Article

Self-Localization of Tethered Drones without a Cable Force Sensor in GPS-Denied Environments

Amer Al-Radaideh * and Liang Sun

Department of Mechanical and Aerospace Engineering, New Mexico State University, Las Cruces, NM 88003, USA; lsun@nmsu.edu
* Correspondence: radaideh@nmsu.edu

Abstract: This paper considers the self-localization of a tethered drone without using a cable-tension force sensor in GPS-denied environments. The original problem is converted to a state-estimation problem, where the cable-tension force and the three-dimensional position of the drone with respect to a ground platform are estimated using an extended Kalman filter (EKF). The proposed approach uses the data reported by the onboard electric motors (i.e., the pulse width modulation (PWM) signals), accelerometers, gyroscopes, and altimeter, embedded in the commercial-of-the-shelf (COTS) inertial measurement units (IMU). A system-identification experiment was conducted to determine the model that computes the drone thrust force using the PWM signals. The proposed approach was compared with an existing work that assumes known cable-tension force. Simulation results show that the proposed approach produces estimates with less than 0.3-m errors when the actual cable-tension force is greater than 1 N.

Keywords: tethered drone; kalman filtering; self-localization; GPS-denied navigation

Citation: Al-Radaideh, A.; Sun, L. Self-Localization of Tethered Drones without a Cable Force Sensor in GPS-Denied Environments. *Drones* **2021**, *5*, 135. https://doi.org/10.3390/drones5040135

Academic Editor: Francesco Nex

Received: 21 September 2021
Accepted: 27 October 2021
Published: 17 November 2021

1. Introduction

Tethered drones have been witnessed in various applications, such as surveillance [1–4], high-rise building cleaning [5,6], infrastructure monitoring [7], wind turbine cleaning and de-icing [8,9], and firefighting [10]. Although the tether would limit the reachable space of the drone compared to a free-flying drone, it offers unique persistent and secured data transmission link and electricity power to the drone [1]. The potential combination of the tether with a hose also provides capabilities of delivering fluid to a targeting area, such as, spraying pesticides on a crop field [11]. The effective use of tethered drones in these applications requires accurate self-localization information. For example, for surveillance/monitoring applications, the meter-level self-localization accuracy would be acceptable, while for applications such as agricultural chemical spraying and wind-turbine and high-rise-building cleaning, the decimeter/centimeter-level accuracy for self-localization would be preferred.

Small drones notably rely on accurate self-location information for guidance, navigation, and control. Drone self-localization typically counts on IMUs [12–14], the Global Positioning System (GPS) [15] (differential GPS [16]), infrared (IR) sensors [17], laser rangefinders [18,19], and optical and vision systems [20–22]. While these sensing systems have successfully supported outdoor applications, extensive investment has been made to enhance the capability of self-localization for drone by improving the GPS infrastructure, utilizing cellular network infrastructure [23], or integrating both technologies for a wider range of applications. However, the self-localization of small drones in GPS-degraded/-denied environments (e.g., indoors and street canyons) is still challenging due to their limited size, payload, power, and flight endurance that have prevented them from carrying high-end sensors for self-localization. This poses critical concerns to the safe operation of drones in GPS-degraded/-denied environments.

In previous studies for the self-localization and control of tethered drones, Lupashin and D'Andrea [24] presented an approach to estimating the two-dimensional (2D) location of the drone with respect to a ground station. Tognon and Franchi [25] presented an observer-based control technique to regulate a tethered drone attached to a moving ground platform. Lima and Pereira [26] presented an EKF-based self-localization approach by assuming a catenary-shape cable for a static drone in hovering and assuming that the cable-tension force is known. Companies have also commercialized tethered drones on the market [27,28]. In our previous work [29,30], we presented both a low pass filter (LPF) and an extended Kalman filter (EKF) to estimate the three-dimensional (3D) location of the drone with respect to a ground platform (see Figure 1) while assuming known cable-tension force. In this paper, we assume the cable-tension force is unknown and we extend our previous work by enabling simultaneously estimation of both the 3D drone location and the cable-tension force, using only the measurements of onboard IMUs and altimeter.

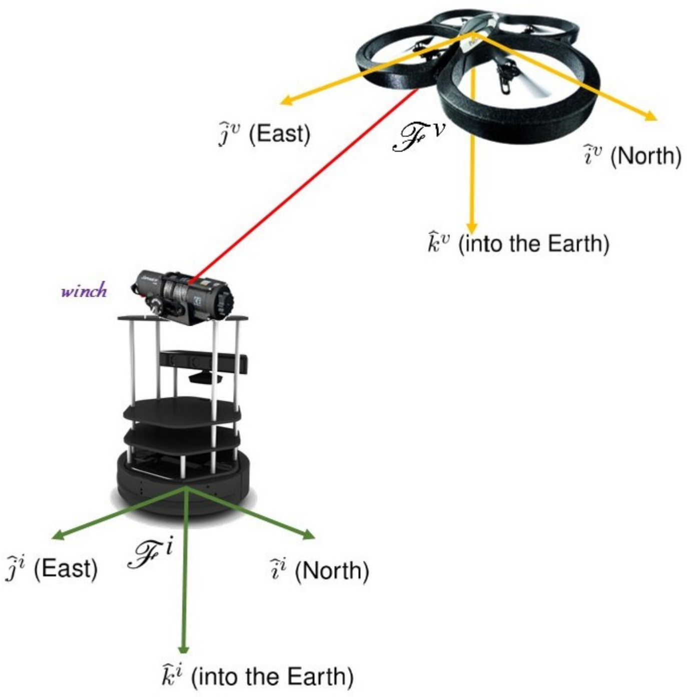


Figure 1. A drone is tethered to a ground robot [29].

To the best of our knowledge, existing literature [24–30] for the self-localization and control of tethered drones has assumed known cable-tension force and accurate drone thrust forces, which, however, are nontrivial to measure directly. The cable-tension force is usually assumed to be measured by a force sensor that is connected in series with the tether. Connecting a COTS cable-tension force sensor underneath the drone will significantly increase the payload of the drone. Connecting the force sensor on the ground platform would be extremely challenging when the tether length varies with the drone movement. The drone thrust force is usually computed using the pulse width modulation (PWM) signals, but such a computational formula is not usually provided by a drone manufacturer, and it is usually unique for each particular drone. Existing work for computing the motor thrust using a PWM signal has focused on identifying the coefficients of a high-order polynomial of the PWM signal using a load cell to measure the thrust force [31–34]. However, setting up such experiments by attaching load cells to the drone motors requires considerable efforts of disassembling drone components. To the best of our knowledge, this paper presents one of the first works that apply the system-identification technique to model the relationship between the motor thrust and PWM signals without disassembling the drone, but only using real flight-test data.

The contribution of this paper includes the development of an EKF that enables the estimation of both the 3D position of a moving drone with respect to a ground platform and the cable-tension force, and the development of a system-identification method to compute the motor thrust force using the PWM signal. The measurements used for the proposed EKF are assumed to be measured by the onboard inertial sensors (e.g., accelerometers and gyroscopes), along with the altimeter (e.g., an ultrasound sensor). We evaluate the proposed EKF in simulations in comparison to the 3-state EKF in [29]. The result shows that when the actual cable-tension force is greater than 1 N, the proposed 4-state EKF produces estimates with less than 0.3-N estimation errors, which are equivalent to the performance of the technique, assuming a known cable-tension force [29].

The remainder of this paper is structured as follows. System dynamics and accelerometer principles are introduced in Section 2. The problem statement and state-space model are introduced in Section 3. The EKF development and system identification for motor coefficients are presented in Sections 4 and 5, respectively. Section 6 shows and discusses the simulation results, and Section 7 concludes the paper. Section 8 presents our future work.

2. System Dynamics and Accelerometer Principles

2.1. Coordinate Frames

We first introduce several key coordinate frames associated with the system dynamics of a drone, i.e., the inertial frame, the vehicle frame, and the body frame [35], as shown in Figure 1.

2.1.1. The Inertial Frame $\mathcal{F}^i$

The inertial coordinate frame is an earth-fixed coordinate system with its origin at a pre-defined location. In this paper, this coordinate system is referred to in the North-East-Down (NED) reference frame. It is common for North to be referred to as the inertial x direction, East to the y direction, and Down to the z direction.

2.1.2. The Vehicle Frame $\mathcal{F}^v$

The origin of the vehicle frame is at the center of mass of a drone. However, the axes of $\mathcal{F}^v$ are aligned with the axes of the inertial frame $\mathcal{F}^i$. In other words, the unit vector $\mathbf{i}^v$ points toward North, $\mathbf{j}^v$ toward East, and $\mathbf{k}^v$ toward the center of the earth.

2.1.3. The Body Frame $\mathcal{F}^b$

The body frame is obtained by rotating the vehicle frame in a right-handed rotation about $\mathbf{i}^v$ by the roll angle, ϕ, about the $\mathbf{j}^v$ axis by the pitch angle, θ, and about the $\mathbf{k}^v$ axis by the yaw angle, ψ. The transformation of the drone 3D position from $\mathbf{p}^b$ in $\mathcal{F}^v$ to $\mathbf{p}^v$ in $\mathcal{F}^b$ is given by

$$\mathbf{p}^b = R_v^b(\phi,\theta,\psi)\mathbf{p}^v, \tag{1}$$

where the transformation matrix, $R_v^b(\phi,\theta,\psi)$, is given by

$$R_v^b(\phi,\theta,\psi) = \begin{pmatrix} \mathrm{c}_\theta \mathrm{c}_\psi & \mathrm{c}_\theta \mathrm{s}_\psi & -\mathrm{s}_\theta \\ \mathrm{s}_\phi \mathrm{s}_\theta \mathrm{c}_\psi - \mathrm{c}_\phi \mathrm{s}_\psi & \mathrm{s}_\phi \mathrm{s}_\theta \mathrm{s}_\psi + \mathrm{c}_\phi \mathrm{c}_\psi & \mathrm{s}_\phi \mathrm{c}_\theta \\ \mathrm{c}_\phi \mathrm{s}_\theta \mathrm{c}_\psi + \mathrm{s}_\phi \mathrm{s}_\psi & \mathrm{c}_\phi \mathrm{s}_\theta \mathrm{s}_\psi - \mathrm{s}_\phi \mathrm{c}_\psi & \mathrm{c}_\phi \mathrm{c}_\theta \end{pmatrix}, \tag{2}$$

where $\mathrm{c}_* = \cos_*$ and $\mathrm{s}_* = \sin_*$.

2.2. Tethered Drone Dynamics

The equations of motion of a drone tethered to a stationary ground station are expressed by a six-degree-of-freedom model consisting of 12 states [35]

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{pmatrix} = R_b^v(\phi, \theta, \psi) \begin{pmatrix} u \\ v \\ w \end{pmatrix}, \tag{3}$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \frac{1}{m} \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix}, \tag{4}$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}, \tag{5}$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{pmatrix} + \begin{pmatrix} \frac{1}{J_x} \tau_l \\ \frac{1}{J_y} \tau_m \\ \frac{1}{J_z} \tau_n \end{pmatrix}, \tag{6}$$

where $(p_n, p_e, p_d)^T \in \mathbb{R}^3$ is defined as the drone position in the NED inertial frame, (u, v, w) is the drone linear velocity vector in the body frame, m is the drone mass, (p, q, r) is the rotational velocity vector in the body frame, (f_x, f_y, f_z) and (τ_l, τ_m, τ_n) are the total external forces and torques applied to the drone in the body frame, respectively, and J_x, J_y, and J_z are moments of inertia of the drone in x, y, and z directions, respectively.

2.3. Accelerometer Principle

The output of COTS accelerometers for drones contains several specific terms that are derived from the drone acceleration and are important for drone controller design and analysis. In this subsection, the normalized kinematic accelerations and specific forces [36] are introduced, which are used in the proposed self-localization methodology.

Kinematic Accelerations and Specific Forces

Let $\eta = (u, v, w)^T$ be the linear velocity vector, $\Omega = (p, q, r)^T$ be the rotational velocity vector of the drone in the body frame, and $\mathbf{f}^b = \left(f_x, f_y, f_z\right)^T$ be the total external force vector in the body frame. Define the kinematic acceleration vector $\mathbf{a}_k^b \triangleq \left(\mathbf{a}_{k,x}^b, \mathbf{a}_{k,y}^b, \mathbf{a}_{k,z}^b\right)^T$ in the body frame as

$$\mathbf{a}_k^b = \frac{\mathbf{f}^b}{mg} = \frac{\dot{\eta}}{g} = \frac{1}{g}\left(\frac{\partial \eta}{\partial t} + \Omega \times \eta\right), \tag{7}$$

of which the components are

$$\mathbf{a}_{k,x}^b = \frac{1}{g}(\dot{u} + qw - rv) = \frac{f_x}{mg}, \tag{8}$$

$$\mathbf{a}_{k,y}^b = \frac{1}{g}(\dot{v} + ru - pw) = \frac{f_y}{mg}, \tag{9}$$

$$\mathbf{a}_{k,z}^b = \frac{1}{g}(\dot{w} + pv - qu) = \frac{f_z}{mg}, \tag{10}$$

where g is the gravitational acceleration constant on Earth. Note that $\mathbf{a}_k^b$ is in units of g. The accelerometer is assumed to be mounted at the center of gravity of a drone.

The output of accelerometers used by drone autopilots is generated in the form of the specific force, $\mathbf{a}_{SF}^b$, also called g-force or mass-specific force (measured in meters/second^2), which is actually an acceleration ratio given by

$$\mathbf{a}_{SF}^b = \frac{\mathbf{f}^b - \mathbf{f}_g^b}{mg} = \mathbf{a}_k^b - \frac{\mathbf{f}_g^b}{mg}, \tag{11}$$

whose components are given by

$$\begin{aligned}
\mathbf{a}^b_{SF,x} &= \mathbf{a}^b_{k,x} + \sin\theta, && (12)\\
\mathbf{a}^b_{SF,y} &= \mathbf{a}^b_{k,y} - \cos\theta\sin\phi, && (13)\\
\mathbf{a}^b_{SF,z} &= \mathbf{a}^b_{k,z} - \cos\theta\cos\phi. && (14)
\end{aligned}$$

2.4. External Forces of Tethered Drone

The total external force vector for a tethered drone in the body frame is given by

$$\mathbf{f}^b = \mathbf{f}^b_{thrust} + \mathbf{f}^b_g + \mathbf{f}^b_{cable}, \tag{15}$$

where $\mathbf{f}^b_{thrust}$ is the thrust force, $\mathbf{f}^b_g$ is the gravity force, and $\mathbf{f}^b_{cable}$ is the cable-tension force, all in the body frame. The gravity force vector of the drone in the vehicle frame, $\mathbf{f}^v_g$, is given by

$$\mathbf{f}^v_g = \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix}. \tag{16}$$

Then, we have

$$\mathbf{f}^b_g = R^b_v \mathbf{f}^v_g = \begin{pmatrix} -mg\sin\theta \\ mg\cos\theta\sin\phi \\ mg\cos\theta\sin\theta \end{pmatrix}. \tag{17}$$

The thrust force vector in the body frame is given by

$$\mathbf{f}^b_{thrust} = \begin{pmatrix} f_{thrust,x} \\ f_{thrust,y} \\ f_{thrust,z} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -(f_F + f_R + f_B + f_L) \end{pmatrix}, \tag{18}$$

where subscripts F, R, B, and L denote the thrust forces provided by the front, right, back, and left motors, respectively. The individual thrust forces have been calculated using the PWM signals commanded to the motors, such as,

$$f_* = k_{motor} \cdot \text{pwm}_*, \tag{19}$$

where $* \in \{F, R, B, L\}$ and k_{motor} is the electric motor coefficient and pwm_* is the PWM motor control signal. However, the mapping between the drone motor thrust force and the PWM signals is much more complicated than the linear relationship shown in (19). We will discuss this more in Section 5.

Since the output of the accelerometer is the total acceleration (see Equation (11)) minus the gravity terms [35]

$$\mathbf{a}^b_{SF} = \frac{\mathbf{f}^b - \mathbf{f}^b_g}{mg} = \mathbf{a}^b_k - \frac{\mathbf{f}^b_g}{mg} = \frac{\mathbf{f}^b_{thrust} + \mathbf{f}^b_{cable}}{mg}, \tag{20}$$

assuming a taut cable, $\mathbf{f}^b_{cable}$ is given by

$$\mathbf{f}^b_{cable} = R^b_v f^v_{cable} \frac{L}{\ell}, \tag{21}$$

where $L = (p_n, p_e, p_d)^T$, $\ell = \sqrt{p_n^2 + p_e^2 + p_d^2}$, and f^v_{cable} is the magnitude of the cable-tension force. We can then obtain

$$\mathbf{f}^b_{cable} = \frac{R^b_v f^v_{cable}}{\sqrt{p_n^2 + p_e^2 + p_d^2}} \left[\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} p_n \\ p_e \\ p_d \end{pmatrix} \right]. \tag{22}$$

Then, Equation (20) can be written as

$$\mathbf{a}^b_{SF} = \begin{bmatrix} \mathbf{a}^b_{SF,x} \\ \mathbf{a}^b_{SF,y} \\ \mathbf{a}^b_{SF,z} \end{bmatrix} = \frac{1}{mg}\left[\begin{pmatrix} 0 \\ 0 \\ -(f_F + f_R + f_B + f_L) \end{pmatrix} - \frac{R^b_v f^v_{cable}}{\sqrt{p_n^2 + p_e^2 + p_d^2}}\begin{pmatrix} p_n \\ p_e \\ p_d \end{pmatrix}\right]. \quad (23)$$

3. Self-Localization of Tethered Drone

3.1. Problem Statement

Consider a scenario where a drone tethered to a ground robot (see Figure 1). In this paper, the ground robot is assumed to be stationary and the tether is assumed to be controlled by a retractable winch that provides a constant cable-tension force. The problem is to estimate the 3D position of the tethered drone with respect to the ground station (i.e., the origin of the vehicle coordinate frame), using the measurements of the accelerometer, gyroscopes, altimeter, and PWM signals onboard the drone.

3.2. State-Space Model for Self-Localization

In our previous work [29], we presented a 3-state state-space model for self-localization by assuming that the cable-tension force is known. In this paper, we develop a 4-state state-space model to estimate the drone 3D location, as well as the cable-tension force.

Define the state vector as

$$\mathbf{x_{4s}} = (p_n, p_e, p_d, f_c)^T \in \mathbb{R}^4 \quad (24)$$

and the system dynamics are given by

$$\dot{\mathbf{x}}_{\mathbf{4s}} = f(\mathbf{x_{4s}}, \mathbf{u}), \quad (25)$$

where **u** is the system input vector. Since we do not know the actual motion plan and the cable-tension force evolution, we will use the following system dynamics to derive the EKF

$$\dot{\mathbf{x}}_{\mathbf{4s}} = f(\mathbf{x_{4s}}, \mathbf{u}) = \mathbf{0}_{4\times 1}. \quad (26)$$

Assuming that the measurements of the 3-axis accelerometers and the altimeter (i.e., the ultrasound sensor) are available and according to Equation (23), the output function is given by

$$\mathbf{y} = h(\mathbf{x_{4s}}) = \begin{pmatrix} \mathbf{a}^b_{SF,x} \\ \mathbf{a}^b_{SF,y} \\ \mathbf{a}^b_{SF,z} \\ -p_d \end{pmatrix} = \begin{bmatrix} \frac{1}{mg}\left[\begin{pmatrix} 0 \\ 0 \\ -(f_F + f_R + f_B + f_L) \end{pmatrix} - \frac{R^b_v f^v_{cable}}{\sqrt{p_n^2+p_e^2+p_d^2}}\begin{pmatrix} p_n \\ p_e \\ p_d \end{pmatrix}\right]_{3\times 1} \\ -p_d \end{bmatrix}. \quad (27)$$

4. Extended Kalman Filter

In this section, we present the application of the EKF (Algorithm 1) [35] technique to estimate the location of the drone and the cable-tension force. The system dynamics and output equations are described in Section 3. We assume that the available sensor measurements include the 3-axis orientations and accelerations that are distorted by white Gaussian noise. Selecting the system state vector at time k is $\mathbf{x}_k = (p_{n,k}, p_{e,k}, p_{d,k})^T \in \mathbb{R}^3$ and the system measurement vector at time step k as $\mathbf{y}_k = \left(\mathbf{a}^b_{SF,x,k}, \mathbf{a}^b_{SF,y,k}, \mathbf{a}^b_{SF,z,k}, -p_{d,k}\right)^T$, the state transition and observation models are given by

$$\hat{\mathbf{x}}_{k+1} = \mathbf{f}(\hat{\mathbf{x}}_k, \mathbf{u}_k, \mathbf{w}_k), \mathbf{w}_k \sim N(0, Q), \quad (28)$$

$$\hat{\mathbf{y}}_{k+1} = \mathbf{h}(\hat{\mathbf{x}}_{k+1}, \mathbf{v}_{k+1}), \mathbf{v}_{k+1} \sim N(0, R), \quad (29)$$

where $\hat{\mathbf{x}}_{k+1}$ and $\hat{\mathbf{y}}_{k+1}$ denote the approximated a posteriori state and observation, respectively, and $\hat{\mathbf{x}}_k$ the a priori estimate of the previous step. The process is characterized by random noise variables $\mathbf{w}_k$ and $\mathbf{v}_k$ that represent respectively the process noise and measurement noise, both of which follow the Gaussian distribution with covariance matrices Q and R, respectively. The diagram in Figure 2 summarizes the EKF process loop [37] with associated equations.

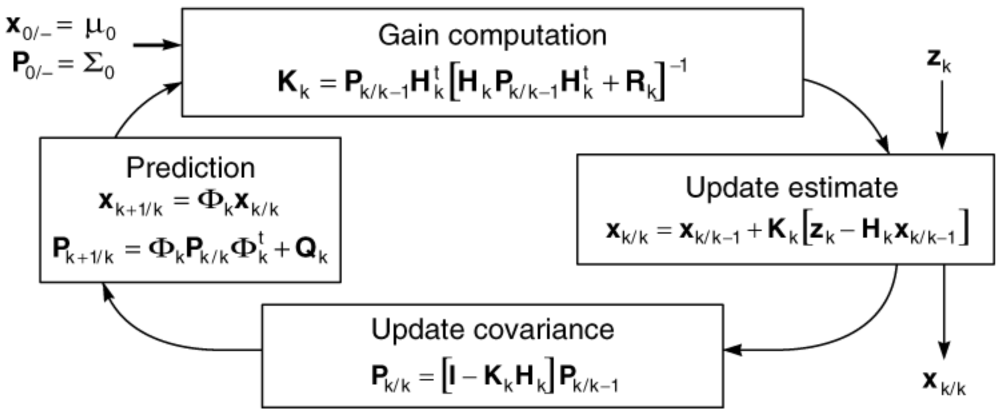

$\mathbf{x}_{0/-} = \mu_0$
$\mathbf{P}_{0/-} = \Sigma_0$
Gain computation
$\mathbf{K}_k = \mathbf{P}_{k/k-1}\mathbf{H}_k^t\left[\mathbf{H}_k\mathbf{P}_{k/k-1}\mathbf{H}_k^t + \mathbf{R}_k\right]^{-1}$
$\mathbf{z}_k$
Prediction
$\mathbf{x}_{k+1/k} = \Phi_k\mathbf{x}_{k/k}$
$\mathbf{P}_{k+1/k} = \Phi_k\mathbf{P}_{k/k}\Phi_k^t + \mathbf{Q}_k$
Update estimate
$\mathbf{x}_{k/k} = \mathbf{x}_{k/k-1} + \mathbf{K}_k\left[\mathbf{z}_k - \mathbf{H}_k\mathbf{x}_{k/k-1}\right]$
Update covariance
$\mathbf{P}_{k/k} = \left[\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right]\mathbf{P}_{k/k-1}$
$\mathbf{x}_{k/k}$

Figure 2. The recursive process of EKF [37].

Algorithm 1 Extended Kalman Filter [35].

```
1: Initialize: $\hat{x}$
2: At each sample time $T_{out}$,
3: for i = 1 to N do {Prediction}
4:     $\hat{x} = \hat{x} + (\frac{T_{out}}{N})\, f(\hat{x}, u)$
5:     $\Phi = \frac{\partial f}{\partial x}(\hat{x}, u)$
6:     $P = P + (\frac{T_{out}}{N})(\Phi P + P\Phi^T + Q)$
7:     Calculate $A$, $P$, and $C$
8: end for
9: if measurement has been received from sensor $i$ then {Correction:Measurement Update}
10:    $H_i = \frac{\partial h_i}{\partial x}(\hat{x}, u)$
11:    $K_i = PH_i^T(R + H_iPH_i^T)^{-1}$
12:    $P = (I - K_iH_i)P$
13:    $\hat{x} = \hat{x} + K_i(y_i[n] - h(\hat{x}, u[n]))$
14: end if
```

The EKF starts by calculating the Jacobian of the $\mathbf{f}(\mathbf{x}, \mathbf{u})$ and $\mathbf{h}(\mathbf{x})$ functions that were derived in Section 3.2. The prediction step before acquiring the measurements is given by

$$\hat{\mathbf{x}}_{k+1/k} = \Phi_k\hat{\mathbf{x}}_{k/k}, \quad (30)$$

$$P_{k+1/k} = \Phi_kP_k\Phi_k^T + Q_k, \quad (31)$$

while the update step after acquiring the measurements is given by

$$K_k = P_{k/k-1}H_k^T[H_kP_{k/k-1}H_k^T + R_k]^{-1}, \quad (32)$$

$$\hat{\mathrm{x}}_{\mathrm{k/k}} = \hat{\mathrm{x}}_{\mathrm{k/k-1}} + K_k(z_k - H_k\hat{\mathrm{x}}_{\mathrm{k/k-1}}), \quad (33)$$

$$P_{k/k} = (I - K_kH_k)P_{k/k-1}, \quad (34)$$

where K is the Kalman gain matrix, and P is the covariance matrix for the state estimate, containing information about the accuracy of the estimate [38]. Figure 3 shows the localization/EKF algorithm flowchart and diagram that is implemented and coded. The Jacobian of $\mathbf{h}(\mathbf{x})$ with respect to $\hat{x}$ is given by

$$\frac{\partial h}{\partial x} = \begin{bmatrix} \left[\left[\frac{-R_v^b f_{cable}^v}{mg.\left(\sqrt{p_n^2+p_e^2+p_d^2}\right)^3} \begin{bmatrix} p_e^2+p_d^2 & -p_n p_e & -p_n p_d \\ -p_n p_e & p_n^2+p_d^2 & -p_e p_d \\ -p_n p_d & -p_e p_d & p_n^2+p_e^2 \end{bmatrix} \right] \right]_{3\times3} & \frac{-R_v^b}{mg} \begin{bmatrix} \frac{p_n}{\sqrt{p_n^2+p_e^2+p_d^2}} \\ \frac{p_n}{\sqrt{p_n^2+p_e^2+p_d^2}} \\ \frac{p_n}{\sqrt{p_n^2+p_e^2+p_d^2}} \end{bmatrix}_{3\times1} \\ \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}_{1\times3} & 0 \end{bmatrix}. \quad (35)$$

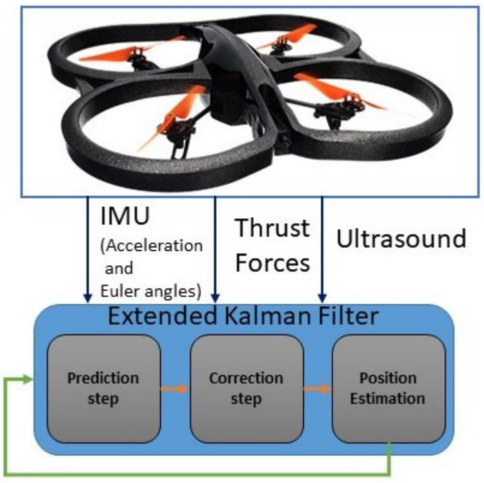


Figure 3. EKF flowchart for tethered drone self-localization [29].

5. System Identification for Motor Coefficients

In order to compute accurate motor thrust forces using the PWM signals, we present a system-identification strategy in this section to obtain function f_* in Equation (19) [39]. The system identification process has to go through a few steps to generate f_* that maps the input PWM signals to the total motor thrust [13,14,40–42]. The first step is to design flight experiments to collect the data with sufficient accuracy and duration. A good experimental design should ensure that the system is excited adequately by the input commands. The collected measurement data are usually processed by noise filtration and bias removal before being used for deriving high-fidelity models. A model structure is usually selected based on a prior knowledge of the input-output relation for model estimation. After that, the collected data are used to generate and update the selected parameters in the model, such that the model output is matched with the output in the data set. The dataset is usually divided into two subsets, which are used for estimation and validation, respectively. Validating the model and analyzing the uncertainty of the estimated model are the final steps before using the model for the application (e.g., control and state estimation). The estimation-validation process may take several iterations before finding the optimal model with the highest fitting percentage that is used to represent the model accuracy [43]. In this paper, the applied system-identification process [44] is summarized in Figure 4, and was implemented using the System Identification Toolbox in MATLAB®.

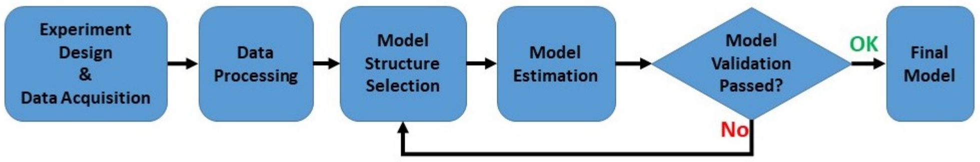


Figure 4. System identification process.

5.1. Experiment Design and Data Acquisition

The input commands to the drone system are the PWM signals of the four motors, and the sensor measurements include the three Euler attitude angles, the 3-axis accelerations, and the altitude. The output of the system-identification model is the total thrust force generated by all four motors, $\mathbf{f}^{\mathbf{b}}_{\mathbf{thrust}}$ (see Equation (18)), which is computed using the accelerometer measurement in the z-axis

$$f_{thrust,z} = mg \cdot R_b^v(\phi, \theta, \psi) \begin{pmatrix} 0 \\ 0 \\ a_z \end{pmatrix}. \tag{36}$$

The input-command sequences for the proposed tethered drone are designed, such that the individual inputs are sufficiently "exciting" system motion and guarantee meaningful identification results [45]. For this reason, indoor flights (see Figure 5) were conducted by first commanding the drone at a steady hovering flight. Then, the roll, pitch, and yaw angles were excited individually, while varying the altitude by changing the collective thrust input commands. Figure 6 shows a collected data set that consists of the computed thrust force (in Newton), the acceleration measurements (±1 g), and the Euler angles (±180 degrees) in response to the PWM commands (from 0 to 255).

5.2. Data Processing

The flight test data were collected using the "rosbags" in the robot operating system (ROS) and imported by MATLAB® for data processing. The data collected from the flight test were re-sampled at 100 Hz, and only the airborne data were selected. The plots between 10 and 140 s in Figure 6 indicate that the drone was in flight. The data are then filtered by a fifth-order Butter-worth low pass filter with a cut-off frequency of 10 Hz. The resulting data are then divided into two subsets for estimation and validation, respectively, as shown in Figure 6e.

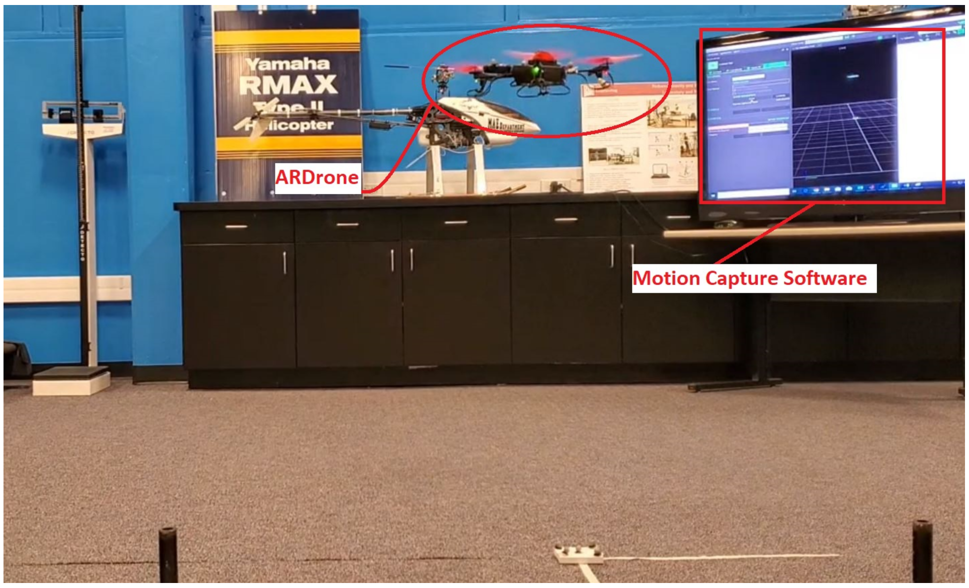
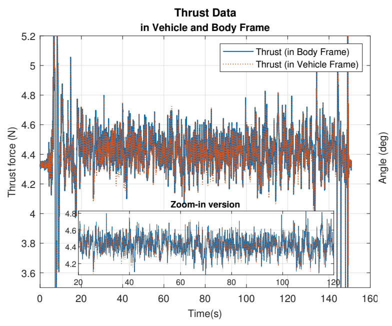
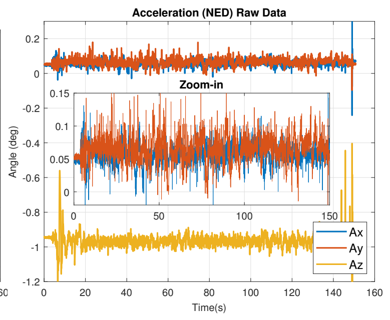


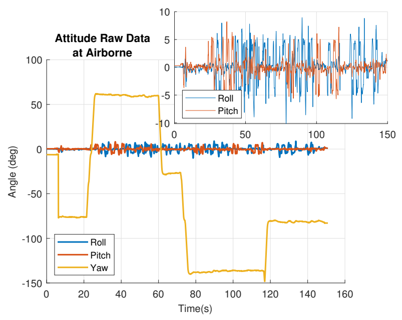
Figure 5. System Identification flight test for thrust modeling.



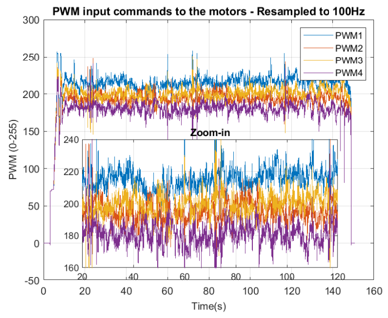
(**a**)



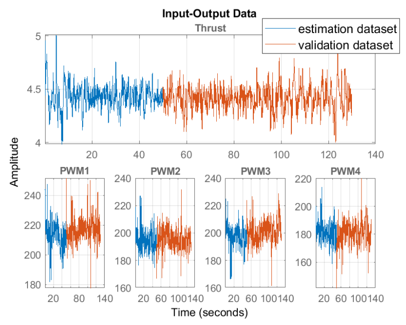
(**b**)



(**c**)



(**d**)



(**e**)

Figure 6. Data collected during a flight test for system identification. (**a**) Computed thrust. (**b**) Accelerations measurement. (**c**) Euler Angles (Attitudes). (**d**) PWM input-commands during the flight test. (**e**) The estimation-validation (filtered) data set from the flight test.

5.3. Model Structure Selection, Estimation, and Validation

In this work, we examined a variety of parametric model structures. Parametric models describe systems using differential equations and transfer functions as black-box models. The general linear-model structure can be represented by

$$y(t) = G(\xi, \eta)u(t) + H(\xi, \eta)e(t), \tag{37}$$

where $u(t)$ and $y(t)$ are the input and output of the system, respectively, $e(t)$ is the system disturbance, $G(\xi, \eta)$ and $H(\xi, \eta)$ are the transfer functions of the deterministic and the stochastic parts of the system, respectively, ξ is the backward shift operator, and η is the parameter vector [39]. A subset of the general linear model structure, can be represented as

$$A(\xi)y(t) = \frac{B(\xi)}{F(\xi)}u(t) + \frac{C(\xi)}{D(\xi)}e(t). \tag{38}$$

By setting one or more of the A, B, C, or D polynomials equal to 1, we can create simpler models, such as autoregressive (AR), autoregressive with exogenous variables (ARX), autoregressive moving average with exogenous input (ARMAX), Box–Jenkins (BJ), and output-error structures [40,41,46]. These methods have their own advantages and disadvantages and are selected based on the dynamics, and the noise characteristics of the system.

A model with more parameters does not necessarily generate more accurate results, as it may capture nonexistent dynamics and noise characteristics. This is where the physical insight into a system is helpful. The model structures that we have tested include the transfer function, process model, black- box ARX, state space, and Box–Jenkins. Black-box modeling is usually a trial-and-error process, where the parameters of various structures are estimated and compared. We started with the simple linear model structure and progressed to more complex ones [46]. ARX is the simplest and the most efficient method that solves linear regression equations in an analytic form with the global minimum of the loss function. The ARX model, therefore, is preferable in this work, as the model order is high. The disadvantage of the ARX model is its weak capability of eliminating disturbances from the system dynamics. The Box–Jenkins structure provides a complete formulation by separating disturbances from the system dynamics.

Transfer function models are commonly used to represent single-input-single-output (SISO) or multiple-input-multiple-output (MIMO) systems [47]. In the MATLAB® System Identification Toolbox, the process model structure describes the system dynamics, in terms of one or more of these elements, such as static gain, time constants, process zero, time delay, and integration [47].

The models generated were designed for prediction and the results demonstrated are for the five-step-ahead prediction [40,41,46,47]. Equations (A1)–(A8) in the Appendix A represent the two highest best fits models: the ARX and state-space models. Table 1 summarizes the quality of the identified models on the basis of fit percentage (Fit%), Akaike's final prediction error (FPE) [48], and the mean-squared error (MSE) [49]. As can be seen from Table 1, the fit percentages for the ARX, Box–Jenkins, and state space models are all above 94%, among which the state-space model has the best fit percentage, whereas the process models and the transfer functions are below 50%.

Table 1. Identification results for 5-step prediction.

Structure	Fit%	FPE	MSE
Transfer Function (mtf)	46%	0.002388	0.002343
Process Model (midproc0)	41.41%	0.002796	0.002778
Black-Box model-ARX Model (marx)	96.77%	8.478×10^{-6}	8.438×10^{-6}
State-Space Models Using (mn4sid)	99.56%	1.589×10^{-7}	1.562×10^{-7}
Box-Jenkins Model (bj)	94.64%	2.339×10^{-5}	2.326×10^{-5}

6. Simulation Results and Discussion

In order to evaluate the feasibility and performance of the proposed 4-state EKF for the tethered drone self-localization, numerical simulations were performed under MATLAB®/Simulink®.

The initial position of the drone is selected as $\mathbf{p}_0 = (0, 0, 0)^T$m and the drone is controlled to follow a circular orbit of 2.5-m radius with a constant velocity of 1 m/s and a varying altitude. The IMUs and ultrasound sensors are assumed to provide measurements with a frequency of 200 Hz [50]. The measurements of the 3-axis accelerometers and the ultrasound sensor are used to generate the outputs of the EKF in Equation (27). We assume that these measurements are corrupted by the Gaussian noise $\mathcal{N}(0, \sigma^2_{acc})$ (for each axis of the accelerometers) and $\mathcal{N}(0, \sigma^2_{ults})$, respectively, where $\sigma^2_{acc} = 0.01$ m/s^2 and $\sigma^2_{ults} = 0.1$ m [31]. Thus, the sensor noise covariance matrix, R, is selected as $R =$

$\text{diag}(\sigma^2_{acc}, \sigma^2_{acc}, \sigma^2_{acc}, \sigma^2_{ults}) = \text{diag}(0.01, 0.01, 0.01, 0.1)$. The 3-axis gyros measurements are used to compute the transformation matrix, R^b_v, in Equation (2). We assume that the 3-axis gyros measurements are corrupted by the Gaussian noise $\mathcal{N}(0, \sigma^2_{gyros})$ (for each axis of the gyros), where $\sigma^2_{gyros} = 0.01^\circ$. Figure 7 shows the noisy sensor measurements and the ones filtered by LPFs. The noisy measurements were directly used by the EKF and the values obtained by an LPF are used in the self-localization approach presented in [30]. The process noise covariance matrix of the EKF was tuned and selected as $Q = \text{diag}(5 \times 10^{-3}, 5 \times 10^{-3}, 5 \times 10^{-3})$. The initial state estimate was chosen to be $\hat{\mathbf{x}}_0 = (1.5, 2.5, 1.5)^T$ m, while the initial error covariance matrix was chosen to be $\mathbf{P}_0 = \mathbf{I}_3$. As for the LPF, it is based on a cutoff frequency set to 2 rad/s.

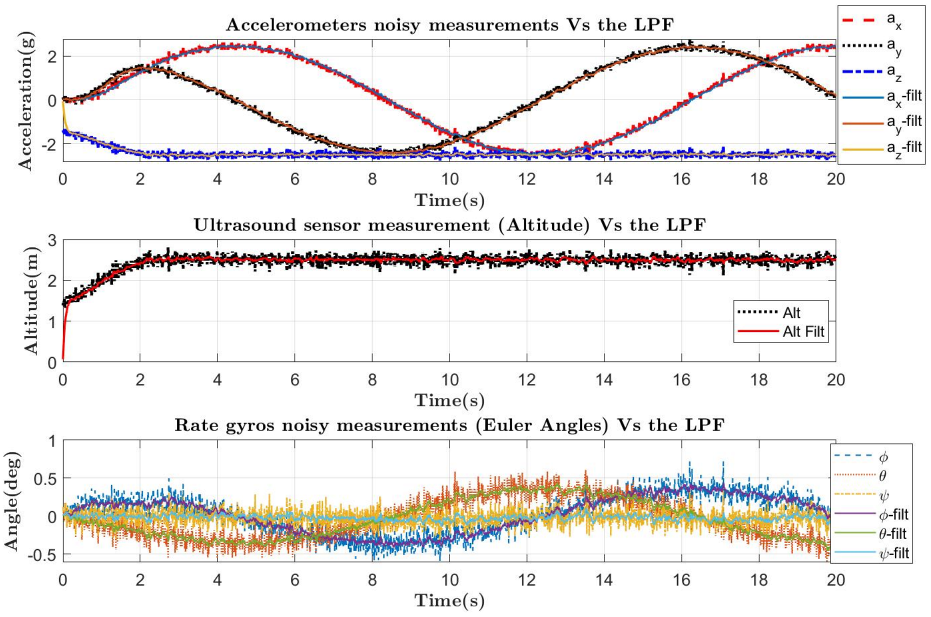
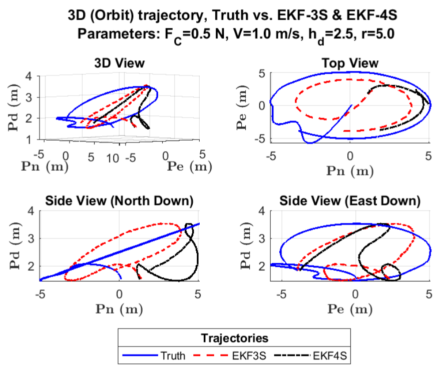


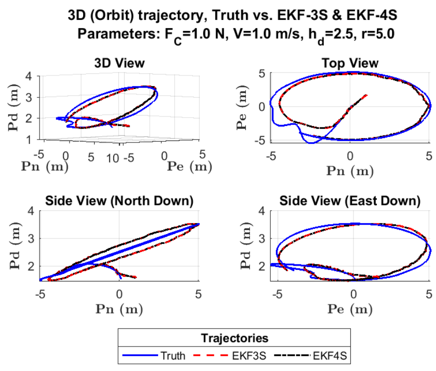
Figure 7. Sensors measurements and its low pass filter (LPF) output [29].

To compare the estimation results obtained from the proposed 4-state EKF and the 3-state EKF in [30], we assume that the 3-state model uses the actual cable-tension force from an onboard force sensor, but the 4-state EKF does not have access to the actual cable-tension force in the estimation process.

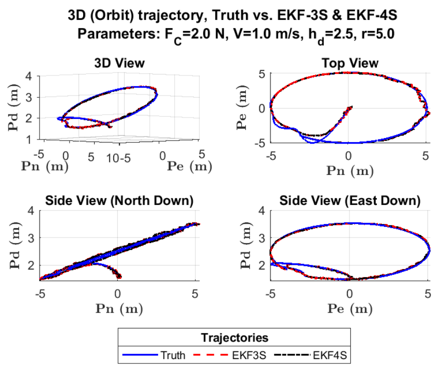
Figure 8 shows the ground-truth drone trajectory ("Truth" in the figure) overlaid with the estimated trajectories generated by the 3-state and 4-state EKFs ("EKF3S" and "EKF4S" in the figure, respectively) in the 3D, top-down, and side views with different magnitudes of the cable-tension force (0.5 N, 1 N, 2 N, 4 N, 6 N, and 10 N). Figure 9 shows the estimated North, East, and Down coordinates generated by the 3-state and 4-state EKFs versus the ground truth under different cable-force magnitudes, respectively. Figure 10 shows the estimation errors corresponding to Figure 9. It can be seen that the magnitude of the cable-tension force affects the accuracy of the position estimates obtained from both EKFs. When the cable-tension force is less than 2 N (see Figures 8a,b and 10a,b), both EKFs are unable to generate accurate estimates. Both EKFs generated very close estimates in the first 15 s, but diverged from each other after that. It seems the 3-state-EKF was able to follow the trend of the ground truth waves with smaller magnitude and slower pace, while the 4-state-EKF estimates become relatively flat after 15 s. When the cable-tension force is greater than 1 N, both EKF estimates start to follow the ground truth with increasing accuracy, but become increasingly noisy (see Figures 8–10b–f, respectively).



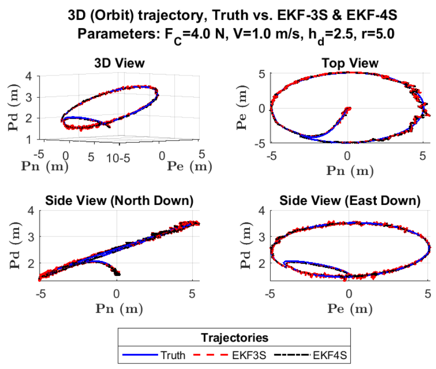
(**a**)



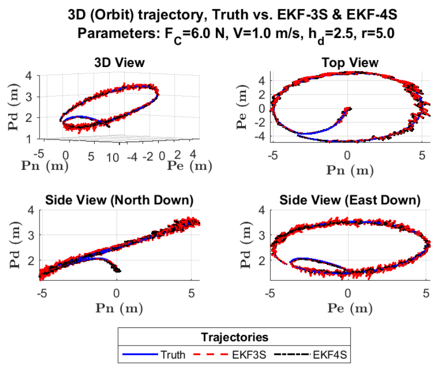
(**b**)



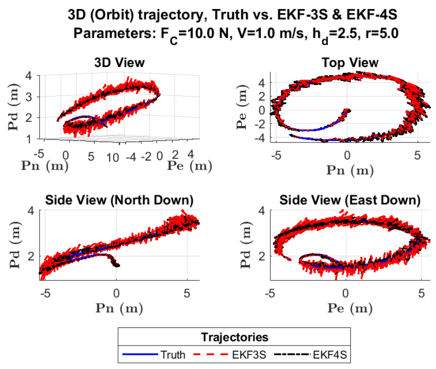
(**c**)



(**d**)



(**e**)



(**f**)

Figure 8. Different views of the ground-truth drone trajectory with a time-varying altitude and the state estimates generated by the 3-state and 4-state EKFs with different cable-tension force magnitudes (f_c): (**a**) $f_c = 0.5$ N, (**b**) $f_c = 1$ N, (**c**) $f_c = 2$ N, (**d**) $f_c = 4$ N, (**e**) $f_c = 6$ N, and (**f**) $f_c = 10$ N.

Figure 11 shows the ground-truth cable force (in blue) and its estimates (in red) using the 4-state EKF under different cable forces. Figure 11a shows that the cable-force estimation started to diverge from the beginning and generated impractical negative values and came back towards the ground truth after 20 s and diverged again after 25 s. This observation matches the position estimates in Figures 8–10. The cable-force estimates for other cases are consistently accurate within a ±0.3 N range.

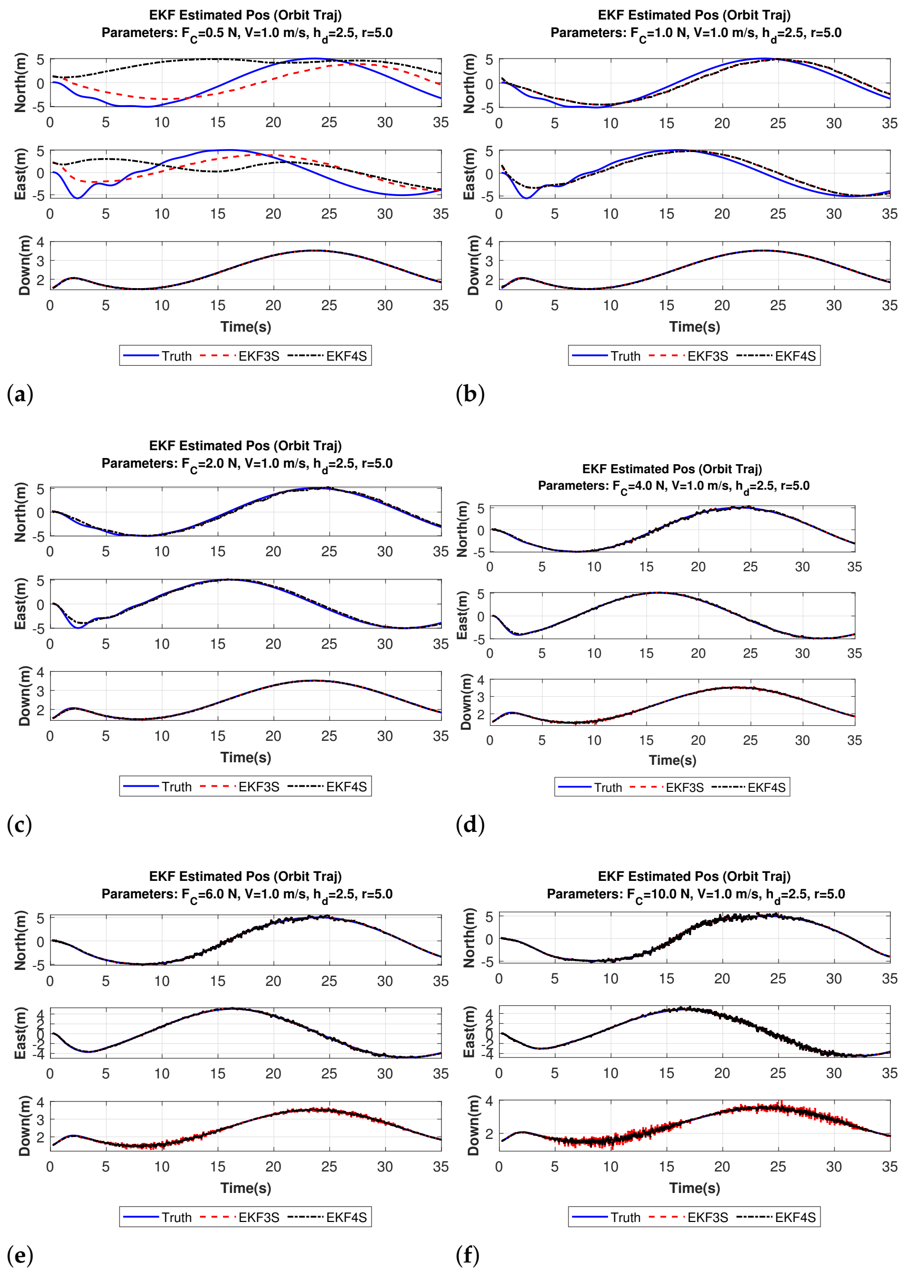


Figure 9. Position estimates using the 3-state and 4-state EKFs under different cable-tension force magnitudes (f_c): (**a**) $f_c = 0.5$ N, (**b**) $f_c = 1$ N, (**c**) $f_c = 2$ N, (**d**) $f_c = 4$ N, (**e**) $f_c = 6$ N, and (**f**) $f_c = 10$ N.

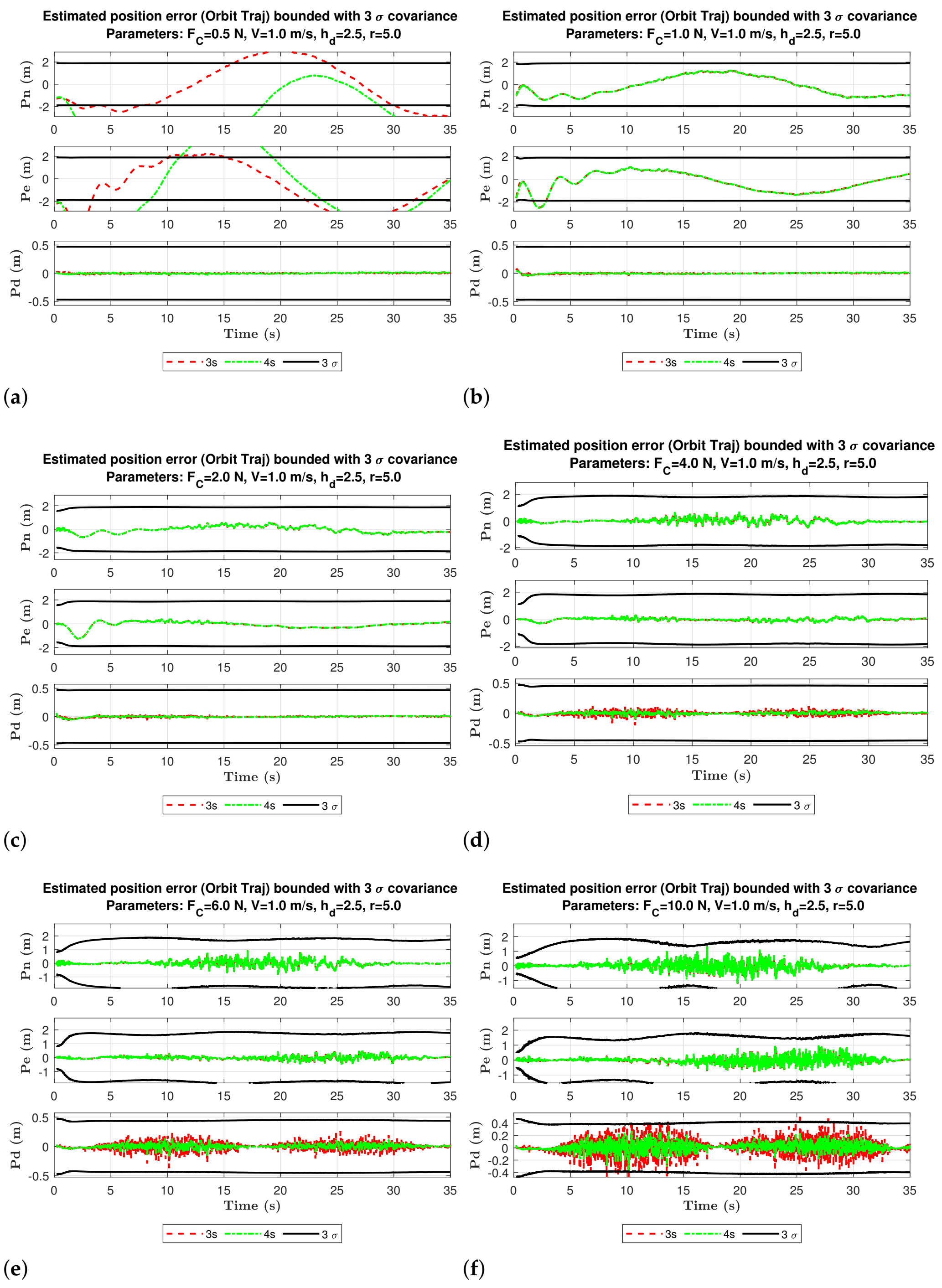


Figure 10. Position estimation errors generated by the 3-state EKF (3s), and 4-state EKF (4s) with different cable force magnitudes (f_c): (**a**) $f_c = 0.5$ N, (**b**) $f_c = 1$ N, (**c**) $f_c = 2$ N, (**d**) $f_c = 4$ N, (**e**) $f_c = 6$ N, and (**f**) $f_c = 10$ N. The 3σ boundaries refer to the 4-state EKF.

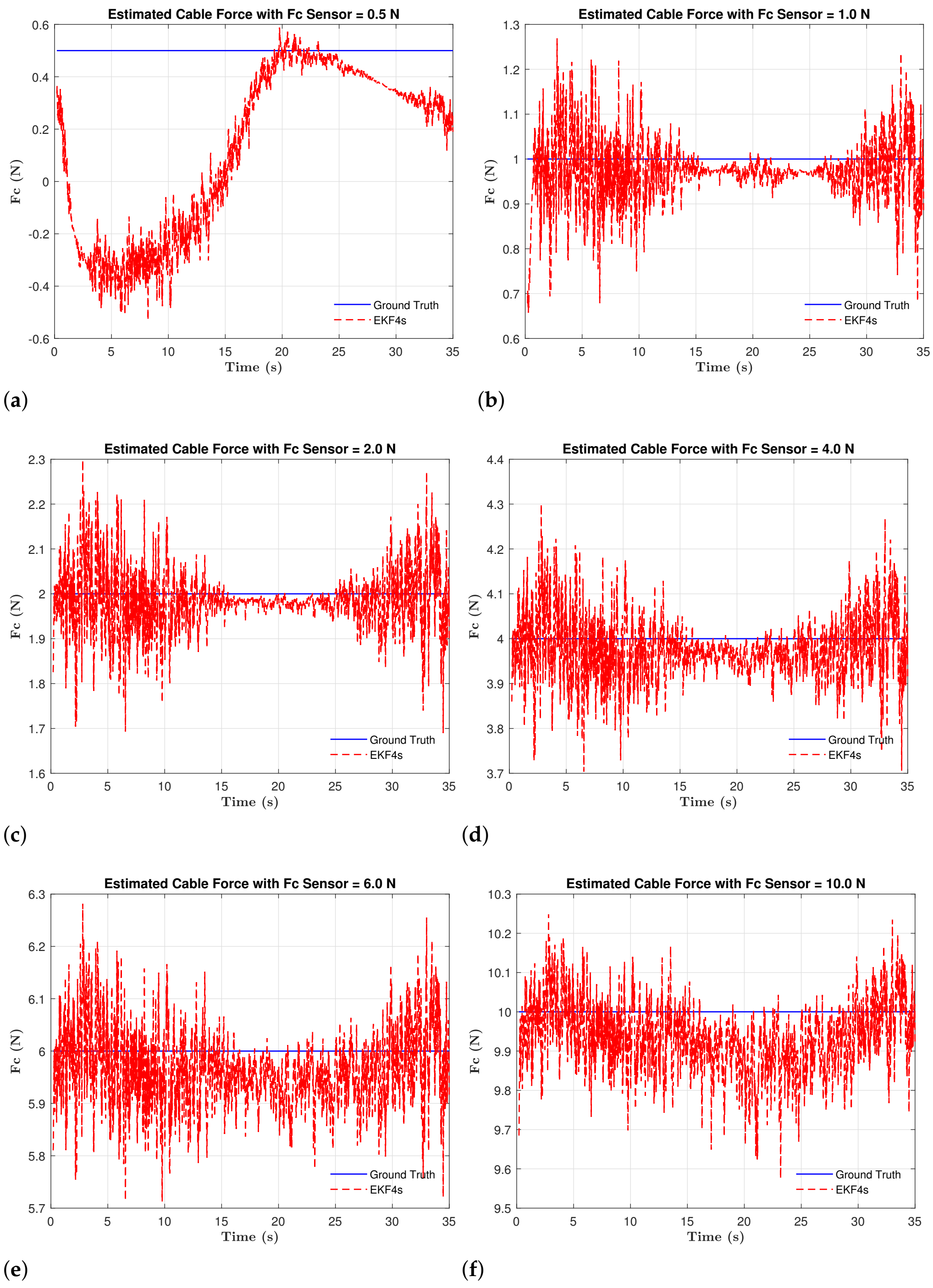


Figure 11. Cable-tension force estimates vs. the ground truth for different cable-tension force magnitudes (f_c): (**a**) $f_c = 0.5$ N, (**b**) $f_c = 1$ N, (**c**) $f_c = 2$ N, (**d**) $f_c = 4$ N, (**e**) $f_c = 6$ N, and (**f**) $f_c = 10$ N.

Table 2 summarizes the estimation results under different cable-force magnitudes using the root mean square error (RMSE) metric. We can see that for small cable-tension force values (i.e., <1 N), the 3-states model produces more accurate position estimates in the north and east directions.

Table 2. Root-Mean-Square-Error (RMSE) comparison of the 3-state (3S) and the 4-state (4S) EKFs.

	$f_c = 0.5$ N		$f_c = 2$ N		$f_c = 4$ N		$f_c = 10$ N	
Position	3S	4S	3S	4S	3S	4S	3S	4S
North (m)	2.022	5.075	0.275	0.276	0.159	0.156	0.236	0.243
East (m)	2.146	3.613	0.294	0.296	0.106	0.105	0.206	0.209
Down (m)	0.010	0.010	0.014	0.013	0.033	0.020	0.120	0.067
f_c (N)	-	0.495	-	0.066	-	0.071	-	0.109

To study the impact of various drone altitudes and velocities on the estimation accuracy, we conducted the simulation with drone altitudes ranging from 1 m to 10 m, and velocities ranging from 0.5 to 3 m/s. Figures 12–14 summarizes the RMSE results using different altitudes, velocities, and cable-tension forces. It can be seen from Figure 12 that the 3-states and 4-states EKFs have no significant difference, however, it can seen that at lower altitude of 1 m the error is around [0.7, 0.9] m in North and East position respectively. The error decreases as the altitude increases and it reaches its lowest value at around 5-m altitude. The error increases again as the altitude increases. It can be seen from Figure 13 that the lower the velocity, the lower the postion estimation error in all directions.

Figure 14 shows that the 4-state and 3-state EKFs provide 3D-position estimates with the same level of accuracy (less than 0.3 m, see Table 2) when the actual cable-tension force magnitude is greater than 1 N. The position estimation accuracy of both 4-state and 3-state EKFs degrades when the cable magnitude is less than 2 N, even though the 3-state EKF uses the true cable-tension force. This implies that to produce accurate position estimation using the proposed 4-state EKF, one needs to maintain the cable-tension force to be above 2 N, which can be realized by using a retractable cable system.

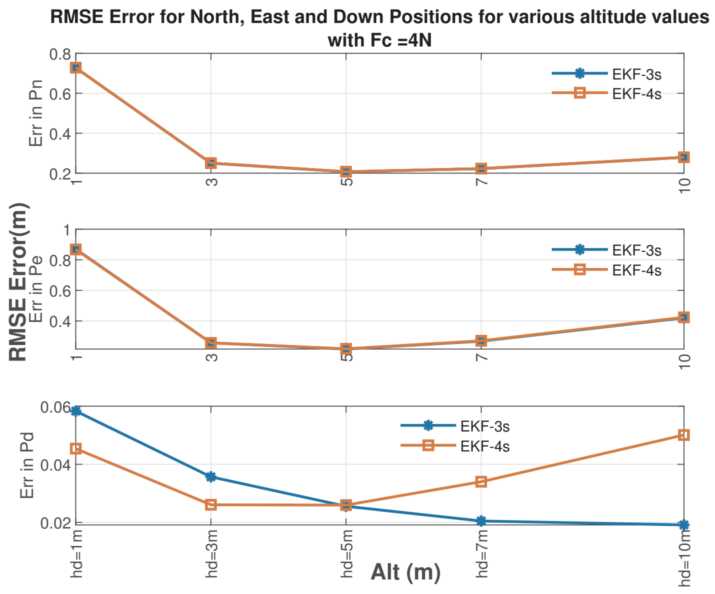


Figure 12. RMSE of 3D position estimates with various drone altitudes.

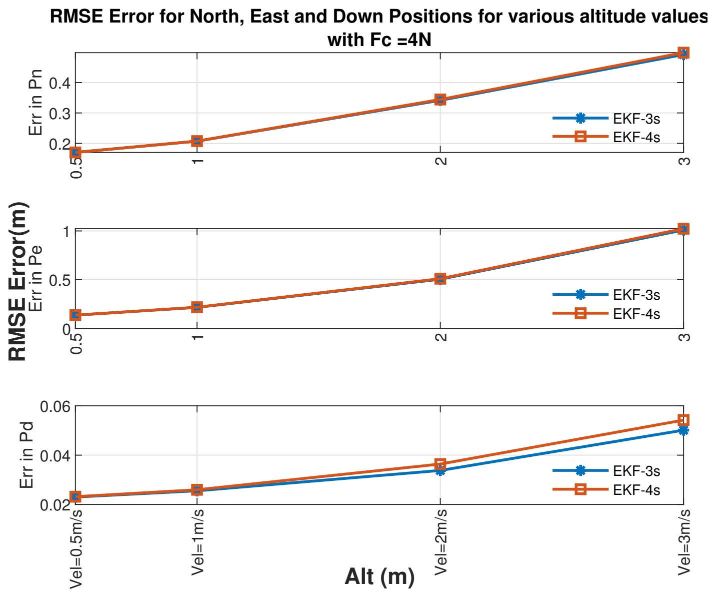


Figure 13. RMSE of 3D position estimates with various drone velocities.

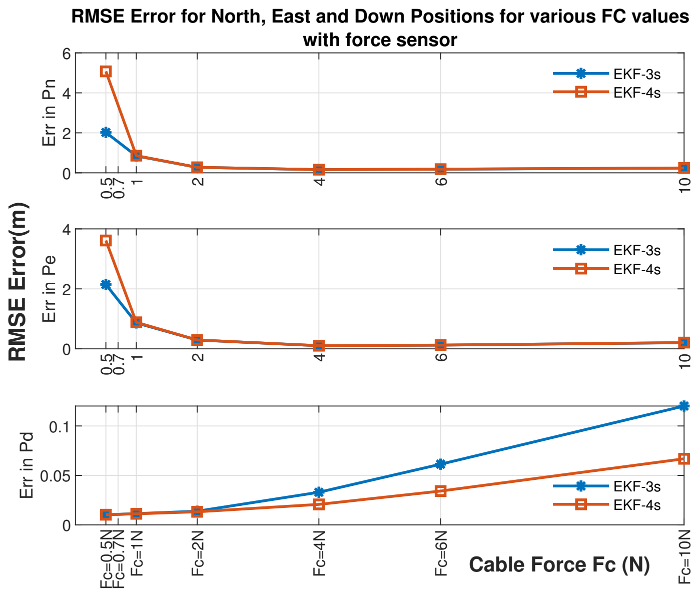


Figure 14. RMSE of 3D position estimates with various cable-tension force magnitudes.

7. Conclusions

In this paper, we present a self-localization technique for a tethered drone without a cable-force sensor in GPS-denied environments. To the best of our knowledge, this is one of the first works that estimate both the cable-tension force and the 3D location of a tethered drone without adding additional onboard sensors. A 4-state extended Kalman filter (EKF) was developed for the estimation, and its performance was compared with an existing 3-state EKF that assumes known cable-tension force. We also studied the impact of various cable-force values, altitudes, and velocities on the performance of both the proposed 4-state and the existing 3-state EKFs. The simulation results reveal that both EKFs produce the 3D drone position estimates with less than 0.3-m RMSE (root mean square error) and the cable-force estimates with less than 0.11 N RMSE, when the actual cable-tension force is greater than 1 N. When the actual cable-tension force is less than 2 N, the proposed 4-state EKF produces estimates with up to 5-m error for and the 3-state EKF with up to 2-m error.

This work facilitates the control and self-localization of a tethered drone by enabling the estimation of the cable-tension force, which eliminates the need of equipping a cable force sensor and reduces the complexity of the control system and data-acquisition system for a tethered drone. This ability makes possible the use of a tethered drone in GPS-degraded/-denied environments for real-world applications that need precise self-localization information with the decimeter-level accuracy, such as agricultural chemical spraying, and wind-turbine and high-rise building cleaning.

8. Future Work

In our future work, we plan to further investigate the position and cable-tension force estimation problem by leveraging the sigma-point Kalman filtering techniques (e.g., the unscented Kalman filter) and machine learning techniques (e.g., the decision tree method). Our hope is that these techniques would improve the overall estimation accuracy, especially when the cable-tension force is lower than 2 N. Moreover, we will develop a hardware experimental platform to evaluate our proposed techniques on a real tethered drone.

Author Contributions: Conceptualization, L.S.; Methodology, A.A.-R. and L.S.; Data curation, A.A.-R.; Formal analysis, A.A.-R. and L.S.; Funding acquisition, L.S.; Investigation, A.A.-R. and L.S.; Methodology, A.A.-R. and L.S.; Software, A.A.-R.; Supervision, L.S.; Validation, A.A.-R. and L.S.; Visualization, A.A.-R.; Writing—original draft, A.A.-R.; Writing—review & editing, A.A.-R. and L.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by both the NASA New Mexico Space Grant Consortium (NMSGC) under the Research Infrastructure Development program, and the NSF I-Corps program (Award # 1950161).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available upon request to the corresponding author.

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

Appendix A

The discrete-time ARX model is given by

$$A(z)y(t) = B(z)u(t) + e(t)A(z) \tag{A1}$$

where

$$\begin{aligned}
A(z) &= 1 - 1.914(\pm 0.004434)z^{-1} && +0.9297(\pm 0.004426)z^{-2} && \text{(A2)}\\
B1(z) &= 0.0007286(\pm 7.391\times 10^{-5})z^{-1} && -0.0007004(\pm 7.409\times 10^{-5})z^{-2} && \text{(A3)}\\
B2(z) &= 0.0006718(\pm 7.234\times 10^{-5})z^{-1} && -0.0006295(\pm 7.234\times 10^{-5})z^{-2} && \text{(A4)}\\
B3(z) &= 0.001004(\pm 7.951\times 10^{-5})z^{-1} && -0.0009491(\pm 7.963\times 10^{-5})z^{-2} && \text{(A5)}\\
B4(z) &= 0.0008196(\pm 7.596\times 10^{-5})z^{-1} && -0.0007239(\pm 7.639\times 10^{-5})z^{-2} && \text{(A6)}
\end{aligned}$$

The discrete-time identified state-space model is given by

$$x(t+Ts) = Ax(t) + Bu(t) + Ke(t) \tag{A7}$$
$$y(t) = Cx(t) + Du(t) + e(t) \tag{A8}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0.3136 \pm 0.03151 & -0.9517 \pm 0.1516 & 0.3403 \pm 0.3207 & 2.394 \pm 0.371 & -4.498 \pm 0.2339 & 3.399 \pm 0.0638 \end{bmatrix} \tag{A9}$$

$$B = \begin{bmatrix} -0.003938 \pm 0.0004003 & -0.002054 \pm 0.0003904 & -0.005297 \pm 0.0004491 & -0.005127 \pm 0.0004363 \\ 0.002901 \pm 0.0003992 & 0.0006675 \pm 0.0003899 & 0.003552 \pm 0.0004534 & 0.003335 \pm 0.0004453 \\ 0.0008814 \pm 0.0002329 & 0.001305 \pm 0.0001607 & 0.001033 \pm 0.0003003 & 0.001325 \pm 0.000276 \\ 0.002195 \pm 0.0002164 & 0.001693 \pm 0.0001207 & 0.002747 \pm 0.0002748 & 0.002982 \pm 0.0002602 \\ 0.0006869 \pm 0.0001376 & 0.001096 \pm 0.00011 & 0.0007627 \pm 0.0001723 & 0.001135 \pm 0.000163 \\ 0.0003934 \pm 0.0001215 & 0.0005386 \pm 9.06 \times 10^{-5} & 0.0003074 \pm 0.0001414 & 0.0006126 \pm 0.0001345 \end{bmatrix} \tag{A10}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A11}$$

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \tag{A12}$$

$$K = \begin{bmatrix} 3.54 \pm 0.02501 \\ 7.881 \pm 0.07874 \\ 13.91 \pm 0.1519 \\ 20.67 \pm 0.2448 \\ 26.98 \pm 0.3524 \\ 31.68 \pm 0.4601 \end{bmatrix}. \tag{A13}$$

References

1. ELISTAIR. Available online: https://elistair.com/orion-tethered-drone/ (accessed on 1 September 2021).
2. Prior, S.D. Tethered drones for persistent aerial surveillance applications. In *Defence Global*; Barclay Media Limited: Manchester, UK, 2015; pp. 78–79.
3. Al Nuaimi, O.; Almelhi, O.; Almarzooqi, A.; Al Mansoori, A.A.S.; Sayadi, S.; Swamidoss, I. Persistent surveillance with small Unmanned Aerial Vehicles (sUAV): A feasibility study. In *Electro-Optical Remote Sensing XII*; International Society for Optics and Photonics: Berlin, Germany, 2018; Volume 10796, p. 107960K.
4. Tarchi, D.; Guglieri, G.; Vespe, M.; Gioia, C.; Sermi, F.; Kyovtorov, V. Search and rescue: Surveillance support from RPAs radar. In Proceedings of the 2017 European Navigation Conference (ENC), Lausanne, Switzerland, 9–12 May 2017; pp. 256–264.
5. Dorn, L. Heavy Duty Tethered Cleaning Drones That Safely Wash Windows of High Altitude Skyscrapers. 2021. Available online: https://laughingsquid.com/aerones-skyscraper-window-washing-drone/ (accessed on 1 September 2021).
6. Kumparak, G. Lucid's Drone Is Built to Clean the Outside of Your House or Office. 2019. Available online: https://techcrunch.com/2019/08/27/lucids-drone-is-built-to-clean-the-outside-of-your-house-or-office/ (accessed on 1 September 2021).
7. What Are the Benefits of Tethered Drones? 2021. Available online: https://elistair.com/tethered-drones-benefits/ (accessed on 1 September 2021).

8. This Drone Can Clean Wind Turbines. 2018. Available online: https://www.irishnews.com/magazine/technology/2018/03/27/news/this-drone-can-clean-wind-turbines-1289122/ (accessed on 1 September 2021).
9. Drones and Robots that Clean Wind Turbines. 2021. Available online: https://www.nanalyze.com/2019/12/drones-robots-clean-wind-turbines/ (accessed on 1 September 2021).
10. Reagan, P.B.J. Fotokite Launches Tethered Drone System for Firefighters. 2019. Available online: https://dronelife.com/2019/04/17/fotokite-launches-tethered-drone-system-for-firefighters/ (accessed on 1 September 2021).
11. Estrada, C.; Sun, L. Trajectory Tracking Control of a Drone-Guided Hose System for Fluid Delivery. In Proceedings of the AIAA Scitech 2021 Forum, Nashville, TN, USA, 11–15 January 2021; p. 1003.
12. Al-Radaideh, A.; Al-Jarrah, M.; Jhemi, A. UAV testbed building and development for research purposes at the american university of sharjah. In Proceedings of the ISMA'10 7th International Symposium on Mechatronics and Its Applications, Sharjah, United Arab Emirates, 20–22 April 2010.
13. Al-Radaideh, A. Guidance, Control and Trajectory Tracking of Small Fixed Wing Unmanned Aerial Vehicles (UAV's) A THESIS IN MECHATRONICS. Master's Thesis, American University of Sharjah, Sharjah, United Arab Emirates, 2009.
14. Al-Radaideh, A.; Al-Jarrah, M.A.; Jhemi, A.; Dhaouadi, R. ARF60 AUS-UAV modeling, system identification, guidance and control: Validation through hardware in the loop simulation. In Proceedings of the 2009 6th International Symposium on Mechatronics and Its Applications, Sharjah, United Arab Emirates, 23–26 March 2009; pp. 1–11.
15. Martin, P.; Salaün, E. The true role of accelerometer feedback in quadrotor control. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 1623–1629.
16. Hoffmann, G.; Huang, H.; Waslander, S.; Tomlin, C. Quadrotor helicopter flight dynamics and control: Theory and experiment. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, CA, USA, 15–18 August 2007.
17. Escareno, J.; Salazar-Cruz, S.; Lozano, R. Embedded control of a four-rotor UAV. In Proceedings of the 2006 American Control Conference, Minneapolis, MN, USA, 14–16 June 2006. [CrossRef]
18. He, R.; Prentice, S.; Roy, N. Planning in information space for a quadrotor helicopter in a GPS-denied environment. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 1814–1820. [CrossRef]
19. Grzonka, S.; Grisetti, G.; Burgard, W. Towards a navigation system for autonomous indoor flying. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 2878–2883.
20. Bouabdallah, S.; Siegwart, R. Full control of a quadrotor. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 153–158.
21. Guenard, N.; Hamel, T.; Mahony, R. A practical visual servo control for an unmanned aerial vehicle. *IEEE Trans. Robot.* **2008**, *24*, 331–340. [CrossRef]
22. Kendoul, F.; Fantoni, I.; Nonami, K. Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles. *Robot. Auton. Syst.* **2009**, *57*, 591–602. [CrossRef]
23. Xu, Q.; Wang, Z.; Gerber, A.; Mao, Z.M. Cellular Data Network Infrastructure Characterization and Implication on Mobile Content Placement. In Proceedings of the ACM SIGMETRICS 2011 International Conference on Measurement and Modeling of Computer Systems, San Jose, CA, USA, 7–11 June 2011; pp. 317—328.
24. Lupashin, S.; D'Andrea, R. Stabilization of a flying vehicle on a taut tether using inertial sensing. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 2432–2438. [CrossRef]
25. Tognon, M.; Dash, S.S.; Franchi, A. Observer-Based Control of Position and Tension for an Aerial Robot Tethered to a Moving Platform. *IEEE Robot. Autom. Lett.* **2016**, *1*, 732–737. [CrossRef]
26. Lima, R.R.; Pereira, G.A. On the Development of a Tether-based Drone Localization System. In Proceedings of the 2021 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 15–18 June 2021; pp. 195–201.
27. Amalthea, A. World Premier: Tethered Drones at Paris Airports. Available online: https://elistair.com/tethered-drone-at-paris-airports/ (accessed on 1 September 2021).
28. Hoverfly—Tethered Drone Technology for Infinite Flight Time. Available online: https://hoverflytech.com/ (accessed on 1 September 2021).
29. Al-Radaidehl, A.; Sun, L. Observability Analysis and Bayesian Filtering for Self-Localization of a Tethered Multicopter in GPS-Denied Environments. In Proceedings of the 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 11–14 June 2019; pp. 1041–1047.
30. Al-Radaideh, A.; Sun, L. Self-localization of a tethered quadcopter using inertial sensors in a GPS-denied environment. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 271–277. [CrossRef]
31. Jeurgens, N.L.M. Implementing a Simulink controller in an AR. Drone 2.0. Master's Thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2016.
32. Capello, E.; Park, H.; Tavora, B.; Guglieri, G.; Romano, M. Modeling and experimental parameter identification of a multicopter via a compound pendulum test rig. In Proceedings of the 2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), Cancun, Mexico, 23–25 November 2015; pp. 308–317.

33. Chovancová, A.; Fico, T.; Chovanec, L.; Hubinsk, P. Mathematical modelling and parameter identification of quadrotor (a survey). *Procedia Eng.* **2014**, *96*, 172–181. [CrossRef]
34. Elsamanty, M.; Khalifa, A.; Fanni, M.; Ramadan, A.; Abo-Ismail, A. Methodology for identifying quadrotor parameters, attitude estimation and control. In Proceedings of the 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Wollongong, NSW, Australia, 9–12 July 2013; pp. 1343–1348.
35. Beard, R.W.; McLain, T.W. *Small Unmanned Aircraft: Theory and Practice*; Princeton University Press: Princeton, NJ, USA, 2012.
36. Rauw, M.O. *FDC 1.2-A Simulink Toolbox for Flight Dynamics and Control Analysis*; Delft University of Technology: Delft, The Netherlands, 2001; pp. 1–7.
37. Levy, L.J. The Kalman filter: Navigation's integration workhorse. *GPS World* **1997**, *8*, 65–71.
38. Orderud, F. *Comparison of Kalman Filter Estimation Approaches for State Space Models with Nonlinear Measurements*; Fagbokforlaget Vigmostad & Bjorke AS: Trondheim, Norway, 2005; pp. 1–8.
39. Angarita, J.E.; Schroeder, K.; Black, J. Quadrotor Model Generation using System Identification Techniques. In Proceedings of the 2018 AIAA Modeling and Simulation Technologies Conference, Kissimmee, FL, USA, 8–12 January 2018; p. 1917.
40. Ljung, L. Approaches to identification of nonlinear systems. In Proceedings of the 29th Chinese Control Conference, Beijing, China, 29–31 July 2010; pp. 1–5.
41. Ljung, L. *Identification of Nonlinear Systems*; Linköping University Electronic Press: Linköping, Sweden, 2007.
42. Al-Radaideh, A.; Jhemi, A.; Al-Jarrah, M.A. System identification of the Joker-3 unmanned helicopter. In Proceedings of the AIAA Modeling and Simulation Technologies Conference, Minneapolis, MN, USA, 13–16 August 2012; p. 4725.
43. Simmons, B.M. System Identification of a Nonlinear Flight Dynamics Model for a Small, Fixed-Wing UAV. Ph.D. Thesis, Virginia Tech, Blacksburg, Virginia, 2018.
44. Andersson, L.; Jönsson, U.; Johansson, K.H.; Bengtsson, J. *A Manual for System Identification*; Laboratory Exercises in System Identification. KF Sigma i Lund AB. Department of Automatic Control, Lund Institute of Technology: Lund, Sweden, 2006, Volume 118.
45. L'Erario, G.; Fiorio, L.; Nava, G.; Bergonti, F.; Mohamed, H.A.O.; Benenati, E.; Traversaro, S.; Pucci, D. Modeling, Identification and Control of Model Jet Engines for Jet Powered Robotics. *IEEE Robot. Autom. Lett.* **2020**, *5*, 2070–2077. [CrossRef]
46. Ljung, L. *System Identification Toolbox. Getting Started Guide Release 2017*; The MathWorks, Inc.: Natick, MA, USA, 2017.
47. Ljung, L. Identification for control: Simple process models. In Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, NV, USA, 10–13 December 2002; Volume 4, pp. 4652–4657.
48. Jones, R.H. Fitting autoregressions. *J. Am. Stat. Assoc.* **1975**, *70*, 590–592.
49. Poli, A.A.; Cirillo, M.C. On the use of the normalized mean square error in evaluating dispersion model performance. *Atmos. Environ. Part A Gen. Top.* **1993**, *27*, 2427–2434. [CrossRef]
50. Monajjemi, M. Ardrone Autonomy: A Ros Driver for Ardrone 1.0 & 2.0. 2012. Available online: https://github.com/AutonomyLab/ardrone_autonomy (accessed on 1 September 2021).