

Journal of Advanced Transportation

Unmanned Aircraft System and its Applications in Transportation

Lead Guest Editor: Daiheng Ni

Guest Editors: Guizhen Yu and Sivakumar Rathinam





Unmanned Aircraft System and its Applications in Transportation

Journal of Advanced Transportation

Unmanned Aircraft System and its Applications in Transportation

Lead Guest Editor: Daiheng Ni

Guest Editors: Guizhen Yu and Sivakumar Rathinam



Copyright © 2017 Hindawi. All rights reserved.

This is a special issue published in "Journal of Advanced Transportation." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Editorial Board

Francesco Bella, Italy
Klaus Bogenberger, Germany
Juan C. Cano, Spain
Giulio E. Cantarella, Italy
Anthony Chen, USA
Steven I. Chien, USA
Antonio Comi, Italy
Gonçalo H. Correia, Netherlands
Emanuele Crisostomi, Italy
Luca D'Acierno, Italy
Andrea D'Ariano, Italy
Alexandre De Barros, Canada
Luigi Dell'Olio, Spain
Cédric Demonceaux, France

Sunder Lall Dhingra, India
Yuchuan Du, China
Juan-Antonio Escareno, France
David F. Llorca, Spain
Jérôme Härri, France
Serge Hoogendoorn, Netherlands
Ángel Ibeas, Spain
Lina Kattan, Canada
Victor L. Knoop, Netherlands
Ludovic Leclercq, France
Seungjae Lee, Republic of Korea
Zhi-Chun Li, China
Yue Liu, USA
Jose R. Martinez-De-Dios, Spain

Andrea Monteriù, Italy
Jose E. Naranjo, Spain
Dongjoo Park, Republic of Korea
Paola Pellegrini, France
Luca Pugi, Italy
Prianka N. Seneviratne, Philippines
Wai Yuen Szeto, Hong Kong
Richard S. Tay, Australia
Martin Trépanier, Canada
Pascal Vasseur, France
Antonino Vitetta, Italy
S. Travis Waller, Australia
Chien-Hung Wei, Taiwan
Jacek Zak, Poland

Contents

Unmanned Aircraft System and Its Applications in Transportation

Daiheng Ni, Guizhen Yu, and Sivakumar Rathinam
Volume 2017, Article ID 7156153, 2 pages

Suspended Load Path Tracking Control Strategy Using a Tilt-Rotor UAV

M. A. Santos, B. S. Rego, G. V. Raffo, and A. Ferramosca
Volume 2017, Article ID 9095324, 22 pages

Doppler Effect-Based Automatic Landing Procedure for UAV in Difficult Access Environments

Jan M. Kelner and Cezary Ziółkowski
Volume 2017, Article ID 8092718, 9 pages

Car Detection from Low-Altitude UAV Imagery with the Faster R-CNN

Yongzheng Xu, Guizhen Yu, Yunpeng Wang, Xinkai Wu, and Yalong Ma
Volume 2017, Article ID 2823617, 10 pages

Using UAV-Based Systems to Monitor Air Pollution in Areas with Poor Accessibility

Oscar Alvear, Nicola Roberto Zema, Enrico Natalizio, and Carlos T. Calafate
Volume 2017, Article ID 8204353, 14 pages

Intelligent Controller Design for Quad-Rotor Stabilization in Presence of Parameter Variations

Oualid Doukhi, Abdur Razzaq Fayjie, and Deok Jin Lee
Volume 2017, Article ID 4683912, 10 pages

Conceptual Design of a Small Hybrid Unmanned Aircraft System

Umberto Papa, Salvatore Ponte, and Giuseppe Del Core
Volume 2017, Article ID 9834247, 10 pages

Surrogate Safety Analysis of Pedestrian-Vehicle Conflict at Intersections Using Unmanned Aerial Vehicle Videos

Peng Chen, Weiliang Zeng, Guizhen Yu, and Yunpeng Wang
Volume 2017, Article ID 5202150, 12 pages

Editorial

Unmanned Aircraft System and Its Applications in Transportation

Daiheng Ni,¹ Guizhen Yu,² and Sivakumar Rathinam³

¹University of Massachusetts, Amherst, MA, USA

²Beihang University, Beijing, China

³Texas A&M University, College Station, TX, USA

Correspondence should be addressed to Daiheng Ni; ni@ecs.umass.edu

Received 11 October 2017; Accepted 11 October 2017; Published 1 November 2017

Copyright © 2017 Daiheng Ni et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Within the past few years, civilian demand for unmanned aircraft system (UAS), commonly referred to as drones, has skyrocketed. Previously only used for military reconnaissance and later for strike programs, UAS has increasingly been considered for a variety of civilian tasks, including infrastructure monitoring, precision agriculture, package delivery services, search and rescue operations, photography, and more. Among many public and private sector agencies, transportation agencies are in a unique position to leverage the emerging technology, due to the nature of this profession, the vast demand, and the great benefit in terms of reducing accidents, mitigating congestion, and cost saving. According to a survey by the American Association of State Highway and Transportation Officials (AASHTO), 33 state departments of transportation (DOTs) have carried out or are exploring applications of UASs in various aspects of transportation, including inspecting bridges, collecting traffic data, and helping crash clear-up.

The purpose of this special issue is to publish high-quality research papers as well as review articles addressing recent advances on UAS and its applications in transportation. From those submitted manuscripts, we have chosen the following six papers to publish on this special issue:

C. Ziółkowski and J. M. Kelner proposed a solution to address UAS vertical take-off and landing. The authors showed the structure of an autonomous system and a Doppler-based navigation procedure that allows for automatic landing approaches. An accuracy evaluation of the

developed solution for VTOL was made on the basis of simulation studies.

P. Chen et al. applied UAS to provide a bird's eye view at an urban intersection, based on which a surrogate safety analysis was conducted on pedestrian-vehicle conflicts. The findings demonstrated that UAS can support intersection safety analysis in an accurate and cost-effective way.

Y. Xu et al. employed UAS flying at low-altitude in combination with region-based convolutional neural network to detect vehicles at signalized intersections. Their test results demonstrated that this approach is robust to illumination changes and cars' in-plane rotation and thus can be applied for vehicle detection from both static and moving UAV platforms.

O. Alvear A. et al. proposed the use of UAVs equipped with off-the-shelf sensors to perform air pollution monitoring tasks. Experimental results showed that an implicit priority guides the construction of pollution maps by focusing on areas where the pollutants' concentration is higher. This way, accurate maps can be constructed in a faster manner when compared to other strategies.

U. Papa et al. reported the conceptual design of a low-cost, electrically powered hybrid unmanned aircraft system. Its capability of extended cruise endurance was investigated by analyzing the design drivers affecting the craft flight time. Experimental results in different testing scenarios and complex environments showed 50% improvement of the flight duration.

O. Doukhi et al. presented the mathematical model of a quadrotor UAV and the design of robust Self-Tuning PID controller. The proposed controller was applied in the inner and outer loop for heading and position trajectory tracking control to handle the external disturbances. The results of numerical simulation demonstrated the effectiveness of this intelligent control strategy.

Daiheng Ni
Guizhen Yu
Sivakumar Rathinam

Research Article

Suspended Load Path Tracking Control Strategy Using a Tilt-Rotor UAV

M. A. Santos,¹ B. S. Rego,¹ G. V. Raffo,^{1,2} and A. Ferramosca³

¹Graduate Program in Electrical Engineering, Federal University of Minas Gerais, 31270-901 Belo Horizonte, MG, Brazil

²Department of Electronic Engineering, Federal University of Minas Gerais, 31270-901 Belo Horizonte, MG, Brazil

³CONICET-Universidad Tecnológica Nacional (UTN), Facultad Regional de Reconquista, 27 de Abril 1000, 3560 Reconquista, Santa Fe, Argentina

Correspondence should be addressed to G. V. Raffo; raffo@ufmg.br

Received 24 March 2017; Accepted 12 June 2017; Published 19 October 2017

Academic Editor: Guizhen Yu

Copyright © 2017 M. A. Santos et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work proposes a control strategy to solve the path tracking problem of a suspended load carried by a tilt-rotor unmanned aerial vehicle (UAV). Initially, the equations of motion for the multibody mechanical system are derived from the load's perspective by means of the Euler-Lagrange formulation, in which the load's position and orientation are chosen as degrees of freedom. An unscented Kalman filter (UKF) is designed for nonlinear state estimation of all the system states, assuming that available information is provided by noisy sensors with different sampling rates that do not directly measure the load's attitude. Furthermore, a model predictive control (MPC) strategy is proposed for path tracking of the suspended load with stabilization of the tilt-rotor UAV when parametric uncertainties and external disturbances affect the load, the rope's length and total system mass vary during taking-off and landing, and the desired yaw angle changes throughout the trajectory. Finally, numerical experiments are presented to corroborate the good performance of the proposed strategy.

1. Introduction

Firstly used mainly for military purposes, recent advances in technology allowed large-scale production of unmanned aerial vehicles (UAVs) and consequently their civilian usage. Nowadays, UAVs receive great interest from engineers and researchers and have a wide range of applications, such as precision agriculture, cargo transportation, cinematographic filming, search-and-rescue missions, surveillance, fire inspection, and archeology. The most commonly found configurations of UAVs are the fixed-wing, helicopter, and quadrotor ones. Due to the vertical take-off and landing (VTOL) and hovering capabilities of helicopter and quadrotor UAVs, they are used in tasks that require high maneuverability in slow velocities. Thereby, fixed-wing UAVs are used in tasks that require improved forward flight, mostly in open environments.

In the last decades, some hybrid aircrafts have drawn substantial attention due to their ability to perform VTOL,

hover, and improved forward flights. The tilt-rotor configuration is among the most popular ones, being provided with fixed wings and rotary wings, and is capable of switching between helicopter and airplane flight modes by tilting its thrusters. Inspired by the capabilities of those aircrafts, recent researches are looking into the design of small-scale tilt-rotor UAVs [1–3]. Their hybrid characteristics offer advantages over fixed-wing and quadrotor UAVs, which also come with design and control challenges since they are complex under-actuated mechanical systems with highly coupled dynamics.

Moreover, an important application of unmanned aerial vehicles is the load transportation in risky and inaccessible zones, allowing dealing with rapid deployment of supplies in search-and-rescue missions [4], vertical replenishment of seaborne vessels [5], and safe landmine detection [6]. However, this kind of task is a challenging subject in terms of modeling and control. Since the payload is often connected to the UAV through a rope, the dynamic behavior of the system varies due to the load's swing, which can destabilize the whole

system if it is not well attenuated. Moreover, the suspended load by a rope adds unactuated degrees of freedom to the system, increasing its underactuation degree. Apart from the exposed, in order to accomplish the load transportation task, the knowledge on the load position is usually required. The problem of estimating the load position then arises, being mostly addressed through visual systems and state estimators [7, 8].

Therefore, due to its hybrid capabilities, a tilt-rotor UAV becomes a promising platform for aerial load transportation, providing improved forward speed when compared with rotary-wing UAVs, which is a desired feature for missions requiring rapid deployment. Furthermore, missions that demand precise positioning of the load cannot be addressed by fixed-wing UAVs, since they are not able to perform hover flights. Thereby, the tilt-rotor UAV can be used in both scenarios.

In the literature, control objectives for aerial load transportation include path tracking of the aircraft with reduced load's swing [9–14], obstacle avoidance [15, 16], transportation by multiple aircrafts [17, 18], and path tracking of the suspended load [19–24]. Since large load swing may destabilize the aircraft, many works address the problem of swing-free aerial load transportation. In [9], a control strategy based on a cascade structure is proposed for load transportation using an unmanned helicopter. A delayed feedback controller is introduced in an outer loop to reduce load's swing, while the inner-loop controller does not take into account the dynamics of the load. Open-loop approaches based on trajectory generation for load transportation using quadrotor UAVs are proposed in [10, 11], in order to obtain reduced load's swing motion. In [13], a robust nonlinear control strategy is proposed in order to transport a suspended load by a quadrotor UAV along a predefined trajectory while avoiding load's swing. The controller is designed based on nonlinear \mathcal{H}_∞ control theory and Lyapunov redesign taking into account the whole dynamics of the system. Some works have also used tilt-rotor UAVs for load transportation. In [12], a nonlinear cascade control strategy is proposed for path tracking of a tilt-rotor UAV with load's swing improvement, which is composed of three levels of feedback linearization controllers. In [14], a model predictive control (MPC) strategy is studied for path tracking of a tilt-rotor UAV while carrying a load.

When precise positioning of the load is of concern, swing-free motion may not be sufficient for the accomplishment of the task; then the problem of path tracking of the suspended load is addressed. For instance, in [19], the authors propose a model-free open-loop approach based on trajectory generation by reinforcement learning for path tracking of the suspended load using a quadrotor UAV. A nonlinear control strategy based on cascade structure and system decoupling is proposed in [20], also for path tracking of a suspended load using a quadrotor UAV. In [22], a nonlinear solution to the suspended load path tracking problem using a quadrotor UAV is presented, in which the authors assume the quadrotor as a system actuated by total thrust and orientation. The control problem is casted into the framework described in [21], which deals with path tracking of underactuated systems driven by directed thrust and angular velocity with

a double-integrator structure. By using tilt-rotor UAVs, in [23, 24], the path tracking problem of the suspended load is solved through the design of control and state estimation strategies based on linearized, time-invariant state-space equations, which did not allow yaw angle tracking, neither the occurrence of changes in the load's mass and rope's length.

This work intends to solve the problem of suspended load path tracking considering a realistic scenario in which the rope's length and total system mass are not constant during take-off and landing maneuvers, the desired yaw angle varies during the mission execution, and the sensors embedded in the UAV have different sampling rates and do not measure directly the load's position and orientation. Due to the tilt-rotor UAV highly nonlinear dynamic behavior, the unscented Kalman filter is chosen as state estimation strategy since it is not based on model linearization. Further, to perform path tracking while dealing with yaw regulation and the rope's length variation without relying only on controller's robustness, the MPC strategy is selected because its model-based nature allows dealing with those problems in a simple fashion way. Therefore, the main contributions of this work are as follows: (i) a detailed modeling in which the multibody system's equations of motion are obtained from the load's point of view, yielding an input-affine state-space representation with the load's position and orientation as state variables; (ii) a nonlinear estimation strategy dealing with multirate noisy sensors that do not provide directly information about the load's pose; and (iii) a control strategy that is able to cope with the path tracking of the suspended load considering yaw angle regulation, to incorporate the rope's length variation to the control problem formulation, and to keep the UAV stabilized while rejecting external disturbances and parametric uncertainties.

The remainder of this paper is organized as follows: Section 2 develops the equations of motion of the tilt-rotor UAV with suspended load from the load's perspective, yielding a state-space representation with the load's position and orientation represented by state variables; Section 3 presents the unscented Kalman Filter design based on the model and measurement equations given by available sensors; Section 4 describes the model predictive control strategy designed to achieve path tracking of the suspended load based on the linearized time-varying model; Section 5 presents the model and design parameters, besides the simulation scenario used to validate the proposed control strategy; in Section 6, the numerical experiments used to demonstrate the good performance of the controller and filter are presented and discussed; finally, Section 7 concludes the work.

2. Tilt-Rotor UAV with Suspended Load Modeling

This section deals with the development of the equations of motion of the tilt-rotor UAV with suspended load, from the load's perspective. The dynamic equations, assuming a multibody system, are obtained through the Euler-Lagrange formulation, in which the coupling between the aircraft and the load is considered naturally.



FIGURE 1: The tilt-rotor UAV with suspended load (CAD model).

2.1. System Description. The computer-aided design (CAD) model of the tilt-rotor UAV with suspended load is shown in Figure 1. The system is regarded as a multibody mechanical system composed of four rigid bodies: (i) the aircraft's main body, composed of Acrylonitrile Butadiene Styrene (ABS) structure, landing gear, batteries, and electronics; (ii) the right thruster group, composed of the right thruster and its tilting mechanism (a revolute joint); (iii) the left thruster group, composed of the left thruster and its tilting mechanism; and (iv) the suspended load group, comprising the load and the rope. The system is actuated through the aircraft's thrusters and tilting mechanisms. For modeling purposes, the following assumptions are made:

- (A1) The rope is rigid and has negligible mass.
- (A2) The rope is connected to the aircraft's geometric center.
- (A3) The main body's center of mass does not coincide with the aircraft's geometric center.
- (A4) The thrusters groups' centers of mass are located at their respective tilting axes.

Regarding the assumptions stated above, it is important to describe the main reasons of assumption (A3), which are as follows: (i) the center of mass is vertically displaced in order to improve the pitch moment; and (ii) the center of mass is displaced with respect to the x -axis of the geometric center, allowing obtaining non-null equilibrium points for the angular positions of the tilting mechanisms and pitch angle. This mechanical feature improves the controllability of the system in hover flight, allowing the projection of the thrusts generated by propellers on the x -axis without the need to tilt the thrusters' group.

2.2. Kinematics from the Load's Perspective. The load's perspective approach consists in formulating the system's kinematics regarding the suspended load as a free rigid body, with the aircraft as a multibody mechanical system rigidly coupled to it. Six reference frames are defined, as shown in Figure 2: (i) the inertial reference frame, \mathcal{F} ; (ii) the aircraft's geometric center frame, \mathcal{B} ; (iii) the main body center of mass frame, \mathcal{E}_1 ; (iv) the right thruster group center of mass frame,

\mathcal{E}_2 ; (v) the left thruster group center of mass frame, \mathcal{E}_3 ; and (vi) the suspended load group center of mass frame, \mathcal{L} . The load's position with respect to the inertial frame \mathcal{F} is denoted by $\boldsymbol{\xi} \triangleq [x \ y \ z]^T$. The displacement vector from \mathcal{L} to \mathcal{B} corresponds to the rope and is expressed in \mathcal{L} by $\mathbf{d}_{\mathcal{B}}^{\mathcal{L}} \triangleq [0 \ 0 \ l]^T$, with l as the rope's length. The displacement vectors from \mathcal{B} to \mathcal{E}_i are model parameters of the tilt-rotor UAV and are denoted by $\mathbf{d}_{\mathcal{E}_i}^{\mathcal{B}}$, expressed in \mathcal{B} , with $i \in \{1, 2, 3\}$.

The orientation of the load with respect to \mathcal{F} is parametrized by Euler angles, $\boldsymbol{\eta} \triangleq [\phi \ \theta \ \psi]^T$, using the ZYX convention about local axes. The associated rotation matrix is given by

$$\begin{aligned} \mathbf{R}_{\mathcal{L}}^{\mathcal{F}} &\triangleq \mathbf{R}_{z,\psi} \mathbf{R}_{y,\theta} \mathbf{R}_{x,\phi} \\ &= \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}. \end{aligned} \quad (1)$$

On the other hand, the orientation of the aircraft's geometric center frame with respect to \mathcal{L} , corresponding to the orientation of the UAV with respect to the rope, is parametrized by two angles, $\boldsymbol{\gamma} \triangleq [\gamma_1 \ \gamma_2]^T$, such that

$$\mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \triangleq \mathbf{R}_{x,-\gamma_1} \mathbf{R}_{y,-\gamma_2} = \begin{bmatrix} c_{\gamma_2} & 0 & -s_{\gamma_2} \\ s_{\gamma_1} s_{\gamma_2} & c_{\gamma_1} & s_{\gamma_1} c_{\gamma_2} \\ c_{\gamma_1} s_{\gamma_2} & -s_{\gamma_1} & c_{\gamma_1} c_{\gamma_2} \end{bmatrix}. \quad (2)$$

The reference frames \mathcal{B} and \mathcal{E}_1 are parallel to each other and attached to the same rigid body; thus the relative orientation is null, that is, $\mathbf{R}_{\mathcal{E}_1}^{\mathcal{B}} = \mathbb{1}_{3 \times 3}$ (in this work $\mathbb{1}_{n \times n}$ is an identity matrix with dimension n , $\mathbf{0}_{n \times m}$ is n by m zeros matrix, and $\mathbb{1}_{n \times m}$ is n by m ones matrix). Furthermore, the orientations of the thrusters' groups with respect to \mathcal{B} are described by

$$\begin{aligned} \mathbf{R}_{\mathcal{E}_2}^{\mathcal{B}} &\triangleq \mathbf{R}_{x,-\beta} \mathbf{R}_{y,\alpha_R} = \begin{bmatrix} c_{\alpha_R} & 0 & s_{\alpha_R} \\ -s_\beta s_{\alpha_R} & c_\beta & s_\beta c_{\alpha_R} \\ -c_\beta s_{\alpha_R} & -s_\beta & c_\beta c_{\alpha_R} \end{bmatrix}, \\ \mathbf{R}_{\mathcal{E}_3}^{\mathcal{B}} &\triangleq \mathbf{R}_{x,\beta} \mathbf{R}_{y,\alpha_L} = \begin{bmatrix} c_{\alpha_L} & 0 & s_{\alpha_L} \\ s_\beta s_{\alpha_L} & c_\beta & -s_\beta c_{\alpha_L} \\ -c_\beta s_{\alpha_L} & s_\beta & c_\beta c_{\alpha_L} \end{bmatrix}, \end{aligned} \quad (3)$$

where α_R and α_L are the tilting angles of the right and left thrusters, respectively, and β is a fixed inclination angle of the thrusters towards the aircraft geometric center, designed to improve the aircraft's controllability [25]. The angular velocities of the system are given by $\boldsymbol{\omega}_{\mathcal{F}\mathcal{L}}^{\mathcal{L}} = \mathbf{W}_\eta \dot{\boldsymbol{\eta}}$, $\boldsymbol{\omega}_{\mathcal{L}\mathcal{B}}^{\mathcal{B}} = \mathbf{Q} \dot{\boldsymbol{\gamma}}$, $\boldsymbol{\omega}_{\mathcal{B}\mathcal{E}_1}^{\mathcal{E}_1} = \mathbf{0}_{3 \times 1}$, $\boldsymbol{\omega}_{\mathcal{B}\mathcal{E}_2}^{\mathcal{E}_2} = \mathbf{a}_y \dot{\alpha}_R$, and $\boldsymbol{\omega}_{\mathcal{B}\mathcal{E}_3}^{\mathcal{E}_3} = \mathbf{a}_y \dot{\alpha}_L$, where, for

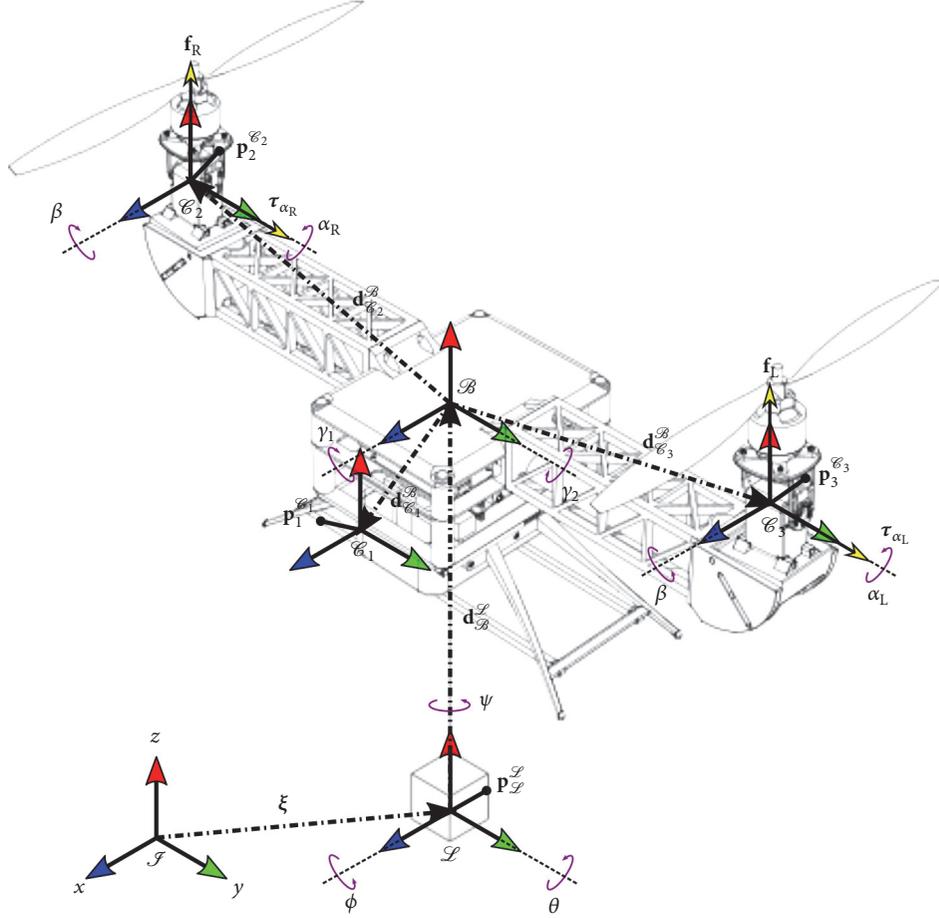


FIGURE 2: Kinematic definitions, input forces, and torques.

instance, $\boldsymbol{\omega}_{\mathcal{L}\mathcal{L}}^{\mathcal{L}}$ denotes the angular velocity of \mathcal{L} with respect to \mathcal{L} , expressed in \mathcal{L} , with

$$\mathbf{W}_{\eta} \triangleq \begin{bmatrix} 1 & 0 & -s_{\theta} \\ 0 & c_{\phi} & s_{\phi}c_{\theta} \\ 0 & -s_{\phi} & c_{\phi}c_{\theta} \end{bmatrix}, \quad \mathbf{Q} \triangleq \begin{bmatrix} -c_{\gamma_2} & 0 \\ 0 & -1 \\ s_{\gamma_2} & 0 \end{bmatrix}, \quad (4)$$

and $\mathbf{a}_y \triangleq [0 \ 1 \ 0]^T$.

From the rigid transformations of the system, the forward kinematics of points that belong to each rigid body are given by

$$\mathbf{p}_{\mathcal{L}}^{\mathcal{L}} = \boldsymbol{\xi} + \mathbf{R}_{\mathcal{L}}^{\mathcal{L}} \mathbf{p}_{\mathcal{L}}^{\mathcal{L}}, \quad (5)$$

$$\mathbf{p}_i^{\mathcal{L}} = \boldsymbol{\xi} + \mathbf{R}_{\mathcal{L}}^{\mathcal{L}} \mathbf{d}_{\mathcal{B}}^{\mathcal{L}} + \mathbf{R}_{\mathcal{L}}^{\mathcal{L}} \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{d}_{\mathcal{E}_i}^{\mathcal{B}} + \mathbf{R}_{\mathcal{L}}^{\mathcal{L}} \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{R}_{\mathcal{E}_i}^{\mathcal{B}} \mathbf{p}_i^{\mathcal{E}_i}, \quad (6)$$

where $\mathbf{p}_{\mathcal{L}}$ is the position of a point that belongs to the suspended load body and \mathbf{p}_i belongs to the rigid body with attached frame \mathcal{E}_i . Taking the time derivatives of (5) and

(6) and making use of several properties of skew-symmetric matrices [26] yield

$$\dot{\mathbf{p}}_{\mathcal{L}}^{\mathcal{L}} = \dot{\boldsymbol{\xi}} + \mathbf{R}_{\mathcal{L}}^{\mathcal{L}} \mathbf{S}(\mathbf{p}_{\mathcal{L}}^{\mathcal{L}})^T \boldsymbol{\omega}_{\mathcal{L}\mathcal{L}}^{\mathcal{L}}, \quad (7)$$

$$\begin{aligned} \dot{\mathbf{p}}_i^{\mathcal{L}} = & \dot{\boldsymbol{\xi}} + \left[\mathbf{R}_{\mathcal{L}}^{\mathcal{L}} \mathbf{S}(\mathbf{d}_{\mathcal{B}}^{\mathcal{L}})^T + \mathbf{R}_{\mathcal{L}}^{\mathcal{L}} \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{S}(\mathbf{d}_{\mathcal{E}_i}^{\mathcal{B}})^T (\mathbf{R}_{\mathcal{E}_i}^{\mathcal{B}})^T \right] \boldsymbol{\omega}_{\mathcal{L}\mathcal{L}}^{\mathcal{L}} \\ & + \mathbf{R}_{\mathcal{L}}^{\mathcal{L}} \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{R}_{\mathcal{E}_i}^{\mathcal{B}} \mathbf{S}(\mathbf{p}_i^{\mathcal{E}_i})^T (\mathbf{R}_{\mathcal{E}_i}^{\mathcal{B}} \mathbf{R}_{\mathcal{E}_i}^{\mathcal{B}})^T \boldsymbol{\omega}_{\mathcal{L}\mathcal{L}}^{\mathcal{L}} \\ & + \left[\mathbf{R}_{\mathcal{L}}^{\mathcal{L}} \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{S}(\mathbf{d}_{\mathcal{E}_i}^{\mathcal{B}})^T + \mathbf{R}_{\mathcal{L}}^{\mathcal{L}} \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{R}_{\mathcal{E}_i}^{\mathcal{B}} \mathbf{S}(\mathbf{p}_i^{\mathcal{E}_i})^T (\mathbf{R}_{\mathcal{E}_i}^{\mathcal{B}})^T \right] \boldsymbol{\omega}_{\mathcal{L}\mathcal{B}}^{\mathcal{B}} \\ & + \mathbf{R}_{\mathcal{L}}^{\mathcal{L}} \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{R}_{\mathcal{E}_i}^{\mathcal{B}} \mathbf{S}(\mathbf{p}_i^{\mathcal{E}_i})^T \boldsymbol{\omega}_{\mathcal{E}_i}^{\mathcal{E}_i}, \end{aligned} \quad (8)$$

where $\mathbf{S}(\cdot)$ denotes an operator that maps a vector to a skew-symmetric matrix [26].

The generalized coordinates of the system are chosen according to their degrees of freedom. For the equations of motion describing explicitly the time evolution of the load's position and orientation, these are included in the generalized

coordinates, which are chosen as

$$\mathbf{q} \triangleq [\xi^T \ \eta^T \ \gamma^T \ \alpha_R \ \alpha_L]^T \in \mathbb{R}^{10}. \quad (9)$$

Note that, due to the chosen perspective for the system kinematics, the position and orientation of the aircraft with respect to \mathcal{S} are not degrees of freedom of the system. Thus, they are not included in (9), and consequently their time evolution will not be described explicitly by the developed equations of motion.

2.3. Kinetic and Potential Energies. Initially, in order to derive the equations of motion through the Euler-Lagrange formulation, the kinetic and potential energies of each body of the mechanical system must be obtained. For each i th rigid body, these energies can be computed through the volume integrals [27]

$$\mathcal{K}_i = \frac{1}{2} \int_{V_i} \rho_i (\dot{\mathbf{p}}_i^{\mathcal{S}})^T (\dot{\mathbf{p}}_i^{\mathcal{S}}) dV_i, \quad (10)$$

$$\mathcal{U}_i = - \int_{V_i} \rho_i \hat{\mathbf{g}}^T \mathbf{p}_i^{\mathcal{S}} dV_i = -m_i \hat{\mathbf{g}}^T \mathbf{o}_{\mathcal{E}_i}^{\mathcal{S}}, \quad (11)$$

respectively, where ρ_i corresponds to its density, V_i to its volume, and $m_i = \int_{V_i} \rho_i dV_i$ to its mass, $\hat{\mathbf{g}} \triangleq [0 \ 0 \ -\hat{g}_z]^T$ denotes the gravitational acceleration vector expressed in \mathcal{S} , and $\mathbf{o}_{\mathcal{E}_i}^{\mathcal{S}}$ is the position vector obtained from the forward kinematics of the origin of \mathcal{E}_i .

The quadratic terms $(\dot{\mathbf{p}}_{\mathcal{S}}^{\mathcal{S}})^T \dot{\mathbf{p}}_{\mathcal{S}}^{\mathcal{S}}$ and $(\dot{\mathbf{p}}_i^{\mathcal{S}})^T \dot{\mathbf{p}}_i^{\mathcal{S}}$, $i \in \{1, 2, 3\}$, are obtained using (7) and (8), respectively. The kinetic energy of the load $\mathcal{K}_{\mathcal{S}}$ and the i th body of the aircraft \mathcal{K}_i are obtained through (10). The total kinetic energy of the system is then computed through $\mathcal{K} = \mathcal{K}_{\mathcal{S}} + \sum_{i=1}^3 \mathcal{K}_i$. Writing the result in the quadratic form $\mathcal{K} = (1/2) \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}}$, and by defining the inertia tensors $\mathbf{I}_{\mathcal{S}} = - \int_{V_{\mathcal{S}}} \rho_{\mathcal{S}} S(\mathbf{p}_{\mathcal{S}}^{\mathcal{S}})^2 dV_{\mathcal{S}}$ and $\mathbf{I}_i = - \int_{V_i} \rho_i S(\mathbf{p}_i^{\mathcal{E}_i})^2 dV_i$, and taking into account the parallel axis theorem [28], yielding $\mathbf{J}_i \triangleq -m_i S(\mathbf{d}_{\mathcal{E}_i}^{\mathcal{S}})^2 + \mathbf{R}_{\mathcal{E}_i}^{\mathcal{S}} \mathbf{I}_i (\mathbf{R}_{\mathcal{E}_i}^{\mathcal{S}})^T$ and $\mathbf{D}_i \triangleq -m_i S(\mathbf{d}_{\mathcal{E}_i}^{\mathcal{S}})^2 + \mathbf{R}_{\mathcal{E}_i}^{\mathcal{S}} \mathbf{J}_i (\mathbf{R}_{\mathcal{E}_i}^{\mathcal{S}})^T$, we have that the inertia matrix $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{10 \times 10}$ is given by

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} (m_{\mathcal{S}} + m) \mathbb{1}_{3 \times 3} & \mathbf{M}_{12} & -\mathbf{R}_{\mathcal{S}}^{\mathcal{S}} \mathbf{R}_{\mathcal{B}}^{\mathcal{S}} S(\mathbf{d}_m) \mathbf{Q} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ * & \mathbf{M}_{22} & \mathbf{M}_{23} & \mathbf{W}_{\eta}^T \mathbf{R}_{\mathcal{B}}^{\mathcal{S}} \mathbf{R}_{\mathcal{E}_2}^{\mathcal{S}} \mathbf{I}_{2\mathbf{a}_y} & \mathbf{W}_{\eta}^T \mathbf{R}_{\mathcal{B}}^{\mathcal{S}} \mathbf{R}_{\mathcal{E}_3}^{\mathcal{S}} \mathbf{I}_{3\mathbf{a}_y} \\ * & * & \mathbf{Q}^T \mathbf{J} \mathbf{Q} & \mathbf{Q}^T \mathbf{R}_{\mathcal{E}_2}^{\mathcal{S}} \mathbf{I}_{2\mathbf{a}_y} & \mathbf{Q}^T \mathbf{R}_{\mathcal{E}_3}^{\mathcal{S}} \mathbf{I}_{3\mathbf{a}_y} \\ * & * & * & \mathbf{a}_y^T \mathbf{I}_{2\mathbf{a}_y} & 0 \\ * & * & * & * & \mathbf{a}_y^T \mathbf{I}_{3\mathbf{a}_y} \end{bmatrix}, \quad (12)$$

with * denoting terms that are deduced by symmetry, and

$$\begin{aligned} \mathbf{M}_{12} &= -m \mathbf{R}_{\mathcal{S}}^{\mathcal{S}} S(\mathbf{d}_{\mathcal{B}}^{\mathcal{S}}) \mathbf{W}_{\eta} - \mathbf{R}_{\mathcal{S}}^{\mathcal{S}} \mathbf{R}_{\mathcal{B}}^{\mathcal{S}} S(\mathbf{d}_m) (\mathbf{R}_{\mathcal{B}}^{\mathcal{S}})^T \mathbf{W}_{\eta}, \\ \mathbf{M}_{22} &= \mathbf{W}_{\eta}^T [\mathbf{I}_{\mathcal{S}} + \mathbf{D} - S(\mathbf{d}_{\mathcal{B}}^{\mathcal{S}}) \mathbf{R}_{\mathcal{B}}^{\mathcal{S}} S(\mathbf{d}_m) (\mathbf{R}_{\mathcal{B}}^{\mathcal{S}})^T \\ &\quad - \mathbf{R}_{\mathcal{B}}^{\mathcal{S}} S(\mathbf{d}_m) (\mathbf{R}_{\mathcal{B}}^{\mathcal{S}})^T S(\mathbf{d}_{\mathcal{B}}^{\mathcal{S}})] \mathbf{W}_{\eta}, \\ \mathbf{M}_{23} &= \mathbf{W}_{\eta}^T [-S(\mathbf{d}_{\mathcal{B}}^{\mathcal{S}}) \mathbf{R}_{\mathcal{B}}^{\mathcal{S}} S(\mathbf{d}_m) + \mathbf{R}_{\mathcal{B}}^{\mathcal{S}} \mathbf{J}] \mathbf{Q}, \end{aligned} \quad (13)$$

where $m \triangleq \sum_{i=1}^3 m_i$, $\mathbf{J} \triangleq \sum_{i=1}^3 \mathbf{J}_i$, $\mathbf{D} \triangleq \sum_{i=1}^3 \mathbf{D}_i$, and $\mathbf{d}_m \triangleq \sum_{i=1}^3 m_i \mathbf{d}_{\mathcal{E}_i}^{\mathcal{S}}$. Note that, from the inertia matrix (12), the four body dynamics are completely coupled, allowing to consider this interaction in the control law design and avoid the need of cascade control structures.

The Coriolis and centripetal forces matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{10 \times 10}$, can be calculated via Christoffel symbols of the first kind [26]. The element from its k th row and j th column is computed through

$$C_{kj} = \sum_{i=1}^{10} \frac{1}{2} \left(\frac{\partial M_{kj}}{\partial q_i} + \frac{\partial M_{ki}}{\partial q_j} - \frac{\partial M_{ij}}{\partial q_k} \right) \dot{q}_i, \quad (14)$$

where M is an element of the inertia matrix and $k, j \in \{1, 2, \dots, 10\}$.

The forward kinematics of each body's center of mass is obtained by making $\mathbf{p}_{\mathcal{S}}^{\mathcal{S}} = \mathbf{0}_{3 \times 1}$ in (5) and $\mathbf{p}_i^{\mathcal{E}_i} = \mathbf{0}_{3 \times 1}$ in (6). Thereafter, the potential energies of the load and of each body of the aircraft are obtained using (11). The total potential energy of the system is then computed by $\mathcal{U} = \mathcal{U}_{\mathcal{S}} + \sum_{i=1}^3 \mathcal{U}_i = -\hat{\mathbf{g}}^T [(m_{\mathcal{S}} + m) \xi + m \mathbf{R}_{\mathcal{S}}^{\mathcal{S}} \mathbf{d}_{\mathcal{B}}^{\mathcal{S}} + \mathbf{R}_{\mathcal{S}}^{\mathcal{S}} \mathbf{R}_{\mathcal{B}}^{\mathcal{S}} \mathbf{d}_m]$.

Then, the gravitational force vector is given by

$$\mathbf{g}(\mathbf{q}) = \frac{\partial \mathcal{U}}{\partial \mathbf{q}} \in \mathbb{R}^{10}. \quad (15)$$

2.4. Generalized Forces. This section obtains the contributions to generalized forces of all nonconservative forces and torques that actuate on the tilt-rotor UAV with suspended load. Let $\mathbf{f} \in \mathbb{R}^3$ and $\boldsymbol{\tau} \in \mathbb{R}^3$ denote a nonconservative force and a nonconservative torque, respectively, $\mathbf{p} \in \mathbb{R}^3$ denote the point of application of \mathbf{f} , and \mathcal{F} be a reference frame rigidly attached to the body to which $\boldsymbol{\tau}$ is applied. According to [29], the contributions of \mathbf{f} and $\boldsymbol{\tau}$ to the generalized forces

can be computed through the following mappings:

$$\mathfrak{f}_f = (\mathcal{J}_p)^T \mathbf{f}^f \in \mathbb{R}^n, \quad (16)$$

$$\mathfrak{f}_\tau = (\mathcal{W}_{\mathcal{F}})^T \boldsymbol{\tau}^f \in \mathbb{R}^n, \quad (17)$$

where $\mathcal{J}_p \triangleq \partial \dot{\mathbf{p}}^f / \partial \dot{\mathbf{q}} \in \mathbb{R}^{3 \times n}$ and $\mathcal{W}_{\mathcal{F}} \triangleq \partial \boldsymbol{\omega}_{\mathcal{F}}^f / \partial \dot{\mathbf{q}} \in \mathbb{R}^{3 \times n}$.

The thrust forces generated by the aircraft's propellers, denoted by \mathbf{f}_R and \mathbf{f}_L , and the torques generated by the servomotors composing the tilting mechanisms, denoted by $\boldsymbol{\tau}_{\alpha_R}$ and $\boldsymbol{\tau}_{\alpha_L}$, correspond to the input forces and torques of the

system. Expressed in their respective thrusters' frames (see Figure 2), they are given by $\mathbf{f}_R^{\mathcal{E}_2} = \mathbf{a}_z f_R$, $\mathbf{f}_L^{\mathcal{E}_3} = \mathbf{a}_z f_L$, $\boldsymbol{\tau}_{\alpha_R}^{\mathcal{E}_2} = \mathbf{a}_y \tau_{\alpha_R}$, and $\boldsymbol{\tau}_{\alpha_L}^{\mathcal{E}_3} = \mathbf{a}_y \tau_{\alpha_L}$, where $\mathbf{a}_z \triangleq [0 \ 0 \ 1]^T$. In the inertial reference frame, these vectors are expressed as $\mathbf{f}_R^f = \mathbf{R}_{\mathcal{E}_2}^f \mathbf{f}_R^{\mathcal{E}_2}$, $\mathbf{f}_L^f = \mathbf{R}_{\mathcal{E}_3}^f \mathbf{f}_L^{\mathcal{E}_3}$, $\boldsymbol{\tau}_{\alpha_R}^f = \mathbf{R}_{\mathcal{E}_2}^f \boldsymbol{\tau}_{\alpha_R}^{\mathcal{E}_2}$, and $\boldsymbol{\tau}_{\alpha_L}^f = \mathbf{R}_{\mathcal{E}_3}^f \boldsymbol{\tau}_{\alpha_L}^{\mathcal{E}_3}$, with $\mathbf{R}_{\mathcal{E}_i}^f \triangleq \mathbf{R}_{\mathcal{I}}^f \mathbf{R}_{\mathcal{L}}^{\mathcal{L}} \mathbf{R}_{\mathcal{B}}^{\mathcal{B}} \mathbf{R}_{\mathcal{E}_i}^{\mathcal{E}_i}$.

This work assumes that the thrust forces are applied to the centers of mass of the respective thrusters' groups, which correspond the origins of \mathcal{E}_2 and \mathcal{E}_3 . Making $\mathbf{p}_i^{\mathcal{E}_i} = \mathbf{o}_{\mathcal{E}_i}^{\mathcal{E}_i} = \mathbf{0}_{3 \times 1}$ in (6) to obtain $\mathcal{J}_{\mathbf{o}_{\mathcal{E}_2}}^f = \partial \dot{\mathbf{o}}_{\mathcal{E}_2}^f / \partial \dot{\mathbf{q}}$ and $\mathcal{J}_{\mathbf{o}_{\mathcal{E}_3}}^f = \partial \dot{\mathbf{o}}_{\mathcal{E}_3}^f / \partial \dot{\mathbf{q}}$ yields

$$\dot{\mathbf{o}}_{\mathcal{E}_i}^f = \left[\mathbb{1}_{3 \times 3} \left(\mathbf{R}_{\mathcal{L}}^f \mathbf{S}(\mathbf{d}_{\mathcal{B}}^{\mathcal{L}})^T + \mathbf{R}_{\mathcal{L}}^f \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{S}(\mathbf{d}_{\mathcal{E}_i}^{\mathcal{B}})^T (\mathbf{R}_{\mathcal{B}}^{\mathcal{L}})^T \right) \mathbf{W}_{\eta} \mathbf{R}_{\mathcal{L}}^f \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{S}(\mathbf{d}_{\mathcal{E}_i}^{\mathcal{B}})^T \mathbf{Q} \mathbf{0}_{3 \times 1} \mathbf{0}_{3 \times 1} \right] \dot{\mathbf{q}}. \quad (18)$$

Then, using (16) and (18) yields

$$\mathfrak{f}_{\mathbf{f}_R} = \left(\mathcal{J}_{\mathbf{o}_{\mathcal{E}_2}}^f \right)^T \mathbf{f}_R^f = \begin{bmatrix} \mathbf{R}_{\mathcal{L}}^f \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{R}_{\mathcal{E}_2}^{\mathcal{E}_2} \mathbf{a}_z \\ \mathbf{W}_{\eta}^T \mathbf{S}(\mathbf{d}_{\mathcal{B}}^{\mathcal{L}}) \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{R}_{\mathcal{E}_2}^{\mathcal{E}_2} \mathbf{a}_z + \mathbf{W}_{\eta}^T \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{S}(\mathbf{d}_{\mathcal{E}_2}^{\mathcal{B}}) \mathbf{R}_{\mathcal{E}_2}^{\mathcal{E}_2} \mathbf{a}_z \\ \mathbf{Q}^T \mathbf{S}(\mathbf{d}_{\mathcal{E}_2}^{\mathcal{B}}) \mathbf{R}_{\mathcal{E}_2}^{\mathcal{E}_2} \mathbf{a}_z \\ 0 \\ 0 \end{bmatrix} f_R, \quad (19)$$

$$\mathfrak{f}_{\mathbf{f}_L} = \left(\mathcal{J}_{\mathbf{o}_{\mathcal{E}_3}}^f \right)^T \mathbf{f}_L^f = \begin{bmatrix} \mathbf{R}_{\mathcal{L}}^f \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{R}_{\mathcal{E}_3}^{\mathcal{E}_3} \mathbf{a}_z \\ \mathbf{W}_{\eta}^T \mathbf{S}(\mathbf{d}_{\mathcal{B}}^{\mathcal{L}}) \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{R}_{\mathcal{E}_3}^{\mathcal{E}_3} \mathbf{a}_z + \mathbf{W}_{\eta}^T \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{S}(\mathbf{d}_{\mathcal{E}_3}^{\mathcal{B}}) \mathbf{R}_{\mathcal{E}_3}^{\mathcal{E}_3} \mathbf{a}_z \\ \mathbf{Q}^T \mathbf{S}(\mathbf{d}_{\mathcal{E}_3}^{\mathcal{B}}) \mathbf{R}_{\mathcal{E}_3}^{\mathcal{E}_3} \mathbf{a}_z \\ 0 \\ 0 \end{bmatrix} f_L. \quad (20)$$

The servomotors' torques are applied to the respective thrusters' bodies, and opposite torques due to reaction are also applied to the aircraft's main body. These pairs of torques are mapped to generalized forces through (17). From the addition of angular velocities [26], we have

$$\boldsymbol{\omega}_{\mathcal{F}}^f = \boldsymbol{\omega}_{\mathcal{I}}^f + \boldsymbol{\omega}_{\mathcal{L}}^f = \mathbf{R}_{\mathcal{L}}^f \boldsymbol{\omega}_{\mathcal{I}}^f + \mathbf{R}_{\mathcal{L}}^f \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{B}} = \left[\mathbf{0}_{3 \times 3} \ \mathbf{R}_{\mathcal{L}}^f \mathbf{W}_{\eta} \ \mathbf{R}_{\mathcal{L}}^f \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{Q} \ \mathbf{0}_{3 \times 1} \ \mathbf{0}_{3 \times 1} \right] \dot{\mathbf{q}}, \quad (21)$$

$$\boldsymbol{\omega}_{\mathcal{F}\mathcal{E}_2}^f = \boldsymbol{\omega}_{\mathcal{I}}^f + \boldsymbol{\omega}_{\mathcal{L}}^f + \boldsymbol{\omega}_{\mathcal{B}\mathcal{E}_2}^f = \left[\mathbf{0}_{3 \times 3} \ \mathbf{R}_{\mathcal{L}}^f \mathbf{W}_{\eta} \ \mathbf{R}_{\mathcal{L}}^f \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{Q} \ \mathbf{R}_{\mathcal{L}}^f \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{R}_{\mathcal{E}_2}^{\mathcal{E}_2} \mathbf{a}_y \ \mathbf{0}_{3 \times 1} \right] \dot{\mathbf{q}}, \quad (22)$$

$$\boldsymbol{\omega}_{\mathcal{F}\mathcal{E}_3}^f = \boldsymbol{\omega}_{\mathcal{I}}^f + \boldsymbol{\omega}_{\mathcal{L}}^f + \boldsymbol{\omega}_{\mathcal{B}\mathcal{E}_3}^f = \left[\mathbf{0}_{3 \times 3} \ \mathbf{R}_{\mathcal{L}}^f \mathbf{W}_{\eta} \ \mathbf{R}_{\mathcal{L}}^f \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{Q} \ \mathbf{0}_{3 \times 1} \ \mathbf{R}_{\mathcal{L}}^f \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{R}_{\mathcal{E}_3}^{\mathcal{E}_3} \mathbf{a}_y \right] \dot{\mathbf{q}}. \quad (23)$$

Comparing (21), (22), and (23) to $\boldsymbol{\omega}_{\mathcal{F}\mathcal{B}}^f = \mathcal{W}_{\mathcal{B}}^f \dot{\mathbf{q}}$, $\boldsymbol{\omega}_{\mathcal{F}\mathcal{E}_2}^f = \mathcal{W}_{\mathcal{E}_2}^f \dot{\mathbf{q}}$, and $\boldsymbol{\omega}_{\mathcal{F}\mathcal{E}_3}^f = \mathcal{W}_{\mathcal{E}_3}^f \dot{\mathbf{q}}$, respectively, and using (17) lead to

$$\mathfrak{f}_{\boldsymbol{\tau}_{\alpha_R}} = (\mathcal{W}_{\mathcal{E}_2}^f)^T \boldsymbol{\tau}_{\alpha_R}^f + (\mathcal{W}_{\mathcal{B}}^f)^T (-\boldsymbol{\tau}_{\alpha_R}^f) \quad (24)$$

$$= \left[(\mathbf{0}_{3 \times 1})^T \ (\mathbf{0}_{3 \times 1})^T \ (\mathbf{0}_{2 \times 1})^T \ 1 \ 0 \right]^T \boldsymbol{\tau}_{\alpha_R},$$

$$\mathfrak{f}_{\boldsymbol{\tau}_{\alpha_L}} = (\mathcal{W}_{\mathcal{E}_3}^f)^T \boldsymbol{\tau}_{\alpha_L}^f + (\mathcal{W}_{\mathcal{B}}^f)^T (-\boldsymbol{\tau}_{\alpha_L}^f) \quad (25)$$

$$= \left[(\mathbf{0}_{3 \times 1})^T \ (\mathbf{0}_{3 \times 1})^T \ (\mathbf{0}_{2 \times 1})^T \ 0 \ 1 \right]^T \boldsymbol{\tau}_{\alpha_L}.$$

This work also takes into account drag torques generated by the propellers. These are reaction torques applied to the thrusters' bodies, due to the blades' acceleration and drag [30]. Assuming steady-state for the angular velocity of the blades, the drag torques are given in the thrusters' reference frames by $\boldsymbol{\tau}_{\text{drag,R}}^{\mathcal{E}_2} = \lambda_R (k_\tau / b) \mathbf{f}_R^{\mathcal{E}_2}$, $\boldsymbol{\tau}_{\text{drag,L}}^{\mathcal{E}_3} = \lambda_L (k_\tau / b) \mathbf{f}_L^{\mathcal{E}_3}$, where k_τ and b are parameters obtained experimentally, and λ_R and λ_L are given according to the direction of rotation of the corresponding propeller: if counter-clockwise, 1; if clockwise, -1. In the inertial reference frame, we then have $\boldsymbol{\tau}_{\text{drag,R}}^f = \mathbf{R}_{\mathcal{E}_2}^f \boldsymbol{\tau}_{\text{drag,R}}^{\mathcal{E}_2}$ and $\boldsymbol{\tau}_{\text{drag,L}}^f = \mathbf{R}_{\mathcal{E}_3}^f \boldsymbol{\tau}_{\text{drag,L}}^{\mathcal{E}_3}$.

As the drag torques are applied to the thrusters' bodies, through (17), (22), and (23), we have

$$\mathfrak{f}_{\boldsymbol{\tau}_{\text{drag,R}}} = (\mathcal{W}_{\mathcal{E}_2}^f)^T \boldsymbol{\tau}_{\text{drag,R}}^f = \lambda_R \frac{k_\tau}{b} \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{W}_{\eta}^T \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{R}_{\mathcal{E}_2}^{\mathcal{E}_2} \mathbf{a}_z \\ \mathbf{Q}^T \mathbf{R}_{\mathcal{E}_2}^{\mathcal{E}_2} \mathbf{a}_z \\ 0 \\ 0 \end{bmatrix} f_R, \quad (26)$$

$$\mathfrak{g}_{\tau_{\text{drag,L}}} = \left(\mathcal{W}_{\mathcal{E}_3} \right)^T \tau_{\text{drag,L}} = \lambda_L \frac{k_{\tau}}{b} \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{W}_{\eta}^T \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{R}_{\mathcal{E}_3}^{\mathcal{B}} \mathbf{a}_z \\ \mathbf{Q}^T \mathbf{R}_{\mathcal{E}_3}^{\mathcal{B}} \mathbf{a}_z \\ 0 \\ 0 \end{bmatrix} f_L. \quad (27)$$

Finally, the total mapping of the system inputs to generalized forces is obtained by summing up the contributions of the thrust forces, servomotor torques, and drag torques. Then, from (19)-(20) and (24)-(27),

$$\begin{aligned} \mathfrak{g}_{\text{in}} &= \mathfrak{g}_{f_R} + \mathfrak{g}_{f_L} + \mathfrak{g}_{\tau_{\alpha_R}} + \mathfrak{g}_{\tau_{\alpha_L}} + \mathfrak{g}_{\tau_{\text{drag,R}}} + \mathfrak{g}_{\tau_{\text{drag,L}}} \\ &= \begin{bmatrix} \mathbf{R}_{\mathcal{L}}^{\mathcal{J}} \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{r}_R & \mathbf{R}_{\mathcal{L}}^{\mathcal{J}} \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{r}_L & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{W}_{\eta}^T \mathbf{\Lambda}_R \mathbf{r}_R & \mathbf{W}_{\eta}^T \mathbf{\Lambda}_L \mathbf{r}_L & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{Q}^T \mathbf{\Gamma}_R \mathbf{r}_R & \mathbf{Q}^T \mathbf{\Gamma}_L \mathbf{r}_L & \mathbf{0}_{2 \times 1} & \mathbf{0}_{2 \times 1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_R \\ f_L \\ \tau_{\alpha_R} \\ \tau_{\alpha_L} \end{bmatrix} \\ &\triangleq \mathbf{L}_{\text{in}}(\mathbf{q}) \mathbf{u}, \end{aligned} \quad (28)$$

where

$$\begin{aligned} \mathbf{r}_R &\triangleq \mathbf{R}_{\mathcal{E}_2}^{\mathcal{B}} \mathbf{a}_z, \\ \mathbf{\Lambda}_R &\triangleq \mathbf{S}(\mathbf{d}_{\mathcal{E}_2}^{\mathcal{L}}) \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} + \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{S}(\mathbf{d}_{\mathcal{E}_2}^{\mathcal{B}}) + \lambda_R \frac{k_{\tau}}{b} \mathbf{R}_{\mathcal{B}}^{\mathcal{L}}, \\ \mathbf{\Gamma}_R &\triangleq \mathbf{S}(\mathbf{d}_{\mathcal{E}_2}^{\mathcal{B}}) + \lambda_R \frac{k_{\tau}}{b} \mathbb{1}_{3 \times 3}, \\ \mathbf{r}_L &\triangleq \mathbf{R}_{\mathcal{E}_3}^{\mathcal{B}} \mathbf{a}_z, \\ \mathbf{\Lambda}_L &\triangleq \mathbf{S}(\mathbf{d}_{\mathcal{E}_3}^{\mathcal{L}}) \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} + \mathbf{R}_{\mathcal{B}}^{\mathcal{L}} \mathbf{S}(\mathbf{d}_{\mathcal{E}_3}^{\mathcal{B}}) + \lambda_L \frac{k_{\tau}}{b} \mathbf{R}_{\mathcal{B}}^{\mathcal{L}}, \\ \mathbf{\Gamma}_L &\triangleq \mathbf{S}(\mathbf{d}_{\mathcal{E}_3}^{\mathcal{B}}) + \lambda_L \frac{k_{\tau}}{b} \mathbb{1}_{3 \times 3}. \end{aligned} \quad (29)$$

Note that, although in this current work no aerodynamic surfaces are considered in the tilt-rotor UAV (see Figure 1), the system modeling developed here is general enough to describe the dynamics of any tilt-rotor carrying a suspended load. The only constraint is that the aircraft needs to be seen as a multibody system with similar frames definitions as those shown in Figure 2. Further, in presence of aerodynamics surfaces (e.g., wings and horizontal and vertical stabilizers), the lift and drag forces generated by them can be added to the model in a straightforward manner, by including these terms in (28), which will allow dealing with both helicopter and airplane flight modes.

In addition, viscous friction is taken into account at the revolute joints of the tilting mechanisms and also at the point of connection between the rope and the tilt-rotor UAV. It is assumed that the friction torques are mapped to generalized forces as

$$\mathfrak{g}_{f_r} = -\mathbf{L}_{f_r} \dot{\mathbf{q}}, \quad (30)$$

where $\mathbf{L}_{f_r} \triangleq \text{diag}(0, 0, 0, 0, 0, 0, \mu_{\gamma}, \mu_{\gamma}, \mu_{\alpha}, \mu_{\alpha})$ with μ_{γ} and μ_{α} being constant parameters.

External disturbances applied to the suspended load are also considered, which may represent wind gusts affecting the system. Defining these disturbances in the inertial reference frame as the force vector $\mathbf{d} \triangleq [d_x \ d_y \ d_z]^T \in \mathbb{R}^3$, and assuming that is applied to the load's center of mass, it can be mapped to generalized forces through (16), yielding

$$\begin{aligned} \mathfrak{g}_{\text{db}} &= \left(\frac{\partial \dot{\mathbf{o}}_{\mathcal{L}}^{\mathcal{J}}}{\partial \dot{\mathbf{q}}} \right)^T \mathbf{d} = [\mathbb{1}_{3 \times 3} \ \mathbf{0}_{3 \times 3} \ \mathbf{0}_{3 \times 2} \ \mathbf{0}_{3 \times 1} \ \mathbf{0}_{3 \times 1}]^T \mathbf{d} \\ &\triangleq \mathbf{L}_{\text{db}} \mathbf{d}, \end{aligned} \quad (31)$$

where $\dot{\mathbf{o}}_{\mathcal{L}}^{\mathcal{J}} = \dot{\boldsymbol{\xi}}$ is obtained by making $\mathbf{p}_{\mathcal{L}}^{\mathcal{J}} = \mathbf{0}_{3 \times 1}$ in (7).

2.5. Equations of Motion. From (12), (14), and (15), the equations of motion of the tilt-rotor UAV with suspended load can be written in the Euler-Lagrange formulation as [27]

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathfrak{g}, \quad (32)$$

where \mathfrak{g} is the total generalized forces vector, obtained by summing up the contributions from the input forces and torques (28), viscous friction torques (30), and external disturbances (31). Thus, substituting $\mathfrak{g} = \mathfrak{g}_{\text{in}} + \mathfrak{g}_{f_r} + \mathfrak{g}_{\text{db}}$ in (32) yields

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + (\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{L}_{f_r}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{L}_{\text{in}}(\mathbf{q}) \mathbf{u} + \mathbf{L}_{\text{db}} \mathbf{d}. \quad (33)$$

Finally, by defining the state vector

$$\mathbf{x} \triangleq [\mathbf{q}^T \ \dot{\mathbf{q}}^T]^T \in \mathbb{R}^{20}, \quad (34)$$

and recalling the input vector defined in (28), the dynamic equations (33) can be written in the state-space representation

$$\begin{aligned} \dot{\mathbf{x}} &= \boldsymbol{\varphi}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \\ &= \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{M}(\mathbf{q})^{-1} [-(\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{L}_{f_r}) \dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) + \mathbf{L}_{\text{in}}(\mathbf{q}) \mathbf{u} + \mathbf{L}_{\text{db}} \mathbf{d}] \end{bmatrix}, \end{aligned} \quad (35)$$

which is nonlinear and highly coupled. Since the load's position and orientation are among the generalized coordinates (9), they are represented by the state variables (34), and, consequently, the load's behavior is described explicitly by (35). On the other hand, the aircraft's position and orientation are described only with respect to the load, thus appearing in (35) only implicitly.

3. Nonlinear State Estimation

This section presents a state estimation strategy to cope with the problem of predicting the states related to the load's pose, using the information of noisy sensors with different sampling rates, in order to gather the information necessary to build the state vector (34). The following sensors are assumed to be available: (i) a Global Positioning System (GPS) equipment to measure the x and y positions of the UAV; (ii) a barometer to measure the UAV's altitude;

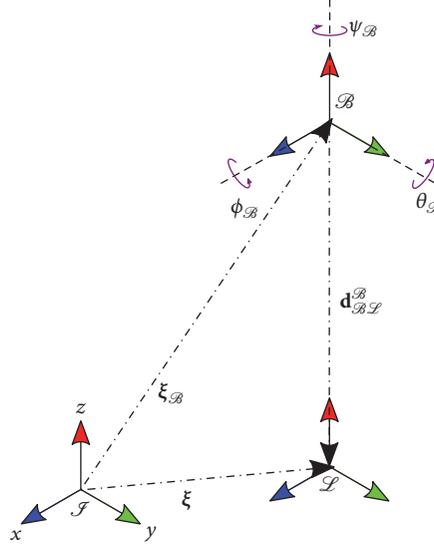


FIGURE 3: Tilt-rotor UAV pose and load's position measured by the available sensors.

(iii) an Inertial Measurement Unit (IMU) that provides the orientation and angular velocity of the UAV; (iv) a camera that gives the position of the load with respect to the UAV; and (v) embedded sensors at the servomotors to provide the tilting angles and their time derivatives.

The measurement model is highly nonlinear and therefore some well-known estimation techniques based on linearized transformations may have limited performance. Thereby, this paper considers the unscented Kalman filter strategy that estimates the mean and covariance of the likelihood distribution by means of the Unscented Transform (UT) and then uses the Kalman Filter (KF) equations to compute the posterior distribution [31, 32].

3.1. Measurement Equation. The relation between the variables measured by the embedded sensors and the state vector \mathbf{x} at time instant k is given by the measurement equation

$$\mathbf{y}_k = \boldsymbol{\pi}(\mathbf{x}_k) + \mathbf{v}_k, \quad (36)$$

where \mathbf{y}_k is the measured vector, $\boldsymbol{\pi}(\mathbf{x}_k)$ corresponds to a nonlinear mapping to be obtained, and \mathbf{v}_k is the measurement noise.

Let $\boldsymbol{\xi}_B \triangleq [x_B \ y_B \ z_B]^T$ be the position of the aircraft with respect to \mathcal{F} (see Figure 3). Then, through forward kinematics, the following holds:

$$\boldsymbol{\xi}_B(\boldsymbol{\xi}, \boldsymbol{\eta}) = \boldsymbol{\xi} + \mathbf{R}_{\mathcal{F}}^B \mathbf{d}_{\mathcal{F}}^B. \quad (37)$$

Additionally, let $\boldsymbol{\eta}_B \triangleq [\phi_B \ \theta_B \ \psi_B]^T$ be the aircraft attitude with respect to \mathcal{F} parametrized by Euler angles using

the local roll-pitch-yaw convention. Therefore,

$$\begin{aligned} \mathbf{R}_{\mathcal{F}}^B &\triangleq \mathbf{R}_{z, \psi_B} \mathbf{R}_{y, \theta_B} \mathbf{R}_{x, \phi_B} \\ &= \begin{bmatrix} c_{\psi_B} c_{\theta_B} & c_{\psi_B} s_{\theta_B} s_{\phi_B} - s_{\psi_B} c_{\phi_B} & c_{\psi_B} s_{\theta_B} c_{\phi_B} + s_{\psi_B} s_{\phi_B} \\ s_{\psi_B} c_{\theta_B} & s_{\psi_B} s_{\theta_B} s_{\phi_B} + c_{\psi_B} c_{\phi_B} & s_{\psi_B} s_{\theta_B} c_{\phi_B} - c_{\psi_B} s_{\phi_B} \\ -s_{\theta_B} & c_{\theta_B} s_{\phi_B} & c_{\theta_B} c_{\phi_B} \end{bmatrix}. \end{aligned} \quad (38)$$

Recalling that the relation $\mathbf{R}_{\mathcal{F}}^B = \mathbf{R}_{\mathcal{F}}^{\mathcal{L}} \mathbf{R}_{\mathcal{L}}^B$ holds and considering $\theta_B \neq \pm\pi/2$, it is possible to state that

$$\begin{aligned} \phi_B(\boldsymbol{\eta}, \boldsymbol{\gamma}) &= \arctan\left(\frac{(\mathbf{R}_{\mathcal{F}}^{\mathcal{L}} \mathbf{R}_{\mathcal{L}}^B)_{32}}{(\mathbf{R}_{\mathcal{F}}^{\mathcal{L}} \mathbf{R}_{\mathcal{L}}^B)_{33}}\right), \\ \theta_B(\boldsymbol{\eta}, \boldsymbol{\gamma}) &= \arcsin\left(-(\mathbf{R}_{\mathcal{F}}^{\mathcal{L}} \mathbf{R}_{\mathcal{L}}^B)_{31}\right), \\ \psi_B(\boldsymbol{\eta}, \boldsymbol{\gamma}) &= \arctan\left(\frac{(\mathbf{R}_{\mathcal{F}}^{\mathcal{L}} \mathbf{R}_{\mathcal{L}}^B)_{21}}{(\mathbf{R}_{\mathcal{F}}^{\mathcal{L}} \mathbf{R}_{\mathcal{L}}^B)_{11}}\right). \end{aligned} \quad (39)$$

Furthermore, the angular velocity provided by the IMU is given by

$$\begin{aligned} \boldsymbol{\omega}_{\mathcal{F}}^B(\boldsymbol{\eta}, \boldsymbol{\gamma}, \dot{\boldsymbol{\eta}}, \dot{\boldsymbol{\gamma}}) &= \boldsymbol{\omega}_{\mathcal{F}}^B + \boldsymbol{\omega}_{\mathcal{L}}^B \\ &= (\mathbf{R}_{\mathcal{F}}^B)^T \mathbf{W}_{\boldsymbol{\eta}} \dot{\boldsymbol{\eta}} + \mathbf{Q} \dot{\boldsymbol{\gamma}}, \end{aligned} \quad (40)$$

where the matrices $\mathbf{W}_{\boldsymbol{\eta}}$ and \mathbf{Q} were defined in (4).

Let $\mathbf{d}_{\mathcal{F}}^B$ denote the displacement vector from \mathcal{B} to \mathcal{L} , expressed in \mathcal{B} , which is the measurement provided by the camera (see Figure 3). Then,

$$\mathbf{d}_{\mathcal{F}}^B(\boldsymbol{\gamma}) = -\mathbf{d}_{\mathcal{L}}^B = -(\mathbf{R}_{\mathcal{F}}^{\mathcal{L}})^T \mathbf{d}_{\mathcal{L}}^{\mathcal{L}} \triangleq -(\mathbf{R}_{\mathcal{F}}^B)^T \mathbf{d}_{\mathcal{F}}^B. \quad (41)$$

Finally, (37), (39), (40), and (41) along with the system's states $\alpha_R, \alpha_L, \dot{\alpha}_R, \dot{\alpha}_L$, and the measurement noise \mathbf{v}_k can be grouped into the nonlinear measurement equation

$$\mathbf{y}_k = \boldsymbol{\pi}(\mathbf{x}_k) + \mathbf{v}_k \triangleq \begin{bmatrix} \xi_{\mathcal{B}}(\boldsymbol{\xi}, \boldsymbol{\eta}) \\ \phi_{\mathcal{B}}(\boldsymbol{\eta}, \boldsymbol{\gamma}) \\ \theta_{\mathcal{B}}(\boldsymbol{\eta}, \boldsymbol{\gamma}) \\ \psi_{\mathcal{B}}(\boldsymbol{\eta}, \boldsymbol{\gamma}) \\ \omega_{\mathcal{F}\mathcal{B}}^{\mathcal{L}}(\boldsymbol{\eta}, \boldsymbol{\gamma}, \dot{\boldsymbol{\eta}}, \dot{\boldsymbol{\gamma}}) \\ \mathbf{d}_{\mathcal{B}\mathcal{L}}^{\mathcal{L}}(\boldsymbol{\gamma}) \\ \alpha_R \\ \alpha_L \\ \dot{\alpha}_R \\ \dot{\alpha}_L \end{bmatrix} + \mathbf{v}_k \quad (42)$$

$$= \begin{bmatrix} \boldsymbol{\xi} + \mathbf{R}_{\mathcal{F}\mathcal{B}}^{\mathcal{L}} \mathbf{d}_{\mathcal{B}\mathcal{L}}^{\mathcal{L}} \\ \arctan\left(\frac{(\mathbf{R}_{\mathcal{F}\mathcal{B}}^{\mathcal{L}} \mathbf{R}_{\mathcal{B}}^{\mathcal{L}})_{32}}{(\mathbf{R}_{\mathcal{F}\mathcal{B}}^{\mathcal{L}} \mathbf{R}_{\mathcal{B}}^{\mathcal{L}})_{33}}\right) \\ \arcsin\left(-(\mathbf{R}_{\mathcal{F}\mathcal{B}}^{\mathcal{L}} \mathbf{R}_{\mathcal{B}}^{\mathcal{L}})_{31}\right) \\ \arctan\left(\frac{(\mathbf{R}_{\mathcal{F}\mathcal{B}}^{\mathcal{L}} \mathbf{R}_{\mathcal{B}}^{\mathcal{L}})_{21}}{(\mathbf{R}_{\mathcal{F}\mathcal{B}}^{\mathcal{L}} \mathbf{R}_{\mathcal{B}}^{\mathcal{L}})_{11}}\right) \\ (\mathbf{R}_{\mathcal{B}}^{\mathcal{L}})^T \mathbf{W}_{\boldsymbol{\eta}} \dot{\boldsymbol{\eta}} + \mathbf{Q}\dot{\boldsymbol{\gamma}} \\ - (\mathbf{R}_{\mathcal{B}}^{\mathcal{L}})^T \mathbf{d}_{\mathcal{B}\mathcal{L}}^{\mathcal{L}} \\ \alpha_R \\ \alpha_L \\ \dot{\alpha}_R \\ \dot{\alpha}_L \end{bmatrix} + \mathbf{v}_k.$$

This work assumes synchronized sensors with sampling rates equal to $10T_s, 2T_s$, and T_s , respectively, for the GPS, camera, and remaining sensors, T_s being the control sampling time. Since not all information is available at time instant k , the dimension of the vector \mathbf{y}_k , as well as the transformation $\boldsymbol{\pi}(\cdot)$, will change with time. Thus, from now on, the variables that have different data acquisition rates will be denoted with the subscript $(\cdot)_{\{T_s\}}$.

3.2. Unscented Kalman Filter. In the unscented Kalman filter (UKF), a fixed number of sigma points are chosen deterministically to capture the mean and covariance of the prior distribution. These points are then propagated through a nonlinear transformation to estimate the posterior distribution [32].

Consider the discrete representation of (35) and the measurement equation (42)

$$\mathbf{x}_k = \boldsymbol{\varphi}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{d}_{k-1}) + \mathbf{w}_{k-1}, \quad (43)$$

$$\mathbf{y}_{k,\{T_s\}} = \boldsymbol{\pi}_{\{T_s\}}(\mathbf{x}_k) + \mathbf{v}_{k,\{T_s\}}, \quad (44)$$

where \mathbf{w}_k is the process noise. Also, let $\widehat{(\cdot)}$ denote estimated variables, $(\cdot)_{m|n}$ denote information at time instant m given measurements up to instant n , and $E[\cdot]$ be the expected value operator. This work assumes that measures of \mathbf{d}_k are always available, and the mean $\widehat{\mathbf{x}}_{0|0} = E[\mathbf{x}_0]$ and covariance $\mathbf{P}_{0|0}^x = E[(\mathbf{x}_0 - \widehat{\mathbf{x}}_{0|0})(\mathbf{x}_0 - \widehat{\mathbf{x}}_{0|0})^T]$ are known. Furthermore, the process and measurement noises are assumed to have zero mean and covariances $E[\mathbf{w}_k \mathbf{w}_k^T] = \mathbf{Q}_k$ and $E[\mathbf{v}_{k,\{T_s\}} \mathbf{v}_{k,\{T_s\}}^T] = \mathbf{R}_{k,\{T_s\}}$, respectively, and the cross-covariance between them is assumed to be null, that is, $E[\mathbf{w}_i \mathbf{v}_{j,\{T_s\}}^T] = \mathbf{0} \forall i, j$.

In order to describe the prior statistics using UT, only $n_{\text{sp}} = 2n$ sigma points are necessary, where n is the dimension of the state vector \mathbf{x} [31]. The sigma points $\mathbf{X}_1, \dots, \mathbf{X}_{n_{\text{sp}}}$ must satisfy

$$\widehat{\mathbf{x}} = \sum_{j=1}^{n_{\text{sp}}} \gamma_j \mathbf{X}_j, \quad (45)$$

$$\mathbf{P}^x = \sum_{j=1}^{n_{\text{sp}}} \gamma_j [\mathbf{X}_j - \widehat{\mathbf{x}}] [\mathbf{X}_j - \widehat{\mathbf{x}}]^T,$$

where γ_j are weights defined as $\gamma_j = 1/2n$ subject to $\sum_{j=1}^{n_{\text{sp}}} \gamma_j = 1$.

The sigma point matrix $\mathbf{X} \triangleq [\mathbf{X}_1 \ \mathbf{X}_2 \ \dots \ \mathbf{X}_{n_{\text{sp}}}]$ is chosen as

$$\mathbf{X} = \widehat{\mathbf{x}} \mathbb{1}_{1 \times n_{\text{sp}}} + \sqrt{n} \left[(\mathbf{P}^x)^{1/2} - (\mathbf{P}^x)^{1/2} \right], \quad (46)$$

where $(\cdot)^{1/2}$ is the Cholesky square root.

The likelihood distribution statistics can be obtained by propagating the sigma points $\mathbf{X}_1, \dots, \mathbf{X}_{n_{\text{sp}}}$ through the nonlinear measurement equation (42), yielding

$$\mathbf{Y}_{j,\{T_s\}} = \boldsymbol{\pi}_{\{T_s\}}(\mathbf{X}_j), \quad j = 1, \dots, n_{\text{sp}} \quad (47)$$

such that

$$\widehat{\mathbf{y}}_{\{T_s\}} = \sum_{j=1}^{n_{\text{sp}}} \gamma_j \mathbf{Y}_{j,\{T_s\}}, \quad (48)$$

$$\mathbf{P}_{\{T_s\}}^y = \sum_{j=1}^{n_{\text{sp}}} \gamma_j \left[\mathbf{Y}_{j,\{T_s\}} - \widehat{\mathbf{y}}_{\{T_s\}} \right] \left[\mathbf{Y}_{j,\{T_s\}} - \widehat{\mathbf{y}}_{\{T_s\}} \right]^T.$$

Algorithm. The UKF algorithm is formed by a forecast and a data assimilation step [32]. The operations presented below must be performed at each measurement step $k = 1, 2, 3, \dots$

(i) *Forecast Step*

(1) Form the sigma points using (46),

$$\begin{aligned} \mathbf{X}_{k-1|k-1} &= \widehat{\mathbf{x}}_{k-1|k-1} \mathbb{1}_{1 \times n_{\text{sp}}} \\ &+ \sqrt{n} \left[(\mathbf{P}_{k-1|k-1}^x)^{1/2} - (\mathbf{P}_{k-1|k-1}^x)^{1/2} \right]. \end{aligned} \quad (49)$$

(2) Propagate the sigma points through the nonlinear transformation (43)

$$\mathbf{X}_{j,k|k-1} = \boldsymbol{\varphi}(\mathbf{X}_{j,k-1|k-1}, \mathbf{u}_{k-1}, \mathbf{d}_{k-1}), \quad \forall j = 1, \dots, n_{\text{sp}}. \quad (50)$$

(3) Compute the predicted mean and covariance using (45) together with the process noise covariance

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \sum_{j=1}^{n_{\text{sp}}} \gamma_j \mathbf{X}_{j,k|k-1}, \\ \mathbf{P}_{k|k-1}^x &= \sum_{j=1}^{n_{\text{sp}}} \gamma_j [\mathbf{X}_{j,k|k-1} - \hat{\mathbf{x}}_{k|k-1}] [\mathbf{X}_{j,k|k-1} - \hat{\mathbf{x}}_{k|k-1}]^T \\ &\quad + \mathbf{Q}_{k-1}. \end{aligned} \quad (51)$$

(4) Form the sigma point matrix with the estimation of $\hat{\mathbf{x}}_{k|k-1}$

$$\begin{aligned} \mathbf{X}_{k|k-1} &= \hat{\mathbf{x}}_{k|k-1} \mathbb{1}_{1 \times n_{\text{sp}}} \\ &\quad + \sqrt{n} \left[(\mathbf{P}_{k|k-1}^x)^{1/2} - (\mathbf{P}_{k|k-1}^x)^{1/2} \right]. \end{aligned} \quad (52)$$

(5) Propagate the sigma points $\mathbf{X}_{j,k|k-1}$ through the measurement model (44)

$$\mathbf{Y}_{j,k|k-1, \{T_s\}} = \boldsymbol{\pi}_{\{T_s\}}(\mathbf{X}_{j,k|k-1}), \quad \forall j = 1, \dots, n_{\text{sp}}. \quad (53)$$

(6) Compute the measurement statistics using (48) together with the measurement noise covariance and the cross-covariance of the states and measurements

$$\begin{aligned} \hat{\mathbf{y}}_{k|k-1, \{T_s\}} &= \sum_{j=1}^{n_{\text{sp}}} \gamma_j \mathbf{Y}_{j,k|k-1, \{T_s\}}, \\ \mathbf{P}_{k|k-1, \{T_s\}}^y &= \sum_{j=1}^{n_{\text{sp}}} \gamma_j [\mathbf{Y}_{j,k|k-1, \{T_s\}} - \hat{\mathbf{y}}_{k|k-1, \{T_s\}}] [\mathbf{Y}_{j,k|k-1, \{T_s\}} - \hat{\mathbf{y}}_{k|k-1, \{T_s\}}]^T + \mathbf{R}_{k, \{T_s\}}, \\ \mathbf{P}_{k|k-1, \{T_s\}}^{xy} &= \sum_{j=1}^{n_{\text{sp}}} \gamma_j [\mathbf{X}_{j,k|k-1} - \hat{\mathbf{x}}_{k|k-1}] [\mathbf{Y}_{j,k|k-1, \{T_s\}} - \hat{\mathbf{y}}_{k|k-1, \{T_s\}}]^T. \end{aligned} \quad (54)$$

(ii) Data Assimilation Step

(1) Compute the filter gain $\mathbf{K}_{k, \{T_s\}}$ and the innovations $\mathbf{v}_{k, \{T_s\}}$

$$\begin{aligned} \mathbf{K}_{k, \{T_s\}} &= \mathbf{P}_{k|k-1, \{T_s\}}^{xy} \left(\mathbf{P}_{k|k-1, \{T_s\}}^y \right)^{-1}, \\ \mathbf{v}_{k, \{T_s\}} &= \mathbf{y}_{k, \{T_s\}} - \hat{\mathbf{y}}_{k|k-1, \{T_s\}}. \end{aligned} \quad (55)$$

(2) Compute the corrected mean and covariance conditional on the measurement information

$$\begin{aligned} \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_{k, \{T_s\}} \mathbf{v}_{k, \{T_s\}}, \\ \mathbf{P}_{k|k}^x &= \mathbf{P}_{k|k-1}^x - \mathbf{K}_{k, \{T_s\}} \mathbf{P}_{k|k-1, \{T_s\}}^y \mathbf{K}_{k, \{T_s\}}^T. \end{aligned} \quad (56)$$

Observe that the UKF algorithm uses at all time instant k all available information from the sensors to estimate the posterior distribution. For those time instants in which the measurement vector is full, that is, it has all sensors information, the estimation of $\hat{\mathbf{x}}_{k|k}$ will be more accurate and the covariance $\mathbf{P}_{k|k}^x$ reduced. On the other hand, when neither GPS nor camera information is available, the data assimilation step will have a less important role on the estimation algorithm, and $\hat{\mathbf{x}}_{k|k}$ will be less accurate and $\mathbf{P}_{k|k}^x$ bigger.

Finally, the state vector estimated in this section, $\hat{\mathbf{x}}_{k|k}$, will be used by the control strategy presented in the next section to perform the state feedback control.

4. Suspended Load Path Tracking Controller

This section describes the model predictive controller design for performing the suspended load path tracking. The main objectives of the control system are as follows: ensure closed-loop stability, reject constant external disturbances and parametric uncertainties, and satisfy constraints on state deviations and control inputs.

Aiming at an improved path tracking control and recalling the system's underactuated behavior, the load's position $\xi = [x \ y \ z]^T$ and its yaw angle ψ are chosen to be regulated, while the other degrees of freedom will be only stabilized.

This paper works with the incremental MPC framework [33], for which the prediction model is obtained using a discrete linear time-varying state-space model that is not affine in the parameters. On one hand, the use of this kind of linear system improves the aircraft's nonlinear dynamics representation, when comparing to time-invariant linearized models, and, particularly for the proposed application, allows yaw movements regulation and copes with the rope's length variation during the take-off and landing. On the other hand, because the time-varying linearized model used in this work

is non-affine in the time-varying parameters, it is not possible to obtain, for instance, a polytopic representation for the linearized model and, then, use some well-know techniques to improve the MPC robustness.

4.1. Linearized Error Dynamics. Seeking to obtain the discrete linear time-varying state-space model to construct the prediction model, the equations of motion (35) must be linearized around a time-varying trajectory. Additionally, due to limited computational resources, this process needs to be done with most of the physical parameters numerically evaluated. However, it is possible to let some physical parameters be variables in a way that they will appear in the linearized Jacobians after finishing the linearization process.

Let \mathbf{x}^{tr} and \mathbf{u}^{tr} denote trajectory values and, for linearization purposes, consider $\mathbf{d} = \mathbf{0}_{3 \times 1}$ (see (31)). This work assumes that the desired trajectory is feasible, that is,

$$\dot{\mathbf{x}}^{\text{tr}} = \boldsymbol{\varphi}(\mathbf{x}^{\text{tr}}, \mathbf{u}^{\text{tr}}, \mathbf{d}). \quad (57)$$

Then, linearizing the state-space equations (35) around these trajectories, through first-order expansion in Taylor series, yields

$$\Delta \dot{\mathbf{x}} = \mathbf{A}(t) \Delta \mathbf{x} + \mathbf{B}(t) \Delta \mathbf{u}, \quad (58)$$

where $\Delta \mathbf{x} \triangleq \mathbf{x} - \mathbf{x}^{\text{tr}}$, $\Delta \mathbf{u} \triangleq \mathbf{u} - \mathbf{u}^{\text{tr}}$, and

$$\begin{aligned} \mathbf{A}(t) &= \left. \frac{\partial \boldsymbol{\varphi}(\mathbf{x}, \mathbf{u}, \mathbf{d})}{\partial \mathbf{x}} \right|_{\substack{\mathbf{x}=\mathbf{x}^{\text{tr}} \\ \mathbf{u}=\mathbf{u}^{\text{tr}}}} \in \mathbb{R}^{20 \times 20}, \\ \mathbf{B}(t) &= \left. \frac{\partial \boldsymbol{\varphi}(\mathbf{x}, \mathbf{u}, \mathbf{d})}{\partial \mathbf{u}} \right|_{\substack{\mathbf{x}=\mathbf{x}^{\text{tr}} \\ \mathbf{u}=\mathbf{u}^{\text{tr}}}} \in \mathbb{R}^{20 \times 4}. \end{aligned} \quad (59)$$

In this work, the trajectory values for \mathbf{x} and \mathbf{u} are given by

$$\mathbf{x}^{\text{tr}} = [(\mathbf{q}^{\text{tr}})^T \ (\dot{\mathbf{q}}^{\text{tr}})^T]^T, \quad (60)$$

$$\mathbf{u}^{\text{tr}} = \mathbf{L}_{\text{in}}(\mathbf{q}^{\text{tr}})^+ [\mathbf{M}(\mathbf{q}^{\text{tr}}) \ddot{\mathbf{q}}^{\text{tr}} + (\mathbf{C}(\mathbf{q}^{\text{tr}}, \dot{\mathbf{q}}^{\text{tr}}) + \mathbf{L}_{\text{fr}}) \dot{\mathbf{q}}^{\text{tr}} + \mathbf{g}(\mathbf{q}^{\text{tr}})], \quad (61)$$

where $\mathbf{L}_{\text{in}}(\mathbf{q}^{\text{tr}})^+$ denotes the left pseudoinverse of $\mathbf{L}_{\text{in}}(\mathbf{q}^{\text{tr}})$ and \mathbf{q}^{tr} , $\dot{\mathbf{q}}^{\text{tr}}$, and $\ddot{\mathbf{q}}^{\text{tr}}$ are provided reference signals with $\mathbf{q}^{\text{tr}} \triangleq [x^{\text{tr}}(t) \ y^{\text{tr}}(t) \ z^{\text{tr}}(t) \ \phi^{\text{eq}} \ \theta^{\text{eq}} \ \psi^{\text{tr}}(t) \ \gamma_1^{\text{eq}} \ \gamma_2^{\text{eq}} \ \alpha_R^{\text{eq}} \ \alpha_L^{\text{eq}}]^T$, in which $(\cdot)^{\text{eq}}$ is the state's equilibrium value. Notice that \mathbf{u}^{tr} , since it is computed using a left pseudoinverse, will be an exact solution to the dynamic equations (33) only if the desired trajectory is feasible.

Therefore, by linearizing the system using (59) with the trajectories defined in (60) and (61) added to the rope's length $l(t)$ as a time-varying parameter, the linearized Jacobians are $\mathbf{A}(\boldsymbol{\zeta}(t))$ and $\mathbf{B}(\boldsymbol{\zeta}(t))$, where $\boldsymbol{\zeta}(t) \triangleq [x^{\text{tr}}(t) \ y^{\text{tr}}(t) \ z^{\text{tr}}(t) \ \psi^{\text{tr}}(t) \ x^{\text{tr}}(t) \ y^{\text{tr}}(t) \ z^{\text{tr}}(t) \ \dot{\psi}^{\text{tr}}(t) \ \dot{x}^{\text{tr}}(t) \ \dot{y}^{\text{tr}}(t) \ \dot{z}^{\text{tr}}(t) \ \dot{\psi}^{\text{tr}}(t) \ l(t)]^T$ is the vector of time-varying parameters.

To improve the trajectory tracking of the regulated variables and provide constant disturbance and parametric uncertainties rejection, the state vector $\Delta \mathbf{x}$ is augmented with

integral actions [34], yielding

$$\Delta \bar{\mathbf{x}} \triangleq \begin{bmatrix} \Delta \mathbf{x} \\ \int (\boldsymbol{\xi} - \boldsymbol{\xi}^{\text{tr}}) \\ \int (\boldsymbol{\psi} - \boldsymbol{\psi}^{\text{tr}}) \end{bmatrix} \in \mathbb{R}^{24}, \quad (62)$$

whose dynamics are given by

$$\begin{aligned} \Delta \dot{\bar{\mathbf{x}}} &= \begin{bmatrix} \mathbf{A}(\boldsymbol{\zeta}(t)) & \mathbf{0}_{20 \times 4} \\ \mathbf{0}_{4 \times 14} & \mathbf{0}_{4 \times 4} \end{bmatrix} \Delta \bar{\mathbf{x}} \\ &+ \underbrace{\begin{bmatrix} \mathbf{B}(\boldsymbol{\zeta}(t)) \\ \mathbf{0}_{4 \times 4} \end{bmatrix}}_{\bar{\mathbf{B}}(\boldsymbol{\zeta}(t))} \Delta \mathbf{u}. \end{aligned} \quad (63)$$

To obtain a discrete prediction model using the incremental form and, thereafter, improve performance with input integrators added to the closed-loop system [33], it will be necessary to map the model (63) from the continuous-time to the discrete-time domain, which yields

$$\Delta \bar{\mathbf{x}}_{k+1} = \bar{\mathbf{A}}(\boldsymbol{\zeta}_k) \Delta \bar{\mathbf{x}}_k + \bar{\mathbf{B}}(\boldsymbol{\zeta}_k) \Delta \mathbf{u}_k, \quad (64)$$

being the matrices $\bar{\mathbf{A}}(\boldsymbol{\zeta}_k)$ and $\bar{\mathbf{B}}(\boldsymbol{\zeta}_k)$ obtained after discretizing the model using a zero-order hold with sampling time T_s (in this work, variables in continuous-time and discrete-time domain are differentiated by the time variable t and the sampling variable k).

Finally, choosing the control increment $\delta \mathbf{u}_k \triangleq \Delta \mathbf{u}_k - \Delta \mathbf{u}_{k-1}$ to be the control input, the extended discrete linearized system can be rewritten in the incremental form as

$$\underbrace{\begin{bmatrix} \Delta \bar{\mathbf{x}}_{k+1} \\ \Delta \mathbf{u}_k \end{bmatrix}}_{\Delta \mathcal{X}_{k+1}} = \underbrace{\begin{bmatrix} \bar{\mathbf{A}}(\boldsymbol{\zeta}_k) & \bar{\mathbf{B}}(\boldsymbol{\zeta}_k) \\ \mathbf{0}_{4 \times 24} & \mathbb{I}_{4 \times 4} \end{bmatrix}}_{\mathcal{A}(\boldsymbol{\zeta}_k)} \underbrace{\begin{bmatrix} \Delta \bar{\mathbf{x}}_k \\ \Delta \mathbf{u}_{k-1} \end{bmatrix}}_{\Delta \mathcal{X}_k} + \underbrace{\begin{bmatrix} \bar{\mathbf{B}}(\boldsymbol{\zeta}_k) \\ \mathbb{I}_{4 \times 4} \end{bmatrix}}_{\mathcal{B}(\boldsymbol{\zeta}_k)} \delta \mathbf{u}_k. \quad (65)$$

4.2. Prediction Model. The state-space model (65) gives the one-step ahead prediction and can be used recursively to obtain the prediction model considering a prediction horizon N_p and control horizon N_c . Thus, considering the case where $N_c < N_p$ and assuming $\delta \mathbf{u}_{k+i} = \mathbf{0}_{4 \times 1}$, $\forall i \geq N_c$, the N_p -step ahead prediction yields

$$\begin{aligned} \Delta \mathcal{X}_{k+N_p} &= \left(\prod_{\ell=1}^{N_p} \mathcal{A}(\boldsymbol{\zeta}_{k+N_p-\ell}) \right) \Delta \mathcal{X}_k \\ &+ \left(\prod_{\ell=1}^{N_p-1} \mathcal{A}(\boldsymbol{\zeta}_{k+N_p-\ell}) \right) \mathcal{B}(\boldsymbol{\zeta}_k) \delta \mathbf{u}_k + \dots \\ &+ \left(\prod_{\ell=1}^{N_p-N_c} \mathcal{A}(\boldsymbol{\zeta}_{k+N_p-\ell}) \right) \mathcal{B}(\boldsymbol{\zeta}_{k+N_c-1}) \delta \mathbf{u}_{k+N_c-1}. \end{aligned} \quad (66)$$

Thereby, defining $\Delta \underline{\mathcal{X}} \triangleq [(\Delta \mathcal{X}_{k+1})^T \cdots (\Delta \mathcal{X}_{k+N_p})^T]^T$ and $\delta \underline{\mathbf{u}} \triangleq [(\delta \mathbf{u}_k)^T \cdots (\delta \mathbf{u}_{k+N_c-1})^T]^T$, it is possible to write the prediction model as

$$\Delta \underline{\mathcal{X}} = \mathcal{P} \Delta \mathcal{X}_k + \mathcal{H} \delta \underline{\mathbf{u}}, \quad (67)$$

where the matrices $\mathcal{P} \in \mathbb{R}^{28 \cdot N_p \times 28}$ and $\mathcal{H} \in \mathbb{R}^{28 \cdot N_p \times 4 \cdot N_c}$ are given by

$$\mathcal{P} = \left[\left(\prod_{\ell=1}^1 \mathcal{A}(\zeta_{k+1-\ell}) \right)^T \left(\prod_{\ell=1}^2 \mathcal{A}(\zeta_{k+2-\ell}) \right)^T \cdots \left(\prod_{\ell=1}^{N_p} \mathcal{A}(\zeta_{k+N_p-\ell}) \right)^T \right]^T,$$

$$\mathcal{H} = \begin{bmatrix} \mathcal{B}(\zeta_k) & \mathbf{0}_{28 \times 4} & \cdots & \mathbf{0}_{28 \times 4} \\ \left(\prod_{\ell=1}^{2-1} \mathcal{A}(\zeta_{k+2-\ell}) \right) \mathcal{B}(\zeta_k) & \mathcal{B}(\zeta_{k+1}) & \cdots & \mathbf{0}_{28 \times 4} \\ \vdots & \vdots & \ddots & \vdots \\ \left(\prod_{\ell=1}^{N_p-1} \mathcal{A}(\zeta_{k+N_p-\ell}) \right) \mathcal{B}(\zeta_k) & \left(\prod_{\ell=1}^{N_p-2} \mathcal{A}(\zeta_{k+N_p-\ell}) \right) \mathcal{B}(\zeta_{k+1}) & \cdots & \left(\prod_{\ell=1}^{N_p-N_c} \mathcal{A}(\zeta_{k+N_p-\ell}) \right) \mathcal{B}(\zeta_{k+N_c-1}) \end{bmatrix}. \quad (68)$$

4.3. Optimization Problem. Consider the standard quadratic cost function

$$\mathcal{J} = \sum_{i=1}^{N_p} \|\mathcal{W}_{k+i} - \Delta \mathcal{X}_{k+i}\|_{\mathcal{Q}}^2 + \sum_{j=0}^{N_c-1} \|\delta \mathbf{u}_{k+j}\|_{\mathcal{R}}^2, \quad (69)$$

where $\mathcal{W}_{k+i} = (\mathcal{X}_{k+i}^{\text{tr}} - \mathcal{X}_k^{\text{tr}})$ for $i = 1, \dots, N_p$, $\mathcal{X}_{k+i}^{\text{tr}}$ being the reference trajectory at the instant $k+i$. Moreover, $\mathcal{Q} \in \mathbb{R}^{28 \times 28}$ and $\mathcal{R} \in \mathbb{R}^{4 \times 4}$ are, respectively, weighting matrices of states error and control effort.

The cost function (69) can be written in the matrix form by means of the prediction model (67) as

$$\mathcal{J} = \left(\mathcal{H} \delta \underline{\mathbf{u}} + \mathcal{P} \Delta \mathcal{X}_k - \underline{\mathcal{W}} \right)^T \cdot \Omega_{\mathcal{Q}} \left(\mathcal{H} \delta \underline{\mathbf{u}} + \mathcal{P} \Delta \mathcal{X}_k - \underline{\mathcal{W}} \right) + \delta \underline{\mathbf{u}}^T \Omega_{\mathcal{R}} \delta \underline{\mathbf{u}}, \quad (70)$$

where $\Omega_{\mathcal{Q}} \triangleq \text{blkdiag}(\mathcal{Q}, \dots, \mathcal{Q}) \in \mathbb{R}^{28 \cdot N_p \times 28 \cdot N_p}$ and $\Omega_{\mathcal{R}} \triangleq \text{blkdiag}(\mathcal{R}, \dots, \mathcal{R}) \in \mathbb{R}^{4 \cdot N_c \times 4 \cdot N_c}$ are block diagonal matrices, and $\underline{\mathcal{W}} \triangleq [(\mathcal{W}_{k+1})^T \cdots (\mathcal{W}_{k+N_p})^T]^T$.

Finally, (70) can be rewritten in the canonical quadratic form as

$$\mathcal{J} = \frac{1}{2} \delta \underline{\mathbf{u}}^T \Lambda \delta \underline{\mathbf{u}} + \mathbf{f}^T \delta \underline{\mathbf{u}} + f_0, \quad (71)$$

where $f_0 = (\mathcal{P} \Delta \mathcal{X}_k - \underline{\mathcal{W}})^T \Omega_{\mathcal{Q}} (\mathcal{P} \Delta \mathcal{X}_k - \underline{\mathcal{W}})$, $\mathbf{f}^T = 2(\mathcal{P} \Delta \mathcal{X}_k - \underline{\mathcal{W}})^T \Omega_{\mathcal{Q}} \mathcal{H}$, and $\Lambda = 2(\mathcal{H}^T \Omega_{\mathcal{Q}} \mathcal{H} + \Omega_{\mathcal{R}})$.

Adding constraints on the objective variable $\delta \underline{\mathbf{u}}$, the most general optimization problem must be solved:

$$\min_{\delta \underline{\mathbf{u}}} \frac{1}{2} \delta \underline{\mathbf{u}}^T \Lambda \delta \underline{\mathbf{u}} + \mathbf{f}^T \delta \underline{\mathbf{u}} + f_0 \quad \text{s.t.} \quad \mathbf{M} \delta \underline{\mathbf{u}} \leq \mathbf{N}. \quad (72)$$

The constraints considered above can be used to limit the control signal amplitude avoiding saturation in the actuators and to limit the maximum state error. In both cases, the constraints must be mapped to the amplitude of the control increment $\delta \mathbf{u}$ as in [14]

$$\mathbf{M} \triangleq \begin{bmatrix} \mathbf{C}_2 \\ -\mathbf{C}_2 \\ \mathcal{H} \\ -\mathcal{H} \end{bmatrix}, \quad \mathbf{N} \triangleq \begin{bmatrix} \mathbf{C}_1 (\mathbf{u}^{\max} - \Delta \mathbf{u}_{k-1}) - \mathbf{u}_k^{\text{tr}} \\ -\mathbf{C}_1 (\mathbf{u}^{\min} - \Delta \mathbf{u}_{k-1}) + \mathbf{u}_k^{\text{tr}} \\ \Delta \mathcal{X}^{\max} - \mathcal{P} \Delta \mathcal{X}_k \\ -\Delta \mathcal{X}^{\min} + \mathcal{P} \Delta \mathcal{X}_k \end{bmatrix}, \quad (73)$$

where $(\cdot)^{\max}$ and $(\cdot)^{\min}$ are, respectively, the maximum and minimum values allowed for the variable, and the matrices \mathbf{C}_1 and \mathbf{C}_2 are defined as

$$\mathbf{C}_1 = \begin{bmatrix} \mathbb{I}_{4 \times 4} \\ \mathbb{I}_{4 \times 4} \\ \vdots \\ \mathbb{I}_{4 \times 4} \end{bmatrix}, \quad \mathbf{C}_2 = \begin{bmatrix} \mathbb{I}_{4 \times 4} & \mathbf{0}_{4 \times 4} & \cdots & \mathbf{0}_{4 \times 4} \\ \mathbb{I}_{4 \times 4} & \mathbb{I}_{4 \times 4} & \cdots & \mathbf{0}_{4 \times 4} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{I}_{4 \times 4} & \mathbb{I}_{4 \times 4} & \cdots & \mathbb{I}_{4 \times 4} \end{bmatrix}. \quad (74)$$

4.4. Applied Control Signal. Using the relations $\Delta \mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_k^{\text{tr}}$ and $\delta \mathbf{u}_k = \Delta \mathbf{u}_k - \Delta \mathbf{u}_{k-1}$, the control signal at the time instant k can be written as

$$\mathbf{u}_k = \mathbf{u}_k^{\text{tr}} + \Delta \mathbf{u}_{k-1} + \delta \mathbf{u}_k, \quad (75)$$

where $\delta \mathbf{u}_k$ results from the optimization problem (72) and \mathbf{u}_k^{tr} is the feedforward term (61).

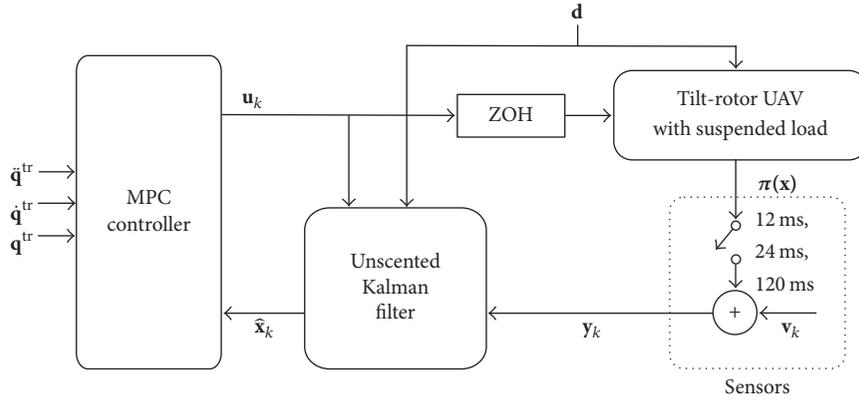


FIGURE 4: The proposed control and state estimation strategy.

5. Simulation Scenario

This section describes the simulation scenario used to evaluate the performance of the proposed control and state estimation strategy. The implemented control structure is presented in Figure 4.

5.1. Model and Design Parameters. The model parameters of the tilt-rotor UAV with suspended load are shown in Table 1. Mass, inertia, and displacement parameters of the aircraft

were obtained from CAD model, designed in Solidworks® software. The parameters related to the suspended load, as well as k_r and b , are the same considered in [12]. The gravitational acceleration is assumed constant, and λ_R and λ_L are given according to the following: the right propeller of the aircraft rotates counter-clockwise and the left one rotates clockwise. From those data, and assuming $\psi = 0$, the following equilibrium points was obtained to construct the vector \mathbf{q}^{tr} used in (60) and (61)

$$\begin{aligned} \phi^{\text{eq}} &= 0, & \theta^{\text{eq}} &= 0, & \gamma_1^{\text{eq}} &= 0.0001563, & \gamma_2^{\text{eq}} &= 0.0287134, \\ \alpha_R^{\text{eq}} &= 0.0288375, & \alpha_L^{\text{eq}} &= 0.0283718, & f_R^{\text{eq}} &= 9.7903455, & f_L^{\text{eq}} &= 9.8252665, \\ \tau_{\alpha_R}^{\text{eq}} &= 0, & \tau_{\alpha_L}^{\text{eq}} &= 0. \end{aligned} \quad (76)$$

For simplicity, the sensors' noise was assumed to have a Gaussian probability distribution, and the measurement error is defined as three times the standard deviation. However, such assumption is not required by the UKF to estimate the system states (34). Additionally, this work considers known disturbances affecting the aircraft, which may represent the presence of wind gusts, and all sensors are synchronized. Table 2 shows the sensors' measurement error and their sampling time. The relations between the measured variables and those presented in (42) are as follows: $\xi_{\mathcal{B}} \equiv \{s_1, s_2, s_3\}$, $\eta_{\mathcal{B}} \equiv \{s_4, s_5, s_6\}$, $\omega_{\mathcal{F}\mathcal{B}} \equiv \{s_7, s_8, s_9\}$, $\mathbf{d}_{\mathcal{B}\mathcal{L}} \equiv \{s_{10}, s_{11}, s_{12}\}$, $\alpha_R \equiv \{s_{13}\}$, $\alpha_L \equiv \{s_{14}\}$, $\dot{\alpha}_R \equiv \{s_{15}\}$, and $\dot{\alpha}_L \equiv \{s_{16}\}$.

For time instants k multiples of 120 ms, the measurement vector is full, being $\mathbf{y}_{k, \{T_s\}} = [s_1 \ s_2 \ s_3 \ s_4 \ s_5 \ s_6 \ s_7 \ s_8 \ s_9 \ s_{10} \ s_{11} \ s_{12} \ s_{13} \ s_{14} \ s_{15} \ s_{16}]^T$, while for those multiples of 24 ms the vector does not have GPS information, and then $\mathbf{y}_{k, \{T_s\}} = [s_3 \ s_4 \ s_5 \ s_6 \ s_7 \ s_8 \ s_9 \ s_{10} \ s_{11} \ s_{12} \ s_{13} \ s_{14} \ s_{15} \ s_{16}]^T$. Finally, for time instants multiples of 12 ms, the measurement vector does not have neither GPS nor camera information; thus $\mathbf{y}_{k, \{T_s\}} = [s_3 \ s_4 \ s_5 \ s_6 \ s_7 \ s_8 \ s_9 \ s_{13} \ s_{14} \ s_{15} \ s_{16}]^T$.

The initial state vector is considered to be precisely known, that is, $\hat{\mathbf{x}}_{0|0} = \mathbf{x}_0$, and the covariance matrices used in the filter algorithm are given by

$$\begin{aligned} \mathbf{P}_{0|0}^x &= 0.001 \cdot \mathbb{1}_{20 \times 20}, \\ \mathbf{Q}_k &= \text{diag}(0.0001 \cdot \mathbb{1}_{2 \times 1}, 0.001, 0.00001 \cdot \mathbb{1}_{3 \times 1}, 0.05 \cdot \mathbb{1}_{2 \times 1}, \\ &\quad 0.001 \cdot \mathbb{1}_{2 \times 1}, 0.01 \cdot \mathbb{1}_{6 \times 1}, 0.05 \cdot \mathbb{1}_{2 \times 1}, 0.1 \cdot \mathbb{1}_{2 \times 1}), \end{aligned} \quad (77)$$

and $\mathbf{R}_{k, \{T_s\}}$ being a diagonal matrix composed by the sensor standard deviation, that is, the measurement error divided by three, regarding the sensors information available at the time instant k .

For discretization of the linearized Jacobians, $\bar{\mathbf{A}}(\zeta(t))$ and $\bar{\mathbf{B}}(\zeta(t))$, in (63), it is assumed that $T_s = 12$ ms. The prediction and control horizons, chosen considering the trade-off between good performance and small computational cost, are $N_p = 100$ and $N_c = 10$. Furthermore, the saturation level of the tilt-rotor UAV actuators and the maximum state error allowed, used in (73), are

$$\begin{aligned}
\Delta x &= [-1, 1], & \Delta y &= [-1, 1], & \Delta z &= [-1, 1], \\
\Delta \phi &= [-0.8, 0.8], & \Delta \theta &= [-0.8, 0.8], & \Delta \psi &= [-0.8, 0.8], \\
\Delta \gamma_1 &= [-0.8, 0.8], & \Delta \gamma_2 &= [-0.8, 0.8], & \Delta \alpha_R &= [-0.8, 0.8], & \Delta \alpha_L &= [-0.8, 0.8], \\
f_R &= [0, 15], & f_L &= [0, 15], & \tau_{\alpha_R} &= [-2, 2], & \tau_{\alpha_L} &= [-2, 2],
\end{aligned} \tag{78}$$

where the error limitations were chosen regarding measurement error and disturbances effect on the system. The actuators' bounds are due to physical constraints.

The Bryson's rule [35] was used as starting point to synthesize the MPC controller's weighting matrices, which are given by

$$\begin{aligned}
\mathcal{Q} &= \text{diag} \left(\frac{40}{2^2}, \frac{40}{2^2}, \frac{20}{2^2}, \frac{5}{(\pi/2)^2}, \frac{5}{(\pi/2)^2}, \frac{10}{(\pi)^2}, \frac{10}{(\pi/2)^2}, \frac{10}{(\pi/2)^2}, \frac{0.1}{(\pi/2)^2}, \frac{0.1}{(\pi/2)^2}, \frac{10}{2^2}, \frac{10}{2^2}, \frac{5}{2^2}, \frac{1}{(\pi/3)^2}, \frac{1}{(\pi/3)^2}, \frac{1}{(\pi/4)^2}, \frac{5}{(3\pi)^2}, \right. \\
&\quad \left. \frac{5}{(3\pi)^2}, \frac{0.1}{(3\pi)^2}, \frac{0.1}{(3\pi)^2}, 40, 40, 40, 20, \frac{40}{(f_R^{\text{eq}} - 15)^2}, \frac{40}{(f_L^{\text{eq}} - 15)^2}, \frac{20}{(\tau_{\alpha_R}^{\text{eq}} - 2)^2}, \frac{20}{(\tau_{\alpha_L}^{\text{eq}} - 2)^2} \right), \\
\mathcal{R} &= \text{diag} \left(\frac{200}{(f_R^{\text{eq}} - 15)^2}, \frac{200}{(f_L^{\text{eq}} - 15)^2}, \frac{1000}{(\tau_{\alpha_R}^{\text{eq}} - 2)^2}, \frac{1000}{(\tau_{\alpha_L}^{\text{eq}} - 2)^2} \right).
\end{aligned} \tag{79}$$

5.2. Desired Trajectory. To explore the controller capabilities, the proposed trajectory to be tracked by the suspended load is composed of several interconnected paths, which are described by polynomial and/or sinusoidal functions. It starts with vertical take-off, followed by straight line tracking with changes in direction, and ends with vertical landing. Along with the desired position, yaw's movements are specified in order to always have the aircraft performing the trajectory head-on (see Figure 5). Moreover, to evaluate the disturbance compensation of the proposed strategy, external forces are applied to the suspended load. Figure 6 shows the disturbance profile for the desired trajectory, which may represent sustained wind gusts affecting the load. The magnitude of the disturbances may seem low at a first glance; however the mass of the load is very small (see Table 1).

6. Numerical Results and Discussion

This section presents the numerical results obtained with the proposed control and state estimation strategy when subjected to the simulation scenario described before. The simulations have been carried out using the MATLAB/Simulink® environment. A detailed analysis about the performance of the control system when solving the path tracking problem of a suspended load carried by a tilt-rotor UAV is provided.

The trajectories performed by the tilt-rotor UAV and the suspended load are shown in Figures 5 and 7. The tracking error is illustrated in Figure 8. The path tracking was performed successfully, from take-off to landing, and throughout the different paths that compose the trajectory, including yaw angle regulation. These results demonstrate the joint performance of the designed MPC controller and the UKF estimator, using the adopted control structure.

During the vertical take-off, the aircraft starts to fly while the load remains in the ground. Only when the distance between the aircraft and the floor is greater than the rope's length, the tilt-rotor UAV starts to carry the load. Vice versa, in the landing maneuver, the aircraft flies free of load once it has touched the ground. Figure 9 highlights this behavior in the first 5 and last 10 seconds of simulation. At the beginning of the simulation, the rope's length increases until it reaches its maximum value; then, the aircraft starts to carry the suspended load and the total mass increases. Likewise, during the landing, last 10 seconds, the load touches the ground and the rope's length decreases until the aircraft landing finishes. Also, Figure 9 shows the total mass reduction when the load touches the ground. The controller was capable of dealing with this problem due to the incorporation of $l(t)$ in the time-varying parameters vector $\zeta(t)$ and the model-based nature of predictive controllers. The load's mass was not incorporated in $\zeta(t)$ for two main reasons: (i) in order to consider the mass variation as a time-varying parameter in the same way that $l(t)$ was considered, the load's mass needs to be estimated or informed to the controller before the flight starts; (ii) it is reasonable to assume that the relation between the UAV's mass and the load's mass is small enough to be rejected by the controller as parametric uncertainty when the actual load's mass is different from the one considered in Table 1.

Figure 10 shows the time evolution of the remaining degrees of freedom, which are kept stable as the trajectory is performed. Since the aircraft's behavior is described implicitly by those variables, one can conclude that the UAV was stabilized along the trajectory. It is important to note that the designed MPC controller was able to stabilize the aircraft without the need of a cascade control structure, since the

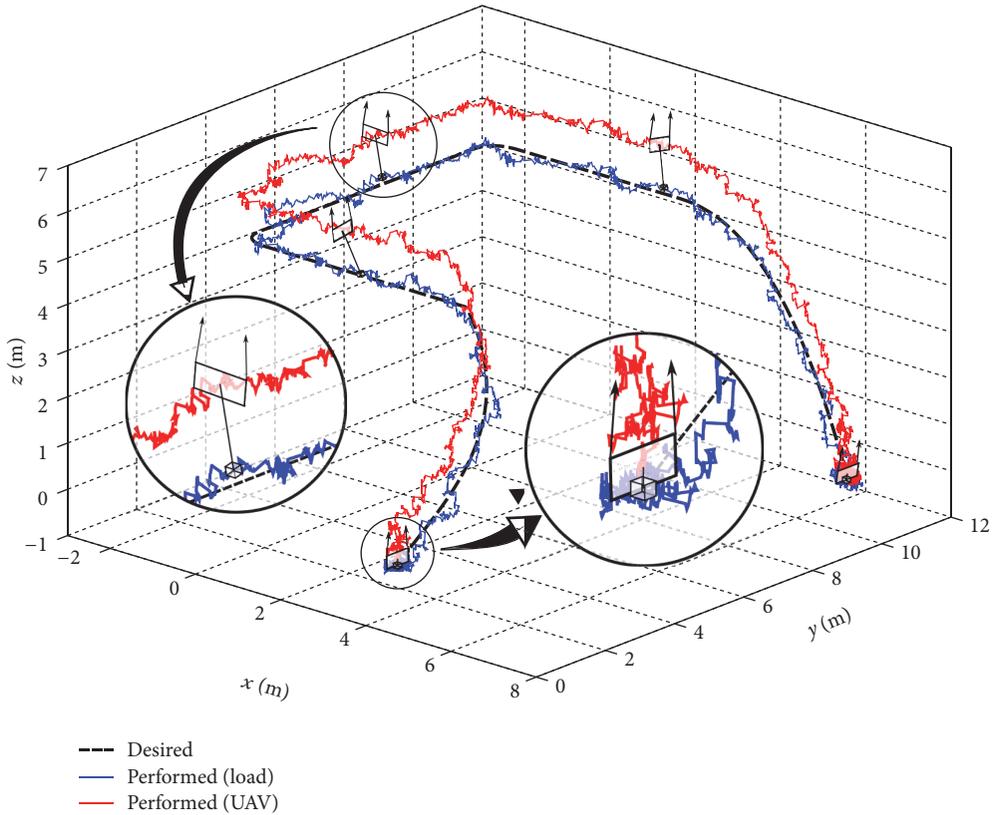


FIGURE 5: Three-dimensional trajectories of the suspended load (blue), the tilt-rotor UAV (red), and the desired trajectory (dashed black). At the beginning and at the end of the path, the rope's length is reduced, while throughout the trajectory the rope reaches its maximum value.

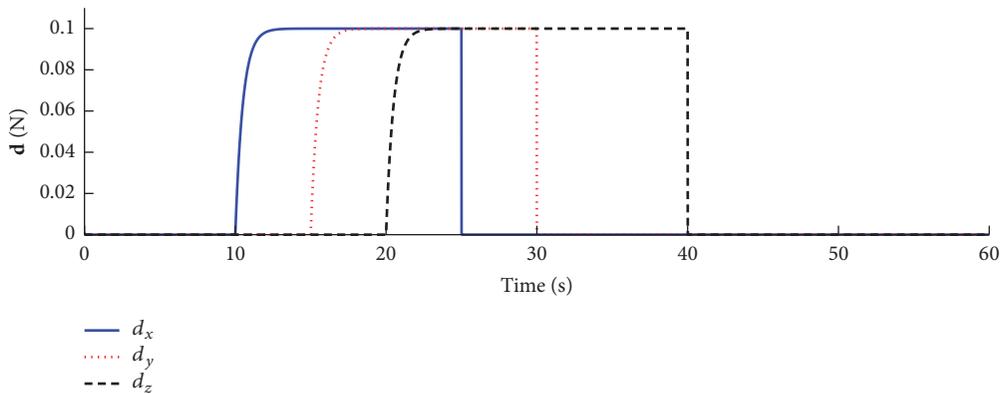


FIGURE 6: Profile of the disturbance forces applied to the load and expressed in the inertial frame \mathcal{F} .

proposed modeling considered all the dynamic couplings between the rigid bodies.

Figure 11 shows the actuator signals generated by the MPC controller. Despite being noisy, the control signals did not reach the actuators' limits. In an experimental setup, the inherent inertial properties of the aircraft actuators, which are commonly brushless DC motors and servomotors, would attenuate such noise naturally.

Figures 12 and 13 present, respectively, the estimation error for the generalized coordinates and for their time derivatives, along with the confidence limits (i.e., three times the standard deviation). In all cases, the estimation error is close to zero and is kept inside the confidence limits, demonstrating the performance of the unscented Kalman filter. Recalling the nonlinearities of the system (35) and the measurement equation (42), the UKF was then able

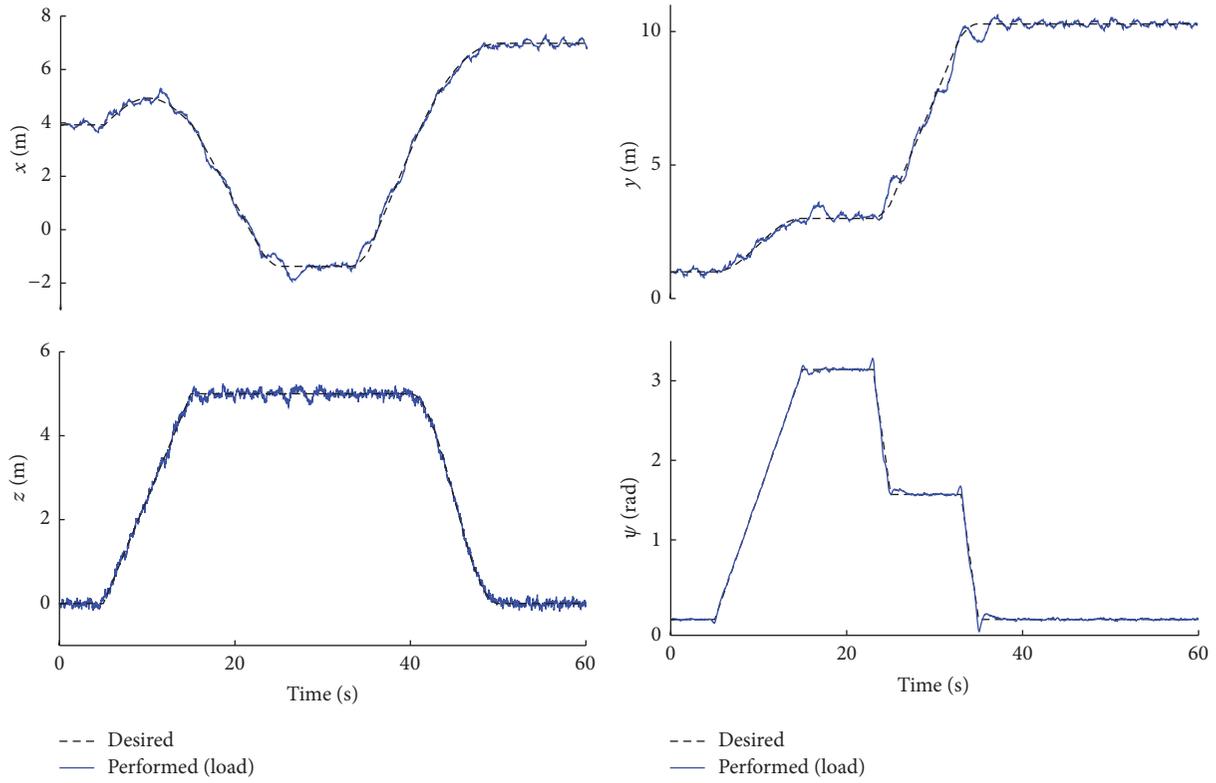


FIGURE 7: Time evolution of the regulated variables (x, y, z, ψ) and their desired trajectories.

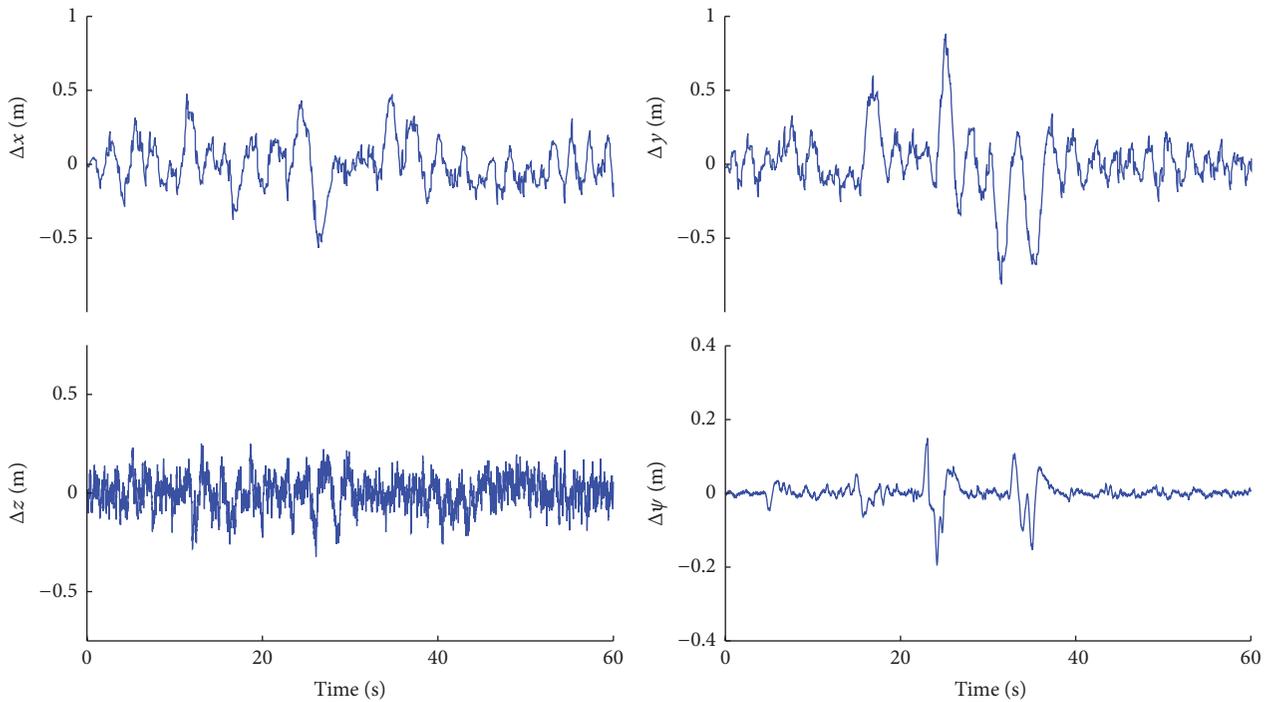


FIGURE 8: Time evolution of the tracking errors. The oscillatory behavior around zero is due to the presence of external disturbances and noisy measurements.

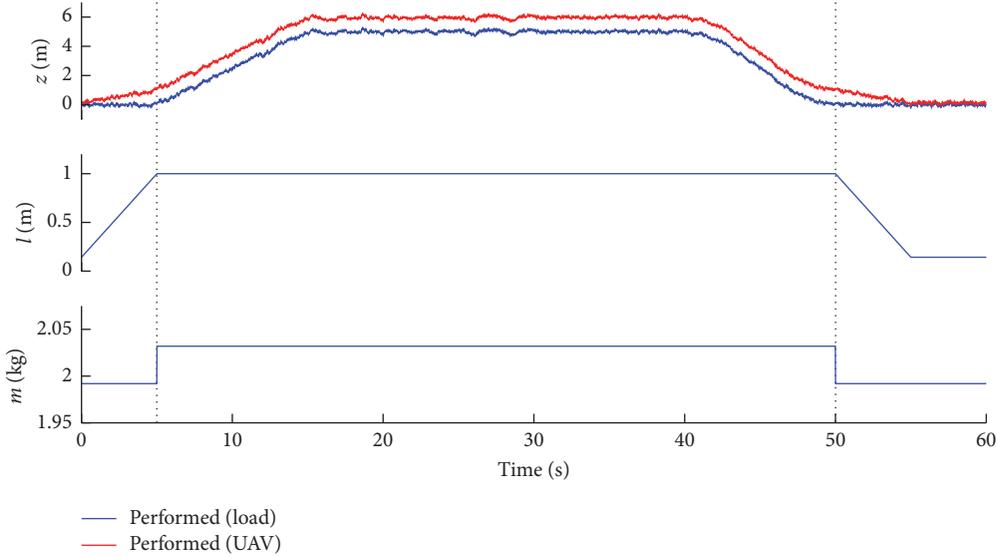


FIGURE 9: Analysis of the time evolution of the load and UAV altitude in take-off and landing maneuvers together with the rope's length and the total mass variation.

TABLE 1: Model parameters of the tilt-rotor UAV with suspended load.

Parameter	Value
$m_{\mathcal{L}}$	0.05000 Kg
m_1	1.70249 Kg
m_2, m_3	0.13973 Kg
l_{\max}	1 m
$\mathbf{d}_{\mathcal{B}}^{\mathcal{L}}$	$[0 \ 0 \ l]^T$ m
$\mathbf{d}_{\mathcal{E}_1}^{\mathcal{B}}$	$[-0.00433 \ 0.00060 \ -0.04559]^T$ m
$\mathbf{d}_{\mathcal{E}_2}^{\mathcal{B}}$	$[0.00002 \ -0.27761 \ 0.05493]^T$ m
$\mathbf{d}_{\mathcal{E}_3}^{\mathcal{B}}$	$[0.00077 \ 0.27761 \ 0.05493]^T$ m
$\mathbf{I}_{\mathcal{L}}$	$2.645 \cdot 10^{-6} \cdot \mathbb{I}_{3 \times 3}$ Kg·m ²
\mathbf{I}_1	$\begin{bmatrix} 3697.66749 & 0.36342 & -9.51029 \\ * & 840.10403 & 0.61804 \\ * & * & 3865.05354 \end{bmatrix} \cdot 10^{-6}$ Kg·m ²
\mathbf{I}_2	$\begin{bmatrix} 441.68245 & 0 & 0 \\ * & 441.67985 & -1.07006 \\ * & * & 0.64418 \end{bmatrix} \cdot 10^{-6}$ Kg·m ²
\mathbf{I}_3	$\begin{bmatrix} 441.68245 & 0 & 0 \\ * & 441.67985 & 1.07006 \\ * & * & 0.64418 \end{bmatrix} \cdot 10^{-6}$ Kg·m ²
$\hat{\mathbf{g}}$	$[0 \ 0 \ -9.81]^T$ m/s ²
k_{τ}	$1.7 \cdot 10^{-7}$ N·m·s ²
b	$9.5 \cdot 10^{-6}$ N·s ²
(λ_R, λ_L)	(1, -1)
β	5°
$\mu_{\gamma}, \mu_{\alpha}$	0.005 N·m/(rad/s)

TABLE 2: Sensors parameters for the considered scenario.

	Sensor	Measurement error	Sampling time
GPS	$\{s_1, s_2\}$	± 0.15 m	120 ms
Barometer	$\{s_3\}$	± 0.51 m	12 ms
IMU	$\{s_4, s_5, s_6\}$	$\pm 2.618 \cdot 10^{-3}$ rad	12 ms
	$\{s_7, s_8, s_9\}$	$\pm 16.558 \cdot 10^{-3}$ rad/s	
Camera	$\{s_{10}, s_{11}\}$	± 0.005 m	24 ms
	$\{s_{12}\}$	± 0.02 m	
Servos	$\{s_{13}, s_{14}\}$	$\pm 5.67 \cdot 10^{-3}$ rad	12 ms
	$\{s_{15}, s_{16}\}$	± 0.50772 rad/s	

to estimate the means and covariances of the posterior distributions in a consistent manner, despite the sensors' different sampling rates.

Some patterns arose in the confidence limits due to the greater sampling rates of the GPS and the camera, whose measurements are available only every 120 ms and 24 ms for performing the data assimilation step, respectively. The other sensors' data are available every 12 ms, which is also the controller sampling time. These patterns are expected since the estimation is more accurate and has smaller confidence limits every time that more data are available to be used in the data assimilation step. This result illustrates that, despite having nonlinear dynamics and sensors with different sampling rates, the unscented Kalman filter is able to recover the state vector from the information provided only by the UAV's embedded sensors.

Based on the presented results, the proposed modeling, control and estimation strategies were demonstrated to be appropriate to solve the problem of path tracking control of the suspended load. A next research step consists in validating the proposed approach considering an experimental setup.

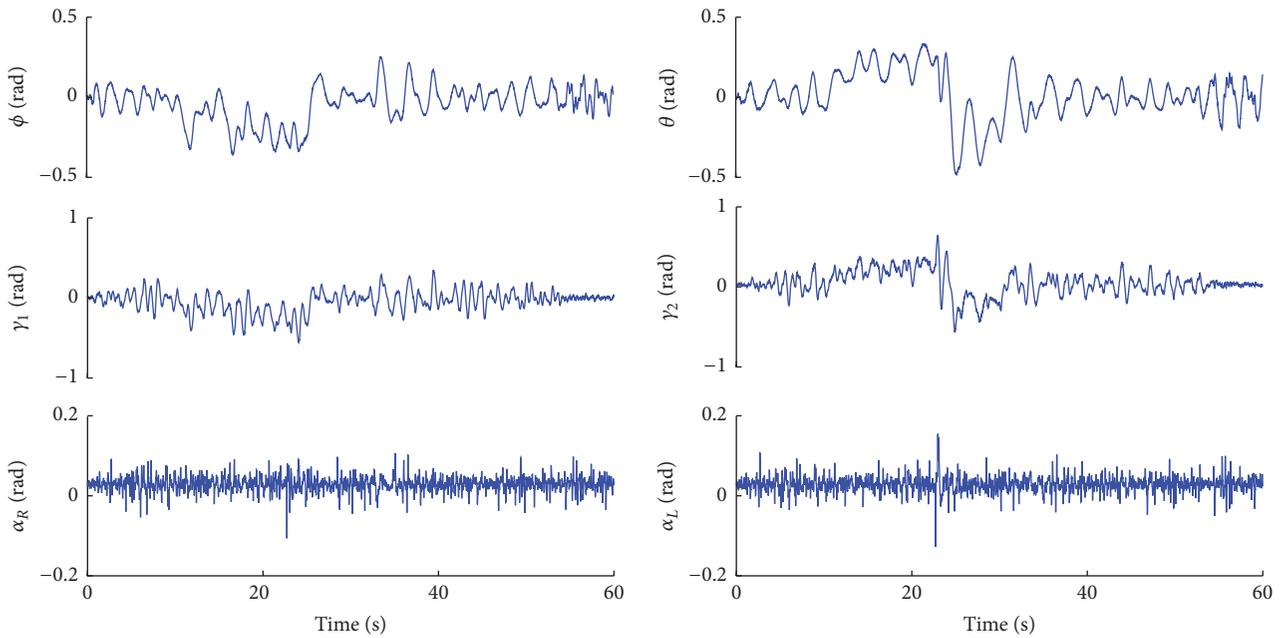


FIGURE 10: Time evolution of the remaining degrees of freedom. The variations on ϕ , θ , γ_1 , and γ_2 are due to changes on their equilibrium points when external disturbances affect the system.

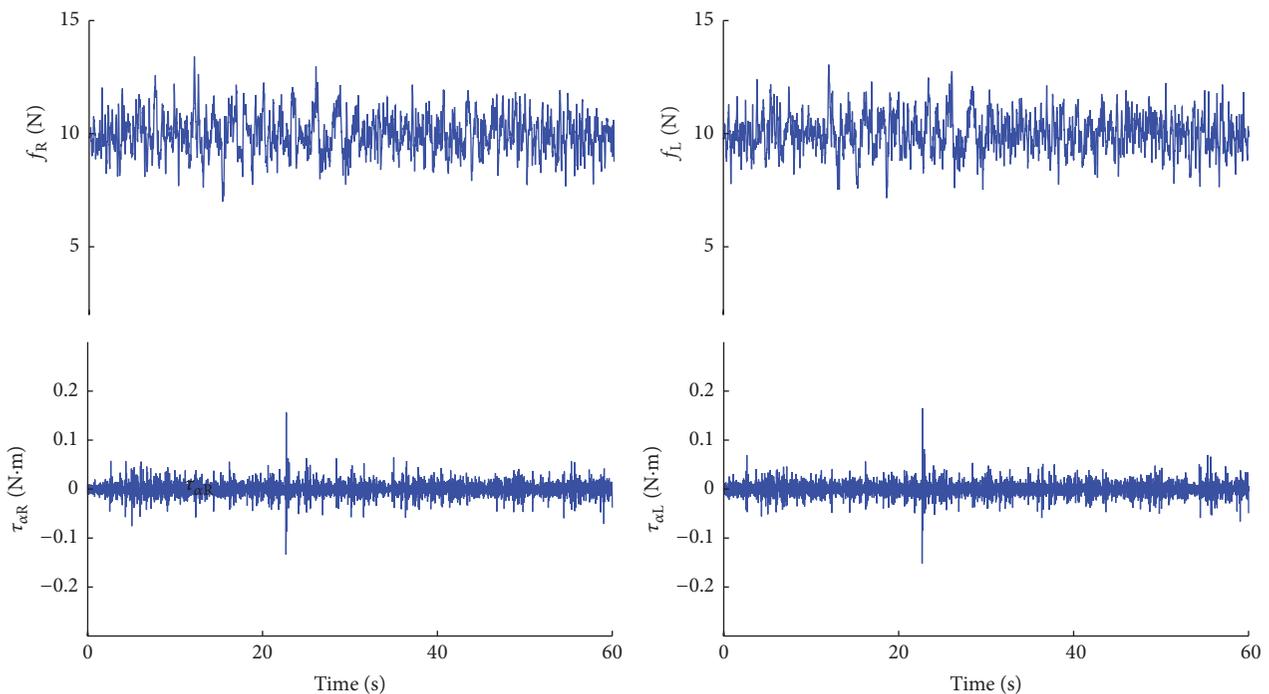


FIGURE 11: Applied thrusts and torques to the tilt-rotor UAV by the propellers and servomotors, respectively.

7. Conclusions

This paper proposed a detailed model and the design of a control and state estimation strategy to solve the path tracking control problem of a suspended load using a tilt-rotor UAV when it is operating in the helicopter flight-mode.

A modeling approach was presented, in which the kinematics of the system were formulated from the load's perspective, being the load's position and orientation chosen as degrees of freedom of the multibody mechanical system. The UAV's position and orientation were described only with respect to the load. By using the Euler-Lagrange formulation,

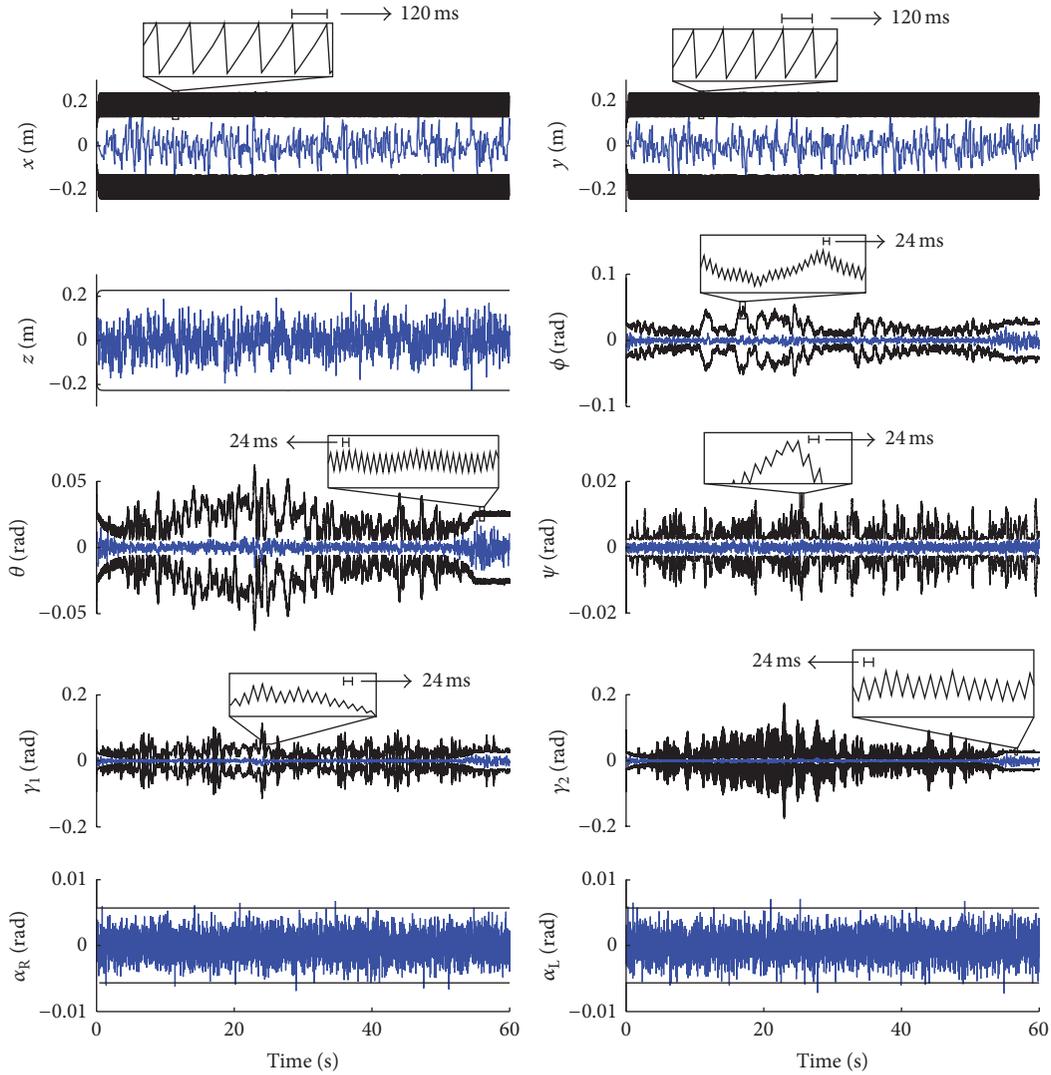


FIGURE 12: Time evolution of the estimation error of the generalized coordinates, along with the confidence limits. Patterns are highlighted to illustrate the estimation behavior in presence of different sampling rates.

the equations of motion were obtained, while taking into account the dynamic coupling between the aircraft and the load, the existence of viscous friction at the suspension point and at the tilting mechanisms, and also external forces affecting the load. The nonlinear state-space representation of the system was obtained, with the load's position and orientation as state variables.

Assuming that the UAV's embedded sensors provide noisy information with different sampling rates, an unscented Kalman filter was proposed for nonlinear state estimation of all the state variables, based on the model attained and measurement equations developed from kinematic concepts. Moreover, based on linearized time-varying state-space equations augmented with integral actions, a model predictive control strategy was designed for path tracking of the suspended load with stabilization of the tilt-rotor UAV. The proposed model predictive controller allows yaw

angle tracking, take-off and landing maneuvers, situations where the rope's length and the total system mass are not constant. The feedback control loop was performed using the estimation provided by the UKF.

The proposed strategy in this work was evaluated through numerical experiments in MATLAB/Simulink environment. The trajectory performed by the load comprised vertical take-off, straight line following with direction changing, and vertical landing, with disturbances being applied to the load and also under total mass and rod's length variation. The unscented Kalman filter was able to estimate the entire state vector from the information provided by the sensors, while the MPC controller was able to perform the control task using the provided estimation, without saturating the aircraft actuators. The presented results demonstrated the good performance of the designed MPC controller and the unscented Kalman filter for path tracking of the load

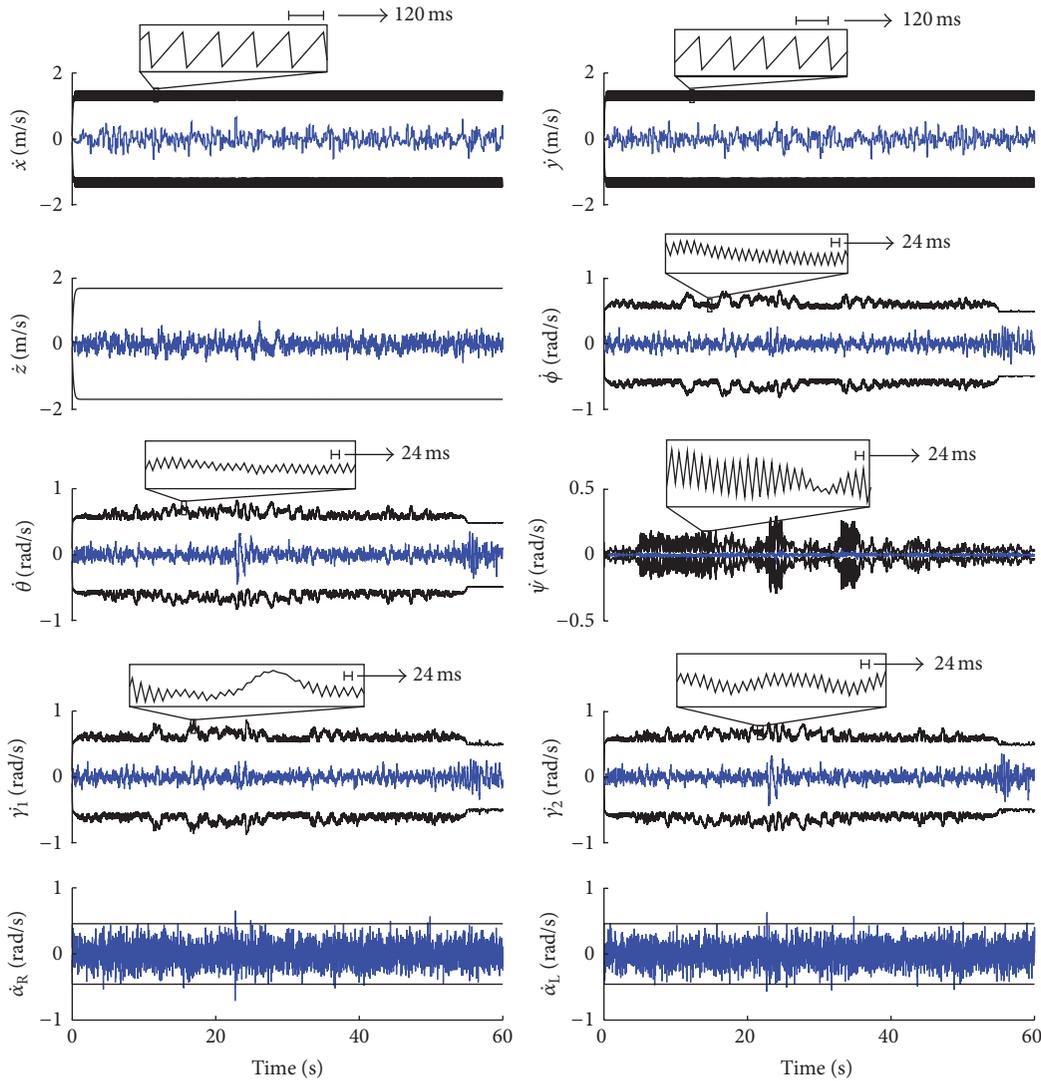


FIGURE 13: Time evolution of the estimation error of the generalized coordinates derivatives, along with the confidence limits. Patterns are highlighted to illustrate the estimation behavior in presence of different sampling rates.

using a tilt-rotor UAV, in this challenging, not yet addressed scenario.

One of the biggest appeals of tilt-rotor UAVs is their ability to take-off and land vertically, perform hover flights, and achieve improved forward velocities when in airplane-like mode. This work dealt only with helicopter-like flights, in which the thrusters' groups motions are limited to small tilting angles. However, as stated before, the inclusion of aerodynamic surfaces into the proposed model is straightforward. Further, the proposed estimation and control strategy is general enough to cope with both flight modes, including the transition between them.

7.1. Future Works and Limitations. The main limitation of the proposed approach is the computational cost associated with the estimation and control algorithms. Since one of the goals of this work was to perform yaw angle tracking, the

use of a nonlinear estimator and a controller based on a linearized time-varying model became necessary. As discussed in Section 4, this model is not affine in the parameters; that is, it is not possible to obtain a polytopic representation for the linearized model. This fact prevents the use of techniques that could allow the prediction horizon reduction, which is directly related to the controller computational cost. Regarding the estimator, the requirement to perform yaw motion on flight, which makes the linearized model unsuitable far from the chosen operational points, together with the inability to derive Jacobian and Hessian matrices from the model derived in Section 2 using numerical computation software, made the choice of a filter with higher computational cost necessary.

Future works will propose ways to reduce the computational cost of the proposed controller allowing implementing it in embedded systems. Approaches that use piecewise linearized models or models identified in subspace models

could be a possible way to address this issue. Also, studying computational efficient ways to implement UKF algorithms will be a future work. Once the problems related to computational cost have been solved and the prototype is finished, future works will present the aircraft aerodynamic design, that is currently being developed, together with simulation and experimental results for load transportation tasks.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Brazilian agencies CNPq, CAPES, FAPEMIG, and Pró-Reitoria de Pesquisa de UFMG. A. Ferramosca would like to acknowledge the Argentinian Agency of Scientific and Technological Development (ANPCyT) under the FONCyT Grant PICT-2016-0283.

References

- [1] N. Amiri, A. Ramirez-Serrano, and R. Davies, "Modelling of Opposed Lateral and longitudinal Tilting dual-fan unmanned aerial vehicle," in *Proceedings of the 18th IFAC World Congress*, pp. 2054–2059, Milano, Italy, September 2011.
- [2] S. Park, J. Bae, Y. Kim, and S. Kim, "Fault tolerant flight control system for the tilt-rotor UAV," *Journal of the Franklin Institute. Engineering and Applied Mathematics*, vol. 350, no. 9, pp. 2535–2559, 2013.
- [3] D. N. Cardoso, G. V. Raffo, and S. Esteban, "A robust adaptive mixing control for improved forward flight of a tilt-rotor UAV," in *Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1432–1437, Rio de Janeiro, Brazil, November 2016.
- [4] M. Bernard, K. Kondak, I. Maza, and A. Ollero, "Autonomous transportation and deployment with aerial robots for search and rescue missions," *Journal of Field Robotics*, vol. 28, no. 6, pp. 914–931, 2011.
- [5] F. Wang, P. Liu, S. Zhao et al., "Guidance, navigation and control of an unmanned helicopter for automatic cargo transportation," in *Proceedings of the 33rd Chinese Control Conference, CCC 2014*, pp. 1013–1020, China, July 2014.
- [6] M. Bisgaard, *Modeling, estimation and control of helicopter slung load system [Ph.D. Thesis]*, Aalborg University, 2008.
- [7] M. Bisgaard, A. la Cour-Harbo, E. N. Johnson, and J. D. Bendtsen, "Vision aided state estimator for helicopter slung load system," in *Proceedings of the 17th IFAC Symposium on Automatic Control in Aerospace*, vol. 40, pp. 425–430.
- [8] M. Bisgaard, A. La Cour-Harbo, and J. D. Bendtsen, "Full state estimation for helicopter slung load system," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 1–15, USA, August 2007.
- [9] M. Bisgaard, A. La Cour-Harbo, and J. D. Bendtsen, "Swing damping for helicopter slung load systems using delayed feedback," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, pp. 1–11, USA, August 2009.
- [10] I. Palunko, P. Cruz, and R. Fierro, "Agile load transportation : Safe and efficient load manipulation with aerial robots," *IEEE Robotics and Automation Magazine*, vol. 19, no. 3, pp. 69–79, 2012.
- [11] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia, "Learning swing-free trajectories for UAVs with a suspended load," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation, ICRA 2013*, pp. 4902–4909, Karlsruhe, Germany, May 2013.
- [12] M. M. Almeida and G. V. Raffo, "Nonlinear control of a tiltrotor UAV for load transportation," in *Proceedings of the 11th IFAC Symposium on Robot Control*, pp. 234–239, 2015.
- [13] G. V. Raffo and M. M. De Almeida, "Nonlinear robust control of a quadrotor UAV for load transportation with swing improvement," in *Proceedings of the 2016 American Control Conference, ACC 2016*, pp. 3156–3162, USA, July 2016.
- [14] M. A. Santos and G. V. Raffo, "Path tracking Model Predictive Control of a Tilt-rotor UAV carrying a suspended load," in *Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1458–1463, Rio de Janeiro, Brazil, November 2016.
- [15] A. La Cour-Harbo and M. Bisgaard, "State-control trajectory generation for helicopter slung load system using optimal control," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, pp. 1–16, USA, August 2009.
- [16] S. Tang and V. Kumar, "Mixed Integer Quadratic Program trajectory generation for a quadrotor with a cable-suspended payload," in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation, ICRA 2015*, pp. 2216–2222, USA, May 2015.
- [17] M. Bernard and K. Kondak, "Generic slung load transportation system using small size helicopters," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation, ICRA '09*, pp. 3258–3264, Japan, May 2009.
- [18] T. Lee, K. Sreenath, and V. Kumar, "Geometric control of cooperating multiple quadrotor UAVs with a suspended payload," in *Proceedings of the 52nd IEEE Conference on Decision and Control, CDC 2013*, pp. 5510–5515, Italy, December 2013.
- [19] I. Palunko, A. Faust, P. Cruz, L. Tapia, and R. Fierro, "A reinforcement learning approach towards autonomous suspended load manipulation using aerial robots," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation, ICRA 2013*, pp. 4896–4901, Karlsruhe, Germany, May 2013.
- [20] K. Sreenath, T. Lee, and V. Kumar, "Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load," in *Proceedings of the 52nd IEEE Conference on Decision and Control, CDC 2013*, pp. 2269–2274, Italy, December 2013.
- [21] P. O. Pereira and D. V. Dimarogonas, "Lyapunov-based generic controller design for thrust-propelled underactuated systems," in *Proceedings of the 2016 European Control Conference (ECC)*, pp. 594–599, Aalborg, Denmark, June 2016.
- [22] P. O. Pereira, M. Herzog, and D. V. Dimarogonas, "Slung load transportation with a single aerial vehicle and disturbance removal," in *Proceedings of the 24th Mediterranean Conference on Control and Automation, MED 2016*, pp. 671–676, Athens, Greece, June 2016.
- [23] B. S. Rego and G. V. Raffo, "Suspended load path tracking control based on zonotopic state estimation using a tilt-rotor UAV," in *Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1445–1451, Rio de Janeiro, Brazil, November 2016.
- [24] B. S. Rego and G. V. Raffo, "Suspended load path tracking by a tilt-rotor UAV," in *Proceedings of the 1st IFAC Conference on Cyber-Physical & Human-Systems*, pp. 229–234, 2016.

- [25] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "Nonlinear H_∞ controller for the quad-rotor helicopter with input coupling," in *Proceedings of the 18th IFAC World Congress*, pp. 13834–13839, Milano, Italy, September 2011.
- [26] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, John Wiley & Sons, Inc, 2006.
- [27] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, Springer Science & Business Media, 2009.
- [28] A. A. Shabana, *Computational Dynamics*, John Wiley and Sons, 3rd edition, 2010.
- [29] T. R. Kane and D. A. Levinson, *Dynamics: Theory and Applications*, McGraw-Hill College, 1985.
- [30] P. Castillo, R. Lozano, and A. Dzul, "Stabilization of a mini rotorcraft with four rotors: experimental implementation of linear and nonlinear control laws," *IEEE Control Systems Magazine*, vol. 25, no. 6, pp. 45–55, 2005.
- [31] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.
- [32] S. Särkkä, *Bayesian Filtering and Smoothing*, vol. 3, Cambridge University Press, Cambridge, UK, 2013.
- [33] J. A. Rossiter, *Model-Based Predictive Control: A Practical Approach*, CRC press, 2013.
- [34] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "An integral predictive/nonlinear H_∞ control structure for a quadrotor helicopter," *Automatica*, vol. 46, no. 1, pp. 29–39, 2010.
- [35] M. A. Johnson and M. J. Grimble, "Recent trends in linear optimal quadratic multivariable control system design," *IEE Proceedings D: Control Theory and Applications*, vol. 134, no. 1, pp. 53–71, 1987.

Research Article

Doppler Effect-Based Automatic Landing Procedure for UAV in Difficult Access Environments

Jan M. Kelner and Cezary Ziółkowski

Institute of Telecommunications, Faculty of Electronics, Military University of Technology, Gen. Sylwester Kaliski Str. No. 2, 00-908 Warsaw, Poland

Correspondence should be addressed to Jan M. Kelner; jan.kelner@wat.edu.pl

Received 23 June 2017; Accepted 14 August 2017; Published 4 October 2017

Academic Editor: Seungjae Lee

Copyright © 2017 Jan M. Kelner and Cezary Ziółkowski. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Currently, almost unrestricted access to low-lying areas of airspace creates an opportunity to use unmanned aerial vehicles (UAVs), especially those capable of vertical take-off and landing (VTOL), in transport services. UAVs become increasingly popular for transporting postal items over small, medium, and large distances. It is forecasted that, in the near future, VTOL UAVs with a high take-off weight will also deliver goods to very distant and hard-to-reach locations. Therefore, UAV navigation plays a very important role in the process of carrying out transport services. At present, during the flight phase, drones make use of the integrated global navigation satellite system (GNSS) and the inertial navigation system (INS). However, the inaccuracy of GNSS + INS makes it unsuitable for landing and take-off, necessitating the guidance of a human UAV operator during those phases. Available navigation systems do not provide sufficiently high positioning accuracy for an UAV. For this reason, full automation of the landing approach is not possible. This paper puts forward a proposal to solve this problem. The authors show the structure of an autonomous system and a Doppler-based navigation procedure that allows for automatic landing approaches. An accuracy evaluation of the developed solution for VTOL is made on the basis of simulation studies.

1. Introduction

Military applications [1–3] were the beginning of the development of unmanned aircraft systems. The increase in their availability contributes to the widespread use of unmanned aerial vehicles (UAVs) in civilian applications. In this case, monitoring large areas of land or sea is the main purpose. It concerns such areas of human activity as an agriculture [4, 5], energetics (i.e., photovoltaic plants [6–8] and high voltage lines [9]), environment protection [10], search and rescue [11], forestry and fire detection [12, 13], water area management [14, 15], and so on. An automation of monitoring procedures, low costs, and minimization of human resources in the UAV exploitation are conducive to the dynamic growth of their use in the civilian applications. The UAV monitoring is mainly based on optical sensors, whereas take-off and landing are usually done in the same place. Currently, almost unrestricted access to low-lying areas of airspace creates an opportunity to use UAVs in transport services. In this case, the place of the take-off and landing is far removed. This significantly hinders

the implementation of navigation procedures, especially at the landing stage. In this article, we present a method of automatic landing approach that can be used especially in UAV transport services over long distances.

The advantage of this mode of transport is its independence of road infrastructure, traffic volume, and difficult terrain conditions. Hence, UAVs are increasingly used to transport long distances to small postal items and medicines in hard-to-reach areas. A practical example of such a solution, developed at the initiative of the United Nations Children's Fund (UNICEF), is the use of UAVs to transport blood samples in Malawi (Africa) [16]. Nowadays, the transportation of blood at close distance between hospitals [17] or other packages [18] is already achieved. In the future, increasing the load capacity of UAVs will enable fast transport services in hard-to-reach environments such as islands, mountainous, desert, and polar areas (i.e., the Arctic and the Antarctic). In this case, vertical take-off and landing (VTOL) UAVs will play a special role, as they do not require landing strips, but only small landing pads. Hence, VTOLs may be used to land at

such locations as islands, oil platforms, vessels, or skyscraper roofs.

The basic method for navigating UAVs over long distances is based on the use of a global navigation satellite system (GNSS) supported by the UAV's own inertial navigation system (INS) [19–22]. However, these systems cannot be directly used in the final stage of the flight, that is, during landing, due to the low positioning accuracy of moving objects inherent in such systems. A practical solution to this problem is to use optical cameras, which allow the operator to remotely control the landing process. However, this method requires the use of a broadband control channel and may only be used during daylight and in good visibility conditions. Under night conditions and in poor visibility, a thermal imaging camera [6, 23–25] or synthetic aperture radar (SAR) [26–28] may be used. Furthermore, long distance wireless communications are characterized by significant delays and, in the case of the hard-to-reach areas, only satellite communications [29] can be used. In these cases, the navigation system requires a large and expensive extension, which prevents its commercial use. High costs and, above all, the necessity to change the destination location (landing area) for the transport also prevent the use of conventional landing navigation systems [20, 21, 30, 31] such as the instrument landing system (ILS), microwave landing system (MLS), or local area augmentation system (LAAS). In this case, the operational range for these systems is limited to the space around large airports.

The high precision required for determining the current position of the object and the required flexibility in terms of landing spots limit the ability to use UAVs for air transport. Therefore, the study of a precise and simple positioning method, which would give the user flexibility in terms of landing locations and conditions, is essential for development of this transport sector. This paper presents a proposal for a solution that uses spatially distinctive features of the Doppler effect. The presented navigation procedure is based on an analytical relationship that describes the Doppler frequency shift (DFS), as a function of the receiver position [32]. This formula is the basis of the signal processing method called the signal Doppler frequency (SDF) [33–35], which is used in the location systems of emission sources [36] and navigation of objects [37, 38]. Using this method enables complete automation of the UAV landing procedure and eliminates the need for a broadband remote control channel. Contemporary navigational systems often make use of pulse signals. Consequently, these systems require large spectrum resources. In contrast to these solutions, the developed system is based on narrowband signals (harmonics), which minimizes the spectrum cost. Simulation studies for VTOL are performed in order to determine the effectiveness of the developed procedure, that is, the accuracy with which the flight trajectory is determined. The obtained results show the position errors that may occur during the VTOL landing process and thus provide the opportunity to evaluate the practical implementation of the developed procedure.

The remainder of the paper is organized as follows. Section 2 presents operation principles of the Doppler-based landing approach system for UAVs. The authors show a

structure of this system and a navigation procedure for the landing phase. Simulation scenarios are described in Section 3. In simulation studies, the authors assume that the vehicle is VTOL-capable. In Section 4, obtained results are presented. Section 5 contains the summary of the paper.

2. Operation Principle of the Proposed Landing System

Assuming that the receiver (i.e., the UAV) moves at constant velocity, v , the relationship between DFS, f_D , and the signal source coordinates, (x, y, z) , is described by [32]

$$f_D(x, y, z, t) \cong f_{D\max} \frac{x - vt}{\sqrt{(x - vt)^2 + y^2 + z^2}}, \quad (1)$$

where $f_{D\max} = f_0 v/c$ is the maximum DFS, f_0 is the carrier frequency of the emitted signal, and c is the speed of light.

Based on (1), it follows that, for known (x, y, z) , direction, and values of v , DFS measurement gives the possibility of determining the coordinates of two possible positions of the receiver. By using a system of two reference sources that are located at the distance r , we eliminate the ambiguity of the result. Averaging the results obtained from several reference sources reduces the error of estimation for the position coordinates of the object. Thus, increasing the number of reference sources increases the accuracy of the positioning of the receiver. The structure of the narrowband automatic landing system based on the SDF method is shown in Figure 1.

The basic elements of the system are the narrowband navigation receiver (NR) and GNSS receiver integrated with INS, both of which are installed in the UAV, and four radio beacons (RBs) that serve as reference signal sources. Three of RBs emit harmonic signals at frequencies f_1 , f_2 , and f_3 , respectively. The fourth RB emits a modulated signal that contains information about the location of each RB relative to the destination UAV landing site. In addition, the narrowband measurement receiver (MR) of this RB measures the current frequency of each RB. This information is also included with the modulating signal. This minimizes the impact of RB instability on the DFS determination accuracy [39]. The frame structure of the modulating signal is presented in Figure 2. Transmission of information contained in the modulation signal frame is based on differential binary phase shift keying (DPSK). Selecting this modulation type makes it easy to eliminate signal modulation. The operation of raising the DPSK signal to the second power provides for the reconstruction of the carrier wave whose frequency includes DFS [40].

As shown in Figures 1 and 4, the UAV navigation process consists of two essential stages: long-range navigation and landing. Long-range navigation uses integrated GNSS + INS, because at this flight stage, accurate positioning is not a critical issue. However, the landing stage requires high positioning accuracy. The standard GNSS receiver cannot achieve such accuracy, and thus navigation stage requires the use of dedicated solutions. A generalized algorithm of the precise UAV positioning at the landing stage is shown in Figure 3.

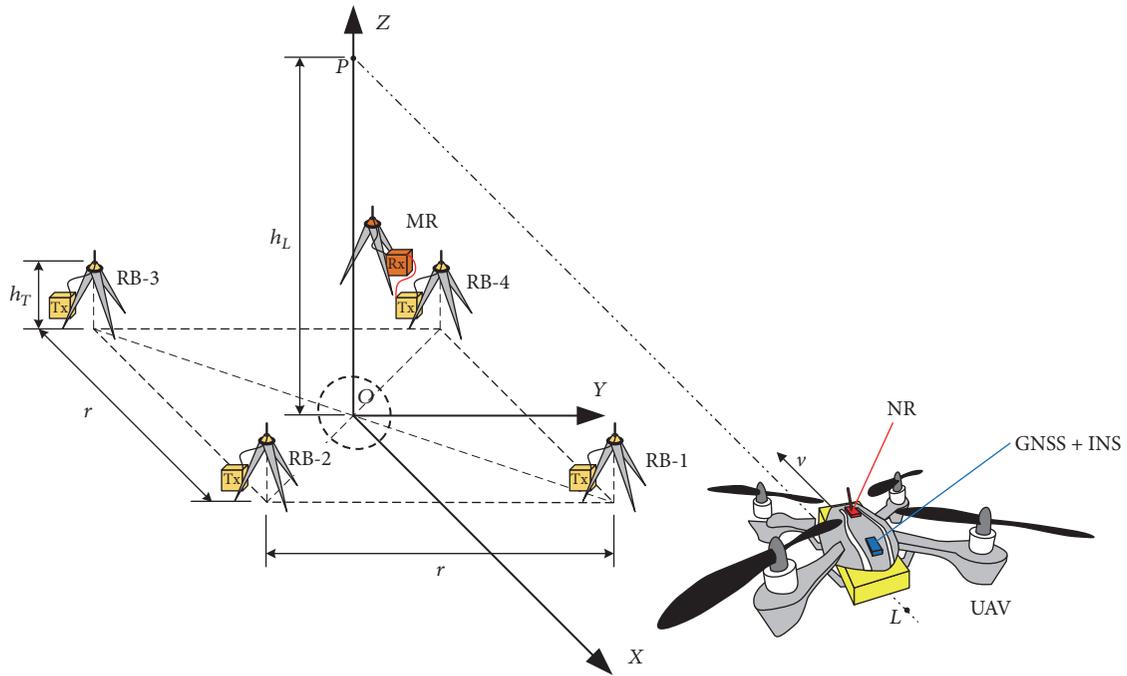


FIGURE 1: Structure of the automatic landing system.

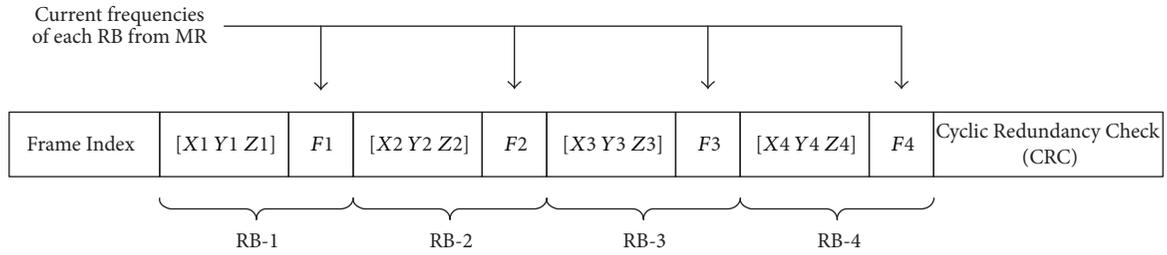


FIGURE 2: Frame structure of the modulating signal.

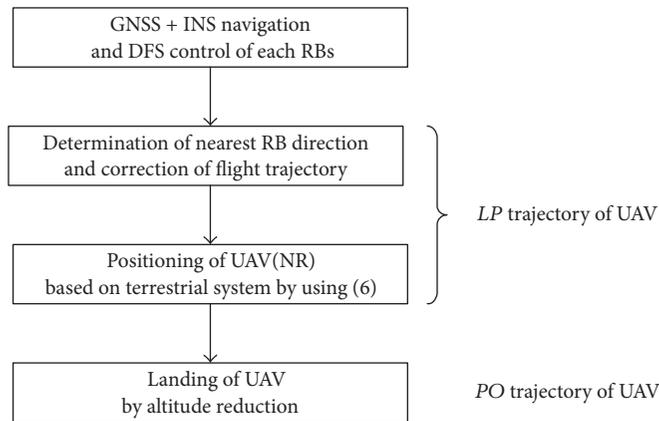


FIGURE 3: Generalized algorithm of the precise UAV positioning at the landing stage.

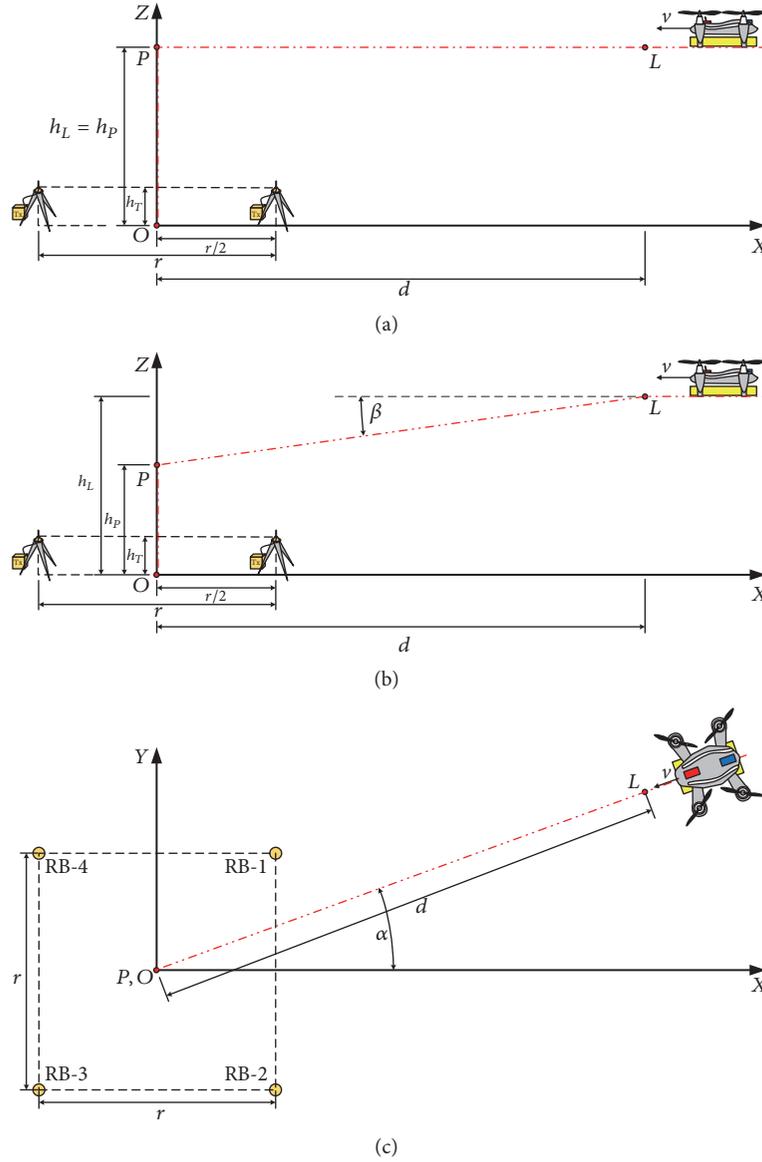


FIGURE 4: Direction of VTOL arrival to landing site with respect to the location of RBs.

Strictly determined DFSs, which occur for a close proximity to particular RBs, are the criterion for the transition of the UAV navigation system to the landing stage (point L , see Figures 1 or 4). At a relatively large distance from the landing site, the azimuth angle, φ , has a dominant impact on f_D . In this case, it is

$$\theta \cong \varphi, \quad \text{so } f_D \cong f_{D_{\max}} \cos \varphi, \quad (2)$$

where θ is the angle between the velocity vector of the receiver (i.e., UAV) and the direction to the signal source (i.e., RB).

Hence, changes of DFSs on the LP (see Figures 1 or 4) line segment are the basis for correcting the UAV flight direction at a fixed altitude. Length of this segment is d . Because UAV is moving in the direction of signal sources, the criterion of this correction is a simultaneous maximization of DFSs for the all RBs' signals. In close proximity, the elevation

angle begins to play a significant impact. If the smallest DFSs decrease to a certain criteria value, for example, $0.8f_{D_{\max}}$, then UAV changes the flight direction by θ and begins to approach toward the nearest RB. During movement, the RBs' coordinates are determined in NR with respect to the system whose O (see Figures 1 or 4) is the origin and the x -axis coincides with the direction of the new trajectory. The formulas that describe the RBs' coordinates are defined using the SDF method [33]

$$\begin{aligned} \tilde{x}_k &= v \frac{t_2 A_k(t_2) - t_1 A_k(t_1)}{t_2 A_k(t_2) - t_1 A_k(t_1)}, \quad \forall_{k=1, \dots, K} \\ \tilde{y}_k^2 + \tilde{z}_k^2 &= \left[v \frac{(t_2 - t_1) A_k(t_2) A_k(t_1)}{A_k(t_2) - A_k(t_1)} \right]^2, \quad \forall_{k=1, \dots, K}, \end{aligned} \quad (3)$$

where $(\tilde{x}_k, \tilde{y}_k, \tilde{z}_k)$ is the estimated coordinates of the k th RB, $A_k(t) = \sqrt{1 - F_k^2(t)/|F_k(t)|}$, $F_k(t) = [c\tilde{f}_{Dk}(t)]/[vf_k(t)]$, $\tilde{f}_{Dk}(t)$ is DFS for the k th RB estimated by NR at t time moment, $f_k(t)$ is the carrier frequency of the k th RB signal measured by MR and obtained from the last received data frame, and t_1 and t_2 are two different time moments of the DFS measurement.

The navigational coordinates of UAV are determined by transforming the RBs' coordinates and the coordinates obtained by the system based on the UAV's flight trajectory

$$\begin{aligned}\tilde{x}_{\text{UAV}k} &= x_k + (\tilde{x}_k \cos \tilde{\alpha} + \tilde{y}_k \sin \tilde{\alpha}) \cos \tilde{\beta} + \tilde{z}_k \sin \tilde{\beta}, \\ &\quad \forall_{k=1, \dots, K} \\ \tilde{y}_{\text{UAV}k} &= y_k - \tilde{x}_k \sin \tilde{\alpha} + \tilde{y}_k \cos \tilde{\alpha}, \quad \forall_{k=1, \dots, K} \\ \tilde{z}_{\text{UAV}k} &= z_k - (\tilde{x}_k \cos \tilde{\alpha} + \tilde{y}_k \sin \tilde{\alpha}) \sin \tilde{\beta} + \tilde{z}_k \cos \tilde{\beta}, \\ &\quad \forall_{k=1, \dots, K},\end{aligned}\quad (4)$$

where $(\tilde{x}_k, \tilde{y}_k, \tilde{z}_k)$ are the real coordinates of the k th RB included in each data frame and $\tilde{\alpha}$ and $\tilde{\beta}$ are the estimated directions of the UAV flight in the azimuth (OXY) and elevation (OXZ) planes determined relative to the destination landing site (see Figure 4) by using (3); that is,

$$\begin{aligned}\tilde{\alpha} &= \text{atan} \left(\frac{1}{K} \sum_{k=1}^K \frac{\tilde{y}_k + y_k}{\tilde{x}_k + x_k} \right) \\ \tilde{\beta} &= \text{atan} \left(\frac{1}{K} \sum_{k=1}^K \frac{\tilde{z}_k + z_k}{\tilde{x}_k + x_k} \right).\end{aligned}\quad (5)$$

In [33, 35, 37], analysis of the SDF method shows that the trajectory location relative to the signal source has a significant influence on the accuracy of the positioning the object. The smallest positioning error occurs when α tends to 90° , that is, when DFS converges to zero. Therefore, to minimize the navigation error, the weighted average coordinates relative to the individual RBs are used to estimate the UAV coordinates [37]

$$\begin{aligned}(\tilde{x}_{\text{UAV}}, \tilde{y}_{\text{UAV}}, \tilde{z}_{\text{UAV}}) \\ = \left(\frac{1}{W} \sum_{k=1}^K w_k \tilde{x}_{\text{UAV}k}, \frac{1}{W} \sum_{k=1}^K w_k \tilde{y}_{\text{UAV}k}, \frac{1}{W} \sum_{k=1}^K w_k \tilde{z}_{\text{UAV}k} \right),\end{aligned}\quad (6)$$

where $w_k = 1 - |F_k(t)|$ and $W = \sum_{k=1}^K w_k$.

RB system simplicity gives us the ability to position the navigation system in any field conditions. Additionally, the minimization of spectral resources is a significant advantage of the presented method compared to the existing solutions.

3. Scenarios of Simulation Studies

The effectiveness of the developed navigation procedure determines the accuracy of determining the current UAV coordinates. In this paper, an assessment of the procedure

accuracy is made on the basis of simulation tests, of which scenarios concern the VTOL navigation. In our studies, the following assumptions and input data are accepted:

- (i) landing point is the origin of the coordinate system;
- (ii) base of the navigation system is four RBs whose positions describe the coordinates, (x_k, y_k, z_k) , where $k = 1, 2, \dots, K$, ($K = 4$); the RBs' coordinates and frequencies, f_k , of the emitted signals are presented in Table 1;
- (iii) the K th RBs emit the modulated DPSK signal, which contains information about the position coordinates and the current frequencies of the individual RBs;
- (iv) bandwidth of the DPSK signal is $B_T = 80$ kHz;
- (v) operating frequency of NR is $f_R = 2.4$ GHz and the reception bandwidth is about $B_R = 500$ kHz;
- (vi) instantaneous DFS is determined every 0.5 s on the basis of the spectral analysis duration of 1.0 s; basic frequency of the spectral analysis is 0.1 Hz;
- (vii) to analyze the Doppler curves, the time windows $T_1 = 5$ s and $T_2 = 10$ s are used;
- (viii) in electromagnetic environment, additive noise is occurred, and the level of the emitted signals at the most distant point of the trajectory provides SNR = 8 dB;
- (ix) VTOL speed is $v = 72$ km/h = 20 m/s and the flight altitude is $h_L = 50$ m.

The research focuses on the impact assessment of various factors, such as the VTOL trajectory position and the flight direction relative to the RBs. In the first stage of simulation studies, two scenarios, Sc. 1 and Sc. 2, are examined ($\alpha = 0$). Figure 4 shows their geometry in the elevation (a and b, resp., for Sc. 1 and Sc. 2) and azimuth plane (c). The second stage of the research is also based on Sc. 1 and Sc. 2. In this case, the study focuses on the impact assessment of the arrival direction in the azimuth plane, α .

We assumed that the UAV is flying using the long-range navigation (GNSS + INS) method at an average altitude of h_L . At the point L , the aircraft navigation system switches to the landing phase, that is, it begins to use the navigation procedure described in Section 2. In Sc. 1, VTOL flies at a constant altitude to the point P located above the landing site. The destination landing point O is reached by reducing the altitude in the vertical direction. In the case of Sc. 2, the flight on the line segment LP is performed at the angle β , assuming that, at the point P , UAV is at the altitude h_P . This angle is determined by

$$\beta = \text{atan} \left(\frac{h_L - h_P}{d} \right), \quad (7)$$

where d is the length of LP graphical projected on the azimuth plane (OXY). In simulation studies, we assume that $d = 400$ m and $h_P = 30$ m.

The VTOL approach direction with respect to the position of RBs has also a significant influence on the navigation

TABLE 1: Location coordinates of RBs and frequencies of signals emitted by them.

k th RB	x_k (m)	y_k (m)	$z_k = h_T$ (m)	f_k (kHz)	Notes
1	20	20	2	2,399,800	Harmonic signal
2	20	-20	2	2,399,850	Harmonic signal
3	-20	-20	2	2,399,900	Harmonic signal
4	-20	20	2	2,400,100	DPSK; RB + MR

accuracy. The results of these studies show the required location of RBs relative to the expected direction of VTOL approach. In simulation studies, Sc. 1 and Sc. 2 are also used to evaluate the impact of the VTOL approach direction on the navigation error.

4. Results

The simulation studies involve the implementation of procedures such as the generation of RBs' harmonic signals, generation of environmental noise, and determination of the VTOL coordinates based on the estimated DFSS. Each generated harmonic signal contains DFS that results from the VTOL position relative to corresponding RB. This DFS is determined on the basis of (1). The generated environmental noise is a normal band signal, whose dynamics relative to the harmonic signal dynamics provides SNR = 8 dB at the range of the navigation system. The VTOL positioning procedure is performed as described in Section 2.

To evaluate the positioning error, ΔR , also called the navigation error, the following metric is used:

$$\Delta R = \sqrt{(\tilde{x}_{UAV} - x_{UAV})^2 + (\tilde{y}_{UAV} - y_{UAV})^2 + (\tilde{z}_{UAV} - z_{UAV})^2}, \quad (8)$$

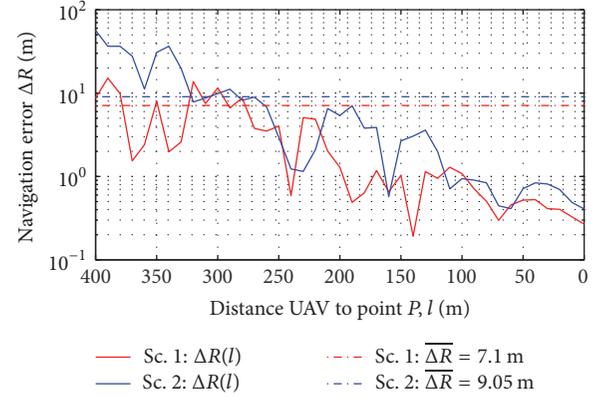
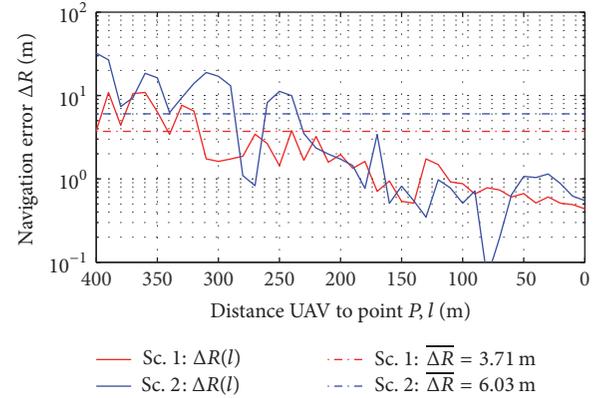
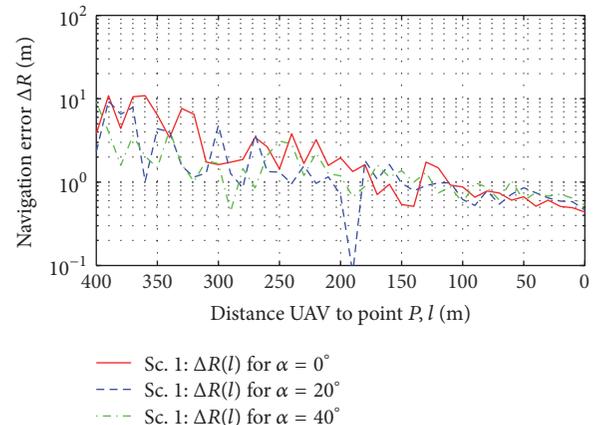
where $(\tilde{x}_{UAV}, \tilde{y}_{UAV}, \tilde{z}_{UAV})$ and $(x_{UAV}, y_{UAV}, z_{UAV})$ are the estimated and real coordinates of UAV, respectively.

The assessment of the impact of the landing trajectory position with respect to RBs is performed for $\alpha = 0^\circ$ and two analyzed temporal windows. Figures 5 and 6 present the navigation error versus the distance to the target landing site for T_1 and T_2 , respectively.

As you can see, the average navigation errors, $\overline{\Delta R}$ on line section $d = 400$ m for Sc. 1, are more than 1.3 and 1.6 times smaller compared to Sc. 2 for T_1 and T_2 , respectively. At the target landing point, these errors reach T_1 Sc. 1: $\overline{\Delta R} = 0.08$ m and Sc. 2: $\overline{\Delta R} = 0.33$ m; on the other hand T_2 Sc. 1: $\overline{\Delta R} = 0.18$ m and Sc. 2: $\overline{\Delta R} = 0.46$ m. For Sc. 2, the navigation error has a large deviation (spread), σ_Δ , which is 12 m, and for Sc. 1, this error deviation does not exceed 5 m.

The direction of the VTOL landing approach relative to the RBs' locations has also a significant impact on ΔR . Based on Sc. 1 for T_1 , simulation studies that show a variability in navigation accuracy as a function of distance from the landing site are performed. The results of these tests, that is, ΔR versus the distance from the landing point, are shown in Figure 7 for selected values $\alpha = 0^\circ$, $\alpha = 20^\circ$, and $\alpha = 40^\circ$.

It can be seen that the proper VTOL approach direction with respect to the landing site minimizes navigation

FIGURE 5: Navigation error for Sc. 1 and Sc. 2 and T_1 .FIGURE 6: Navigation error for Sc. 1 and Sc. 2 and T_2 .FIGURE 7: Navigation error for $\alpha = 0^\circ$, $\alpha = 20^\circ$, and $\alpha = 40^\circ$ - Sc. 1.

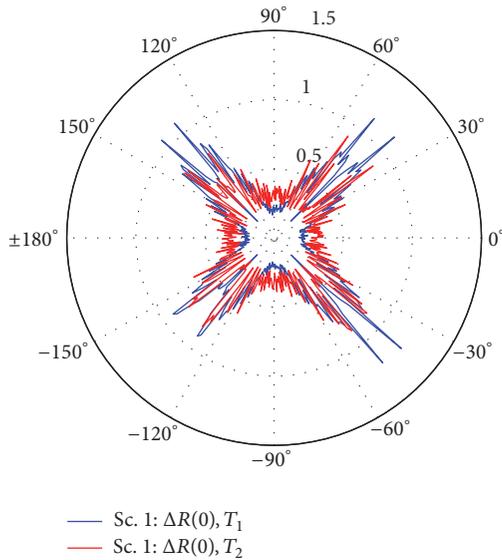


FIGURE 8: Final navigation error versus arrival direction for Sc. 1.

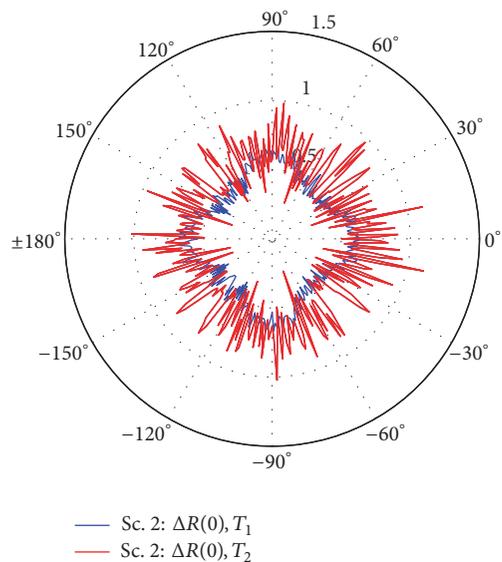


FIGURE 9: Final navigation error versus arrival direction for Sc. 2.

errors. This fact results from the comparison of mean errors, which are 7.10 m, 2.66 m, and 3.10 m, respectively, for $\alpha = \{0^\circ, 20^\circ, 40^\circ\}$.

However, the navigation error that occurs at the destination landing point is most significant. Hence, the influence assessment of the approach direction in the azimuth plane on the final error of the VTOL positioning is made on the basis of simulation research. For Sc. 1 and Sc. 2, the obtained results are presented in Figures 8 and 9, respectively.

For Sc. 1, the obtained results show that the average errors of the final position for all directions are smaller than 0.5 m for T_1 and T_2 . In the case of Sc. 2, these errors are 0.52 m and 0.63 m for T_1 and T_2 , respectively. In Figure 8, we see that there are four crucial sectors of the VTOL arrival for which ΔR can reach up to 0.90 m and 1.3 m for T_1 and T_2 ,

respectively. These cases occur when the approach direction overlap with the directions set by pairs of RBs. In the case of Sc. 2 for T_2 , we can see that a distribution of the final errors has a more uniform character. This is also evidenced by the error deviations, which for Sc. 1 are larger (0.22 m) than for Sc. 2 (0.08 m). For T_1 , the error deviations are similar, that is, 0.14 m and 0.16 m for Sc. 1 and Sc. 2, respectively.

The results of the simulation tests show the high positioning accuracy of the aircraft in the landing phase. This indicates the reasonability of the practical implementation of the developed procedure. Based on the results, it can be concluded that the smaller navigation errors were obtained for Sc. 1 and the longer analysis time, that is, for T_1 . However, given the nature of the VTOL flight, the smaller time window may be more practical. In this case, the location of the flight trajectory shown in Sc. 2 may be better, due to the independence of the navigation error from the direction of landing approach.

5. Conclusions

This paper provides the navigation procedure that enables the automatic landing approach of VTOL. The developed procedure is based on the Doppler effect and can be made using a simple short-range navigation system that is mounted anywhere at the target UAV landing. Around this place, RBs are deployed, which transmit the harmonic signals and the narrowband signal containing the information about their positions. In addition, the transmission of frequency corrections ensures that the influence of frequency instability of signal sources is minimized. This navigation system can work completely independently of GNSS and requires small spectrum resources.

In the paper, the authors evaluated the impact of the UAV trajectory, the direction of landing approach relative to RBs, and the temporal window of the signal analysis on the accuracy of the developed procedure. This assessment was made on the basis of simulation studies. The results show the high precision of the VTOL positioning. Proper selection of parameters shows that the navigation error near the destination landing point is less than 1 m. The executed research has shown that the developed procedure can contribute to the full automation of the UAV landing process, which may be important for their use in transport.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] M. Tortonesi, C. Stefanelli, E. Benvegna, K. Ford, N. Suri, and M. Linderman, "Multiple-UAV coordination and communications in tactical edge networks," *IEEE Communications Magazine*, vol. 50, no. 10, pp. 48–55, 2012.
- [2] M. A. Ma'Sum, M. K. Arrofi, G. Jati et al., "Simulation of intelligent Unmanned Aerial Vehicle (UAV) for military surveillance," in *Proceedings of the 2013 5th International Conference on*

- Advanced Computer Science and Information Systems, ICACSIS 2013*, pp. 161–166, September 2013.
- [3] D. Orfanus, E. P. De Freitas, and F. Eliassen, “Self-Organization as a Supporting Paradigm for Military UAV Relay Networks,” *IEEE Communications Letters*, vol. 20, no. 4, pp. 804–807, 2016.
 - [4] J. A. J. Berni, P. J. Zarco-Tejada, L. Suárez, and E. Fereres, “Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, pp. 722–738, 2009.
 - [5] J. Wijitdechakul, S. Sasaki, Y. Kiyoki, and C. Koopipat, “UAV-based multispectral image analysis system with semantic computing for agricultural health conditions monitoring and real-time management,” in *Proceedings of the 2016 International Electronics Symposium (IES)*, pp. 459–464, Denpasar, Indonesia, September 2016.
 - [6] P. B. Quater, F. Grimaccia, S. Leva, M. Mussetta, and M. Aghaei, “Light Unmanned Aerial Vehicles (UAVs) for cooperative inspection of PV plants,” *IEEE Journal of Photovoltaics*, vol. 4, no. 4, pp. 1107–1113, 2014.
 - [7] F. Grimaccia, M. Aghaei, M. Mussetta, S. Leva, and P. B. Quater, “Planning for PV plant performance monitoring by means of unmanned aerial systems (UAS),” *International Journal of Energy and Environmental Engineering*, vol. 6, no. 1, pp. 47–54, 2015.
 - [8] S. Daliento, A. Chouder, P. Guerriero et al., “Monitoring, diagnosis, and power forecasting for photovoltaic fields: a review,” *International Journal of Photoenergy*, vol. 2017, Article ID 1356851, 13 pages, 2017.
 - [9] S. Babak, M. Myslovych, and R. Sysak, “Module structure of UAV-based computerized systems for remote environment monitoring of energy facilities,” in *Proceedings of the 17th International Conference Computational Problems of Electrical Engineering, CPEE 2016*, September 2016.
 - [10] Y. Lu, D. Macias, Z. S. Dean, N. R. Kreger, and P. K. Wong, “A UAV-Mounted Whole Cell Biosensor System for Environmental Monitoring Applications,” *IEEE Transactions on Nanobioscience*, vol. 14, no. 8, pp. 811–817, 2015.
 - [11] D. Erdos, A. Erdos, and S. E. Watkins, “An experimental UAV system for search and rescue challenge,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 28, no. 5, pp. 32–37, 2013.
 - [12] K. A. Ghamry, M. A. Kamel, and Y. Zhang, “Cooperative forest monitoring and fire detection using a team of UAVs-UGVs,” in *Proceedings of the 2016 International Conference on Unmanned Aircraft Systems, ICUAS 2016*, pp. 1206–1211, June 2016.
 - [13] C. Yuan, Z. Liu, and Y. Zhang, “Vision-based forest fire detection in aerial images for firefighting using UAVs,” in *Proceedings of the 2016 International Conference on Unmanned Aircraft Systems, ICUAS 2016*, pp. 1200–1205, June 2016.
 - [14] S. Rathinam, P. Almeida, Z. Kim et al., “Autonomous searching and tracking of a river using an UAV,” in *Proceedings of the 2007 American Control Conference, ACC*, pp. 359–364, July 2007.
 - [15] D. Popescu, L. Ichim, and T. Caramihale, “Flood areas detection based on UAV surveillance system,” in *Proceedings of the 19th International Conference on System Theory, Control and Computing, ICSTCC 2015 - Joint Conference SINTES 19, SACCS 15, SIMSIS 19*, pp. 753–758, October 2015.
 - [16] UNICEF website, <https://www.unicef.org/>.
 - [17] AirVein, [1-Aug-], <http://www.airvein.pl/>.
 - [18] G. Hoareau, J. J. Liebenberg, J. G. Musial, and T. R. Whitman, “Package transport by unmanned aerial vehicles,” *US0068265 A1*, 2016.
 - [19] P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems*, Artech House, Boston, MA, USA, 2nd edition, 2013.
 - [20] M. S. Grewal, A. P. Andrews, and C. G. Bartone Wiley-Interscience, Hoboken, 3rd edition, 2013.
 - [21] E. D. Kaplan, *Understanding GPS: Principles and applications*, Artech House, Boston, MA, USA, 2nd edition, 2005.
 - [22] F. van Diggelen, *A-GPS: Assisted GPS, GNSS, and SBAS*, Artech House, 2009.
 - [23] S. Ward, J. Hensler, B. Alsalam, and L. F. Gonzalez, “Autonomous UAVs wildlife detection using thermal imaging, predictive navigation and computer vision,” in *Proceedings of the 2016 IEEE Aerospace Conference, AERO 2016*, March 2016.
 - [24] F. S. Leira, T. A. Johansen, and T. I. Fossen, “Automatic detection, classification and tracking of objects in the ocean surface from UAVs using a thermal camera,” in *Proceedings of the 2015 IEEE Aerospace Conference, AERO 2015*, March 2015.
 - [25] H. Sheng, H. Chao, C. Coopmans, J. Han, M. McKee, and Y. Chen, “Low-cost UAV-based thermal infrared remote sensing: Platform, calibration and applications,” in *Proceedings of the 2010 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, MESA 2010*, pp. 38–43, July 2010.
 - [26] C. Leśnik, P. Serafin, and A. Kawalec, “Azimuth ambiguity suppression in SAR images using Doppler-sensitive signals,” *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 63, no. 1, pp. 221–227, 2015.
 - [27] B. Dawidowicz, A. Gadoś, A. Gorzelańczyk et al., “First polish SAR trials,” *IEE Proceedings: Radar, Sonar and Navigation*, vol. 153, no. 2, pp. 135–143, 2006.
 - [28] M. Caris, S. Stanko, M. Malanowski et al., “Mm-Wave SAR demonstrator as a test bed for advanced solutions in microwave imaging,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 29, no. 7, pp. 8–15, 2014.
 - [29] L. J. Ippolito, “Satellite Communications Systems Engineering: Atmospheric Effects, Satellite Link Design and System Performance,” *Satellite Communications Systems Engineering: Atmospheric Effects, Satellite Link Design and System Performance*, pp. 1–376, 2008.
 - [30] *Instrument flying handbook*, CreateSpace Independent Publishing Platform, 2013.
 - [31] M. Kayton and W. R. Fried, *Avionics Navigation Systems*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 1997.
 - [32] J. Rafa and C. Ziolkowski, “Influence of transmitter motion on received signal parameters – Analysis of the Doppler effect,” *Wave Motion*, vol. 45, no. 3, pp. 178–190, 2008.
 - [33] J. M. Kelner, *Analizadopplerowskiej methodlokalizacjizródelemisjifalradiowych, rozprawadoktorska (Analysis of the Doppler location method of the radio waves emission sources*, Military University of Technology, 2010.
 - [34] P. Gajewski, C. Ziolkowski, and J. M. Kelner, “Using SDF method for simultaneous location of multiple radio transmitters,” in *Proceedings of the 2012 19th International Conference on Microwaves, Radar and Wireless Communications, MIKON 2012*, pp. 634–637, May 2012.
 - [35] A. T. de Hoop, “Comments on: J. Rafa, C. Ziolkowski, “Influence of transmitter motion on received signal parameters—an analysis of the Doppler effect”, *Wave Motion* 45 (2008) 178–190 [MR2449241],” *Wave Motion. An International Journal Reporting Research on Wave Phenomena*, vol. 46, no. 1, pp. 89–91, 2009.

- [36] N. A. Lehtomaki, T. C. Poling, and D. A. Schuyler, "Single platform Doppler geolocation," Article ID 130794, pp. 03–2015, 2015.
- [37] J. M. Kelner, C. Ziółkowski, and L. Nowosielski, "Local navigation system for VTOLs used on the vessels," in *Proceedings of the IEEE/ION Position, Location and Navigation Symposium, PLANS 2016*, pp. 415–421, April 2016.
- [38] J. M. Kelner, C. Ziółkowski, L. Nowosielski, and M. Wnuk, "Reserve navigation system for ships based on coastal radio beacons," in *Proceedings of the IEEE/ION Position, Location and Navigation Symposium, PLANS 2016*, pp. 393–402, April 2016.
- [39] J. M. Kelner, C. Ziółkowski, and P. Marszalek, "Influence of the frequency stability on the emitter position in SDF method," in *Proceedings of the 2016 International Conference on Military Communications and Information Systems, ICMCIS 2016*, May 2016.
- [40] J. M. Kelner and C. Ziółkowski, "The use of SDF technology to BPSK and QPSK emission sources' location," *Przegląd Elektrotechniczny*, vol. 91, no. 3, pp. 61–65, 2015.

Research Article

Car Detection from Low-Altitude UAV Imagery with the Faster R-CNN

Yongzheng Xu,^{1,2} Guizhen Yu,^{1,2} Yunpeng Wang,^{1,2} Xinkai Wu,^{1,2} and Yalong Ma^{1,2}

¹Beijing Key Laboratory for Cooperative Vehicle Infrastructure Systems and Safety Control,
School of Transportation Science and Engineering, Beihang University, Beijing 100191, China

²Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, SiPaiLou No. 2, Nanjing 210096, China

Correspondence should be addressed to Guizhen Yu; yugz@buaa.edu.cn

Received 2 December 2016; Revised 12 July 2017; Accepted 25 July 2017; Published 29 August 2017

Academic Editor: Pascal Vasseur

Copyright © 2017 Yongzheng Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

UAV based traffic monitoring holds distinct advantages over traditional traffic sensors, such as loop detectors, as UAVs have higher mobility, wider field of view, and less impact on the observed traffic. For traffic monitoring from UAV images, the essential but challenging task is vehicle detection. This paper extends the framework of Faster R-CNN for car detection from low-altitude UAV imagery captured over signalized intersections. Experimental results show that Faster R-CNN can achieve promising car detection results compared with other methods. Our tests further demonstrate that Faster R-CNN is robust to illumination changes and cars' in-plane rotation. Besides, the detection speed of Faster R-CNN is insensitive to the detection load, that is, the number of detected cars in a frame; therefore, the detection speed is almost constant for each frame. In addition, our tests show that Faster R-CNN holds great potential for parking lot car detection. This paper tries to guide the readers to choose the best vehicle detection framework according to their applications. Future research will be focusing on expanding the current framework to detect other transportation modes such as buses, trucks, motorcycles, and bicycles.

1. Introduction

Unmanned aerial vehicles (UAVs) hold promise of great value for transportation research, particularly for traffic data collection (e.g., [1–5]). UAVs have many advantages over ground based traffic sensors [2]: great maneuverability and mobility, wide field of view, and zero impact on ground traffic. Due to the high cost and challenges of image processing, UAVs have not been extensively exploited for transportation research. However, with the recent price drop of off-the-shelf UAV products and widely applications of surveillance video technologies, UAVs are becoming more prominent in transportation safety, planning, engineering, and operations.

For UAV based applications in traffic monitoring, one essential task is vehicle detection. This task has become challenging due to the following reasons: varying illumination conditions, background motions due to UAV movements, complicated scenes, and different traffic conditions (congested or noncongested). Many traditional techniques, such as background subtraction [6], frame difference [7], optical

flow [8], and so on, can only achieve low accuracy; and some methods, such as frame difference and optical flow, can only detect moving vehicles. In order to improve detection accuracy and efficiency, many object detection schemes have been applied for vehicle detection from UAV images, including Viola-Jones (V-J) object detection scheme [9], the linear support machine (SVM) with histogram of orientated gradient (HOG) features [10] (SVM + HOG), and Discriminatively Trained Part Based Models (DPM) [11]. Generally, these object detection schemes are less sensitive to image noise and complex scenarios therefore are more robust and efficient for vehicle detection. However, most of these methods are sensitive to objects' in-plane rotation; that is, only objects in one particular orientation can be detected. Furthermore, many methods, like V-J, are sensitive to illumination changes.

In recent years, convolutional neural network (CNN) has shown impressive performance on object classification and detection. The structure of CNN was first proposed by LeCun et al. [12]. As a feature learning architecture, CNN contains convolution and max-pooling layers. Each convolutional

layer of CNN generates feature maps using several different convolution kernels on the local receptive fields from the preceding layer. The output layer in the CNN combines the extracted features for classification. By applying down-pooling, the sizes of feature map can be decreased and the extracted features become more complex and global. Many studies [13–15] have shown that CNN can achieve promising performance in object detection and classifications.

However, directly combining CNN with sliding window strategy has difficulties to precisely localize objects [16, 17]. To address above issues, region-based CNN, that is, R-CNN [18], SPPnet [19], and Fast-R-CNN have, been proposed to improve object detection performance. But the region proposal generation step consumes too much computation time. Therefore, Ren et al. further improved Fast R-CNN [20] and developed the Faster R-CNN [21], which achieves state-of-the-date object detection accuracy with real-time detection speed. Inspired by the success of Faster R-CNN [21] in object detection, this research aims to apply Faster R-CNN [21] for vehicle detection from UAV imagery.

The rest of the paper is organized as follows: Section 2 briefly reviews some related work about vehicle detection with CNN from UAV images, followed by the methodological details of the Faster R-CNN [21] in Section 3. Section 4 presents a comprehensive evaluation of the Faster R-CNN for car detection. Section 5 presents a discussion on some key characteristics of Faster R-CNN. Finally, Section 6 concludes this paper with some remarks.

2. Related Work

A large amount of research has been performed on vehicle detection over the years. Here we only focus on vehicle detection with CNN from UAV images. Some of the most related work is reviewed here.

Pérez et al. [22] developed a traditional object detection framework based on the sliding window strategy with a classifier. This paper designed a simple CNN network instead of using traditional classifiers (SVM, Boosted Trees, etc.). As the sliding window strategy is time-consuming when handling multiscale objects detection, the framework of [22] is time-consuming for vehicle detection from UAV images.

Ammour et al. [23] proposed a two-stage car detection method, including candidate regions extraction and classification stage. In the candidate regions extraction stage, the authors employed the mean-shift algorithm [24] to segment images. Then fine-tuned VGG16 model [25] was used to extract region feature. Finally, SVM was used to classify the features into “car” and “non-car” objects. The proposed framework of [23] is similar to R-CNN [18], which was time-consuming when generating region proposals. Besides, different models should be trained for the three separate stages, which increases the complexity of [23].

Chen et al. [15] proposed a hybrid deep convolutional neural network (HDNN) for vehicle detection in satellite images to handle large-scale variance of vehicles. However, when applying HDNN for vehicle detection from satellite images, it takes about 7-8 seconds to detect one image even using Graphics Processing Unit (GPU).

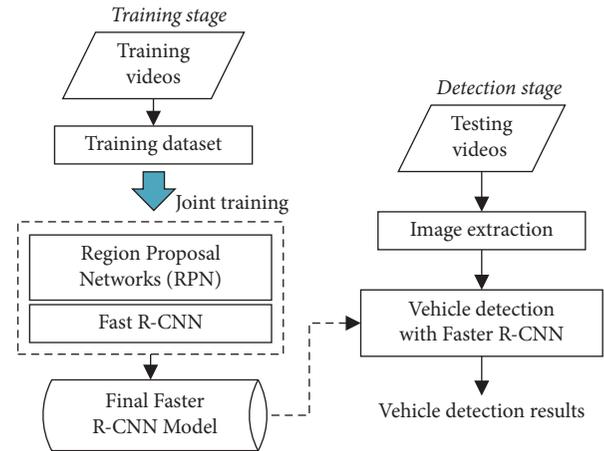


FIGURE 1: Car detection framework with the Faster R-CNN.

Inspired by the success of Faster R-CNN in both detection accuracy and detection speed, this work proposed a car detection method based on Faster R-CNN [21] to detect cars from low-altitude UAV imagery. The details of the proposed method are presented in the following section.

3. Car Detection with Faster R-CNN

Faster R-CNN [21] has achieved state-of-the-art performance for multiclass object detection in many fields (e.g., [19]). But so far no direct application of Faster R-CNN on car detection from low-altitude UAV imagery, particularly under urban environment, has been applied. This paper aims to fill this gap by proposing a framework for car detection from UAV images using Faster R-CNN, as shown in Figure 1.

3.1. Architecture of Faster R-CNN. The Faster R-CNN consists of two modules: the Regional Proposal Network (RPN) and the Fast R-CNN detector (see Figure 2). RPN is a fully convolutional network for efficiently generating region proposals with a wide range of scales and aspect ratios which will be fed into the second module. Region proposals are rectangular regions which may or may not contain candidate objects. Fast R-CNN detector, the second module, is used to refine the proposals. The RPN and Fast R-CNN detector share the same convolutional layers, allowing for joint training. The Faster R-CNN runs through the CNN only once for the entire input image and then refines object proposals. Due to the sharing of convolutional layers, it is possible to use a very deep network (e.g., VGG16 [25]) for generating high-quality object proposals. The entire architecture is a single and unified network for object detection (see Figure 2).

3.2. Fast R-CNN Detector. The Fast R-CNN detector takes multiple regions of interest (RoIs) as input. For each RoI (see Figure 2), a fixed-length feature vector is extracted by the RoI pooling layer from the convolutional layer. Each feature vector is fed into a sequence of fully connected (FC) layers. The final outputs of the detector through the softmax layer and the bounding-box regressor layer include (1) softmax

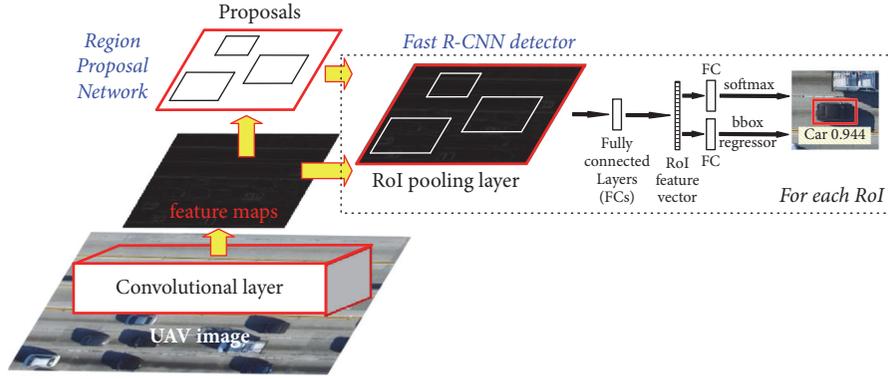


FIGURE 2: The architecture of Faster R-CNN, from [20, 21].

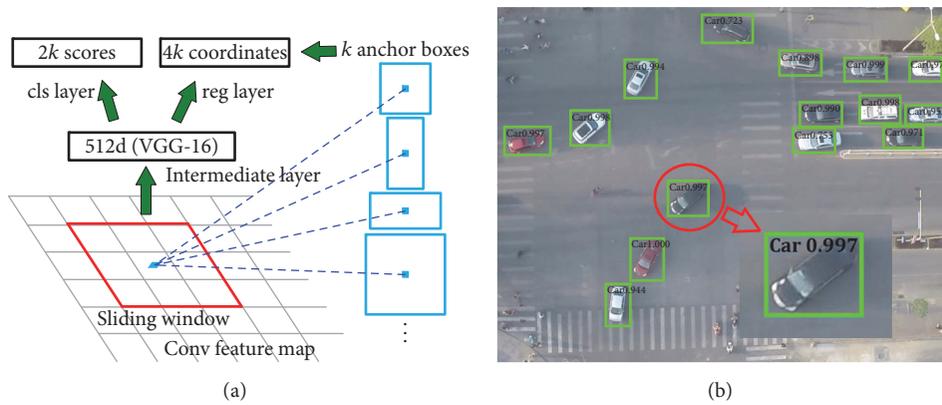


FIGURE 3: (a) Region Proposal Network (RPN), from [21]. (b) Car detection using RPN proposals on our UAV image.

probabilities which estimate over K object classes plus the “background” class and (2) related bounding-box (bbox) values. In this research, the value of K is 1, namely, the object classes only contain one object “passenger car” plus the “background” class.

3.3. Region Proposal Networks and Joint Training. When using RPN to predict car proposals from UAV images, the RPN takes a UAV image as input and outputs a set of rectangular car proposals (i.e., bounding boxes), each with an objectness score. In this paper, the VGG-16 model [25], which has 13 shareable convolutional layers, was used as the Faster-RCNN convolutional backend.

The RPN utilizes sliding windows over the convolutional feature map output by the last shared convolutional layer to generate rectangular region proposals for each position (see Figure 3(a)). A $n \times n$ spatial window (filter) was convolved with the input convolutional feature map. Then each sliding window is projected to a lower-dimensional feature (512-d for VGG-16), by convolving with two 1 by 1 filters, respectively, for a box-regression layer (reg) and a box-classification layer (cls). For each sliding window location, k possible proposals (i.e., anchors in [21]) were generated in the cls layer. For the reg layer, $4k$ outputs were generated to encode the coordinates of k bounding boxes. Meanwhile, $2k$ objectness scores were

output in the cls layer to estimate probability whether each proposal contains a car or a non-car object (see Figure 3(b)).

As many proposals highly overlap with each other, nonmaximum suppression (NMS) was applied to merge proposals that have high intersection-over-union (IoU). After NMS, the remaining proposals were ranked based on the object probability score, and only the top N proposals are used for detection.

For training RPNs, each proposal is assigned a binary class label which indicates whether the proposal is an object (i.e., car) or just background. A positive training example is designated if the proposal overlaps with a ground-truth box with an IoU more than a predefined threshold (0.7 in [21]), or if it has the highest IoU with a ground-truth.

A proposal will be assigned as a negative example if its maximum IoU is lower than the predefined threshold (0.3 in [21]) for all ground-truth boxes. Following the multitask loss in Fast R-CNN network [20], the RPN is trained by a multitask loss, which is defined as

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{\text{cls}}} \sum_i L_{\text{cls}}(p_i, p_i^*) + \lambda \frac{1}{N_{\text{reg}}} \sum_i p_i^* L_{\text{reg}}(t_i, t_i^*), \quad (1)$$

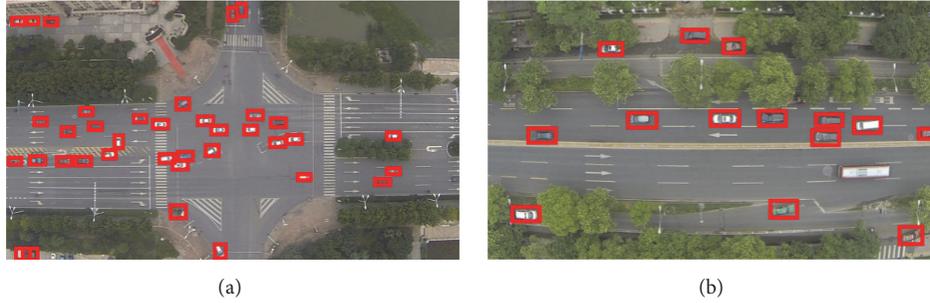


FIGURE 4: Car detection. (a) Signalized intersection; (b) arterial road.

where i is the index of an anchor and p_i is the predicted probability of anchor i being an object. The ground-truth label p_i^* is 1 if the anchor is positive and 0 if the anchor is negative. The multitask loss has two parts, a classification component L_{cls} and a regression component L_{reg} . In (1), t_i is a vector representing the 4 parameterized coordinates of the predicted bounding-box; and t_i^* is the vector of the ground-truth box associated with a positive anchor. These two terms are normalized by N_{cls} and N_{reg} and weighted by a balancing parameter λ . In the released code [26], the cls term in (1) is normalized by the minibatch size (i.e., $N_{\text{cls}} = 256$), the reg term is normalized by the number of anchor locations (i.e., $N_{\text{reg}} \sim 2,400$), and λ is set as 10.

Bounding-box regression is to find the best nearby ground-truth box of an anchor box. The parameterization of the 4 coordinates of an anchor is described as follows:

$$\begin{aligned}
 t_x &= \frac{(x - x_a)}{w_a}, \\
 t_y &= \frac{(y - y_a)}{h_a}, \\
 t_w &= \log\left(\frac{w}{w_a}\right), \\
 t_h &= \log\left(\frac{h}{h_a}\right), \\
 t_x^* &= \frac{(x^* - x_a)}{w_a}, \\
 t_y^* &= \frac{(y^* - y_a)}{h_a}, \\
 t_w^* &= \log\left(\frac{w^*}{w_a}\right), \\
 t_h^* &= \log\left(\frac{h^*}{h_a}\right),
 \end{aligned} \tag{2}$$

where x , y , w , and h denote the bounding-box's center coordinates, width, and height, respectively. x , x_a , and x^* are for the predicted box, anchor box, and ground-truth box, respectively. Similar definitions apply for y , w , and h .

The bounding-box regression is achieved by using features with the same spatial size on the feature maps. A set of k bounding-box regressors are trained to adapt for varying size.

Since the RPN and Fast R-CNN detector can share the same convolutional layers, these two networks can be trained jointly to learn a unified network through the following 4-step training algorithm: first, training the RPN as described above; second, training the detector network using proposals generated by the RPN trained in the first step; third, initializing RPN training by the detector network but only train the RPN specific layers; and finally, training the detector network using the new RPN's proposals. Figure 4 shows two screenshots of car detection with the Faster R-CNN.

4. Experiments

4.1. Data Set Descriptions. The airborne platform used in this research is a DJI Phantom 2 quadcopter integrated with a 3-axis stabilized gimbal (see Figure 5).

Videos are collected by a Gopro Hero Black Edition 3 camera mounted on the UAV. The resolution of the videos is 1920×1080 and the frame rate is 24 frames per second (f/s). The stabilized gimbal is used to stabilize the videos and eliminate video jitters caused by UAV therefore greatly reducing the impact from external factors, such as wind. In addition, an On-Screen Display (OSD), an image transmission module, and a video monitor are installed in the system for data transmission and airborne flying status monitoring and control.

A UAV image dataset is built for training and testing the proposed car detection framework. For training video collection, we followed the following two key suggestions: (1) collecting videos with cars of different orientations; (2) collecting videos with cars of a wide range of scales and aspect ratios. To collect videos with cars of different orientations, UAV videos from signalized intersections were recorded; since cars at intersections have different orientations while making turning. To collect videos covering cars of a wide range of scales and aspect ratio, UAV videos at different flight height, ranging from 100 m to 150 m, were recorded. In this work, UAV videos were collected from two different signalized intersections. For each intersection, videos 1-hour long were captured. Totally, videos two hours long were collected for building the training and testing datasets.

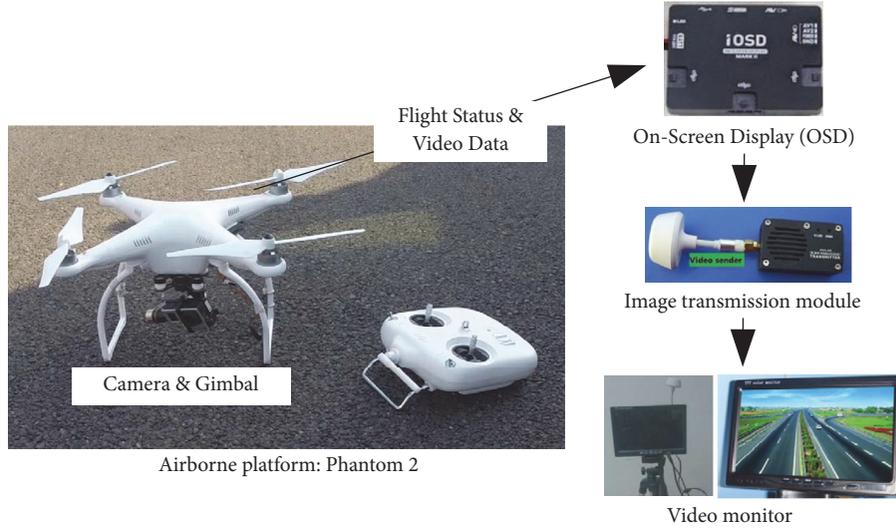


FIGURE 5: UAV system architecture.

In our experiment, the training and testing datasets include 400 and 100 images, respectively. Note the images for training and testing are collected from different UAV videos. The whole dataset contains 400 images with 12,240 samples for training and 100 images with 3,115 samples for testing. Note the samples for training and testing are collected from different UAV videos. Training and testing samples are annotated using the tool LabelImg [27]. During the testing and training stage, in order to avoid the same car in consecutive frames being used too many times, images were extracted every 10 seconds from UAV videos.

4.2. Training Faster R-CNN Model. Faster-RCNN was powerful in multiclass object detection. But in this research, we only trained the Faster-RCNN model for passenger cars. Particularly, we applied the VGG-16 model [25]. For the RPN of the Faster-RCNN, 300 RPN proposals were used. The source code of Faster R-CNN was from [26]. GPU was used during the training. The main configurations of the computer used in this research are

- (i) CPU: Intel Core i7 hexa-core 5930 K@3.5 GHz, 32 GB DDR4;
- (ii) Graphics card: Nvidia TITAN X, 12 GB GDDR5;
- (iii) Operating system: Linux (Ubuntu 14.04).

The training and detection implementation in this paper is all performed on the open source code released by the authors of Faster R-CNN [21]. The inputs for training and testing are images with the original size (1920 × 1080) without any preprocessing steps.

4.3. Performance Evaluation

4.3.1. Evaluation Indicator. The performance of car detection by Faster R-CNN is evaluated by four typical indicators:

detection speed (frames per second, f/s), Correctness, Completeness, and Quality, as defined in (3):

$$\begin{aligned} \text{Correctness} &= \frac{TP}{TP + FP}, \\ \text{Completeness} &= \frac{TP}{TP + FN}, \\ \text{Quality} &= \frac{TP}{TP + FP + FN}, \end{aligned} \quad (3)$$

where TP is the number of “true” detected cars; FP is the number of “false” detected objects which are non-car objects; and FN is the number of cars missed. In particular, Quality is considered as the strictest criterion, which contains both possible detection errors (false positives and false negatives).

4.3.2. Description of Algorithms for Comparison. To comprehensively evaluate the car detection performance of Faster R-CNN from UAV images, four other algorithms were included for comparison. The four algorithms are

- (1) ViBe, a universal background subtraction algorithm [6];
- (2) Frame difference [7];
- (3) The AdaBoost method using Haar-like features (V-J) [9];
- (4) Linear SVM classifier with HOG features (HOG + SVM) [10].

As ViBe [6] and frame difference [7] are sensitive to background motions, image registration [28] is applied first to compensate UAV motions and delete UAV video jitters. The time for image registration is included in the detection time for these two methods. The performance indicators are calculated based on the same 100 images as the testing dataset.

TABLE 1: Car detection results.

Metrics	ViBe	Frame difference	V-J	HOG + SVM	Faster R-CNN
Correctness (%)	76.64%	78.17%	84.74%	84.33%	98.43%
Completeness (%)	38.65%	39.78%	41.89%	43.18%	96.40%
Quality (%)	34.58%	35.80%	38.96%	39.97%	94.94%
Detection speed (f/s)					
CPU mode	7.42	11.83	3.38	1.45	0.018
GPU mode	N/A	N/A	20.61	6.82	2.10

Note, for ViBe and Frame Difference, the postprocessing for blob segmentation results is very important for the final car detection accuracy as blob segmentation using ViBe and Frame Difference may yield segmentation errors. In this work, two rules are designed to screen out segmentation errors: (1) the area of a detected blob is too large (2 times larger than that of a normal passenger car) or too small (smaller than 1/2 of a normal passenger car); (2) the aspect ratio of the minimum enclosing rectangle of a detected blob is larger than 2. Note, the area of the normal passenger car was obtained by human. If any of the two rules is met, the detected blob will be screened out as segmentation errors.

The V-J [9] and HOG + SVM [10] methods are trained on 12,240 positive samples and 30,000 negative samples. These 12,240 samples only contain cars orientated in the horizontal direction. Besides, all positive samples are normalized to a compressed size of 40×20 . The performance evaluations of Faster R-CNN, V-J, and HOG + SVM are run on our testing dataset (100 images, 3,115 testing samples).

4.3.3. Experiment Results. The testing results of five methods are presented in Table 1. The detection speed was an average of the 100 tested images. To comprehensively evaluate the performance of different algorithms on both CPU and GPU architectures, detection speeds for V-J, HOG + SVM, and Faster R-CNN were tested on the i7 CPU and the high-end GPU, respectively.

The results show that Faster R-CNN achieved the best *Quality* (94.94%) compared with other four methods. ViBe and Frame Difference achieved fast detection speed under CPU mode but with very low *Completeness*. The reason is that many stopped cars (such as cars waiting at the traffic light) are recognized as background objects, therefore generating many false negatives and leading to a low *Completeness*. Only when those stopped cars run again could they be detected. As many moving non-car objects (such as tricycles and moving pedestrians) lead to false positives, the *Correctness* of those two methods is low (76.64% and 78.17%, resp.).

Although the two object detection schemes V-J and HOG + SVM are nonsensitive to image background motions compared with ViBe and Frame Difference, the *Completeness* of these two methods is also as low as 41.61% and 42.89%, respectively, which is only slightly higher than that of ViBe and Frame Difference. The reason, as mentioned in Section 1, is that both V-J and HOG + SVM are sensitive to objects' in-plane rotation. Only cars in the same orientation with the positive training samples could be detected. In this paper,

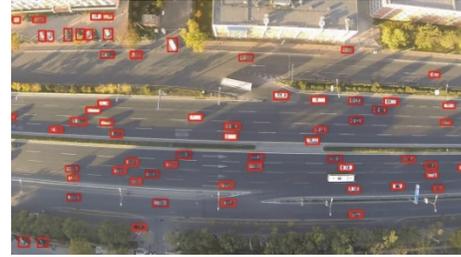


FIGURE 6: Car detection under illumination changing condition using Faster R-CNN.

only cars in the horizontal direction can be detected. A sensitivity analysis of the impact of cars' in-plane rotations has been provided in Discussion.

The method of Faster R-CNN achieved the best performance (*Quality*, 94.94%) among all five methods. As Faster R-CNN can intelligently learn the information of orientation, aspect ratio, and scale during training, this method is not sensitive to cars' in-plane rotation and scale variations. Therefore, Faster R-CNN achieves high *Correctness* (98.43%) and *Completeness* (96.40%).

Though Faster R-CNN achieved 2.1 f/s under GPU mode, which is slower than other methods, 2.1 f/s can still satisfy real-time applications.

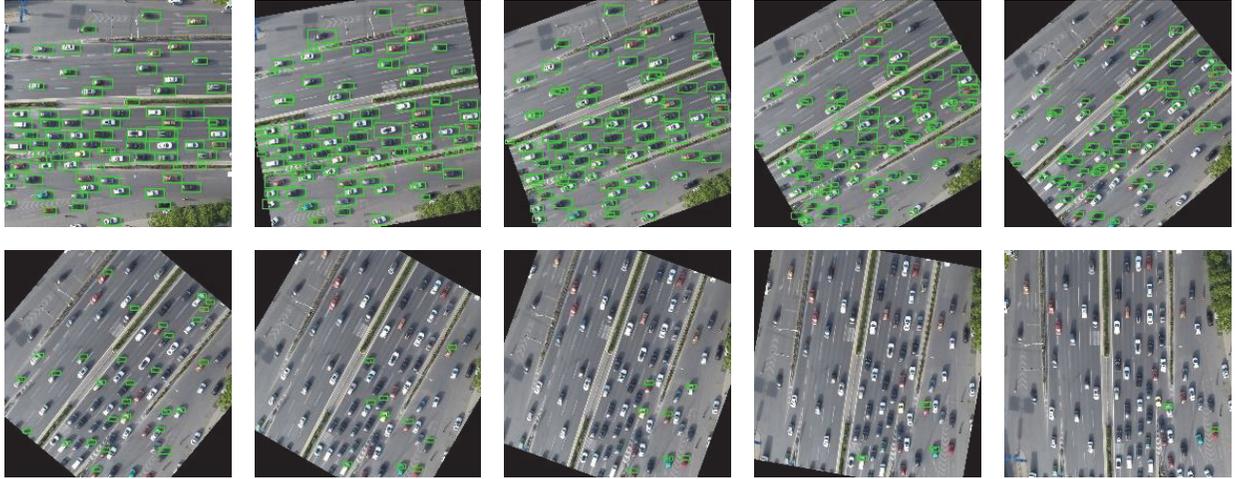
5. Discussion

5.1. Robustness to Illumination Changing Condition. For car detection from UAV videos, one most challenging issue is the illumination changing. Our testing datasets (100 images, 3,115 testing samples) do not contain cars in such scenes; for example, cars travel from an illumination (or shadowed) area to a shadowed (or illumination) area. Therefore, we further conducted an experiment using a 10 min long video captured under illumination changing condition to evaluate the performance of the Faster R-CNN (see Figure 6).

The testing results are highlighted in Table 2. The results show that Faster R-CNN achieved a *Completeness* of 94.16%, which is slightly lower than that in Table 1 (96.40%), due to the oversaturation of the image sensor under strong illumination condition. The *Correctness* of Faster R-CNN is 98.26%. The results shown in Table 2 confirm that illumination changing condition has little impact on the accuracy of vehicle detection using Faster R-CNN.

TABLE 2: Vehicle detection under illumination changing condition.

Metrics	ViBe	Frame difference	V-J	HOG + SVM	Faster R-CNN
Correctness (%)	81.91%	80.15%	87.27%	88.45%	98.26%
Completeness (%)	67.90%	64.69%	81.36%	82.38%	94.16%
Quality (%)	59.05%	55.76%	72.73%	74.38%	92.61%

FIGURE 7: Car detection by HOG + SVM using image dataset which contain cars orientated in different orientations (0° , 10° , 20° , 30° , 40° , 50° , 60° , 70° , 80° , and 90°).

The methods of ViBe and Frame Difference achieved higher *Quality* than that shown in Table 1. That is because this test scene is an arterial road (see Figure 6), where most cars were running fast along the road; therefore these moving cars can be easily detected by ViBe and Frame Difference. However, many black cars that have similar color as the road surface and cars under strong illuminations could not be detected; therefore, the *Completeness* of ViBe and Frame Difference are still low (67.90% and 64.69%, resp.). The V-J and HOG + SVM methods achieved higher *Completeness* (81.36% and 82.38%, resp.) than those shown in Table 1 (41.61% and 42.89%, resp.); because most of these cars in this testing scene (see Figure 6) are orientated in the horizontal direction; thus these vehicles can be successfully detected by V-J and HOG + SVM. However, the *Completeness* of these two methods is significantly lower than that of the Faster R-CNN. As argued by some research [29], methods like the V-J method are sensitive to lighting conditions.

5.2. Sensitivity to Vehicles' In-Plane Rotation. As mentioned in Section 1, methods like V-J and HOG + SVM are sensitive to vehicles' in-plane rotation. As the vehicle orientations are generally unknown in UAV images, the detection rates (*Completeness*) of different methods may be affected significantly by the vehicles' in-plane rotation.

To analyze the sensitivity of different methods to vehicles' in-plane rotation, experiments are conducted based on dataset which contains vehicles orientated in different directions (see Figure 7). The dataset contains 5 groups of images; each group contains 19 images which orientated in

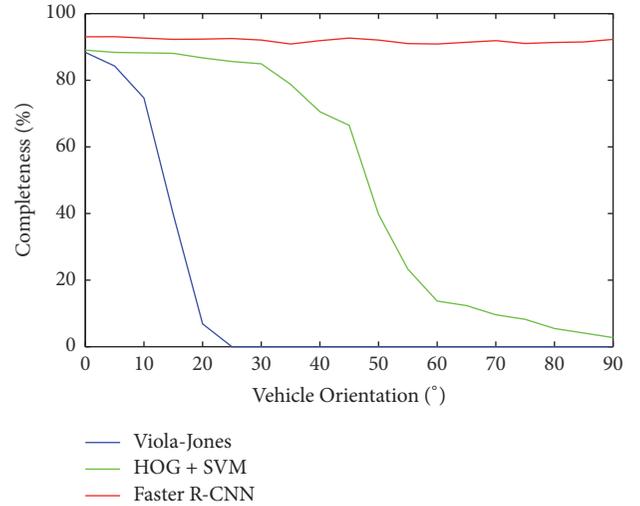


FIGURE 8: Sensitivity to vehicles' in-plane rotation.

different orientations as $0^\circ, 5^\circ, 10^\circ, \dots, 85^\circ, 90^\circ$ at an interval of 5° .

From Figure 8 we can see that the *Completeness* of the V-J downgrades significantly as the vehicles' orientation exceeds 10 degrees. Compared to V-J, HOG + SVM is less sensitive to vehicles' in-plane rotation, but the *Completeness* of HOG + SVM still downgrades significantly when the vehicles' orientation exceeds about 45 degrees.

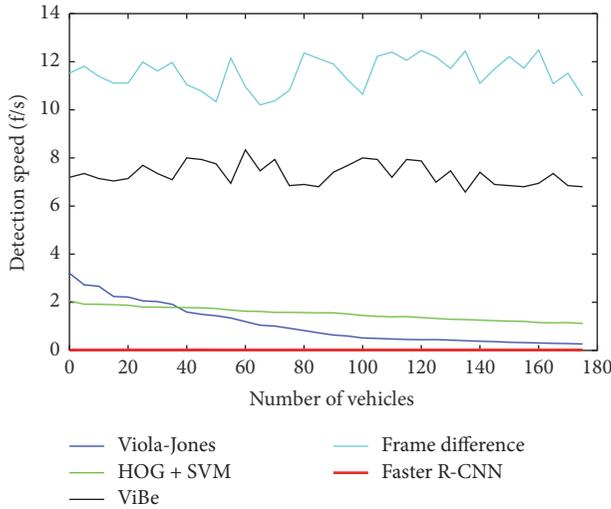


FIGURE 9: Sensitivity of detection speed to different detection load (tested on i7 CPU).

Compared with V-J and HOG + SVM, Faster R-CNN is insensitive to vehicles' in-plane rotation (the red curve in Figure 8). The reason is that the Faster R-CNN can automatically learn the information of orientation, aspect ratio, and scale of vehicles from vehicle training samples during the training. Therefore, Faster R-CNN is insensitive to vehicles' in-plane rotation.

5.3. Sensitivity of Detection Speed to Different Detection Load. Detection speed is crucial for real-time applications. Detection speed can be easily affected by many factors, such as the detection load (i.e., the number of detected vehicles in one image), hardware configuration, and video resolution. Among these factors, the most important factor is detection load.

To comprehensively explore the speed characteristic of Faster R-CNN, experiments on images which contain different number of detected vehicles have been conducted (see Figure 9). Other four methods are also included for comparison. To fairly evaluate the detection speed of different algorithms on different architectures, the speed tests are performed on the i7 CPU and the high-end GPU, respectively. We explored the detection speed on i7 CPU for all five methods (see Figure 9) and explored the detection speed on GPU for VJ, HOG + SVM, and Faster R-CNN (see Figure 10).

From Figure 9 we can see that the detection speeds of V-J and HOG + SVM are monotonically decreasing with the increase of the number of detected vehicles. The V-J method presents a higher descending rate than HOG + SVM as the number of detected vehicles increases. The speed curves of ViBe and Frame Difference are unsmooth, but we can see that the increase of the number of detected vehicles has little influence on the detection speed of the two methods.

The detection speed of Faster R-CNN was very slow under CPU mode (see Figure 9). Under GPU mode (see Figure 10), the detection speed of Faster R-CNN was about 2 f/s. From Figures 9 and 10, we can find that the Faster R-CNN holds

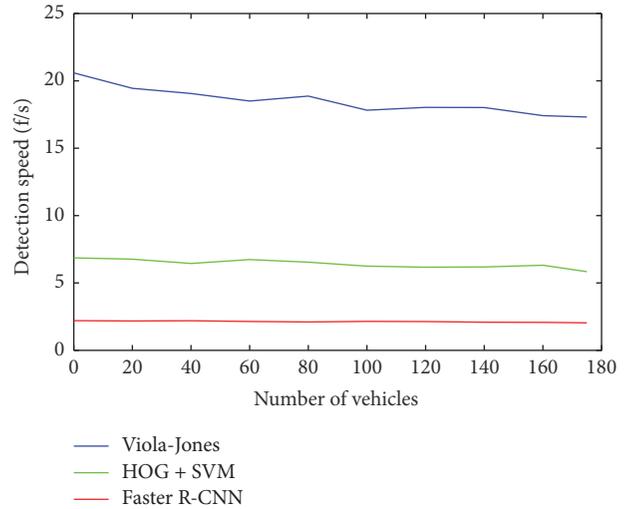


FIGURE 10: Sensitivity of detection speed to different detection load (tested on GPU).

TABLE 3: Training cost.

Metrics	V-J	HOG + SVM	Faster R-CNN
Training time	6.8 days	5 minutes	21 hours

similar speed characteristic as the ViBe and Frame Difference but with a smooth speed curve. The detection load almost has no influence on the detection speed of Faster R-CNN. The reason is that when detecting vehicles using Faster R-CNN, the method is applied on the entire image. In the proposal regions generation stage, 2000 RPN proposals are generated from the original image [21]. The top-300 ranked proposal regions are fed into the Fast R-CNN [20] to check whether the proposal region contains one car. The computational cost is almost the same for each frame; therefore, the detection speed of Faster R-CNN is nearly insensitive to detection load.

5.4. Training Cost Comparison. When applying the Faster R-CNN for vehicle detection, one important issue that should be considered is the computational cost of training procedures. As the training samples may change, it is necessary to efficiently update the Faster R-CNN model to satisfy the requirement of vehicle detection. The training costs of three different methods are shown in Table 3. Because the open source code of Faster R-CNN can only support training function under GPU mode, only training time under GPU mode was provided. For V-J and HOG + SVM, as the open source code only supports CPU mode, only training time under CPU mode was provided.

As shown in Table 3, the AdaBoost method using Haar-like features (V-J) trained on 12,240 positive samples and 30,000 negative samples takes about 6.8 days. The training procedure was only run on CPU without parallel computing or other acceleration schemes. The linear SVM classifier with HOG features (HOG + SVM) shares the fastest training speed among all the three methods. It only takes 5 minutes on the same training set as the V-J method. Although HOG + SVM

has the fastest training speed, its detection performance is significantly lower than that of Faster R-CNN (see Table 1). The training of Faster R-CNN takes about 21 hours to complete. For practical applications, 21 hours is acceptable, as the annotation of training samples may take several days. For example, in this paper, the annotation of the whole dataset (12,240 training samples and 3,115 testing samples, totally 500 images) using the tool LabelImg [27] costs 4 days by two research fellows.

6. Concluding Remarks

Inspired by the impressive performance achieved by Faster R-CNN on object detection, this research applied this method for passenger car detection from low-altitude UAV imagery. The experimental results demonstrate that Faster R-CNN can achieve highest Completeness (96.40%) and Correctness (98.43%) with real-time detection speed (2.10 f/s), compared with four other popular vehicle detection methods.

Our tests further demonstrate that Faster R-CNN is robust to illumination changing and cars' in-plane rotation; therefore, Faster R-CNN can be applied for vehicle detection from both static and moving UAV platforms. Besides, the detection speed of Faster R-CNN is insensitive to the detection load (i.e., the number of detected vehicles). The training cost of Faster R-CNN network is about 21 hours, which is acceptable for practical applications.

It should be emphasized that this research provided a rich comparison of different vehicle detection techniques which covers a lot of aspects of object detection challenges that are usually partially covered in object detection papers: detection rate without in-plane rotation, sensitivity to in-plane rotation, detection speed, and sensitivity to the number of vehicle in the image as well as the training cost. This paper tries to guide the readers to choose the best framework according to their applications.

However, due to the lack of enough training samples, this research only tested the Faster-RCNN networks for passenger cars. Future research will expand this method for the detection of other transportation modes such as buses, trucks, motorcycles, and bicycles.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is partially supported by the Fundamental Research Funds for the Central Universities and partially by the National Science Foundation of China under Grant nos. 61371076 and 51278021.

References

- [1] A. Angel, M. Hickman, P. Mirchandani, and D. Chandnani, "Methods of analyzing traffic imagery collected from Aerial

platforms," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 2, pp. 99–107, 2003.

- [2] M. Hickman and P. Mirchandani, "Airborne traffic flow data and traffic management," in *Proceedings of the 75 Years of the Fundamental Diagram for Traffic Flow Theory: Greenshields Symposium*, pp. 121–132, 2008.
- [3] B. Coifman, M. McCord, R. G. Mishalani, and K. Redmill, *Surface Transportation Surveillance from Unmanned Aerial Vehicles*, 2004.
- [4] J. Leitloff, D. Rosenbaum, F. Kurz, O. Meynberg, and P. Reinartz, "An operational system for estimating road traffic information from aerial images," *Remote Sensing*, vol. 6, no. 11, pp. 11315–11341, 2014.
- [5] B. Coifman, M. McCord, R. Mishalani, M. Iswalt, and Y. Ji, "Roadway traffic monitoring from an unmanned aerial vehicle," *IEEE Proceedings-Intelligent Transport Systems*, vol. 153, no. 1, pp. 11–20, 2006.
- [6] O. Barnich and M. van Droogenbroeck, "ViBe: a universal background subtraction algorithm for video sequences," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1709–1724, 2011.
- [7] A. C. Shastry and R. A. Schowengerdt, "Airborne video registration and traffic-flow parameter estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 4, pp. 391–405, 2005.
- [8] H. Yalcin, M. Hebert, R. Collins, and M. J. Black, "A flow-based approach to vehicle detection and background mosaicking in airborne video," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR '05)*, p. 1202, San Diego, CA, USA, June 2005.
- [9] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 511–518, December 2001.
- [10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 886–893, June 2005.
- [11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.
- [12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.
- [13] Y. Huang, R. Wu, Y. Sun, W. Wang, and X. Ding, "Vehicle logo recognition system based on convolutional neural networks with a pretraining strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1951–1960, 2015.
- [14] J. Tang, C. Deng, G.-B. Huang, and B. Zhao, "Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 3, pp. 1174–1185, 2015.
- [15] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 10, pp. 1797–1801, 2014.
- [16] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. Lecun, "Pedestrian detection with unsupervised multi-stage feature learning," in *Proceedings of the 26th IEEE Conference on Computer Vision*

- and Pattern Recognition (CVPR '13)*, pp. 3626–3633, IEEE, June 2013.
- [17] R. Vaillant, C. Monrocq, and Y. Le Cun, “Original approach for the localization of objects in images,” *IEE Proceedings: Vision, Image and Signal Processing*, vol. 141, no. 4, pp. 245–250, 1994.
 - [18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 580–587, Columbus, Oh, USA, June 2014.
 - [19] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
 - [20] R. Girshick, “Fast R-CNN,” in *Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV '15)*, pp. 1440–1448, December 2015.
 - [21] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
 - [22] A. Pérez, P. Chamoso, V. Parra, and A. J. Sánchez, “Ground vehicle detection through aerial images taken by a UAV,” in *Proceedings of the 17th International Conference on Information Fusion, (FUSION '14)*, July 2014.
 - [23] N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan, and M. Zuair, “Deep learning approach for car detection in UAV imagery,” *Remote Sensing*, vol. 9, no. 4, p. 312, 2017.
 - [24] D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
 - [25] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *Computer Science*, 2014.
 - [26] Faster R-CNN, 2016, <https://github.com/rbgirshick/py-faster-rcnn>.
 - [27] LabelImg, 2016, <https://github.com/tzutalin/labelImg>.
 - [28] Y. Ma, X. Wu, G. Yu, Y. Xu, and Y. Wang, “Pedestrian detection and tracking from low-resolution unmanned aerial vehicle thermal imagery,” *Sensors*, vol. 16, no. 4, p. 446, 2016.
 - [29] R. Padilla, C. F. F. Costa Filho, and M. G. F. Costa, “Evaluation of Haar Cascade Classifiers Designed for Face Detection,” in *Proceedings of the International Conference on Digital Image Processing*, 2012.

Research Article

Using UAV-Based Systems to Monitor Air Pollution in Areas with Poor Accessibility

Oscar Alvear,^{1,2} Nicola Roberto Zema,³ Enrico Natalizio,³ and Carlos T. Calafate¹

¹Department of Computer Engineering, Universitat Politècnica de València, Camino de Vera, S/N, 46022 Valencia, Spain

²Department of Electrical Engineering, Electronics and Telecommunications, Universidad de Cuenca, Av. 12 de Abril, S/N, Cuenca, Ecuador

³Laboratoire Heudiasyc, Sorbonne Universités, Université de Technologie de Compiègne, CNRS, 57 Avenue de Landshut, CS 60319, 60203 Compiègne Cedex, France

Correspondence should be addressed to Oscar Alvear; osal@doctor.upv.es

Received 8 March 2017; Revised 16 June 2017; Accepted 9 July 2017; Published 7 August 2017

Academic Editor: Guizhen Yu

Copyright © 2017 Oscar Alvear et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Air pollution monitoring has recently become an issue of utmost importance in our society. Despite the fact that crowdsensing approaches could be an adequate solution for urban areas, they cannot be implemented in rural environments. Instead, deploying a fleet of UAVs could be considered an acceptable alternative. Embracing this approach, this paper proposes the use of UAVs equipped with off-the-shelf sensors to perform air pollution monitoring tasks. These UAVs are guided by our proposed Pollution-driven UAV Control (PdUC) algorithm, which is based on a chemotaxis metaheuristic and a local particle swarm optimization strategy. Together, they allow automatically performing the monitoring of a specified area using UAVs. Experimental results show that, when using PdUC, an implicit priority guides the construction of pollution maps by focusing on areas where the pollutants' concentration is higher. This way, accurate maps can be constructed in a faster manner when compared to other strategies. The PdUC scheme is compared against various standard mobility models through simulation, showing that it achieves better performance. In particular, it is able to find the most polluted areas with more accuracy and provides a higher coverage within the time bounds defined by the UAV flight time.

1. Introduction

Industrial growth has brought unforeseen technological advances to our societies. Unfortunately, the price to pay for these advances has been an increase of air pollution worldwide, affecting both our health [1] and our lifestyle.

Air quality monitoring is relevant not only for the people living in urban areas, but also because it directly affects crops and different animals/insects in rural environments [2]. Thus, different solutions for measuring air quality should be sought for such environments.

For abovementioned reasons, environmental organizations and governmental institutions are beginning to consider the monitoring of environmental pollutants as a primary goal [3, 4].

The majority of methods used insofar to keep track of air pollution in major cities rely on fixed monitoring stations [5].

However, the use of such dedicated architectures and hardware for pollution monitoring is outmatched, in theory, by the use of *crowdsensing* [6] in areas with a high population density. Also, new ground-vehicle-based mobile sensors, which would theoretically be able to cover the same areas as the fixed solutions while employing a reduced number of agents, are emerging as a viable alternative (e.g., [7, 8]).

With respect to the widespread use of small pollution monitoring sensors embedded in mobile vehicles, the possible scenarios can be divided into two main classes:

- (i) *Urban environments*, where it is possible to embed the sensors on a wide set of vehicles like bicycles [9, 10] or cars [8].
- (ii) *Rural and industrial areas*, where vehicular traffic is scarce and limited to the main transportation arteries.

In the latter case, crowdsensing often fails to provide enough data to obtain realistic measurements having the required granularity.

Regarding crowdsensing approaches, projects like [11–13] relied on crowdsensing solutions to monitor pollution in urban areas. However, in rural and industrial zones, available options are quite more limited. In the particular case of large rural or industrial areas, a fleet of mobile vehicles could be efficiently used to cover the vast distances associated with them. Furthermore, the use of autonomous sensor carriers is even more encouraged in this case due to the following considerations:

- (i) The relative absence of civilian population to be taken care of during robotic operations.
- (ii) Stable and regulated positioning of obstacles.
- (iii) Fewer constraints concerning UAV flight laws
- (iv) Safety and security concerns, as some areas could be dangerous to access for human operators.

Since, in these environments, ground access is usually hindered and full of obstacles, the most feasible way to implement a fleet of mobile pollution monitoring robots is via Unmanned Aerial Vehicles (UAVs) [14].

Taking the aforementioned issues into consideration, in this paper we propose the use of UAVs equipped with commercial and off-the-shelf (COTS) devices and sensors to implement a service of air pollution monitoring that leverages the use of bioinspired approaches as its main control strategy. These choices allow covering a specific area automatically and enable discovering the pollution distribution of a large area by prioritizing the most polluted zones inside it.

We show that, using our chemotaxis-based approach for UAV path control, it is possible to achieve faster and more accurate estimations about the location of the most polluted areas with respect to classical area-search approaches. Our analysis also takes into account uncertainty-based considerations in the sensor sampling operations.

This paper is organized as follows: in Section 2 we refer to some related works addressing UAV-based sensing, UAV mobility models, and UAV control protocols. Sections 3 and 4 present an overview of the UAV Configuration and the UAV Control System, respectively. In Section 5, we compare our algorithm against the Billiard and Spiral mobility models via simulation. Section 6 discusses the open issues in air pollution monitoring using UAVs. Finally, in Section 7, we present the conclusions of our work.

2. Related Works

UAV-based solutions have experienced a very substantial increase in the last decade, especially in the past five years. Back in 2004, NASA experts defined a wide set of civil applications for UAVs [15], highlighting their potential in the near future in areas such as commercial, Earth Sciences, national security, and land management. This preliminary report was ratified years later by authors such as Hugenholtz et al. [16], who explained how the use of UAVs could revolutionize research methods in the fields of Earth Sciences

and remote sensing. In [17], authors display the results of a detailed study on different UAVs aspects, showing their applicability in Agriculture and Forestry, Disaster Monitoring, Localization and Rescue, Surveillance, Environmental Monitoring, Vegetation Monitoring, Photogrammetry, and so on.

If we focus specifically on research using quadrotor multicopters, authors like Gupte et al. [18] and Colomina and Molina [19] consider that, given their high maneuverability, compactness, and ease of use, different applications for these devices are being found in areas including civil engineering, search and rescue, emergency response, national security, military surveillance, border patrol, and surveillance, as well as in other areas such as Earth Sciences, where they can be used to study climate change, glacier dynamics, and volcanic activity or for atmospheric sampling, among others.

In our case, we are more interested in atmospheric sampling to measure air pollution levels. In this research area, Anderson and Gaston [20] highlight the applicability of UAVs in the field of ecology, emphasizing that the spatial and temporal resolutions of the data obtained by traditional methods often fail to adapt well to the requirements of local ecology-oriented research. Furthermore, the use of UAVs, when flying at low altitudes and speeds, offers new opportunities in terms of ecological phenomena measurements, enabling the delivery of data with a finer spatial resolution. In fact, Zhang and Kovacs [21] explain how the images taken by small UAVs are becoming an alternative to high-resolution satellite images, which are much more expensive, to study the variations in crop and soil conditions. Specifically, the use of UAVs is considered a good alternative given its low cost of operation in environmental monitoring, its high spatial and temporal resolution, and its high flexibility in the scheduling of image acquisitions. A good example of this use can be found in the work of Bellvert et al. [22], which shows how, by using a multicopter equipped with a thermal camera, it was possible to obtain a very precise map of water levels in a vineyard, thereby achieving significant advances in the field of precision agriculture.

Focusing on our topic, despite the presence of several works related to air pollution monitoring using Unmanned Aerial Systems (UAS), the majority of these involve, mainly, swarm creation or communication interaction between them. An example of such work is [23], where authors propose a mobility model for a group of nodes following “Virtual Tracks” (highways, valley, etc.) operating in a predefined “Switch Station” mode, through which nodes can split or merge with another group of nodes.

Different works have been done related to mounting sensors in Unmanned Aerial Vehicles. In this regard, Erman et al. [24] use an UAV equipped with a sensor to create a Wireless Sensor Network, thereby enabling each UAV to act as a sink or as a node, but it does not try to optimize the monitoring process. Teh et al. [25] propose a fixed-wind aircraft carrying a sensor node that acts as a mobile gateway, showing the communication between the UAV and different static base stations which monitor the pollution. In this case, the UAV only recovers the data collected by the stations. Khan et al. [26] propose the design of a lightweight laser-based

sensor for measuring trace gas species using UAVs, mainly analyzing how the optical sensor captures the air pollution samples. In [27], authors use a large-sized aircraft equipped with ozone sensors to cover a wide area in an automated manner, showing how the UAV improves the sampling granularity.

If we analyze works related to mobility models for UAS mobility control that could be used for air pollution monitoring tasks, we can observe that basically no work focuses on the coverage improvement for a certain area.

For instance, in [28], authors propose a mobility model based on the Enhanced Gauss-Markov model to eliminate or limit the sudden stops and sharp turns that the random waypoint mobility model typically creates. Also, in [29], authors present a semirandom circular movement (SRCM) based model. They analyze the coverage and network connectivity by comparing results against a random waypoint mobility model.

The authors of [30] compare their models against random waypoint-based, Markov-based, and Brownian-motion-based algorithms to cover a specific area, analyzing the influence of the use of collision avoidance systems in the time to achieve full area coverage. The work in [31] compares the results of using the “Random Mobility Model” and the “Distributed Pheromone Repel Mobility Model” as direction decision engines (next waypoint) in UAV environments. The authors of [32] propose an algorithm to cover a specific area; it selects a point in space along with the line perpendicular to its heading direction and then drives the UAV based on geometric considerations.

There are works focusing on using UAVs for specific tasks involving autonomous movements. An example is [33], where authors present a mobility model for the self-deployment of an Aerial Ad Hoc Network in a disaster scenario in order to create a flying and flexible communications infrastructure that victims can use. The mobility model proposed is based mainly on the Jaccard dissimilarity metric to control the deployment of the Unmanned Aerial Vehicles composing the network. A similar work is presented in [34], where instead an in-network density analysis is used to select the physical areas that need to be visited by a flying robot.

Focusing solely on existing proposals addressing mobility models, we can find works such as [35] where authors propose the Paparazzi Mobility Model (PPRZM) by defining five types of movements—Stay-On, Waypoint, Eight, Scan, and Oval—following a defined state machine with different probabilities to change between states. There are even studies following animal-based navigation patterns. An example of such work is [36], where authors investigate the UAV placement and navigation strategies with the end goal of improving network connectivity, using local flocking rules that aerial living beings like birds and insects typically follow.

The use of UAVs for air pollution monitoring in a specific area using multirotor drones is, however, still not present in scientific literature, and this work can be seen as one of the first approaches in this direction. Our contribution can be divided into two parts: (i) the design of a low-cost and open-source UAV equipped with off-the-shelf sensors for monitoring tasks and (ii) the deployment of a protocol called

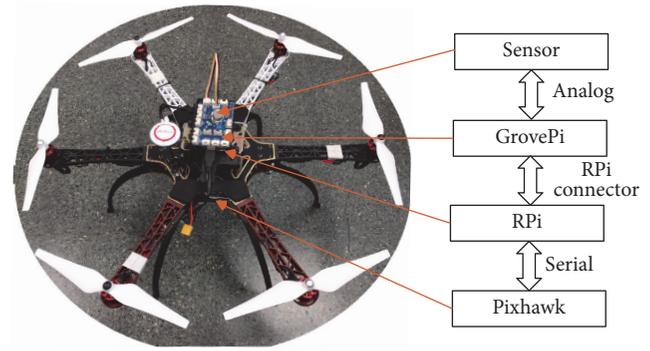


FIGURE 1: Proposed UAV with air pollution sensors.

PdUC (Pollution-driven UAV Control) to automatically track a target area by focusing on the most polluted regions.

3. Overview of the Proposed Solution

To implement a solution for air pollution monitoring using UAVs we have to consider, like in any cyber-physical system, two main aspects: (i) the hardware configuration and (ii) the control process for controlling the system behavior.

By following these guides, our proposal can also be split into two parts: (i) the physical configuration of the UAV and the environmental sensors and (ii) the algorithm to control the UAV for automatically monitoring a specific area called Pollution-driven UAV Control (PdUC).

Even though we are not proposing, in this paper, an implementation using real UAVs, we nevertheless present the specification of the devised cyber-physical system.

3.1. UAV Configuration. We have designed a scheme to dynamically drive the UAV by connecting the UAV control module to a Raspberry Pi [37] and connecting the latter to the set of pollution sensors via an analog converter. The scheme is shown in Figure 1.

The UAV is driven using a Pixhawk Autopilot [38, 39], which controls its physical functioning. The Raspberry Pi is mounted over the UAV chassis and connected to the Pixhawk through a serial port. The sensors are connected to the Raspberry Pi using a Grove Raspberry Hat (GrovePi) [40], which allows connecting different kinds of COTS sensors easily. Specifically, we are using

- (i) *Pixhawk Autopilot*: a high-performance flight control module suitable for several types of autonomous vehicles including multirotors, helicopters, cars, boats, and fixed-wing aircrafts. It is developed under the independent, open hardware Pixhawk project, and it has two main components: (i) an Autopilot hardware provides an industry standard autopilot module designed as a 168 MHz Cortex M4F CPU with 3D ACC/Gyro/MAG/Baro sensors, microSD slot, 5 UARTs, CAN, I2C, SPI, ADC, and so on; (ii) an Autopilot software that includes a real-time Operating System (RTOS) with a POXIS-style environment to control the drone.

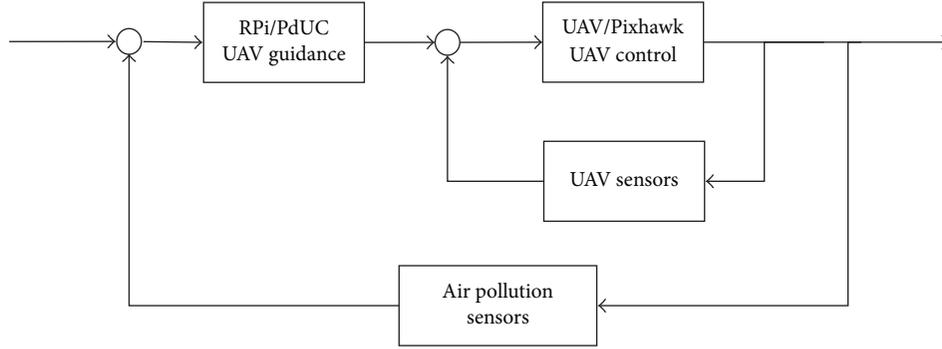


FIGURE 2: UAV control loop.

- (ii) *Raspberry Pi*: one of the most popular single-board-computers (SBC) worldwide. It is a low-cost and small-sized piece of hardware that allows exploring computing and that supports different Operating Systems. The most popular of them is Raspbian, which is based on Debian, although Ubuntu Mate or Windows 10 IoT Core can also be installed, thereby allowing using several programming languages. Besides, all Raspberry Pi versions benefit from several input/output ports operating at 5 V, thus being ideal for all sorts of IoT projects.
- (iii) *GrovePi*: extension board that allows connecting several analog/digital *grove* ports to a Raspberry Pi in an easy way. It has several grove ports: seven digital ports, three analog ports, three I2C ports, one serial port to the GrovePi, and a serial connector to the Raspberry Pi.
- (iv) *Grove Sensors*: sensors which use a grove-standardized connector, providing an easy connection to different boards like GrovePi. There are several COTS environmental sensors such as CO₂, CO, or alcohol. Specifically, we mostly focus on ozone sensors (MQ131).

Figure 2 shows the closed-loop control scheme of our proposal. The Pixhawk Autopilot is responsible for the physical control system of the UAV (lower level), while the Raspberry Pi is in charge of the Guidance system (higher level) determining the way forward.

3.2. Autonomous Driving. To deploy an algorithm for automatically monitoring a specific area we have analyzed, first of all, different existing possibilities that could be useful to our goals.

So, to elaborate the proposed PdUC solution, we have used specific techniques such as the metaheuristics and optimization algorithms described below.

3.2.1. Chemotaxis Metaheuristic. The use of rotary-wing UAVs, equipped with chemical sensors and tasked to survey large areas, could follow chemotactic [41] mobility patterns, since their flight behavior could easily implement the

following two-phase algorithm: first, read a pollution concentration while hovering; next, follow a chemotactic step.

Chemotaxis metaheuristics are based on bacteria movement. In this model, the microorganisms react to a chemical stimulus by moving towards areas with a higher concentration of some components (e.g., food) or moving away from others (e.g., poison). In our system, we have considered the following adaptation of the chemotaxis. Let us consider an agent i moving on a Euclidean plane, located at position \vec{P}_j^i from an absolute reference axis, and moving along time in sequential steps j . For every chemotactic step, a new position \vec{P}_j^i is calculated based on the previous one, defined by x_{j-1}^i and y_{j-1}^i , plus a step size d^i applying a random direction θ_j^i , as specified in (1).

$$\vec{P}_j^i = \begin{pmatrix} x_{j-1}^i \\ y_{j-1}^i \end{pmatrix} + \begin{pmatrix} d^i \times \cos(\theta_j^i) \\ d^i \times \sin(\theta_j^i) \end{pmatrix}, \quad (1)$$

$$\theta_j^i = \begin{cases} \theta_{j-1}^i + \alpha_j^i, & p_j^i \geq p_{j-1}^i, \\ -\theta_{j-1}^i + \beta_j^i, & p_j^i < p_{j-1}^i. \end{cases} \quad (2)$$

The direction θ_j^i , as shown in (2), is calculated on the basis of the concentration value of a certain chemical component, sampled by an agent i at step j : p_j^i . With respect to the previously sampled value p_{j-1}^i , the following two types of movements are contemplated: *Run* and *Tumble*. In the former, Run, when the component concentration is increased with respect to the previous sample, the movement continues to follow the same direction as before (θ_{j-1}^i) plus a random angle α_j^i . Regarding the latter, Tumble, when the concentration is decreasing, the movement takes a turn in the opposite direction $-\theta_{j-1}^i$, plus a random angle β_j^i . Notice that both α_j^i and β_j^i are used to introduce variability and to maximize the gradient, allowing reaching the most polluted areas faster.

3.2.2. Particle Swarm Optimization. Particle Swarm Optimization (PSO) is a technique introduced in [42] where a solution to a problem is represented as a particle p^i moving in a D-dimensional space at a time t ; each particle p^i

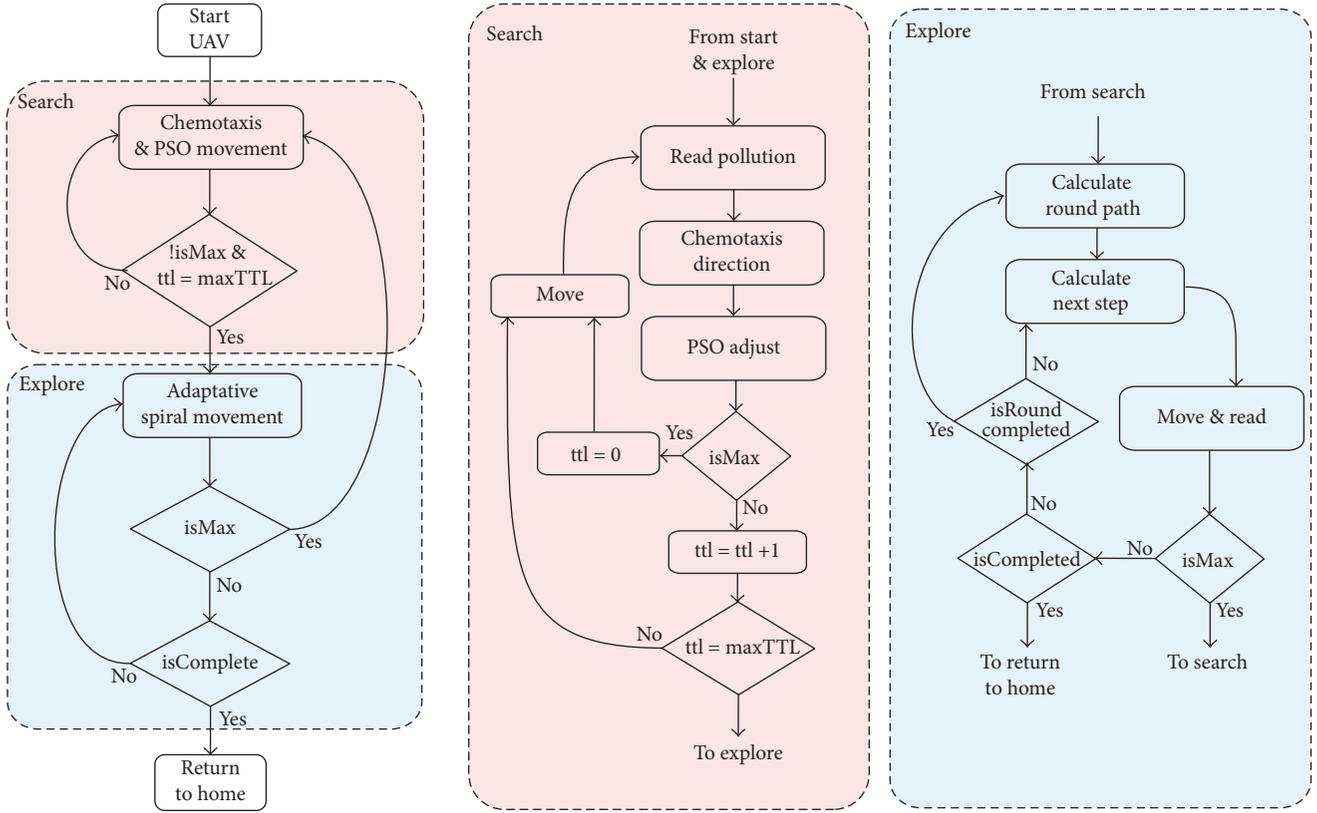


FIGURE 3: Overview of different states associated with the PdUC algorithm.

maintains its position p_t^i and its best performance position p_b^i . To determine the next position p_{t+1}^i , PSO calculates the stochastic adjustment in the direction of the previous local best position of i 's p_b^i element, along with the general best position of any element p_b^g , as shown in

$$p_{t+1}^i = \alpha \cdot p_t^i + U(0, \beta) \cdot (p_b^i - p_t^i) + U(0, \beta) \cdot (p_b^g - p_t^i), \quad (3)$$

where α and β are constants to calibrate the algorithm and $U(0, \beta)$ is a random number between $[0, \beta]$.

4. Proposed Autonomic Solution

To consistently drive the UAVs, so as to achieve the desired area coverage goals, we have devised the following algorithm, which incorporates a chemotactic approach.

4.1. PdUC Algorithm. In this context, we have developed an algorithm called Pollution-driven UAV Control (PdUC), based on the chemotaxis metaheuristic concept, to search an area for the highest pollution concentration levels. Once this pollution hotspot is found, the flying drone covers the whole area by following a spiral movement, starting from the most polluted location.

Our algorithm is composed of two phases: (i) a search phase, in which the UAV searches for a globally maximum

pollution value, and (ii) an exploration phase, where the UAV explores the surrounding area, following a spiral movement, until one of the following conditions occurs: it covers the whole area, the allowed flight time ends, or it finds another maximum value, in which case it returns to the search phase, as shown Figure 3.

The exploration phase is based mainly on two previously described techniques: a chemotaxis metaheuristic and a local particle swarm optimization algorithm. As detailed in Algorithm 1, initially, before the UAV starts its first movement, it samples the pollution value and puts it in a buffer. For each chemotactic step, it starts to hover, collects another sample, and compares it with the previous one. If the sampling variation is positive (increasing), the UAV follows a “Run” chemotaxis direction, with a random α_j^i of $[-30, 30]$ degrees. Otherwise, if the sampling variation is decreasing, the UAV calculates the “Tumble” chemotaxis direction in the reverse orientation with a random β_j^i of $[-150, 150]$ degrees, although modified by the actual maximum value reached (md^i), as shown in Figure 4. Equation (4) denotes the formula to calculate the new direction, and γ specifies the weight of the md^i , which must be between 0 and 1.

To determine when PdUC has found a maximum local value, we use a TTL (time-to-live) counter. When PdUC finds a maximum value, the TTL is reset and increasing until a new maximum pollution value is found or until the maximum TTL value is reached. In this case, PdUC reverts to

```

(1) while isSearching do
(2)    $p_2^{pollution} \leftarrow CurrentPollution(p_2)$ 
(3)    $\nabla poll \leftarrow p_2^{pollution} - p_1^{pollution}$ 
(4)    $p_1 \leftarrow p_2$ 
(5)   if  $\nabla poll > 0$  then
(6)      $tfl \leftarrow 0$ 
(7)      $p_2 \leftarrow Run(p_1)$ 
(8)      $p_{max} \leftarrow p_2$ 
(9)   else
(10)     $p_2 \leftarrow Tumble(p_1)$ 
(11)     $tfl \leftarrow tfl + 1$ 
(12)     $p_2 \leftarrow AdjustPSO(p_2, p_{max})$ 
(13)    if isInsideArea( $p_2$ ) then
(14)      MoveTo( $p_2$ )
(15)    else
(16)       $p_2 \leftarrow Tumble(p_1)$ 
(17)    if  $tfl > tfl_{max}$  then
(18)      isSearching  $\leftarrow$  false
(19)      isExploring  $\leftarrow$  true
(20) end

```

ALGORITHM 1: PdUC search phase.

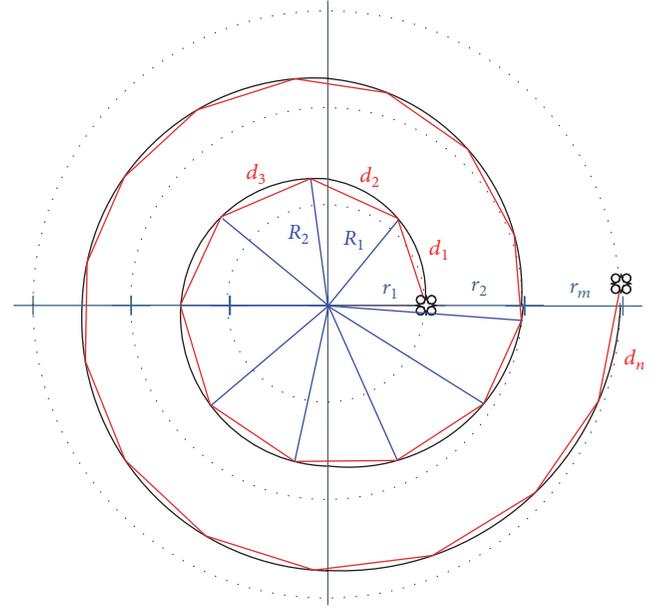


FIGURE 5: PdUC algorithm: exploration phase.

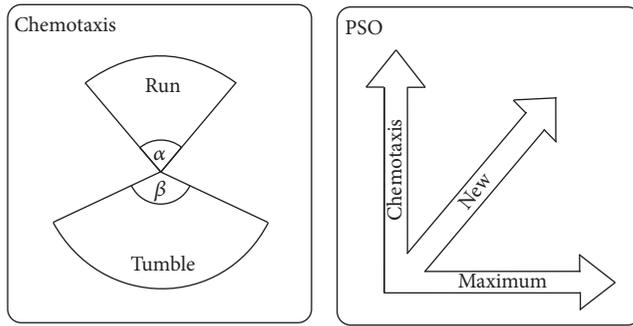


FIGURE 4: PdUC algorithm: calculation of a new direction.

the exploration phase since it considers that a local maximum value has been found.

$$\theta_j^i = \begin{cases} \theta_{j-1}^i + \alpha_j^i, & \text{Run,} \\ (1 - \gamma)(-\theta_{j-1}^i + \beta_j^i) + \gamma md^i, & \text{Tumble.} \end{cases} \quad (4)$$

Once a maximum value is reached, the next phase is to explore the surrounding area. As shown in algorithm 2, this is achieved by following an Archimedean spiral similar to the one depicted in Figure 5. Starting from the maximum value, it covers the surrounding area by applying a basic step size d_j^i and changing it depending on the detected pollution variations, a procedure that is similar to the *finding* phase. If the variation is less than a preset value c^i , the step size increases until reaching $3 \times d_j^i$; otherwise, it decreases until d_j^i is reached. If a maximum pollution value is found, PdUC automatically returns to the exploration phase. Finally, once

the whole area is covered, the UAV changes to a *return-to-base* (RTB) mode to finish the exploration.

4.2. Algorithm Optimization. Next, analyzing the overall behavior, we have introduced some modifications to optimize the performance of the proposed PdUC algorithm.

4.2.1. Spiralling with Alternating Directions. As shown in Figure 6, to avoid large steps in the exploration phase when the spiral center is next to a border, the direction of the spiral will alternate for each round to allow minimizing the length of some of the steps. To this purpose, for each spiral round, we calculate the direction adopted as being the opposite direction with reference to the previously used one. The system can get the general size of the area to search, as well as its borders, before starting the mission. This procedure takes place in line 4 of Algorithm 2. In detail, it follows

$$\theta_{s,r} = \begin{cases} \alpha + \beta_{s,c}, & \text{if } r \text{ is even,} \\ \alpha - \beta_{s,c}, & \text{if } r \text{ is odd,} \end{cases} \quad (5)$$

$$p_{s,r} = \begin{pmatrix} x_s \\ y_s \end{pmatrix} = \begin{pmatrix} x_c + R_s \times \cos(\theta_{s,r}) \\ y_c + R_s \times \sin(\theta_{s,r}) \end{pmatrix},$$

where $\theta_{s,r}$ defines the angle in round r and step s , α is the initial angle, and β_s is the angle in step s . Using it, angle $\theta_{s,r}$ and the next point p_s are calculated using as a reference the coordinates for the spiral center (x_c and y_c) and radius R_s .

4.2.2. Skipping Previously Monitored Areas. As shown in Figure 7, to avoid monitoring the same area multiple times, all samples, which were taken within the area monitored during the exploration phase, are internally stored. For this purpose,

```

(1) while isExploring do
(2)   round  $\leftarrow$  round + 1
(3)   roundsize  $\leftarrow$   $2\pi \cdot (\text{round} + \text{round}_{\text{next}})/2$ 
(4)   rounddirection  $\leftarrow$  -previousdirection
(5)   anglecount  $\leftarrow$   $2\pi/(\text{round}_{\text{size}}/d)$ 
(6)   step  $\leftarrow$  0
(7)   angle  $\leftarrow$  0
(8)   while step < roundsize and isExploring do
(9)     if isInsideArea(p2) and isNotMonitored(p2) then
(10)      p2pollution  $\leftarrow$  CurrentPollution(p2)
(11)      if p2 > pmax then
(12)        store(pmax)
(13)        store(round)
(14)        isExploring = false
(15)        isSearching = true
(16)        pmax = p2
(17)      else
(18)         $\nabla \text{poll} \leftarrow p2_{\text{pollution}} - p1_{\text{pollution}}$ 
(19)        MoveTo(p2)
(20)      step  $\leftarrow$  step + d
(21)      angle  $\leftarrow$  angle + anglecount  $\times$  rounddirection
(22)      p1  $\leftarrow$  p2
(23)      p2  $\leftarrow$  NextPoint(p1, angle, step)
(24)      previousdirection  $\leftarrow$  rounddirection
(25) end

```

ALGORITHM 2: PdUC exploration phase.

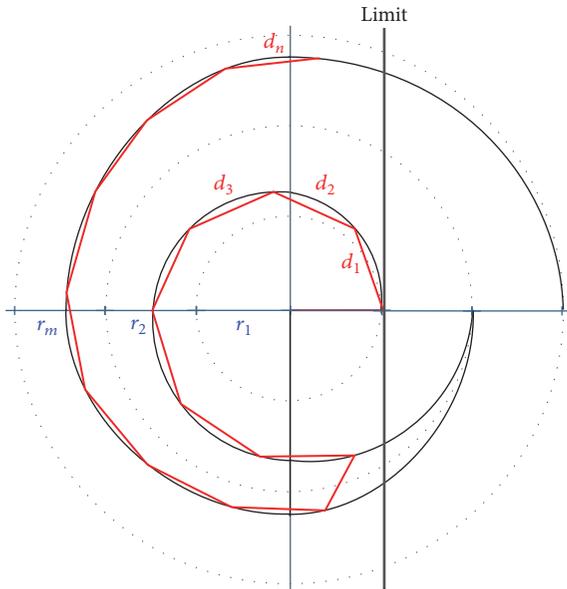


FIGURE 6: PdUC algorithm: alternating spiral direction.

PdUC maintains a list containing the location of the central position of all spirals with their respective radius to determine the monitored areas (as a circumference determined by a center and a radius). Next, in the exploration phase, all points inside these circles are omitted for the sake of celerity, as shown in line 9 of Algorithm 2.

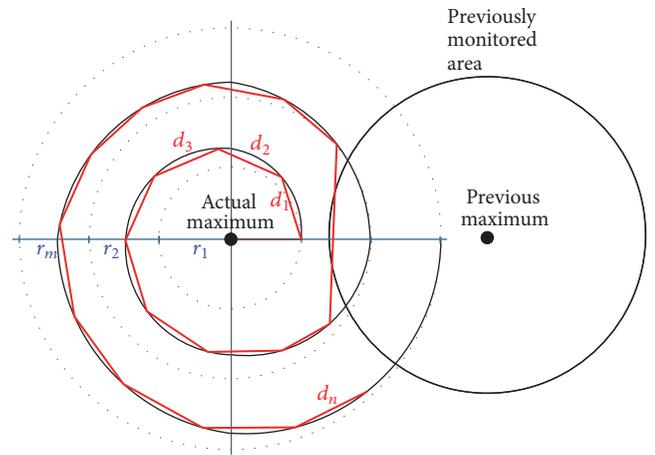


FIGURE 7: PdUC algorithm: skipping monitored areas.

5. Validation and Simulation

To validate our protocol, we have run several simulations with different configurations implemented in the OMNeT++ simulation tool, as shown in Figure 8.

To prepare a suitable data environment, we have created various pollution distribution maps representing ozone levels to be used as inputs for testing. These pollution maps were generated using the *R Graph* tool [43] and following a Kriging-based interpolation [44]. In particular, a Gaussian distribution is used to adjust the parameters coming from

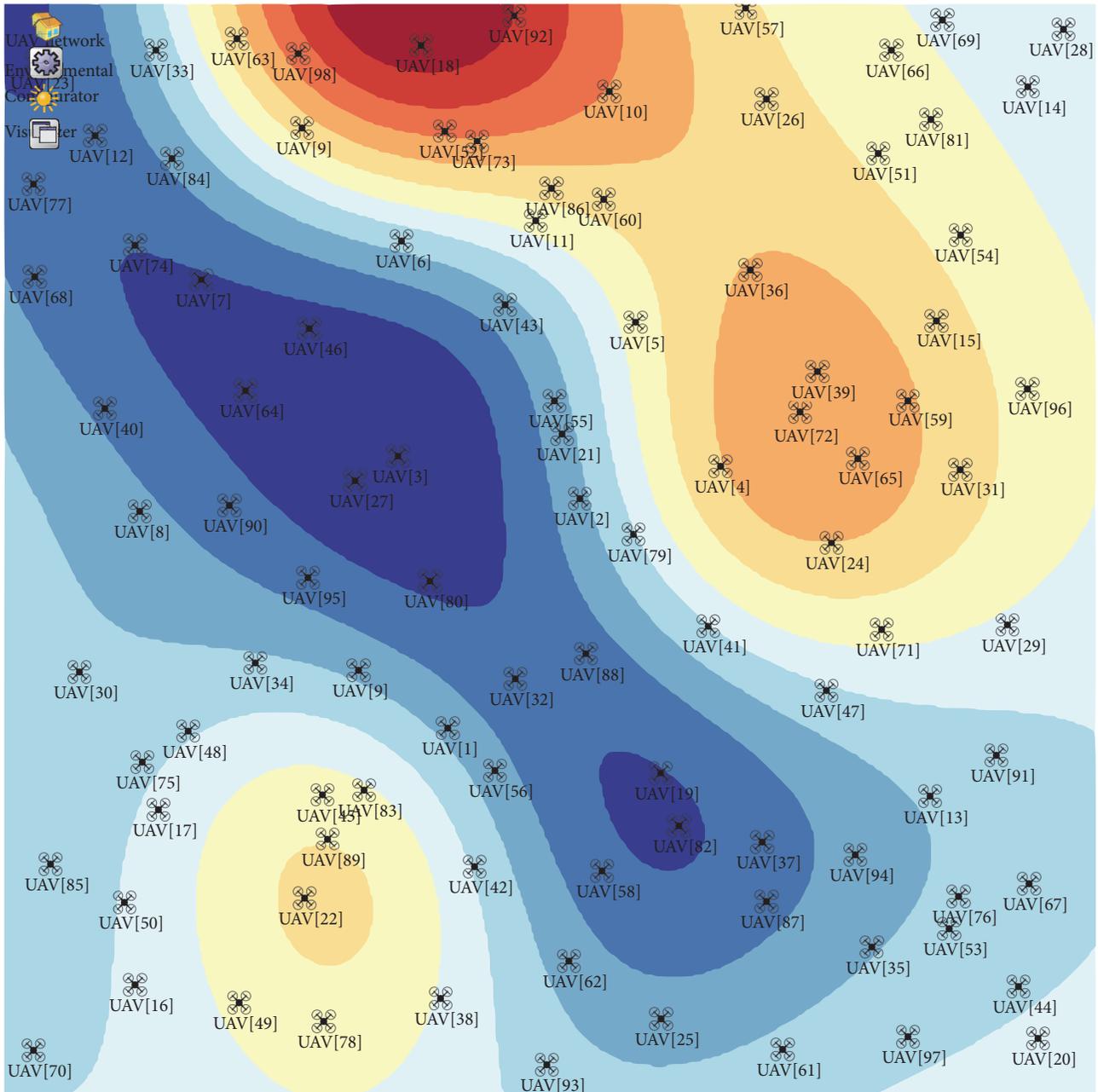


FIGURE 8: Example simulation scenario showing possible initial UAV positions over a randomly generated pollution map.

random data sources of ozone concentration. The actual values range between 40 and 180 ppb, thereby providing a realistic ozone distribution.

Figure 9 shows some samples of the created maps, which have the highest pollution concentration (areas in red) located at completely different positions due to the stochastic scenario generation procedure adopted.

Using the previously created data as input, we have run several simulations using OMNeT++, comparing our protocol against both the Billiard and Spiral mobility patterns. In the simulator, we have created a mobility model

implementation of PdUC. In addition, to simulate the sampling process, we have configured OMNeT++ to periodically perform measurements taken from the pollution distribution map defined for the test.

Figure 10 shows an example of the path followed by an UAV using the PdUC algorithm as a guidance system. As expected, the UAV starts a search process throughout the scenario until it locates a position with the highest degree of pollution (local maximum). Afterward, it follows a spiral pattern to gain awareness of the surrounding gradients. If, while following the spiral-shaped scan path, it finds a higher

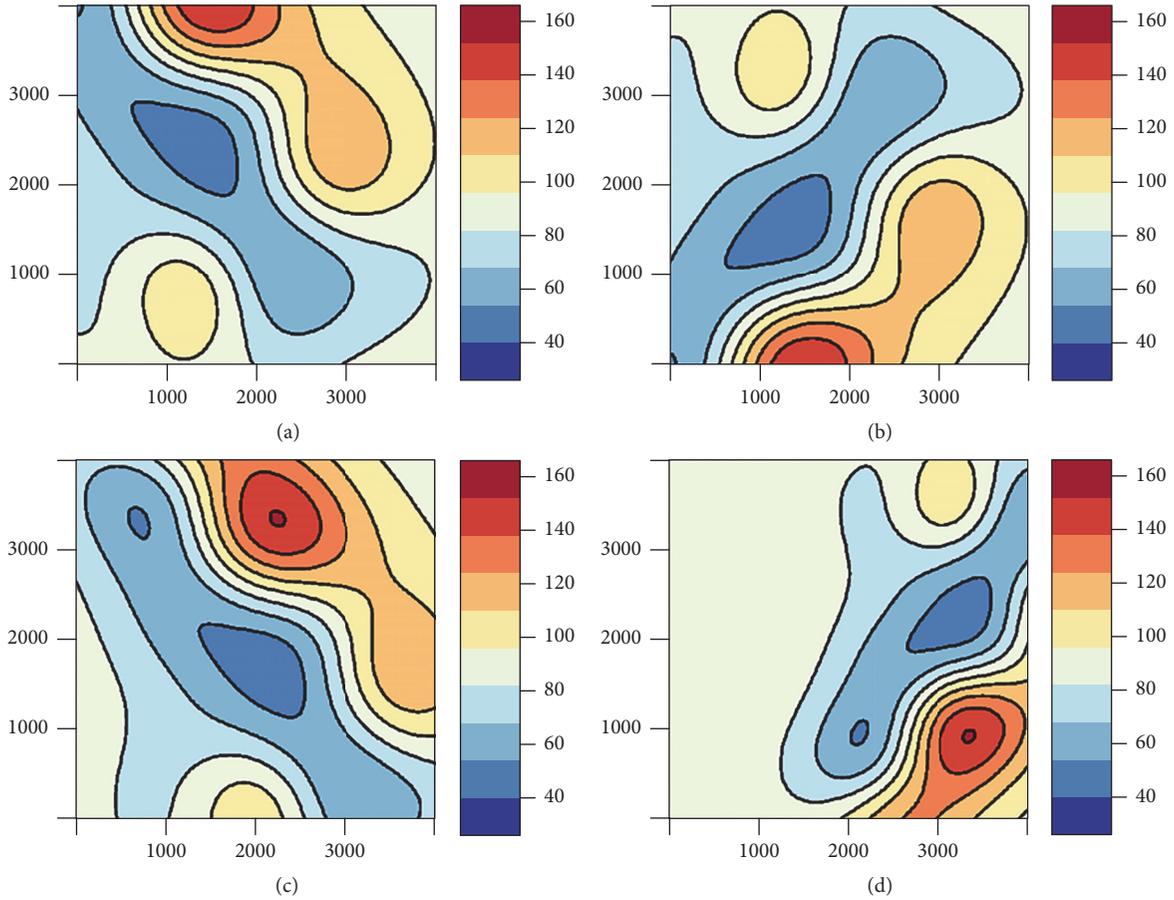


FIGURE 9: Pollution distribution examples used for validation.

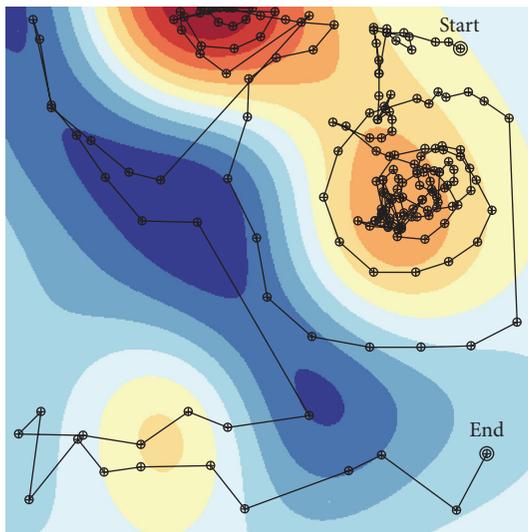


FIGURE 10: Example of an UAV path when adopting the PdUC mobility model.

pollution value, the algorithm again switches to the search phase. Finally, when the entire target area has been sampled, the algorithm finishes.

TABLE 1: Simulation parameters.

Parameter	Value
Area	4 × 4 Km
Pollution range	[40–180] ppb
Sampling error	10 ppb
Max. speed	20 m/s
Sampling time	4 seconds
Step distance	100 m
Mobility models	Billiard, Spiral, and PdUC

To compare the three options under study, we recreate, using the R Graph tool, the pollution distribution maps using the simulation output as the input for the Kriging-based interpolation. In this way, we obtain new pollution maps for comparison against the ones used as reference.

Table 1 summarizes the parameters used in the simulations.

Since we are proposing the PdUC algorithm for rural environments, the simulation area defined is a 4 × 4 Km area. As indicated above, the pollution distribution relies on synthetic maps that are generated by combining a random Kriging interpolation following a Gaussian model with values between 40 and 180 units based on the Air Quality Index

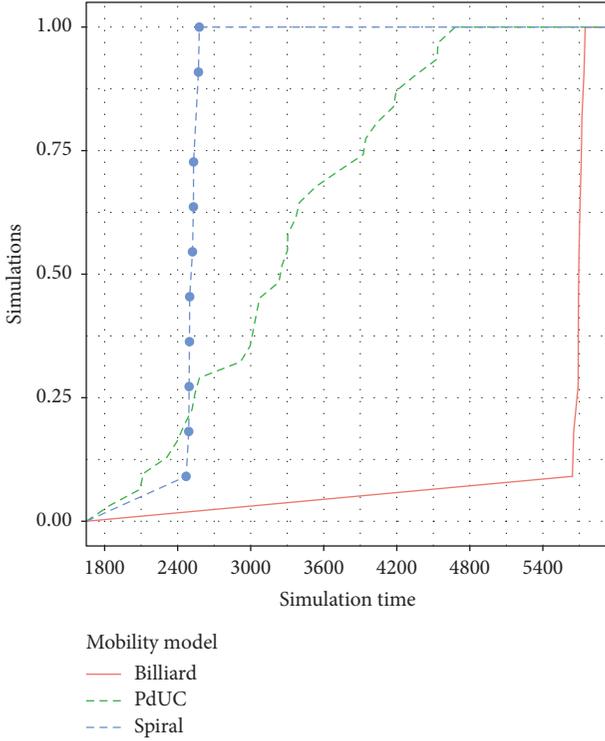


FIGURE 11: Cumulative Distribution Function of the time spent at covering the complete area for the Billiard, Spiral, and PdUC mobility models.

(AQI) [45]. Since samples are taken using off-the-shelf sensors, which are not precise, we introduce a random sampling error of ± 10 ppb based on real tests using the MQ131 (Ozone) sensor. In our simulation, we set the maximum UAV speed to 20 m/s, a value achievable by many commercial UAVs. The step distance defined between consecutive samples is 100 meters. Once a new sampling location is reached, the monitoring time per sample is defined to be 4 seconds.

The mobility models used are Billiard, Spiral, and PdUC. These models have different assumptions regarding the initial UAV position. In the *Billiard* model, the UAV starts in a corner of the target area and then covers the whole area by “bouncing” when reaching the borders. The *Spiral* model starts at the center of the area to cover and then gradually moves to the periphery of the scenario following a spiral pattern. Finally, PdUC is set to start at a random position within the target area.

We now proceed by analyzing the time required to cover the entire area using each of the approaches being tested. For this purpose, we defined 100 simulations for each model (Billiard, Spiral, and PdUC) and determined the required time to cover the whole area, estimating the pollution map afterward.

For each run, the starting position of the UAV is randomly set on the map, as shown in Figure 8.

Figure 11 shows the Cumulative Distribution Function relative to the time required to cover the whole area for the three mobility models. It can be seen that the Billiard and

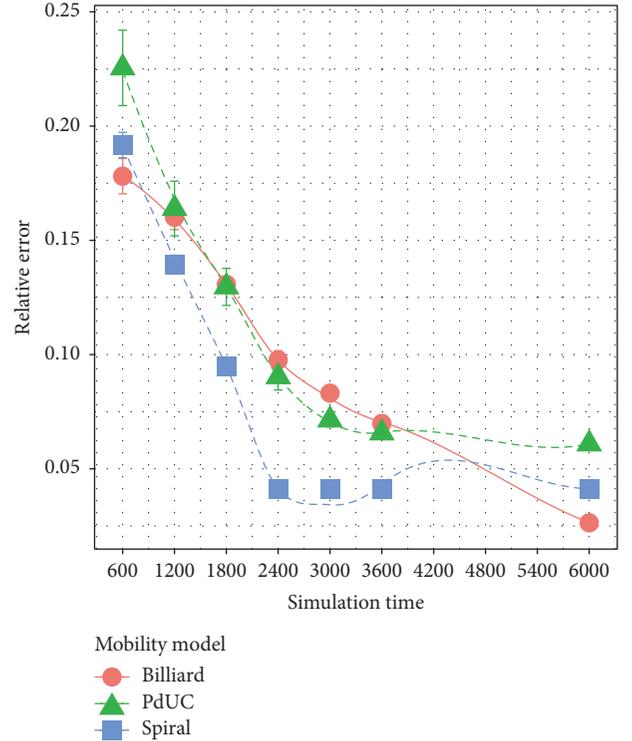


FIGURE 12: Relative error comparison between the PdUC, Billiard, and Spiral mobility models at different times, when analyzing all the values.

Spiral models do not depend on the start position, spending a nearly constant time (5600 and 2600 seconds, resp.) for each configuration defined. In the case of the PdUC mobility model, since it reacts to air pollution, the time required to cover the complete area varies between 1800 and 4300 seconds, depending on the start position.

Due to battery restrictions, it is interesting to analyze how fast each mobility model discovers the most polluted areas and how accurately does it recreate the pollution distribution. For this purpose, we analyze the relative error for the three mobility models at different time instants (600, 1200, 1800, 2400, 3000, and 6000 seconds); this error is defined by

$$e_t = \frac{\sum_{i=1}^m \sum_{j=1}^n |(s_{x,y,t} - b_{x,y}) / \Delta b|}{m \cdot n}, \quad (6)$$

where e_t is the relative error at time t ; $s_{x,y,t}$ is the recreated pollution value at position (x, y) using the samples taken during simulation until time t , $b_{x,y}$ is the reference pollution value at position (x, y) , and n and m are the dimensions of the target area, respectively.

Figure 12 shows the temporal evolution of the relative error between the three mobility models (Billiard, Spiral, and PdUC) and the original one. We can observe that all mobility models have roughly the same behavior: they start with a high relative error, which is foreseeable since we are using Kriging interpolation to recreate the pollution distribution, and it tends to the mean value when the number of samples

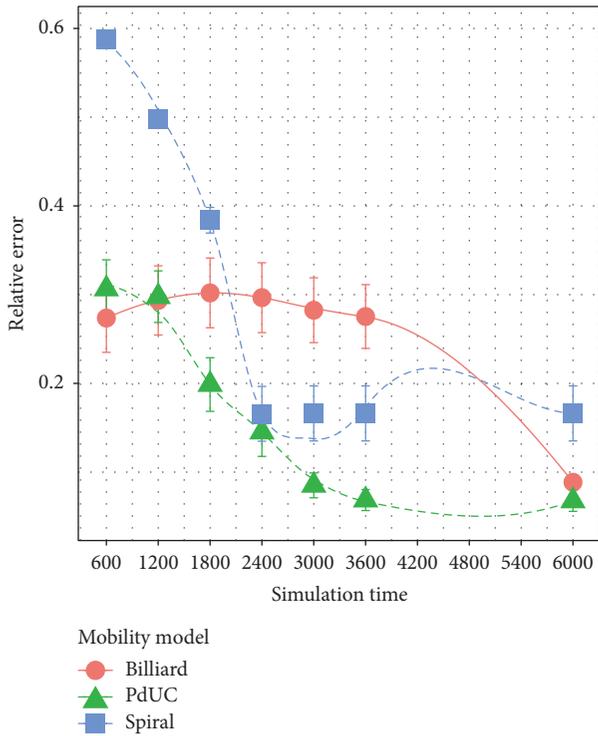


FIGURE 13: Relative error comparison between PdUC, Billiard, and Spiral mobility models at different times when only considering values higher than 120 ppb.

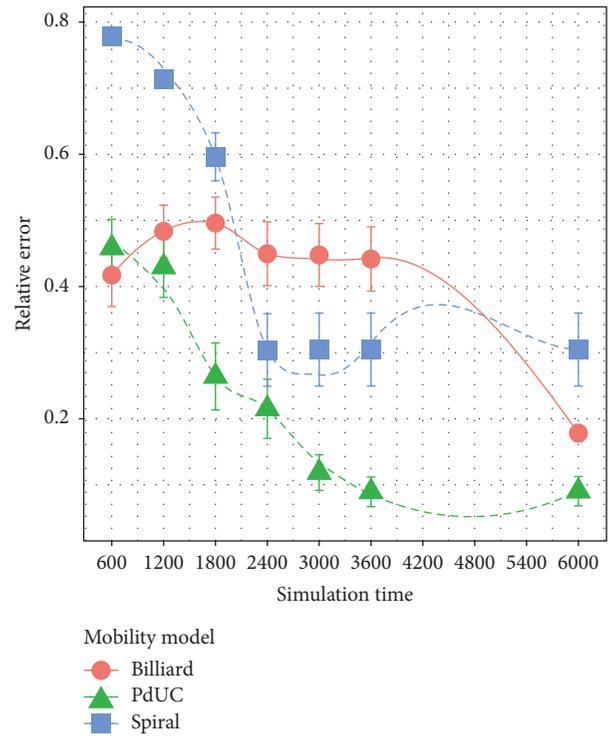


FIGURE 14: Relative error comparison between PdUC, Billiard, and Spiral mobility models at different times when only considering values higher than 150 ppb.

is not enough. Then, as more samples become available, the spatial interpolation process quickly becomes more precise.

Although the three mobility models are similar, the spiral approach achieves a better performance in terms of relative error reduction. However, if we analyze only the most polluted regions, that is, regions characterized by values higher than a certain threshold (120 and 150 ppm in our case, based on AQI [45]), we find that PdUC clearly provides better results.

Figures 13 and 14 show the comparison between the Billiard, Spiral, and PdUC mobility models at different times when only focusing on air pollution values higher than 120 and 150 ppb, respectively. These results show that PdUC clearly provides better results than the Billiard and Spiral movement patterns, outperforming their accuracy from nearly the beginning of the experiment (1200 seconds) and reaching the lowest relative error values in just 3600 seconds, with these two other mobility approaches more than doubling the error values for the same time. In particular, the Billiard mobility pattern requires about 6000 seconds to achieve a similar degree of accuracy (120 ppb case), while the Spiral approach is not able to achieve values as low as PdUC in any of the cases. This occurs because PdUC focuses on the highest values in the chemotaxis-based phase. PdUC always prioritizes the most polluted areas in detriment of less polluted ones, thus allowing obtaining, at least, details about the region with the highest pollution values.

To complete our study, Figure 15 presents an example of the evolution of predicted pollution values for the whole target area and for the three algorithms under analysis (Spiral, Billiard, and PdUC), at different times (1200 s, 2400 s, 3600 s, and 6000 s). We can observe that PdUC is able to quickly find the most polluted areas, while the effectiveness of other approaches highly depends on the actual location of pollution hotspots in order to detect them at an early stage.

6. Open Issues

Unmanned Aerial Systems (UAS) have been quickly adopted in different application areas due to their flexibility and relatively low cost. Focusing on the environmental monitoring area, in a previous work we introduced the idea of using UAVs for air pollution monitoring [46] by equipping them with off-the-shelf sensors. Instead, in the current paper, we introduce an algorithm called PdUC to guide a single UAV in the task of monitoring a specific area. However, there are still several open issues related to this topic.

Until now, we have only considered operations limited to a single UAV. The next step in our research is to introduce multiple-UAVs and the associated cooperation schemes. The following aspects need to be addressed to follow this research line:

- (i) *Cooperation.* To maximize the effectiveness and reduce mapping times, it is advisable to have several

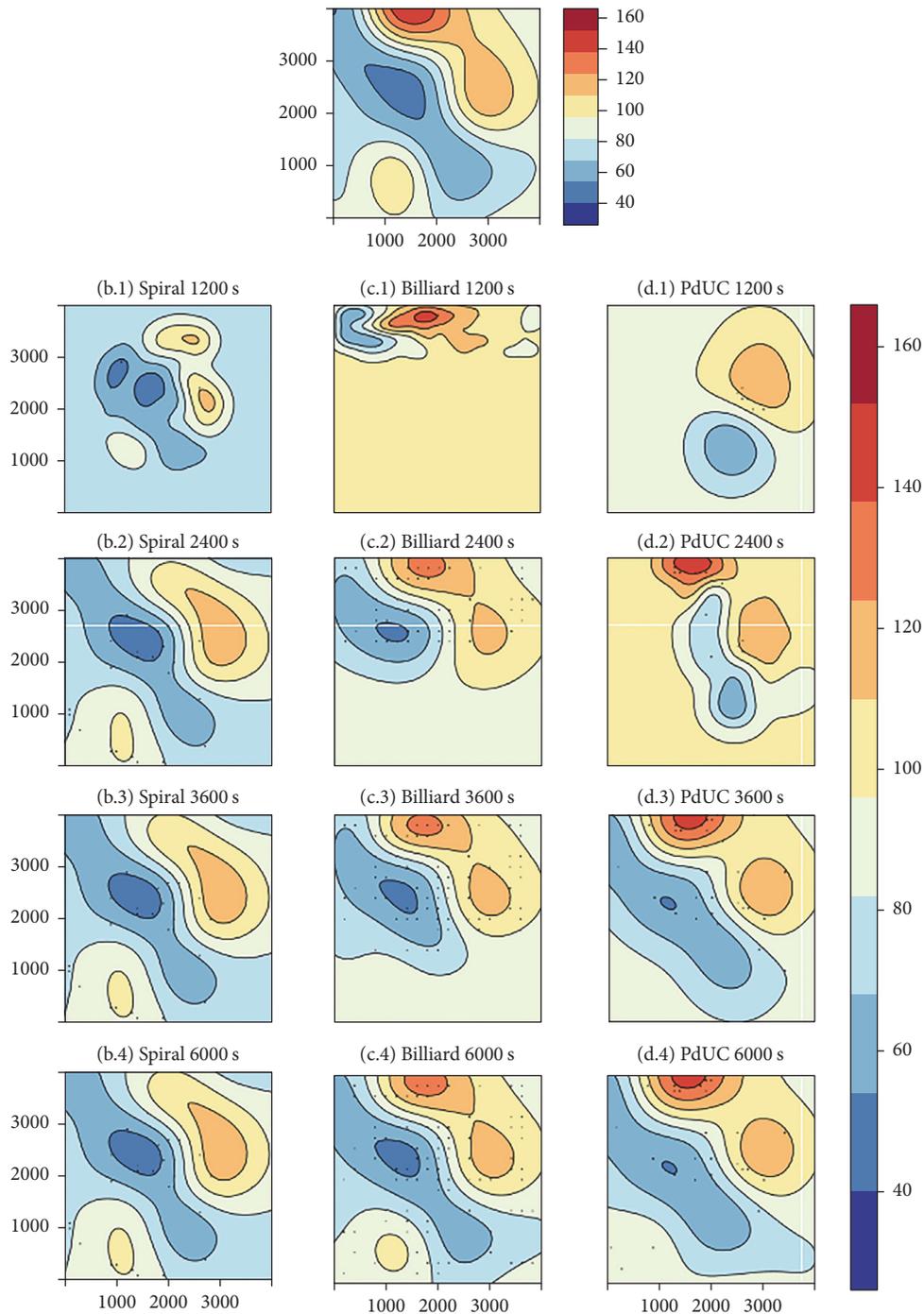


FIGURE 15: Visual representation of the estimation output for the PdUC, Billiard, and Spiral mobility models at different times.

UAVs that cooperate with each other to achieve the same task, thereby accelerating the whole process and avoiding battery exhaustion before completing the monitoring process.

- (ii) *Collision Avoidance*. Since the different UAVs are expected to have some degree of autonomy regarding their mobility pattern, a correct coordination between nearby UAVs is required to avoid collisions when flying at a close range.

- (iii) *Communications*. To achieve the aforementioned goals of cooperation and collision avoidance, communications between UAVs and between UAVs and a central management unit are required.

On the other hand, using mobile sensors installed on UAVs introduces new issues to the sensing process that should also be addressed:

- (i) *Altitude*. Despite the fact that currently most pollution studies are made at a ground level, the use

of UAVs allows determining the concentration of pollutants at different heights, allowing determining if there are layers of pollutants that can cause health problems in rugged mountainsides.

- (ii) *Influence of the Wind*. The sampling procedure includes sensors that are sensitive to the wind conditions. In addition, wind causes the overall pollution map to be more dynamic. In this context, both issues deserve more scrutiny.

7. Conclusions

Despite the fact that we have several options to monitor air pollution in urban scenarios, being crowdsensing an emerging approach arousing great interest, finding an adequate approach for industrial or rural areas remains a pending task.

Recently, Unmanned Aerial Systems have experienced unprecedented growth, offering a platform for the fast development of solutions due to their flexibility and relatively low cost; in fact, they can be good options to solve the previous requirements, allowing monitoring remote areas that are difficult to access.

In this paper, we propose a solution where we equip an UAV with off-the-shelf sensors for monitoring tasks, using a Pixhawk Autopilot for UAV control, and a Raspberry Pi for sensing and storing environmental pollution data.

To automatically analyze pollution values within a target area, we also propose an adaptive algorithm for autonomous navigation called Pollution-based UAV Control system (PdUC). This algorithm allows an UAV to autonomously monitor a specific area by prioritizing the most polluted zones. In particular, PdUC combines different concepts including a chemotaxis metaheuristic, a local particle swarm optimization (PSO), and an Adaptive Spiralling technique, to create an algorithm able to quickly search for hotspots having high pollution values, and to cover the surrounding area as well, thereby obtaining a complete and detailed pollution map of the target region.

To validate our proposal, we compared the proposed PdUC solution against the Billiard and Spiral mobility models through simulations implemented in OMNeT++. Simulation experiments show that PdUC offers significantly better performance at reducing prediction errors, especially regarding the accuracy achieved for the high-values range.

Conflicts of Interest

The received funding does not lead to any conflicts of interest regarding the publication of the manuscript.

Acknowledgments

This work has been partially carried out in the framework of the DIVINA Challenge Team, which is funded under the Labex MS2T program. Labex MS2T is supported by the French Government, through the program “Investments for the Future” managed by the National Agency for Research (Reference: ANR-11-IDEX-0004-02). This work was also

supported by the “Programa Estatal de Investigación, Desarrollo e Innovación Orientada a Retos de la Sociedad, Proyecto I+D+I TEC2014-52690-R,” the “Programa de Becas SENESCYT de la República del Ecuador,” and the Research Direction of University of Cuenca.

References

- [1] A. Seaton, D. Godden, W. MacNee, and K. Donaldson, “Particulate air pollution and acute health effects,” *The Lancet*, vol. 345, no. 8943, pp. 176–178, 1995.
- [2] Q. S. McFrederick, J. C. Kathilankal, and J. D. Fuentes, “Air pollution modifies floral scent trails,” *Atmospheric Environment*, vol. 42, no. 10, pp. 2336–2348, 2008.
- [3] D. Mage, G. Ozolins, P. Peterson et al., “Urban air pollution in megacities of the world,” *Atmospheric Environment*, vol. 30, no. 5, pp. 681–686, 1996.
- [4] H. Mayer, “Air pollution in cities,” *Atmospheric Environment*, vol. 33, no. 24–25, pp. 4029–4037, 1999.
- [5] P. S. Kanaroglou, M. Jerrett, J. Morrison et al., “Establishing an air pollution monitoring network for intra-urban population exposure assessment: A location-allocation approach,” *Atmospheric Environment*, vol. 39, no. 13, pp. 2399–2409, 2005.
- [6] O. Alvear, W. Zamora, C. Calafate, J.-C. Cano, and P. Manzoni, “An architecture offering mobile pollution sensing with high spatial resolution,” *Journal of Sensors*, vol. 2016, Article ID 1458147, 2016.
- [7] A. Adam-Poupart, A. Brand, M. Fournier, M. Jerrett, and A. Smargiassi, “Spatiotemporal modeling of ozone levels in Quebec (Canada): a comparison of kriging, land-use regression (LUR), and combined bayesian maximum entropy—LUR approaches,” *Environmental Health Perspectives*, vol. 122, no. 9, pp. 970–976, 2014.
- [8] M. Pujadas, J. Plaza, J. Terés, B. Artíñano, and M. Millán, “Passive remote sensing of nitrogen dioxide as a tool for tracking air pollution in urban areas: The Madrid urban plume, a case of study,” *Atmospheric Environment*, vol. 34, no. 19, pp. 3041–3056, 2000.
- [9] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, “BikeNet: a mobile sensing system for cyclist experience mapping,” *ACM Transactions on Sensor Networks*, vol. 6, pp. 87–101, 2009.
- [10] M. André, “The ARTEMIS European driving cycles for measuring car pollutant emissions,” *Science of the Total Environment*, vol. 334–335, pp. 73–84, 2004.
- [11] M. Brkovic and V. Sretovic, “Urban sensingsmart solutions for monitoring environmental quality: Case studies from serbia in. 48th ISOCARP International Congress: Fast Forward: Planning in a (hyper) dynamic urban context,” Perm, Russia, 2012.
- [12] S.-C. Hu, Y.-C. Wang, C.-Y. Huang, and Y.-C. Tseng, “Measuring air quality in city areas by vehicular wireless sensor networks,” *Journal of Systems and Software*, vol. 84, no. 11, pp. 2005–2012, 2011.
- [13] Y. Cheng, X. Li, Z. Li et al., “AirCloud: a cloud-based air-quality monitoring system for everyone,” in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems (SenSys '14)*, pp. 251–265, Memphis, TN, USA, November 2014.
- [14] M. Dunbabin and L. Marques, “Robots for environmental monitoring: Significant advancements and applications,” *IEEE Robotics and Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

- [15] T. H. Cox, C. J. Nagy, M. A. Skoog, I. A. Somers, and R. Warner, Civil uav capability assessment.
- [16] C. H. Hugenholtz, B. J. Moorman, K. Riddell, and K. Whitehead, "Small unmanned aircraft systems for remote sensing and Earth science research," *Eos, Transactions American Geophysical Union*, vol. 93, no. 25, p. 236, 2012.
- [17] G. Pajares, "Overview and current status of remote sensing applications based on unmanned aerial vehicles (UAVs)," *Photogrammetric Engineering and Remote Sensing*, vol. 81, no. 4, pp. 281–329, 2015.
- [18] S. Gupte, P. I. T. Mohandas, and J. M. Conrad, "A survey of quadrotor unmanned aerial vehicles," in *Proceedings of the Proceeding of the IEEE Southeastcon*, pp. 1–6, Orlando, Fla, USA, March 2012.
- [19] I. Colomina and P. Molina, "Unmanned aerial systems for photogrammetry and remote sensing: a review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 92, pp. 79–97, 2014.
- [20] K. Anderson and K. J. Gaston, "Lightweight unmanned aerial vehicles will revolutionize spatial ecology," *Frontiers in Ecology and the Environment*, vol. 11, no. 3, pp. 138–146, 2013.
- [21] C. Zhang and J. M. Kovacs, "The application of small unmanned aerial systems for precision agriculture: a review," *Precision Agriculture*, vol. 13, no. 6, pp. 693–712, 2012.
- [22] J. Bellvert, P. J. Zarco-Tejada, J. Girona, and E. Fereres, "Mapping crop water stress index in a 'Pinot-noir' vineyard: Comparing ground measurements with thermal remote sensing imagery from an unmanned aerial vehicle," *Precision Agriculture*, vol. 15, no. 4, pp. 361–376, 2014.
- [23] B. Zhou, K. Xu, and M. Gerla, "Group and swarm mobility models for ad hoc network scenarios using virtual tracks," in *Proceedings of the OrlCOM 2004 - 2004 IEEE Military Communications Conference*, pp. 289–294, November 2004.
- [24] A. T. Erman, L. V. Hoesel, P. Havinga, and J. Wu, "Enabling mobility in heterogeneous wireless sensor networks cooperating with UAVs for mission-critical management," *IEEE Wireless Communications*, vol. 15, no. 6, pp. 38–46, 2008.
- [25] S. K. Teh, L. Mejias, P. Corke, and W. Hu, "Experiments in integrating autonomous uninhabited aerial vehicles(uavs) and wireless sensor networks," in *Proceedings of the Australasian Conference on Robotics and Automation (ACRA 08, The Australian Robotics and Automation Association Inc)*, Canberra, Australia, 2008, <https://eprints.qut.edu.au/15536/>.
- [26] A. Khan, D. Schaefer, L. Tao et al., "Low power greenhouse gas sensors for unmanned aerial vehicles," *Remote Sensing*, vol. 4, no. 5, pp. 1355–1368, 2012.
- [27] S. Illingworth, G. Allen, C. Percival et al., "Measurement of boundary layer ozone concentrations on-board a Skywalker unmanned aerial vehicle," *Atmospheric Science Letters*, vol. 15, no. 4, pp. 252–258, 2014.
- [28] J.-D. M. M. Biomo, T. Kunz, and M. St-Hilaire, "An enhanced Gauss-Markov mobility model for simulations of unmanned aerial ad hoc networks," in *Proceedings of the 7th IFIP Wireless and Mobile Networking Conference, WMNC 2014*, Vilamoura, Portugal, May 2014.
- [29] W. Wang, X. Guan, B. Wang, and Y. Wang, "A novel mobility model based on semi-random circular movement in mobile ad hoc networks," *Information Sciences*, vol. 180, no. 3, pp. 399–413, 2010.
- [30] D. Orfanus and E. P. De Freitas, "Comparison of UAV-based reconnaissance systems performance using realistic mobility models," in *Proceedings of the 2014 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, ICUMT 2014*, pp. 248–253, Saint Petersburg, Russia, October 2014.
- [31] E. Kuiper and S. Nadjm-Tehrani, "Mobility models for UAV group reconnaissance applications," in *Proceedings of the 2nd International Conference on Wireless and Mobile Communications (ICWMC '06)*, pp. 33–43, July 2006.
- [32] Y. Wan, K. Namuduri, Y. Zhou, and S. Fu, "A smooth-turn mobility model for airborne networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 7, pp. 3359–3370, 2013.
- [33] J. Sanchez-Garcia, J. M. Garcia-Campos, S. L. Toral, D. G. Reina, and F. Barrero, "A Self Organising Aerial Ad Hoc Network Mobility Model for Disaster Scenarios," in *Proceedings of the 8th International Conference on Developments in eSystems Engineering, DeSe 2015*, pp. 35–40, December 2015.
- [34] O. Briante, V. Loscri, P. Pace, G. Ruggeri, and N. R. Zema, "COMVIVOR: An Evolutionary Communication Framework Based on Survivors' Devices Reuse," *Wireless Personal Communications*, vol. 85, no. 4, pp. 2021–2040, 2015.
- [35] O. Bouachir, A. Abrassart, F. Garcia, and N. Larriue, "A mobility model for UAV ad hoc network," in *Proceedings of the 2014 International Conference on Unmanned Aircraft Systems, ICUAS 2014*, pp. 383–388, Orlando, FL, USA, May 2014.
- [36] P. Basu, J. Redi, and V. Shurbanov, "Coordinated flocking of UAVS for improved connectivity of mobile ground nodes," in *Proceedings of the MILCOM 2004 - 2004 IEEE Military Communications Conference*, pp. 1628–1634, November 2004.
- [37] R. P. Foundation, "Raspberry pi," <https://www.raspberrypi.org/>.
- [38] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: a system for autonomous flight using onboard computer vision," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '11)*, pp. 2992–2997, IEEE, Shanghai, China, May 2011.
- [39] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: a micro aerial vehicle design for autonomous flight using onboard computer vision," *Autonomous Robots*, vol. 33, no. 1-2, pp. 21–39, 2012.
- [40] D. Industries, Grovepi, <http://www.dexterindustries.com/grovepi/>.
- [41] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences. An International Journal*, vol. 237, pp. 82–117, 2013.
- [42] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micromachine and Human Science*, pp. 39–43, Nagoya, Japan, October 1995.
- [43] R. Core Team, "R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria," 2016, <http://www.R-project.org/>.
- [44] M. L. Stein, *Interpolation of Spatial Data: Some Theory for Kriging*, Springer, New York, NY, USA, 1999.
- [45] U. S. E. P. Agency, "Air Quality Index," 2015, <http://cfpub.epa.gov/airnow/index.cfm?action=aqibasics.aqi>.
- [46] O. Alvear, C. T. Calafate, E. Hernández, J.-C. Cano, and P. Manzoni, "Mobile Pollution Data Sensing Using UAVs," in *Proceedings of the 13th International Conference on Advances in Mobile Computing and Multimedia, MoMM 2015*, pp. 393–397, Brussels, Belgium, December 2015.

Research Article

Intelligent Controller Design for Quad-Rotor Stabilization in Presence of Parameter Variations

Oualid Doukhi, Abdur Razzaq Fayjie, and Deok Jin Lee

Smart Autonomous System Laboratory, Kunsan National University, Jeollabuk, Republic of Korea

Correspondence should be addressed to Deok Jin Lee; deokjlee@kunsan.ac.kr

Received 24 March 2017; Accepted 5 June 2017; Published 18 July 2017

Academic Editor: Guizhen Yu

Copyright © 2017 Oualid Doukhi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The paper presents the mathematical model of a quadrotor unmanned aerial vehicle (UAV) and the design of robust Self-Tuning PID controller based on fuzzy logic, which offers several advantages over certain types of conventional control methods, specifically in dealing with highly nonlinear systems and parameter uncertainty. The proposed controller is applied to the inner and outer loop for heading and position trajectory tracking control to handle the external disturbances caused by the variation in the payload weight during the flight period. The results of the numerical simulation using gazebo physics engine simulator and real-time experiment using AR drone 2.0 test bed demonstrate the effectiveness of this intelligent control strategy which can improve the robustness of the whole system and achieve accurate trajectory tracking control, comparing it with the conventional proportional integral derivative (PID).

1. Introduction

The payload variation has been playing an important role in the field of transportation. The control for the unmanned aerial vehicles (UAV) becomes more challenging when it involves the payload variation over time. With the increasing popularity of autonomous UAVs in the field of military and civilian such as surveillance, disaster monitoring, rescue mission, firefighting, and package delivery [1], where carrying and transporting external payload is essential to complete the mission successfully, the safe and fast transportation of fragile or dangerous payloads by UAVs has become an interesting and important topic for the recent researchers in the field. The control of such UAVs requires the UAV stability and constant performance with the changing payload weight during the flight. But these UAVs are inherently unstable, nonlinear, and multivariable with complex and highly coupled dynamics [2]. It is also very sensitive to unmolding dynamics, parametric uncertainties, and external disturbances. Thus, it makes the control design for UAVs with payload variation very critical. The critical design of such controllers leads the field of control theory to introduce new intelligent types of controllers [3].

Recent researches on the same field show different control methodologies. Bouabdallah and Siegwart present the

real-time tests with an integral backstepping controller for an autonomous vehicle [4] and Boudjedir et al. mentioned an Adaptive Neural Control technique at the presence of sinusoidal disturbance [5] which is tested only in a simulation environment. Even though there exists another controller depicting the adaptive nature using sliding mode, control in the presence of external disturbances is described in [6], but it holds the simulation environment only. The work in [7] describes a Fuzzy PID for quadrotor with space fixed-point position control in a real-time flight test. Kim and Lee extend their control idea to nonlinear attitude stabilization and tracking control for an autonomous hexarotor vehicle [8]. A new approach is taken in the field of control by the design of an Intelligent controller for unmanned quadrotor [9] which seems to be promising. But the discussed works lack the analysis of the payload weight variation effects and the performance of the UAV in presence of such variations on the weight. This phenomenon is an integral part of the environments like mountains or unstructured areas for real UAV applications where UAV needs to drop the payload gradually without landing. The most practical example to view the importance of such system is the application in firefighting. Other applications like package delivery and mail delivery involve maximizing the number of delivery within

an area where package or mail drop from air helps to make delivery easier. In hazard situations like fire control, such system can be used as continuous spray of water or fire control gases. The application also extends to the field of agriculture with medicine spray in a crop field.

It is extremely painful that there exists a few successful research focused on payload weight variation and its effect on the UAV performance. These studies include a fixed wing UAV that has been used for payload dropping mission [10]. It lacks the stationary flight capability which leads the researcher to focus on Vertical Take-Off and Landing (VTOL) vehicles such as quadrotor UAV which offers the ability to fly in any direction and even to stop their movement staying hovering above a specific target.

In [11] a robust adaptive control is designed for quadrotor payload add and drop applications in simulation environment. Another research presented a payload drop application of unmanned quadrotor helicopter using Gain-Scheduled PID and model predictive control techniques in real-time experiment [12] but the work analyzes the effect only in altitude. Muhammad presented the simulation work with the PID controller for payload drop of a quadrotor UAV [13]. To handle such complex UAV systems to unload package or to drop payload without landing, a robust flight controller is needed which must be intelligent enough to keep the UAV stable in different circumstances. The intelligent controller must adapt itself in a short duration of time with the sudden changes in the payload which enable the UAV to be stabilized. The design of such controllers aims to avoid the damage of the system and guarantees the safety operation for the surrounding environment and people present.

This work studies the effects of payload weight variation and analyzes the behavior of the UAV in a 3-dimensional space (x, y, z) along with the attitude (roll, pitch, and yaw angles) instead of the single axis z (altitude) consideration of most of the present works. The work contributes to the design of a robust intelligent self-tuning PID controller based on fuzzy logic to ensure the trajectory tracking performance in the presence of payload weight variations. The proposed control idea is tested both in the simulation as well as in real-time environment.

The paper introduces the existing research works which are similar to the topic of the paper, the pros and cons of such control ideas, the problem definitions, and the proposed idea in Section 1. The next part describes the dynamic model of the quadrotor in Section 2 and the controls development is presented in Section 3. Section 4 presents the simulation environment and asymptotic tracking results obtained in that environment which is further extended to real-time environment and the real flight test results are presented in Section 5. Finally, the paper concludes with the promising result and uses of proposed idea in near future to handle payload variation on a UAV effectively.

2. Description and Dynamics of the Quadrotor

The work uses a quadrotor, AR Drone 2.0, that was unveiled at CES Las Vegas 2012. It comes with 720p camera and the onboard sensors (like ultrasonic altimeter and 3-axis

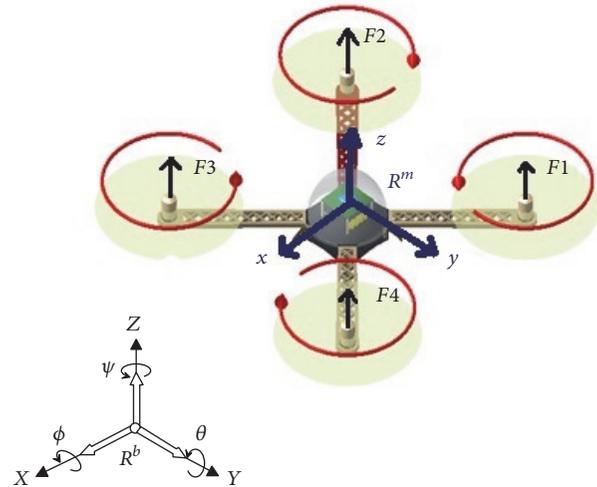


FIGURE 1: Quadrotor configuration frame scheme with body-fixed and inertia frames.

gyroscope, along with a 3-axis accelerometer and magnetometer) which allow the quadrotor for greater control. The wifi communication with 802.11n standard provides better communication with the ground station. The rotors are powered by 15 watts and brushless motors are powered by an 11.1 Volt lithium polymer battery. This provides approximately 12 minutes of flight time at a speed of 5 m/s (11 mph). In order to design a flight controller, one must first understand the quadrotor movements, its dynamics, and consequently its dynamic equations. This understanding is necessary not just for the design of the controller, but also to ensure that the simulations of the vehicle behavior are closer to the reality when the command is applied.

The quadrotor is classified in the category of the most complex flying systems given the number of physical effects that affect its dynamics, namely, aerodynamic effects, gravity, gyroscopic effects, friction, and moment of inertia. The schematic configuration of a quadrotor we adopted in this study is shown in Figure 1. Four rotors are usually placed at the ends of two orthogonal axes. The rotation for two propellers needed to be in a clockwise direction while the other two are in a counterclockwise direction. The modeling of flying robots is a delicate task as the dynamics of the system are strongly nonlinear and fully coupled. In order to understand the model dynamics developed in this work, the working hypotheses are given below [14]:

- (1) The structure of the quadrotor is assumed to be rigid and symmetric.
- (2) The propellers are rigid.
- (3) Thrust and drag forces are proportional to the square of the rotors speed.
- (4) The center of mass and origin of the coordinate system related to the quadrotor structure coincide.

To evaluate the mathematical model of the quadrotor, we use two reference frames, a fixed frame linked to the ground called inertia frame \mathbf{R}^b and another mobile frame \mathbf{R}^m with its

TABLE 1: Quadrotor parameters.

Symbol	Quantity	Value
k_1	Thrust factor	$8.048 \times 10^{-6} \text{ N}\cdot\text{S}^2$
k_2	Drag coefficient	$2.43 \times 10^{-7} \text{ N}\cdot\text{m}\cdot\text{s}^2$
I_{xx}	Moment of inertia along x direction	$0.002237568 \text{ kg}\cdot\text{m}^2$
I_{yy}	Moment of inertia along y direction	$0.002985236 \text{ kg}\cdot\text{m}^2$
I_{zz}	Moment of inertia along z direction	$0.00480374 \text{ kg}\cdot\text{m}^2$
l	Center distance between the gravity of quadrotor and each propeller	0.1785 m
m	Total mass of the quadrotor	0.450 kg
J_r	Rotor inertia	$2.029585 \times 10^{-5} \text{ kg}\cdot\text{m}^2$
g	Gravity	9.81 m/s^2

origin at the center of mass of the quadrotor. By varying the rotor speeds, the thrust forces $F_{i=1\dots 4}$ can be changed which results in the six principal motions as follows:

- (1) Roll rotation ϕ followed by translation along O_y
- (2) Pitch rotation θ followed by translation along O_x
- (3) Yaw rotation ψ
- (4) Translation along O_z (the altitude).

Let $q = [P, r]^T \in R^6$ be the generalized coordinates for the quadrotor where $P = [x, y, z]^T$ denotes the absolute position in the inertial frame \mathbf{R}^b , and $r = [\phi, \theta, \psi]^T$ is the vector consists of three Euler angles ϕ , θ and ψ . $\vartheta = [\Omega_1, \Omega_2, \Omega_3]^T$ denotes the vector containing angular velocities of roll, pitch, and yaw with respect to the body-fixed frame \mathbf{R}^m . The rotation matrix from body frame to earth frame can be obtained as follows:

$$R = \begin{bmatrix} c\phi c\theta & s\phi s\theta c\psi - s\psi c\phi & c\phi s\theta c\psi + s\psi s\phi \\ s\phi c\theta & s\phi s\theta s\psi + c\psi c\theta & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}, \quad (1)$$

where $s(\cdot)$ and $c(\cdot)$ are abbreviations for $\sin(\cdot)$ and $\cos(\cdot)$ functions, respectively.

Using the Newton-Euler formulation, the rotational and translational dynamics are written in the following equation with the corresponding parameters presented in Table 1:

$$\begin{aligned} \ddot{x} &= (c_\phi s_\theta c_\psi + s_\phi s_\psi) \frac{1}{m} U_1, \\ \ddot{y} &= (c_\phi s_\theta s_\psi - s_\phi c_\psi) \frac{1}{m} U_1, \\ \ddot{z} &= -g + (c_\phi c_\theta) \frac{1}{m} U_1, \\ \ddot{\phi} &= \dot{\theta} \dot{\psi} \left(\frac{I_{yy} - I_{zz}}{I_{xx}} \right) - \frac{J_r}{I_{xx}} \dot{\theta} \Omega_d + \frac{1}{I_{xx}} U_2, \\ \ddot{\theta} &= \dot{\phi} \dot{\psi} \left(\frac{I_{zz} - I_{xx}}{I_{yy}} \right) + \frac{J_r}{I_{yy}} \dot{\phi} \Omega_d + \frac{1}{I_{yy}} U_3, \\ \ddot{\psi} &= \dot{\theta} \dot{\phi} \left(\frac{I_{xx} - I_{yy}}{I_{zz}} \right) + \frac{1}{I_{zz}} U_4, \end{aligned} \quad (2)$$

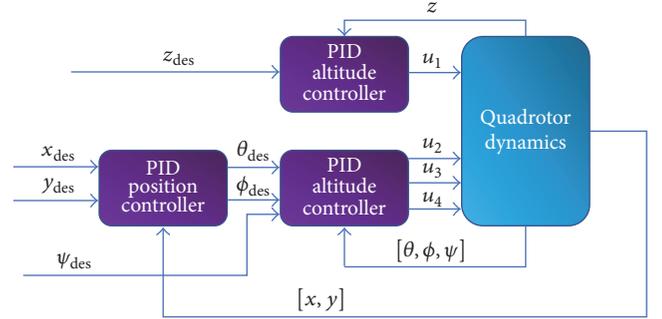


FIGURE 2: PID controller structure.

where

$$\Omega_d = \omega_2 + \omega_4 - \omega_1 - \omega_4 \quad (3)$$

and U_i ($i = 1, 2, 3, 4$) is the virtual control input defined as follows:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} K_1 & K_1 & K_1 & K_1 \\ 0 & -IK_1 & 0 & IK_1 \\ -IK_1 & 0 & IK_1 & 0 \\ K_2 & -K_2 & K_2 & -K_2 \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}. \quad (4)$$

3. Control Development

The work develops a self-tuning PID controller for position and heading trajectory tracking control to deal with the mass variance of the quadrotor when the payload is gradually changed.

3.1. Design of PID Controller. The proportional integral derivative (PID) controller is designed to ensure the trajectory tracking capability of $\{\psi\}$ and $\{x, y, z\}$ along the desired yaw angle trajectory $\{\psi_d\}$ and $\{x_d, y_d, z_d\}$ position as illustrated in Figure 2.

The based control laws of PID controller for position and heading take the following form:

$$\begin{aligned} U_4 &= K_{p\psi} \cdot e_\psi + K_{d\psi} \cdot \dot{e}_\psi + K_{i\psi} \cdot \int e_\psi \cdot dt, \\ U_{1x} &= K_{px} \cdot e_x + K_{dx} \cdot \dot{e}_x + K_{ix} \cdot \int e_x \cdot dt, \\ U_{1y} &= K_{py} \cdot e_y + K_{dy} \cdot \dot{e}_y + K_{iy} \cdot \int e_y \cdot dt, \\ U_1 &= K_{pz} \cdot e_z + K_{dz} \cdot \dot{e}_z + K_{iz} \cdot \int e_z \cdot dt, \end{aligned} \quad (5)$$

where

$$\begin{aligned} e_\psi &= \psi_{\text{des}} - \psi, \\ e_x &= x_{\text{des}} - x, \\ e_y &= y_{\text{des}} - y, \\ e_z &= z_{\text{des}} - z \end{aligned} \quad (6)$$

and K_p, K_d, K_i are parameters tuned without payload onboard. U_{1x} and U_{1y} are virtual control inputs used to generate the desired roll ϕ_d and pitch θ_d angles as illustrated in Figure 2:

$$\begin{aligned} \phi_d &= \frac{1}{g} (U_{1x} \sin \psi_{\text{des}} - U_{1y} \cos \psi_{\text{des}}), \\ \theta_d &= \frac{1}{g} (U_{1x} \cos \psi_{\text{des}} - U_{1y} \sin \psi_{\text{des}}). \end{aligned} \quad (7)$$

3.2. Fuzzy Logic for Self-Tuning of PID. Fuzzy logic is a tool for reasoning propositions that can be manipulated with mathematical precepts. It has been suggested by researchers that measurements, process modeling, and control can never be exact for real and complex processes. Also, there are uncertainties such as incompleteness, randomness, and ignorance of data in the process model [15].

The seminal work of Zadeh introduced the concept of fuzzy logic to model human reasoning from imprecise and incomplete information by giving definitions to vague terms and allowing construction of a rule base (Zadeh, 1965, 1973). Fuzzy logic can incorporate human experiential knowledge and give it an engineering flavor to model and control such ill-defined systems with nonlinearity and uncertainty. The fuzzy logic methodology usually deals with reasoning and inference on a higher level, such as semantic or linguistic. Fuzzy logic requires knowledge in order to reason which is provided by a person who knows the process or machine (the expert). One of the most important components of fuzzy logic is membership function. Membership functions group input data into sets, such as temperatures that are too cold, motor speeds that are acceptable, and distances that are too far. The fuzzy controller (see Figure 4) assigns the input data a grade between 0 and 1 based on how well it fits into each membership function [16].

Mathematically a fuzzy set is described as a set A in X . The set A is characterized by a membership function $\mu_A(x)$

associated with each point in x , a real number in the interval $[0, 1]$, with the values of $\mu_A(x)$ at x representing the grade of membership of x in A .

The function that characterizes the fuzziness of a fuzzy set A in X , which associates each point in X with a real number in the interval $[0, 1]$, is known as membership function.

Fuzzification and Defuzzification. The process that allows converting a numeric value into a fuzzy input is called fuzzification.

Defuzzification is the reverse process of fuzzification. Mathematically, the defuzzification of a fuzzy set is the process of conversion of a fuzzy quantity into a crisp value. This is necessary when a crisp value is to be provided from a fuzzy system to the user.

Inference Mechanism. Inference is the process of formulating a nonlinear mapping from a given input space to output space. The mapping then provides a basis from which decisions can be made. The process of fuzzy inference involves all the membership functions, operators, and if-then rules [16].

3.3. Self-Tuning of the Designed PID Controller. Traditional PID designed in Section 3.1 comes with a lot of limitations. For example, its three parameters are tuned for a specific operating condition such as with constant payload or without payload. Therefore, such mass uncertainty may significantly affect the fly control performance. To solve this problem of designed PID, a self-tuning mechanism based on fuzzy logic is developed which has the capability of adjusting the parameters of designed PID online which enables the previously developed PID to handle the sudden or gradual change of the payload weight. We know that the weight change practically has a big and inevitable effect not only in the altitude stability of the quadrotor but also in x - y position. More specifically, in the hovering condition, a quadrotor with its onboard load is seen as a heavy body [13], thus requiring more upward thrust to attain the desired altitude. Once the payload is dropped or gradually changed, the quadrotor will reach a higher altitude, thus, producing overshooting on its desired height. The self-tuning intelligent PID controller must monitor this overshoot and quickly compensates for this and return the quadrotor to its desired altitude by lowering the upward thrust by slowing the speed of the brushless DC motor (BLDC) or in the opposite way when the payload is added to compensate the undershoot.

The self-tuning mechanism is designed for automatic-tuning of each of PID controller parameters so that the system becomes robust to the mass uncertainty. The inputs to the fuzzy mechanism are error and change in the error. It produces the output of the PID parameters " K_p ", " K_i ", " K_d " as presented in Figure 3.

3.4. Design of Membership Function (MF). The work designed five triangular membership functions nl, ns, ze, ps, and pl which represent negative large, negative simple, zero, positive simple, and positive large for each input variable (see Figure 5) and seven triangular membership functions pvs,

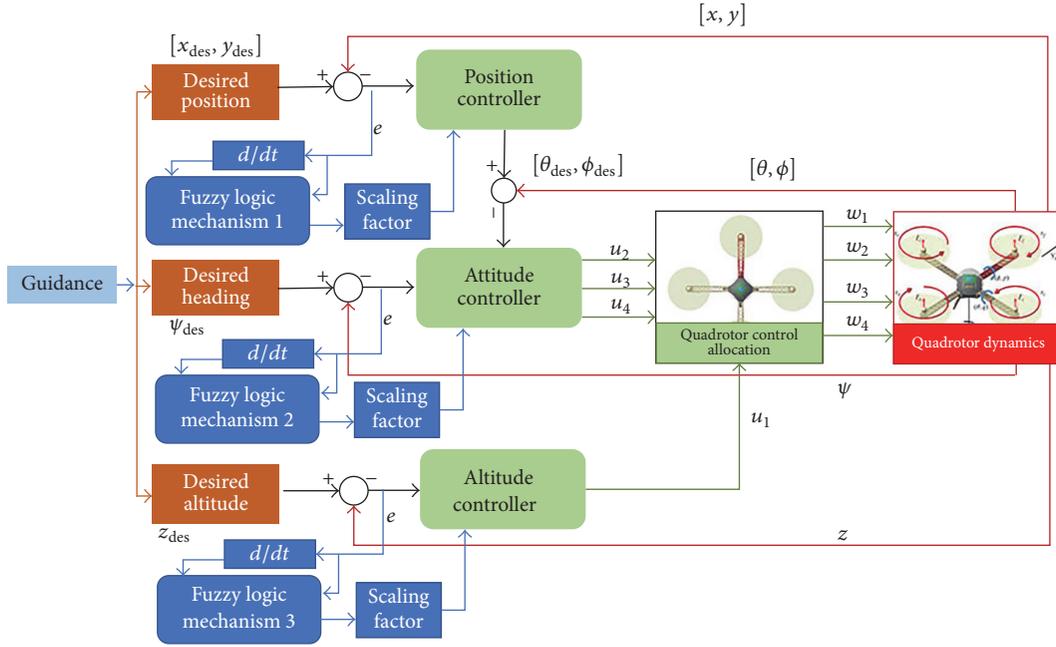


FIGURE 3: Overall structure for the developed control mechanism.

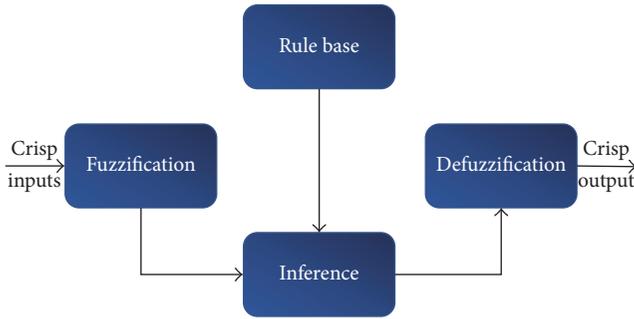


FIGURE 4: Fuzzy structure.

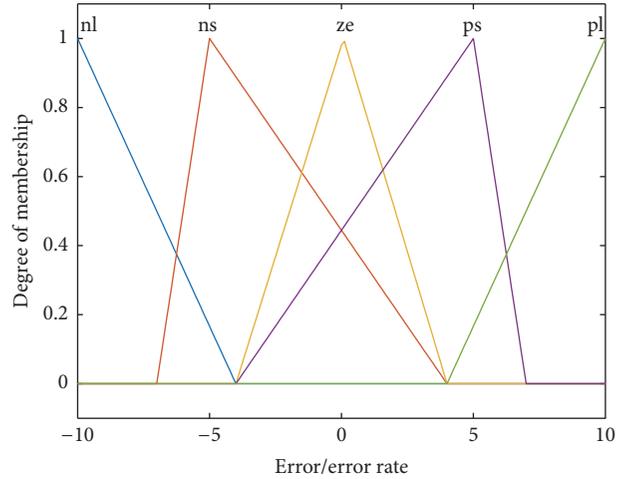


FIGURE 5: Membership functions designed for the inputs.

ps, pms, pm, pml, pl, and pvl for each output variables (see Figures 6–8). Based on experience, 25 fuzzy rules are defined for all possible combinations of the inputs. The “centroid method” is used for defuzzification. The three-dimensional surface of chosen fuzzy rules for the controller design is shown in Figures 9–11. Fuzzy controller is made of “if-then” rules. The selected rules for the quadrotor control process are shown in Tables 2–4. The equation for the control algorithm is given as

$$U(t) = U(t)_{SP} + U(t)_{sd} + U(t)_{SI}, \quad (8)$$

where the control signals self-tuning proportional, self-tuning Integral, and self-tuning derivative are

$$\begin{aligned} U(t)_{SP} &= Sk_p F_p e(t), \\ U(t)_{sd} &= Sk_d F_d \dot{e}(t), \\ U(t)_{SI} &= Sk_i F_i \int e(t) dt, \end{aligned} \quad (9)$$

where F_p, F_d and F_i are scaling factors; hence, three fuzzy controller outputs are

$$\begin{aligned} Sk_p &= f_p \left(e, \frac{de_p}{dt} \right), \\ Sk_d &= f_d \left(e, \frac{de_d}{dt} \right), \\ Sk_i &= f_i \left(e, \frac{de_i}{dt} \right). \end{aligned} \quad (10)$$

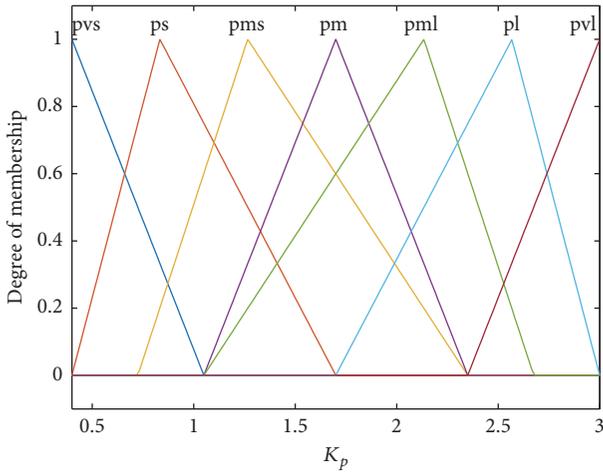


FIGURE 6: Membership functions designed for the output k_p .

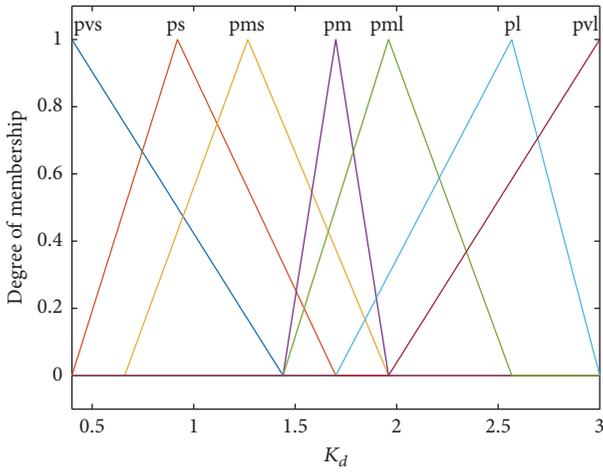


FIGURE 7: Membership functions designed for the output k_d .

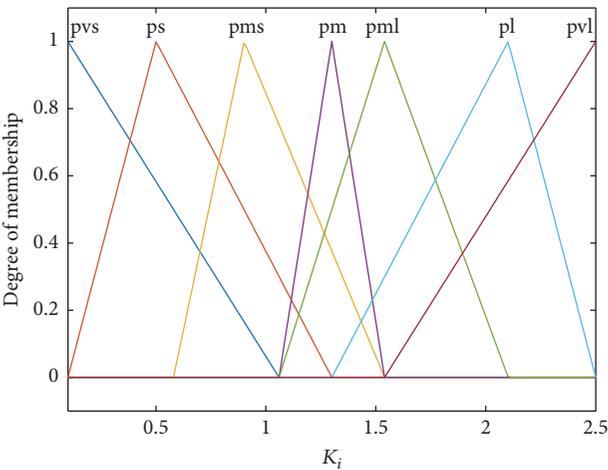


FIGURE 8: Membership functions designed for the output k_i .

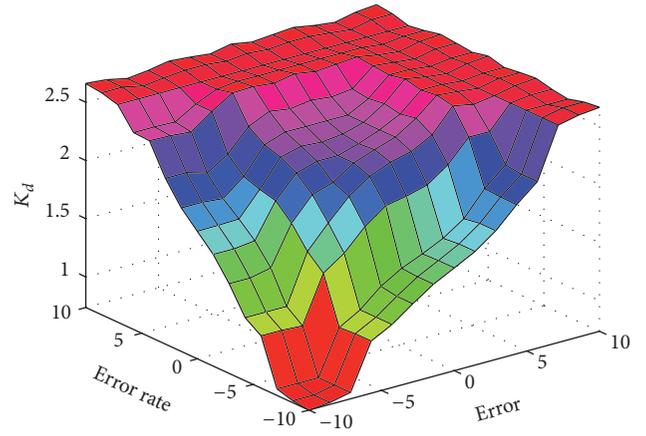


FIGURE 9: Error and derivative error for the output k_d .

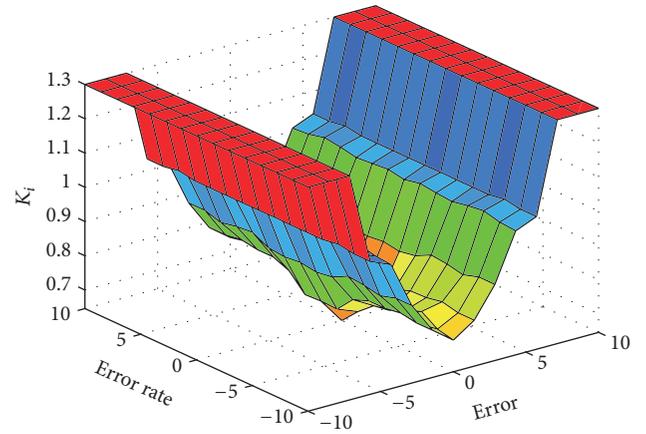


FIGURE 10: Error and derivative error for the output k_i .

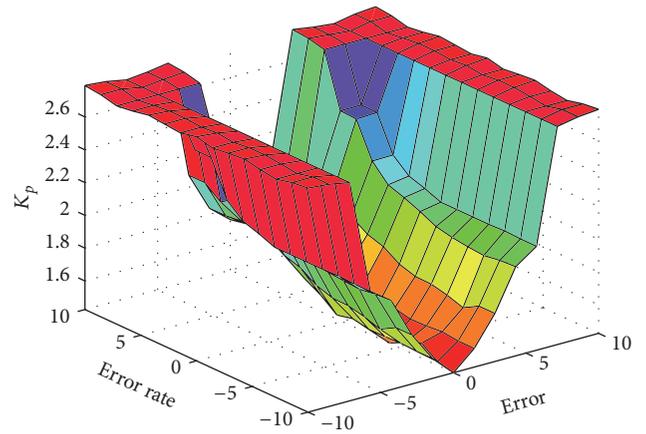


FIGURE 11: Error and derivative error for the output k_p .

Therefore, the self-tuning control input can be written as

$$\begin{aligned}
 U(t) = & f_p \left(e, \frac{de_p}{dt} \right) F_p e(t) + f_d \left(e, \frac{de_d}{dt} \right) F_d e(t) \\
 & + f_i \left(e, \frac{de_i}{dt} \right) F_i e(t).
 \end{aligned}
 \tag{11}$$

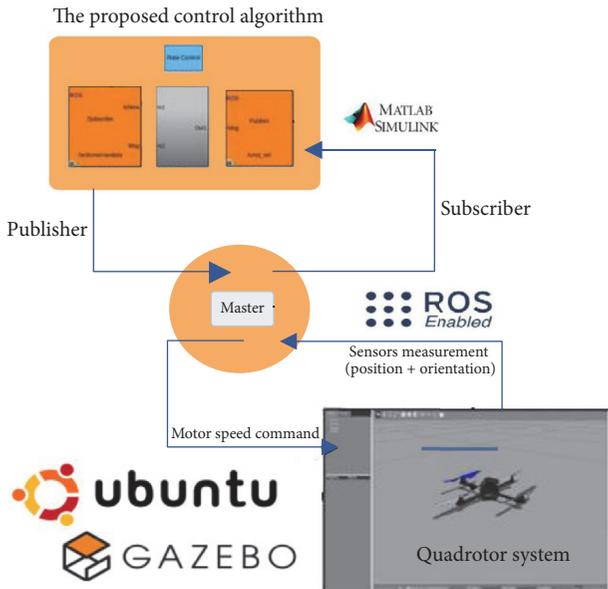


FIGURE 12: Simulation environment configuration.

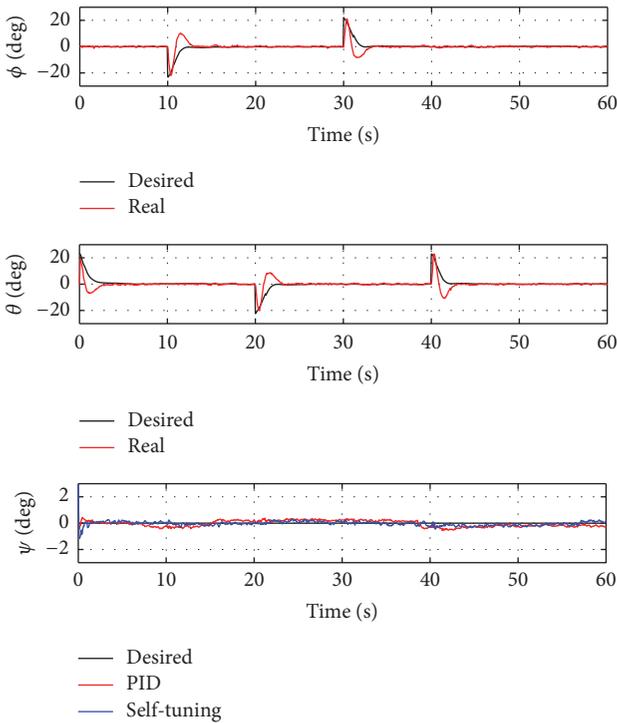


FIGURE 13: Response curves of quadrotor UAV attitude (ϕ, θ, ψ).

4. Simulation Setup and Results

Simulations have been conducted using robot operating system (ROS) interfaced with Matlab/Simulink and gazebo physic engine simulator as illustrated in Figure 12, in order to verify the efficiency of the proposed self-tuning PID controller for heading and position control and benchmark with a regular PID.

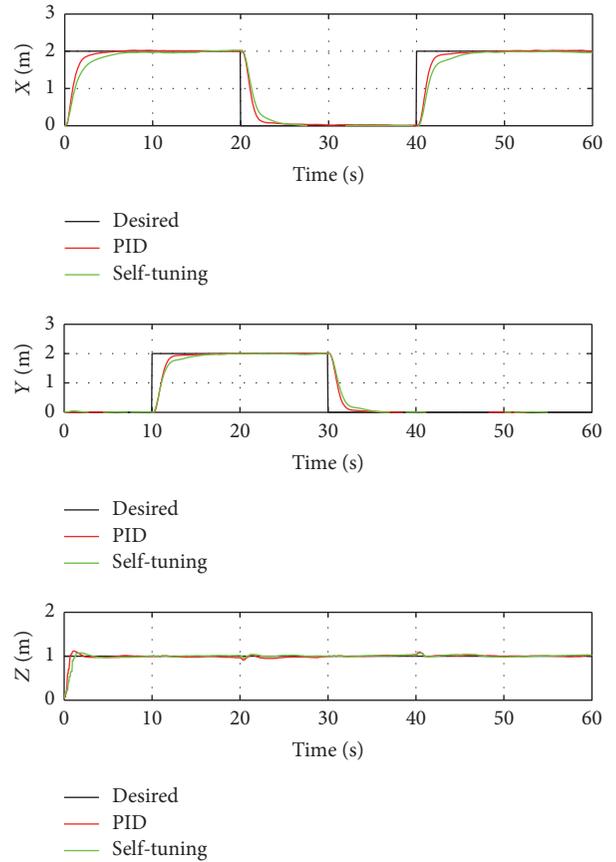


FIGURE 14: Response curves of quadrotor UAV position (x, y, z).

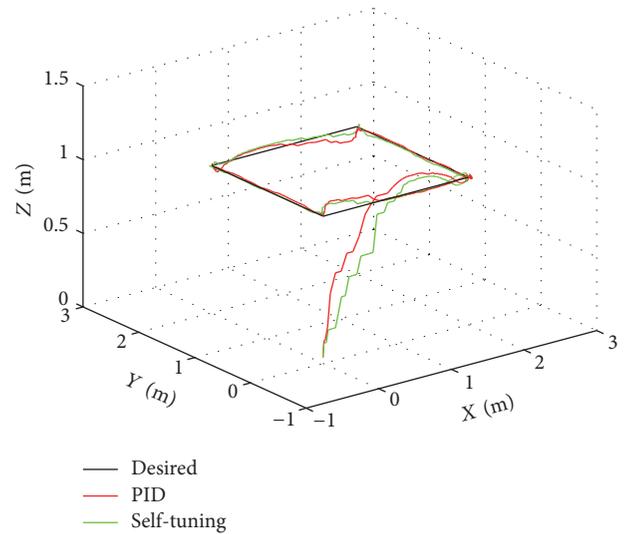


FIGURE 15: Three-dimensional response curves of quadrotor UAV position (x, y, z).

Firstly PID parameters are tuned in normal case quadrotor without payload; from Figures 13–15, we can see that the proposed self-tuning controller gives us very good result in terms of trajectory tracking performances. In the next section to demonstrate the effectiveness of the proposed controller against adverse condition, like payload weight variation, we

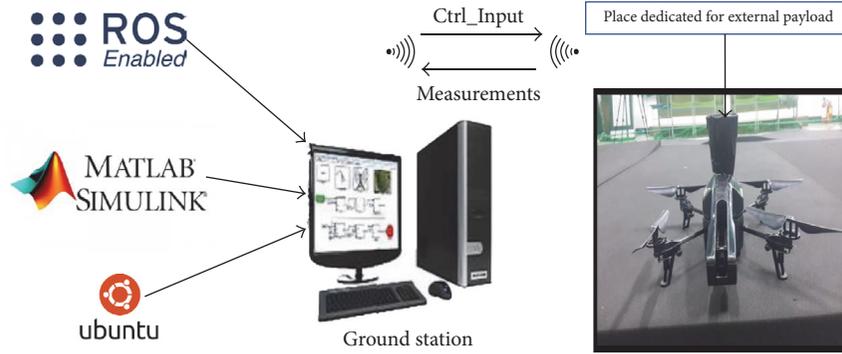


FIGURE 16: Experimental environment setup.

TABLE 2: Fuzzy rule for k_p .

de/e	NL	NS	ZE	PS	PL
NL	PVL	PVL	PVL	PVL	PVL
NS	PML	PML	PML	PL	PVL
ZE	PVS	PVS	PS	PMS	PMS
PS	PML	PML	PML	PL	PVL
PL	PVL	PVL	PVL	PVL	PVL

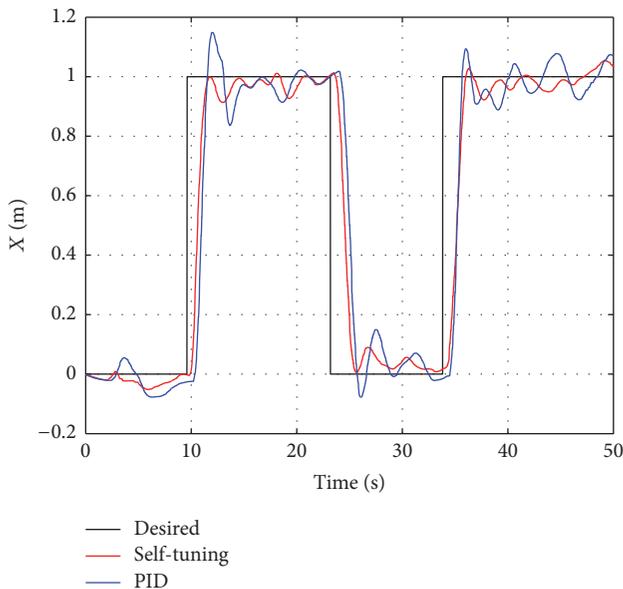


FIGURE 17: Quadrotor x position under payload with variable weight.

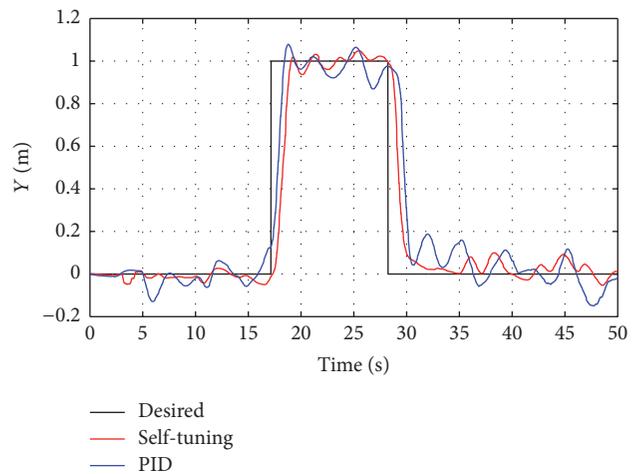


FIGURE 18: Quadrotor Y position under payload with variable weight.

need to do some experimental tests; we change the weight gradually and then we analyze the effect of this change in the position tracking performance and we compare it with conventional PID.

5. Experiment Setup and Results

In this section to validate the position and heading trajectory tracking performance experiment tests (the experimental

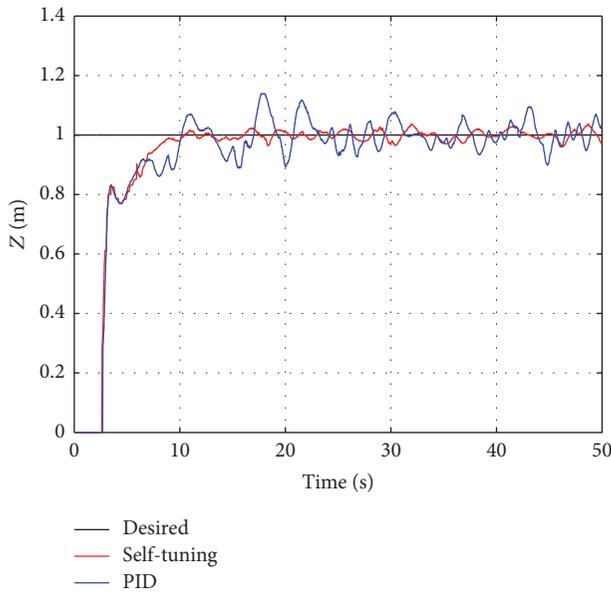
video is available at the URL <https://www.youtube.com/channel/UC6eedfqCWGpj1YqANGdzfIg>) are implemented on the Ardrone 2.0 testbed as presented in Figure 16. Firstly the quadrotor is on the ground with initial payload of 20 g onboard; after takeoff the quadrotor was ordered to follow square trajectory, and then during the fly maneuver we change gradually the weight by adding external load till we reach the maximum load supported by our system which is about 60 g; from Figures 17–21 we can see that the the self-tuning PID algorithm based on fuzzy logic that automatically adjust the parameter gain within rage $k_p, K_d \in [0.4 \ 3]$, $K_i \in [0.1 \ 2.5]$ is able to make the quadrotor reach the desired trajectory, without overshoot and with

TABLE 3: Fuzzy rule for k_d .

de/e	NL	NS	ZE	PS	PL
NL	PVS	PML	PM	PL	PVL
NS	PMS	PML	PL	PVL	PVL
ZE	PM	PL	PL	PVL	PVL
PS	PML	PVL	PVL	PVL	PVL
PL	PVL	PVL	PVL	PVL	PVL

TABLE 4: Fuzzy rule for k_i .

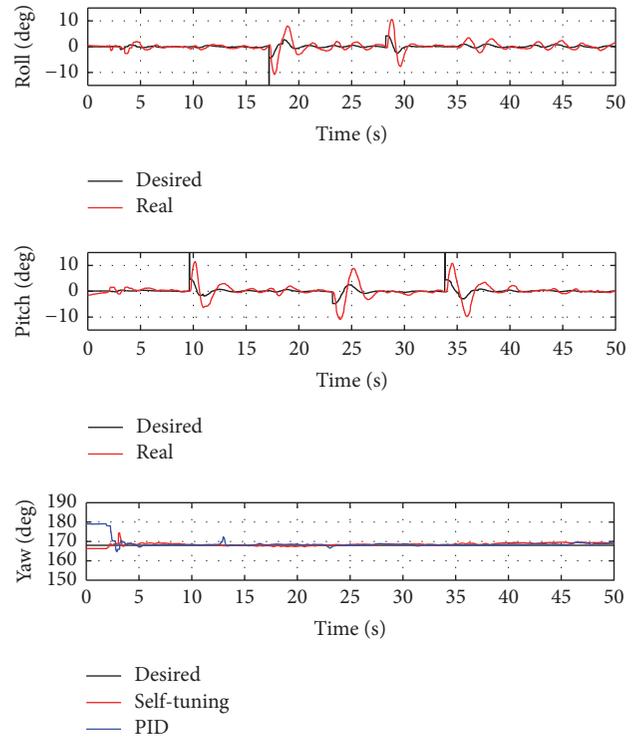
de/e	NL	NS	ZE	PS	PL
NL	PM	PM	PM	PM	PM
NS	PMS	PMS	PMS	PMS	PMS
ZE	PS	PS	PVS	PS	PS
PS	PMS	PMS	PMS	PMS	PMS
PL	PM	PM	PM	PM	PM

FIGURE 19: Quadrotor z position under payload with variable weight.

slightly oscillating motion compared to the conventional PID controller system which presents about 19 cm of overshoot in the altitude and position, because the control parameters $k_p = 1.5$ $K_d = 0.8$ $K_i = 0.3$ are tuned in the initial case with constant payload.

6. Conclusion

This paper presents a self-tuning PID control algorithm to achieve quadrotor aerial vehicle's performance with a variable payload. The comparison between the conventional PID and self-tuning PID base on fuzzy logic has been conducted; the result shows that both control methodologies are acceptable in the case of no variation in the payload weight, experimental results demonstrate the superiority of the proposed self-tuning PID comparing to the PID in case of variable

FIGURE 20: Response curves of quadrotor UAV attitude (ϕ , θ , ψ).

payload weight, and it provides good performance in terms of disturbance reduction and trajectory tracking.

Conflicts of Interest

The received funding did not lead to any conflicts of interest regarding the publication of this manuscript.

Acknowledgments

This research was supported by Unmanned Vehicles Advanced Core Technology Research and Development

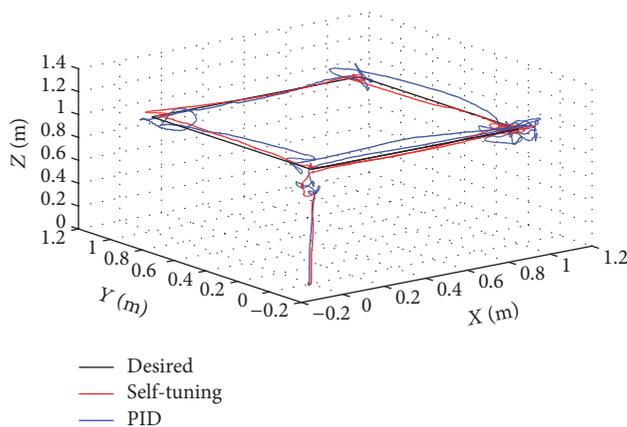


FIGURE 21: Three-dimensional response curves of quadrotor UAV position (x, y, z) with variable payload weight.

Program through the National Research Foundation of Korea (NRF) and Unmanned Vehicle Advanced Research Center (UVARC) funded by the Ministry of Science, ICT and Future Planning, Republic of Korea (NRF-2016M1B3A1A01937245), and was also supported by the Human Resource Training Program for Regional Innovation and Creativity through the Ministry of Education and National Research Foundation of Korea (no. 2014H1C1A10166733).

References

- [1] J. M. Maddalon, K. J. Hayhurst, D. M. Koppen, J. M. Upchurch, and A. T. Morris, "Perspectives on unmanned aircraft classification for civil airworthiness standards," *National Aeronautics and Space Administration, Langley Research Center Hampton, Virginia*, pp. 2199–23681, 2013.
- [2] S. Gardecki, W. Giernacki, J. Goslinski, and A. Kasinski, "An adequate mathematical model of four-rotor flying robot in the context of control simulations," *Journal of Automation, Mobile Robotics & Intelligent Systems*, vol. 8, no. 2, 2014.
- [3] W. Li, C. Yang, Y. Jiang, X. Liu, and C. Su, "Motion Planning for Omnidirectional Wheeled Mobile Robot by Potential Field Method," *Journal of Advanced Transportation*, vol. 2017, Article ID 4961383, 11 pages, 2017.
- [4] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 153–158, San Diego, Calif, USA, November 2007.
- [5] H. Boudjedir, O. B. Fouad Yacef, and N. Rizoug, "Adaptive neural network for a quadrotor unmanned aerial vehicle," *International Journal in Foundations of Computer Science & Technology*, vol. 2, no. 4, 2012.
- [6] A. Modirrousta and M. Khodabandeh, "Adaptive robust sliding mode controller design for full control of quadrotor with external disturbances," in *Proceedings of the 2014 2nd RSI/ISM International Conference on Robotics and Mechatronics, ICRoM 2014*, pp. 870–877, Tehran, Iran, October 2014.
- [7] J. Ma and R. Ji, "Fuzzy PID for quadrotor space fixed-point position control," in *Proceedings of the 2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC '16)*, pp. 721–726, Harbin, China, July 2016.
- [8] H. Kim and D. J. Lee, "Nonlinear attitude stabilization and tracking control techniques for an autonomous hexa-rotor vehicle," in *Wireless Communications, Networking and Applications, Lecture Notes in Electrical Engineering*, Q. A. Zeng, Ed., vol. 348, pp. 1279–1292, Springer India, New Delhi, India, 2016.
- [9] B. S. Jun, J. YoungPil, P. J. Cho UkRae, D. J. Lee, and C. K. To, "Intelligent control system design of a unmanned quadrotor robot," *Applied Mechanics and Materials*, vol. 548-549, pp. 917–921, 2014.
- [10] G. S. Hadi, R. Varianto, B. Riyanto, and A. Budiyo, "Autonomous uav system development for payload dropping mission," *Journal of Instrumentation, Automation and Systems*, vol. 1, no. 2, 2014.
- [11] B. J. Emran, J. Dias, L. Seneviratne, and G. Cai, "Robust adaptive control design for quadcopter payload add and drop applications," in *Proceedings of the 34th Chinese Control Conference, CCC 2015*, pp. 3252–3257, Hangzhou, China, July 2015.
- [12] I. Sadeghzadeh, M. Abdolhosseini, and Y. M. Zhang, "Payload drop application of unmanned quadrotor helicopter using gain-scheduled PID and model predictive control techniques," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7506, no. 1, pp. 386–395, 2012.
- [13] M. A. Muflikhun, I. H. V. Gue, E. R. Magsino, and A. Y. Chua, "A Pid Controller for Payload Drop of A Quadrotor UAV," HUST, Hanoi, Vietnam, 2014.
- [14] M. Moussid, A. Sayouti, and H. Medromi, "Dynamic modeling and control of a hexarotor using linear and nonlinear methods," *International Journal of Applied Information Systems (IJ AIS) –ISSN: 2249-0868 Foundation of Computer Science FCS*, vol. 9, no. 5, pp. 9–17, 2015.
- [15] R. L. Galveza, E. P. Dadiosa, and A. A. Bandala, "Fuzzy logic based path planning for quadrotor," in *Proceedings of the 2015 Annual PAASE Meeting and Symposium De La Salle University*, Manila, Philippines, February 2015.
- [16] N. Siddique and H. Adeli, *Computational Intelligence Synergies of Fuzzy Logic, Neural networks and Evolutionary Computing*, Wiley, 2013.

Research Article

Conceptual Design of a Small Hybrid Unmanned Aircraft System

Umberto Papa,¹ Salvatore Ponte,² and Giuseppe Del Core¹

¹*Department of Science and Technology, University of Naples “Parthenope”, Naples, Italy*

²*Department of Industrial and Information Engineering, University of Campania “Luigi Vanvitelli”, Aversa, Italy*

Correspondence should be addressed to Umberto Papa; umberto.papa@uniparthenope.it

Received 3 November 2016; Revised 18 April 2017; Accepted 7 May 2017; Published 30 May 2017

Academic Editor: Guizhen Yu

Copyright © 2017 Umberto Papa et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

UAS (Unmanned Aircraft System) technologies are today extremely required in various fields of interest, from military to civil (search and rescue, environmental surveillance and monitoring, and entertainment). Besides safety and legislative issues, the main obstacle to civilian applications of UAS systems is the short time of flight (endurance), which depends on the equipped power system (battery pack) and the flight mission (low/high speed or altitude). Long flight duration is fundamental, especially with tasks that require hovering capability (e.g., river flow monitoring, earthquakes, devastated areas, city traffic monitoring, and archeological sites inspection). This work presents the conceptual design of a Hybrid Unmanned Aircraft System (HUAS), merging a commercial off-the-shelf quadrotor and a balloon in order to obtain a good compromise between endurance and weight. The mathematical models for weights estimation and balloon static performance analysis are presented, together with experimental results in different testing scenarios and complex environments, which show 50% improvement of the flight duration.

1. Introduction

Unmanned Aircraft Systems (UAS) play an important role in various military and civil applications, particularly for monitoring and surveillance of areas (urban traffic, coast guard patrolling, border patrolling, detection of illegal imports, archeological site prospection, etc.), climate research (weather forecast, river flow), agricultural studies, air composition and pollution studies, inspection of electrical power lines, monitoring gas or oil pipe lines, entertainment, and TV: their diffusion is mainly due to the capacity to perform dangerous, sensitive, environmentally critical, or dull tasks with low costs and increased manoeuvrability and survivability. Most civilian UAS missions require flying speeds lower than 50 kts (70 km/hr) at low altitudes, with hovering capability to perform close proximity inspection (e.g., air quality measurements at high spatial resolution [1], powerline inspection, subsurface geology and agriculture [2], mineral resource analysis, or incident control by police and fire services) [3–5]. Some missions also require significant flight duration time. Increasing endurance generally implies additional costs in terms of fuel consumption and airframe complexity, resulting

in reduced efficiency of payload and/or reduced range for size, mass, and financial cost.

Currently, a broad range of UAS exists, from small and lightweight fixed-wing aircrafts to rotor helicopters, large-wingspan airplanes, and quadcopters, generally providing persistence beyond the capabilities of manned vehicles [6]. Table 1 [7] classifies UAS with respect to mass, range, flight altitude, and endurance.

VTOL (Vertical Take-Off and Landing) aircrafts provide many advantages with respect to Conventional Take-Off and Landing (CTOL) vehicles, first of all the capability of hovering and the small area required for take-off and landing. Among VTOL aircrafts, such as conventional helicopters and crafts with rotors like the tilt-rotor and fixed-wing aircrafts with directed jet thrust capability, the quadrotor or quadcopter is often preferred, especially in the academic research on mini- or microsize UAS. Quadrotors are a good alternative to conventional rotorcrafts, due to their ability to hover and move without the complex systems of linkages and blade elements typical of a standard single-rotor vehicle [5] [8, Ch. 1]. By employing four rotors to generate differential thrust, the quadrotor gains flexibility, swift manoeuvrability, and

TABLE 1: Extract of UAV categories defined by UVS International (Unmanned Vehicle Systems Association). Figures on category name “mini” depend on different countries.

Category name	Mass [kg]	Range [km]	Flight altitude [m]	Endurance [hours]
Micro	<5	<10	250	1
Mini	<25/30/150	<10	150/250/300	<2
Close range	25–250	10–30	3000	2–4
Medium range	50–250	30–70	3000	3–6
High altitude, long range	>250	>70	>3000	>6

TABLE 2: Comparison among different VTOL concepts and design issues (see text for explanation).

Design driver	A	B	C	D	E	F	G	H
Mechanics simplicity	1	1	1	1	3	4	3	4
Aerodynamics complexity	1	1	1	1	1	3	1	4
Low-speed flight	2	4	3	2	3	4	4	4
Stationary flight	1	4	4	2	4	3	4	4
Control cost	2	1	2	1	1	3	4	3
Payload/volume	2	2	3	1	2	1	4	3
Maneuverability	3	4	3	3	2	1	2	3
High-speed flight	3	2	2	3	4	1	1	3
Miniaturization	2	2	2	4	3	1	4	3
Power cost	3	2	2	3	2	4	2	1
Survivability/Endurance	2	1	1	3	3	3	3	1
<i>Total quality index</i>	<i>22</i>	<i>24</i>	<i>24</i>	<i>24</i>	<i>28</i>	<i>28</i>	<i>32</i>	<i>33</i>

increased payload. Table 2, adapted from [5], gives quality indexes (from 1 = bad to 4 = very good) for some design issues pertaining to different VTOL vehicle concepts, namely, bird-like (A), single-rotor (B), tandem rotors (C), insect-like (D), axial rotor (E), blimp (F), coaxial rotors (G), and quadrotor (H).

The quadrotor has good ranking among VTOL vehicles, but it still keeps some drawbacks. For example, the craft size is rather large; the energy consumption is great; therefore the flight time is short, the control algorithms are very complex, as only four actuators have to control the six degrees of freedom (DOF) of the craft (a quadrotor is a typical example of an underactuated system), and the changing aerodynamic interference patterns between the rotors have to be taken into account [5, Chap. 3] [8, Parts I and II].

Endurance is a fundamental task in UAS design. Generally, it depends on the payload and the aircraft mission (search and rescue, inspection tasks, etc.). Typical UAS endurance are in the range from 30 min (micro UAS) to 24 hours (high altitude, long endurance, or HALE, vehicles). Table 3 classifies UAS in terms of mass, range, flight altitude, and endurance.

Extended flight duration has been mainly addressed by means of optimization of the electric propulsion system [9, 10], use of hybrid-electric systems [11], investigation on innovative power sources [12, 13], use of solar-powered



FIGURE 1: The adopted small quadrotor (Conrad 450 ARF 35 Mhz).

HALE [14], or use of a variable pitch propeller [15]. This work proposes an alternative approach to improve endurance of a nonexpensive (micro-UAV), commercial quadrotor, by applying a balloon to reduce weight and power consumption.

Little research is available in the literature in the field of low-altitude, low-speed UAVs equipped with balloons, except for some work related to subsurface geology detection with a hybrid UAV equipped with an airbag [16]. Our approach aims for a multimission platform with dedicated payload, suitable for a broad range of applications in the fields of Simultaneous Localization and Mapping (SLAM), exploration, search and rescue, remote sensing, and environmental monitoring.

The mini quadrotor used for our investigations is the Conrad Quadcopter 450 ARF [17], shown in Figure 1.

The quadrotor is equipped with a payload composed of a Global Positioning System (GPS) receiver module, an Inertial Measurement Unit (IMU), a sonar altimeter, a small camera module, and main microcontroller hardware (Arduino, Raspberry). Such a payload was selected considering a multimission data gathering platform (attitude measurement, GPS data collection, remote sensing, vision-based navigation, etc. [18–20]). Typical endurance of the chosen quadrotor is less than 2 hours (without payload), whereas in our case (motor propeller plus payload) the flight time has found to be less than 1 hour [18]. Balloons are not easily manoeuvrable, but a hybrid solution (quadcopter merging) could effectively exploit the advantages of a quadrotor vehicle (flexibility, well-designed structure, and security) and the strength points of a balloon (low noise, low energy consumption, and buoyancy providing most of the flight lift), increasing endurance by reducing power consumption [2, 20, 21].

The preliminary design involves determination of weights, gross static lifting capability of the balloon (i.e., the portion of the balloon’s total lift attributable to its buoyancy), lifting gas properties, atmospheric conditions in the flight range, and influence of the balloon size on lift. Although the dynamics and aerodynamics of the HUAS operations must be considered in a complete design of the balloon, we will consider only the balloon static performance, following the approach described in [22].

The paper is structured as follows: in Section 2 we present the HUAS conceptual design. In Section 3 we describe the mathematical models to estimate weight and size of

TABLE 3: UAS classification based on mass, distance covered, altitude, and flight endurance [5].

Category name	Mass [kg]	Range [km]	Flight altitude [m]	Endurance [hours]
Micro	<5	<10	250	1
Mini	5–100	<10	150–300	<2
Close range	25–250	10–30	3000	2–4
Medium range Low endurance	50–250	30–70	3000	3–6
High altitude, long range Medium altitude long endurance (MALE)	>250	>70	>3000	>6
High altitude long endurance (HALE)	>1000	>100	5000–15000	<24
	>10000	>100	>15000	24+

the inflated balloon and to calculate its static performance, presenting preliminary results in Section 4. In Section 5 we present our conclusions and future research challenges.

2. HUAS Conceptual Design

The most interesting improvement created by the HUAS is the aerostatic lift provided by the balloon, which, combined with the fan lift of the propulsion system, allows the vehicle to achieve easy take-off, climbing, hovering, and landing with reduced power consumption. The main issues which inspired us to propose a hybrid solution are as follows:

- (1) Effective control of the HUAS flight path by static lift (provided by the balloon) and rotor power: this allows hovering, flying, climbing, and landing at any height, improving flexibility of the mission and enlarging the range of applicability.
- (2) Low operating speed, low-altitude missions, operation in discontinues trajectories: the HUAS could be effectively used in high-resolution spatiotemporal sampling applications and in monitoring known environments.
- (3) Minimization of complexity of the fuselage structure, drive mechanisms, and engine systems: this implies simpler manufacturing process and shorter production cycle.
- (4) Low noise, low vibration, low turbulence generation: the HUAS does not disturb the environment that is being monitored or measured and reduces sensor noise and potential hardware malfunction due to vibration.
- (5) Reduced cost of energy and power systems and rapid prototyping: the HUAS is an advanced low-cost system easily viable for potential commercial operators.

A stand-alone balloon is not easily manoeuvrable mainly for its big inertia, but in static conditions it is very reliable.

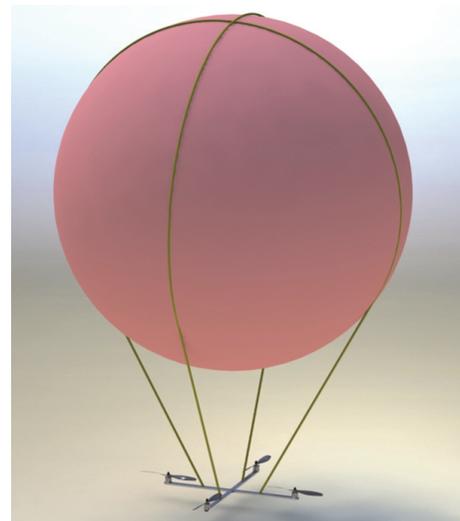


FIGURE 2: HUAS design concept.

It is possible to manage and control the balloon by means of propeller speed changes. Creating hybrid lift (static lift and fan lift) to achieve take-off, hovering, and landing with reduced energy consumption would improve flight duration. The overall system is able to correct any change of flight attitude due to voluntary actions or instability induced by the balloon.

Figures 2 and 3 show the structure of the HUAS, with a basic balloon, support lines, safety lines, and a synthetic model of the quadcopter. The balloon diameter has been chosen to be 1.5 m. Thrust propellers improve manoeuvrability of the HUAS. A simple web frame is chosen to wrap the balloon and link to quadrotor; it is a nonrigid solution, able to keep the two systems linked but independent, each with its own function: the balloon for static gross lift (mainly upward) and the quadrotor for flight control. The buoyancy of the balloon provides most of the flight lift, reducing power consumption and increasing endurance. 3D CAD software

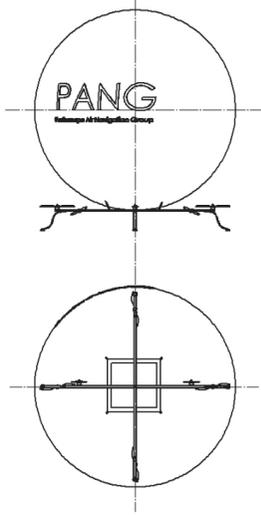


FIGURE 3: HUAS 3D CAD.



FIGURE 4: LiPo battery packs used in research tests.

was used to make a preliminary structure for successive balloon size validation.

The installed battery packs are lightweight LiPo (Lithium Polymer), with capacities of 1800 mAh and 4000 mAh @11.1 V, respectively (Figure 4).

Endurance in minutes, without the balloon, for a normal mission (take-off, hovering, and landing) is shown in Table 4.

3. Estimation of Weights and Balloon Sizing

Each configuration was checked through a static analysis in CAD software, choosing a solution which allowed us to achieve good compromise between weight and lift. The analytical techniques described in [22] were applied, considering three subsystems of the HUAS:

- (i) Inflation gas
- (ii) Balloon structure
- (iii) Quadrotor system

Weight and the static performance of each part were determined separately and successively added. Inertia and

TABLE 4: Endurance for the two LiPo batteries used.

Capacity of LiPo battery	Take-off and landing	Hovering
4000 mAh	<5 min	≈40 min
1800 mAh	<5 min	≈15 min

dynamic properties were not considered, postponing their evaluation to future work.

3.1. Take-Off Weight Estimation. For a correct evaluation of the gross static lift provided by the balloon, it is necessary to estimate the HUAS take-off gross weight (W_{TO}), which depends on all components of the HUAS (the quadcopter structure, payload, battery pack, support lines, safety lines, tether lines, and the balloon itself).

The net force acting on the balloon tether line equals the gross static lift of the balloon less the constant tare of the bag. Equation (1) gives the gross static lift L of the balloon [22]:

$$L = (\rho_{\text{air}} - \rho_g) V_g, \quad (1)$$

where V_g is the volume occupied by the gas when the balloon is fully inflated. In ideal conditions, the gross lift is equal to W_{TO} , given by

$$W_{TO} = W_{\text{eq}} + W_B + W_{\text{batt}} + W_p, \quad (2)$$

where weights are referred to empty quadcopter W_{eq} , balloon (lines, gas, and bag included) W_B , battery pack W_{batt} , and payload (sensors onboard) W_p . Obviously, W_B depends on the balloon size and is the unknown parameter, whereas the other contributions to W_{TO} are constant and known. Using a weight coefficient $k_x = W_x/W_{TO}$ (with x representing any of the subscripts in the right side of (2)), (2) is rewritten as

$$k_{\text{eq}} + \frac{W_B}{W_{TO}} + k_{\text{batt}} + k_p = 1 \quad (3)$$

and derive W_{TO} as follows:

$$W_{TO} = -\frac{W_B}{(k_{\text{eq}} + k_{\text{batt}} + k_p)}. \quad (4)$$

The UAS system considered in this work has a total weight ($W_{\text{eq}} + W_{\text{batt}} + W_p$) of about 3 kg: this is, according to Archimedes' law, the buoyancy requirement.

3.2. Balloon Static Performance and Sizing. A classical airship has various configurations: flabby balloon with ballonet (air chamber), partially inflated and fully inflated (Figure 5). It can also be composed of two parts: a hull and a tail fin assembly. In this work, a fully inflated configuration (logging balloon, without tail fin) was chosen.

For safe flight conditions, the gas volume in fully inflated configuration must always be less than or equal to the total volume. Correct sizing must also take into account temperature and pressure influences on the lifting gas, which

TABLE 5: Lifting gas comparison.

Lifting gas	Density at sea level and 0°C	Lifting force of 1 m ³ of gas
Helium (He)	0.178 kg/m ³	11.8 N
Hydrogen (H)	0.090 kg/m ³	10.9 N
Air	1.292 kg/m ³	##

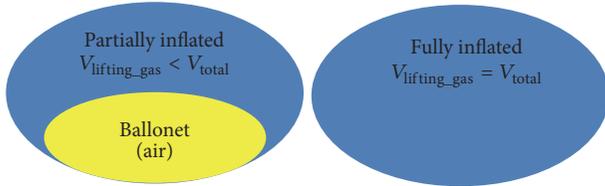


FIGURE 5: Balloon configurations.

obviously must have density lower than air density. Among several available types of lifting gases (hot air, hydrogen, helium, ammonia, etc.), we have chosen helium (He), due to its availability and readiness to use without many control systems (valve, pipes, etc.). Table 5 compares lifting forces of He and hydrogen.

From (1), we consider a lifting force for 1 m³ of gas at sea level and 0°C. Lift decreases proportionally with the altitude (temperature and pressure). The evaluation of the lifting force is only referred to 1 m³ of lifting gas and does not include the bag and lines of the balloon.

The balloon was modelled as a sphere, which contains lifting gas sufficient to equalize, more or less, W_{TO} . The main mission parameters to be defined are altitude (pressure altitude (pressure altitude is the reading of an altimeter when adjusted to the standard MSL (mean sea level) atmospheric pressure of 1013.25 mbar (29.98 inches of mercury); it does not correspond to the actual elevation): PA, temperature T), gas density, and air density. Typical PA values in the range 6–20 m (20–65 ft) and air temperature of 288 K (15°C), reasonably constant in the selected altitude range, were considered. The density-pressure-temperature nomogram (Figure 6) can be used to determine the inflation requirements of the balloon.

The blue dashed line in Figure 6 shows the initial inflation in the selected operational conditions and determines a balloon gas density of 0.168 kg/m³. The design volume V_D of the spherical balloon and the weight of the inflation gas, W_{He} , were computed through use of the perfect gas law, giving

$$V_D = \frac{W_{TO}}{\left(1 - \left(\frac{p_g}{p_{air}}\right) \left(\frac{R_{air}}{R_g}\right)\right) (g p_{air} / R_{air} T^*)}, \quad (5)$$

$$W_{He} = \rho_g V_D, \quad (6)$$

where $T^* = T_g / T_{air}$ and V_D is referred to a fully inflated balloon.

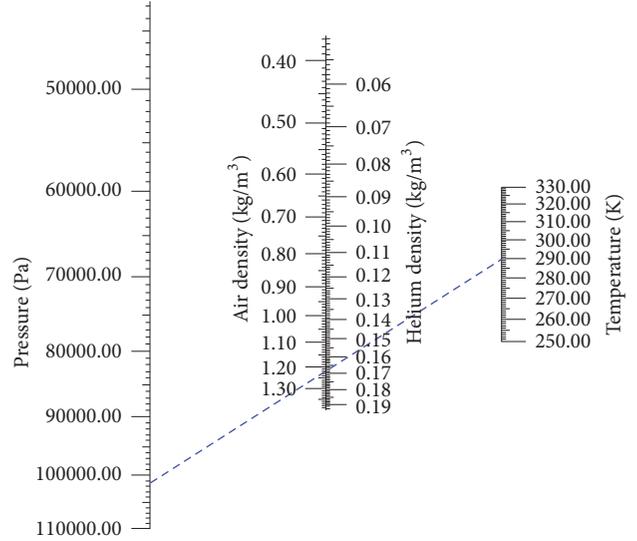
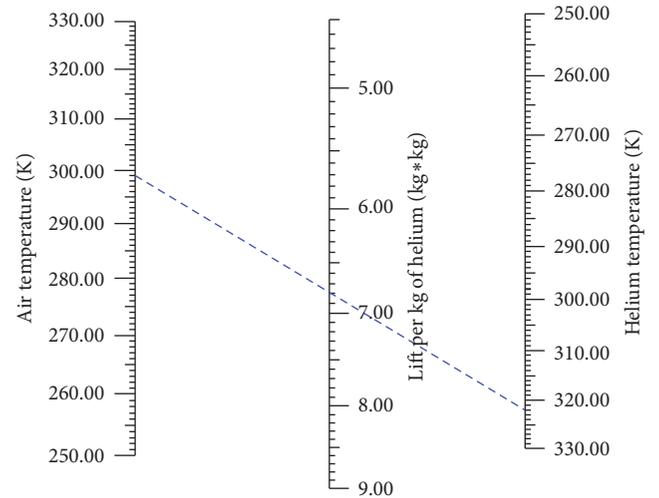


FIGURE 6: Density-pressure-temperature nomogram and initial inflation determination (dashed line).

FIGURE 7: Nomogram for the expected gross static lift ($T_{air} = 299$ K, $T_{He} = 322$ K).

Equation (7) gives the design volume (cubic feet are the preferred unit in the compressed-gas industry) required in standard conditions, V_{STD} :

$$V_{STD} = \frac{\rho_g}{\rho_{STD}} \frac{W_{TO}}{\left(1 - \left(\frac{p_g}{p_{air}}\right) \left(\frac{R_{air}}{R_g}\right)\right) (g p_{air} / R_{air} T^*)}. \quad (7)$$

It is now possible to calculate the expected gross static lift (L_{exp}), given by (1), considering the nomogram for initial inflation, shown in Figure 7. From the blue dashed line L_{exp} is found to be 6.78 kg per kg of He, considering the maximum expected temperatures: $T_{air} = 299$ K ($\approx 27^\circ\text{C}$) and $T_{He} = 322$ K ($\approx 49^\circ\text{C}$).

Generally, L depends on atmospheric pressure and temperature, but we considered a constant pressure value, due

to the low altitude and the small altitude range (0 to 19 m) chosen.

The gross static lift L is given by

$$L = \frac{L_{\text{exp}} \rho_g W_{\text{TO}}}{(1 - (p_g/p_{\text{air}})(R_{\text{air}}/R_g))(\rho_{\text{STD}} g p_{\text{air}}/R_{\text{air}} T^*)}. \quad (8)$$

The design volume V_D is the main parameter determining gross static lift for a specific diameter of the balloon. Various diameters have been considered, in order to attain good efficiency ($L(V_D)/\text{Drag} \approx 1$), at low altitude and low speed.

Recalling that W_{TO} is equal to 3 kg, the balloon weight W_B given by

$$W_B = W_{\text{He}} + W_{\text{bag}} + W_{\text{lines}} \quad (9)$$

has to be added. W_{He} is given by (6), and the remaining weights are a function of V_D and the material used for the bag and the lines. In Section 4 we will show a set of diameters and the corresponding solutions.

The balloon thickness is obviously dependent on the helium pressure, which in turn depends on temperature, volume of the bag (V_D), and number of moles of He. Using the ideal gas law and assuming operational conditions at $T = 15^\circ\text{C}$ (288.15 K) and $p = 1031.25$ hPa, the helium pressure, which must be sustained by the structure (bag), is given by

$$p_{\text{He}} = \frac{nRT}{V_D}. \quad (10)$$

Young-Laplace law [23] can be used to derive the pressure difference $\Delta p = p_{\text{He}} - p_{\text{air}}$, across the interface between air and helium, assuming a spherical balloon of radius r :

$$\Delta p = \frac{2\tau(r)}{r}, \quad (11)$$

where τ is the surface tension (obviously, Δp should be less than the maximum tension of the balloon, to avoid bursting). The material chosen for the balloon was PVC (polyvinyl chloride), whose mechanical characteristics are summarized in Table 6.

It turned out that the tension forces τ are small and the balloon pressure is very close to the atmospheric pressure. Therefore, thickness in the range 0.10–0.30 mm is sufficient to guarantee the necessary lift, keeping the balloon away from bursting conditions.

4. Preliminary Results

The test field and the atmospheric conditions are very important in order to evaluate the lifting properties of the inflation gas used for our HUAS. Experimental results were acquired considering three cases (modelling three different environmental conditions):

- (i) Standard operating conditions (International Standard Atmosphere, ISA, 0 km mean sea level (MSL), temperature 15.0°C (288.15K), density 1.225 kg/m³, and pressure 1013.25 hPa)

TABLE 6: PVC mechanical characteristics.

Property	Units	Method	Value
Specific weight	g/cm ³	ISO 1183	1.42
Yielding tension	MPa	DIN EN ISO 527	58
Elastic modulus	MPa	DIN EN ISO 527	3000
Hardness SHORE D	—	ISO 868	82

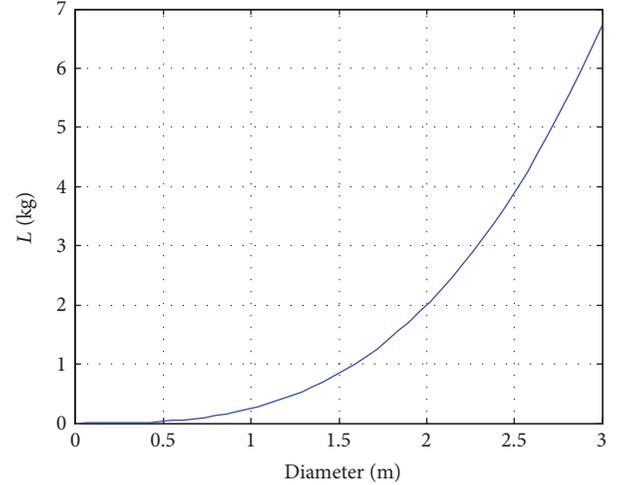


FIGURE 8: Gross static lift versus balloon diameter.

- (ii) Complex scenario (ISA +25°C and ISA -25°C)

In this paper only the temperature variation of air and of the inflation gas is considered. Relative humidity is not taken into account due to its negligible influence on the lifting properties. Air speed has been set equal to 0 (hovering condition). No variation on the payload mass was taken into account, since there is no fuel consumption. Table 7 summarizes the expected operating conditions and the helium expected gross static lift in the standard operating conditions.

Helium density and the expected gross static lift are quite similar (see Figures 6 and 7) and were considered constant. On average, we have a gross static lift (from (8)) of 2.84 kg. In this case, one kg of He is sufficient to lift the UAS and sensors ($W_{\text{eq}} + W_{\text{batt}} + W_p$).

Using a spherical configuration, the total gross static lift becomes a function of the balloon diameter, as shown in Figure 8 (see also (6)).

Figure 9 shows the dependence of the gas weight (to be added to obtain W_{TO}) on the balloon diameter.

In order to analyse the effects of temperature on the static lift, we performed several tests in different conditions, namely, ISA -25°C and ISA +25°C, that is, -10°C and +40°C, as operational boundaries in which the HUAS is expected to work.

By using the nomograms of Figures 6 and 7 and (8), an estimate of the expected helium gross static lift is shown in Table 8 and Figures 10 and 11.

We found little differences with respect to the standard condition, concluding that the balloon can provide the required lift even in complex scenarios, since its inflation

TABLE 7: Lifting properties of He in standard (ISA) operating conditions.

Pressure altitude [m]	Air or helium temperature [°C]	Helium expected gross static lift [kg per kg of He]	Helium density [kg/m ³]	Air density [kg/m ³]
7	17.1045	6.78	0.1681	1.2241
10	17.0850	6.78	0.1681	1.2238
13	17.0655	6.78	0.1681	1.2234
16	17.0460	6.78	0.1682	1.2231
19	17.0265	6.78	0.1682	1.2227

TABLE 8: Results in ISA -25°C and ISA +25°C operating conditions.

Pressure altitude [m]	Air or helium temperature [°C]	Helium expected gross static lift [kg per kg He]	Helium density [kg/m ³]	Air density [kg/m ³]	Gross static lift (see (8)) [kg]
7	-10.609	6.23	0.161	1.17	2.61
10	-10.609	6.23	0.161	1.17	2.61
13	-10.608	6.23	0.161	1.17	2.61
16	-10.605	6.23	0.161	1.17	2.61
19	-10.605	6.23	0.161	1.17	2.61
7	40.253	6.21	0.158	1.15	2.61
10	40.248	6.21	0.158	1.15	2.61
13	40.233	6.21	0.158	1.15	2.61
16	40.230	6.21	0.158	1.15	2.61
19	40.226	6.21	0.158	1.15	2.61

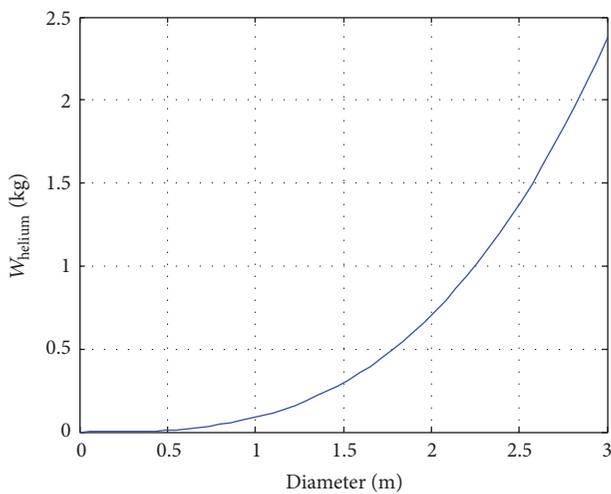


FIGURE 9: Gas weight versus balloon diameter.

requirement is not dependent on temperature in the range from -10 to +40°C.

The total balloon weight W_B is estimated considering a PVC bag with thickness in the range 0.18–0.28 mm. The weight of the link lines is reasonably constant within the considered thickness range.

A diameter of 2.7 meters provides full W_{TO} equivalence (bag and lines included, see Figure 9), but in terms of control

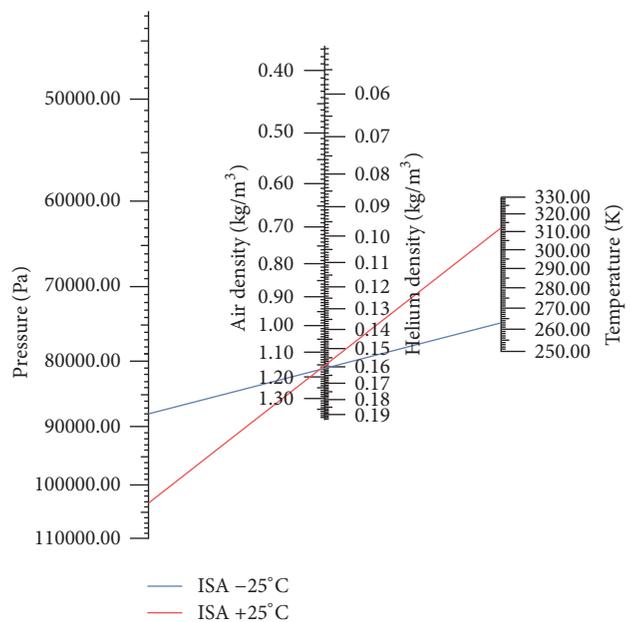


FIGURE 10: Air and helium densities in ISA -25°C and ISA +25°C.

and manoeuvring it is a poor choice, since such a value involves high drag force, depending on body sectional area, acting opposite to the relative motion. This, in turn, involves weak manoeuvrability and control. Since our goal is to reduce

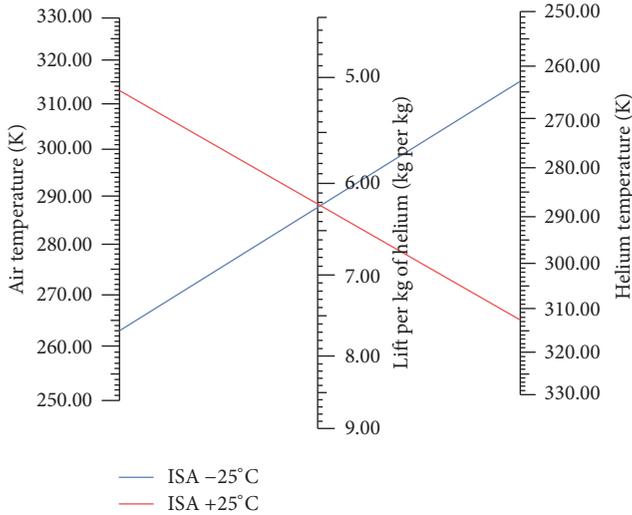


FIGURE 11: Expected static lift per kg of He in ISA -25°C and ISA $+25^{\circ}\text{C}$. The value (almost the same for both cases) has been found to be about 6.22 kg per kg of He.

power consumption of the rotors to increase endurance, it is not necessary to maximize the total gross lift (i.e., $L = W_{\text{TO}}$). In this preliminary design (Figure 12 shows the relationships among design parameters), a solution with gross static lift less than W_{TO} has been chosen, resulting in low diameter and low drag force, with acceptable manoeuvrability and control.

Under these hypotheses, considering the initial inflation parameters, it is possible to estimate gross static lift values for a specific balloon diameter. The generated gross lift is 67% of W_{TO} , for a 2-meter balloon diameter: in this condition the HUAS is able to come back to the ground by its own gravity. During landing and hovering, the lift provided by the balloon and quadcopter can help to get the desired hovering height; so, the total lift is less than the HUAS weight (sensors included). If the propeller speed decreases, the HUAS can land due to its own weight.

Table 9 shows increased performance of the HUAS in the hovering phase, with respect to the standard UAS (without balloon), showing 50% and 40% improvement on the hovering time with the 4000 mAh and 1800 mAh LiPo battery pack used, respectively. No significant change in take-off and landing time was observed.

5. Conclusions and Further Work

This paper reported the main phases of the conceptual design of a low-cost (less than 2 k€), electrically powered hybrid UAS (quadrotor + airship). We investigated the HUAS capability of extended cruise endurance by analysing the design drivers affecting the craft flight time. With respect to conventional electric-motor/battery powered UAS, in which increasing endurance requires heavier batteries, with a consequent weight increment, we propose a solution that increases endurance by using a balloon, resulting in a favourable endurance/weight ratio.

TABLE 9: HUAS versus UAS endurance.

	UAS		HUAS	
	Take-off and landing	Hovering	Take-off and landing	Hovering
4000 mAh	<5 min	≈ 40 min	<5 min	≈ 60 min
1800 mAh	<5 min	≈ 15 min	<5 min	≈ 25 min

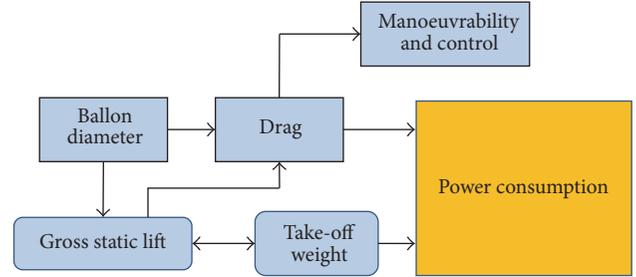


FIGURE 12: Relationships among design parameters.

The installation of a balloon with a diameter of about 2 meters provided significant increase in flight time, as demonstrated in test missions developed in situ (urban traffic monitoring scenario). As shown in Table 9, the HUAS endurance (with 1800 mAh LiPo battery) increased by 50%. The effect of temperature changes on the static lift provided by the balloon (inflated with helium) has been found to be negligible, providing an average value of 2.84 kg lift (27.8 N) per kg of He.

Further investigations will focus on stability and control problems [3, 22] of the HUAS, caused by wind flow (e.g., vertical/horizontal gusts [24, 25]), critical scenarios in take-off and landing (e.g., sloped terrain, obstacles), and pendulum effects during left/right turning, which make trajectory tracking and attitude stabilization challenging tasks [26]. These effects mainly depend on nonlinearities like coupling between the quadrotor and the balloon. Nonlinear techniques [27] are currently under study, together with a structural modification of the craft. The main theoretical aspects to be analysed in further developments will concern optimal control techniques for the following critical issues:

- (1) Stabilization and attitude control during the hovering phase and capability of tracking straight-line trajectories [28, 29]
- (2) Transition between flight modes and operation near ground (especially in take-off and landing missions, where the HUAS is supposed to smoothly reach a desired hovering height or to descend from initial height to 0 m)
- (3) Collision avoidance

As a preliminary step, two additional propellers (Figure 13) will be added to the existing configuration in order to provide thrust for flight direction changing. To remove the nonlinearity introduced by pendulum moments, the tether lines will be reduced, obtaining a final UAS + balloon single structure shown in Figure 14. Preliminary simulations and

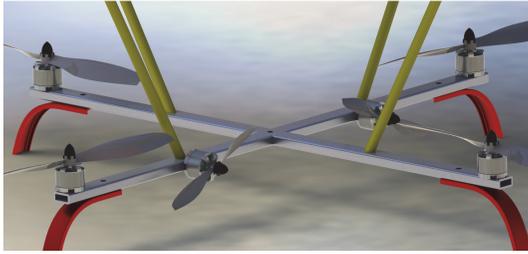


FIGURE 13: Latest HUAS configuration, with directional propellers.



FIGURE 14: Newest configuration to reduce nonlinearities.

dynamic studies confirm improved HUAS stability, allowing us to tune the mechanical design for optimal control sensitivity and disturbance rejection.

Another issue to be explored in further work is the purity of the lifting gas (it is not easy to find 100% pure He), which could impair the gas performance during the mission. The impact of nonpure helium on the lifting properties is currently under examination.

Nomenclature

ρ_g and ρ_{air} :	Gas density and air density (kgm^{-3})
g :	Gravitational acceleration (9.8066 ms^{-2})
L :	Static lift (kg)
p , p_g , and p_{air} :	Pressure, gas pressure, and air pressure (Pa)
R :	Perfect gas constant ($8.314 \text{ Pa m}^3 \text{ mol}^{-1} \text{ K}^{-1}$)
T_g and T_{air} :	Gas temperature and air temperature (K)
V_D :	Balloon design volume (m^3)
V_g :	Gas volume (m^3)
W and W_{TO} :	Weight and take-off gross weight (kg).

Disclosure

This research is a thesis topic in Flight Mechanics/Dynamics and Aeronautical Systems at the Department of Science and Technology of the University of Naples “Parthenope,” with a team of many students studying different HUAS-related issues and developing and proposing possible solutions (under the authors’ supervision).

Conflicts of Interest

The authors Umberto Papa, Giuseppe Del Core, and Salvatore Ponte declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The balloon structure and the integration of electronic sub-systems were performed at the Laboratory of Flight Dynamics of the University of Naples “Parthenope,” Italy.

References

- [1] T. Villa, F. Gonzalez, B. Miljevic, Z. D. Ristovski, and L. Morawska, “An overview of small unmanned aerial vehicles for air quality measurements: Present applications and future perspectives,” *Sensors (Switzerland)*, vol. 16, no. 7, article no. 1072, 2016.
- [2] Y. Inoue, S. Morinaga, and A. Tomita, “A blimp-based remote sensing system for low-altitude monitoring of plant variables: A preliminary experiment for agricultural and ecological applications,” *International Journal of Remote Sensing*, vol. 21, no. 2, pp. 379–385, 2000.
- [3] S. N. Ghazbi, Y. Aghli, M. Alimohammadi, and A. A. Akbari, “Quadrotors unmanned aerial vehicles: a review,” *International Journal on Smart Sensing and Intelligent Systems*, vol. 9, no. 1, pp. 309–333, 2016.
- [4] Z. Sarris, “Survey of UAV applications in civil markets,” in *Proceedings of the 9th IEEE Mediterranean Conference on Control and Automation*, Technical University of Crete, Corfu, Greece, June 2001.
- [5] R. Austin, “Unmanned Aircraft Systems: UAVS Design, Development and Deployment,” *Unmanned Aircraft Systems: UAVS Design, Development and Deployment*, 2010.
- [6] Office of the Secretary of Defense, *Unmanned Aircraft Systems Roadmap: 2005–2030*, Office of the Secretary of Defense, Washington, DC, USA, 2005.
- [7] H. Eisenbeiss, “A mini unmanned aerial vehicle (UAV): system overview and image acquisition,” in *Proceedings of the International Workshop on Processing and Visualization using High-Resolution Imagery*, Institute for Geodesy and Photogrammetry, ETH-Hoenggerberg, Zurich, Switzerland, 2004.
- [8] K. Nonami, F. Kendoul, S. Suzuki, W. Wang, and D. Nakazawa, “Autonomous flying robots: Unmanned aerial vehicles and micro aerial vehicles,” *Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles*, pp. 1–329, 2010.
- [9] X. Liu and R. Zhu, “Endurance optimization for micro air vehicles,” *Acta Aeronautica et Astronautica Sinica*, vol. 29, no. 1, 2008.
- [10] O. Gur and A. Rosen, “Optimizing electric propulsion systems for unmanned aerial vehicles,” *Journal of Aircraft*, vol. 46, no. 4, pp. 1340–1353, 2009.
- [11] J. Y. Hung and L. F. Gonzalez, “On parallel hybrid-electric propulsion system for unmanned aerial vehicles,” *Progress in Aerospace Sciences*, vol. 51, pp. 1–17, 2012.
- [12] T. Chang and H. Yu, “Improving Electric Powered UAVs’ Endurance by Incorporating Battery Dumping Concept,” in *Proceedings of the Asia-Pacific International Symposium on Aerospace Technology, APISAT 2014*, pp. 168–179, chn, September 2014.
- [13] N. Lapeña-Rey, J. A. Blanco, E. Ferreyra, J. L. Lemus, S. Pereira, and E. Serrot, “A fuel cell powered unmanned aerial vehicle for low altitude surveillance missions,” *International Journal of Hydrogen Energy*, vol. 42, no. 10, pp. 6926–6940, 2017.
- [14] X. Z. Gao, Z. X. Hou, Z. Guo, J. X. Liu, and X. Q. Chen, “Energy management strategy for solar-powered high-altitude

- long-endurance aircraft,” *Energy Conversion and Management*, vol. 70, pp. 20–30, 2013.
- [15] G. Ferrarese, F. Giuliotti, and G. Avanzini, “Modeling and simulation of a quad-tilt rotor aircraft,” in *Proceedings of the 2nd IFAC Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pp. 64–70, Compiegne, France, November 2013.
- [16] Y. Wang, S. Le, and Z. Fan, “A new type of rotor + airbag hybrid unmanned aerial vehicle,” *International Journal of Energy Science*, vol. 3, no. 3, pp. 183–187, 2013.
- [17] CONRAD, 2016 <http://www.conrad.com/ce/en/product/208000/QUADROCOPTER-450-ARF-35-MHz>.
- [18] U. Papa and G. Del Core, “Design and assembling of a low-cost mini UAV quadcopter system,” Tech. Rep., Department of Science and Technology, University of Naples “Parthenope”, Naples, Italy, 2014.
- [19] U. Papa and G. Del Core, “Design of sonar sensor model for safe landing of an UAV,” in *Proceedings of the 2nd IEEE International Workshop on Metrology for Aerospace, MetroAeroSpace 2015*, pp. 346–350, Benevento, Italy, June 2015.
- [20] H. Timmis, “Robot integration engineering a GPS module with the arduino,” in *Practical Arduino Engineering*, chapter 5, pp. 97–131, Springer Science+ Business Media, New York, NY, USA, 2011.
- [21] W. W. Carson and P. A. Peters, *Gross Static Lifting Capacity of Logging Balloons*, USDA Forest Service Research Note, 1971.
- [22] S. A. Barton, “Stability analysis of an inflatable vacuum chamber,” *Journal of Applied Mechanics, Transactions ASME*, vol. 75, no. 4, pp. 0410101–0410108, 2008.
- [23] G. K. Batchelor, *An Introduction to Fluid Dynamics*, Cambridge University Press, 1967.
- [24] War Department, “Technical Manual of Airship Aerodynamics,” TM 1-320, US Government Printing Office, Washington, DC, USA, 1941.
- [25] L. T. Redd, R. M. Bennett, and S. R. Bland, “Experimental and analytical determination of stability parameters for a balloon tethered in a wind,” NASA TN D-7222, 1973.
- [26] G. Cai, B. M. Chen, K. Peng, M. Dong, and T. H. Lee, “Modeling and control system design for a UAV helicopter,” in *Proceedings of the 14th Mediterranean Conference on Control and Automation*, pp. 600–606, Ancona, Italy, 2006.
- [27] I. Fantoni and R. Lozano, *Non-linear Control for Underactuated Mechanical Systems*, Springer-Verlag, London, UK, 2002.
- [28] N. G. Shakev, A. V. Topalov, O. Kaynak, and K. B. Shiev, “Comparative results on stabilization of the quad-rotor rotorcraft using bounded feedback controllers,” *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1, pp. 389–408, 2012.
- [29] F. Kendoul, D. Lara, I. Fantoni-Coichot, and R. Lozano, “Real-time nonlinear embedded control for an autonomous quadrotor helicopter,” *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, pp. 1049–1061, 2007.

Research Article

Surrogate Safety Analysis of Pedestrian-Vehicle Conflict at Intersections Using Unmanned Aerial Vehicle Videos

Peng Chen,^{1,2} Weiliang Zeng,³ Guizhen Yu,^{1,2} and Yunpeng Wang^{1,2}

¹School of Transportation Science and Engineering, Beijing Key Laboratory for Cooperative Vehicle Infrastructure System and Safety Control, Beihang University, Beijing 100191, China

²Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, Sipailou No. 2, Nanjing 210096, China

³Institution of Materials and Systems for Sustainability, Nagoya University, Furo-cho, Chikusa, Nagoya 464-8603, Japan

Correspondence should be addressed to Weiliang Zeng; weiliangzeng49@gmail.com

Received 22 March 2017; Accepted 24 April 2017; Published 18 May 2017

Academic Editor: Zhi-Chun Li

Copyright © 2017 Peng Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Conflict analysis using surrogate safety measures (SSMs) has become an efficient approach to investigate safety issues. The state-of-the-art studies largely resort to video images taken from high buildings. However, it suffers from heavy labor work, high cost of maintenance, and even security restrictions. Data collection and processing remains a common challenge to traffic conflict analysis. Unmanned Aerial Systems (UASs) or Unmanned Aerial Vehicles (UAVs), known for easy maneuvering, outstanding flexibility, and low costs, are considered to be a novel aerial sensor. By taking full advantage of the bird's eye view offered by UAV, this study, as a pioneer work, applied UAV videos for surrogate safety analysis of pedestrian-vehicle conflicts at one urban intersection in Beijing, China. Aerial video sequences for a period of one hour were analyzed. The detection and tracking systems for vehicle and pedestrian trajectory data extraction were developed, respectively. Two SSMs, that is, Postencroachment Time (PET) and Relative Time to Collision (RTTC), were employed to represent how spatially and temporally close the pedestrian-vehicle conflict is to a collision. The results of analysis showed a high exposure of pedestrians to traffic conflict both inside and outside the crosswalk and relatively risking behavior of right-turn vehicles around the corner. The findings demonstrate that UAV can support intersection safety analysis in an accurate and cost-effective way.

1. Introduction

Pedestrian safety at intersections remains a critical issue. With the dramatic increasing of urban traffic flow, the major threat to pedestrians comes from frequent interactions with turning vehicles at crosswalk. Though crosswalks are operated to give pedestrians prioritized right of way over vehicles, still around 30% of the total traffic accident fatalities in China are pedestrians according to the accident statistics from the Ministry of Public Security of China [1]. The National Highway Traffic Safety Administration Report [2] indicated that pedestrians account for 15% of the fatalities in traffic accidents in the US. Japan National Policy Agency [3] stated that more than one-third of the total traffic accident fatalities in Japan are pedestrians at signalized and unsignalized crosswalks. Pedestrian safety has become a major concern worldwide.

So far the reactive strategies for the purpose of improving pedestrian safety have been primarily based on identifying sites with high crash rates. It is subject to less crash records or validity losing due to changes of road system and operation. On the other hand, traffic conflict technique (TCT) represents an efficient approach to enable a preventive strategy development. Surrogate safety measures (SSMs) serve as near-crash indicators to measure spatial and temporal proximity of road users. In the context of safety assessment and improvement of urban intersections, the conflict between pedestrians and turning vehicles needs special attention. However, there are still limited applications of SSM on pedestrian-vehicle conflict assessment [4]. One possible reason is that pedestrian exposure to the risk of collision is difficult to measure directly, since this would involve tracking the movements of all people at all time [5].

Data collection and processing remains a common challenge to pedestrian studies.

Field surveys of pedestrian-vehicle conflict are costly to conduct and suffer from inter- and intraobserver variability for the repeatability and consistency of results [6]. Video detection, as alternative data collection procedure to relieve the issues and limitations of manual data collection, has attracted considerable interest. It provides a reliable way to collect road users' positions in time and space, that is, trajectories, that benefit the detailed analysis of pedestrian-vehicle conflict [7]. However, this method is relatively expensive and in practice it is difficult to collect and process video data at a large scale over a long period of time. In order to enable a view of both pedestrians and conflicting vehicles at the monitored intersection, video cameras should be installed high enough, for example, mounted on existing poles located near the intersection. This usually brings heavy labor work and high cost of maintenance and even is not allowed due to security restrictions. Furthermore, the synchronization among multiple cameras for one intersection is complicated and requires much extra effort. Last, to extract pedestrian and vehicle trajectory data from videos with a desirable accuracy and efficiency remains a difficult problem.

Unmanned Aerial Systems (UASs) or Unmanned Aerial Vehicles (UAVs), known for easy maneuvering, outstanding flexibility, and low costs, are considered to be a novel aerial sensor. UAVs can be launched and deployed within minutes and exchange with the control center in real time. While in the last decade UAVs have been frequently employed in the military, civilian applications of UAVs still face several technical and institutional barriers, for example, strict airspace and route restrictions. In recent years, an increasing number of countries such as China and US have begun to consider and evaluate flexible air traffic control rules. For instance, the China Air Traffic Control Center promised to open up the low altitude space (lower than 1000 m) management in the following years. Such emerging trend presents a great opportunity for the transportation departments to fully explore the potential of UAVs in road traffic network surveillance. The equipped sensors on the UAVs such as high-resolution camera, radar, and infrared camera can provide bird's eye view over an intersection or a large area. The entire images and video can be further processed to monitor traffic flow interaction and evaluate traffic state evolution. Thus, UAVs can be an effective aerial traffic information gathering platform. This study will investigate the potential of applying UAV videos for surrogate safety analysis of pedestrian-vehicle conflicts in an accurate and cost-effective way. To the best of the authors' knowledge, it will be the first attempt to employ UAVs for detailed safety assessment at intersections.

The remainder of the paper is organized as follows. A thorough literature review on UAV applications in transportation engineering and operation as well as SSMs for pedestrian-vehicle conflict assessment is presented first. Then the process of data acquisition using UAVs is introduced and the procedures of trajectory extraction are elaborated. Next, postextracted SSMs at one urban intersection in Beijing, China, are investigated in detail by referring to intersection geometry, traffic volume, and signal control strategy. Last,

conclusions are drawn and recommendations are provided for future consideration.

2. Literature Review

2.1. UAV Applications in Transportation Engineering and Operation. UAV, as an aerial traffic information gathering platform, has been becoming more prominent in transportation engineering and operation. For instance, by utilizing aerial images captured from UAVs, the Washington State Department of Transportation evaluated the use of a UAV as an avalanche control tool on mountain slopes above state highways [8]. The Michigan Department of Transportation tested five main UAV platforms with a combination of optical, thermal, and LiDAR sensors to assess critical transportation infrastructure and issues such as bridges, confined spaces, traffic flow, and roadway assets [9]. The Utah Department of Transportation examined the use of high-resolution aerial photography obtained from UAVs to monitor and document State Roadway structures and associated issues [10]. The Florida Department of Transportation investigated the feasibility of using surveillance video from UAVs for traffic control and incident management [11].

Perhaps the most important role that UAVs could fill is providing a rapid response to incidents [12]. Since time of response is vital to victim survivability and eventual health state, a UAV could fly directly to an incident ahead of emergency responders. The timely aerial video images transmitted back to the operators will allow rapid assessment of the situation and proper allocation of emergency response resources. Besides emergency-based applications, UAVs are also valuable for traffic management and monitoring applications. Coifman et al. (2006) demonstrated several applications by using data from a UAV flying in an urban environment, for example, determining level of service, estimating average annual daily travel, documenting intersection operations, and measuring Origin-Destination flows. Cheng et al. [13] presented a method for detecting and counting vehicles from UAV video flow. Hart and Gharaibeh [14] used micro-UAVs as a tool for collecting condition and inventory data for roadside infrastructure assets. Yu and David [15] investigated the feasibility of using high-resolution images acquired by the small UAV in work zone management, traffic congestion, safety, and environmental impact studies.

The spatial perspectives offered by UAVs from the air demonstrate to be more promising than presently available ground-based views for traffic management and monitoring. Useful information can be derived from UAV video for both offline planning and real-time management. To this end, vision-based detection and frame-to-frame matching to track road users are important. However, in practice accurate detection and tracking from the UAV platform is a challenging task due to platform motion, image instability, the relatively small size of the objects, varied appearance, and so forth. Such technical issues may impose limitations to transportation professionals in a variety of intensive research and applies uses. Recently, by using UAV images, Xu et al. [16] developed a new hybrid vehicle detection scheme which integrated the Viola-Jones and linear Support

Vector Machine (SVM) classifier with Histogram of Oriented Gradient (HOG) feature methods; Ma et al. [17] developed a pedestrian detection and tracking system using thermal infrared images recorded from UAVs. The proposed detection and tracking approaches would facilitate a more detailed analysis of road users' behavior and interaction based on accurate trajectory data extracted from UAV video. As an extension of the above work, the aim of this paper is to apply UAV video for surrogate safety analysis of pedestrian-vehicle conflict at intersections.

2.2. SSMs for Pedestrian-Vehicle Conflict Assessment. As an alternative to crash risk estimation based on limited crash data, SSMs serve as near-crash indicators to measure the severity and frequency of traffic conflict events. Numerous SSMs have been suggested for safety evaluation of traffic facilities as shown in Allen et al. [18], Gettman and Head [19], and HSM [20]. In general, a SSM is supposed to satisfy two conditions in order to be useful for safety applications [4]: (1) a measurable or observable noncrash event that is physically related in a predictable and reliable way to crashes and (2) a practical method for converting or calibrating the noncrash event into a corresponding crash frequency and/or severity.

In the case of pedestrian-vehicle conflict at intersections, turning vehicles typically have to filter through conflicting pedestrian flow at crosswalk during permitted signal phase as implemented in China and US. Under the mixed impact of surrounding environment, crosswalk geometry, signal operation, and pedestrians moving in different directions, turning vehicles might take risky behavior by not yielding to pedestrians or passing through small gaps in pedestrian flow, which poses a threat to pedestrian safety. The most commonly used SSMs for pedestrian conflict assessment include but not limited to the following measures:

- (i) Time to Collision (TTC), which is defined as the time that remains until a collision between two road users would have occurred if the collision course and speed difference are maintained [21].
- (ii) Postencroachment Time (PET), which is defined as the time difference between the moment when an offending road user leaves an area of potential collision and the moment of arrival of a conflicted road user possessing the right of way [22].
- (iii) Time to Zebra (TTZ), which is a variation of TTC in order to estimate frequency and severity of critical encounters between crossing pedestrians and vehicles that are approaching the crosswalk [23].
- (iv) Deceleration-to-Safety Time (DST), which is the necessary deceleration to reach a nonnegative PET value if the movement of the conflicting road users remains unchanged [24].
- (v) Gap Time (GT), which is defined as the time lapse between the completion time of encroachment by one road user and the arrival time of the interacting road user if they continue with the same speed and path [25].

In general, Allen et al. [18], Gettman and Head [19], and Gettman et al. [26] found that TTC and PET are ranked as the most accurate measures for the analysis of safety at intersections in light of ease of measurement, consistency over time, and relation to other measures. TTC requires estimating the time remaining to the conflict point at each time instant in the case of pedestrian-vehicle conflict, while to measure PET, only their passing times at conflict point are necessary. Due to its simplicity, PET is also amenable to automated measurement methods using techniques such as video image processing. Another important property of PET is that it is continuous from crash-free operations to crash occurrences with a distinct boundary at zero. The smaller value of PET implies a greater risk of vehicle-pedestrian collisions. In practice, Songchitruksa and Tarko [27] demonstrated the usefulness of the number of short PETs in explaining the variability of crash counts and concluded that the frequency of short PETs is a potential indicator in discriminating varying safety levels across survey sites.

However, PET has inherent drawbacks in its ability to accurately capture conflict severity [6]. For example, the events in which the approaching vehicle decelerated to near-stop to avoid collision with the conflicting pedestrian may have PET values that do not reflect that true severity of the interaction. On the other hand, the main advantage of TTC is its ability to capture the severity of an interaction in an objective and quantitative way. Thus, a combination of these SSMs would be necessary to help identify all the dangerous interactions between vehicles and pedestrians.

3. Methodology

3.1. Detection and Tracking. In order to investigate pedestrian-vehicle conflict, road users should be detected and then tracked frame-to-frame in UAV video. In this study, we extract the trajectories for vehicles and pedestrians, respectively, at intervals of every 0.04 s by using the detection and tracking system developed in our previous studies [16, 17]. A brief introduction is provided below.

An image processing system for automated vehicle trajectory extraction was developed based on UAV videos [16]. As shown in Figure 1, the system mainly includes three modules: (1) video stabilization; (2) vehicle detection; and (3) vehicle tracking. Note that, due to UAV motions, image registration algorithm [29] was first applied to stabilize UAV videos. The object detection framework of faster R-CNN [30] was applied for vehicle detection. Then the algorithm of kernelized correlation filters (KCF) [31] was applied for vehicle tracking. The trajectory of one vehicle will be derived after the tracking was finished.

Similar to the work of Beymer et al. [32], several entry and exit regions were set as shown in Figure 2. Commonly, the entry and exit regions are set at the upstream and downstream of a signalized intersection. Once a vehicle entered the entry region it will be detected and tracked; the tracking will be finished until the vehicle entered the exit region. Left-turn and right-turn vehicles can be distinguished by the information of entry and exit included in the trajectories. For example, trajectories of left-turn vehicles from the

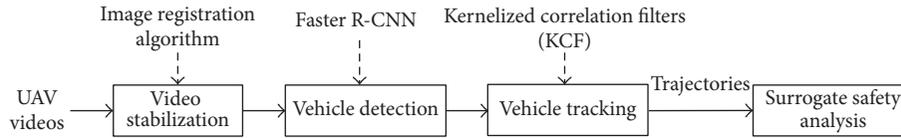


FIGURE 1: The workflow of vehicle detection and tracking.

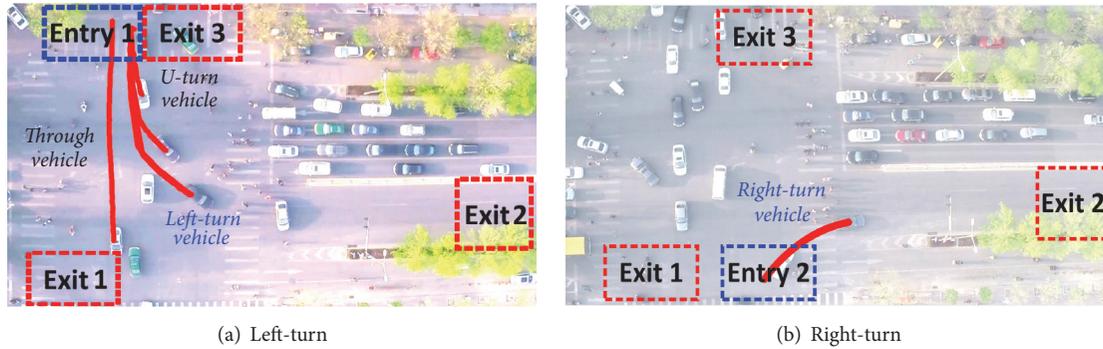


FIGURE 2: The configuration of entry and exit regions for turning vehicles.

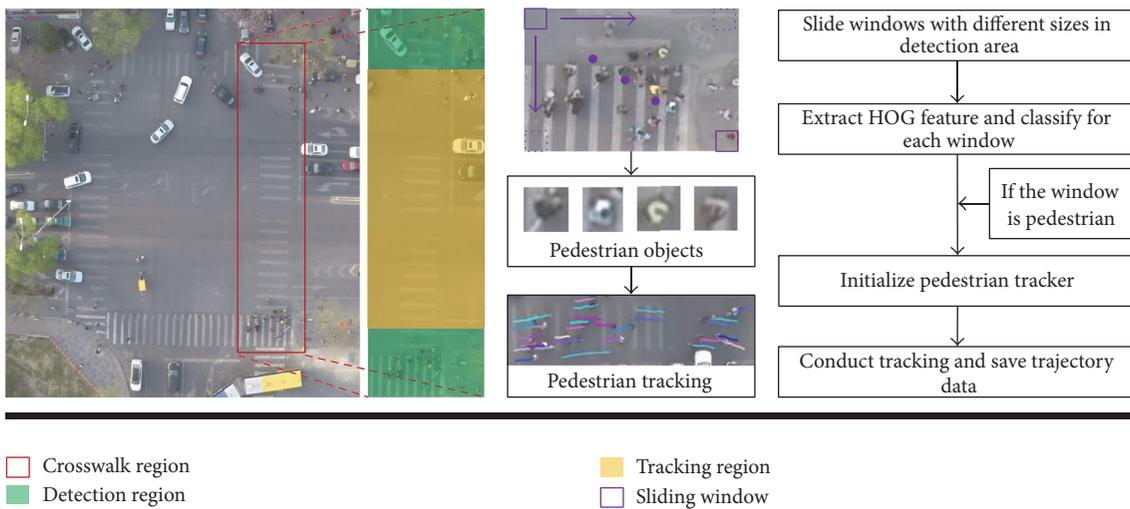


FIGURE 3: The workflow of pedestrian detection and tracking.

southbound approach can be obtained between Entry 1 and Exit 2 by distinguishing from through and U-turn vehicles as shown in Figure 2(a); similarly, trajectories of right-turn vehicles from the northbound approach can be obtained between Entry 2 and Exit 2 as in Figure 2(b).

Pedestrian detection and tracking from the UAV platform is a challenging task due to the small size of the objects and the high-density crowd. A semiautomatic pedestrian detection and tracking system was developed for pedestrian trajectory data extraction from UAV aerial images [17]. The developed system consists of two components, that is, detector and tracker. The detector is responsible for automatic aerial pedestrian detection. The tracker is used for pedestrian tracking and extracting trajectory coordinate data. Figure 3 illustrates the overall workflow of the system. First, pedestrians are detected by detector in detection area (green region as

shown in Figure 3), which is manually set up according to the crosswalk location. A general and machine learning-based method is employed for constructing pedestrian detector. It includes two stages, that is, pedestrian feature descriptor extraction and pedestrian classification. HOG feature, as a kind of local gradient feature, has been proved to perform well in pedestrian detection problems [17, 33]. Hence, our pedestrian detector employs HOG feature as pedestrian descriptor. By utilizing multisize sliding window method, detector scans the detection area. For each window, a linear SVM classifier [34] is used to classify window as pedestrian or nonpedestrian. When pedestrians are detected, their coordinates will be fed as inputs for initialization of trackers. Then the trackers apply the pyramidal Lucas–Kanade method [35] to compute the local sparse optical flow, together with a secondary detection in the search region for correcting the



FIGURE 4: (a) Tracking trajectory visualization and (b) tracking point visualization.

drift when tracking pedestrians. A tracker is responsible for only one pedestrian object. After tracking, the pedestrian trajectory data can be saved for further analysis. Figure 4 shows the pedestrian tracking results and trajectory visualization.

Note that the tracked positions or trajectories might contain measurement errors. Kalman filtering (KF) was used to correct the errors and smooth the raw trajectory data. The KF computes the best estimate of the state vector (i.e., position coordinates) by minimizing the squared error according to the estimation of the past state and the present state. The image coordinates were converted to geographic coordinates by projective transformation.

The available observations are trajectory profiles based on time series. From these data, all relevant quantities of vehicles and pedestrians, such as positions, velocities, and acceleration, can be derived either directly or by applying finite differences. The ordinary differential equations for speed and acceleration can be solved as follows:

$$\begin{aligned} \vec{v}_\alpha(t) &= [\vec{v}_\alpha^x(t), \vec{v}_\alpha^y(t)] = \left[\frac{x_\alpha(t+1) - x_\alpha(t-1)}{2\Delta t}, \right. \\ &\quad \left. \frac{y_\alpha(t+1) - y_\alpha(t-1)}{2\Delta t} \right], \\ \vec{f}_\alpha(t) &= [\vec{f}_\alpha^x(t), \vec{f}_\alpha^y(t)] \\ &= \left[\frac{(x_\alpha(t+1) - x_\alpha(t)) - (x_\alpha(t) - x_\alpha(t-1))}{(\Delta t)^2}, \right. \\ &\quad \left. \frac{(y_\alpha(t+1) - y_\alpha(t)) - (y_\alpha(t) - y_\alpha(t-1))}{(\Delta t)^2} \right], \end{aligned} \quad (1)$$

where $\vec{v}_\alpha(t)$ is the speed vector for vehicle/pedestrian α at time t , $\vec{f}_\alpha(t)$ is the acceleration vector for α at time t , x_α and y_α are the positions in the x and y directions, respectively, and Δt is the time interval for trajectory extraction, that is, 0.04 s.

3.2. SSM Measurement. Traditional traffic conflict technique usually use PET and TTC to represent the probability of collision or how close the conflict is to a collision [7, 36].

In the context of vehicle conflict assessment, PET is defined as the time difference between the moment when the first vehicle passed the conflict area and the moment of arrival of the second vehicle subsequently at the same area. In the context of vehicle-pedestrian conflict assessment, PET can be similarly defined as the time difference between the departure of the encroaching pedestrian from the potential collision point and the arrival of the conflicting vehicle at the point, or vice versa. However, as PET only considers the last moment of the interaction, it has limitations in indicating pedestrian safety during the course of vehicle-pedestrian interaction.

Alternatively, TTC has been commonly implemented as a measure of conflict severity for the whole interaction process. It was originally defined as the time that remains for the paired vehicles before they collide, if both continue at their present speeds along their respective trajectories. TTC can be easily detected in the rear-end conflict situation because the trajectories of the paired vehicles are assumed to be overlapped. However, it cannot be detected (or does not exist) in most of the interactions if the trajectories of the paired users intersect, for example, the pedestrian-vehicle conflict and the conflict between left-turn and opposing through vehicles. In the rear-end conflict, the following vehicle will definitely collide with the leader vehicle if the speed of the follower is higher. However, for the pedestrian-vehicle conflict, the cases that the pedestrian and the vehicle occupy the trajectory intersection point at the same moment are rare. To overcome this problem, we use the Relative Time to Collision (RTTC) as the indicator to measure the conflict severity. As shown in Figure 5, RTTC is defined as the time difference between the first road user arriving at the potential conflicting location and the second road user arriving at this location if they keep their current speeds. It should be noted that the TTC can be detected only when the RTTC equals zero. The RTTC can be formulated as follows:

$$\text{RTTC}(t) = \frac{\vec{P}_c(t) - \vec{P}_v(t)}{\vec{v}_v(t)} - \frac{\vec{P}_c(t) - \vec{P}_p(t)}{\vec{v}_p(t)}, \quad (2)$$

where $\vec{P}_c(t)$ is the potential conflicting point at the moment t , which is the intersecting point determined by the moving

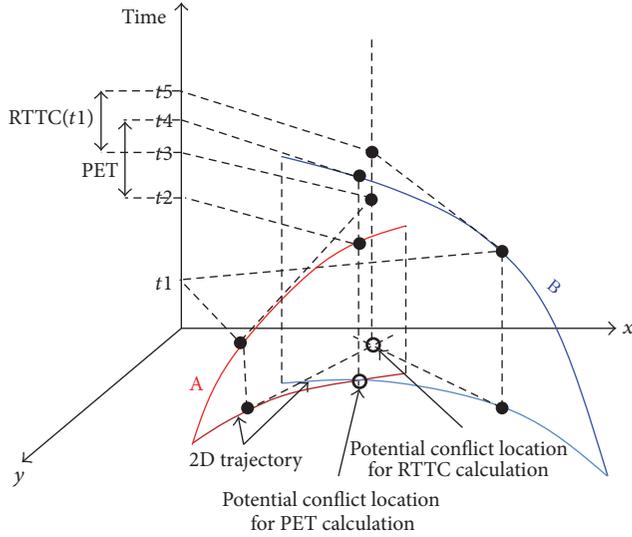


FIGURE 5: RTTC and PET of pedestrian-vehicle conflict identified on the time-space diagram (modified from [28]).

directions of two road users [37], as shown in Figure 5; $\vec{P}_v(t)$ is the location of the vehicle at the moment t ; $\vec{P}_p(t)$ is the location of the pedestrian at the moment t ; $\vec{v}_v(t)$ is the speed of the vehicle at the moment t ; $\vec{v}_p(t)$ is the speed of the pedestrian at the moment t .

In data processing, both RTTC and PET are obtained as a function of paired vehicle-pedestrian speeds and spacing. A time-space diagram identifying RTTC and PET for a pedestrian-vehicle conflict event is illustrated in Figure 5. The trajectories of the crossing pedestrian and the turning vehicle are represented by curve A and curve B, respectively. In such cases, RTTC at instant t_1 can be obtained as

$$\text{RTTC}(t_1) = t_5 - t_3. \quad (3)$$

The PET for such a conflict event can be obtained as

$$\text{PET} = t_4 - t_2 \quad (4)$$

Note that RTTC is instant varying and continually calculated between conflicting vehicles and pedestrians. Thus, a set of RTTC values will be obtained for each conflict. The minimum RTTC (RTTC_{\min}) can be extracted from this set to indicate the maximum severity of this interaction. In this study, only traffic events with associated minimum RTTC of less than 3 s are considered for safety assessment. This value was selected by considering the close proximity of road users in space and time based on the work of Sayed and Zein [38]. On the other hand, by referring to the empirical work by Ni et al. [36], only the conflicting events with PET values of less than 3 s are included for safety assessment.

4. Experiment Results and Analysis

4.1. Study Site. The selected study site is the intersection of Huayuan Road and Beitucheng Road in the Haidian District

of Beijing, China. The intersection is located on a key route to the downtown area, characterized by higher vehicle volume and medium-to-high pedestrian demand during peak hours. For signal control, this intersection is fixed-time controlled with a cycle length of approximately 120 s. The yellow time durations are 3 s and the all-red durations are 1 s at all the approaches. The three-phase control plan is presented in Figure 6. Note that, for the northbound and southbound approaches, both left-turn and right-turn phases are permitted, indicating that potential conflicts exist when turning vehicles filter through conflicting pedestrian streams at crosswalk. The length and width of the crosswalks in the north-south directions are 35 m and 5.5 m, respectively. The green phase for pedestrians at the two crosswalks is 45 s. The light volume of bicycles at the site ensures that bicycle interference with pedestrian flow can be roughly neglected.

Experiments were conducted using aerial videos captured by an optical camera (GoPro Hero Black Edition 3) with a 1920×1080 resolution mounted on a quadrotor UAV (model: Phantom 2). Figure 7 shows the basic components of the UAV platform. A 3-axis gimbal is mounted on the UAV to stabilize the videos and eliminate video jitters caused by UAV, thus greatly reducing the impact from external factors, for example, wind. Besides, an on-screen display, an image transmission module and a video monitor are installed in the system for data transmission and airborne flying status monitoring and control. The flexibility of this device enables wide coverage of the scene from a top-down view. The flight altitude was set approximately 100 m above the ground in this experiment. Aerial video was recorded by the UAV from 5 PM to 6 PM on 17 of April 2015.

4.2. Data Extraction. In total, the dataset consists of the trajectories of 1494 pedestrians and 282 right-turn vehicles. The visualization of the extracted trajectories is presented in Figure 8. The significant variation of pedestrians and turning vehicles' trajectories can be identified. It is worth mentioning that not a small number of pedestrians walked outside the crosswalk during pedestrian green phase and some rushed into crosswalks without necessarily heeding approaching turning vehicles during pedestrian flashing green phase, which may increase the occurrence probability of severe conflicts. Besides, by taking the westbound approach as an example, Figure 9 compares the distributions of the extracted right-turn vehicle trajectories at three cross-sections. Though right-turning vehicles entered the intersection centered around the middle point of the through-right lane, as shown in Figure 9(a), the exiting positions of right-turning vehicles are widely distributed through the cross-sections 2 and 3, as in Figures 9(b) and 9(c). It may lead to widely distributed conflict points on crosswalk. In reality, pedestrians and vehicles behave by anticipating the behavior of each other to avoid collision. On signalized crosswalk, pedestrian trajectories are supposed to be under the interaction between pedestrian flow, conflicting vehicles, pedestrian signal control, intersection geometry, and so forth [37, 39]. Turning vehicle trajectories are also sensitive to intersection corner radius, turning angle (i.e., the angle between entering and exit approaches), and vehicle speed.

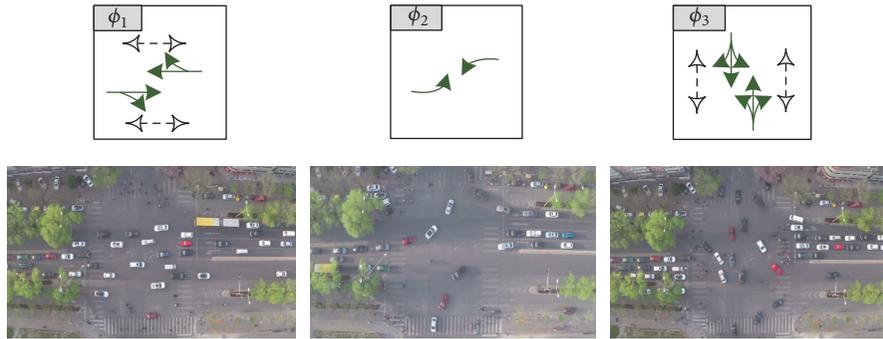


FIGURE 6: Signal phasing at study site.



FIGURE 7: Quadrotor UAV for video collection.

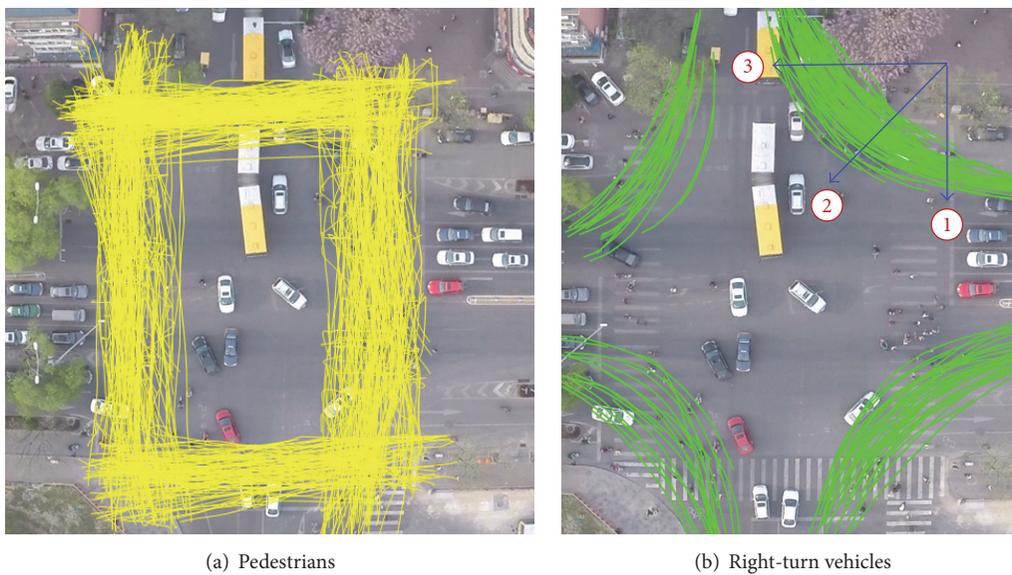


FIGURE 8: Visualization of extracted trajectories.

The accurately extracted trajectories from UAV video offer a good basis for intersection safety assessment.

4.3. *SSM Analysis.* Based on extracted trajectory data, traffic conflicts between pedestrians and right-turn vehicles were identified and SSMs, that is, PET and RTTC, were calculated

accordingly. In terms of SSMs, the conflict analysis aims to identify conflict frequency, severity and location (conflict points).

The spatial distribution of small PETs (which are less than 3 s in this study) is shown in Figure 10. A smaller PET value indicates a higher probability of collision occurring at

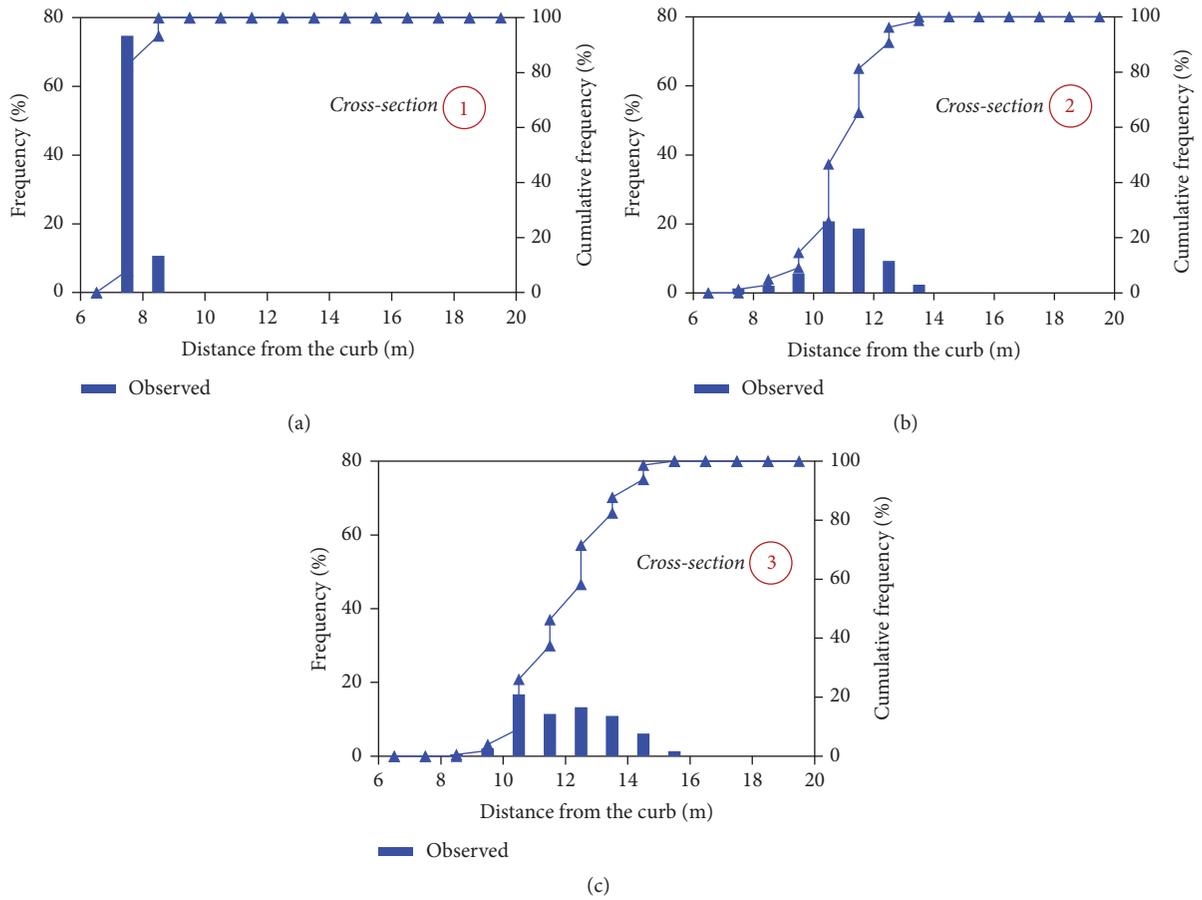


FIGURE 9: Comparison of extracted right-turn vehicle trajectories at cross-sections.



FIGURE 10: Spatial distribution of the number of small PETs.

the end of the vehicle-pedestrian interaction. It shows that most of the small PETs as well as the related conflict points are widely distributed on the area in front of downstream exiting approaches. The details of the statistics are shown in

Figure 11. It is found that 57% of the small PETs occur inside the crosswalk, while 43% outside the crosswalk. Even though the pedestrians are supposed to walk inside the crosswalk, not a small number of them walk outside the crosswalk

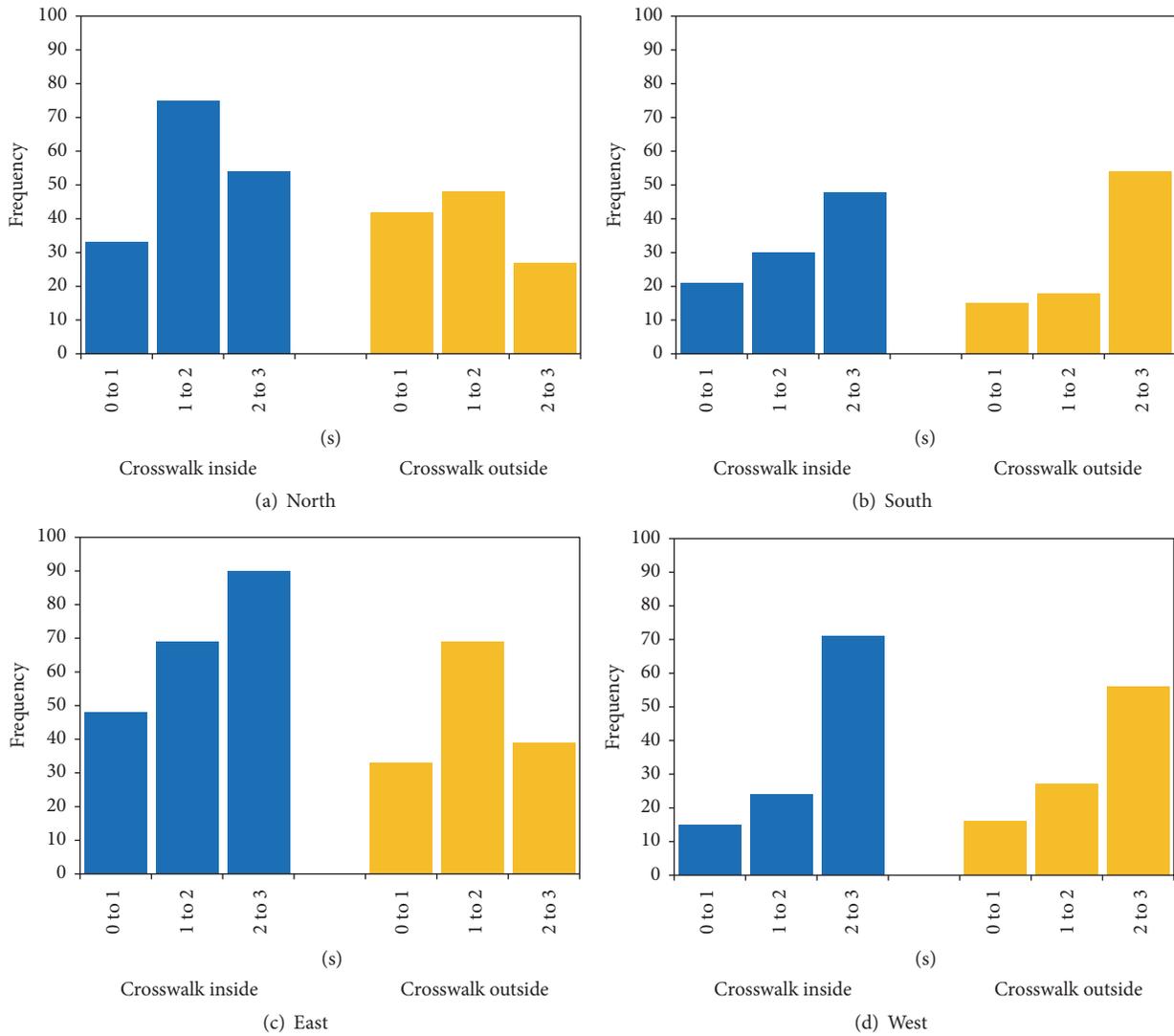


FIGURE 11: The number of small PETs inside and outside of the crosswalk.

under complex interactions with opposing pedestrians and turning vehicles. If we look into the percentage of severe conflicts ($PET < 1s$), it is found that 20% of the PETs are less than 1s for the pedestrians walking inside the crosswalk, while there are 25% of the PETs that are less than 1s for the pedestrians walking outside the crosswalk. It indicates that walking outside the crosswalk is more dangerous. A possible reason is that drivers might not recognize and yield to the pedestrians if the pedestrians do not walk inside the crosswalk. Figure 11 shows that almost 64% of the small PETs and 70% of the severe conflicts ($PET < 1s$) occur at the northern and eastern crosswalks due to the large pedestrian flow.

Different from PET, RTTC reflects the potential conflict severity during the course of vehicle-pedestrian interaction. Figure 12 presents the spatial distribution of pedestrian-vehicle conflicts with the minimum RTTC of less than 3 s by heat mapping. It clearly shows that most of the severe conflicts occur outside the crosswalk. Two reasons may lead to this

phenomenon. First, not a small number of pedestrians walk outside the crosswalk, which may cause the severe conflict with right-turn vehicles. Second, the potential conflict point for calculating RTTC is determined by the intersection of the current speed vectors of the paired users, which may distribute sparsely due to the flexible change of pedestrian movement.

In general, there are two types of vehicle-pedestrian conflict, that is, vehicle yielding to pedestrian, also known as pedestrian passing first (PPF), and pedestrian yielding to vehicle, also known as vehicle passing first (VPF). The PPF and VPF cases are compared because these two types of conflicts can result in different safety performance. Figure 13 provides the details of statistics of the critical PPF and VPF ($RTTC < 1s$) inside and outside the crosswalk. It is found that VPF occupies 55%, while PPF occupies 45% for the total critical RTTCs. It indicates that VPF is more dangerous than PPF. This is because both the vehicle and the pedestrian do not yield to each other and the vehicle tries to accelerate to



FIGURE 12: Spatial distribution of $RTTC_{min}$.

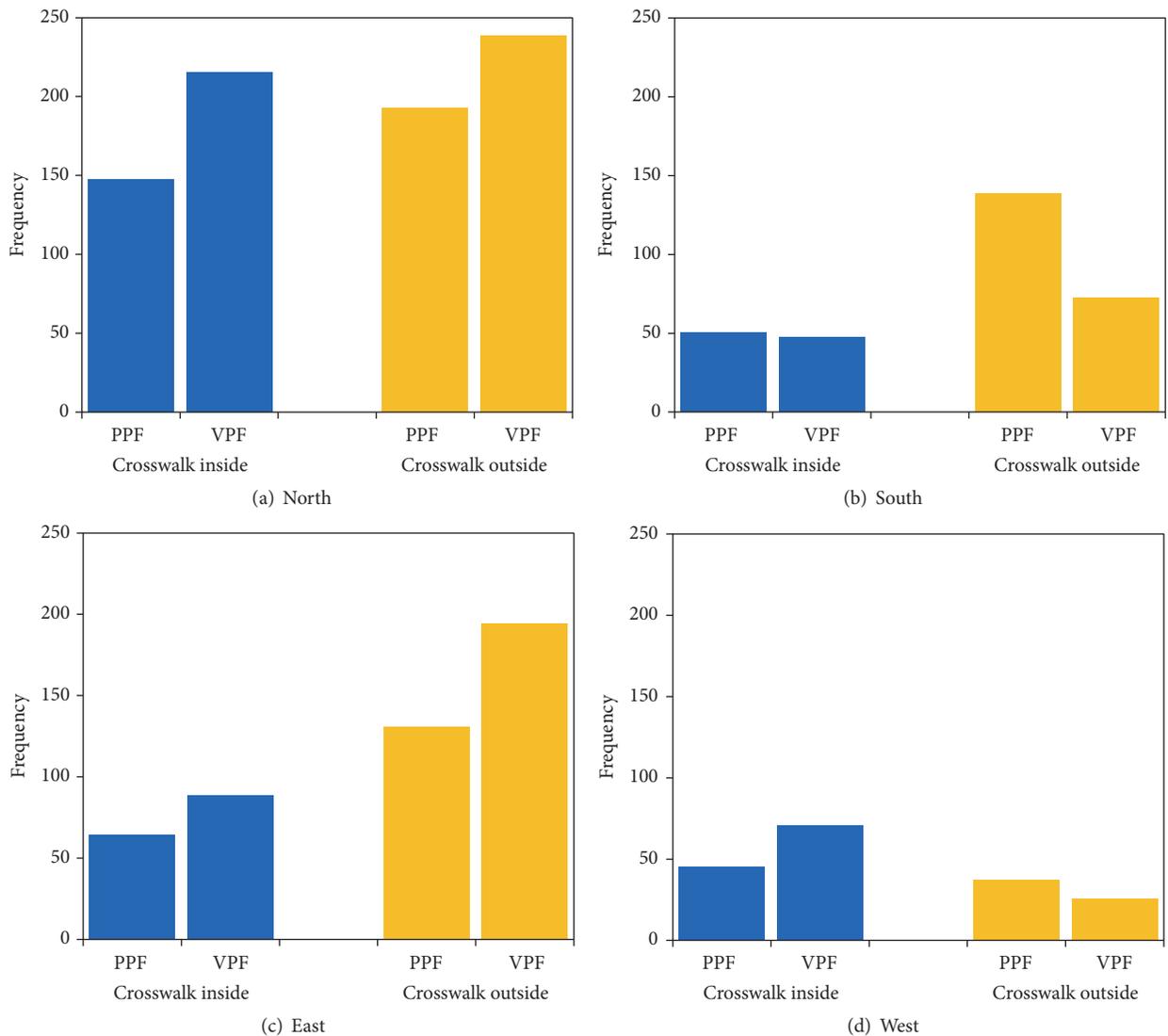


FIGURE 13: The number of critical RTTC inside and outside of the crosswalk.

pass the conflict area first. For the PPF cases, most of the vehicles decelerate to yield to the pedestrians, even though the vehicle may not totally stop and reaccelerate quickly once the pedestrians pass the conflict area. It was also found that a large percent of VPF occur outside the crosswalk at the northbound and eastbound due to the high traffic flow of right-turn vehicles and pedestrians at these two crosswalks. The statistics of PET in Figure 11 also show the potential risk at these crosswalks. Because drivers may not notice the pedestrian outside the crosswalk, the frequent critical confliction may lead to accidents. Thus, we propose the recommendation that a special pedestrian phase should be set to separate the right-turn vehicles from the southbound and eastbound. In practice, varying pedestrian and vehicular behavior may lead to misunderstanding of others' decisions, which may result in safety problems. Considering widely distributed conflict points, both pedestrians and turning vehicles must pay attention to a broader area where conflicts may occur.

5. Conclusions and Future Work

Despite the prominent advantage of UAVs for emergency and traffic monitoring, there has been no research yet to employ UAVs for detailed safety assessment at intersections. In practice, accurate detection and tracking from UAVs is a challenging task due to platform motion, image instability, the relatively small size of the objects and varied appearance, and so forth. This study, as a pioneer work, investigated the feasibility of applying UAV video for surrogate safety analysis of pedestrian-vehicle conflict at intersections. By taking full advantage of the bird's eye view offered by UAV, the image processing systems for automated vehicle trajectory extraction and semiautomatic pedestrian trajectory extraction were developed, respectively. Based on the trajectory data collected from one urban intersection in Beijing, China, two SSMs, that is, PET and RTTC, were employed to represent the frequency, severity, and location of pedestrian-vehicle conflict. The results of analysis showed a high exposure of pedestrians to traffic conflict both inside and outside the crosswalk and relatively risking behavior of right-turn vehicles around the corner. The findings demonstrate that UAV can support intersection safety analysis in an accurate and cost-effective way.

Still, there are some limitations of this study. Firstly, due to the limitations of top-down views, the characteristics of pedestrian heterogeneity, for example, gender and age, cannot be identified in the video and thus are not discussed in this study. In pedestrian safety analysis, a recognized key issue [40] is that pedestrians of different gender and age behave differently on crosswalk. This aspect will be investigated in future studies with aerial videos captured at a lower flight altitude and with better visibility. Secondly, it is necessary to further study the interactions between pedestrian groups and vehicle (platoons), since the safety assessment for group interactions and pairwise intersections can be different [36]. Thirdly, the threshold values of PET and RTTC in this study were derived from previous studies, which may be influenced by the configuration of intersections and crosswalks and

need further investigation. The pedestrian-vehicle conflict at other intersections with different intersection geometries, traffic volumes, and signal control strategies is supposed to be analyzed and compared in terms of SSMs with more UAV videos. Last but not least, it is essential in practice for transportation authorities to equip and manage UAVs in order to quickly detect and evaluate intersection safety, especially for the sites with frequent and severe traffic conflicts. Thus, it will be interesting to analyze the cost-benefit of establishing an integrated UAV network with fixed detectors by accounting for various UAV speed, admissible airspace, and operational budget constraints.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors appreciate the National Natural Science Foundation of China (no. 51308475 and no. U1564212) for funding support of this research. Special thanks go to Mr. Yongzheng Xu and Mr. Yalong Ma for their efforts in the UAV data analysis.

References

- [1] Ministry of Public Security of China, Road Traffic Accident Statistics, 2011.
- [2] Traffic Safety Facts, *2015 Motor Vehicle Crashes: Overview*, National Highway Traffic Safety Administration, U.S. Department of Transportation, Washington, DC, USA, 2016, DOT HS 812 318.
- [3] National Policy Agency, Fatal traffic accident in 2012, Japan, <http://www.npa.go.jp/toukei/index.htm>.
- [4] A. Tarko, G. Davis, N. Saunier, T. Sayed, and S. Washington, "Surrogate Measures of Safety White Paper," ANB20(3) *Subcommittee on Surrogate Measures of Safety and ANB20 Committee on Safety Data Evaluation and Analysis*, Transportation Research Board, 2009.
- [5] R. Greene-Roesel, M. C. Diogenes, and D. Ragland, "Estimating Pedestrian Accident Exposure," California PATH Research Report, UCB-ITS-PRR-2010-32, 2010.
- [6] K. Ismail, T. Sayed, N. Saunier, and C. Lim, "Automated analysis of pedestrian-vehicle conflicts using video data," *Transportation Research Record*, no. 2140, pp. 44–54, 2009.
- [7] T. Sayed, M. H. Zaki, and J. Autey, "Automated safety diagnosis of vehicle-bicycle interactions using computer vision analysis," *Safety Science*, vol. 59, pp. 163–172, 2013.
- [8] E. D. McCormack, "The Use of Small Unmanned Aircraft by the Washington State Department of Transportation," Report No. WA-RD 703.1, Washington State Department of Transportation, Washington, DC, USA, 2008.
- [9] C. Brooks, R. Dobson, D. Banach et al., "Evaluating the Use of Unmanned Aerial Vehicles for Transportation Purposes," Report No. RC-1616, Michigan Department of Transportation, Michigan, Mich, USA, 2015.
- [10] S. L. Barfuss, A. Jensen, and S. Clemens, "Evaluation and Development of Unmanned Aircraft (UAV) for UDOT Needs,"

- Tech. Rep., Utah Department of Transportation, Utah, Utah, USA, 2012.
- [11] H. A. Latchman, T. Wong, J. Shea et al., *Airborne Traffic Surveillance Systems: Proof of Concept Study for the Florida Department of Transportation*, University of Florida, Florida Department of Transportation, Gainesville, FL, USA, 2005.
 - [12] B. Coifman, M. McCord, R. G. Mishalani, and K. Redmill, "Surface transportation surveillance from unmanned aerial vehicles," in *Proceedings of the 83rd Annual Meeting of the Transportation Research Board*, Washington, DC, USA, 2004.
 - [13] P. Cheng, G. Zhou, and Z. Zheng, "Detecting and counting vehicles from small low-cost UAV images," in *Proceedings of the American Society for Photogrammetry and Remote Sensing Annual Conference, ASPRS '09*, vol. 1, pp. 138–144, Maryland, Md, USA, March 2009.
 - [14] W. S. Hart and N. G. Gharaibeh, "Use of micro unmanned aerial vehicles in roadside condition surveys," in *Proceedings of the 1st Congress of the Transportation and Development Institute of ASCE*, pp. 80–92, Chicago, Ill, USA, March 2011.
 - [15] G. Yu and R. M. David, "Development of UAV-Based Remote Sensing Capabilities for Highway Applications," *Research and Innovative Technology Administration (RITA)*, 2012.
 - [16] Y. Xu, G. Yu, Y. Wang, X. Wu, and Y. Ma, "A hybrid vehicle detection method based on viola-jones and HOG + SVM from UAV images," *Sensors*, vol. 16, no. 8, p. 1325, 2016.
 - [17] Y. Ma, X. Wu, G. Yu, Y. Xu, and Y. Wang, "Pedestrian detection and tracking from low-resolution unmanned aerial vehicle thermal imagery," *Sensors*, vol. 16, no. 4, p. 446, 2016.
 - [18] B. L. Allen, B. T. Shin, and P. J. Cooper, "Analysis of traffic conflicts and collisions," *Transportation Research Record*, vol. 667, pp. 67–74, 1978.
 - [19] D. Gettman and L. Head, "Surrogate Safety Measures from Traffic Simulation Models," Final Report, Publication No. FHWA-RD-03-050, Federal Highway Administration, Washington, DC, USA, 2003.
 - [20] Highway Safety Manual, American Association of State Highway and Transportation Officials, USA, 2010.
 - [21] J. C. Hayward, "Near-miss determination through use of a scale of danger," *Highway Research Record*, vol. 384, pp. 24–34, 1968.
 - [22] J. Cooper, "Experience with traffic conflicts in canada with emphasis on post encroachment time techniques," in *International Study of Traffic Conflict Techniques*, vol. 75, pp. 75–96, 1984.
 - [23] A. Varhelyi, "Dynamic Speed Adaption Based on Information Technology—A Theoretical Background," in *Bulletin 142*, Department of Traffic Planning and Engineering, Lund University, Lund, Sweden, 1996.
 - [24] C. Hupfer, "Deceleration to safety time (DST)—a useful figure to evaluate traffic safety," in *Proceeding of the ICTCT Conference Proceedings of Seminar 3*, Department of Traffic Planning and Engineering, Lund University, Lund, Sweden, 1997.
 - [25] J. Archer, *Methods for the Assessment and Prediction of Traffic Safety at Urban Intersections and Their Application in Micro-Simulation Modeling*, Academic Thesis, Royal Institute of Technology, Stockholm, Sweden, 2004.
 - [26] D. Gettman, L. Pu, T. Sayed, and S. Shelby, "Surrogate Safety Assessment Model and Validation," FHWA Report, Publication No.: FHWA-HRT-08-051, Federal Highway Administration, Washington, DC, USA, 2008.
 - [27] P. Songchitruksa and A. P. Tarko, "The extreme value theory approach to safety estimation," *Accident Analysis and Prevention*, vol. 38, no. 4, pp. 811–822, 2006.
 - [28] Y. Zhang, D. Yao, T. Z. Qiu, L. Peng, and Y. Zhang, "Pedestrian safety analysis in mixed traffic conditions using video data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1832–1844, 2012.
 - [29] A. C. Shastry and R. A. Schowengerdt, "Airborne video registration and traffic-flow parameter estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 4, pp. 391–405, 2005.
 - [30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 1, 2016.
 - [31] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.
 - [32] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, "A real-time computer vision system for measuring traffic parameters," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pp. 495–501, San Juan, Puerto Rico, June 1997.
 - [33] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 886–893, San Diego, Calif, USA, June 2005.
 - [34] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
 - [35] J. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the Algorithm," 2016, http://robots.stanford.edu/cs223b04/algo_tracking.pdf.
 - [36] Y. Ni, M. Wang, J. Sun, and K. Li, "Evaluation of pedestrian safety at intersections: a theoretical framework based on pedestrian-vehicle interaction patterns," *Accident Analysis and Prevention*, vol. 96, pp. 118–129, 2016.
 - [37] W. Zeng, P. Chen, H. Nakamura, and M. Iryo-Asano, "Application of social force model to pedestrian behavior analysis at signalized crosswalk," *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 143–159, 2014.
 - [38] T. Sayed and S. Zein, "Traffic conflict standards for intersections," *Transportation Planning and Technology*, vol. 22, no. 4, pp. 309–323, 1999.
 - [39] W. Zeng, H. Nakamura, and P. Chen, "A modified social force model for pedestrian behavior simulation at signalized crosswalks," *Procedia—Social and Behavioral Sciences*, vol. 138, pp. 521–530, 2014.
 - [40] W. Daamen and S. Hoogendoorn, "Calibration of pedestrian simulation model for emergency doors by pedestrian type," *Transportation Research Record*, vol. 2316, pp. 69–75, 2012.