  orthogonal matrix obtained by Gram–Schmidt Orthogonalization. Matrices  $T, W$  are derived from the Jordanian Specification Decomposition of  $\tilde{P}_{ki}$ , and  $\tilde{x}_{ki}$  represents the state estimation that satisfies the first  $i$  constraints, and  $\tilde{P}_{ki}$  is the covariance of  $\tilde{x}_{ki}$ .

So the upper bound can transform into

$$\begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} z_{ki} \leq \frac{b_{ki} - \Phi_{ki}^T \tilde{x}_{ki}}{(\Phi_{ki}^T \tilde{P}_{ki} \Phi_{ki})^{1/2}} \leq d_{ki} \quad (66)$$

similarly, the lower bound can transform into

$$[ 1 \ 0 \ \dots \ 0 ] z_{ki} \geq \frac{a_{ki} - \Phi_{ki}^T \tilde{x}_{ki}}{(\Phi_{ki}^T \tilde{P}_{ki} \Phi_{ki})^{1/2}} \geq c_{ki} \tag{67}$$

then we obtain a normalized scalar constraint,

$$c_{ki} \leq [ 1 \ 0 \ \dots \ 0 ] z_{ki} \leq d_{ki} \tag{68}$$

Remove the part beyond the constraint in the Gaussian probability function and calculate the rest probability function and transform the cut-off probability into region area of 1, then we obtain

$$pdf(\xi) = \begin{cases} \alpha \exp(-\xi^2/2), & \xi \in [c_{ki}, d_{ki}] \\ 0, & \text{others} \end{cases} \tag{69}$$

where

$$\alpha = \frac{\sqrt{2}}{\sqrt{\pi}[\text{erf}(d_{ki}/2) - \text{erf}(c_{ki}/2)]} \tag{70}$$

and  $\text{erf}(\cdot)$  is the error function

$$\text{erf}(t) = \frac{2}{\sqrt{\pi}} \int_0^t \exp(-\gamma^2) d\gamma \tag{71}$$

After adding the constraint, the mean and variance of the state estimate are

$$\tilde{z}_{k,i+1} = [ \mu \ 0 \ \dots \ 0 ]^T \tag{72}$$

$$\text{Cov}(\tilde{z}_{k,i+1}) = (\sigma^2, 1, \dots, 1) \tag{73}$$

where

$$\mu = \alpha[\exp(-c_{ki}^2/2) - \exp(-d_{ki}^2/2)] \tag{74}$$

$$\sigma^2 = \alpha[\exp(-c_{ki}^2/2)(c_{ki} - 2\mu) - \exp(-d_{ki}^2/2)(d_{ki} - 2\mu)] + \mu^2 + 1 \tag{75}$$

Then perform an inverse transformation. We can obtain the state estimation and co-variance that meets the first constraint. Repeat  $i - 1$  times, and we obtain a state estimation. Figure 5 shows the schematic diagram of constrained algorithm by one real positioning result in experiments.

$$\tilde{x}_{k,i+1} = TW^{1/2} \rho^T \tilde{z}_{k,i+1} + \tilde{x}_{ki} \tag{76}$$

$$\tilde{P}_{k,i+1} = TW^{1/2} \rho^T \text{Cov}(\tilde{z}_{k,i+1}) \rho W^{1/2} T^T \tag{77}$$

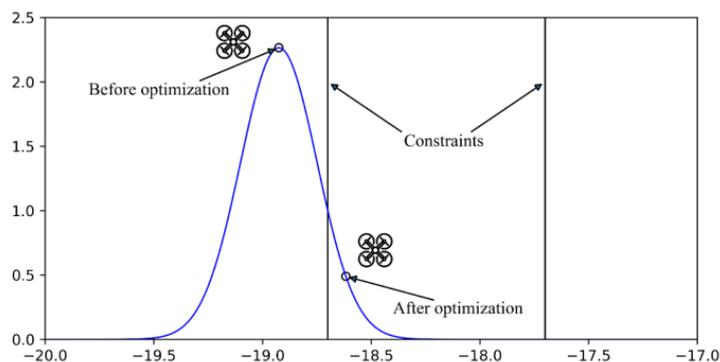


Figure 5. Schematic diagram of constrained algorithm based on probability.

#### 4. Experiment Results and Discussion

The indoor aircraft platform is independently designed to conduct a series of experiments, which are the WiFi-based and IMU-based positioning methods, the ESKF-based combined positioning method, and the constrained ESKF method for optimization. In this chapter, after introducing the platform and testing environment, we show the detailed results of all the above experiments.

##### 4.1. UAV Platform and Testing Environment

Figure 6 is the quadcopter we designed for the indoor corridor environment. It includes a Pixhawk flight controller, an onboard computer Raspberry Pi 3B+, and an optical flow sensor. The stability of the basic flight of the quadcopter is guaranteed by a flight controller. DroneKit project was used to control the copter's movement. In addition, note that the optical flow only participated in maintaining fixed-point flight, not in positioning and navigation. Before real flight, we carried out a series of experiments to guarantee feasibility. The simulation model we used is SITL, whose biggest advantage is that the simulated sensor data can be obtained directly from the existing flight dynamics model library. And the simulation experiment platform includes SITL, MAVProxy, and Mission Planner.



Figure 6. Indoor aircraft platform.

The simulated experiment site is the same as the site where the fingerprint database was established before as shown in Figure 7. The square corridor is roughly a  $20\text{ m} \times 30\text{ m}$  rectangle with a corridor width of 1.8 m.

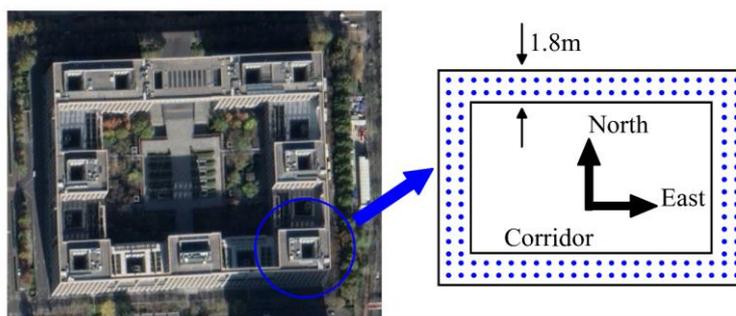


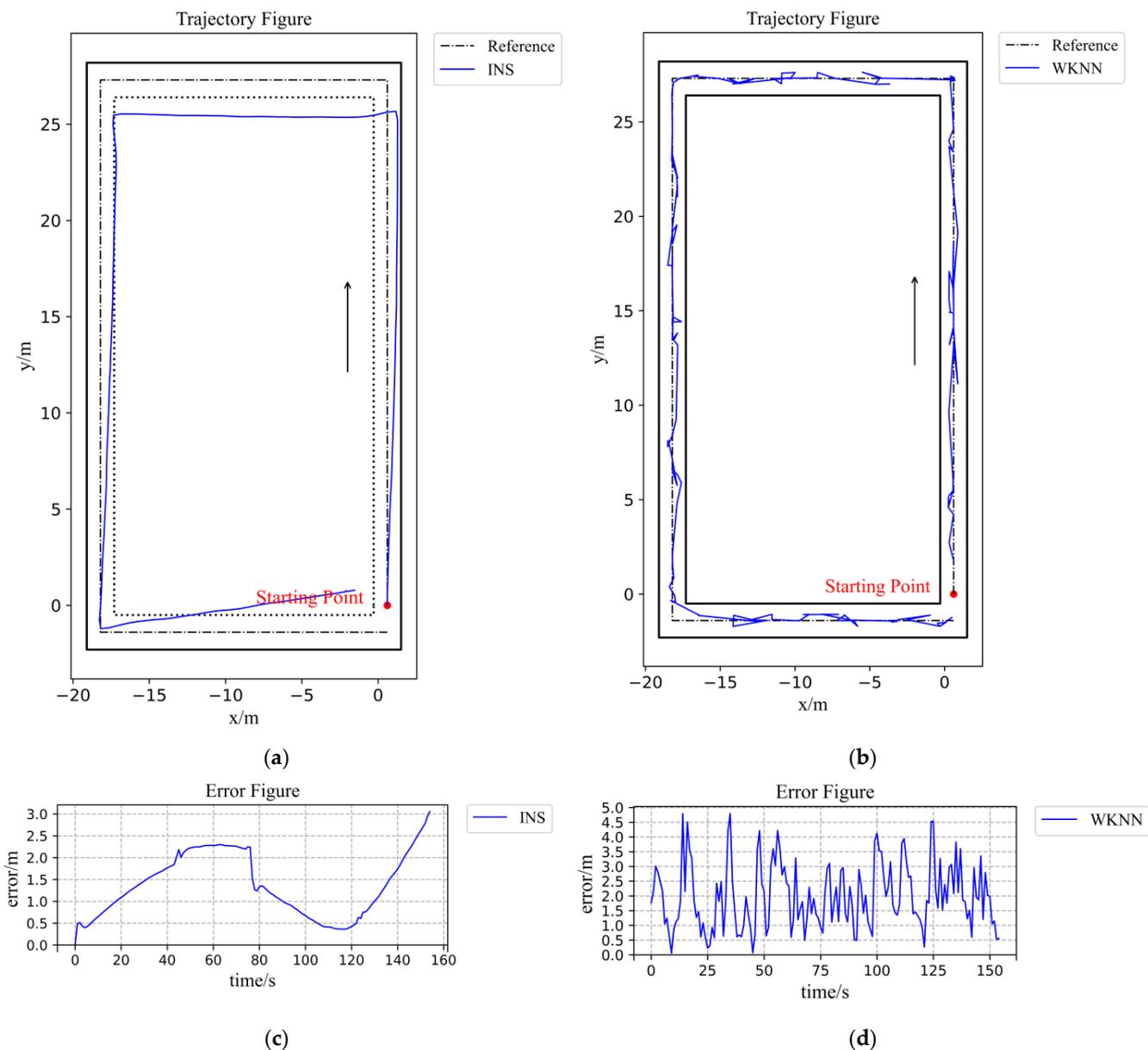
Figure 7. Experiment sites and its simplified diagram.

##### 4.2. WiFi-Based and IMU-Based Positioning Experiment Results and Discussion

###### 4.2.1. WiFi-Based Positioning Experiment Results and Discussion

The square grid was used as a reference to build the fingerprint database as shown in Figure 7. The reference points are separated by 1.2 m, and 158 reference points were arranged in the corridor. Considering that different receiving directions may cause interference to the WiFi signal strength, we took 10 measurements in each of the four directions at each reference point and then took the average of the measurement values as the signal strength and stored it in the fingerprint database.

The WiFi positioning experiment results are shown in Figure 8 and Table 1. In the trajectory figure, the black line is the corridor wall, the dotted line is the reference trajectory, and the blue line is zone partition-based WKNN trajectory. The positioning trajectory does not hit the wall, and the smoothness is very poor. The data points are moving together and jumping back and forth, which means in the further navigation part that the stability of the actual UAV flight cannot be satisfied. With a maximum error of 4.79 m, removing a minimum error of 0.01 m, a zone partition-based WKNN algorithm can reach an average error of 1.98 m.



**Figure 8.** Figures of IMU-based and WiFi-based experiments results: (a) Figure of IMU-based positioning trajectory; (b) Figure of WiFi-based positioning trajectory. (c) Error figure of IMU-based positioning trajectory; (d) Error figure of WiFi-based positioning trajectory.

**Table 1.** Errors of WiFi positioning algorithm.

Algorithm	Min/m	Max/m	Mean/m
Zone partition-based WKNN algorithm	0.01	4.79	1.98

### 4.2.2. IMU-Based Positioning Experiment Results and Discussion

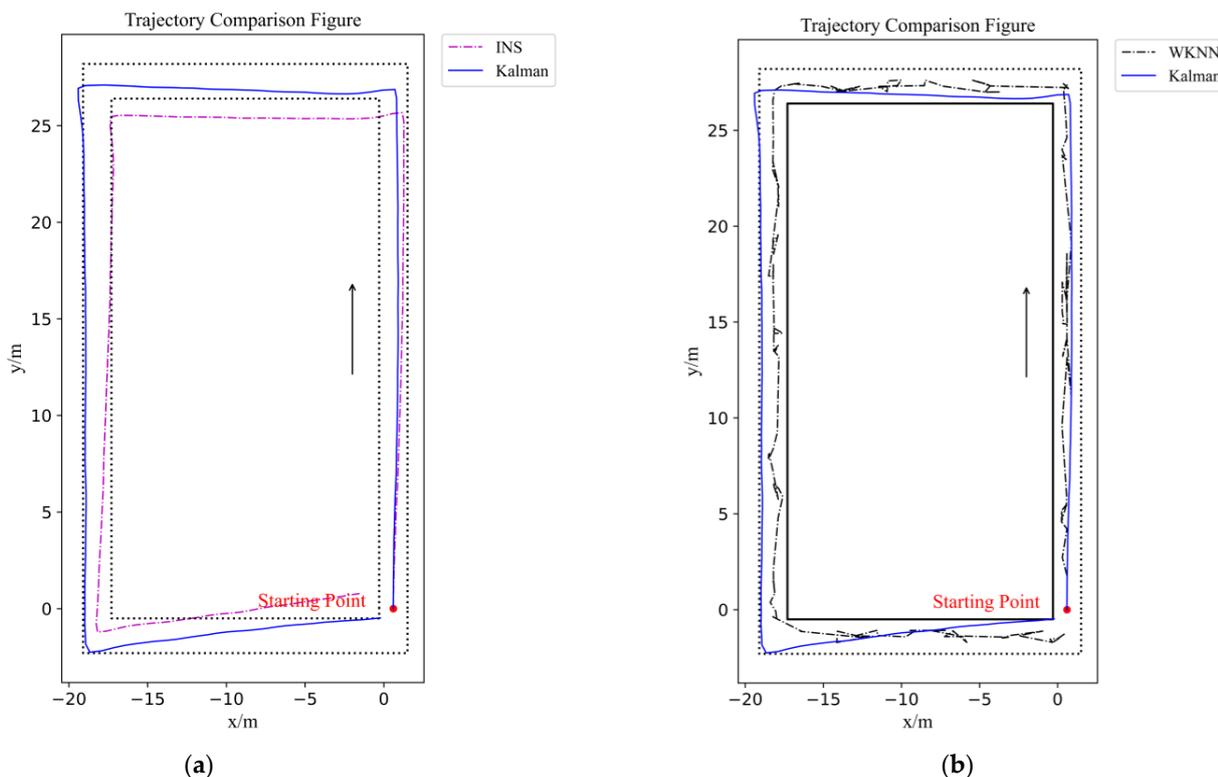
The IMU-based positioning experiment results are shown in Figure 8 and Table 2. In Figure 8a, the dotted line is the reference trajectory, and the blue line is the IMU-based trajectory, which exhibits an obvious cumulative error due to the two integrations. Only by realizing the IMU-based positioning, the experiment result deviates from the reference trajectory too much. Although it is smoother than the WiFi-based positioning method, the UAV cannot finish the whole flight safely. With a maximum error of 3.06 m, removing a minimum error of 0.001 m, the IMU-based positioning algorithm can reach a mean error of 1.37 m.

**Table 2.** Errors of INS positioning algorithm.

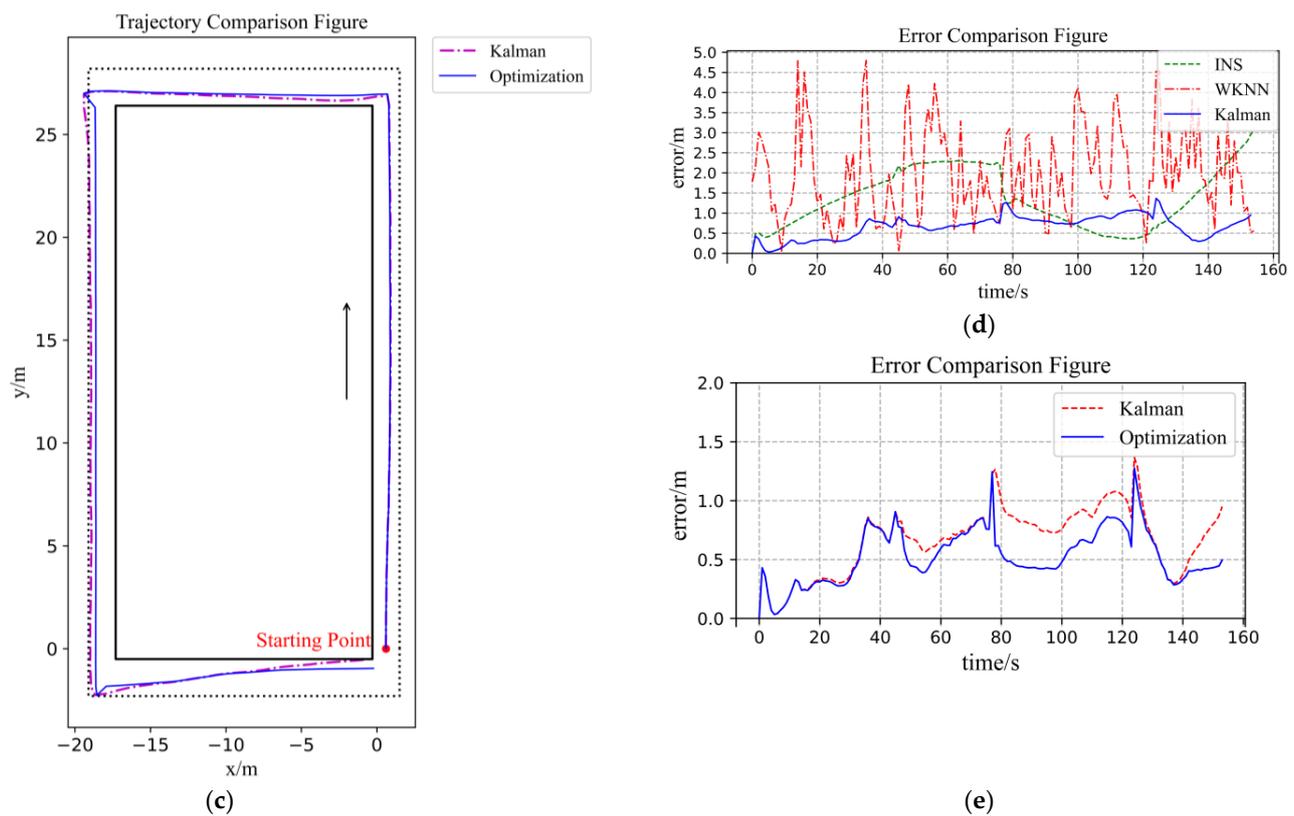
Algorithm	Min/m	Max/m	Mean/m
IMU-based positioning algorithm	0.001	3.06	1.37

### 4.3. ESKF for Combined Positioning Method Experiment Result and Discussion

The ESKF result is shown in Figure 9 and Table 3, and the comparisons among all three algorithms are shown in Figure 9d. After the ESKF data fusion, the average value of positioning deviation decreased to 0.67 m, and the positioning accuracy improved by 51.09% compared to the IMU-based algorithm and 66.16% compared to the WiFi-based algorithm. The maximum error also decreased significantly; it decreased by 55.23% compared to the IMU-based algorithm and by 71.4% compared to the WiFi-based algorithm. The positioning accuracy is evidently enhanced through the ESKF data fusion algorithm. In terms of the positioning trajectory, the ESKF-based data fusion result combines specific features of the other two algorithms, which made the combined trajectory smoother and closer to the reference line. However, because there was an obvious drift with the IMU sensors, the combined trajectory partly overlapped with the corridor walls, which will bring risk in further indoor navigation control.



**Figure 9.** Cont.



**Figure 9.** Figures of ESKF-based and constrained ESKF experiment results and comparison figures: (a) Figure of IMU-based and ESKF positioning trajectory comparison; (b) Figure of WiFi-based and ESKF positioning trajectory comparison; (c) Figure of ESKF and constrained ESKF positioning trajectory comparison; (d) Error figure of three positioning methods; (e) Error figure of data fusion method and optimization positioning method.

**Table 3.** Comparisons of errors of three algorithms.

Algorithm	Min/m	Max/m	Mean/m
WiFi-based positioning algorithm	0.01	4.79	1.98
IMU-based positioning algorithm	0.001	3.06	1.37
ESKF-based data fusion algorithm	0.001	1.37	0.67

#### 4.4. Constrained ESKF Positioning Experiment Result and Discussion

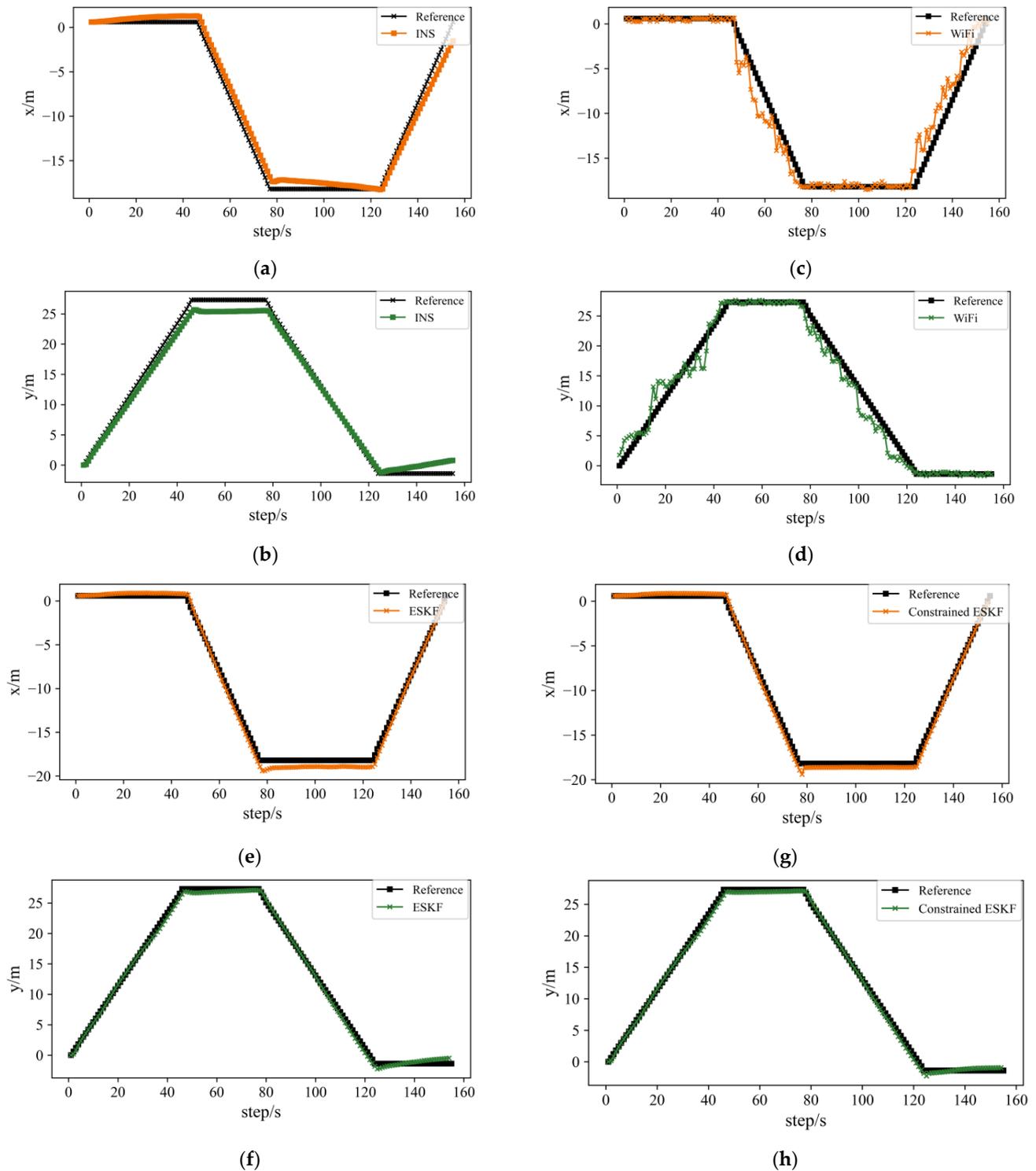
The constrained ESKF optimization result is shown in Figure 9. In addition, the comparison of the two algorithm errors is shown in Table 4. The optimization operation effectively increased the accuracy of the ESKF-based combined positioning method. The maximum error decreased by 6.57%, and the positioning accuracy improved by 20.9%. With a mean error of 0.53 m and a maximum error of 1.28 m, the UAV positioning accuracy obtained great protection.

**Table 4.** Comparisons of errors of three algorithms.

Algorithm	Min/m	Max/m	Mean/m
ESKF-based data fusion algorithm	0.001	1.37	0.67
Constrained ESKF optimization algorithm	0.001	1.28	0.53

Figure 10 is a summary of the set of indoor corridor environment positioning methods, which are the WiFi-based and IMU-based positioning methods, the ESKF-based data

fusion method, and the probability-based constrained method. From (a), (b) and (c), and (d) we can easily see the deviation caused by IMU sensors drift and fluctuation in the WiFi-based positioning method. After the ESKF-based data fusion method, the accuracy improved significantly. After implementing the probability-based constrained method, the positioning results further improved.



**Figure 10.** Figures of all methods comparison with reference trajectory: (a,b) are comparison figures of IMU-based method; (c,d) are comparison figures of WiFi-based method; (e,f) are comparison figures of ESKF-based data fusion method; (g,h) are comparison figures of constrained ESKF method.

## 5. Conclusions

With the progress of urbanization, the area of buildings is becoming larger and larger, there are more and more giant buildings, and the market demand for indoor drones is also increasing. Due to its light weight, small size, and flexible motion, indoor drones can easily adapt to narrow spaces, and are of great significance for completing tasks, such as indoor reconnaissance, indoor rescue, and target pickup. At present, the research hotspots of indoor drone navigation mainly focus on visual technology and SLAM technology, and there are few studies on WiFi-based drone navigation and positioning methods. Based on the indoor positioning technology of pedestrians and combining the characteristics of the indoor environment, we proposed a positioning method based on the fusion of WiFi and IMU measurements and obtained good experimental results.

Firstly, we carry out a series of simulation experiments for the IMU-based and WiFi-based positioning methods and obtained an average error of 1.98 m for WiFi-based positioning and an average error of 1.37 m for IMU-based system. There are specific features among them. The IMU-based positioning method can obtain smoother results but deviates greatly from the reference trajectory. WiFi-based positioning can obtain closer results to the reference trajectory, but the data jump back and forth, which cannot guarantee the stability of further navigation control. The ESKF-based data fusion method improved the positioning accuracy by 51.09% compared with the IMU-based method and by 66.16% compared with the WiFi-based method. To further improve accuracy, the probability-based optimization method is realized, which improves the accuracy of positioning by 20.9%.

**Author Contributions:** Conceptualization, Z.L. and Y.Z.; methodology, Z.L.; software, Z.L.; validation, Z.L. and Y.Z.; formal analysis, Z.L.; investigation, Z.L.; resources, Y.Z.; data curation, Z.L.; writing—original draft preparation, Z.L.; writing—review and editing, Z.L. and Y.Z.; visualization, Z.L.; supervision, Y.Z.; project administration, Y.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was sponsored by the National Key Research and Development Program of China under Grant 2018YFF0216004. The authors fully appreciate the financial support.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Wan, X.; Zhan, X. The research of indoor navigation system using pseudolites. *Procedia Eng.* **2011**, *15*, 1446–1450. [[CrossRef](#)]
2. Wang, J. Pseudolite applications in positioning and navigation: Progress and problems. *Positioning* **2002**, *1*, 48–56. [[CrossRef](#)]
3. Tomic, T.; Schmid, K.; Lutz, P.; Domel, A.; Kassecker, M.; Mair, E.; Grix, I.; Ruess, F.; Suppa, M.; Burschka, D. Toward a Fully Autonomous UAV: Research Platform for Indoor and Outdoor Urban Search and Rescue. *IEEE Robot. Autom. Mag.* **2012**, *19*, 46–56. [[CrossRef](#)]
4. Valenti, F.; Giaquinto, D.; Musto, L.; Zinelli, A.; Bertozzi, M.; Broggi, A. Enabling Computer Vision-Based Autonomous Navigation for Unmanned Aerial Vehicles in Cluttered GPS-Denied Environments. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, Hawaii, USA, 4–7 November 2018; pp. 3886–3891.
5. Wang, F.; Cui, J.; Phang, S.K.; Chen, B.M.; Lee, T.H. A mono-camera and scanning laser range finder based UAV indoor navigation system. In Proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 28–31 May 2013; pp. 694–701.
6. Zhang, X.W.; Du, Y.S.; Chen, F.; Qin, L.L.; Ling, Q. Indoor Position Control of a Quadrotor UAV with Monocular Vision Feedback. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; pp. 9760–9765.
7. Zhang, Y.; Cai, Z.; Zhao, J.; You, Z.; Wang, Y. Feature-Based Monocular Real-Time Localization for UAVs in Indoor Environment. In *Proceedings of the 2017 Chinese Intelligent Automation Conference*; Springer: Singapore, 2018; pp. 357–366. [[CrossRef](#)]
8. Salazar, S.; Romero, H.; Gómez, J.; Lozano, R. Real-time stereo visual servoing control of an UAV having eight-rotors. In Proceedings of the 2009 6th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Mexico City, Mexico, 10–13 January 2009; pp. 1–11.

9. Gerhart, G.R.; Soloviev, A.; Gage, D.W.; Rutkowski, A.J.; Shoemaker, C.M. Fusion of inertial, optical flow, and airspeed measurements for UAV navigation in GPS-denied environments. In Proceedings of the Unmanned Systems Technology XI, Orlando, FL, USA, 13–17 April 2009.
10. Mora Granillo, O.D.; Zamudio Beltran, Z. Real-Time Drone (UAV) Trajectory Generation and Tracking by Optical Flow. In Proceedings of the 2018 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE), Cuernavaca, Mexico, 26–29 November 2018; pp. 38–43.
11. Yang, H.; Zhang, Y.; Huang, Y.; Fu, H.; Wang, Z. WKNN indoor location algorithm based on zone partition by spatial features and restriction of former location. *Pervasive Mob. Comput.* **2019**, *60*, 101085. [[CrossRef](#)]
12. Chen, Z.; Zou, H.; Yang, J.; Jiang, H.; Xie, L. WiFi Fingerprinting Indoor Localization Using Local Feature-Based Deep LSTM. *IEEE Syst. J.* **2020**, *14*, 3001–3010. [[CrossRef](#)]
13. Li, Y.; Zhang, P.; Niu, X.; Zhuang, Y.; Lan, H.; El-Sheimy, N. Real-time indoor navigation using smartphone sensors. In Proceedings of the 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Banff, AB, Canada, 13–16 October 2015; pp. 1–10.
14. Nguyen, D.-V.; Nashashibi, F.; Nguyen, T.-H.; Castelli, E. Indoor Intelligent Vehicle localization using WiFi received signal strength indicator. In Proceedings of the 2017 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM), Nagoya, Japan, 19–21 March 2017; pp. 33–36.
15. Jian, H.X.; Hao, W. WIFI Indoor Location Optimization Method Based on Position Fingerprint Algorithm. In Proceedings of the 2017 International Conference on Smart Grid and Electrical Automation (ICSGEA), Changsha, China, 27–28 May 2017; pp. 585–588.
16. Joseph, R.; Sasi, S.B. Indoor Positioning Using WiFi Fingerprint. In Proceedings of the 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), Kottayam, India, 21–22 December 2018; pp. 1–3.
17. Lemic, F.; Behboodi, A.; Handziski, V.; Wolisz, A. Increasing Interference Robustness of WiFi Fingerprinting by Leveraging Spectrum Information. In Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, Liverpool, UK, 26–28 October 2015; pp. 1200–1208.
18. Ting-Ting, X.; Xing-Yu, L.; Ke, H.; Min, Y. Study of fingerprint location algorithm based on WiFi technology for indoor localization. In Proceedings of the 10th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2014), Beijing, China, 26–28 September 2014; pp. 604–608.
19. Schatzberg, U.; Banin, L.; Amizur, Y. Enhanced WiFi ToF indoor positioning system with MEMS-based INS and pedometer information. In Proceedings of the 2014 IEEE/ION Position, Location and Navigation Symposium—PLANS 2014, Monterey, CA, USA, 5–8 May 2014; pp. 185–192.
20. Naseri, H.; Homaeinezhad, M.R. Improving measurement quality of a MEMS-based gyro-free inertial navigation system. *Sens. Actuators A Phys.* **2014**, *207*, 10–19. [[CrossRef](#)]
21. Huang, H.; Zeng, Q.; Chen, R.; Meng, Q.; Wang, J.; Zeng, S. Seamless Navigation Methodology optimized for Indoor/Outdoor Detection Based on WIFI. In Proceedings of the 2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS), Wuhan, China, 22–23 March 2018; pp. 1–7.
22. Biswas, J.; Veloso, M. WiFi localization and navigation for autonomous indoor mobile robots. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 4379–4384.