# Model-Free Integrated Navigation of Small Fixed-Wing UAVs Full State Estimation in Wind Disturbance

Yue Yang, *Student Member, IEEE*, Xiaoxiong Liu, Xuhang Liu, Yicong Guo, and Weiguo Zhang

***Abstract*—This paper presents a model-free distributed multi-sensor extended Kalman filter (DMSEKF) full state estimation algorithm to provide long-term convergent flight parameters for small fixed-wing unmanned aerial vehicles (UAVs). The full state has the attitude, velocity, position, airspeed, and 2D horizontal wind speed. The airspeed and wind speed are estimated in wind disturbance to provide more robust perception information. The model-free estimator has a low-cost standard sensor suite, including an IMU, a magnetometer, a barometer, a GPS module, and an airspeed tube, rather than the aerodynamic model of the UAVs to increase the multi-sensor fusion algorithm versatility in various UAVs. Then, the full state integrated navigation model is established based on the onboard sensor suite fused by the distributed tightly-coupled EKF. In addition, a consistent multiple sensors data processing method is designed to synchronize the time node of all onboard sensors. Finally, the proposed algorithm is verified through the experimental flight sensor data. The results demonstrate that the proposed algorithm can provide a reliable full state vector and achieve an effective solution performance during the UAVs flight application.**

***Index Terms*—Model-free, Kalman filter, full state estimation, UAVs, wind disturbance, integrated navigation model.**
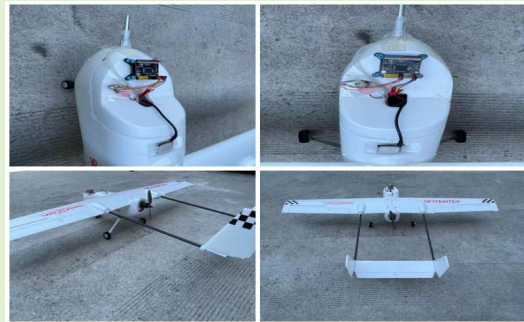
I. INTRODUCTION

SMALL fixed-wing UAVs are widely employed in many applications [1]–[3] due to their excellent maneuverability and economy. The small UAVs usually perform different tasks at a relatively low flying speed, which is a little higher than the wind speed. The flight performance is greatly affected by the wind [4]–[6]. Thus, estimating wind speed can improve the UAVs flight control system (UFCS) [7] and increase survivability in windy disturbance situations. Additionally, the state parameters (attitude, velocity, position, airspeed, and 2d horizontal wind speed) are essential for the small fixed-wing UAVs are in various flight conditions. Furthermore, the UAVs, as a multi-sensor system [8], mount multiple low-cost devices, including the inertial measurement unit (IMU), magnetometer (MAG), barometer (BARO), global positioning system (GPS),

and air data system (ADS). Then depending on some advanced sensor infusion algorithm [9], [10], and provide the required state estimation vector to ensure the long-term autonomous flight.

In the classic UAVs integrated navigation [11], the state vector is usually the attitude, velocity, and position depending on the IMU, MAG, and GPS combined with some fusion methods, including Mahony complementary filter [12], Madgwick filter [13], Kalman filter [14], and artificial intelligence algorithm [15]. The classic navigation can achieve a satisfactory solution when in the normal flight condition and be used in the inner-outer loop control system, path planning, and avoidance. However, the navigation performance will be affected by the wind disturbance [16] resulting in degraded calculation accuracy even divergence. Thus, wind sensing and estimation [17] along the UAV flight trajectory will be very meaningful and tackled by solving the wind triangulation, which has been researched in some papers using a standard sensor suite.

The system model of the state estimation in wind disturbance can be divided into the aerodynamic model-aided and the complete model-free or model-independent. The aerodynamic model-aided method [18] needs to be designed by known aerodynamic model coefficient about the force and moment. However, the corresponding coefficient will not be
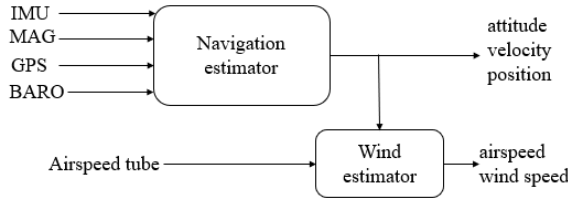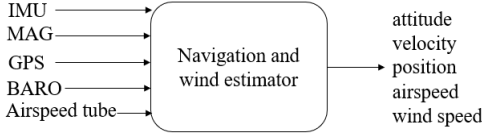
Fig. 1. Loosely-coupled scheme.

Fig. 2. Tightly-coupled scheme.

correct when the UAVs' model changes and be obtained again using computational fluid dynamics(CFD). Furthermore, the designed aerodynamic model [19] is pretty complex and unsuitable for real flight applications. Compared with the model-aided, the model-free system [20], [21] only relies on the onboard mounted sensor to be widely employed in various aircraft without aircraft modeling or tuning of the modeling uncertainties, which is useful for fault detection and isolation [22] of the ADS system. Meanwhile, the model-free system is often combined with the nonlinear Kalman filters, mainly EKF and UKF, to provide the real-time online wind speed and other parameters. A sensing system using this approach [18] can quickly work without requiring any changes to the filter, except for tuning for different sensors.

In addition, there are mainly two fusion frameworks on full state estimation consisting of the loosely-coupled and tightly-coupled in many studies. The loosely-coupled scheme [23], [24] shown in Fig. 1, also named the cascaded scheme, has the navigation estimator and wind estimator. This scheme starts from the navigation estimator to calculate the navigation states as the input and then runs the wind estimator to obtain the airspeed and wind speed. The navigation estimator and wind estimator operate independently and do not affect each other. In contrast, the tightly-coupled scheme [25], [26] shown in Fig. 2 takes the airspeed and wind speed as the estimated state, and only one estimator runs from all available observation values by a more compact fusion structure, it can improve the navigation solution accuracy and algorithm robustness.

Considering the above-detailed analysis and research, this paper proposes a novel fusion framework based on model-free and tightly-coupled scheme. The primary contribution of this paper relative to other studies is as follows.

1. A full state mathematical model combined with the low-cost onboard sensor and the estimated state vector are designed by the strapdown inertial navigation system (SINS) [26] and first-order random walk model [27].

2. Aiming at the full state solution of the onboard sensor suite, this paper proposes a multi-sensor, tightly-coupled, and distributed fusion algorithm. Additionally, an effective data time consistency process is designed for the different devices' sample frequencies.

3. Experimental flight tests are performed on the self-developed fixed-wing UAVs platform that verifies the proposed algorithm's performance in the real flight environment.

The rest of this paper is organized as follows. In Section II, the state mathematical model is established. The full state integrated navigation model is proposed in Section III. Then, the experimental test results and discussion in Section IV demonstrate the superiority of the proposed algorithm. Finally, the conclusion is given in Section V.

II. STATE MATHEMATICAL MODEL STATEMENT

As the first step of integrated navigation, the estimated state vector of the mathematical model is established based on the multiple sensors rather than the aircraft dynamic model. The state vector includes the unit quaternion, velocity, position, angular velocity bias, acceleration bias, airspeed, and 2d horizontal wind speed.

The unit quaternion is often used for the fast attitude solution and the rigid body transformation. An arbitrary attitude model can be derived by using a unit quaternion.

$$\boldsymbol{q}(t_k) = \boldsymbol{q}(t_{k-1}) \otimes \boldsymbol{q}(\Delta t) \tag{1}$$

where $\boldsymbol{q}$ consists of 4 elements, and $\boldsymbol{q} = q_0 + q_1\boldsymbol{i} + q_2\boldsymbol{j} + q_3\boldsymbol{k}$. $\boldsymbol{q}(t_k)$ and $\boldsymbol{q}(t_{k-1})$ are the unit quaternion at time t_k and t_{k-1}, respectively. $\boldsymbol{q}(\Delta t)$ is calculated as follows.

$$\boldsymbol{q}(\Delta t) = \sum_{n=0}^{\infty} \frac{(\frac{1}{2}\boldsymbol{\Omega}_\omega \Delta t)^n}{n!} \approx \boldsymbol{I}_{4\times4} + \frac{\boldsymbol{\Omega}_\omega \Delta t}{2} \tag{2}$$

where $\boldsymbol{\Omega}_\omega$ is the anti-symmetric matrix of the three-axis angular velocity $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$ in the body frame, and $\boldsymbol{\omega}$ is modeled by the following equation.

$$\hat{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{n}_\omega + \boldsymbol{b}_\omega \tag{3}$$

where $\hat{\boldsymbol{\omega}}$ is measured angular velocity in the body frame, $\boldsymbol{n}_\omega$ is the gyroscope noise. $\boldsymbol{b}_\omega$ is the gyroscope bias and generally be modeled as the first-order Markov process with the zero-mean Gaussian noise as follow.

$$\boldsymbol{b}_\omega(t_k) = \boldsymbol{b}_\omega(t_{k-1}) exp\frac{-T_s}{T_\omega} + \boldsymbol{n}_{\boldsymbol{b}_\omega(t_k)} \tag{4}$$

where T_ω is the gyroscope correction time, T_s is the sampling time, $\boldsymbol{n}_{\boldsymbol{b}_\omega(t_k)}$ denotes the zero-mean Gaussian noise.

The velocity and position can be derived by the dead-reckoning algorithm [28] in the navigation frame by the equation (5) and (6). It should note that the body frame is "x-y-z" (front-right-down), and the navigation frame is "n-e-d" (north-east-down).

$$\boldsymbol{V}(t_k) = \boldsymbol{V}(t_{k-1}) + \boldsymbol{C}_b^n(t_k)\boldsymbol{a}(t_k)\Delta t + \boldsymbol{g}\Delta t \tag{5}$$

$$\boldsymbol{P}(t_k) = \boldsymbol{P}(t_{k-1}) + \frac{1}{2}(\boldsymbol{V}(t_k) + \boldsymbol{V}(t_{k-1}))\Delta t \tag{6}$$

where $\boldsymbol{V}(t_k)$ and $\boldsymbol{P}(t_k)$ are velocity and position at time t_k, respectively. $\boldsymbol{C}_b^n(t_k)$ is the direction cosine matrix from the body frame to the navigation frame. $\boldsymbol{g}$ is the local gravity vector. $\boldsymbol{a}$ is the acceleration value at time t_k in the body frame, and obtained by the following equation.

$$\hat{\boldsymbol{a}} = (\boldsymbol{a} + \boldsymbol{C}_n^b\boldsymbol{g}) + \boldsymbol{n}_a + \boldsymbol{b}_a \tag{7}$$

where $\hat{\boldsymbol{a}}$ is the three-axis measured acceleration in the body frame, and $\boldsymbol{n}_a$ is the accelerometer noise. $\boldsymbol{b}_a$ is the accelerometer bias and assumed to the first-order Markov process with zero-mean Gaussian noise as follow.

$$\boldsymbol{b}_a(t_k) = \boldsymbol{b}_a(t_{k-1}) exp \frac{-T_s}{T_a} + \boldsymbol{n}_{\boldsymbol{b}_a(t_k)} \tag{8}$$

where T_a is the accelerometer correction time, $\boldsymbol{n}_{\boldsymbol{b}_a(t_k)}$ is also the zero-mean Gaussian noise.

The airspeed and 2d horizontal wind speed are assumed to the random walk process by the equation (9) and (10) based on the previous works [23], [24].

$$\boldsymbol{V}_a(t_k) = \boldsymbol{V}_a(t_{k-1}) + \boldsymbol{n}_{V_a(t_k)} \tag{9}$$

$$\boldsymbol{V}_w(t_k) = \boldsymbol{V}_w(t_{k-1}) + \boldsymbol{n}_{V_w(t_k)} \tag{10}$$

where $\boldsymbol{V}_a(t_k)$ and $\boldsymbol{V}_w(t_k)$ are the airspeed and wind speed, $\boldsymbol{n}_{V_a(t_k)}$ ans $\boldsymbol{n}_{V_w(t_k)}$ are the corresponding Gaussian process noise.

III. FULL STATE INTEGRATED NAVIGATION ESTIMATION

A. System Model Description

Establishing a concrete fusion model for the multi-sensor integrated navigation of UAVs is vital to obtain the full state vector. Considering the nonlinearity of the integrated navigation in the real flight environment, this paper establishes the system model assumed to satisfy the Gaussian process.

$$\begin{cases} \boldsymbol{x}(t_k) = \boldsymbol{f}(\boldsymbol{x}(t_{k-1})) + \boldsymbol{v}(t_{k-1}) \\ \boldsymbol{z}^i(t_k) = \boldsymbol{h}^i(\boldsymbol{x}(t_k)) + \boldsymbol{\eta}^i(t_k) \end{cases} \tag{11}$$

where $\boldsymbol{x}(t_k)$ is the state vector, $\boldsymbol{v}(t_{k-1})$ is the Gaussian process white noise with covariance matrix $\boldsymbol{Q}_{k-1}$. $\boldsymbol{f}(\boldsymbol{x}(t_{k-1}))$ and $\boldsymbol{h}^i(\boldsymbol{x}(t_k))$ are the nonlinear system dynamic and observation equation, respectively. $\boldsymbol{\eta}(t_k)$ is the Gaussian observation white noise with covariance matrix $\boldsymbol{R}_k$. $\boldsymbol{z}(t_k)$ is the observation vector. The i is number of the observation equation of system model.

The vector $\boldsymbol{x}(t_k)$ is designed in this paper with 19-dimensional state denoted by the equation (12).

$$\boldsymbol{x}(t_k) = [\boldsymbol{q}(t_k), \boldsymbol{V}(t_k), \boldsymbol{P}(t_k), \boldsymbol{b}_\omega(t_k), \boldsymbol{b}_a(t_k), \boldsymbol{V}_a(t_k), \boldsymbol{V}_w(t_k)]^{\mathrm{T}} \tag{12}$$

The vector $\boldsymbol{z}^i(t_k)$ is the i-th observed value from the various onboard sensors of the self-developed small UAVs, including the accelerometer, magnetometer, GPS, barometer, and airspeed tube.

$$\boldsymbol{z}^1(t_k) = \hat{\boldsymbol{a}}(t_k) = [a_x(t_k), a_y(t_k), a_z(t_k)]^{\mathrm{T}} \tag{13}$$

$$\boldsymbol{z}^2(t_k) = \hat{\boldsymbol{m}}(t_k) = [m_x(t_k), m_y(t_k), m_z(t_k)]^{\mathrm{T}} \tag{14}$$

$$\boldsymbol{z}^3(t_k) = \psi^{heading}(t_k) = \begin{bmatrix} \psi_{mag}^{heading}(t_k) \\ \psi_{gps}^{heading}(t_k) \end{bmatrix} \tag{15}$$

$$\boldsymbol{z}^4(t_k) = \begin{bmatrix} \boldsymbol{V}^{gps}(t_k) \\ \boldsymbol{P}^{gps}(t_k) \end{bmatrix} = \begin{bmatrix} V_n^{gps}(t_k), V_e^{gps}(t_k), V_d^{gps}(t_k) \\ P_n^{gps}(t_k), P_e^{gps}(t_k), P_d^{gps}(t_k) \end{bmatrix}^{\mathrm{T}} \tag{16}$$

$$\boldsymbol{z}^5(t_k) = P_d^{baro}(t_k) \tag{17}$$

$$\boldsymbol{z}^6(t_k) = V^{air}(t_k) \tag{18}$$

$$\boldsymbol{z}^7(t_k) = \boldsymbol{V}_a(t_k) \tag{19}$$

Sensors	Sample frequency	Observation	Unit
gyroscope	200Hz	$\boldsymbol{\omega}$	rad/s
accelerometer	200Hz	$\boldsymbol{a}$	m/s^2
magnetometer	50Hz	$\boldsymbol{m}$	$gauss$
GPS	10Hz	$\boldsymbol{V}^{gps}, \boldsymbol{P}^{gps}$	$m/s, m$
barometer	50Hz	P_d^{baro}	m
airspeed tube	50Hz	V^{air}	m/s

where $\hat{\boldsymbol{a}}(t_k)$ is three-axis acceleration from the accelerometer, $\hat{\boldsymbol{m}}(t_k)$ is three-axis magnetic field strength, $\psi^{heading}(t_k)$ is the measured heading angle, $\boldsymbol{V}^{gps}(t_k)$ and $\boldsymbol{P}^{gps}(t_k)$ are observed velocity and position from the GPS, $P_d^{baro}(t_k)$ is the observed value of barometer, $V^{air}(t_k)$ is the airspeed observed value, $\boldsymbol{V}_a(t_k)$ is the estimated airspeed. Table I lists the sample frequency and types of observation values for all used low-cost sensors.

B. Observation Model

About the system model, there are divided into the dynamic model and observation model. Among them, the dynamic model $\boldsymbol{f}(\boldsymbol{x}(t_k))$ is the same as the state mathematical model in section II. Thus, this section establishes the observation model from the various used sensors.

1) Accelerometer:

$$\boldsymbol{z}^1(t_k) = \hat{\boldsymbol{a}}(t_k) = \boldsymbol{C}_n^b(t_k)\boldsymbol{g} + \boldsymbol{\eta}^1(t_k) \tag{20}$$

where $\boldsymbol{\eta}^1(t_k)$ is the acceleration observation noise.

2) Magnetometer:

$$\boldsymbol{z}^2(t_k) = \hat{\boldsymbol{m}}(t_k) = \boldsymbol{C}_n^b(t_k) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \boldsymbol{\eta}^2(t_k) \tag{21}$$

where $\boldsymbol{\eta}^2(t_k)$ is the magnetic observation noise.

3) Heading Angle:

$$\boldsymbol{z}^3(t_k) = \psi^{heading}(t_k) = atan(\frac{2(q_0q_3 + q_1q_2)}{1 - 2(q_2^2 + q_3^2)}) + \boldsymbol{\eta}^3(t_k) \tag{22}$$

where $\boldsymbol{\eta}^3(t_k)$ is the heading angle observation noise.

4) GPS:

$$\boldsymbol{z}^4(t_k) = \begin{bmatrix} \boldsymbol{V}^{gps}(t_k) \\ \boldsymbol{P}^{gps}(t_k) \end{bmatrix} = \boldsymbol{h}^4(\boldsymbol{x}(t_k)) + \boldsymbol{\eta}^4(t_k) \tag{23}$$

where $\boldsymbol{\eta}^4(t_k)$ is the GPS observation noise.

5) Barometer:

$$\boldsymbol{z}^5(t_k) = P_d^{baro}(t_k) = \boldsymbol{h}^5(\boldsymbol{x}(t_k)) + \boldsymbol{\eta}^5(t_k) \tag{24}$$

where $\boldsymbol{\eta}^5(t_k)$ is the barometer observation noise.

6) Airspeed Tube:

The relationships of wind triangular about the wind speed $\boldsymbol{V}_w$, ground speed $\boldsymbol{V}$, and airspeed $\boldsymbol{V}_a$ are shown in Fig. 3, which can be formulated as follows.

$$\vec{\boldsymbol{V}}(t_k) = \boldsymbol{C}_b^n(t_k)\vec{\boldsymbol{V}}_a(t_k) + \vec{\boldsymbol{V}}_w(t_k) \tag{25}$$
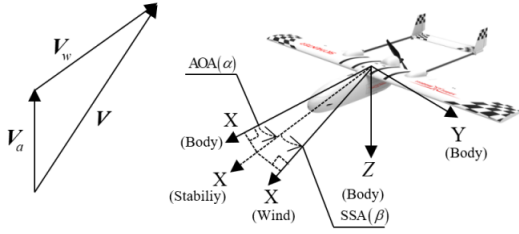
Fig. 3. Wind triangle illustration.

where

$$\vec{V}_a(t_k) = \begin{bmatrix} |\vec{V}_a(t_k)| \cos\alpha \cos\beta \\ |\vec{V}_a(t_k)| \sin\beta \\ |\vec{V}_a(t_k)| \sin\alpha \cos\beta \end{bmatrix}. \tag{26}$$

where α and β are angle of attack and sideslip angle, which can be calculate as follows.

$$\begin{cases} \alpha = tan^{-1}(\dfrac{w}{u}) \\ \beta = sin^{-1}(\dfrac{v}{\sqrt{u^2+v^2}}) \end{cases} \tag{27}$$

where u, v, and w denote the velocity in the body frame, and $[u, v, w]^{\mathrm{T}} = \boldsymbol{C}_n^b \boldsymbol{V} = \boldsymbol{C}_n^b [V_n, V_e, V_d]^{\mathrm{T}}$.

$$z^6(t_k) = V^{air}(t_k) = \boldsymbol{H}^6_{k|k-1}\boldsymbol{x}(t_k) + \boldsymbol{\eta}^6(t_k) \tag{28}$$

$$z^7(t_k) = \boldsymbol{V}_a(t_k) = \sqrt{(\boldsymbol{V}(t_k))^2 - (\boldsymbol{V}_w^2(t_k))^2} + \boldsymbol{\eta}^7(t_k) \tag{29}$$

where $\boldsymbol{\eta}^6(t_k)$ and $\boldsymbol{\eta}^7(t_k)$ are all the airspeed observation noise. The equation (28) is used to correct the measured value (V^{air}) of the airspeed tube to estimate the airspeed ($\boldsymbol{V}_a$), which will be analyzed in the experiment results of Section IV. The equation (29) is to estimate the 2d horizontal wind speed ($\boldsymbol{V}_w$).

C. Multi-Sensor Distributed Fusion Estimation

There are two fusion methods to process the different sensors. One is the filtering estimator, the other is the nonlinear optimization. The nonlinear optimization uses sensor data at all sampling times to calculate the state vector and store all data. Thus, the memory capacity keeps increasing when the sensor data is always collected, resulting in a heavy computation burden for embedded devices. Compared with the nonlinear optimization, the filtering estimator, based on first-order Markov hypothesis, calculates the current state vector $\boldsymbol{x}(t_k)$ depending on the previous state vector $\boldsymbol{x}(t_{k-1})$, and don't consider on historical state from the t_0 to t_{k-1}, it can improve the real-time calculation.

This paper uses the filtering estimator to fuse the mounted sensor, and the estimator has the KF, EKF, UKF, and CKF. The KF is only suitable for the linear system model. If applied to the nonlinear system, the KF filtering solution will be divergent. Although the UKF and CKF can solve the nonlinear problem, their common principle is to use a set of sample points to simulate a nonlinear system, which needs too many points resulting in the calculation burden. However, the traditional EKF is widely used for engineering applications to provide satisfactory solution accuracy and real-time estimation.

Based on the traditional EKF and designed nonlinear integrated navigation model, this paper proposes a distributed multi-sensor EKF (DMSEKF) full state estimation algorithm. Meanwhile, the entire algorithm framework is divided into initialization, full state prediction, and observation correction.

1) Initialization: The setting of initial value $\hat{\boldsymbol{x}}_{0|0}$ and $\hat{\boldsymbol{P}}_{0|0}$ in the filtering algorithm is calculated by the equation (30).

$$\begin{cases} \hat{\boldsymbol{x}}_{0|0} = E(\boldsymbol{x}_{0|0}) \\ \hat{\boldsymbol{P}}_{0|0} = E[(\boldsymbol{x}_{0|0} - \hat{\boldsymbol{x}}_{0|0})(\boldsymbol{x}_{0|0} - \hat{\boldsymbol{x}}_{0|0})^{\mathrm{T}}] \end{cases} \tag{30}$$

where $\hat{\boldsymbol{x}}_{0|0}$ and $\hat{\boldsymbol{P}}_{0|0}$ are the estimated state vector and covariance matrix in initial time, respectively. $\boldsymbol{x}_{0|0}$ is the true state vector value, $E(.)$ denotes the expectation.

2) Full State Prediction: The estimated state vector is calculated by the following equation.

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1|k-1}) \tag{31}$$

where $\boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1|k-1})$ is nonlinear full state integrated navigation dynamic function.

The corresponding state covariance matrix can be obtained by the equation (32).

$$\hat{\boldsymbol{P}}_{k|k-1} = \boldsymbol{F}_{k-1|k-1}\hat{\boldsymbol{P}}_{k-1|k-1}\boldsymbol{F}^{\mathrm{T}}_{k-1|k-1} + \boldsymbol{Q}_{k-1|k-1} \tag{32}$$

where $\boldsymbol{F}_{k|k-1}$ is the state transition matrix and be derived in Appendix I. $\boldsymbol{Q}$ is the process noise matrix with the dimension of 19×19.

3) Sensor Observation Correction: The sample frequencies of the mounted sensor of small fixed-UAVs are shown in Table I, and if all sensor data is processed uniformly, it will cause the solution to diverge result from the sparsity of the observation matrix $\boldsymbol{H}$. Thus, this paper designs a distributed fusion correction according to the difference in sampling frequency.

$$\boldsymbol{K}^i_{k|k-1} = \frac{\hat{\boldsymbol{P}}_{k|k-1}\boldsymbol{H}^{i,\mathrm{T}}_{k|k-1}}{\boldsymbol{H}^i_{k|k-1}\hat{\boldsymbol{P}}_{k|k-1}\boldsymbol{H}^{i,\mathrm{T}}_{k|k-1} + \boldsymbol{R}^i} \tag{33}$$

where $\boldsymbol{K}^i_{k|k-1}$ is the filtering gain, $\boldsymbol{H}^i_{k|k-1}$ is the observation matrix, $\boldsymbol{R}^i$ is the sensor observation matrix. i is the number of the used sensors observed value, and $i = 1, \ldots, 7$. The $\boldsymbol{H}^i_{k|k-1}$ is derived in the following equations.

$$\boldsymbol{H}^1_{k|k-1} = \frac{\partial \boldsymbol{h}^1(\boldsymbol{x}(t_k))}{\partial(\boldsymbol{x}(t_k))} = [\boldsymbol{H}^1_{3\times4} \quad \boldsymbol{0}_{3\times15}] \tag{34}$$

$$\boldsymbol{H}^2_{k|k-1} = \frac{\partial \boldsymbol{h}^2(\boldsymbol{x}(t_k))}{\partial(\boldsymbol{x}(t_k))} = [\boldsymbol{H}^2_{3\times4} \quad \boldsymbol{0}_{3\times15}] \tag{35}$$

$$\boldsymbol{H}^3_{k|k-1} = \frac{\partial \boldsymbol{h}^3(\boldsymbol{x}(t_k))}{\partial(\boldsymbol{x}(t_k))} = [\boldsymbol{H}^3_{1\times4} \quad \boldsymbol{0}_{1\times15}] \tag{36}$$

$$\boldsymbol{H}^4_{k|k-1} = \frac{\partial \boldsymbol{h}^4(\boldsymbol{x}(t_k))}{\partial(\boldsymbol{x}(t_k))} = [\boldsymbol{0}_{6\times4} \quad \boldsymbol{I}_{6\times6} \quad \boldsymbol{0}_{6\times9}] \tag{37}$$

$$\boldsymbol{H}^5_{k|k-1} = \frac{\partial \boldsymbol{h}^5(\boldsymbol{x}(t_k))}{\partial(\boldsymbol{x}(t_k))} = [\boldsymbol{0}_{1\times9} \quad \boldsymbol{I}_{1\times1} \quad \boldsymbol{0}_{1\times9}] \tag{38}$$

$$\boldsymbol{H}^6_{k|k-1} = [\boldsymbol{0}_{1\times16} \quad \boldsymbol{I}_{1\times1} \quad \boldsymbol{0}_{1\times2}] \tag{39}$$

$$\boldsymbol{H}^7_{k|k-1} = \frac{\partial \boldsymbol{h}^7(\boldsymbol{x}(t_k))}{\partial(\boldsymbol{x}(t_k))}$$
$$= [\boldsymbol{0}_{1\times4} \quad \boldsymbol{H}^{7,1}_{1\times3} \quad \boldsymbol{0}_{1\times9} \quad \boldsymbol{0}_{1\times1} \quad \boldsymbol{H}^{7,2}_{1\times2}] \tag{40}$$
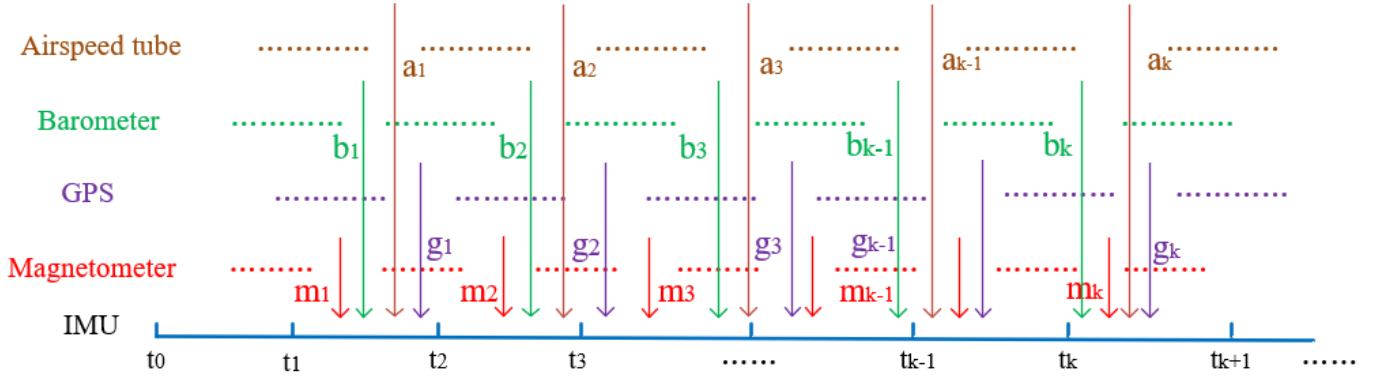
Fig. 4. Multi-sensor data synchronization process.

where $\boldsymbol{H}_{k|k-1}^i$ is the corresponding sensor observation matrix, and the detailed derivation is in Appendix II.

$$\hat{\boldsymbol{x}}_{k|k}^i = \hat{\boldsymbol{x}}_{k|k-1}^i + \boldsymbol{K}_{k|k-1}^i (\boldsymbol{z}^i(t_k) - \boldsymbol{h}^i(\boldsymbol{x}(t_k))) \qquad (41)$$

$$\hat{\boldsymbol{P}}_{k|k}^i = (\boldsymbol{I} - \boldsymbol{K}_{k|k-1}^i \boldsymbol{H}_{k|k-1}^i) \hat{\boldsymbol{P}}_{k|k-1}^i \qquad (42)$$

where $\hat{\boldsymbol{x}}_{k|k}^i$ and $\hat{\boldsymbol{P}}_{k|k}^i$ are the corrected state vector and covariance matrix. The entire algorithm pseudo code is shown in Algorithm 1.

Algorithm 1 The Multi-Sensor Full State Estimation Algorithm

Input: Set initial state vector and covariance matrix: $\hat{\boldsymbol{x}}_{0|0}$, $\hat{\boldsymbol{P}}_{0|0}$;
Output: Estimate state vector and covariance matrix: $\hat{\boldsymbol{x}}_{k|k}$, $\hat{\boldsymbol{P}}_{k|k}$

1: **while** $t > 0$ **do** (t is the algorithm running time)
2:   Calculate the predicted state vector and covariance matrix by the equations (31) and (32), $\boldsymbol{u}_k$ is the IMU input parameters including the angular velocity and acceleration;
3:   $(\hat{\boldsymbol{x}}_{k/k-1}, \hat{\boldsymbol{P}}_{k/k-1}) = \boldsymbol{StatePred}(\hat{\boldsymbol{x}}_{k-1|k-1}, \hat{\boldsymbol{P}}_{k-1|k-1}, \boldsymbol{u}_k)$.
4:   **if** the observed value of the mounted sensor is coming **then**
5:     Calculate the i-th corrected state vector and covariance matrix by the equations (33)-(42), $\boldsymbol{z}_k^i$ is the i-th observed sensor value used in the small fixed-wing UAVs in this paper;
6:   $(\hat{\boldsymbol{x}}_{k|k}^i, \hat{\boldsymbol{P}}_{k|k}^i) = \boldsymbol{ObserUpdate}(\hat{\boldsymbol{x}}_{k/k-1}, \hat{\boldsymbol{P}}_{k/k-1}, \boldsymbol{z}_k^i)$
7:     Obtain the fusion result $\hat{\boldsymbol{x}}_{k|k}$ and $\hat{\boldsymbol{P}}_{k|k}$
8:   **end if**
9: **end while**

---

4) Multi-Sensor Data Time Consistency Process: Multiple sensors data sampling nodes are different, and it is crucial to unify all sensor data on one time loop. Thus, this paper proposes a consistent processing method for multi-sensor data synchronization to obtain an accurate state solution.

The IMU time node is run at a specific frequency and as filtering running the main loop. When the various sensor data is coming, the corresponding time node will be unified into the IMU time node shown in Fig. 4. In this paper, the filtering

Fig. 5. The "SkyHunter" fixed-wing UAV.

fusion time update runs about 200Hz by the IMU sample frequency. When the magnetometer sample time instant is in this time loop between the last and next IMU time instant, the magnetometer measured data is coming into the IMU time update. At this time, the corresponding filtering observation update starts to run as once correction.

Similarly, when the GPS sample time instant is in the IMU time, the GPS filtering observation update runs. When the barometer sample time instant is in the IMU time loop, the barometer filtering observation update runs. When the airspeed sample time instant is in the IMU time loop, the airspeed filtering observation update runs. Then, the entire filtering main loop is running in the task scheduling system of the embedded devices.

IV. EXPERIMENTAL TEST RESULTS AND DISCUSSION
A. Small Fixed-Wing UAVs Platform

This paper designs a self-developed flight platform shown in Fig. 5 to collect the multiple low-cost sensors information and verify various research topics, including sensor fusion state estimation, flight control, and path planning. Meanwhile, the corresponding UAVs specification is summarized in Table II.

Additionally, an onboard self-designed autopilot system shown in Fig. 6 features the Ublox-m8n GPS receiver with 5m position error and a 0.5m/s velocity error, the MS5611 barometer provided the height value, and the HMC5883 magnetometer provided the magnetic field value. Moreover, this paper uses a 6-axis MPU6500 IMU with the ±3% gyro sensitivity error, 0.01 $dps/\sqrt{Hz}$ gyro rate noise, ±3% accel sensitivity error, and 300 $\mu g/\sqrt{Hz}$ accel noise. The airspeed

TABLE II
THE UAVs SPECIFICATION

SkyHunter fixed-wing UAV	Specification
Flying Weight	3.0-3.5 kg
Wing span	1800 mm
Length	1400 mm
Flight time	100 minutes
Li-po battery	5000-10000 mah

TABLE III
THE ATTITUDE ACCURACY COMPARISON OF THE ALGORITHMS

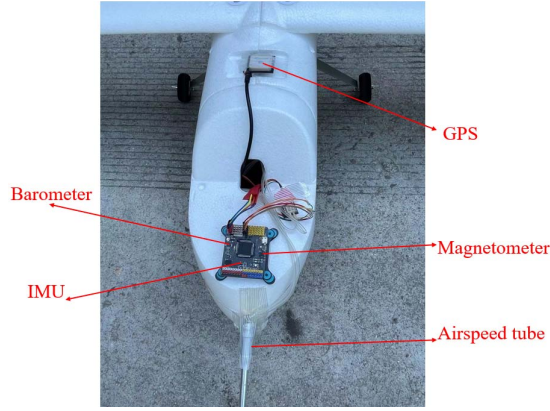| Algorithms	Accuracy (degree)	roll	ptich	yaw
Alg1	MAE	6.012	2.154	2.454
	STD	5.902	2.621	2.845
	RMSE	5.042	2.023	2.483
Alg2	MAE	4.453	1.236	1.831
	STD	4.176	1.481	1.412
	RMSE	4.285	1.804	1.251
Proposed	MAE	2.041	0.831	0.691
	STD	1.901	0.885	0.615
	RMSE	1.891	0.644	0.726

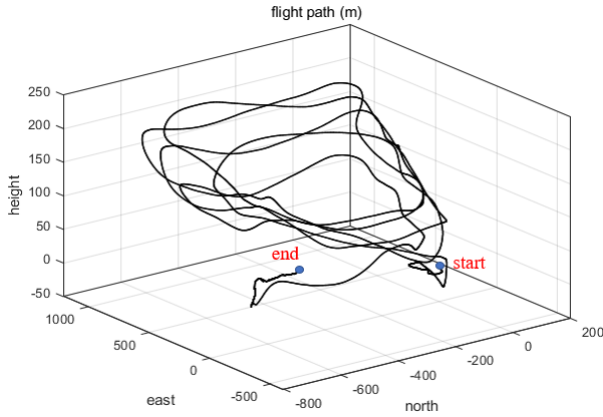Fig. 6. Onboard self-designed autopilot system.

Fig. 7. The flight path.

tube obtains the static and dynamic pressure measurements to calculate the airspeed value.

B. Experiment Result Analysis

The UAVs' flight path is shown in Fig. 7, and the entire flight process includes taking-off, climbing, circling, declining, and landing. Subsequently, this paper designs a two-part comparative experiment to compare the estimator's performance more comprehensively within the full state vector. About the attitude, velocity, and position of the full state vector, the proposed estimator is compared with the conventional algorithm (Alg1) [10] and the multi-sensor fusion algorithm (Alg2) [8]; And about the airspeed and wind speed, the proposed estimator is compared with the model-free algorithm (Alg3) [23] and the model-aided algorithm (Alg4) [18].

Fig. 8 depicts the attitude results of the different algorithms in the roll error, pitch error, and yaw error. It can be found

that the attitude solution accuracy of the proposed algorithm is better than Alg1 and Alg2. This paper uses the multiple measured values from the onboard sensors to increase the system's observability. When the flying speed of the UAVs is relatively low, the observation values of the accelerometer and magnetometer are used to correct the attitude. When the flying speed is high, the observation values of the heading observed values from the magnetometer and GPS begin to be fused into the system framework to improve the robustness of the attitude solution.

To more intuitively analyze the attitude performance between the listed algorithms, this paper introduces three mathematical-statistical accuracy methods, including the mean absolute errors (MAE), standard deviation (STD), and root mean square errors (RMSE) in Table III. It can be seen that the MAE, STD, and RMSE of the proposed algorithm are smaller than Alg1 and Alg2. In addition, the roll accuracy of the proposed algorithm is about 1.944 degree, the pitch about 0.793 degree, the yaw about 0.677 degree. The reason for the poor roll accuracy is many circling maneuvers during the UAVs flight shown in Fig. 7, the harmful acceleration caused by circling maneuvers degrades the quality of the UAVs' lateral control, which further affects the roll calculation.

Fig. 9 depicts the velocity results of the different algorithms in the V_n error, V_e error, and V_d error. The characteristics of the state mathematical model in Section II determine that the attitude error propagates into the velocity solution and influences the accuracy of velocity estimation. It needs to be noted that the smaller the velocity error, the more accurate the velocity calculation. Further, reliable velocity accuracy is useful to long-term outer-loop control of the UAVs. Compared with the other two algorithms, this paper has a better performance attributed to the GPS and airspeed tube observation model to correct the velocity bias. Table IV shows the velocity accuracy for all the algorithms. It can be found that the V_n accuracy of the proposed algorithm is about $0.026m/s$, the V_e accuracy about $0.038m/s$, the V_d accuracy about $0.498m/s$, which is higher than the GPS velocity accuracy ($0.5m/s$). Thus, the fusion-based solution is better than the single sensor, especially in a complex and unknown environment.

Fig. 10 depicts the position results of the different algorithms in the P_n error, P_e error, and P_d error. It can be found that Alg1 and Alg2 only relying on GPS measurements is not ideal, resulting in poor position error when UAVs are in wind disturbance. By contrast, this paper takes the
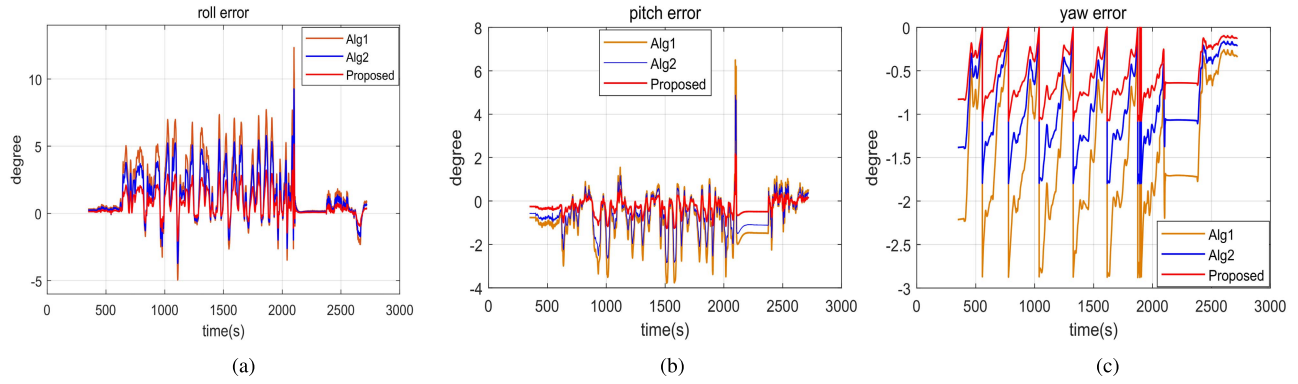
Fig. 8. The attitude results of roll error, pitch error, and yaw error.
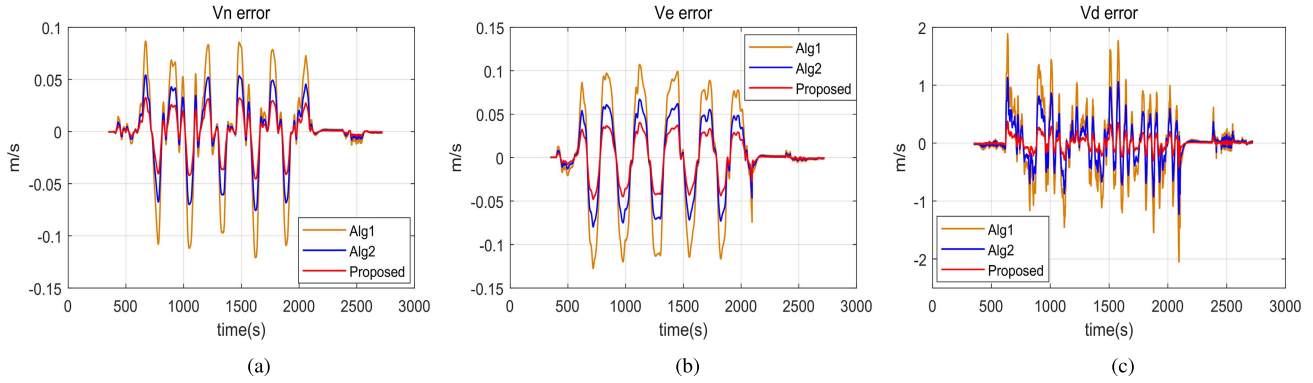

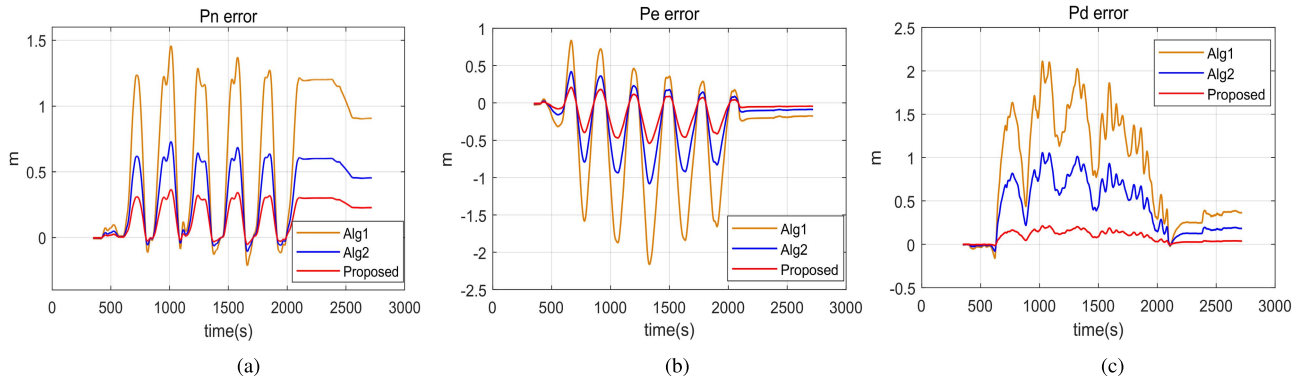Fig. 9. The velocity results of V_n error, V_e error, and V_d error.

Fig. 10. The position results of P_n error, P_e error, and P_d error.

airspeed tube as an observation model to estimate the wind speed improving the position accuracy in wind disturbance. In addition to the GPS observation model, the barometer is also introduced into the system framework of the proposed algorithm to increase the height calculation accuracy shown in Fig. 10(c). Table V shows the position accuracy for all the algorithms. For the proposed algorithm, the P_n accuracy is about $0.351m$, and the P_e accuracy is about $0.419m$, the P_d accuracy is about $0.306m$, which is also higher than the single-point GPS position accuracy ($5m$). The position is also useful for outer loop control in some flight tasks, including path planning, avoidance, and circling operation.

The raw airspeed value (V^{air}) from the airspeed tube has a large deviation and cannot be used in the filtering fusion. Thus, this paper takes the airspeed as an estimated state ($\boldsymbol{V}_a$) different from other studies. Fig. 11 shows that the comparison experiment is performed between the estimated airspeed in different filters and raw airspeed. The R of Fig. 11 is the airspeed observation noise covariance matrix, and the setting of R affects the filtering performance of estimated airspeed. When the R is 1 or bigger than 1 shown in Fig. 11(a), the solution about the estimated airspeed is failing. If the R is 0.5 shown in Fig. 11(b) or 0.05 shown in Fig. 11(c), the estimated airspeed has a certain time lag compared with the
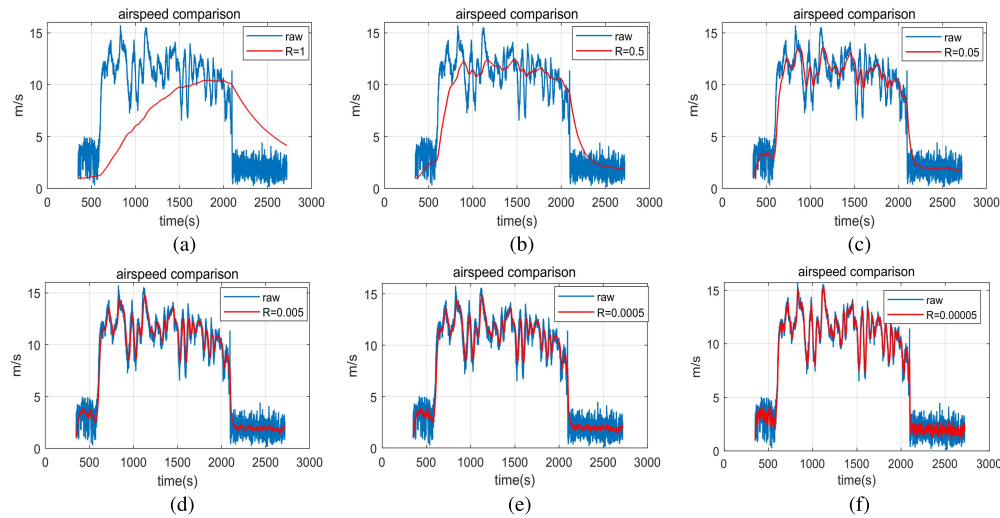
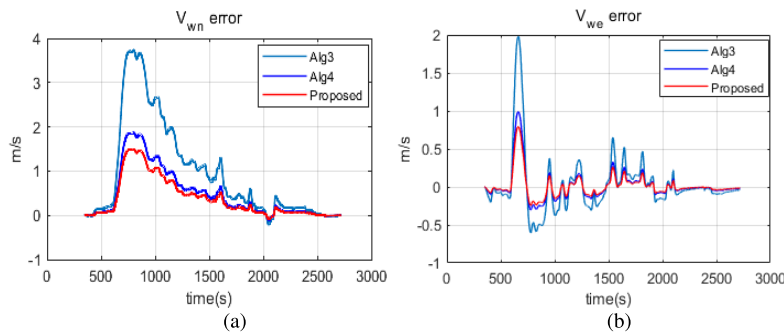Fig. 11. The airspeed comparison in different filters.

Fig. 12. The 2d horizontal wind speed error comparison.

TABLE IV
THE VELOCITY ACCURACY COMPARISON OF THE ALGORITHMS

Algorithms	Accuracy (m/s)	V_n	V_e	V_d
Alg1	MAE	0.082	0.100	1.302
	STD	0.073	0.092	0.939
	RMSE	0.088	0.089	0.901
Alg2	MAE	0.056	0.782	0.841
	STD	0.045	0.049	0.830
	RMSE	0.051	0.051	0.816
Proposed	MAE	0.022	0.029	0.532
	STD	0.026	0.025	0.491
	RMSE	0.030	0.022	0.473

TABLE V
THE POSITION ACCURACY COMPARISON OF THE ALGORITHMS

Algorithms	Accuracy (m)	P_n	P_e	P_d
Alg1	MAE	1.503	1.602	2.104
	STD	1.421	1.549	1.981
	RMSE	1.418	1.572	2.006
Alg2	MAE	0.553	0.634	1.023
	STD	0.601	0.625	0.951
	RMSE	0.525	0.596	1.155
Proposed	MAE	0.381	0.381	0.325
	STD	0.312	0.425	0.294
	RMSE	0.361	0.451	0.301

raw airspeed. The setting about the R is 0.005 shown in Fig. 11(d) or 0.00005 shown in Fig. 11(e) has a good filtering solution and fits the raw airspeed well. Although the R is 0.000005 shown in Fig. 11(f) can better track the raw airspeed compared with the others, the suppression to sensor noise is poor, resulting in unsatisfying filtering estimation. Thus, the appropriate parameter setting about the airspeed observation noise covariance matrix affects the real-time performance of filter estimation and greatly influences sensor noise processing. In addition, a well-estimated airspeed is also very important for calculating wind speed.

Fig. 12 shows the estimated 2d horizontal wind speed in listed algorithms. Alg3 uses the airspeed tube and GPS module

TABLE VI
THE CALCULATION COST OF ALG3, ALG4, AND PROPOSED

Algorithms	Alg3	Alg4	Proposed
Time(mesc)	5.213	13.51	7.004

to estimate the 2d wind speed based on the EKF and not solve the other state vector. Thus, the state dimension of Alg3 is smaller than the proposed algorithm in this paper, which means that the calculation cost of Alg3 is lower than the proposed algorithm. However, Fig. 12 denotes that the wind speed accuracy of Alg3 is worse compared with the other two algorithms. In addition, Alg4, based on the aerodynamic
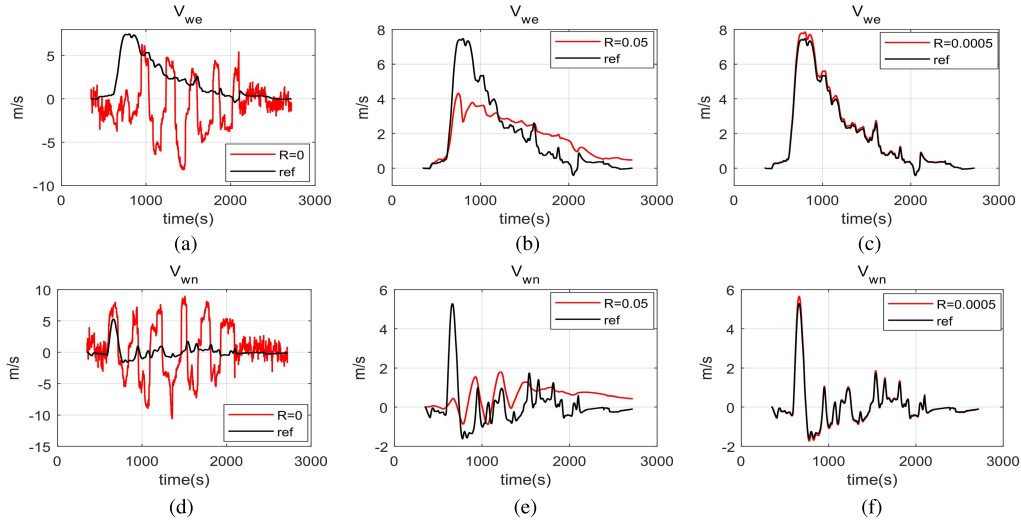
Fig. 13. The wind speed comparison.

model-aided method, is better than Alg3 in solution accuracy, and is about the same as the proposed algorithm. Nevertheless, the calculation cost of Alg4 is the largest shown in Table VI. Although the aerodynamic model-aided method can improve accuracy, it is difficult to balance between accuracy and calculation cost. In addition, Alg4 is very sensitive to the forces and moments of the UAV aerodynamic model. Once the UAV type is changed, the model needs to be redesigned and lacks robustness.

Fig. 13 shows that the 2d wind speed comparison when the setting parameter of airspeed observation noise covariance matrix (R) is different. It has been verified that the appropriate setting parameter (R) can obtain the excellent airspeed, and then the reliable wind speed can be estimated by the equation (29). Thus, it can be found that the accuracy of estimated airspeed is also vital to obtain the 2d wind speed.

V. CONCLUSION

The paper presents a novel model-free multi-sensor full state estimation of small fixed-wing UAVs for the attitude, velocity, position, airspeed, and 2d horizontal wind speed. Specifically, a multi-sensor mathematical model is established to obtain the full state depending on the IMU, magnetometer, barometer, GPS, and airspeed tube. Then, a distributed tightly-coupled fusion algorithm based on the model-free method and EKF is proposed, and the multiple observation models are designed from the various sensor measured values. Meanwhile, to synchronize the time node of all used sensors in the main loop, this paper designs a data processing method to improve the consistency of the sensor's sampling frequency. The detailed experiments are performed to verify the proposed algorithm's superiority and provide a reliable and long-term convergent full state vector estimation in wind disturbance. Although the proposed algorithm is mainly applied to the small UAVs, it can also be used in other robots' localization and state estimation.

APPENDIX I
FULL STATE TRANSITION MATRIX

The full state transition matrix is listed as in the following equations. (43) and (44), as shown at the top of the next page.

APPENDIX II
SENSOR OBSERVATION MATRIX

The all sensor observation matrix $\boldsymbol{H}^i_{k|k-1}(i = 1, \ldots\ldots, 7)$ is listed as follows.

$$\boldsymbol{H}^1_{k|k-1} = \frac{\partial \boldsymbol{h}^1(\boldsymbol{x}(t_k))}{\partial(\boldsymbol{x}(t_k))} = [\boldsymbol{H}^1_{3\times4} \quad \boldsymbol{0}_{3\times15}] \tag{45}$$

$$\boldsymbol{H}^1_{3\times4} = g \begin{bmatrix} 2q2 & -2q3 & 2q0 & -2q1 \\ -2q1 & -2q0 & -2q3 & 2q2 \\ -2q0 & 2q1 & 2q2 & -2q3 \end{bmatrix} \tag{46}$$

where the g is the component of gravitational acceleration $\boldsymbol{g} = [0, 0, g]^{\mathrm{T}}$, and g is about 9.80665 m/s^2.

$$\boldsymbol{H}^2_{k|k-1} = \frac{\partial \boldsymbol{h}^2(\boldsymbol{x}(t_k))}{\partial(\boldsymbol{x}(t_k))} = [\boldsymbol{H}^2_{3\times4} \quad \boldsymbol{0}_{3\times15}] \tag{47}$$

$$\boldsymbol{H}^2_{3\times4} = 2 \begin{bmatrix} q0 & q1 & -q2 & -q3 \\ q3 & q2 & q1 & q0 \\ -q3 & q2 & q1 & q0 \end{bmatrix} \tag{48}$$

$$\boldsymbol{H}^3_{k|k-1} = \frac{\partial \boldsymbol{h}^3(\boldsymbol{x}(t_k))}{\partial(\boldsymbol{x}(t_k))} = [\boldsymbol{H}^3_{1\times4} \quad \boldsymbol{0}_{3\times15}] \tag{49}$$

$$\boldsymbol{H}^3_{1\times4} = \begin{bmatrix} H^2_1 & H^2_2 & H^2_3 & H^2_4 \end{bmatrix} \tag{50}$$

$$H^2_1 = -(2q3)/(((2q0q3 + 2q1q2)^2/(2q2^2 + 2q3^2 - 1)^2 + 1) * (2q2^2 + 2q3^2 - 1)) \tag{51}$$

$$H^2_2 = -(2q2)/((((2q0q3 + 2q1q2)^2/(2q2^2 + 2q3^2 - 1)^2 + 1) * (2q2^2 + 2q3^2 - 1)) \tag{52}$$

$$H^2_3 = -((2q1)/(2q2^2 + 2q3^2 - 1) - (4q2(2q0q3 + 2q1q2))/(2q2^2 + 2q3^2 - 1)^2)/((2q0q3 + 2q1q2)^2/(2q2^2 + 2q3^2 - 1)^2 + 1) \tag{53}$$

$$H^2_4 = -((2q0)/(2q2^2 + 2q3^2 - 1) - (4q3(2q0q3 + 2q1q2))/(2q2^2 + 2q3^2 - 1)^2)/((2q0q3$$

$$\boldsymbol{F}_{k-1|k-1} = \frac{\partial \boldsymbol{f}_{k-1|k-1}}{\partial \boldsymbol{x}_{k-1|k-1}} = [\frac{\partial \boldsymbol{f}_{k-1|k-1}}{\partial \boldsymbol{q}_{k-1}}, \frac{\partial \boldsymbol{f}_{k-1|k-1}}{\partial \boldsymbol{V}_{k-1}}, \frac{\partial \boldsymbol{f}_{k-1|k-1}}{\partial \boldsymbol{P}_{k-1}}, \frac{\partial \boldsymbol{f}_{k-1|k-1}}{\partial \boldsymbol{b}_{\omega,k-1}}, \frac{\partial \boldsymbol{f}_{k-1|k-1}}{\partial \boldsymbol{b}_{a,k-1}}, \frac{\partial \boldsymbol{f}_{k-1|k-1}}{\partial \boldsymbol{V}_{a,k-1}}, \frac{\partial \boldsymbol{f}_{k-1|k-1}}{\partial \boldsymbol{V}_{w,k-1}}] \quad (43)$$

$$= \begin{bmatrix} (\boldsymbol{I}_{4\times4} + \frac{\boldsymbol{\Omega}_{\omega}\Delta t}{2})_{4\times4} & \boldsymbol{0}_{4\times3} & \boldsymbol{0}_{4\times3} & \boldsymbol{0}_{4\times3} & \boldsymbol{0}_{4\times3} & \boldsymbol{0}_{4\times1} & \boldsymbol{0}_{4\times2} \\ \boldsymbol{F}^1_{3\times4} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\boldsymbol{C}^n_b\Delta t & \boldsymbol{0}_{3\times1} & \boldsymbol{0}_{3\times2} \\ \boldsymbol{0}_{3\times4} & \Delta t\boldsymbol{I}_{3\times3} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times1} & \boldsymbol{0}_{3\times2} \\ \boldsymbol{0}_{3\times4} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & exp\frac{-T_s}{T_{\omega}}\boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times1} & \boldsymbol{0}_{3\times2} \\ \boldsymbol{0}_{3\times4} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & exp\frac{-T_s}{T_a}\boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times1} & \boldsymbol{0}_{3\times2} \\ \boldsymbol{0}_{1\times4} & \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} & \boldsymbol{I}_{1\times1} & \boldsymbol{0}_{1\times2} \\ \boldsymbol{0}_{2\times4} & \boldsymbol{0}_{2\times3} & \boldsymbol{0}_{2\times3} & \boldsymbol{0}_{2\times3} & \boldsymbol{0}_{2\times3} & \boldsymbol{0}_{2\times1} & \boldsymbol{I}_{2\times2} \end{bmatrix} \quad (44)$$

$$\boldsymbol{F}^1_{3\times4} = \begin{bmatrix} (2a_xq_0 - 2a_yq_3 + 2a_zq_3) & (2a_xq_1 + 2a_yq_2 + 2a_zq_2) & (2a_yq_1 - 2a_xq_2 + 2a_zq_1) & -(2a_xq_3 + 2a_yq_0 - 2a_zq_0) \\ (2a_xq_3 + 2a_yq_0 - 2a_zq_1) & -2a_yq_1 - 2a_xq_2 + 2a_zq_0) & (2a_xq_1 + 2a_yq_2 + 2a_zq_3) & (2a_xq_0 - 2a_yq_3 + 2a_zq_2) \\ (2a_yq_1 - 2a_xq_2 + 2a_zq_0) & (2a_xq_3 + 2a_yq_0 - 2a_zq_1) & -(2a_xq_0 - 2a_yq_3 + 2a_zq_2) & (2a_xq_1 + 2a_yq_2 + 2a_zq_3) \end{bmatrix}$$

$$+ 2q_1q_2)^2/(2q_2^2 + 2q_3^2 - 1)^2 + 1) \quad (54)$$

$$\boldsymbol{H}^7_{k|k-1} = \frac{\partial \boldsymbol{h}^7(\boldsymbol{x}(t_k))}{\partial(\boldsymbol{x}(t_k))}$$

$$= [\boldsymbol{0}_{1\times4} \quad \boldsymbol{H}^{7,1}_{3\times3} \quad \boldsymbol{0}_{2\times9} \quad \boldsymbol{0}_{2\times1} \quad \boldsymbol{H}^{7,2}_{2\times2}] \quad (55)$$

$$\boldsymbol{H}^{7,1}_{3\times3} = \begin{bmatrix} V_n/\sqrt{(V_n^2 + V_{w,n}^2)} & 0 & 0 \\ 0 & V_e/\sqrt{(V_e^2 + V_{w,e}^2)} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (56)$$

$$\boldsymbol{H}^{7,2}_{2\times2} = \begin{bmatrix} V_{w,n}/\sqrt{(V_n^2 + V_{w,n}^2)} & 0 \\ 0 & V_{w,e}/\sqrt{(V_e^2 + V_{w,e}^2)} \end{bmatrix} \quad (57)$$

REFERENCES

[1] M. Rossi and D. Brunelli, "Autonomous gas detection and mapping with unmanned aerial vehicles," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 4, pp. 765–775, Apr. 2016.

[2] L. Yang, Q. Sun, and Z.-S. Ye, "Designing mission abort strategies based on early-warning information: Application to UAV," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 277–287, Jan. 2020.

[3] I. U. Khan, I. M. Qureshi, M. A. Aziz, T. A. Cheema, and S. B. H. Shah, "Smart IoT control-based nature inspired energy efficient routing protocol for flying ad hoc network (FANET)," *IEEE Access*, vol. 8, pp. 56371–56378, 2020.

[4] M. Coombes, T. Fletcher, W.-H. Chen, and C. Liu, "Decomposition-based mission planning for fixed-wing UAVs surveying in wind," *J. Field Robot.*, vol. 37, no. 3, pp. 440–465, Dec. 2020.

[5] A. Wenz and T. A. Johansen, "Moving horizon estimation of air data parameters for UAVs," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 3, pp. 2101–2121, Jun. 2020.

[6] P. Tian, H. Chao, M. Rhudy, J. Gross, and H. Wu, "Wind sensing and estimation using small fixed-wing unmanned aerial vehicles: A survey," *J. Aerosp. Inf. Syst.*, vol. 18, no. 3, pp. 132–143, Mar. 2021.

[7] K. Peng, G. Cai, B. M. Chen, M. Dong, K. Y. Lum, and T. H. Lee, "Design and implementation of an autonomous flight control law for a UAV helicopter," *Automatica*, vol. 45, no. 10, pp. 2333–2338, Oct. 2009.

[8] J. Peng, P. Zhang, L. Zheng, and J. Tan, "UAV positioning based on multi-sensor fusion," *IEEE Access*, vol. 8, pp. 34455–34467, 2020.

[9] X. Ding, Z. Wang, L. Zhang, and C. Wang, "Longitudinal vehicle speed estimation for four-wheel-independently-actuated electric vehicles based on multi-sensor fusion," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12797–12806, Nov. 2020.

[10] Y. Yang, X. X. Liu, W. G. Zhang, X. X. Liu, and Y. C. Guo, "Multi-level variable dimension extended Kalman filter algorithm for UAV," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2019, pp. 1615–1620.

[11] H. Nourmohammadi and J. Keighobadi, "Fuzzy adaptive integration scheme for low-cost SINS/GPS navigation system," *Mech. Syst. Signal Process.*, vol. 99, pp. 434–449, Jan. 2018.

[12] R. Kottath, P. Narkhede, V. Kumar, V. Karar, and S. Poddar, "Multiple model adaptive complementary filter for attitude estimation," *Aerosp. Sci. Technol.*, vol. 69, pp. 574–581, Oct. 2017.

[13] A. S. Al-Fahoum and M. S. Abadir, "Design of a modified Madgwick filter for quaternion-based orientation estimation using AHRS," *Int. J. Comput. Electr. Eng.*, vol. 10, no. 3, pp. 174–186, 2018.

[14] X. Zhou, X. Xu, Y. Yao, and H. Zhao, "A robust quaternion Kalman filter method for MIMU/GPS in-motion alignment," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, 2021.

[15] J. Li, N. Song, G. Yang, M. Li, and Q. Cai, "Improving positioning accuracy of vehicular navigation system during GPS outages utilizing ensemble learning algorithm," *Inf. Fusion*, vol. 35, pp. 1–10, May 2017.

[16] L. Qian and H. H. T. Liu, "Path-following control of a quadrotor UAV with a cable-suspended payload under wind disturbances," *IEEE Trans. Ind. Electron.*, vol. 67, no. 3, pp. 2021–2029, Mar. 2020.

[17] M. B. Rhudy, Y. Gu, J. N. Gross, and H. Chao, "Onboard wind velocity estimation comparison for unmanned aircraft systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 1, pp. 55–66, Feb. 2017.

[18] W. Youn, H. Choi, A. Cho, S. Kim, and M. B. Rhudy, "Aerodynamic model-aided estimation of attitude, 3-D wind, airspeed, AOA, and SSA for high-altitude long-endurance UAV," *IEEE Sensors J.*, vol. 20, no. 14, pp. 7929–7940, Jul. 2020.

[19] W. Youn, H. S. Choi, H. Ryu, S. Kim, and M. B. Rhudy, "Model-aided state estimation of HALE UAV with synthetic AOA/SSA for analytical redundancy," *IEEE Trans. Aeros. Elec. Syst.*, vol. 56, no. 6, pp. 4300–4314, Dec. 2020.

[20] M. B. Rhudy, M. L. Fravolini, Y. Gu, M. R. Napolitano, S. Gururajan, and H. Chao, "Aircraft model-independent airspeed estimation without Pitot tube measurements," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 3, pp. 1980–1995, Jul. 2015.

[21] K. Sun, C. D. Regan, and D. Gebre-Egziabher, "Observability and performance analysis of a model-free synthetic air data estimator," *J. Aircr.*, vol. 56, no. 4, pp. 1471–1486, Jul. 2019.

[22] M. L. Fravolini, G. D. Core, U. Papa, P. Valigi, and M. R. Napolitano, "Data-driven schemes for robust fault detection of air data system sensors," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 1, pp. 234–248, Jan. 2019.

[23] A. Cho, J. Kim, S. Lee, and C. Kee, "Wind estimation and airspeed calibration using a UAV with a single-antenna GPS receiver and Pitot tube," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 1, pp. 109–117, Jan. 2011.

[24] M. Brossard, J.-P. Condomines, and S. Bonnabel, "Tightly coupled navigation and wind estimation for mini UAVs," in *Proc. AIAA Guid., Navigat., Control Conf.*, Jan. 2018, p. 1843.

[25] J. W. Langelaan, N. Alley, and J. Neidhoefer, "Wind field estimation for small unmanned aerial vehicles," *J. Guid., Control, Dyn.*, vol. 34, no. 4, pp. 1016–1030, 2011.

[26] Y. Liang and Y. Jia, "A nonlinear quaternion-based fault-tolerant SINS/GNSS integrated navigation method for autonomous UAVs," *Aerosp. Sci. Technol.*, vol. 40, pp. 191–199, Jan. 2015.

[27] M. B. Rhudy, M. L. Fravolini, M. Porcacchia, and M. R. Napolitano, "Comparison of wind speed models within a pitot-free airspeed estimation algorithm using light aviation data," *Aerosp. Sci. Technol.*, vol. 86, pp. 21–29, Mar. 2019.

[28] H. Mokhtarzadeh and D. Gebre-Egziabher, "Performance of networked dead reckoning navigation system," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 5, pp. 2539–2553, Oct. 2016.

Xuhang Liu received the B.S. and M.S. degrees from the School of Automation, Northwestern Polytechnical University, Xi'an, China, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree in control science and engineering. His current research interests include nonlinear filtering, sensor fusion, and UAV navigation.

Yue Yang (Student Member, IEEE) received the M.S. degree from the School of Automation, Northwestern Polytechnical University, Xi'an, China, in 2018, where he is currently pursuing the Ph.D. degree in control science and engineering. His current research interests include vision navigation, multi-sensor fusion, state estimation, SLAM, and UAV integrated navigation.

Yicong Guo received the B.S. degree from the School of Automation, Northwestern Polytechnical University, Xi'an, China, in 2017, where he is currently pursuing the Ph.D. degree in control science and engineering. His current research interests include multi UAV cooperative control, guidance, path planning, and UAV navigation.

Xiaoxiong Liu received the bachelor's, master's, and Ph.D. degrees from Northwestern Polytechnical University. He is currently an Associate Professor with the School of Automation, Northwestern Polytechnical University. His research interests include aircraft flight control technology, UAV autonomous control technology, guidance and navigation technology, visual location guidance and navigation technology, fault diagnosis, and fault tolerance flight control technology.

Weiguo Zhang was a Visiting Researcher at the University of Salford, Salford, U.K., from 1988 to 1991. He is currently a Professor with the School of Automation, Northwestern Polytechnical University, China. His research interests include navigation, guidance and control, intelligent flight control, and fault-tolerant control.