ABSTRACT Real-time and stable positioning data is essential for the UAV to perform various tasks. The traditional multi-sensor data fusion algorithm needs to know the measurement noise of sensor data, and even if there are corresponding adaptive methods to estimate the noise, most methods cannot deal with time-varying noise. In addition, traditional fusion algorithms usually are complicated, causing a large amount of calculation. In this paper, a multi-sliding window classification adaptive unscented Kalman filter (MWCAUKF) method with timestamp sort updating was proposed, which can improve the accuracy and stability of positioning. This method consists of three phases. First, according to the timestamp of sensor data, the multi-sensor data are added with fusion filtering in order. Then it estimates the measurement noise of multiple sensors through multiple sliding Windows. Finally, the sensor data classification method is adopted to deal with the filter instability caused by time-varying noise. Both theoretical analysis and experimental results show that this method has a low computational cost, high accuracy, and good stability.

INDEX TERMS Multi-sensor fusion, unmanned aerial vehicle, positioning, adaptive Kalman filtering.

I. INTRODUCTION Positioning is the core of UAV, which plays an important role in UAV navigation [1], autonomous control [2] and the execution of various tasks. With the rapid development of science and technology, the accuracy and reliability of positioning have been put forward more and more high requirements. It is difficult for a single sensor to meet the positioning needs of UAV, and multi-sensor fusion positioning is the current development trend of UAV positioning [3]. That is mainly because multiple information sources can be fully utilized to supplement each other, forming a positioning system with multiple redundancies, higher positioning accuracy, and stability [4].

At present, research on UAV location algorithm based on multi-sensor data fusion is mainly based on Kalman filter [5] data fusion. Among them, multi-sensor data fusion can be divided into centralized fusion Kalman filter [7], [9] and distributed fusion Kalman filter [8], [10] according to different fusion modes. Although the fusion accuracy of centralized fusion filter is relatively high, it has the disadvantages of large computation, complicated computation and high requirement on the central processor. The distributed fusion algorithm is different from the centralized fusion algorithm [11]. Instead, it does not feed observations from each sensor directly into the fusion center. The distributed fusion algorithm uses local sensor data to predict the motion state of the target first, and then sends the prediction data to the fusion center. Therefore, it has the advantages of small amount of transmitted data, high reliability, and high scalability. However, in the process of one-step prediction by distributed fusion method, data dimensions obtained by sensors are different due to the different performance of various sensors. Therefore, various data need to be adjusted in corresponding dimensions [12]. Moreover, the data frequency obtained by various sensors is different, so it is necessary to synchronize the time of various data in the fusion center, which can the optimal order of fusion be determined [13]. These processes before data fusion add many complex processing and computing processes to the process of multi-sensor data fusion.

For the conventional Kalman filter algorithm [14], [15], the model structure and noise statistical characteristics of dynamic systems are required to be known, but in practical applications, the models and noise statistical characteristics of many systems are unknown, resulting in the Kalman filter algorithm losing its optimality. It is estimated that the accuracy decreases or even diverges. Adaptive Kalman filtering can solve this problem. Common adaptive filtering algorithms include Sage-Husa adaptive Kalman filtering algorithm [16], strong tracking filtering algorithm [17], and fuzzy...
adaptive algorithm [18]. Sage-Husa adaptive Kalman filter has the characteristics of high accuracy and instability, while the strong tracking filter has low accuracy but has extremely strong tracking ability about state changes. Literature [19] combines the advantages and disadvantages of Sage-Husa adaptive filtering algorithm and strong tracking filtering algorithm. The fuzzy adaptive filtering algorithm has the problem of fuzzy parameter selection. Similarly, the fading memory filtering algorithm [28], [29] also has the problem of fading factor selection. The H∞ filtering [20] algorithm is very complex and computationally intensive, so it requires high system hardware.

Aiming at the above problems, this paper proposes a new multi-sensor fusion adaptive Kalman filter framework and applies it to the positioning system of UAV. The main contributions of this paper are as follows.

- Aiming at the problem that various sensor data needs time synchronization in data fusion, this paper proposes a filter update method based on timestamp sort, which effectively avoids the problem of multi-sensor time synchronization. At the same time, this method greatly reduces the calculation amount of the filtering process. With this method, sensor measurements can be easily fused together regardless of the sensor update rates.

- The proposed adaptive Kalman filtering method based on multi-sliding window noise estimation can estimate the measurement noise of multiple sensors simultaneously. Traditional adaptive filtering can only estimate a single sensor noise. Different from traditional adaptive filtering, this paper can estimate the measurement noise of multiple sensors in one filter at the same time. During the filtering process, the filtering parameters are dynamically adjusted to improve filtering accuracy.

- Aiming at the multi-sliding window noise estimation, the adaptive Kalman filter cannot effectively deal with the filter divergence caused by time-varying noise. This paper proposes a classification method for noise to improve the stability of filtering.

In the Section II of this article, the differences between the work in this paper and the current related work are introduced. Section III gives a brief introduction to the entire system. The system is described in detail in sections IV and V. Section VI analyzes the performance of the system. Section VII gives specific experimental results and experimental analysis. Conclusions and future work will be discussed in Section VIII.

II. RELATIVE WORK

There are a lot of work-related to UAV positioning in multi-sensor fusion. Some related work referred to in this paper is compared.

For multi-sensor data fusion, the time synchronization of the data directly affects the accuracy of the fusion. A method of sequence measurement updating is proposed in literature [22]. The updated time node in this method is as same as the IMU data measurement time node. Other sensors are used as auxiliary sensors to correct IMU data, and their update nodes must be on the IMU update nodes. This method requires time synchronization for all sensors. The specific method of time synchronization is given in literature [23]. But this article focuses on time sorting. In this paper, all sensor data are sorted by time. Data fusion is performed in the chronological order of data acquisition without the need for data synchronization. Therefore, the method in this paper transforms multi-sensor fusion into the fusion of single data and system state data.

The complicated filtering process leads to a huge increase in the amount of calculation. In literature [24], an improved Kalman filter is proposed, which reduces the calculation of the filtering process. Furthermore, the proposed subband Kalman filtering method in literature [25], which effectively reduces the computation of filtering. Besides, literature [26] proposed a weighted fusion recombination innovation strategy, which reduced the calculation amount under the condition of suppressing the influence of noise on the fusion result. Compared with the centralized fusion method and the distributed fusion method, the calculation of the distributed algorithm is much less. A comparison method of calculation is given in [27]. This article is based on the timestamp sorting update method. The measurement update of the data is the filtered fusion update. In this method, only one measurement data needs to be processed for one filtering update, which avoids processing all sensor data at one time. This processing method effectively reduces the amount of calculation.

Adaptive filtering is very effective for sensors with unknown measurement noise. The fading memory filtering method [28], [29] reduces the influence of outdated data on the current state data by introducing fading memory factor into the filtering. This approach brings the system model close to the current real model. However, how to choose the appropriate fading memory factor is a difficult problem faced by this method. This paper uses a sliding window-based noise statistics method to avoid the selection of fading factors, remove outdated data in time, and use the updated data to evaluate the entire filtering process. The data fusion method [30], [31] based on the adaptive filtering of fuzzy logic avoids the selection of fading memory factors, but it has problems such as low accuracy, easy dispersion, and difficulty in handling strong time-varying noise. Similar to the algorithm in this paper, a random weighted sliding window adaptive filtering method is proposed in the literature [6]. Because the selection of weights is relatively random, this algorithm is not applicable when dealing with strong variability noise. The strong tracking filtering algorithm proposed in [39] effectively addresses the problem of filtering divergence caused by time-varying noise. Based on the characteristics of the sensor data, this paper uses a classification method to adjust the measurement covariance and process covariance accordingly to solve the problem of strong time-varying noise.
data filter the invalid data in advance, which has a greater impact on improving the fault tolerance of the filter fusion system. For the sensors mentioned in this paper, the validity of analyzing the data of these sensors is as follows. For example, whether GPS data is valid can be determined by the number of satellites received by the GPS receiver. When the number of satellites received by GPS is greater than or equal to 4, we consider that the GPS data is reliable, otherwise, the data is unreliable. Similarly, whether the lidar data is valid is determined by the number of data points scanned. One frame of lidar data includes 180 data points. When the lidar scan data points are less than 30, the lidar data is considered to be unusable in the scanning matching algorithm [37]. In order to continuously output the positioning data, the validity of IMU data is not verified in this paper. Other sensor data can be verified separately according to the sensor features.

In this paper, the coordinate system of each sensor needs to be converted to a unified coordinate system [32]. This paper uses east-north-up (ENU) coordinates as the global coordinate system. Longitude, latitude, and altitude are based on the geodetic (LLA) coordinate system. The formula for converting the LLA coordinate system to the ENU coordinate system is given in the literature [40]. This article assumes that the attitude angle obtained by the IMU is data in the ENU coordinate system. The relationship between global acceleration and the acceleration obtained by the IMU is as follows:

\[ a_{enu} = R_B^I a_{IMU} \]  

where \( a_{enu} \) is the acceleration of the UAV in the ENU coordinate system, and \( a_{IMU} \) is the acceleration of the UAV in the IMU coordinate system. \( R_B^I \) is the rotation matrix from the IMU coordinate to the ENU coordinate system.

\[ R = \begin{bmatrix} c\psi s\phi & c\phi s\psi - c\psi s\phi & c\psi c\phi + s\psi s\phi \\ c\theta s\psi & s\phi s\psi + c\phi c\theta & c\phi s\psi - c\psi s\phi \\ -s\theta & c\theta c\phi & c\theta s\phi \end{bmatrix} \]  

where \( \psi, \theta, \) and \( \phi \) are the UAV’s ENU-frame yaw, pitch, and roll, respectively. \( c \) and \( s \) designate the cosine and sine functions. In this paper, the default lidar coordinate axis is parallel to the ENU coordinate axis, so the formula for transferring the polar coordinates \((r_i, \xi_i)\) of the lidar scanning data to the ENU coordinate system is as follows:

\[ Point_i = \begin{bmatrix} r_i \cos(\xi_i) \\ r_i \sin(\xi_i) \\ 0 \end{bmatrix} + P_0 \]  

where \( Point_i \) represents the coordinates of the ith scanning point of the lidar in the ENU coordinate system, and \( P_0 \) represents the current position of the lidar in the ENU coordinate system. The position coordinates obtained by the scan matching algorithm are the position coordinates under the ENU coordinate system.

It can be known from the above that GPS and lidar can obtain the UAV position. IMU can obtain the attitude angle,
acceleration and angular velocity of the UAV. To represent the state of UAV, the state quantity is defined as follows:

\[\mathbf{x} = [P_x, P_y, P_z, \theta, \phi, \psi, V_x, V_y, V_z, \omega_x, \omega_y, \omega_z, a_x, a_y, a_z] \]

where \((P_x, P_y, P_z)\) are the positions of the UAV in the ENU coordinate system, \((V_x, V_y, V_z)\) are the velocity of the UAV in the ENU coordinate system, \((\omega_x, \omega_y, \omega_z)\) and \((a_x, a_y, a_z)\) are the angular velocity and acceleration of the IMU. The conversion relationship is as follows:

\[
\begin{bmatrix}
\theta_{k+1} \\
\phi_{k+1} \\
\psi_{k+1}
\end{bmatrix} = 
\begin{bmatrix}
\theta_k \\
\phi_k \\
\psi_k
\end{bmatrix} + T_f^B \begin{bmatrix}
\omega_x \Delta t \\
\omega_y \Delta t \\
\omega_z \Delta t
\end{bmatrix},
\]

\[T_f^B = \begin{bmatrix}
1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\
0 & \cos \phi & -\sin \phi \\
0 & \sin \phi \cos \phi & \cos \phi \cos \theta
\end{bmatrix}\]

\[V_{k+1} = V_k + R^B_{\text{IMU}} \Delta t \]
\[P_{k+1} = P_k + V_k \Delta t + \frac{1}{2} R^B_{\text{IMU}} (\Delta t)^2 \]

where \(x_k \in \mathbb{R}^n\) is the state vector of the system at time step \(k\). \(z^s_k \in \mathbb{R}^{m(s)}\) is the measurement vector of the \(s\)-th sensor at time step \(k\). \(m(s)\) represents dimension of the \(s\)-th sensor. \(w_k \in \mathbb{R}^n\) and \(v^s_k \in \mathbb{R}^{m(s)}\) represent additive process noise and measurement noise of the \(s\)-th sensor, respectively. \(f(\cdot)\) is the process model and \(h^s(\cdot)\) is the non-linear function of the \(s\)-th sensor measurement model. The process noise \(w_k\) and the measurement noise \(v^s_k\) are uncorrelated time-varying white Gaussian noise, and the measurement noise between the sensors is uncorrelated. Its statistical characteristics satisfy:

\[
\begin{cases}
E(w_k) = q_k, & \text{cov}(w_k, w_j) = Q_k \delta_{kj} \\
E(v^s_k) = r^s_k, & \text{cov}(v^s_k, v^s_j) = R^s_k \delta_{kj} \\
\text{cov}(w_k, v^s_k) = 0
\end{cases}
\]

among them, \(Q_k\) is a non-negative definite matrix, \(R^s_k\) is a positive definite matrix, and

\[\delta_{kj} = \begin{cases}
0 & k \neq j \\
1 & k = j
\end{cases}
\]

For the nonlinear system described by equation (9), the multi-sensor UKF filtering process is as follows when the noise of each sensor is known.

1) TIME UPDATE (PREDICTION STEP)
For each time step \(k = 1, 2, \ldots\), we need first to calculate sigma points and then time-update using time-update equations. The calculation method of points is as follows:

\[
\begin{cases}
\xi_{i,k-1} = x_{k-1} & i = 0 \\
\xi_{i,k-1} = x_{k-1} + \sqrt{(n+\lambda)P_{k-1}} & i = 1, 2, \ldots, n \\
\xi_{i,k-1} = x_{k-1} - \sqrt{(n+\lambda)P_{k-1}} & i = n+1, n+2, \ldots, 2n
\end{cases}
\]

where \(\xi\) is sigma-point. \(\lambda\) is the adjustable parameter that controls the distribution of Sigma-points around \(x_{k-1}\). \((\sqrt{(n+\lambda)P_{k-1}})_{i}\) is the \(i\)-th column of the root mean square of the matrix \((n+\lambda)P_{k-1}\). \(x\) is the system state, and \(P\) is the error covariance matrix.

The Sigma points transformed by the process model \(f(\cdot)\) are

\[\xi_{i,k|k-1} = f(\xi_{i,k-1}) \quad (i = 0, 1, \ldots, 2n)\]
\[ \hat{x}_{k|k-1} = \sum_{i=0}^{2n} a_{ij} \hat{\xi}_{i,k|k-1} + q_k \]  
(14)

\[ \hat{P}_{k|k-1} = \sum_{i=0}^{2n} a_{ij}(\hat{\xi}_{i,k|k-1} - \hat{x}_{k|k-1})(\hat{\xi}_{i,k|k-1} - \hat{x}_{k|k-1})^T + Q_k \]  
(15)

2) TIMESTAMP SORTING UPDATE (CORRECTION STEP)

For each available measurement \( z_k^s = z^s(t_k) \) \((s = 1, 2, \ldots, N)\) at time instant \( t = t_k \), we can update the state estimate \( \hat{x}_k \) and error covariance matrix \( P_k \). There may be multiple measurements at the same time, and the filter will fuse them in the order in which they are stored. After the measurement model \( h^i(\cdot) \) is transformed again, the Sigma point is

\[ y_{i,k|k-1} = h^i(\hat{\xi}_{i,k|k-1}) \quad (i = 0, 1, 2, \ldots, 2n) \]  
(16)

Calculate the measured predicted value and its covariance

\[ \begin{align*}
\hat{z}_{k|k-1}^s &= \sum_{i=0}^{2n} a_{ij} y_{i,k|k-1} + r_k^s \\
\hat{P}_{z_{k|k-1}^s} &= \sum_{i=0}^{2n} a_{ij}(y_{i,k|k-1} - \hat{z}_{k|k-1}^s)(y_{i,k|k-1} - \hat{z}_{k|k-1}^s)^T + R_k^s
\end{align*} \]  
(17)

(18)

Calculate the covariance between the predicted value of the state and the measured predicted value

\[ \hat{P}_{s_{k|k-1}} = \sum_{i=0}^{2n} a_{ij}(\hat{\xi}_{i,k|k-1} - \hat{x}_{k|k-1})(y_{i,k|k-1} - \hat{z}_{k|k-1}^s)^T \]  
(19)

Determine the Kalman gain matrix

\[ \begin{align*}
K_k &= \hat{P}_{s_{k|k-1}} \hat{P}_{z_{k|k-1}}^{-1} \\
\hat{x}_k &= \hat{x}_{k|k-1} + K_k \hat{z}_{k|k-1} \\
\hat{P}_k &= \hat{P}_{k|k-1} - K_k \hat{P}_{s_{k|k-1}} (K_k)^T
\end{align*} \]  
(20)

(21)

(22)

After finishing the last measurement-update loop, the state and covariance estimates at time \( t = t_k \) are given by \( \hat{x}_k, \hat{P}_k \). These results become the initial condition for the next time-update step.

In the above steps, \( r_k^s \) and \( R_k^s \) respectively represent the measurement noise and measurement covariance of the \( s \)-th sensor at time step \( k \). \( \hat{z}_{k|k-1}^s \) and \( \hat{P}_{z_{k|k-1}} \) respectively represent the measured prediction and covariance of the \( s \)-th sensor at time step \( k \). \( \hat{P}_{s_{k|k-1}} \) and \( K_k \) respectively represent the Cross covariance and Kalman gain of the \( s \)-th sensor at time step \( k \). In equation (21), \( \hat{z}_{k|k-1} \) is the measurement value of the \( s \)-th sensor. According to the specific type of measured data, the measured noise type and the measured covariance type in the filtering process can be obtained.

B. MULTI-WINDOW ADAPTIVE UKF MODEL

In many cases, system noise and measurement noise are unknown. Therefore, it is necessary to estimate the noise. According to the nonlinear system described by equation (9), it is easy to know

\[ \begin{align*}
q_k &= E[w_k] = E[x_k - f(x_k|1)] \\
r_k^s &= E[v_k^s] = E[z_k^s - h^s(x_k)] \\
Q_k &= \text{cov}(w_k w_k^T) = E[(x_k - f(x_k|1) - q_k)(x_k - f(x_k|1) - q_k)^T] \\
R_k^s &= \text{cov}(v_k^s v_k^{sT}) = E[(z_k^s - h^s(x_k) - r_k^s)(z_k^s - h^s(x_k) - r_k^s)^T]
\end{align*} \]  
(23)

However, since the true state \( x_k \) cannot be observed during the filtering process, it is not possible to directly apply the equation (23) to estimate the system noise statistics. Consider a time window of width \( W \), and assume that the system noise statistics are constant or very small within this window. In equation (23), using the estimated value \( \hat{x}_k \) or predicted value \( \hat{x}_{k|k-1} \) of \( x_k \) instead of \( x_k \), and calculate the expected value by using the sliding window approximation method to obtain the following suboptimal estimator. Among them, the size of \( W \) should be moderate, and \( W \) is set to 10 in this paper. IMU data experiences 100 ms in a window, LIDAR is 1 s, and GPS is 10 s. IMU and LIDAR can be considered that the noise does not change within a window. Because the GPS window is too long, it will be discussed in Section V-C.

\[ \begin{align*}
\hat{q}_k &= \frac{1}{W} \sum_{j=1}^{W} (\hat{x}_{k-j} - f(\hat{x}_{k-1-j})) \\
\hat{r}_k^s &= \frac{1}{W} \sum_{j=1}^{W} (z_{k-j}^s - h^s(\hat{x}_{k-1-j})) \\
\hat{Q}_k &= \frac{1}{W} \sum_{j=1}^{W} [((\hat{x}_{k-j} - f(\hat{x}_{k-1-j}) - \hat{q}_{k-j}))^T] \\
\hat{R}_k^s &= \frac{1}{W} \sum_{j=1}^{W} [(z_{k-j}^s - h^s(\hat{x}_{k-1-j}) - \hat{r}_{k-j}^s)^T]
\end{align*} \]  
(24)

among them, \( f(\hat{x}_{k-1-j}) \) represents the posterior mean value obtained by transforming the state estimation value \( \hat{x}_{k-1-j} \) by the non-linear function \( f(\cdot) \). For KF, the accuracy value of \( f(\hat{x}_{k-1-j}) \) can be obtained by state transition matrix transformation; however, for non-linear UKF, \( f(\hat{x}_{k-1-j}) \) can only be obtained by UT transformation with third-order Taylor
accuracy approximation [14], that is:

$$f(\hat{x}_{k-1-j}) = \sum_{i=0}^{2n} \omega_i f(\xi_{i,k-1-j}) \tag{25}$$

Similarly, it can only be obtained by a third-order Taylor approximation, ie

$$h(\hat{x}_{k-j-1}) = \sum_{i=0}^{2n} \omega_i h'(\xi_{i,k-j-1}) \tag{26}$$

In the literature [6], the noise statistical estimator formula (24) has been proved to be an unbiased estimate of $q_k$ and $r_k^s$, but the estimates of $Q_k^s$ and $R_k^s$ are biased.

However, an unbiased estimation formula can be obtained through a biased proof as follows

$$\hat{Q}_k = \frac{1}{W} \sum_{j=1}^{W} [\hat{P}_{k-j} + K_{k-j}^s \hat{z}_{k-j} - \hat{z}_{k-j}]^T (K_{k-j}^s)^T$$

$$- \sum_{i=0}^{2n} \omega_i (\xi_{i,k-j-1-j} - \hat{x}_{k-j-1-j})$$

$$\hat{z}_{k-j}(\hat{x}_{k-j-1-j} - \hat{z}_{k-j-1-j})^T \tag{27}$$

$$\hat{R}_k = \frac{1}{W} \sum_{j=1}^{W} [\hat{z}_{k-j} - \hat{z}_{k-j}]^T - \sum_{i=0}^{2n} \omega_i (Y_{i,k-1-j-1-j} - \hat{z}_{k-j}(\hat{x}_{k-j-1-j} - \hat{z}_{k-j-1-j})^T \tag{28}$$

Here $\hat{z}_{k-j} = \hat{z}_{k-1-j} - \hat{z}_{k-j-1}$. For a window width of $W$, $W$ measurements are made in a short time $(t_k - W, t_k)$. Because the system noise is constant or the change is small in the statistics window $W$, there is

$$\begin{cases} q_k = q_{k-j} \\ r_k^i = r_k^{i-j} \\ Q_k = Q_{k-j} \\ R_k^i = R_k^{i-j} \end{cases} \text{ (j = 1, 2, \ldots, N)} \tag{29}$$

Therefore, $N$ sliding windows are needed for $N$ sensors. Estimate their measurement noise through $N$ sliding windows.

$$\begin{cases} \hat{r}_k^i = \frac{1}{W} \sum_{j=1}^{W} (\hat{z}_{k-j} - h(\hat{x}_{k-j}))\
\hat{R}_k^i = \frac{1}{W} \sum_{j=1}^{W} [\hat{z}_{k-j}]^T - \sum_{i=0}^{2n} \omega_i (Y_{i,k-1-j-1-j} - \hat{z}_{k-j}(\hat{x}_{k-j-1-j} - \hat{z}_{k-j-1-j})^T \tag{30}$$

$r_k^i$ and $R_k^i$ are estimates of measurement error and measurement covariance of the $s$-th sensor data.

### C. TIME-VARYING NOISE PROCESSING

In the UKF filter for noise estimation of the multi-sliding window, the method of replacing the true value with the optimal estimation value is used to estimate the measurement noise and process noise. The premise of this method is that in the set window, the time is short enough to make equation (29) true. In this paper, multiple windows are set for multi-sensor noise estimation. Each window has the same width, but experiences different time. This results in some windows being unsatisfied with the formula (29). This leads to inaccurate estimation of process noise and measurement noise, which eventually leads to filter divergence.

**FIGURE 3. Schematic diagram of statistical deviation of GPS data.**

In this paper, sensors are classified based on whether their measurement errors accumulate over time. For example, the displacement predicted by IMU will increase the cumulative error over time. GPS data is different from IMU data, which does not increase the measurement error over time. When GPS data and IMU data are fused together in this paper, sliding window estimation of IMU noise is more accurate. Since GPS needs to go through $W$ seconds under the window width of $W$, GPS noise estimation is not accurate. As shown in Fig. 3, the solid line represents the filtered trajectory, and the dotted line represents the real trajectory. The green circle represents the magnitude of the process covariance $\hat{Q}_k$, and the process noise $\hat{q}_k$ becomes very small. The yellow circle indicates the actual measurement covariance of the GPS sensor $\hat{R}_k^{GPS}$ is the statistical measurement covariance of the GPS sensor and does not change much with time. As time increases, the filtered trajectory of the solid arrow shifts more and more, but the statistical process noise and process covariance do not change much. Among them, the statistical measurement error $\hat{r}_k^{GPS}$ of GPS is getting larger and larger.

Data abnormality capture is performed according to the noise statistical result $\hat{r}_k^{GPS}$. According to experience, set the GPS positioning noise upper limit to $pr^{GPS}$. When $\hat{r}_k^{GPS} > pr^{GPS}$ we think that the data is biased. When this occurs, the corresponding process noise and process covariance need to be adjusted.

$$\hat{q}_k = \frac{1}{W} \sum_{j=1}^{W} (\hat{x}_{k-j} - f(\hat{x}_{k-1-j}) - \hat{r}_k^{GPS}) (\frac{\Delta t_k^{GPS}}{\Delta t_q}) \tag{31}$$

$$\hat{Q}_k = \hat{Q}_k \ast \frac{\hat{r}_k^{GPS}}{\hat{q}_k} \tag{32}$$

where $\Delta t_k^{GPS}$ represents the time interval between two GPS data. Similarly, $\Delta t_q$ represents the average time interval between filter updates.
Because process noise is closely related to process covariance and state quantities, the above method can be used to quickly converge the filter to the global true value.

VI. PERFORMANCE ANALYSIS OF FILTERING METHOD

A. FILTER METHOD ACCURACY ANALYSIS

For the filtering method proposed in this paper, covariance analysis methods are used for accuracy analysis [36]. Assume that the data state dimension of all sensors is \( n \), the observation data dimension is \( m \), and the total number of sensors is \( N \), set the time elapsed during the filter fusion process to \( T \). It is assumed the filtering has converged within this time \( T \).

Theorem 1: Under the above conditions, distributed filter fusion and TSU filter fusion have the same fusion accuracy.

Proof: According to the simplified distributed filtering process and TSU filtering process shown in Appendix, the process covariance of TSU filtering process is set as \( P_{T} \), and the distributed filtering process covariance is set as \( P_{s} \).

\[
H_{s} = H_{k}^{N+1-i} \\
R_{s} = R_{k}^{N+1-i} \\
K_{s} = K_{k}^{N+1-i} \\
P_{T} = P_{k} \left( \prod_{i=1}^{N} \left[ I - K_{k}^{N+1-i}H_{k}^{N+1-i} \right] \right) \\
N \times \prod_{i=1}^{N} \left[ I - K_{k}^{N+1-i}H_{k}^{N+1-i} \right] \times P_{k|i-1}
\]

When \( N = 1 \), there is

\[
K_{s} = K_{k}^{N+1-i} \\
P_{T} = P_{k} \left( \prod_{i=1}^{N} \left[ I - K_{k}^{N+1-i}H_{k}^{N+1-i} \right] \right) \times P_{k|i-1}
\]

In the case of filter convergence [41], the final process covariance will be stabilized as long as the iteration times are sufficient. That is:

\[
\prod_{k=s}^{T} \left[ I - K_{k}^{N+1-i}H_{k}^{N+1-i} \right] \times P_{k|i-1}\]

So there is

\[
P_{T} = \prod_{s=1}^{N} \prod_{k=s}^{T} \left[ I - K_{k}^{N+1-i}H_{k}^{N+1-i} \right] \times P_{k|i-1}
\]

B. ANALYSIS OF CALCULATION OF FILTERING FRAMEWORK

The analysis of the calculation problem is mainly the analysis of the calculation problem caused by the filtering fusion method. According to the filtering process shown in Appendix, the calculation is calculated. When calculating calculations, follow these guidelines:

1) The types of calculations are summarized into four types: addition, assignment, multiplication, and division;
2) One assignment operation is equivalent to one addition operation;
3) Matrix operations are calculated as operations between matrix elements.

Table 1 shows the calculation formula for the single filter fusion. It can be seen from the table that when \( N = 1 \), the calculation amount of the three filtering frames is equal. When \( N \geq 2 \), the calculation of the TSU method is less than the calculation of distributed and does not increase the calculation with the increase of the number of sensors. In order to quantify the advantages of this algorithm, two different values of \( m, n, \) and \( N \) are discussed here. The results are shown in Table 2 and Table 3.

From the statistical results of the calculation of the filtering framework given in Table 2 and Table 3, compared with the distributed filtering method, the filtering method in this paper reduces the total computation by 62.7% and 82.3% respectively. It can be seen from Table 1 that with the increase of the number of system sensors, state dimension, and observation dimension, the advantages of the algorithm in this paper is computation will become more and more obvious.
TABLE 1. Filter method calculations statistics.

| Operation | Distributed | TSU | Distributed-TSU |
|-----------|-------------|-----|-----------------|
| Addition  | \( N [(m-1)! + 1]m^3 \) + \( n^3 + 2n^2 + 2n \) + \( 3Nm^2 + 2Nm^3 + 2Nn + Nn \) | \( (m-1)! + 1)m^3 \) + \( 3n^3 + 2n^2 + n \) + \( 3mn^2 + 2mn^3 \) + \( 3mn^2 + 2mn^3 + 2mn \) | \( (N-1)(m-1)!m^3 \) + \( m^3 + 2n^3 + n \) + \( 3mn^2 + 2mn^3 \) + \( 3mn^2 + 2mn^3 + 2mn \) |
| Multiplication | \( N [(m^n - m - 1) + m^2] \) + \( n^2 + n^2 + 2Nn + 2Nm^3 \) + \( 3Nm^2 + 2Nm^3 \) + \( 2Nnm^3 + 2Nn \) | \( m^2 [(m^n - m - 1) + m^2] \) + \( n^2 + n^2 + 2mn + mn^3 \) + \( 3mn^2 + 2mn^3 \) + \( 2mn + mn^3 + 2mn \) | \( (N-1)m^2 \) | 
| Division | \( \frac{N}{m^2} \) | \( \frac{m^2}{m^2} \) | \( \frac{(N-1)m^2}{m^2} \) |
| SUM | \( N [(m^n - m - 1) + m^2] \) + \( n^2 + n^2 + 2Nn + 2Nm^3 \) + \( 3Nm^2 + 2Nm^3 \) + \( 2Nnm^3 + 2Nn \) + \( 6Nm^2 + 4Nm^2 + 4Nnm^3 + 2Nnm + Nm^2 \) | \( m^2 [(m^n - m - 1) + m^2] \) + \( n^2 + n^2 + 2mn + mn^3 \) + \( 3mn^2 + 2mn^3 \) + \( 2mn + mn^3 + 2mn \) + \( 6mn^2 + 4mn^2 + 4mn^2 + 4mn^3 + 4mn^3 + 4mn^2 + 4mn^2 \) | \( (N-1)m^2 \) |

TABLE 2. Filter framework calculations statistics \((m = 3, n = 3, N = 3)\).

| Operation | Distributed | TSU | Distributed-TSU |
|-----------|-------------|-----|-----------------|
| Addition  | 690         | 264 | 426             |
| Multiplication | 774         | 282 | 492             |
| Division  | 27          | 9   | 18              |
| SUM       | 1491        | 555 | 936             |

TABLE 3. Filter framework calculations statistics \((m = 5, n = 5, N = 6)\).

| Operation | Distributed | TSU | Distributed-TSU |
|-----------|-------------|-----|-----------------|
| Addition  | 9155        | 1680| 7475            |
| Multiplication | 19530       | 3380| 16150          |
| Division  | 150         | 25  | 125             |
| SUM       | 28835       | 5085| 23750          |

TABLE 4. Simulation sensor error setting value.

| Error Type       | Drift     | Gaussian Noise |
|------------------|-----------|----------------|
| IMU acceleration | X: 0.1m/s | X: 0.35m/s     |
| GPS position     | X: 0.1m/s | Y: 0.1m/s      |
| Z: 0.1m/s       | Z: 0.35m/s| Z: 0.01m       |
| Lidar ranging    | 0%        | 0.1% of range  |

VII. EXPERIMENT AND ANALYSIS
A. SIMULATION
The simulation experiment is carried out on the gazebo [33] simulation platform under the Linux system. The UAV is simulated by the open-source simulation software package Hector_quadrotor [34]. The noise and drift of simulation sensor are set in the simulation scene, as shown in Table 4.

We build a simulation scenario as shown in Fig. 4 in the gazebo simulation environment, and set cuboid obstacles to provide obstacles for the simulation of 2D lidar positioning [37]. The red track is the preset track of the UAV. The UAV takes off in the upper right corner of the red rectangular trajectory in the picture and flies counterclockwise.

In order to verify the performance of the adaptive filtering proposed in this paper, experiments were performed on GPS and IMU sensors. The algorithm in this paper proposes a multi-sliding window classification adaptive unscented Kalman filter based on the timestamp sort update filter. The proposed MWCAUKF is compared with standard EKF, UKF, and improved Sage-Husa adaptive unscented Kalman filter (SHAUKF). The comparison chart of simulation trajectory is shown in Fig. 5.
From the positioning of trajectory in Fig. 5, MWCAUKF is obviously closer to the true trajectory than standard UKF and EKF trajectories. Compared with the improved SHAUKF, MWCAUKF is not easy to distinguish the advantages and disadvantages from the trajectory. Fig. 6 and Fig. 7 further compare the two aspects of position error and velocity error.

From Fig. 6 and Fig. 7, it can be seen that the MWCAUKF has smaller position and velocity errors than the standard UKF and EKF. In addition, the MWCAUKF algorithm has a smaller trajectory error fluctuation than SHAUKF, EKF, and UKF. It can be seen that MWCAUKF has better performance in tracking real data. Therefore, compared with the standard EKF, UKF and SHAUKF, our method has higher fusion accuracy, smaller volatility, and can quickly converge to the real value. Specific experimental data are shown in Table 5.

### TABLE 5. Simulation experiment error average data.

| Error type       | EKF  | UKF  | SHAUKF | MWCAUKF |
|------------------|------|------|--------|---------|
| X-axis position error m | 0.1939 | 0.4837 | 0.3197 | 0.1448  |
| Y-axis position error m | 0.2562 | 0.5489 | 0.2866 | 0.1926  |
| X-axis speed error m/s | 0.1267 | 0.0930 | 0.0911 | 0.0831  |
| Y-axis speed error m/s | 0.1594 | 0.1193 | 0.1113 | 0.0860  |

From the data in Table 5, when the noise is unknown, the average error of the position and velocity of the algorithm in this paper is better than the current SHAUKF adaptive filtering algorithm. Table 5 shows that standard UKF and EKF filters have larger errors than SHAUKF when the noise is unknown. Therefore, compared with EKF, UKF and SHAUKF, the algorithm in this paper has smaller average error in position and velocity, so the text algorithm has the highest accuracy in data fusion.

The algorithm design in this paper is aimed at multi-sensor data fusion. In order to reflect the applicability and effectiveness of the algorithm for multiple sensors, lidar data fusion is added. The positioning trajectory is shown in Fig. 8.

It can be seen from the experimental trajectory of Fig. 8 that with the addition of a sensor, the coincidence between the positioning trajectory and the real trajectory increases. Among them, the average error of X-axis velocity is 0.0715m/s, the average error of Y-axis velocity is 0.1173m/s, the position error of X-axis is 0.0713m, and the position error of Y-axis is 0.0653m. It can be seen that when a lidar sensor is added, the positioning accuracy is significantly improved, indicating that the algorithm in this paper is effective for multi-sensor fusion.

### B. REAL EXPERIMENT

In order to verify the system model, the adaptive Kalman filter based on multi-sensor fusion is applied to the UAV positioning system. The multi-sensor fusion UAV positioning systems require related hardware and software environments.
In the hardware environment, the flight platform uses DJI’s M100 quad-rotor drone. The drone itself carries an IMU module to provide IMU data; it uses RPLIDAR-A2 lidar from Silan Technology; Raspberry Pi is used as a data fusion processor; GPS receiving device model is Ublox8030. The software environment is consistent with the simulation, and the program is debugged under ROS (Robot Operating System).

The algorithm proposed in this paper is applied to the positioning system of real UAV, and experiments are carried out in relatively open outdoor scenes. Since DJI’s fused data is relatively accurate [35], the GPS data of DJI’s M100 drone is used as the comparison data of the positioning data. The experimental trajectory comparison chart is shown in Fig. 10.

It can be seen from the Fig. 10 that the UAV flight distance is about 100m. Compared with the GPS positioning data of the M100, all experimental trajectories have the same offset problem. This problem is a coordinate offset problem caused by different positioning satellites [38]. DJI’s GPS uses Global Positioning System (GPS). Ublox8030 GPS uses Bei-dou Satellite Navigation System (DBS). The experiment only compares the advantages and disadvantages of the algorithm in this paper with the algorithms of UKF, EKF and SHAUKF. Therefore, this offset can be ignored. As can be seen from Fig. 10, compared with the positioning trajectories under the other three filtering algorithms, the trajectory obtained by the algorithm in this paper is the closest to the M100 data trajectory.

Fig. 11 and Fig. 12 also illustrate the excellent performance of the MWCAUKF algorithm. The position and velocity of MWCAUKF are closer to the DJI’s own GPS data. Compared with the other three filtering algorithms, MWCAUKF has less volatility and better tracking ability for real data. Table 6 shows the actual experimental data. These data are the average error compared to DJI’s GPS data.

From the experimental data, we can see that the MWCAUKF algorithm is still applicable in real scenarios. Compared with DJI’s GPS positioning data, the average positioning error on the X-axis is 0.4407m and the Y-axis is 1.0433m. In addition, MWCAUKF algorithm has higher positioning accuracy than SHAUKF algorithm. The overall data shows that the MWCAUKF algorithm has advantages in data accuracy in terms of position and velocity of the UAV. By combining Fig. 11, Fig. 12 and Table 6, it can be seen that MWCAUKF data has higher fusion accuracy compared with UKF, EKF, and SHAUKF, higher tracking of data truth value, lower volatility of position and velocity, and better stability of fusion algorithm.
When the UAV is performing the task, the sensor data may be invalid. GPS may have weak satellite reception or satellite signal loss. Lidar can only work in the presence of obstacles. Aiming at the performance problem of filtering fault tolerance, this paper conducted experiments on the roof of the experimental building of the school. The continuity of the experimental trajectory verifies the fault tolerance and stability of the filtering framework in this paper for the fusion of multiple sensor data. The experimental trajectory is shown in the actual image map as shown in Fig. 13.

The roof of the building in the yellow box in Fig. 13(a) is shown in Fig. 13(b). The red trace in Fig. 13(a) is the actual trajectory. As can be seen from the figure, the positioning trajectory coincides with the actual roof corridor. During the experiment, the UAV flew around the corridor on the roof of the building. The specific trajectory is shown in Fig. 14. As can be seen from the figure, in the two areas of A and B, the GPS positioning is affected due to the position of the adjacent buildings, and the positioning is not accurate. This article solves this problem by merging data from lidar. Similarly, when the data scanned by the lidar is insufficient, the positioning of the UAV cannot be completed, and GPS positioning can be used as compensation to achieve stable positioning of the UAV. Experiments prove that the positioning system can handle the positioning in the case of GPS failure or lidar failure, and achieve stable positioning of the UAV.
VIII. CONCLUSION

It can be seen from the experimental data that the filtering method proposed in this paper has a lower computational cost than the distributed filtering algorithm. In this method, there is no need to perform strict time synchronization on the sensor data, and the measurement data of the sensors can be easily fused regardless of the update rate of the sensors. Compared with the standard UKF, EKF, and Suga-Husa adaptive unscented Kalman filter, the proposed algorithm has higher accuracy and better stability. It can track real data well compared with the standard UKF, EKF, and Suga-Husa method proposed in this paper has a lower computational cost.

It can be obviously found that TSU filtering form and kalman filtering form is almost the same. The meaning of the symbol is consistent with the text.

REFERENCES

[1] E. Schnipke, S. Reidling, J. Meiring, W. Jeffers, M. Hashemi, R. Tan, A. Nemati, and M. Kumar, “Autonomous navigation of UAV through GPS-denied indoor environment with obstacles,” in Proc. AIAA Infotech Aerosp., Jan. 2015, pp. 5–9.
[2] J.-J. Xiong and E.-H. Zheng, “Position and attitude tracking control for a quadrotor UAV,” ISA Trans., vol. 53, no. 3, pp. 725–731, May 2014.
[3] H. Li, X. Li, W. Ding, Y. Shi, and Y. Li, “Multi-sensor based high-precision direct georeferencing of medium-altitude unmanned aerial vehicle images,” Int. J. Remote Sens., vol. 38, nos. 8–10, pp. 2577–2602, May 2017.
[4] P. Roy, “Enhanced Synthetic Vision Systems and Multi-Sensor Data Fusion to Improve Operational Capabilities of Small Tactical UAV,” in Proc. AIAA Infotech Aerosp. Conf. Exhibit, May 2007, p. 2952.
[5] R. E. Kalman, “A new approach to linear filtering and prediction problems,” J. Basic Eng., vol. 82, no. 1, pp. 35–45, Mar. 1960.
[6] S. Gao, G. Hu, and Y. Zhong, “Windowing and random weight-based adaptive unscented Kalman filter,” J. Adapt. Control Signal Process., vol. 29, no. 2, pp. 201–223, Feb. 2015.
[7] X. M. Wang, W. Q. Liu, and Z. L. Deng, “Robust centralized fusion steady-state Kalman filter with uncertain parameters and measurement noise variances,” Appl. Mech. Mater., vols. 701–702, pp. 532–537, Dec. 2014.
[8] Y. Xia, X. Peng, J. Pan, Y. Xiang, and Q. Gao, “Multisensor distributed weighted Kalman filter fusion with network delays, stochastic uncertainties, autocorrelated, and cross-correlated noises,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 48, no. 5, pp. 716–726, May 2018.
[9] N. Sahu, R. Gupta, A. Kumar, R. Bahl, and I. N. Kar, “Centralized Kalman filter for fusion of multiple on-board auxiliary sensors with INS for underwater navigation,” in Proc. IEEE Underwater Technol.(UT), Apr. 2019, pp. 1–6.
[10] H. Lin and S. Sun, “Distributed fusion estimator for multisensor multisite systems with correlated noises,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 48, no. 7, pp. 1131–1139, Jul. 2018.
[11] J. Feng, Z. Wang, and M. Zeng, “Distributed weighted robust Kalman filter fusion for uncertain systems with autocorrelated and cross-correlated noises,” Inf. Fusion, vol. 14, no. 1, pp. 78–86, Jan. 2013.
[12] M. A. Bakr and S. Lee, “Distributed multisensor data fusion under unknown correlation and data inconsistency,” Sensors, vol. 17, no. 11, p. 2472, Oct. 2017.
[13] H. Lin and S. Sun, “Globally optimal sequential and distributed fusion state estimation for multi-sensor systems with cross-correlated noises,” Automatica, vol. 101, pp. 128–137, Mar. 2019.
[14] G. Mao, S. Drake, and B. D. O. Anderson, “Design of an extended kalman filter for UAV localization,” in Proc. Inf. Decis. Control, Feb. 2007, pp. 224–229.
[15] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” Proc. IEEE, vol. 92, no. 3, pp. 401–422, Mar. 2004.
[16] A. P. Sage and W. G. Husa, “Adaptive filtering with unknown prior statistics,” in Proc. Joint Autom. Control Conf., no. 7, 1969, pp. 760–769.
[17] F. Tian, J. Y. Zheng, and T. Zhang, “Sensor fault diagnosis for an UAV control system based on a strong tracking Kalman filter,” Appl. Mech. Mater., vols. 687–691, pp. 270–274, Dec. 2014.
P. Cheng, S. Dang, F. He, and K. Wang, “Adaptive fuzzy Kalman filter based on INS/RADAR integrated navigation system,” in Proc. 2nd Int. Conf. Autom. Control Inf. Eng. (ICACE), vol. 119, 2017, pp. 5–9.

X. X. Wang and L. Zhao, “Adaptive fusion filtering algorithm and its application for INS/GPS integrated navigation system,” J. Astronaut., vol. 31, pp. 2504–2511, 2010.

X. Li and W. G. Zhang, “Multi-sensor integrated navigation for intelligent vehicles based on H∞filter,” J. Syst. Simul., vol. 19, no. 18, pp. 4168–4173, 2007.

T. Moore and D. Stouch, “A generalized extended kalman filter implementation for the robot operating system,” in Intelligent Autonomous Systems, vol. 13. Cham, Switzerland: Springer, 2016, pp. 335–348.

S.-M. Oh, “Multisensor fusion for autonomous UAV navigation based on the unscented Kalman filter with sequential measurement updates,” in Proc. IEEE Conf. Multisensor Fusion Integr., Sep. 2010, pp. 217–222.

J. Xiong, L. Shu, Q. Wang, W. Xu, and C. Zhu, “A scheme on indoor tracking of ship dynamic positioning based on distributed multi-sensor data fusion,” IEEE Access, vol. 5, pp. 379–392, 2017.

H. Fan, “Speed measurement of ultrasonic motor based on Kalman filter algorithm,” Small Special Elect. Mach., vol. 11, p. 1, Apr. 2015.

L. I. Z. Xu-Jing and H. L. Wei, “Speech enhancement based on sub-band kalman filtering,” Signal Process., vol. 25, no. 9, pp. 1474–1478, Sep. 2009.

L. Liu, A. Yang, X. Tu, M. Fei, and W. Naeem, “Distributed weighted fusion estimation for uncertain networked systems with transmission time-delay and cross-correlated noises,” Neurocomputing, vol. 270, pp. 54–65, Dec. 2017.

W. Chenglin, L. Bing, and G. E. Quan-Bo, “A data fusion algorithm based on filtering step by step,” Acta Electron. Sinica, vol. 32, no. 8, pp. 1264–1267, 2004.

Y. Geng, Z. Cui, H. Zhang, and J. Fang, “Adaptive fading Kalman filter with applications in integrated navigation system,” J. Beijing Univ. Aeronaut. Astronaut., vol. 30, no. 5, pp. 434–437, 2004.

J. Zhang, Z.-H. Jin, and W.-F. Tian, “A suboptimal Kalman filter with fading factors for DGPS/MEMS-IMU/magnetic compass integrated navigation,” in Proc. IEEE Int. Conf. Intell. Transp. Syst., vol. 2, May 2004, pp. 1229–1234.

M. K. Al-Sharman, B. J. Emran, M. A. Jaradat, H. Najjaran, R. Al-Husari, and Y. Zweiri, “Precision landing using an adaptive fuzzy multi-sensor data fusion architecture,” Appl. Soft Comput., vol. 69, pp. 149–164, Aug. 2018.

C.-H. Tseng, S.-F. Lin, and D.-J. Iwo, “Fuzzy adaptive cubature Kalman filter for integrated navigation systems,” Sensors, vol. 16, no. 8, p. 1167, Jul. 2016.

Y. Dai, Z.-Y. Tang, Z.-B. Zhu, and L. Zhang, “Uniform calibration of networked radar based on unmanned aerial vehicle and GPS,” in Proc. 3rd Int. Conf. Control, Autom. Robot. Sens. Netw. (ICCAENS), Aug. 2017, pp. 61–64.

M. Schmittner, A. Lakina, V. Vacek, J. Das, C. P. Buskirk, S. Rees, J. Szitapanovits, R. Grosou, and V. Kumar, “OpenUAV: A UAV testbed for the CPS and robotics community,” in Proc. ACM/IEEE 9th Int. Conf. Cyber-Phys. Syst. (ICCPS), Apr. 2018, pp. 130–139.

W. A. Arokiasamy, P. Vadakkepat, K. C. Tan, and D. Srinivasan, “Vector directed path generation and tracking for autonomous unmanned aerial/ground vehicles,” in Proc. IEEE Congr. Evol. Comput. (CEC), Jul. 2016, pp. 1375–1381.

S. Świerczynski and A. Felski, “Determination of the position using receivers installed in UAV,” in Proc. Eur. Navigat. Conf. (ENC), Apr. 2019, pp. 1–4.

A. X. Yu, W. D. Hu, and W. H. Zhou, “Performance comparison of multisensor measurement fusion algorithms,” J. Nat. Univ. Defense Technol., vol. 25, no. 6, pp. 39–44, 2001.

S. Kohlbrecher, O. von Stryk, J. Meyer, and U. Klingauf, “A flexible and scalable SLAM system with full 3D motion estimation,” in Proc. IEEE Int. Symp. Saf., Secure, Rescue Robot., Nov. 2011, pp. 155–160.

L. Changsheng, C. Chenglin, W. Liangliang, and Q. Yi, “Analysis of positioning accuracy and convergence of BDS/GPS dual-mode static PPP,” J. Hebei Univ. Technol. (Natural Sci.), vol. 40, pp. 481–485, Mar. 2017.

D. H. Zhou and P. M. Frank, “Strong tracking filtering of nonlinear time-varying stochastic systems with coloured noise: Application to parameter estimation and empirical robustness analysis,” Int. J. Control, vol. 65, no. 2, pp. 295–307, Sep. 1996.

W. Wang, G. Chen, R. Wu, D. Lu, and L. Wang, “A low-complexity spoofing detection and suppression approach for ADS-B,” in Proc. Integr. Commun., Navigat. Survell. Conf. (ICNS), Apr. 2015, p. K2.

T. Hamel and C. Samson, “Position estimation from direction or range measurements,” Automatica, vol. 82, pp. 137–144, Aug. 2017.

JING PENG received the B.S. degree in science from East China Jiaotong University, Jiangxi, China, in 2017. He is currently pursuing the master’s degree in computer science with the South China University of Technology. His current research interests include multisensor data fusion and simultaneous localization and mapping.

PING ZHENG received the B.S. degree in mechanical engineering, and the M.S. and Ph.D. degrees in robotics from Tianjin University, Tianjin, China, in 1985, 1988, and 1994, respectively. He is currently a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His research interests include intelligent networked robotics, intelligent networked manufacturing, human–computer interaction, and real-time embedded systems.

LANXIANG ZHENG received the B.S. degree in electronic information engineering from Shenzhen University, Shenzhen, China, in 2017. He is currently pursuing the M.S. degree in computer science with the South China University of Technology. His current research interests include distributed control, obstacle detection, and collaborative obstacle avoidance.

JIA TAN received the B.S. degree in electronic information science and technology from the Wuhan University of Technology, Wuhan, China, in 2008, and the M.S. degree in software engineering from the South China University of Technology, in 2012, where he is currently pursuing the Ph.D. degree in computer science. His current research interests include robotic software architecture, multimobile robot systems, and mobile wireless ad-hoc networks.