

# Identifying Aerodynamics of Small Fixed-Wing Drones Using Inertial Measurements for Model-Based Navigation

Aman Sharma | Gabriel François Laupré | Jan Skaloud

Geodetic Engineering Laboratory (TOPO),  
École Polytechnique Fédérale de Lausanne  
(EPFL), Lausanne, Switzerland

## Correspondence

Aman Sharma  
EPFL ENAC IIE TOPO, Station 18,  
CH-1015 Lausanne.  
Email: [aman.sharma@epfl.ch](mailto:aman.sharma@epfl.ch)

## Abstract

The success of drone missions is incumbent on an accurate determination of the drone pose and velocity, which are collectively estimated by fusing inertial measurement unit and global navigation satellite system (GNSS) measurements. However, during a GNSS outage, the long-term accuracy of these estimations are far from allowing practical use. In contrast, vehicle dynamic model (VDM)-based navigation has demonstrated significant improvement in autonomous positioning during GNSS outages. This improvement is achieved by incorporating mathematical models of aerodynamic forces/moments in the sensor fusion architecture. Such an approach, however, relies on a knowledge of aerodynamic model parameters, specific to the operating vehicle. We present a novel calibration algorithm to identify these parameters from the flight data of two geometrically different drones. The identified parameters, when used in the VDM framework, show a significant reduction in navigation drift during GNSS outages. Moreover, the obtained results show that the proposed algorithm is independent of the choice of fixed-wing platform and prior knowledge of aerodynamics.

## Keywords

extended Kalman filter, observability Gramian, partial update, recursive least squares, Schmidt–Kalman filter

## 1 | INTRODUCTION

In the past, drones have been predominantly used by the military to ascertain several safeguards pertinent to national security. However, the past two decades have experienced a dramatic technological boom in the fields of robotics, computer vision, and deep learning, which has consequently opened the doors to innovation in the way drones are being utilized today and their plausible usage in the future. Non-military applications of drones include the fields of healthcare (Hiebert et al., 2020), cinematography (Alcántara et al., 2020), geographic mapping (Cledat et al., 2020), disaster management (Erdelj et al., 2017), precision agriculture (Puri et al., 2017), search and rescue (Hayat et al., 2020), weather forecast (Leuenberger et al., 2020), wildlife monitoring (Hodgson et al., 2018), entertainment (Kim et al., 2018), and more. In this article, we envisage addressing the notion of navigation safety

(in terms of pursuing a programmed path with better tolerance to the absence of a global navigation satellite system [GNSS]) using fixed-wing drones without dependence on illumination conditions (i.e., a vision system), as may be required for certain beyond-visual-line-of-sight missions.

The majority of aerial navigation systems are based on the integration of an inertial navigation system (INS) and GNSS (Bryson & Sukkarieh, 2015; Nonami et al., 2010). GNSS provides position/velocity/time data at a low frequency, whereas INS facilitates position/velocity/attitude (PVA) data at a high frequency. The fusion of data outputs from these two systems provides a navigation solution (position/velocity/attitude/time [PVAT]) with sufficient short-term and long-term accuracy. However, during a GNSS outage, this solution is based solely on INS in the horizontal direction, while the vertical channel can be bounded by a barometer. The accuracy of such a solution, based on dead reckoning, is predominantly dictated by the quality of the inertial measurement unit (IMU). For small drones, the long-term accuracy of this solution is so low that the position uncertainty after 1 min of GNSS outage is far from allowing practical utility.

Vehicle dynamic model (VDM)-based navigation systems (Khaghani & Skaloud, 2018; Laupré et al., 2021) have demonstrated significant mitigation of drift in autonomous positioning during GNSS outages. This mitigation is achieved by incorporating mathematical models of aerodynamic forces and moments (experienced by the drone) in the sensor fusion framework. These models play a key role in implicitly rejecting physically impossible movements suggested by the IMU (Khaghani & Skaloud, 2016, 2018). The aerodynamic model proposed in Ducard (2009) has been widely used in recent works (Khaghani & Skaloud, 2018; Laupré et al., 2019; Laupré & Skaloud, 2020, 2021; Mwenegoha et al., 2019a, 2019b, 2020, 2021) involving VDM for a fixed-wing drone, whereas Laupré et al. (2021) proposed an aerodynamic model for a delta-wing platform. We make use of these same functional models to evaluate the performance of our calibration strategy in a comparable setting. We refer the readers to Sendobry (2014), Cork (2014), and Khaghani (2018) for an extensive review of the usage of VDM in the domain of navigation.

The aforementioned aerodynamic models contain certain constant—yet unknown—parameters that are specific to the operating platform. Development of a (flight) data-driven methodology for estimating these unknown parameters is an attractive and inexpensive alternative to computational fluid dynamics (CFD) and wind tunnel testing (Dsouza et al., 2016; Schwithal et al., 2016; Wisnoe et al., 2009) and motivates further exploration of the potential of such an alternative. The existing flight-data-based method, as detailed in Laupré & Skaloud (2021) and Khaghani & Skaloud (2018), makes use of an extended Kalman filter (EKF) to simultaneously estimate all of these parameters in addition to the wind and navigation states as part of a state-space framework. This approach has two major drawbacks: i) the auxiliary states representing the model parameters require a good initial guess to prevent filter divergence (due to nonlinearity and observability) and ii) the simultaneous estimation causes some of the states to compensate for others (Laupré & Skaloud, 2021). Existing works do not detail how these parameters are initialized. It seems that some prior knowledge, or a guess, is indeed used to find the first set of working parameters and/or zero-wind conditions are expected. These parameters, based on prior knowledge, are later refined and tuned using different methodologies presented in Khaghani (2018), Laupré & Skaloud (2021), and Laupré et al. (2021) for obtaining better results.

In Khaghani & Skaloud (2018), the authors argue that for the purpose of navigation, in-flight estimation of parameters is sufficient to improve autonomous positioning compared with inertial coasting while minimizing design effort. However, they also assert that these parameters are practically constant and that they can be

better estimated in a calibration scenario. In Laupré & Skaloud (2021), a calibration scenario is created by utilizing post-processed kinematic (PPK) GNSS data as compared with standalone data. Additionally, precise observations of absolute attitude are incorporated using an external camera and photogrammetry. The obtained parameters from the calibration scenario are then used to initialize the EKF in a test flight executing the VDM-based navigation algorithm (Khaghani & Skaloud, 2018), yielding better results.

The above-described existing approaches to aerodynamic calibration have contributed significantly toward improving the quality of model-based navigation. However, these approaches rely heavily on *a priori* estimates of model parameters and do not provide a definite answer as to how to initialize these parameters for a new platform. Moreover, the reliance of these approaches on a state-space framework that simultaneously estimates the model parameters (alongside navigation states, wind, sensor biases, etc.) causes the different elements of the state vector to compensate for each other (Laupré & Skaloud, 2021). We view this simultaneous estimation structure as a conjunction of nuisance states (constant model parameters and time-varying wind) and states of importance (navigation states). In Brink (2017), the authors point out that nuisance states are usually only mildly observable, and estimating them in the traditional sense may often cause filter divergence. The authors also question the existence of sufficient unique information in a given set of measurements such that the update of a nuisance state is not correlated to any other state. The existing VDM-based navigation methodology is composed of a multitude of states (47 or more), making it challenging to analyze the uniqueness of the information brought in by each measurement. Therefore, we propose a methodology to first decouple the estimation of the wind velocity and the parameters associated with aerodynamic moments and forces. Secondly, we analyze the uniqueness of information associated with each measurement by developing a heuristic approach based on the observability Gramian (Chen, 1999). Thereafter, we utilize a Schmidt–Kalman filter framework to update most observable parameters within nuisance states to obtain a sufficiently *good a priori* estimate of model parameters. Finally, we validate the proposed methodology using the VDM-based navigation framework, wherein we use the calibrated model parameters to either i) initialize the EKF and fine-tune the parameters in-flight or ii) keep the parameters fixed and not estimate them in-flight while adding a new (scale) state to compensate for different weather conditions. The proposed method is found to be independent of the choice of drone and the shape of its wings, allowing for a set of aerodynamic coefficients to be estimated with limited prior knowledge of the platform.

The remainder of the article is organized as follows: Section 2 discusses the relevant mathematical background for a general understanding of the VDM. Section 3 is dedicated to the proposed decoupled aerodynamic calibration algorithm. Then, Section 4 details the experimental setup, followed by the obtained results and a discussion in Section 5 and Section 6, respectively. Finally, we conclude our findings in Section 7.

## 2 | BACKGROUND

In this section, we review the relevant theoretical framework pertinent to our work. We begin with commonly used estimators: recursive least squares (RLS), Kalman filter, and Schmidt–Kalman variant. Secondly, we discuss the VDM and its incorporation into a sensor fusion architecture for two drones, and finally, we review a drone-based wind estimation methodology.

## 2.1 | Linear Estimators

Let  $\mathbf{x} \in \mathbb{R}^n$  be the state vector and  $\mathbf{z} \in \mathbb{R}^p$  be the vector of measurements (or observations).

### 2.1.1 | RLS

Let Equation (1) describe the observation model:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (1)$$

where  $\mathbf{H} \in \mathbb{R}^{p \times n}$ ,  $\mathbf{v} \in \mathbb{R}^p$  denotes white noise with covariance  $\mathbf{R}$  and  $k \in \{0, 1, 2, \dots\}$ . The state estimate ( $\hat{\mathbf{x}}$ ) and its covariance ( $\mathbf{P}$ ) after each observation ( $k$ ) are given by the following set of recursive equations:

$$\mathbf{K}_k = \mathbf{P}_{k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (2)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k-1}) \quad (3)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k-1} \quad (4)$$

### 2.1.2 | Kalman Filter

In addition to the aforementioned observation model in Equation (1), let the process model (or linearized state dynamics) be characterized by the following difference equation:

$$\mathbf{x}_k = \mathbf{F}_{k-1} \mathbf{x}_{k-1} + \mathbf{w}_k \quad (5)$$

where  $\mathbf{F} \in \mathbb{R}^{n \times n}$  and  $\mathbf{w} \in \mathbb{R}^n$  denote white noise with covariance  $\mathbf{Q}$ . Then, the state estimate and its covariance are given by the following set of recursive equations in two phases: prediction and update. Prediction involves the following operations:

$$\tilde{\mathbf{x}}_k = \mathbf{F}_{k-1} \hat{\mathbf{x}}_{k-1} \quad (6)$$

$$\tilde{\mathbf{P}}_k = \mathbf{F}_{k-1} \hat{\mathbf{P}}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{Q}_k \quad (7)$$

For the update,  $\mathbf{K}_k$  follows Equation (2) with  $\mathbf{P}_{k-1} \leftarrow \tilde{\mathbf{P}}_k$ ,  $\hat{\mathbf{x}}_k$  follows Equation (3) with  $\hat{\mathbf{x}}_{k-1} \leftarrow \tilde{\mathbf{x}}_k$ , and  $\hat{\mathbf{P}}_k$  follows Equation (4) with  $\mathbf{P}_{k-1} \leftarrow \tilde{\mathbf{P}}_k$ . It can be seen that these equations are similar to that of RLS (wherein there are no state dynamics). An RLS estimator is a Kalman filter with  $\mathbf{F} = \mathbf{I}$  and  $\mathbf{Q} = 0$  (deterministic process).

### 2.1.3 | Schmidt-Kalman Filter

Continuing with the Kalman filter framework, let  $\gamma_c \in [0, 1]$  and:

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}^i \\ \hat{\mathbf{x}}^c \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{P}} = \begin{bmatrix} \hat{\mathbf{P}}^{ii} & \hat{\mathbf{P}}^{ic} \\ \hat{\mathbf{P}}^{ci} & \hat{\mathbf{P}}^{cc} \end{bmatrix} \quad (8)$$

with superscripts  $c$  and  $i$  denoting the considered (or nuisance) states and states of importance, respectively. Then, the estimated value and covariance of *considered* states are governed by the following equations in the update steps:

$$\hat{\mathbf{x}}^c \leftarrow \gamma_c \tilde{\mathbf{x}}^c + (1 - \gamma_c) \hat{\mathbf{x}}^c \quad (9)$$

$$\hat{\mathbf{P}}^{cc} \leftarrow \gamma_c^2 \tilde{\mathbf{P}}^{cc} + (1 - \gamma_c^2) \hat{\mathbf{P}}^{cc} \quad (10)$$

## 2.2 | VDM-Based Navigation System

This section summarizes the paradigm of drone navigation based on the VDM. We refer the readers to Khaghani & Skaloud (2016) and Khaghani & Skaloud (2018) for further details.

### 2.2.1 | From Inertial to VDM-Based Navigation

A classical sensor fusion architecture, based on an EKF, estimates the navigation states by combining an INS, GNSS, and/or other sensors. The INS governs the state dynamics (process model), whereas GNSS data are treated as a measurement (or observation). Let  $\mathbf{x}_n$  represent the PVA navigation states of the drone; then, the process model is of the following form:

$$\dot{\mathbf{x}}_n = \mathbf{f}(\mathbf{x}_n, \mathbf{z}_a, \mathbf{z}_\omega) \quad (11)$$

where  $\mathbf{z}_a$  and  $\mathbf{z}_\omega$  are accelerometer and gyroscope measurements. It should be noted that these measurements directly serve as forcing inputs of the above differential equation (Equation (11)). However, a VDM-based process model incorporates mathematical models of aerodynamic forces and moments experienced by the drone instead of directly using raw inertial measurements. Let  $\mathbf{x}_n$  be the position, velocity, attitude, and angular velocity of the drone (13 states with attitude representation in quaternion); then, the VDM-based process model is of the following form:

$$\dot{\mathbf{x}}_n = \mathbf{f}(\mathbf{x}_n, \mathbf{x}_p, \mathbf{x}_w, \mathbf{u}) \quad (12)$$

where  $\mathbf{x}_p \in \mathbb{R}^\zeta$  denotes the vector of aerodynamic model parameters, with  $\zeta \in \mathbb{Z}^+$  denoting the number of aerodynamic model parameters. It should be noted that  $\zeta$  depends on the drone platform and its aerodynamics. Generally speaking,  $\mathbf{x}_p = [\mathbf{C}_f^T \mathbf{C}_m^T]^T$ , where  $\mathbf{C}_f, \mathbf{C}_m$  denote force and moment parameters, respectively. For a conventional fixed-wing drone, these parameters are later defined in Equation (36) and Equation (38), thereby yielding  $\zeta = 21$ . However, for a delta-wing platform,  $\zeta = 44$ .  $\mathbf{x}_w \in \mathbb{R}^3$  denotes the wind velocity in the navigation frame, and  $\mathbf{u} \in \mathbb{R}^v$  denotes autopilot control commands, with  $v$  denoting the number of independent actuators on the drone platform. For a fixed-wing platform, the control commands consist of the propeller, ailerons, elevators, and rudder, thereby yielding  $v = 4$ . However, for a delta-wing platform,  $v = 3$ .

It should be noted that the autopilot commands and wind serve as the forcing inputs of the above differential equation (Equation (12)). The IMU is treated as an external measurement in addition to GNSS and possibly other sensors (if available). As the model parameters  $\mathbf{x}_p$  and wind  $\mathbf{x}_w$  are *a priori* unknown, they are

**TABLE 1**  
Dimensionality of State Space

Drone	Non-fixed		Fixed	
	TP2	eBeeX	TP2	eBeeX
Navigation	13	13	13	13
Parameters	21	44	1	1
Actuators	04	03	04	03
Wind	03	03	03	03
Bias	06	06	06	06
Total	47	69	27	26

estimated as part of the augmented state simultaneously alongside the navigation states. Auxiliary states modeling accelerometer and gyroscope biases  $\mathbf{x}_e$  are also added to the complete state vector. The dimensionality of state space for the two platforms is presented in Table 1 (based on the implementation of Khaghani & Skaloud (2018) and Laupré et al. (2021)). It is very evident that the dimension of the VDM-based navigation filter (47 or 69 when the parameters are not fixed) is much larger than that of the conventional INS/GNSS system (16 or 22 based on Tomé et al. (2000)). Later, in this article, we discuss the possibility of reducing the dimensionality of the state vector. The dimension of this reduced state vector can be inferred from Table 1 under the header “Fixed.”

## 2.2.2 | VDM

Two fixed-wing platforms are used in this study. The first platform, TP2, is a custom-made fixed-wing drone of a conventional shape (see Figure 1). The second platform, eBeeX, is a commercially available delta-wing platform (see Figure 2).

### 2.2.2.1 | TP2

The aerodynamic model presented by Equation (13)–Equation (19) (Ducard, 2009) is used for the conventional fixed-wing drone:

$$F_T^b = \rho n^2 D^4 \left( C_{F_T,1} + C_{F_T,2} J + C_{F_T,3} J^2 \right) \quad (13)$$

$$F_x^w = \bar{q} S \left( C_{F_x,1} + C_{F_x,\alpha} \alpha + C_{F_x,\alpha^2} \alpha^2 + C_{F_x,\beta^2} \beta^2 \right) \quad (14)$$

$$F_y^w = \bar{q} S \left( C_{F_y,1} \beta \right) \quad (15)$$

$$F_z^w = \bar{q} S \left( C_{F_z,1} + C_{F_z,\alpha} \alpha \right) \quad (16)$$

$$M_x^b = \bar{q} S b \left( C_{M_x,a} \delta_a + C_{M_x,\beta} \beta + C_{M_x,\tilde{\omega}_x} \tilde{\omega}_x + C_{M_x,\tilde{\omega}_z} \tilde{\omega}_z \right) \quad (17)$$

$$M_y^b = \bar{q} S \bar{c} \left( C_{M_y,1} + C_{M_y,e} \delta_e + C_{M_y,\tilde{\omega}_y} \tilde{\omega}_y + C_{M_y,\alpha} \alpha \right) \quad (18)$$

$$M_z^b = \bar{q} S b \left( C_{M_z,\delta_r} \delta_r + C_{M_z,\tilde{\omega}_z} \tilde{\omega}_z + C_{M_z,\beta} \beta \right) \quad (19)$$

where  $F_T^b$  denotes the thrust force and  $F_x^w, F_y^w$ , and  $F_z^w$  denote drag, lateral, and lift forces.  $M_x^b, M_y^b$ , and  $M_z^b$  denote aerodynamic moments.  $b, S, \bar{c}$ , and  $D$  are the wing span, wing surface, mean aerodynamic chord, and propeller diameter, respectively. These quantities are summarized in Table 2 under the header “TP2.” The air density is denoted by  $\rho$ , whereas  $\bar{q}$  is the dynamic pressure defined as  $\rho V^2/2$ , with  $V$  denoting the airspeed given by  $V = \|\mathbf{v}_g - \mathbf{x}_w\|$ . Here,  $\mathbf{v}_g$  denotes the ground velocity of the drone in the navigation frame. In contrast, the vector entity  $\mathbf{V}_b = [V^x \ V^y \ V^z]^T = \mathbf{v}_g^b - \mathbf{x}_w^b$  denotes the relative velocity of the drone with respect to the wind in the body frame.  $\alpha$  and  $\beta$  denote the angle of attack and side slip, respectively, and are computed using the following trigonometric equations:  $\alpha = \arctan \frac{V^z}{V^x}$  and  $\beta = \arcsin \frac{V^y}{V}$ .  $J$  is defined as  $V/(D\pi n)$ , with  $n$  denoting the propeller rotation speed. The non-dimensional angular velocities are defined as  $\tilde{\omega}_x = b\omega_x/(2V)$ ,  $\tilde{\omega}_y = \bar{c}\omega_y/(2V)$ , and  $\tilde{\omega}_z = b\omega_z/(2V)$ , where  $[\omega_x \ \omega_y \ \omega_z]^T$  denotes the angular velocity of the drone with respect to the navigation frame. The deflections of the control surfaces (i.e., aileron, elevator, and rudder) are denoted by  $\delta_a, \delta_e$ , and  $\delta_r$ , respectively. Finally, the aerodynamic model parameters are represented by  $C$  in Equation (13)–Equation (19).



FIGURE 1 Three-dimensional model of TP2



FIGURE 2 eBeeX

TABLE 2  
Vehicle Parameters

Parameter	Notation	TP2	eBeeX	Units
Wing span	$b$	1.63	1.17	$m$
Wing surface	$S$	0.3439	0.32	$m^2$
Mean aerodynamic chord	$\bar{c}$	0.225	0.295	$m$
Propeller diameter	$D$	0.362	0.23	$m$
Mass	$m_0$	2.771	1.45	$kg$

### 2.2.2.2 | *eBeeX*

The aerodynamic model for the delta-wing drone (Laupré et al., 2021) is characterized by Equation (20)–Equation (25) for force and moment coefficients.

$$C_{F_x} = C_{F_{x0}} + C_{F_{x\alpha}} \alpha + C_{F_{x\alpha^2}} \alpha^2 + C_{F_{x\beta}} \beta + C_{F_{x\delta_e}} \delta_e + C\theta_{A_{F_x}} \quad (20)$$

$$C_{F_y} = C_{F_{y\beta}} \beta + C_{F_{y\beta^2}} \beta^2 + C_{F_{y\delta_a}} \delta_a + C\theta_{A_{F_y}} \quad (21)$$

$$C_{F_z} = C_{F_{z0}} + C_{F_{z\alpha}} \alpha + C_{F_{z\alpha^2}} \alpha^2 + C_{F_{z\beta}} \beta + C_{F_{z\delta_e}} \delta_e + C\theta_{A_{F_z}} \quad (22)$$

$$C_{M_x} = C_{M_{x\beta}} \beta + C_{M_{x\beta^2}} \beta^2 + C_{M_{x\delta_a}} \delta_a + C\theta_{A_{M_x}} \quad (23)$$

$$C_{M_y} = C_{M_{y0}} + C_{M_{y\alpha}} \alpha + C_{M_{y\alpha^2}} \alpha^2 + C_{M_{y\delta_e}} \delta_e + C\theta_{A_{M_y}} \quad (24)$$

$$C_{M_z} = C_{M_{z\beta}} \beta + C_{M_{z\beta^2}} \beta^2 + C_{M_{z\delta_a}} \delta_a + C\theta_{A_{M_z}} \quad (25)$$

Here,  $\delta_a = \frac{-\delta_L + \delta_R}{2}$ ,  $\delta_e = \frac{\delta_L + \delta_R}{2}$ , and all of the other terms have already been defined. It should be noted that a delta-wing drone does not have elevators and ailerons typical of a conventional aircraft geometry; rather, it has two independent control surfaces commonly referred to as elevons. The elevon deflections are denoted by  $\delta_L$  and  $\delta_R$ , where the subscripts  $L$  and  $R$  denote left and right deflection, respectively. Additionally,  $C\theta$  is defined as follows:

$$C\theta_{A_i} = C_{A_{i\tilde{\omega}_x}} \tilde{\omega}_x + C_{A_{i\tilde{\omega}_y}} \tilde{\omega}_y + C_{A_{i\tilde{\omega}_z}} \tilde{\omega}_z \quad (26)$$

with  $A \in \{F, M\}$  distinguishing forces from moments and  $i \in \{x, y, z\}$  defining the three axes. Subsequently,  $F_i^w = \bar{q} S C_{F_i} \forall i \in \{x, y, z\}$  and  $M_j^b = \bar{q} S \epsilon C_{M_j} \forall j \in \{x, y, z\}$  such that  $\epsilon = b$  for  $j \in \{x, z\}$  and  $\epsilon = \bar{c}$  for  $j \in \{y\}$ . In this way, the force and moment equations for *eBeeX* are similar to Equation (14)–Equation (19). The thrust models for both platforms are identical. Physical quantities such as the wing surface  $S$ , span  $b$ , mean aerodynamic chord  $\bar{c}$ , and propeller diameter  $D$  are summarized in Table 2 under the header “eBeeX.”

### 2.2.3 | *IMU Observation Model*

The observation model of the IMU, presented in Khaghani & Skaloud (2018), is given by the following set of equations:

$$\mathbf{z}_f = \underbrace{\mathbf{f}^b}_{f(C_F, \mathbf{x}_w, \mathbf{u})} + \underbrace{\dot{\boldsymbol{\omega}}_b}_{f(C_M, \mathbf{x}_w, \mathbf{u})} \times \mathbf{r}_{bl} + \boldsymbol{\omega}_b \times \boldsymbol{\omega}_b \times \mathbf{r}_{bl} + \mathbf{w}_f \quad (27)$$

$$\mathbf{z}_\omega = \boldsymbol{\omega}_b + \mathbf{w}_\omega \quad (28)$$

where  $\mathbf{f}^b$  denotes the model of specific force at the center of gravity of the drone,  $\boldsymbol{\omega}_b$  is the angular velocity of the drone with respect to an inertial frame,  $\dot{\boldsymbol{\omega}}_b$  denotes the model of angular acceleration,  $\mathbf{r}_{bl}$  denotes the lever arm between the drone’s center of gravity and the IMU, and  $\{\mathbf{w}_f, \mathbf{w}_\omega\}$  together denote measurement noise. The specific force is given as follows:



$$\mathbf{f}^b(C_F, \mathbf{x}_w, \mathbf{u}) = \frac{1}{m_0} \begin{bmatrix} F_T^b \\ 0 \\ 0 \end{bmatrix} + (\mathbf{R}_b^w)^T \begin{bmatrix} F_x^w \\ F_y^w \\ F_z^w \end{bmatrix}; \quad (29)$$

$$\text{with } \mathbf{R}_b^w = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \text{ and}$$

$$\dot{\boldsymbol{\omega}}_b(C_M, \mathbf{x}_w, \mathbf{u}) = (\mathbf{I}^b)^{-1} \begin{bmatrix} M_x^b \\ M_y^b \\ M_z^b \end{bmatrix} - \boldsymbol{\omega}_b \times \mathbf{I}^b \boldsymbol{\omega}_b, \quad (30)$$

where  $\mathbf{I}^b$  denotes the inertia tensor and  $m_0$  is the drone mass.

### 2.3 | Wind Estimation Based on Vehicle Kinematics

As highlighted in Equation (12), wind serves as one of the forcing inputs of the differential equation modeling the VDM state dynamics; therefore, its estimation is important for a reliable navigation solution. In this section, we review one of the drone-based wind estimation methodologies (Johansen et al., 2015), which has been both theoretically and practically validated on three different platforms. This methodology is based on the Bucy–Kalman filter (continuous-time counterpart of the discrete-time Kalman filter) and makes use of a Pitot tube as an external measurement. Unlike the original work, we choose to present the methodology in discrete time. Let  $\mathbf{x}_w \in \mathbb{R}^3$  be the wind velocity in the navigation frame and  $\gamma \in \mathbb{R}^+$  be the scale factor of the Pitot tube; then, the process model is of the following form:

$$\begin{bmatrix} \mathbf{x}_{w_{k+1}} \\ \gamma_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{w_k} \\ \gamma_k \end{bmatrix} + \begin{bmatrix} \mathbf{w}_{w_k} \\ \mathbf{w}_{\gamma_k} \end{bmatrix} \quad (31)$$

with  $\{\mathbf{w}_{w_k}, \mathbf{w}_{\gamma_k}\}$  denoting white noise. The observation model is of the following form:

$$z_k + v_k = h(\mathbf{x}_{w_k}, \gamma_k) = \mathbf{d}^T \mathbf{R}_l^b \mathbf{x}_{w_k} + u_k \gamma_k \quad (32)$$

where  $\mathbf{d} = [1 \ 0 \ 0]$ ,  $u$  is the airspeed measured by the Pitot tube,  $\mathbf{R}_l^b$  is the rotation matrix from the navigation frame to the body frame,  $z$  is the drone's longitudinal velocity, obtained as a result of sensor fusion (INS/GNSS), and  $v$  denotes white noise. Incorporation of Equation (31) and Equation (32) as the process and observation models of the Kalman filter, respectively, yields the wind velocity.

Observability of the system is asserted by computing the observability Gramian at the end of the flight by using Equation (33) and finding it to be positive definite:

$$\mathbf{W}_0(k) = \sum_{\tau=0}^k (\mathbf{F}^T)^\tau \mathbf{H}^T \mathbf{H} \mathbf{F}^\tau \quad (33)$$

It should be noted that  $\mathbf{F}$  and  $\mathbf{H}$  are generally defined by Equation (5) and Equation (1), respectively. In this context,  $\mathbf{F} = \mathbf{I}$  (deduced from Equation (31)) and  $\mathbf{H} = [\mathbf{d}^T \mathbf{R}_l^b u]$  (deduced from Equation (32)). To ensure observability, the

attitude of the drone should not be constant. The results reported in Johansen et al. (2015) show that planar wind is estimated with a lower uncertainty than vertical wind. This difference is due to prevailing maneuvers in yaw rather than in pitch. We refer the readers to any book on advanced systems and control, for example, Chen (1999), for an exhaustive description of the observability Gramian.

This methodology assumes that the wind varies relatively slowly with respect to time. Albeit limited by the rotational movement of the aircraft, these variations can be estimated by defining a cut-off frequency in the Kalman gain of the EKF. Wind velocity frequencies that are lower than this cut-off frequency are captured, whereas the other frequencies are filtered. The authors provide general comments on tuning the Kalman filter; however, no systematic methodology is presented, suggesting that empirical testing is used in their estimator.

### 3 | DECOUPLED AERODYNAMIC CALIBRATION

We define *calibration* as the phase wherein aerodynamic model parameters are estimated subject to known navigation states  $\mathbf{x}_n$ . Meanwhile, *application* is defined as the phase wherein navigation states, wind, and possibly other states are estimated subject to known (or reasonably known) aerodynamic model parameters.

The calibration procedure is premised on the knowledge of navigation states in addition to control commands and geometric parameters. The navigation states are obtained by employing post-processing differential techniques, based on an optimal (Kalman) smoother, on INS/GNSS data. This approach results in an accuracy at the centimeter level for position, 0.01 [m/s] for velocity, and  $\sim 0.1$  [deg] for attitude, as highlighted in many studies, such as Niu et al. (2015). In contrast, control commands are logged by the autopilot, and all geometric parameters (including the inertia tensor) are known.

Substitution of navigation states, control commands, and geometric parameters in Equation (27) for TP2 results in a nonlinear observation model involving aerodynamic model parameters and wind ( $21 + \{3 \times \text{number of observations}\}$  unknowns). Such substitution has no decoupling effect. A similar substitution in Equation (30) results in a nonlinear model involving aerodynamic moment parameters and wind ( $11 + \{3 \times \text{number of observations}\}$  unknowns). For such a substitution to work, it is assumed that gyroscope measurements can be numerically differentiated. Although this substitution has no decoupling effect, it gives credible evidence that if the wind is estimated *a priori* (e.g., via Johansen et al. (2015)), the model in Equation (30) is linear in aerodynamic moment parameters. Consequently, if both wind and aerodynamic moment parameters are known, then the model in Equation (27) is linear in aerodynamic force parameters. As a result, we have three linear and decoupled estimators to collectively carry out aerodynamic calibration of drones using inertial data and Pitot tube observations (during windy conditions). The calibration strategy is illustrated in Figure 3. In the next subsection, we introduce the methodology by first presenting a general model structure based on the Schmidt–Kalman filter, which is shown to be the same for all three estimators. Secondly, we detail the mathematics associated with the estimators for the conventional drone platform. Thirdly, we propose a heuristic based on the observability Gramian to discern the most observable states. Finally, we present a calibration algorithm as a combination of the aforementioned three steps.

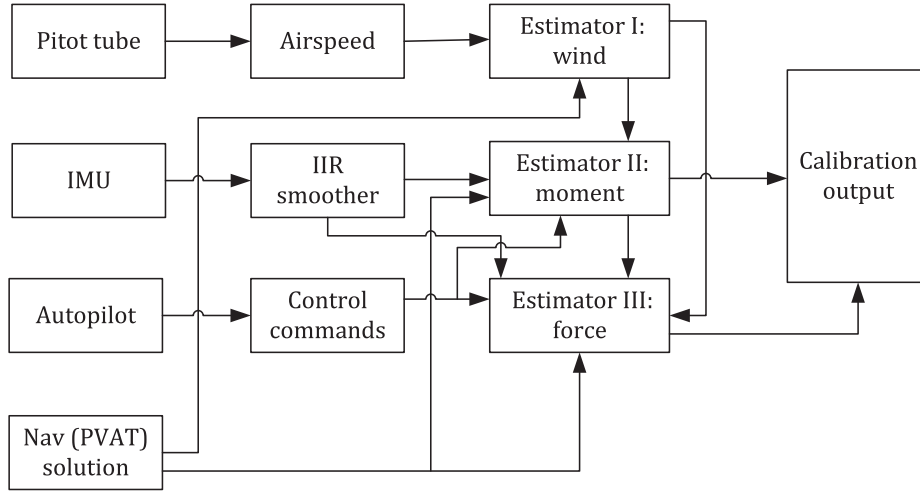


FIGURE 3 Calibration procedure as a sequence of three decoupled estimators IIR: infinite impulse response.

### 3.1 | Generic Model Structure

The process model is of the following form, with  $\mathbf{w}$  as the process noise:

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{x}_{k+1}^i \\ \mathbf{x}_{k+1}^c \end{bmatrix} = \begin{bmatrix} \mathbf{x}_k^i \\ \mathbf{x}_k^c \end{bmatrix} + \mathbf{w}_k = \mathbf{x}_k + \mathbf{w}_k \quad (34)$$

The observation model is of the form given in Equation (1). The previously reviewed works (Johansen et al., 2015; Khaghani & Skaloud, 2018) have shown that the observability of aerodynamic model parameters and wind is dependent on the trajectory, and hence, they are only mildly observable. Studies from Brink (2017) have shown that estimating nuisance states (or mildly observable states) in a traditional setting can impact filter accuracy and consistency. Therefore, we make use of a partial-update Schmidt–Kalman filter (PSKF) (Brink, 2017), which has been shown to be effective in estimating both constant and time-varying nuisance states. The PSKF facilitates a generalized estimation approach wherein only a portion of the traditional full filter update can be applied to selected states. In this way, according to observability conditions, the nuisance terms can be treated as either full filter states and updated or as *considered* states and not updated. Such a framework has proven to be unbiased and consistent (Brink, 2017).

### 3.2 | Estimator I: Wind

As *a priori* knowledge of wind is a key requirement for linearizing Equation (30), we make use of the methodology proposed in Johansen et al. (2015) (reviewed in Section 2.3) to estimate the wind. However, as pointed out in the original work, the observability of some of the states in Equation (31) is dependent on the trajectory (change in attitude); therefore, these states are estimated only when observable. This estimation is carried out by making use of a PSKF (discussed in Section 3.5), in contrast to the standard Kalman filter proposed in Johansen et al. (2015).

### 3.3 | Estimator II: Moments

By substituting formerly obtained wind-related entities ( $\alpha$ ,  $\beta$ , and  $V$ ) and moments (from Equation (17)–Equation (19)) in Equation (30) for *TP2* and then rearranging, the following observation model is obtained:

$$\mathbf{z}_m + \mathbf{v}_m = \mathbf{H}_m \mathbf{C}_m \quad \text{with} \quad (35)$$

$$\mathbf{H}_m = \mathbf{D}_m \begin{bmatrix} \delta_a & \beta & \tilde{\omega}_x & \tilde{\omega}_z & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \delta_e & \tilde{\omega}_y & \alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \delta_r & \tilde{\omega}_z & \beta \end{bmatrix},$$

$$\mathbf{D}_m = \text{diag}(b, \bar{c}, b), \mathbf{z}_m = \frac{1}{\bar{q}S} (\mathbf{I}^b \dot{\omega}_b + \omega_b \times \mathbf{I}^b \omega_b),$$

$$\mathbf{C}_m = \left[ C_{M_x \alpha} C_{M_x \beta} C_{M_x \tilde{\omega}_x} C_{M_z \tilde{\omega}_z} C_{M_y 1} C_{M_y e} C_{M_y \tilde{\omega}_y} C_{M_x \alpha} C_{M_z \delta_r} C_{M_z \tilde{\omega}_z} C_{M_z \beta} \right]^T \quad (36)$$

$\mathbf{v}_m$  denotes white noise, and  $\dot{\omega}_b$  is obtained by employing a high-order (e.g., eighth-order in our case) differentiator on a sequence of gyroscopic observations in which deterministic errors have been removed (from pre-calibration and INS/GNSS integration). Such a differentiator is, for instance, designed using the Parks–McClellan algorithm<sup>1</sup> (based on the Remez exchange algorithm and Chebyshev approximation theory). It should be noted that the quantities ( $\alpha$ ,  $\beta$ , and  $\mathbf{V}$ ) required to identify  $\mathbf{C}_m$  in Equation (35) are taken from estimator I.

The process model is as given in Equation (34). This model results in an  $11 \times 11$  process-noise covariance matrix, which is non-trivial to tune practically. Moreover, as model parameters associated with aerodynamic moments are practically constant ( $\dot{\mathbf{C}}_m(t) = 0$ ) (Khaghani & Skaloud, 2018), the complexity of the estimator is reduced by design by setting the process noise to zero. This effectively turns the Kalman filter into an RLS moment parameter estimator. It should be noted that white noise can also be used; however, selecting its form and strength is not straightforward and requires adaptation to the stability of the meteorological conditions. In the end, we employ a partial-update framework to address observability concerns (as discussed in Section 3.5).

### 3.4 | Estimator III: Forces

By substituting previously estimated wind and moment parameters in Equation (27) for *TP2* and then rearranging, the observation model is obtained:

$$\mathbf{z}'_f + \mathbf{v}_f = \mathbf{H}_f \mathbf{C}_f \quad \text{with} \quad (37)$$

$$\mathbf{H}_f = \frac{1}{m_0} \left[ \mathbf{A}_{33} (\mathbf{R}_b^w)^T \mathbf{A}_{37} \right], \mathbf{A}_{33} = \rho D^4 \begin{bmatrix} n^2 & \frac{nV}{D\pi} & \left( \frac{V}{D\pi} \right)^2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{A}_{37} = \bar{q}S \begin{bmatrix} 1 & \alpha & \alpha^2 & \beta^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \alpha \end{bmatrix}$$

<sup>1</sup><https://ch.mathworks.com/help/signal/ref/firpm.html>

$$\mathbf{C}_f = \left[ C_{F_T1} C_{F_T2} C_{F_T3} C_{F_x1} C_{F_x\alpha} C_{F_x\alpha2} C_{F_x\beta2} C_{F_y1} C_{F_z1} C_{F_z\alpha} \right]^T \quad (38)$$

$$\mathbf{z}'_f = \mathbf{z}_f - \boldsymbol{\omega}_b \times \boldsymbol{\omega}_b \times \mathbf{r}_{bI} - \left[ (\mathbf{I}^b)^{-1} (\mathbf{M}^b - \boldsymbol{\omega}_b \times \mathbf{I}^b \boldsymbol{\omega}_b) \right] \times \mathbf{r}_{bI} \quad (39)$$

where  $m_0$  is the mass of the drone,  $\mathbf{z}_f$  is the accelerometer reading,  $\mathbf{M}^b = \begin{bmatrix} M_x^b & M_y^b & M_z^b \end{bmatrix}^T$ ,  $\mathbf{v}_f$  denotes white noise, and the other terms are already defined. The process model is as given in Equation (34). Following the same reasoning as in the estimation of moment parameters, the process noise is chosen to be zero. This corresponds to an RLS force parameter estimator with a partial-update framework. Again, the entities  $(\mathbf{C}_m, \alpha, \beta, \mathbf{V})$  needed to identify  $\mathbf{C}_f$  are taken from the two previous estimators.

### 3.5 | Heuristic of Uniqueness

In the previous subsections, we presented process and observation models of three decoupled linear estimators. Now, we introduce a heuristic to segregate the system state of these estimators into nuisance and observable states so as to carry out partial updates. We take inspiration from Johansen et al. (2015), wherein the observability of wind is evaluated only at the end of the trajectory by computing the observability Gramian and finding it to be full rank. This metric asserts that the system is observable at the end; however, it does not convey any information regarding the specific states that are observable or precisely when they are observable. We address these questions using the following empirical approach.

For the chosen generic model structure, applicable to all three estimators (wind, moments, and forces), with  $\mathbf{F} = \mathbf{I}$ , the observability Gramian can be computed after each observation by the following recursive summation of the normal equation:

$$\mathbf{W}_0(k) = \mathbf{W}_0(k-1) + \mathbf{H}_k^T \mathbf{H}_k \quad (40)$$

Let  $\mathbf{V}_k = [\mathbf{v}_1^k \ \dots \ \mathbf{v}_n^k]$  and  $\Lambda_k = \text{diag}[\lambda_1^k, \dots, \lambda_n^k]$  such that  $\mathbf{W}_0(k) \mathbf{V}_k = \mathbf{V}_k \Lambda_k$ . It should be noted that  $(\mathbf{v}_j^k, \lambda_j^k)$  denote the  $j$ -th eigenvector/eigenvalue pair of  $\mathbf{W}_0(k)$ . These eigenvalues are sorted in ascending order (1 denoting the smallest value and  $n$  the largest). We compute these values using the MATLAB<sup>2</sup> function `eig`. Let  $k = K$  denote the last observation; then, we define a closeness matrix:

$$\Gamma_q = \mathbf{V}_q^T \mathbf{V}_K \forall q \in \{1, 2, \dots, K\} = \begin{bmatrix} (\mathbf{v}_1^q)^T \mathbf{v}_1^K & (\mathbf{v}_1^q)^T \mathbf{v}_2^K & \dots & (\mathbf{v}_1^q)^T \mathbf{v}_n^K \\ (\mathbf{v}_2^q)^T \mathbf{v}_1^K & (\mathbf{v}_2^q)^T \mathbf{v}_2^K & \dots & (\mathbf{v}_2^q)^T \mathbf{v}_n^K \\ \vdots & & & \\ (\mathbf{v}_n^q)^T \mathbf{v}_1^K & (\mathbf{v}_n^q)^T \mathbf{v}_2^K & \dots & (\mathbf{v}_n^q)^T \mathbf{v}_n^K \end{bmatrix} \quad (41)$$

Note that  $\Gamma_K$  is an identity matrix. Moreover, each element of  $\Gamma_q$  is a scalar product between the eigenvectors of  $\mathbf{W}_0(q)$  and  $\mathbf{W}_0(K)$ . We then write  $\Gamma_q$  in the following form:

$$\Gamma_q = \begin{bmatrix} \mathbf{g}_q^1 & \mathbf{g}_q^2 & \dots & \mathbf{g}_q^n \end{bmatrix} \quad (42)$$

<sup>2</sup><https://ch.mathworks.com/products/matlab.html>

Subsequently, we compute the maximum value along each column vector  $\mathbf{g}_q^j \forall j \in \{1, 2, \dots, n\}$ :

$$\max(\Gamma_q) = \left[ \max(|\mathbf{g}_q^1|) \quad \max(|\mathbf{g}_q^2|) \quad \dots \quad \max(|\mathbf{g}_q^n|) \right] = \left[ g_{qa}^1 \quad g_{qb}^2 \quad \dots \quad g_{qc}^n \right]^T \quad (43)$$

where  $|\mathbf{g}_q^j|$  denotes the element-wise absolute value of  $\mathbf{g}_q^j$ . In other words, the  $a$ -th eigenvector of  $\mathbf{W}_0(q)$  is closest to the first eigenvector of  $\mathbf{W}_0(K)$  and so on. In this way, it is numerically possible to approximately associate the eigenvalues of the observability Gramian at each epoch to a chosen basis, which are the eigenvectors of  $\mathbf{W}_0(K)$ .

At the end of this step, we have  $n$  time series (of length  $K$ ) of eigenvalues of  $\mathbf{W}_0$ . Each time series is characterized by one of the eigenvectors of  $\mathbf{W}_0(K)$ . Let  $\underline{\lambda}_j(k)$  denote such a time series. As the system satisfies conditions of persistence of excitation (Green & Moore, 1986) for uniform observability, there exists  $\epsilon > 0$  such that  $\mathbf{W}_0(K) > \epsilon \mathbf{I}$  (Kalman & Bucy, 1961). As a result, for some sufficiently large  $K$ ,  $\mathbf{W}_0(K)$  is positive definite. Therefore, eigenvalues of the observability Gramian tend to grow with each observation subject to excitation/dynamics. This trend is experimentally shown in Section 5. Our methodology, thus far, approximately associates this growth to an eigenvector of  $\mathbf{W}_0(K)$ .

To ascertain the uniqueness of information brought in by each observation, we compute the difference between adjacent elements of each time series:

$$\nabla \underline{\lambda}_j(k) = \underline{\lambda}_j(k) - \underline{\lambda}_j(k-1) \quad (44)$$

If  $\nabla \underline{\lambda}_j(k) > \text{threshold}$ , then new information is available and some states are observable. To use this heuristic for the generic model structure, it is imperative to first change the basis of state space to the basis created with the eigenvectors of  $\mathbf{W}_0(K)$ . This approach causes the states to correspond to a sufficiently large  $\nabla \underline{\lambda}_j(k)$  being updated while the others are considered.

For any of the three estimators, a general procedure is summarized in the form of an algorithm:

1. **Build the observability Gramian.** For epoch  $k \in \{1, 2, \dots, K\}$ ,
  - (a) Compute  $\mathbf{H}_k$  and  $\mathbf{W}_0(k)$
  - (b) Compute the eigenvalues ( $\Lambda_k$ ) and eigenvectors ( $\mathbf{V}_k$ ) of  $\mathbf{W}_0(k)$
2. **Form the new basis.** At the last epoch  $K$ ,
  - (a) Having  $\mathbf{W}_0(K)$ , find the basis  $\mathbf{V}_K$
  - (b) Compute the state-transformation matrix:  $\mathbf{T} = \mathbf{V}_K^T$
3. **Compute the evolution in observability.** For epoch  $k \in \{1, 2, \dots, K\}$ ,
  - (a) Compute  $\Gamma_k = \mathbf{V}_k^T \mathbf{V}_K$
  - (b) Compute  $\max(\Gamma_k)$
  - (c) Rearrange the eigenvalues into  $n$  time series  $\underline{\lambda}_j(k)$  for  $j \in \{1, 2, \dots, n\}$
  - (d) Compute the difference  $\nabla \underline{\lambda}_j(k)$
4. Initialize the state  $\mathbf{x}$  estimator. For  $k \in \{1\}$ ,
  - (a) Transform the initial states, initial covariance, and process noise (if non-zero);  $\mathbf{x}' \leftarrow \mathbf{T}\mathbf{x}$ ,  $\mathbf{P}' \leftarrow \mathbf{T}\mathbf{P}\mathbf{T}^T$ ,  $\mathbf{Q}' \leftarrow \mathbf{T}\mathbf{Q}\mathbf{T}^T$
5. Estimate the “transformed” states  $\mathbf{x}'$ . For  $k \in \{2, \dots, K\}$ ,
  - (a) If  $\nabla \underline{\lambda}_j(k) > \text{threshold} \forall j \in \{1, 2, \dots, n\}$ , update the state, else *consider*; a partial update is carried out in this step

6. Transform back to the original basis. For  $k \in \{1, \dots, K\}$ ,
  - (a) Invert the basis to obtain the estimated states  $\mathbf{x} \leftarrow \mathbf{T}^{-1}\mathbf{x}'$  and covariance  $\mathbf{P} \leftarrow \mathbf{T}^T\mathbf{P}'\mathbf{T}$  in their original basis set

The partial-update algorithm detailed above is carried out sequentially for the three estimators in the following order: i) wind estimation with non-zero process noise, ii) moment parameter estimation with zero process noise, iii) force parameter estimation with zero process noise. A similar procedure is followed for the other fixed delta-wing platform to carry out aerodynamic calibration.

## 4 | EXPERIMENTAL SETUP

In this section, we describe the two drone platforms used in this research and the associated relevant avionics.

### 4.1 | Drone Platform

We test the proposed methodology on two platforms with distinctly different geometries. The first platform is an in-house-developed conventional fixed-wing drone with the shape of hobby plane “Mentor” from multiplex (Multiplex HiTec power peak, 2022), referred to as *TP2*. The second platform is a commercial delta-wing drone, *eBeeX-RTK*, from senseFly SA, Switzerland, which is henceforth referred to as *eBeeX*. Images of the two platforms are presented in Figure 1 and Figure 2.

### 4.2 | Avionics

Implementation of the proposed calibration algorithm is incumbent on the availability of flight data. In this section, the necessary custom hardware development for data acquisition is detailed for the two platforms. It should be noted that these developments are based on size, weight, and power constraints that are different for each drone.

#### 4.2.1 | *TP2*

This platform has been used in previous research (Khaghani & Skaloud, 2018; Laupré & Skaloud, 2021). However, to improve the quality of inertial data, we have developed a state-of-the-art payload comprised of an IMU from Sensoror, the STIM 318 (Stim 318, 2022), which represents one of the best available options on the market in terms of performance per size and weight. Its bias instability is  $0.3^\circ/h$  and  $2\mu g$  for gyroscopes and accelerometers, respectively. Some of the other parameters of the STIM 318 can be inferred from Table 3. An image of the developed payload is shown in Figure 4(a).

The IMU data stream, at a frequency of 500 Hz, is connected to a dedicated board called Sentiboard (Albrektsen & Johansen, 2018), which is responsible for time-tagging the data in a common global positioning system (GPS) reference time. A second serial port on the Sentiboard handles messages from the GNSS receiver, where the time reference is provided by a pulse-per-second (PPS) signal issued at the same frequency as the GNSS messages. We use a multi-constellation (minimum GPS and GLONASS), multi-frequency (minimum L1 and L2) TOPCON B125 GNSS

**TABLE 3**  
Parameters of Experimental IMUs  
RMS: root mean square.

STIM 318			ADIS 16475 -2			NavChip v1		
Parameters	Value	Unit	Parameters	Value	Unit	Parameters	Value	Unit
Accelerometer								
Bias 1-year stability	1.5	mg	Repeatability	1.4	mg	In-run bias stability	0.05	mg
Bias instability	0.003	mg	In-run bias stability	3.6	$\mu\text{g}$	Bias accuracy	8	mg
Velocity random walk	0.015	$\text{m/s}/\sqrt{h}$	Velocity random walk	0.012	$\text{m/s}/\sqrt{h}$	Noise density (RMS)	50	$\mu\text{g}/\sqrt{h}$
Gyroscope								
Bias range	250	deg/h	Repeatability	0.7	deg/s	In-run bias stability	10	deg/h
Drift rate stability	3	deg/h	In-run bias stability	2.5	deg/h	Noise density (RMS)	0.003	$\text{deg/s}/\sqrt{h}$
Bias instability	0.3	deg/h	Angular random walk	0.15	$\text{deg}/\sqrt{h}$	Angle random walk	0.18	$\text{deg}/\sqrt{h}$
Angular random walk	0.15	$\text{deg}/\sqrt{h}$				Bias accuracy	0.2	deg/s
Other parameters								
Size (length $\times$ width $\times$ height)	44.8 $\times$ 38.6 $\times$ 21.5		11 $\times$ 15 $\times$ 11			13.5 $\times$ 24 $\times$ 9		mm
Weight	55		1.3			6		g
Power	1800		145			214		mW
Cost	~7000		~500			~500		Euro
Year	2020		2021			2011		-

receiver, and a three-frequency antenna, which gathers code and phase measurements for post-processing differential position and velocity references. Meanwhile, the control commands are stored in the autopilot for later use. The autopilot is the open-source PixHawk (Meier et al., 2011) (v.2) running a modified firmware of PX4 (PX4 autopilot, 2009) that accommodates the aforementioned GNSS receiver and synchronizes its internal time with respect to GPS time. In addition, we have also integrated an in-house-developed redundant IMU (R-IMU) board consisting of four ADIS-16475 sensors. Some of the parameters of this IMU can be inferred from Table 3. The data from these IMUs are recorded on the R-IMU board itself against a GPS time stamp.

#### 4.2.2 | *eBeeX-RTK*

The inertial data are acquired by means of the custom payload presented in Figure 4(b). The developed payload consists of an R-IMU board, and the inertial data for the two IMUs<sup>3</sup> are stored internally on the board. Some of the parameters of this IMU can be inferred from Table 3.

This board is connected to the *eBeeX* via a USB cable that provides i) power for the electronics, ii) GNSS data from the receiver, and iii) the PPS signal for synchronizing the IMU clocks with GPS time. The GNSS receiver used in this study is a

<sup>3</sup>NavChip v.1 from InterSense-Thales <https://www.intersense.com/navchip>



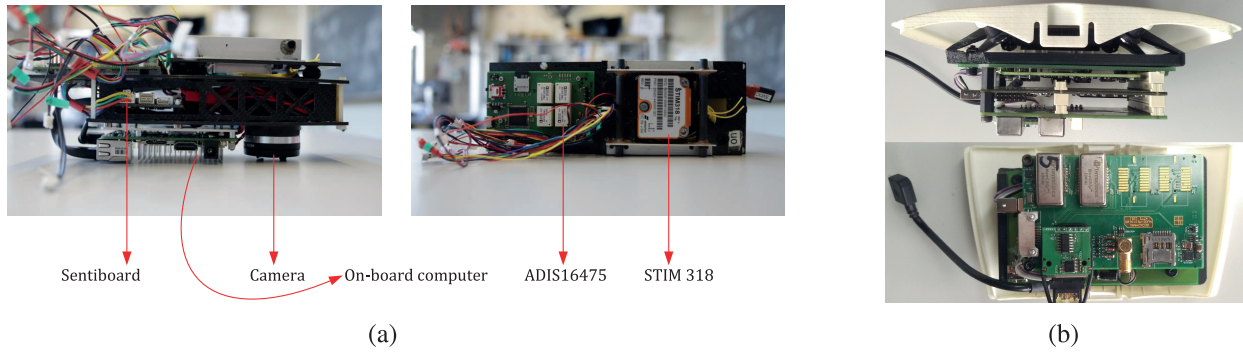


FIGURE 4 Payload: (a) *TP2* payload and (b) *eBeeX* payload

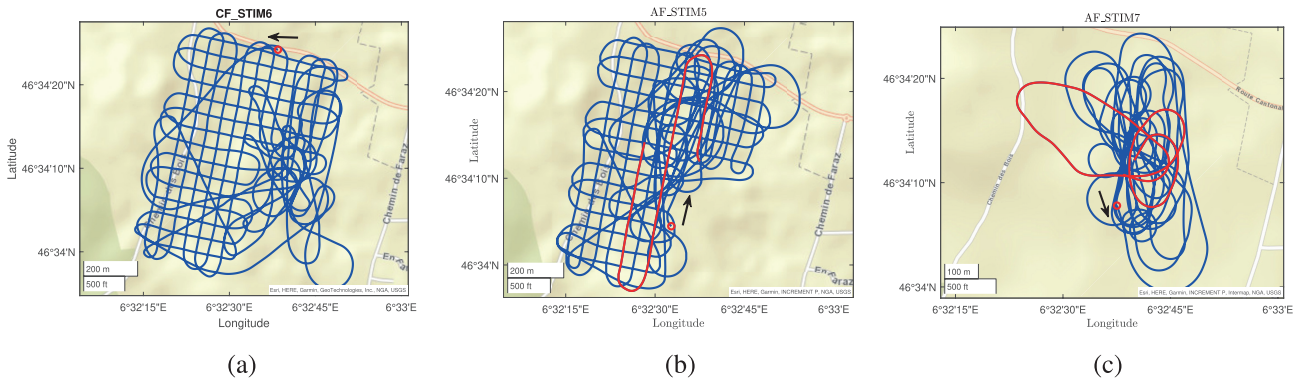


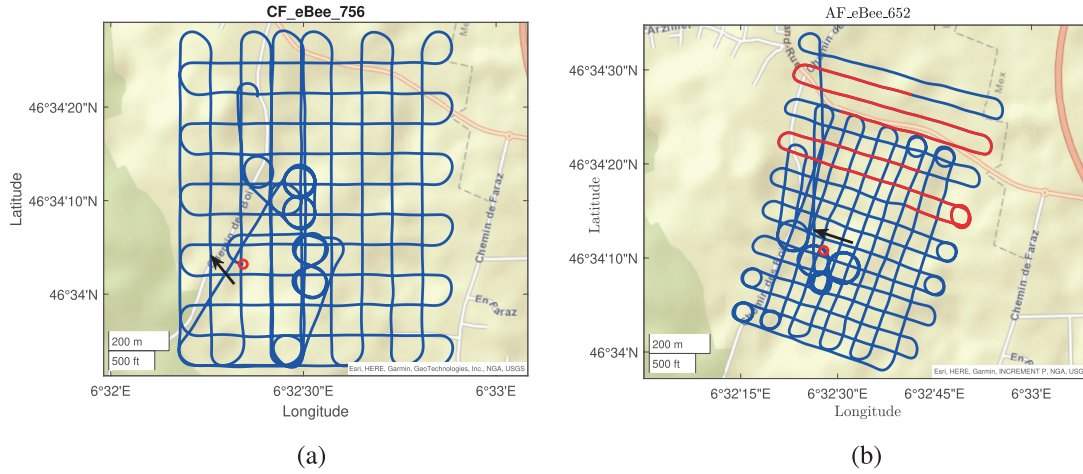
FIGURE 5 *TP2* flight trajectories with *CF\_STIM6* (a) as a calibration flight and *AF\_STIM5* (b) and *AF\_STIM7* (c) as application flights. The 2-min simulated GNSS outages are depicted in red.

multi-constellation (GPS, GLONASS) Septentrio AsteRx-m2 with dual-frequency capacity (L1, L2) alongside a Maxtena M1227 antenna. The flight control commands, autopilot data, and raw GNSS observations are recorded within the autopilot in GPS time.

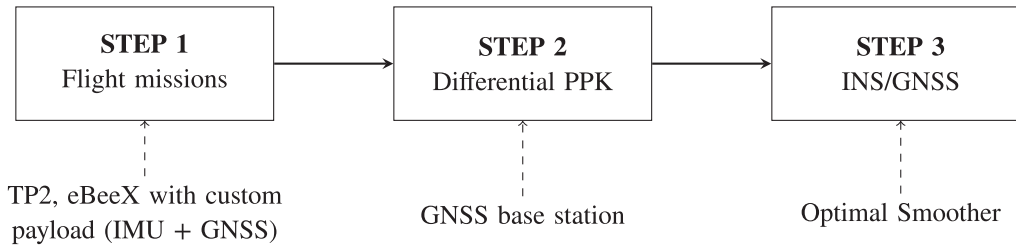
### 4.3 | Experimental Flights

Experimental flights of the aforementioned drones were conducted in a rural region in Switzerland ( $46.5788^\circ$  N,  $6.5394^\circ$  E) between 2019 and 2021, and the flight data were logged using the avionics described in Section 4. At least two flights per drone were used in the analysis: one for calibration and one for application. The calibration and application flights for *TP2* are shown in Figure 5, whereas those for *eBeeX* are shown in Figure 6. It should be noted that a 2-min-long GNSS outage is simulated later in the application flights to evaluate the performance of the proposed calibration methodology. We have marked a portion of the trajectory in Figure 5 and Figure 6 in red to explicitly highlight the outage.

The beginning of the trajectory is marked with a red circle, and a black arrow indicates the initial flying direction. For a calibration flight, the proposed methodology is implemented to compute the aerodynamic model parameters. These parameters are subsequently used in an application flight, wherein they are either i) held fixed throughout the flight or ii) used as an initial guess with a low uncertainty within the VDM-EKF framework.



**FIGURE 6** *eBeeX* flight trajectories with *CF\_eBee\_756* (a) as calibration and *AF\_eBee\_652* (b) as application flights The 2-min simulated GNSS outages are depicted in red.



**FIGURE 7** Steps for obtaining the reference trajectory

#### 4.4 | Reference Trajectory

To evaluate the performance of the proposed methodology in the VDM-EKF framework, the reference trajectories are obtained as a result of post-processing the GNSS and inertial data, as shown in Figure 7. The flight mission is performed for the two drones: *TP2* and *eBeeX*. Both drones carry a dedicated payload to collect IMU and GNSS data (step 1), as described in Section 4.2.1 and Section 4.2.2. During the missions, a GNSS base station, hosting a multi-constellation and multi-frequency receiver, JAVAD Triumph-LS (minimum L1, L2 on GPS and GLONASS), records the code and phase signals at 10 Hz for the precise differential PPK<sup>4</sup> solution (step 2) of the trajectory. This approach allows cm accuracy for position and cm/s accuracy for velocity to be obtained. The precise position and velocity of the trajectory are then combined using optimal smoothing with the inertial data using an INS/GNSS software<sup>5</sup> to yield a reference trajectory. The obtained reference trajectory, used for calibration and navigation performance comparison, provides the required accuracies of cm in positioning, cm/s in velocity, 0.1 *deg* in roll and pitch, and 0.2 *deg* in yaw. Although the NavChip IMU is not as accurate as STIM318, its efficacy to produce similar navigation accuracy in a post-processing scenario has been demonstrated by an independent comparison with i) photogrammetry on a drone (Rehak & Skaloud, 2015) and ii) navigation-grade INS on a helicopter flying at speeds comparable to those of a drone (Clausen & Skaloud, 2020).

<sup>4</sup>Inertial Explorer, ver 8.5-9, NovAtel Inc., 2020

<sup>5</sup>POSProc, ver. 2.1.4, Applanix Corp., 1999

## 5 | RESULTS

In this section, the proposed aerodynamic calibration methodology is validated by means of rigorous experimentation. Following the aforementioned experimental workflow for *TP2*, during the calibration phase, we first present empirical evidence relating our heuristic of uniqueness to the observability of the parameters of interest. Next, we compare some of the IMU observations with those obtained using calibrated model parameters. Then, we present the performance of these calibrated parameters in an application flight. Additionally, we evaluate the invariance of calibrated model parameters to different IMUs (STIM 318 and ADIS-16475). Subsequently, we study the performance of uncalibrated (or random) model parameters in a VDM-based navigation system. Finally, the entire methodology is repeated for a delta-wing platform (*eBeeX*), and the obtained results are presented as a proof of concept. This experimental workflow is summarized in Figure 8.

### 5.1 | Evolution of the Observability Gramian of the Wind Estimator

We present empirical and graphical evidence pertaining to the evolution of the observability Gramian in an experimental setting. To the best of our knowledge, this is the first time that such results have been presented in this context within the domain of navigation. To present these findings, we chose to elucidate the wind estimator because of its simplicity and familiarity in the scientific community.

The procedure detailed in Section 3 is implemented for each estimator. For the wind estimator, the dimensionality of the state space is four ( $j = 4$ ) and  $\mathbf{H}_k \in \mathbb{R}^{1 \times 4}$ ,

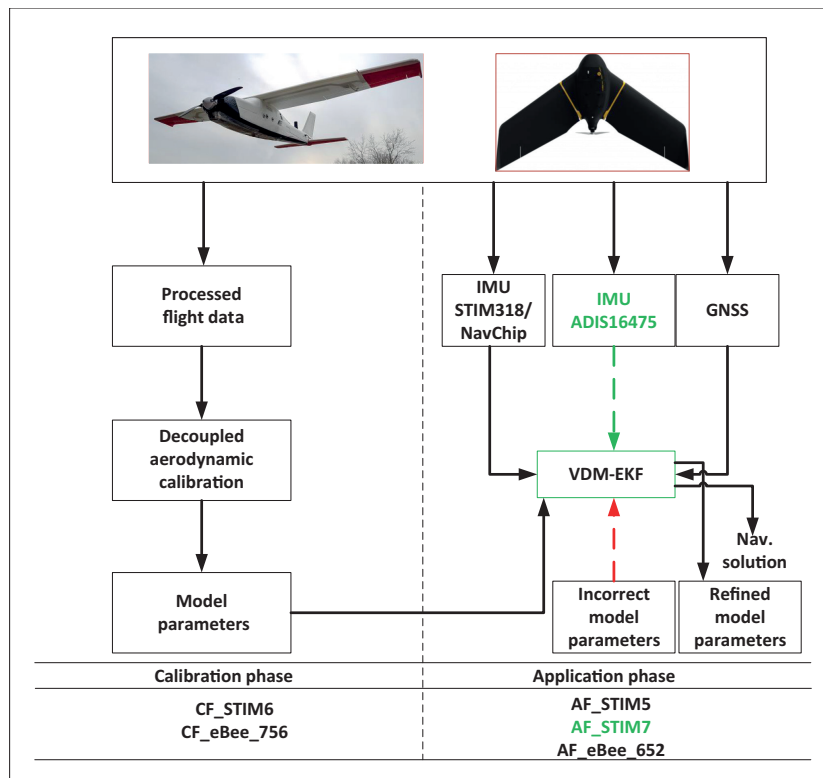


FIGURE 8 Experimental pipeline

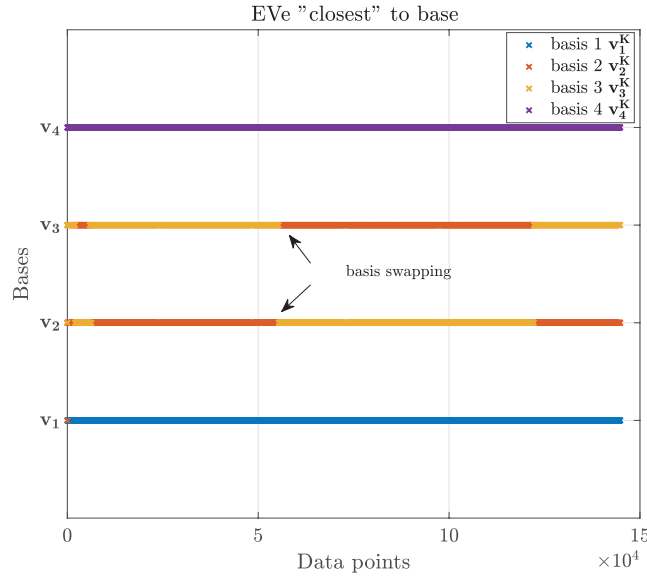


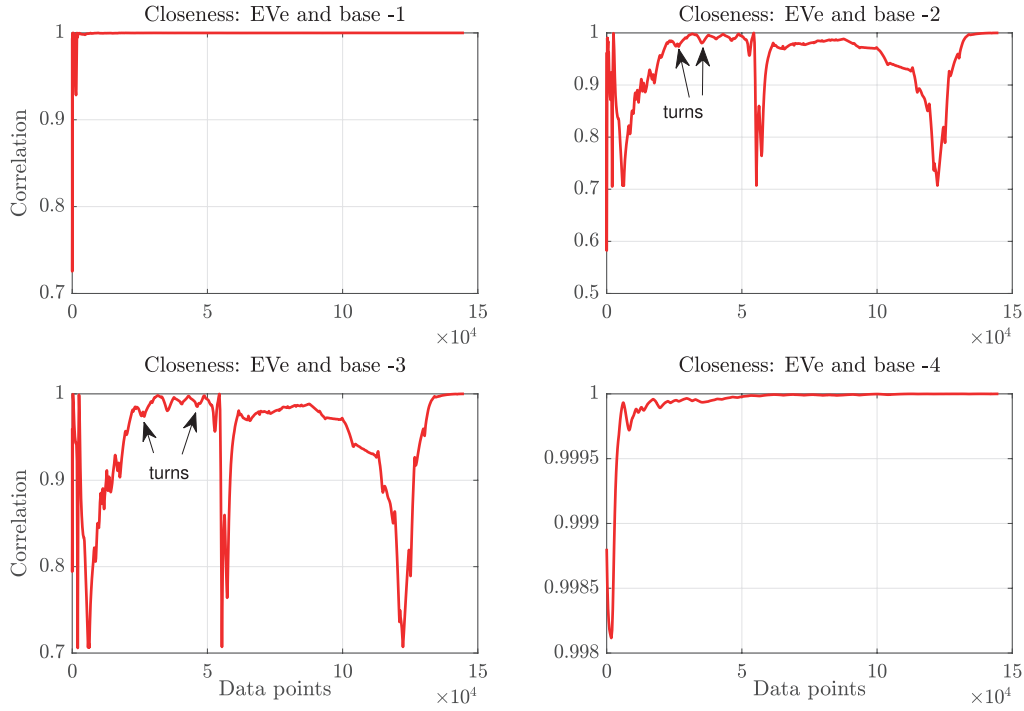
FIGURE 9 Tracking the eigenvector (EVe) of the observability Gramian

whereas the observability Gramian  $\mathbf{W}_0(k) \in \mathbb{R}^{4 \times 4}$ . Upon calculating the eigenvalues and eigenvectors of  $\mathbf{W}_0(k)$  after all observations  $k \in \{1, 2, \dots, K\}$ , we choose eigenvector of  $\mathbf{W}_0(K)$  as the basis of the state space. It should be noted that  $\mathbf{W}_0(k)$  is symmetric and the eigenvector matrix  $\mathbf{V}_k$  is orthonormal, thereby making it a rotation matrix in  $\mathbb{R}^4$ . Subsequently, the closeness matrix ( $\Gamma_k$ ) is computed as a dot product between  $\mathbf{V}_k$  and the basis ( $\mathbf{V}_K$ ). This step is followed by the  $\max(\Gamma_k)$  operation. This operation determines which eigenvector of  $\mathbf{W}_0(k)$  is most closely aligned with the basis vector. For the wind estimator, this evaluation of the closest eigenvector is reported in Figure 9, and the corresponding evolution of the numerical value  $\max(\Gamma_k)$  is shown in Figure 10.

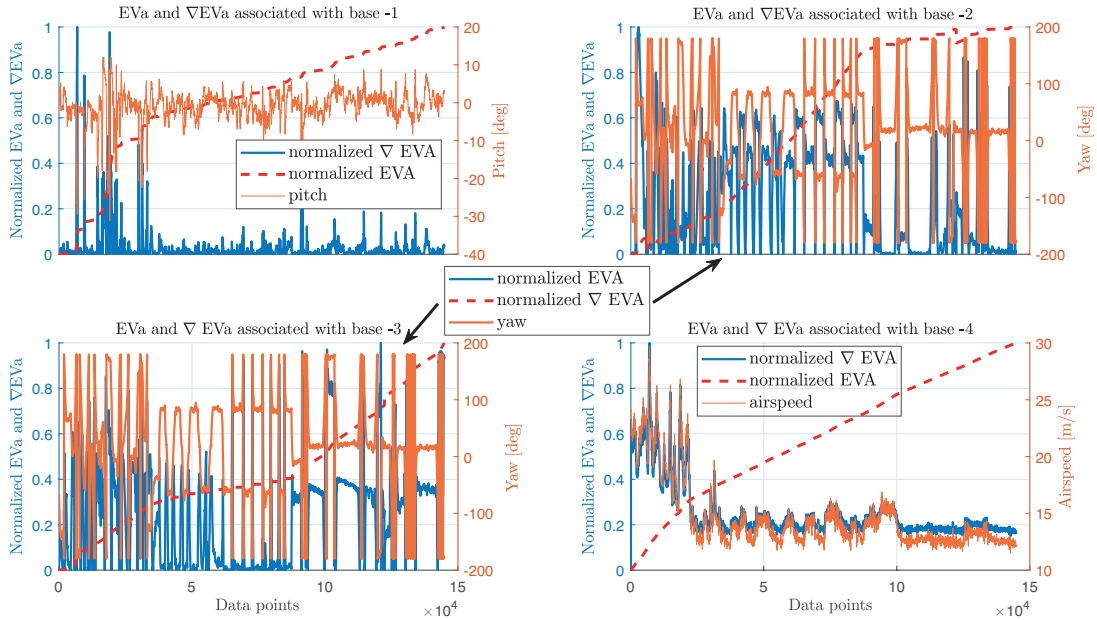
As  $\mathbf{V}_k^T \mathbf{V}_K = \mathbf{I}$  for  $k = K$ , the value of  $\max(\Gamma_k) = [1 \ 1 \ 1 \ 1]^T$ . This can be clearly seen in Figure 10, wherein all of the trends eventually converge to unity. As a result, at  $k = K$ , the first (second and so on) basis vector is the same as the first (second and so on) eigenvector of  $\mathbf{W}_0(k)$ . This can be seen in Figure 9, wherein all of the trends converge to the index of the basis vector. The plot in Figure 9 is a special graphical representation in which i) the x axis  $\in \{1, 2, \dots, K\}$  accounts for the discrete-time events at which airspeed measurements are made and ii) the y axis  $\in \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$  represents the eigenvectors of  $\mathbf{W}_0(k)$ , each made up of one of four colors (blue, red, yellow, purple). As discussed earlier, the eigenvalues and eigenvectors of  $\mathbf{W}_0$  are calculated for each discrete-time event, and then their association to the basis is established (using the closeness metric). In this graph, each basis vector is color-coded, and each  $\mathbf{v}_j$  is represented as a *quantized* time series of these colors. For instance, the time series  $\mathbf{v}_1$  consists primarily of basis vector 1 (blue) except for a few samples of basis vector 2 (red) at the very beginning.

In this way, the eigenvalues of  $\mathbf{W}_0(k) \forall k$  are approximately associated with basis vectors. The evolution of these eigenvalues (red dashed line) is shown in Figure 11, which displays a generally growing trend, except for certain small variations in basis vectors 2 and 3, corresponding to a swapping of basis.

In Figure 11 (red), we present the normalized incremental growth (IG) of the eigenvalue using Equation (44). This IG is then compared with the drone's attitude and Pitot measurement, collectively shown by the right-hand axis of Figure 11.



**FIGURE 10** Maximum closeness in wind observability  
EVE: eigenvector.



**FIGURE 11** Evolution of eigenvalues (EVA) associated with the basis

We observe the following:

1. IG along basis vector 1 is correlated to pitch: The peaks (both positive and negative) in pitch lead to a nonzero IG along basis vector 1. However, this IG is mostly close to zero as the drone performs fewer pitching maneuvers.
2. IG along basis vectors 2 and 3 is correlated to yaw: The nonzero IG peaks along these basis vectors alternate with changes in heading. Moreover, the IG shows

complementary trends, i.e., a peak in one basis vector corresponds to a sharp decrease in the other.

3. IG along basis vector 4 is correlated to Pitot measurements: This IG is always greater than zero.

These observations are consistent with findings of previous works (Johansen et al., 2015) showing that i) a drone must change its heading for the horizontal wind to be observable, ii) pitching maneuvers are essential for better observability of down wind, and iii) the Pitot scale factor is always observable as long as the drone is airborne.

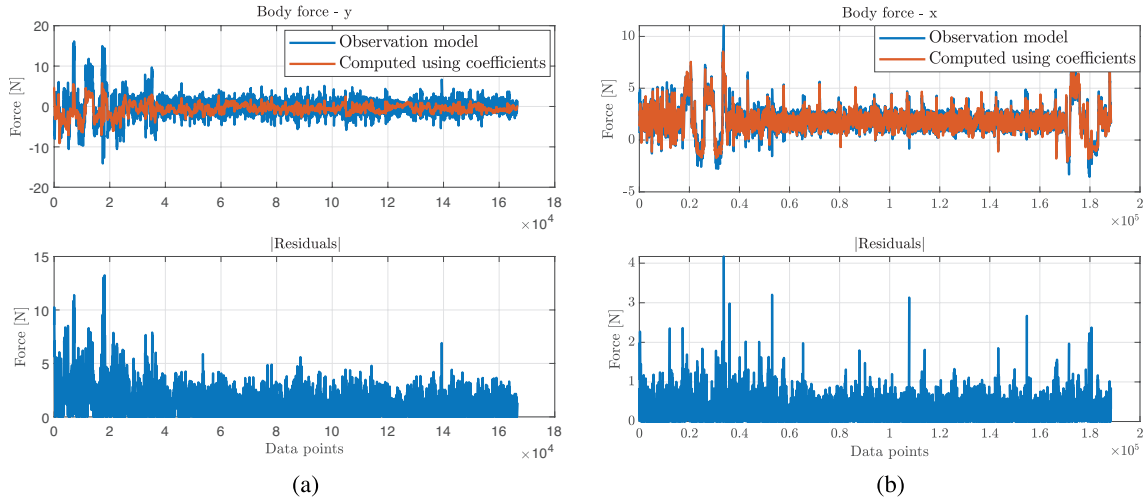
By following this methodology, it is possible to approximately track the growth of eigenvalues of the observability Gramian, rather than evaluating these values only after the last observation, as in Johansen et al. (2015). The IG along basis vectors is primarily responsible for observability. For the wind estimator, it appears that i) basis vector 1 corresponds to down wind, ii) basis vectors 2 and 3 correspond to horizontal wind, and iii) basis vector 4 corresponds to the Pitot scale factor. This interpretation is intuitive, thanks to a low-dimensional state space; however, making such one-to-one correspondences for a high-dimensional state space may not always be possible (as in the case of moment and force parameters). Nevertheless, the mathematics of the methodology prove effective in discerning the most observable states from the rest after each observation. In the next subsection, the same methodology is utilized in computing the aerodynamic model parameters associated with moments and forces.

Note: There exist certain observations that are not accepted for estimation because of the following reasons: i) Non-uniqueness of  $\max(\Gamma_k) \rightarrow$  it was found that in some situations, two eigenvectors are close to the same basis vector, leading to an eventual reduction in the dimensionality of tracked eigenvector space, and ii) transitioning from one closest eigenvector to the other, implying sharp peaks in IG, which do not necessarily correspond to observability. The causes of both situations seem to be related to the empirical nature of the algorithm and observation noise. However, such data points are easy to detect from a software viewpoint and are hence excluded from the estimation. Collectively, such cases form less than 0.5% of the total number of observations for all three estimators.

## 5.2 | Aerodynamic Model Parameters

Following the partial-update methodology presented in Section 3, we sequentially estimate wind, followed by moment, and finally force parameters using sensor data from the calibration flight. To do so, we follow the recommendations of Laupré & Skaloud (2021) in terms of making use of GNSS position and velocity derived from PPK data. Additionally, we make use of an infinite impulse response smoother with a cutoff frequency of  $f_c = 30$  and 40 Hz (from visual data inspection using a fast Fourier transform) for the gyroscope and accelerometer, respectively. This approach eliminates high-frequency noise from inertial sensors (which would affect subsequent numerical differentiation), while precluding any time delays. The signal quality of the onboard *smoothed* IMU (STIM318) is far superior to that of the autopilot's IMU.

We then compare these IMU observations to those obtained using calibrated parameters. For this comparison, we choose to focus on forces, as the forces are implicitly dependent on moments and wind because of the cascaded design of the estimation framework. Calculating forces using accelerometer measurements is a



**FIGURE 12** (top) Juxtaposition of the  $y$  force computed using the IMU and calibrated model and their residuals (bottom) for (a) the  $y$  force for *TP2* and (b) the  $x$  force for *eBeeX*

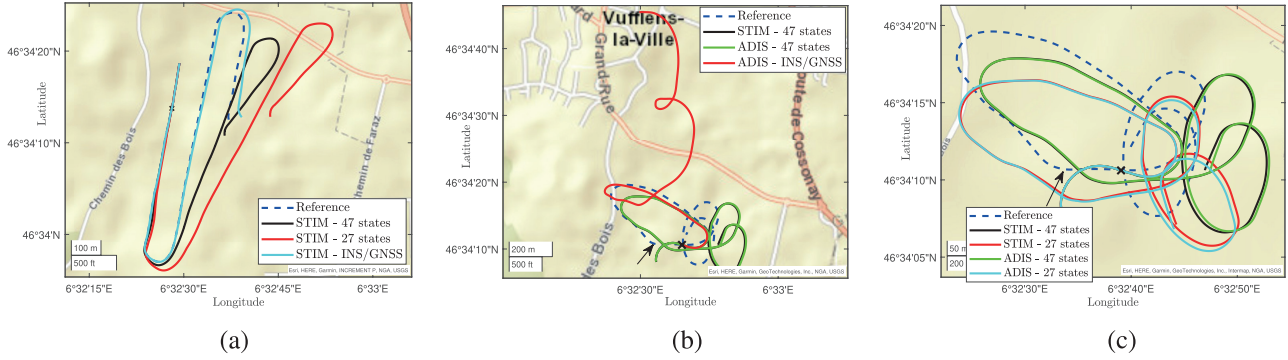
straightforward operation involving multiplication by the mass of the drone. We graphically show one of the estimated quantities—body force along the  $y$  axis—against the value computed using the IMU in Figure 12(a). This figure clearly shows that the force estimated by the model parameters qualitatively follows trends similar to those computed directly from the IMU. Additionally, the estimated forces appear to be filtered versions of IMU data.

### 5.3 | VDM-Based Navigation

In this section, we use the aerodynamic model parameters obtained from Section 5.2 in the VDM-based navigation framework (Khaghani & Skaloud, 2018) for an application flight (different from that of the calibration flight). We use two possible architectures of aerodynamic sensor fusion, depending on how the obtained model parameters are used:

1. **47-state estimator:** When model parameters are included in the state vector of the EKF-VDM framework, as in Khaghani & Skaloud (2018) and Lauré & Skaloud (2021), with their *a priori* estimates provided by the proposed calibration strategy. This results in a 47-dimensional state space, as discussed in Section 2.2.1.
2. **27-state estimator:** When model parameters are considered as known constants, with their values provided by the proposed calibration strategy. This results in a 26-dimensional state vector of the EKF-VDM framework. However, a new state referred to as the aerodynamic scale factor is introduced to compensate for changes in atmospheric conditions between the calibration and application flights. This scale factor is multiplied by the dynamic pressure ( $\bar{q}$ ) for the calculation of forces and moments in the VDM-based process and observation models. This approach effectively results in a 27-dimensional state space.

Both of the aforementioned approaches yield encouraging results and mitigate navigation drift during a GNSS outage, as shown in Figure 13(a) for application flight *AP\_STIM5* and Figure 13(c) for application flight *AP\_STIM7*. Figure 13(c) also



**FIGURE 13** *TP2* flight *AF\_STIM5* (a) with an outage after 15 min and flight *AF\_STIM7* (b) with an outage after 4 min and (c) invariance of the calibrated VDM to different IMUs. The GNSS outage starts at the black cross.

portrays the use of a considerably smaller (and less expensive) IMU (ADIS-16475), which is discussed later in Section 5.4. It should be noted that the drone platform for application flight *AP\_STIM5* differs from that of the calibration flight and application flight *AP\_STIM7* because of hardware modifications.

#### 5.4 | Model Invariance to Choice of IMU

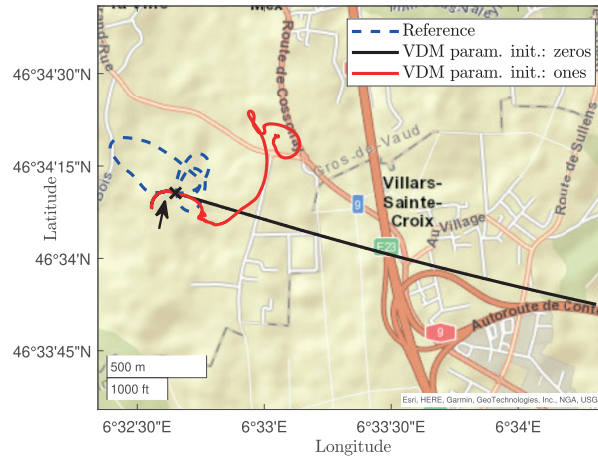
We compare the navigation solution obtained by fusing the identified aerodynamic parameters with two different IMUs: i) STIM-318 and ii) ADIS-16475. For this comparison, we treat the data from application flight *AF\_STIM7* using both of the aforementioned architectures (47/27-state estimator). The outcome, presented in Figure 13(c), shows that the navigation solutions for the two IMUs are similar during a GNSS outage. This result indicates that dynamics identified from a high-grade IMU, using the proposed methodology, can be *prima facie* fused with measurements from a lower-grade inertial sensor to obtain similar results. In contrast, if ADIS is used in an INS/GNSS-driven navigation system, the trajectory obtained during a GNSS outage quickly drifts away from the reference and deviates from the true trajectory by 1 km at the end of 2 min. This result is graphically shown in Figure 13(b) in red with the legend “ADIS-INS/GNSS.”

#### 5.5 | Model Parameter Initialization in the EKF Framework

After having analyzed the goodness of the obtained model parameters for *TP2* in the previous section, we shift our focus to studying the influence of these parameters on state estimation when our initial knowledge is of insufficient quality.

To emphasize the importance of the correct initialization of aerodynamic parameters in a real scenario, different initial sets of parameters were tested in the VDM-based navigation system. These tests were carried out on data from application flight *AF\_STIM7* with a 47-dimensional state vector. The initialization sets used in this test are as follows: (i) all ones, (ii) all zeros (approximated to  $10^{-7}$ ), and (iii) all random values with  $N \sim (0, 1)$ . The effect of these incorrect initial parameters was assessed during a simulated 2-min GNSS outage after 4 min of flight. The obtained results are shown in Figure 14. The scenario in which the VDM parameters are initialized with all zeros results in a straight-line (black) trajectory because





**FIGURE 14** A 2-min GNSS outage after 4 min with intentionally wrong initial VDM parameters for flight *AF\_STIM7*

the model does not produce tangible moments or forces. The second scenario in which the initial values are set to ones (red) is comparable to free INS. The last scenario is not visible on the plot, as numerical instability occurred within 1 s. Clearly and in comparison to Figure 13(c), the communicated methodology outperforms randomly initialized parameters in a model-based navigation framework.

The results obtained with incorrect initialization are consistent with existing simulation studies in a more general framework (Laupré & Skaloud, 2020). For the sake of completeness and to highlight the relevance of the proposed methodology, we briefly review the major findings of the sensitivity studies presented in Laupré & Skaloud (2020):

1. The initialization of unknown model parameters must be below 40% of their true value to prevent filter divergence and/or software failure (due to numerical instability).
2. In addition to the threshold of 40% mentioned above, there is a need for sufficient dynamics to reasonably estimate these unknown parameters in flight.

The consequences of these erroneously estimated model parameters are manifold. First, the incorrect estimated parameters lead to improper forces and moments, which are subsequently compensated for by other states such as wind, IMU biases, or even large errors in drone velocity, as mentioned in Laupré & Skaloud (2021). In the case of a GNSS outage, the solution becomes rapidly unusable and potentially worse than a free INS solution. Another issue with incorrect VDM parameters is the possible numerical instability of the filter, which can potentially lead to software failure.

## 5.6 | Validation with eBeeX

The entire calibration and application procedure was repeated for a delta-wing drone. However, here, we skip the repetitive details and present the major findings succinctly.

We begin by graphically comparing the estimated body force along the  $x$  axis against the force computed using the IMU. This comparison is presented in

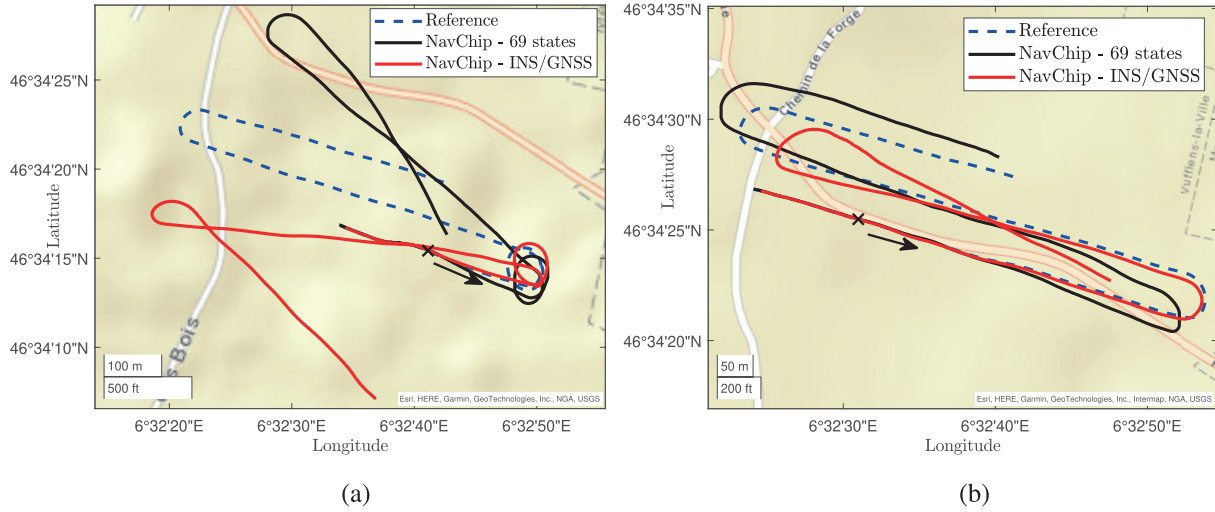


FIGURE 15 *eBeeX* application flight *AF\_eBee\_652* with a 2-min outage after (a) 24 or (b) 28 min

Figure 12(b). Because the force model of *eBeeX* contains a large number of parameters compared with *TP2*, the two signals match better and the residual is lower, confirming the correctness of model identification with respect to the observations. However, these results do not necessarily guarantee better navigation performance. This aspect is further discussed in the following paragraph and later in Section 6.1.

To validate the goodness of the obtained parameters, the application flight was processed two times with GNSS outages simulated after 24 and 28 min. For this test, the model parameters are included in the state, thereby yielding a 69-dimensional state vector (as discussed in Section 2.2.1). The *a priori* estimate of these parameters is obtained using the proposed calibration strategy. The navigation results are portrayed in Figure 15(a) and Figure 15(b) alongside the INS solution with IMU data being pre-calibrated; the deterministic noise error was removed thanks to Clausen (2019). In contrast, the 26-dimensional estimator, which considers model parameters to be constant, did not yield encouraging results. A plausible reason for such a result, based on the following evidence, seems to be over-fitting: (i) low residual error during calibration, (ii) a large number of unknown parameters, and (iii) large navigation drift/numerical instability when model parameters are held fixed during the application phase. It is also important to highlight that in Laupré et al. (2021), the trajectory used for calibration and application is one and the same, while independent testing is enforced here.

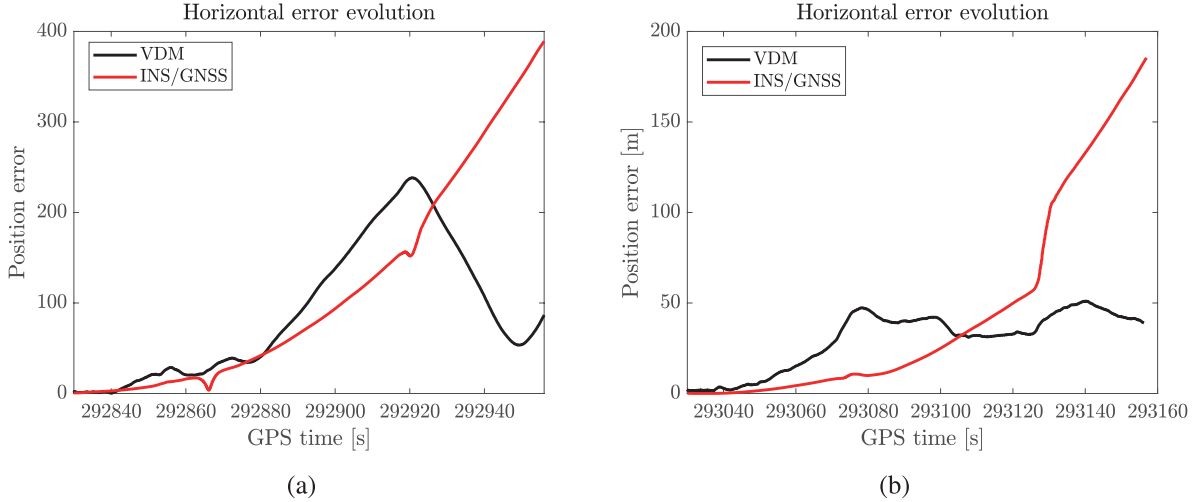
## 5.7 | Performance Summary

We present the performance of calibrated aerodynamic model parameters during GNSS outages in Table 4. It should be noted that the navigation error at the end of the outage under the header “VDM” in Table 4 corresponds to an estimator consisting of 47 (for *TP2*) or 69 (for *eBeeX*) states. Additionally, navigation error curves for the two outages of *AF\_eBee\_652* are portrayed in Figure 16(a) and Figure 16(b), together with certain statistical indicators listed in Table 5. The obtained results show that VDM-based navigation significantly outperforms conventional sensor fusion for the small MEMS IMU, ADIS, used on *TP2*. In contrast, STIM318,

**TABLE 4**

Summary of Calibration Performance for Different Flights and IMUs in a VDM-EKF Framework

Flight	Outage after (minutes)	Outage duration (minutes)	IMU	Error at the end of outage (m)	
				INS/GNSS	VDM
<i>AF_STIM5</i>	15	02	STIM 318	44	41
<i>AF_STIM7</i>	04	02	ADIS 16475	1062	155
<i>AF_eBee_652</i>	24	02	NavChip v1	394	88
<i>AF_eBee_652</i>	28	02	NavChip v1	185	39

**FIGURE 16** Horizontal error evolution during a 2-min GNSS outage of *AF\_eBee\_652* after (a) 24 or (b) 28 min**TABLE 5**Error Statistics for *AF\_eBee\_652*

RMS: root mean square.

Outage after [min]	RMS error [m]		Median error [m]		Mean error [m]	
	INS/GNSS	VDM	INS/GNSS	VDM	INS/GNSS	VDM
24	171	120	85	73	124	95
28	74	35	22	35	48	32

which can almost be classified as a tactical-grade IMU (costing ~\$7500), is used for aerodynamic calibration and obtaining high-precision reference navigation data. A recent work (Paul et al., 2023) has suggested that using VDM is beneficial for lower-quality IMUs. This finding is consistent with the obtained results, wherein VDM does not yield similar navigation results for the high-quality IMU (STIM318) for the chosen trajectory/outage duration. Yet, based on the navigation error curve for *eBeeX* shown in Figure 16, the error at the end of the outage in Table 4, and the statistics listed in Table 5, the long-term performance of the VDM-EKF framework is better than that of INS/GNSS for the MEMS IMU, NavChip, whereas INS/GNSS performs better in the short term. It should be noted that these results are based on the choice of the aerodynamic model (Laupré et al., 2021), which is most likely over-parameterized for a drone of this shape; this aspect is briefly discussed above in Section 5.6 and later in Section 6.1.

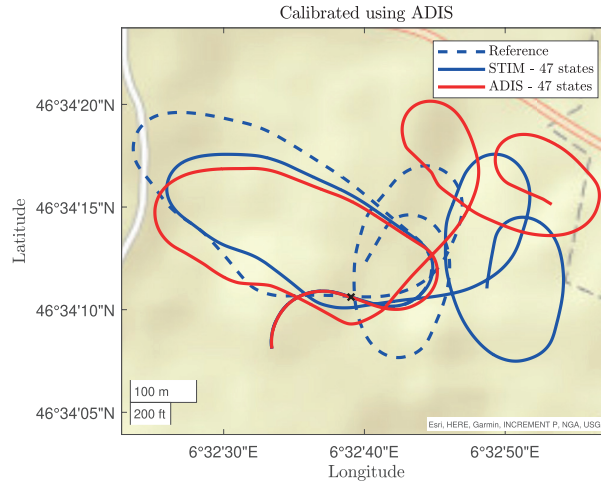


FIGURE 17 Testing model parameters, identified using ADIS, in the VDM-EKF framework for *AF\_STIM7*

## 5.8 | Calibration with a Lower-Quality MEMS IMU

We identified the aerodynamic model parameters of *TP2* using ADIS 16475, as opposed to STIM 318 in previous investigations, yet on the same flight, namely, *CF\_STIM6*. Subsequently, we tested the identified parameters on the application flight, *AF\_STIM7*, by fusing the parameters for both STIM and ADIS separately. The results are shown in Figure 17, whereas the results obtained by calibration using STIM have already been presented in Figure 13(c). Our findings reveal that calibration using the higher-grade IMU results in better test performance of the VDM-EKF with the lower-grade IMU, compared with calibration using the lower-grade IMU. Conversely, calibration using the lower-grade IMU does not result in as good a test performance as the former, although this approach still performs better than the free INS (as shown in Figure 13(b)). It should be noted that while our preliminary results are promising, further statistical analysis is needed to provide stronger evidence for using our methodology with different grades of IMUs. We recognize that the evaluation of IMU grade is an important aspect of future research in this area, but we do emphasize this aspect in this paper.

## 6 | ADDITIONAL DISCUSSION

In this section, certain nuances associated with the proposed methodology are discussed, and references to further reading are provided wherever possible.

### 6.1 | Aerodynamic Modeling

Although not stated explicitly, the presented mathematical formulations for identifying the aerodynamics of drones are dependent on the choice of model. In this article, we have relied on already existing functional models, be it for *TP2* or *eBeeX*. However, while carrying out the calibration procedure, we observed that there are certain basis vectors along which there is hardly any significant growth in the eigenvalues of the observability Gramian. Although we could have chosen the path of eliminating certain parameters, we opted instead to infrequently update

the parameters in the partial-update framework. Otherwise, this opens a new research direction—model selection—which is outside the scope of this article. This appears to be largely true for *eBeeX*, whose aerodynamic model comprises 44 parameters. At the end of the calibration phase, the mean residual error for the specific forces and moments of *eBeeX* is found to be much lower than that of *TP2*. Yet, when these parameters are held fixed in an application flight, *TP2* clearly outperforms *eBeeX*. However, for *TP2*, there seems to be a scope to reduce residual error by including more parameters while simultaneously ensuring that there is no over-fitting. **It is also important to note that if the residual error were zero, then the VDM would converge to the INS, which is not desirable except for an IMU of excellent quality (e.g., navigation grade).**

In addition to the choice of a functional model, it is also important to ensure that the parameters associated with the model are observable. This observability is closely related to the trajectory of the platform, which should satisfy the conditions of persistence of excitation. Such an excitation heavily relies on drone dynamics, which (in the function of guidance and planning) may be difficult to achieve in fully automated missions. For *TP2*, this condition is achieved by manual control, whereas this option is not available for the commercial platform, *eBeeX*. This difference is evident from Figure 5 and Figure 6, where *TP2* trajectories have significantly more dynamics than *eBeeX* trajectories.

## 6.2 | Alternative Methods

In this section, we comment on the benefits of the proposed methodology over alternative approaches. The existing strategies have largely addressed aerodynamic modeling in a VDM-based navigation system by pursuing the following approaches.

1. Using prior knowledge of the platform's aerodynamics followed by further in-flight refinement of model parameters, as in Khaghani & Skaloud (2018), Laupré et al. (2021), and Laupré & Skaloud (2021). However, this knowledge may not always be available for every drone. We have shown that a random initialization of these parameters in a filtering framework leads to filter divergence (or software failure). Therefore, a strategy for systematically identifying a set of working aerodynamic parameters from flight data serves as an attractive alternative for scenarios in which very limited prior knowledge of the platform is available.
2. Using CFD and/or wind tunnel experimentation to obtain a working set of aerodynamic parameters, as in Mwenegoha et al. (2020). Despite being rigorous, these approaches are setup-expensive and time-consuming and require specific expertise to obtain tangible results. Here, we have demonstrated excellent navigation performance by making use of model parameters obtained solely using available flight data. The proposed approach serves as an inexpensive and efficient substitute for wind-tunnel/CFD-based aerodynamic characterization used in model-based navigation.

## 6.3 | Parameter Initialization

In this article, we claim that our methodology does not require an initial guess of model parameters for the purpose of calibration. However, all of the presented

estimation equations are purely recursive. This might be tantamount to a need for an initial guess unless it can be systematically computed. We follow the latter approach in order to free the methodology from the burden of *prior knowledge* by making use of the observability Gramian and linear regression. The developed calibration software chooses certain observations, equaling the number of states (10/11 for *TP2*, for instance), corresponding to the highest IG of eigenvalues along each basis vector for running classical (weighted) linear regression:

$$\mathbf{x}_1 = [\mathcal{H}^T \mathbf{R}^{-1} \mathcal{H}]^{-1} \mathcal{H}^T \mathbf{R}^{-1} \mathcal{Z} \text{ with, } \mathcal{H} = \begin{bmatrix} \mathbf{H}_{e_1} \\ \vdots \\ \mathbf{H}_{e_n} \end{bmatrix}, \mathcal{Z} = \begin{bmatrix} \mathbf{z}_{e_1} \\ \vdots \\ \mathbf{z}_{e_n} \end{bmatrix}$$

where the cardinality of  $\{e_1, e_2, \dots, e_n\}$  is equal to the dimension of the state space. The subscript  $e_j$  denotes the most *exciting entity* for basis vector  $j$ . In other words,  $\mathbf{H}_{e_j}$  and  $\mathbf{z}_{e_j}$  denote the observation matrix and the measurement for which the highest IG of eigenvalues is observed along basis vector  $j$ .

This approach provides an initial guess to start off the estimators. It should be noted that if the first  $n$  temporal observations are selected (assuming that  $n$  is sufficiently small, as above), there is no guarantee that the regressor matrix will be full rank, thereby causing the entire framework to break down. This phenomenon was indeed observed during the rudimentary phase of this research. Thus, we recommend choosing these observations via a reliable metric, as described above, to preclude implementation-related problems. Moreover, by means of independent testing, we found that random initialization of these parameters in the EKF-VDM framework leads to filter divergence or complete breakdown of the software for both platforms. **Therefore, either a priori knowledge or an independent methodology for obtaining these parameters a priori is indispensable to run the VDM-based navigation system.**

## 6.4 | Comparison With Existing Methods

The presented methodology takes inspiration from several widely known concepts, including RLS (Gelb, 1974), Kalman filtering (Bryson & Sukkarieh, 2015), Schmidt–Kalman filtering (Brink, 2017), partial-update frameworks (Brink, 2017), principal component analysis (Moore, 1981), the observability Gramian (Chen, 1999), the accumulation of normal equations (Gelb, 1974), and cascaded estimators (Haupt et al., 1996). In this section, we emphasize the similarities/differences between some of the aforementioned existing methodologies and the proposed technique.

In Haupt et al. (1996), the authors formally propose a two-step estimator to handle nonlinearity in a state estimation framework. Standard nonlinear recursive estimators, such as the EKF, linearize the cost function in order to use the widely known linear Kalman filter equations. This approach results in an estimate that is biased while simultaneously rendering the estimation schema very sensitive to *a priori* estimates of unknown states (about which the cost function is linearized) (Haupt et al., 1996). This extremely important finding validates our choice to i) use three linear estimators for obtaining unbiased estimates of aerodynamic parameters and ii) perform a rudimentary sensitivity analysis of *a priori* unknown model parameters in a VDM-based navigation filter. Although our work differs from the research presented in Haupt et al. (1996), both approaches deal with nonlinearity using a cascaded framework.

State-space representation of a system is not unique, and there are infinitely many representations. This feature is widely referred to as state-space invariance to a change of basis. In this work, a special basis set (the eigenvectors of the observability Gramian) is used to build a partial-update framework that is updated using a metric based on the evolution of the observability Gramian. We have relied on Khaghani & Skaloud (2018), Brink (2017), and Johansen et al. (2015) to determine that the state estimation is trajectory-dependent and that a part of the state vector is mildly observable. We have also relied on Brink (2017) to understand how conventional filter updates of these mildly observable states may lead to filter divergence/inconsistency unless a partial-update framework is employed. Therefore, in sum, we have made use of well-established *prior art*; however, the proposed holistic reorganization of this *art* in a practical setting for model-based navigation is our original contribution to the scientific community.

## 6.5 | Assumptions, Advantages, and Limitations

The proposed methodology relies on the following assumptions: i) existence of an aerodynamic model that is linear in parameters and ii) a knowledge of geometric parameters of the platform.

Conversely, this methodology comes with the following advantages: i) The proposed algorithm facilitates a simple and scalable framework that takes recorded flight data as input and outputs usable model parameters. ii) The calibrated model parameters present the potential to significantly reduce the state-space dimension of the VDM-navigation filter from 47 to 27 for *TP2*, thereby lowering the computational burden. iii) The algorithm is proven to be generalizable and can be extended to different drone geometries.

The algorithm, in its current form, cannot calibrate in real time (while the drone is airborne) because of its reliance on i) smoothed data, ii) available PPK GNSS data, and iii) the existence of the observability Gramian until the last observation for the partial update. However, as the purpose of the methodology is to calibrate, its advantages significantly outweigh its limitations. In other words, a fixed-wing drone can be flown once to collect the necessary flight data for offline aerodynamic calibration. Subsequently, the calibrated aerodynamic model can be directly used for online real-time flight tests. It is worth mentioning that Brink (2017) proposes a real-time metric for conducting partial updates by obtaining a *proxy* of information content:

$$\mathbf{I}_{\text{EKF}} = \tilde{\mathbf{P}}^{-1} \mathbf{K} \mathbf{R}^{-1} \mathbf{K}^T \left[ \tilde{\mathbf{P}}^{-1} \right]^T$$

However, no literature-based backing is provided regarding the origin of this metric, nor is any empirical/mathematical proof presented. Nevertheless, the approach presents very encouraging results, in terms of preventing filter divergence (albeit in a purely simulated setting).

## 7 | CONCLUSION

In this article, we have presented a methodology for identifying aerodynamic model parameters of small fixed-wing drones, solely from recorded flight data, for applications in model-based navigation. The analysis presented in this article shows that a reasonably good *a priori* estimate of aerodynamic model parameters is indispensable for implementing a model-based navigation filter. Failure to

correctly initialize these parameters results in filter divergence or, worse, software failure. The parameters obtained from the proposed calibration algorithm were successfully tested and validated on two different fixed-wing drone platforms during GNSS outages. In sum, this methodology has been shown to be independent of i) the fixed-wing drone geometry and ii) prior knowledge of aerodynamic model parameters. Because of these two major advantages, the presented work may serve as an inexpensive substitute for wind tunnel experimentation and fluid dynamics simulations for model-based navigation. Moreover, the obtained calibration results show a significant improvement in autonomous navigation with respect to inertial coasting for low-quality MEMS IMUs.

## ACKNOWLEDGMENTS

i) We thank Dr. Mohammad Abedini, Kenneth Joseph Paul, and Didier Negretto, colleagues from the Geodetic Engineering Laboratory, EPFL, for their contribution in hardware development. ii) This project has received funding from the European Union's Horizon 2020 Research and Innovation Program under the Marie Skłodowska-Curie grant agreement No. 754354. iii) This work was partially supported by the Swiss National Foundation (200021182072).

## REFERENCES

- Albrektsen, S. M., & Johansen, T. A. (2018). User-configurable timing and navigation for UAVs. *Sensors*, 18(8), 2468. <https://doi.org/10.3390/s18082468>
- Alcántara, A., Capitán, J., Torres-González, A., Cunha, R., & Ollero, A. (2020). Autonomous execution of cinematographic shots with multiple drones. *IEEE Access*, 8, 201300–201316. <https://doi.org/10.1109/ACCESS.2020.3036239>
- Brink, K. M. (2017). Partial-update schmidt-kalman filter. *Journal of Guidance, Control, and Dynamics*, 40(9), 2214–2228. <https://doi.org/10.2514/1.G002808>
- Bryson, M., & Sukkarieh, S. (2015). UAV localization using inertial sensors and satellite positioning systems. *Handbook of Unmanned Aerial Vehicles*, 433–460. [https://doi.org/10.1007/978-90-481-9707-1\\_3](https://doi.org/10.1007/978-90-481-9707-1_3)
- Chen, C. (1999). *Linear system theory and design*. Oxford University Press.
- Clausen, P. (2019). Calibration aspects of INS navigation (Doctoral dissertation, EPFL). Restricted to EPFL only <https://infoscience.epfl.ch/record/264793>
- Clausen, P., & Skaloud, J. (2020). On the calibration aspects of MEMS-IMUs used in micro UAVs for sensor orientation. *Proc. of the IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Portland, Oregon, 1457–1466. <https://doi.org/10.1109/PLANS46316.2020.9110160>
- Cledat, E., Jospin, L., Cucci, D., & Skaloud, J. (2020). Mapping quality prediction for RTK/PPK-equipped micro-drones operating in complex natural environment. *ISPRS Journal of Photogrammetry and Remote Sensing*, 167, 24–38. <https://doi.org/10.1016/j.isprsjprs.2020.05.015>
- Cork, L. R. (2014). *Aircraft dynamic navigation for unmanned aerial vehicles* (Doctoral dissertation, Queensland University of Technology). <https://eprints.qut.edu.au/71396/>
- Dsouza, R. D., Salim, S., Shankar, A., Safwan, M., & D'sa, S. (2016). Wind tunnels: State of art survey and future scope for testing micro air vehicles. *American Academic Scientific Research Journal for Engineering, Technology, and Sciences*, 19(1), 25–41. [https://asrjetsjournal.org/index.php/American\\_Scientific\\_Journal/article/view/1627](https://asrjetsjournal.org/index.php/American_Scientific_Journal/article/view/1627)
- Ducard, G. J. (2009). Nonlinear aircraft model. *Fault-tolerant Flight Control and Guidance Systems: Practical Methods for Small Unmanned Aerial Vehicles*, Springer London. 27–41. <https://doi.org/10.1007/978-1-84882-561-1>
- Erdelj, M., Natalizio, E., Chowdhury, K. R., & Akyildiz, I. F. (2017). Help from the sky: Leveraging UAVs for disaster management. *IEEE Pervasive Computing*, 16(1), 24–32. <https://doi.org/10.1109/MPRV.2017.11>
- Gelb, A. (Ed.). (1974). *Applied optimal estimation*. M.I.T. Press. Retrieve from <https://mitpress.mit.edu/9780262570480/applied-optimal-estimation/>
- Green, M., & Moore, J. B. (1986). Persistence of excitation in linear systems. *Systems & Control Letters*, 7(5), 351–360. [https://doi.org/10.1016/0167-6911\(86\)90052-6](https://doi.org/10.1016/0167-6911(86)90052-6)
- Haupt, G. T., Kasdin, N. J., Keiser, G. M., & Parkinson, B. W. (1996). Optimal recursive iterative algorithm for discrete nonlinear least-squares estimation. *Journal of Guidance, Control, and Dynamics*, 19(3), 643–649. <https://doi.org/10.2514/3.21669>
- Hayat, S., Yanmaz, E., Bettstetter, C., & Brown, T. X. (2020). Multi-objective drone path planning for search and rescue with quality-of-service requirements. *Autonomous Robots*, 44, 1183–1198. <https://doi.org/10.1007/s10514-020-09926-9>



- Hiebert, B., Nouvet, E., Jeyabalan, V., & Donelle, L. (2020). The application of drones in healthcare and health-related services in North America: A scoping review. *Drones*, 4(3), 30. <https://doi.org/10.3390/drones4030030>
- Hodgson, J. C., Mott, R., Baylis, S. M., Pham, T. T., Wotherspoon, S., Kilpatrick, A. D., Raja Segaran, R., Reid, I., Terauds, A., & Koh, L. P. (2018). Drones count wildlife more accurately and precisely than humans. *Methods in Ecology and Evolution*, 9(5), 1160–1167. <https://doi.org/10.1111/2041-210X.12974>
- Johansen, T. A., Cristofaro, A., Sørensen, K., Hansen, J. M., & Fossen, T. I. (2015). On estimation of wind velocity, angle-of-attack and sideslip angle of small UAVs using standard sensors. *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, Denver, Colorado, 510–519. <https://doi.org/10.1109/ICUAS.2015.7152330>
- Kalman, R. E., & Bucy, R. S. (1961). New results in linear filtering and prediction theory. *Journal of Fluids Engineering*. <https://doi.org/10.1115/1.3658902>
- Khaghani, M. (2018). *Vehicle dynamic model based navigation for small UAVs* (Doctoral dissertation, EPFL). <https://infoscience.epfl.ch/record/253624>
- Khaghani, M., & Skaloud, J. (2016). Autonomous vehicle dynamic model-based navigation for small UAVs. *NAVIGATION*, 63(3), 345–358. <https://doi.org/10.1002/navi.140>
- Khaghani, M., & Skaloud, J. (2018). Assessment of VDM-based autonomous navigation of a UAV under operational conditions. *Robotics and Autonomous Systems*, 106, 152–164. <https://doi.org/10.1016/j.robot.2018.05.007>
- Kim, S. J., Jeong, Y., Park, S., Ryu, K., & Oh, G. (2018). A survey of drone use for entertainment and AVR (augmented and virtual reality). *Augmented Reality and Virtual Reality*, 339–352. Springer. [https://doi.org/10.1007/978-3-319-64027-3\\_23](https://doi.org/10.1007/978-3-319-64027-3_23)
- Laupré, G., Khaghani, M., & Skaloud, J. (2019). Sensitivity to time delays in VDM-based navigation. *Drones*, 3(1), 11. <https://doi.org/10.3390/drones3010011>
- Laupré, G., Longobardi, P., Skaloud, J., & Charlaix, J.-C. (2021). Model based navigation of delta-wing UAV-in-flight calibration and autonomous performance. *European Journal of Navigation*, 21(1), 22–30.
- Laupré, G., & Skaloud, J. (2020). On the self-calibration of aerodynamic coefficients in vehicle dynamic model-based navigation. *Drones*, 4(3), 32. <https://doi.org/10.3390/drones4030032>
- Laupré, G., & Skaloud, J. (2021). Calibration of fixed-wing UAV aerodynamic coefficients with photogrammetry for VDM- based navigation. *Proc. of the 2021 International Technical Meeting of the Institute of Navigation*, 775–786). <https://doi.org/10.33012/2021.17867>
- Leuenberger, D., Haefele, A., Omanovic, N., Fengler, M., Martucci, G., Calpini, B., Fuhrer, O., & Rossa, A. (2020). Improving high- impact numerical weather prediction with lidar and drone observations. *Bulletin of the American Meteorological Society*, 101(7), E1036–E1051. <https://doi.org/10.1175/BAMS-D-19-0119.1>
- Meier, L., Tanskanen, P., Fraundorfer, F., & Pollefeys, M. (2011). Pixhawk: A system for autonomous flight using onboard computer vision. *2011 IEEE International Conference on Robotics and Automation*, 2992–2997. <https://doi.org/10.1109/ICRA.2011.5980229>
- Moore, B. (1981). Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1), 17–32. <https://doi.org/10.1109/TAC.1981.1102568>
- Multiplex hitec power peak. (2022). <https://www.multiplex-rc.de/>. Accessed: 2022-08-10
- Mweneogha, H. A., Moore, T., Pinchin, J., & Jabbal, M. (2019a). Enhanced fixed wing UAV navigation in extended GNSS outages using a vehicle dynamics model and raw GNSS observables. *Proc. of the 32nd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2019)*, Miami, Florida, 2552–2565). <https://doi.org/10.33012/2019.17056>
- Mweneogha, H. A., Moore, T., Pinchin, J., & Jabbal, M. (2019b). Model-based autonomous navigation with moment of inertia estimation for unmanned aerial vehicles. *Sensors*, 19(11), 2467. <https://doi.org/10.3390/s19112467>
- Mweneogha, H. A., Moore, T., Pinchin, J., & Jabbal, M. (2020). A model-based tightly coupled architecture for low-cost unmanned aerial vehicles for real-time applications. *IEEE Access*. <https://doi.org/10.1109/access.2020.3038530>
- Mweneogha, H. A., Moore, T., Pinchin, J., & Jabbal, M. (2021). Error characteristics of a model-based integration approach for fixed-wing unmanned aerial vehicles. *Journal of Navigation*, 74(6), 1353–1366. <https://doi.org/10.1017/S0373463321000424>
- Niu, X.-j., Zhang, Q., Gong, L.-l., Liu, C., Zhang, H., Shi, C., Wang, J., & Coleman, M. (2015). Development and evaluation of GNSS/INS data processing software for position and orientation systems. *Survey Review*, 47(341), 87–98. <https://doi.org/10.1179/1752270614Y.0000000099>
- Nonami, K., Kendoul, F., Suzuki, S., Wang, W., & Nakazawa, D. (2010). *Autonomous flying robots: Unmanned aerial vehicles and micro aerial vehicles*. Springer Science & Business Media. <https://doi.org/10.1007/978-4-431-53856-1>
- Paul, K. J., Cucci, D. A., & Skaloud, J. (2023). Model-based constraints for trajectory determination of quad-copters: Design, calibration & merits for direct orientation. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, 100030. <https://doi.org/10.1016/j.ojphoto.2023.100030>

- Puri, V., Nayyar, A., & Raja, L. (2017). Agriculture drones: A modern breakthrough in precision agriculture. *Journal of Statistics and Management Systems*, 20(4), 507–518. <https://doi.org/10.1080/09720510.2017.1395171>
- Px4 autopilot. (2009). <https://px4.io/>
- Rehak, M., & Skaloud, J. (2015). Fixed-wing micro aerial vehicle for accurate corridor mapping. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-1/W1, 23–31. <https://dx.doi.org/10.5194/isprsannals-II-1-W1-23-2015>
- Schwithal, J., Rohlf, D., Looye, G., & Liersch, C. M. (2016). An innovative route from wind tunnel experiments to flight dynamics analysis for a highly swept flying wing. *CEAS Aeronautical Journal*, 7(4), 645–662. <https://doi.org/10.1007/s13272-016-0214-3>
- Sendobry, D.-I. A. (2014). Control system theoretic approach to model based navigation. *Ingenieurwiss. Verlag*. <http://dx.doi.org/10.13140/RG.2.2.22171.87846>
- Stim 318. (2022). <https://www.sensor.com/products/inertial-measurement-units/stim318/>. Accessed: 2022-07-14
- Tomé, P., Cunha, T., Cunha, S., & Bastos, L. (2000). Evaluation of a DGPS/IMU integrated navigation system. *Proc. of the 13th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS 2000)*, Salt Lake City, Utah, 2233–2242. <https://www.ion.org/publications/abstract.cfm?articleID=1640>
- Wisnoe, W., Nasir, R. E. M., Kuntjoro, W., & Mamat, A. M. I. (2009). Wind tunnel experiments and CFD analysis of blended wing body (BWB) unmanned aerial vehicle (UAV) at mach 0.1 and mach 0.3. *International Conference on Aerospace Sciences and Aviation Technology*, Vol. 13, 1–15. <https://doi.org/10.21608/asat.2009.23441>

**How to cite this article:** Sharma, A., François Laupré, G., & Skaloud, J. (2023). Identifying aerodynamics of small fixed-wing drones using inertial measurements for model-based navigation. *NAVIGATION*, 70(4). <https://doi.org/10.33012/navi.611>