aerospace

MDPI

Article

Filtering and Estimation of State and Wind Disturbances Aiming Airship Control and Guidance

Apolo Silva Marton [1], José Raul Azinheira [2], André Ricardo Fioravanti [1], Ely Carneiro De Paiva [1,*], José Reginaldo H Carvalho [3] and Ramiro Romankevicius Costa [4]

[1] School of Mechanical Engineering, University of Campinas, R. Mendeleyev, 200, Campinas 13083-180, SP, Brazil
[2] Department of Mechanical Engineering, Instituto Superior Técnico, Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
[3] Institute of Computing of the Federal University of Amazonas, Av. General Rodrigo Octavio Jordão Ramos, 1200-Coroado I, Manaus 69067-005, AM, Brazil
[4] Federal Institute of Education, Science and Technology of São Paulo, Campus Campinas, R. Heitor Lacerda Guedes, 1000, Campinas 13059-581, SP, Brazil
* Correspondence: elypaiva@unicamp.br

Abstract: Good state and wind estimation is a requirement for the development of guidance and control techniques for airships. However, usually this information is not directly available from the airship sensors. The typical solution applies filtering, estimation and sensor fusion methods. This paper presents a comparative study, evaluating three solutions for the state estimation of NOAMAY airship. We also present alternative versions for the crucial estimation of the wind velocity, combining Kalman filters with a data-driven Neural Network. Finally, we present special solutions to usual problems encountered in filtering implementation as the mitigation of delays caused by second-order filters. The sensors set considered is composed of a global positioning system, an inertial measurement unit and a one-dimensional Pitot probe. Comparative simulation results are presented with the use of a realistic nonlinear model of the airship.

Keywords: wind estimation; guidance; kalman filter; neural network

Citation: Marton, A.S.; Azinheira, J.R.; Fioravanti, A.R.; De Paiva, E.C.; Carvalho, J.R.H.; Costa, R.R. Filtering and Estimation of State and Wind Disturbances Aiming Airship Control and Guidance. *Aerospace* **2022**, *9*, 470. https://doi.org/10.3390/aerospace9090470

Academic Editors: Alberto Rolando and Carlo E.D. Riboldi

Received: 27 May 2022
Accepted: 15 August 2022
Published: 23 August 2022

1. Introduction

Airships with autonomous operation capability are in the focus of worldwide investigation. Airships are particularly suited to scenarios that demand long endurance, high payload and low operational risk, such as surveillance and environmental monitoring [1].

Airship control is a big challenge for the control systems designers as they are under-actuated vehicles with strong nonlinear undamped dynamics [2]. Further, many nonlinear control approaches like sliding mode, backstepping and incremental dynamic inversion require the knowledge of vehicle velocities and accelerations, that may not be available from sensors [3]. Thus, an accurate estimation of the state and its derivatives is fundamental to the design of efficient control and guidance techniques.

Another crucial point is the wind estimation, which is particularly important for lighter-than-air vehicles [4,5]. The work of [6] showed that a good wind knowledge can result in more accurate geolocation of a ground target. Additionally, the crab-angle between ground track and heading, obtained from estimating the wind conditions, can improve the control performance of an airship in trajectory following. Finally, when the objective is ground-hover at constant altitude, the blimp must align itself against the wind and use its forward propulsion to balance the aerodynamic drag [7]. Therefore, by knowing the wind speed and direction control performance can be improved.

For both state and wind estimation, the solution relies on filtering, estimation and fusion of the sensory data. Common approaches are the Extended Kalman Filter (EKF) and

the Unscented Kalman Filter (UKF). As an example, in [8], it is proposed an approach for estimating the angle of attack and sideslip angle by the kinematic equations of motion of an aerobatic UAV. Meanwhile, with the same kinematic equations, in [6] an extended Kalman flter (EKF) is proposed for estimating the wind heading and velocity using an aircraft with a single GPS and Pitot tube. In [9], a wind velocity observer also based on the kinematics is proposed for small UAVs with experimental results. Similarly, in [10] is also proposed an EKF for wind velocity estimation, however applied to a stationary stratospheric airship in simulation environment. Then in [11] are presented four model-based solutions considering an aircraft with four diferent possible configurations of sensors, as is shown in Figure 1.

Figure 1. NOAMAY Airship.

This work is placed in the context of project InSAC https://www2.eesc.usp.br/insac accessed on 25 May 2022. The project aims to develop an autonomous airship for performing environmental monitoring tasks in remote Amazon rain forest areas. Such tasks include aerodynamic flights and ground-hover (i.e., keep a stationary position with respect to a ground target).

Firstly, the paper presents a comparative study evaluating different different solutions for the state/derivatives estimation of NOAMAY airship, which is instrumented with a pitot tube and an Inertial Measurement Unit (IMU) including: accelerometer, gyroscopes, GPS, barometer and thermometer.

In a second moment, for the wind estimation, this work presents an alternative version of a model based on wind velocity estimator using the EKF combined with a data-driven approach of estimation using a Neural Network (NN). The main tool to validate the proposed estimation approaches is a dynamical realistic nonlinear model of an airship in Simulink/MATLAB. This tool is a result of the research group efforts since the seminal project AURORA, which was improved during the projects DIVA and DRONI [12].

Finally, we present possible solutions to mitigate common problems arriving in filtering implementation like the control input redundancy and the delays caused by second-order filters [3]. For the input redundancy, related to an overactuated control system, we propose filtering the less important commanded input, an approach that is known as Washout Filter (WOF) [13]. Another critical point is the drawback that appears in the second-order filters used to estimate state derivatives. The natural dynamics delay produced by these filters may harm the control system feedback. One possible solution to mitigate this delay is the so called Input Scaling Gain (ISG), that was first proposed by our research group in 2015 as a scalar factor gain [14]. In this paper, we generalize the input scale gain to the MIMO case. To illustrate these two design solutions we present briefly the concept of incremental controllers, that are used to show the benefits of these filtering techiques, although not limited to these kind of controllers.

2. Airship Model

The main tool to validate the proposed control approach is a dynamical realistic nonlinear model of a robotic airship, which can be expressed in a state-space form as following:

$$\dot{\boldsymbol{\xi}} = g(\boldsymbol{\xi}, \mathbf{x}, \mathbf{d}), \tag{1a}$$

$$\dot{\mathbf{x}} = f(\boldsymbol{\xi}, \mathbf{x}, \mathbf{d}, \mathbf{u}), \tag{1b}$$

where:

- the kinematic states $\boldsymbol{\xi} = [\mathbf{P^T} \;\; \mathbf{\Phi^T}]^\mathbf{T}$ include the cartesian position $\mathbf{P} = [P_N \;\; P_E \;\; P_D]^T$ and angular position $\mathbf{\Phi} = [\phi \;\; \theta \;\; \psi]^T$ in the North-East-Down (NED) frame (SI units);
- the dynamic states $\mathbf{x} = [\boldsymbol{V}^\mathbf{T} \;\; \mathbf{\Omega}^\mathbf{T}]^\mathbf{T}$ include the linear speed $\boldsymbol{V} = [u \;\; v \;\; w]^T$ and angular speed $\mathbf{\Omega} = [p \;\; q \;\; r]^T$ in the body frame (SI units);
- the input vector $\boldsymbol{u} = [\delta_e \;\; \delta_a \;\; \delta_r \;\; \delta_0 \;\; \delta_q \;\; \mu_0]^T$ includes: δ_e, δ_a and δ_r which are elevator, aileron and rudder deflections; δ_0 as the normalized thrusters voltage; δ_q as the differential voltage between the front-back thrusters; μ_0 as the common vectoring angle of the thrusters;
- and, finally, the disturbance vector $\mathbf{d}$ that includes wind velocities and gust variables (SI units), as is shown in Figure 2.

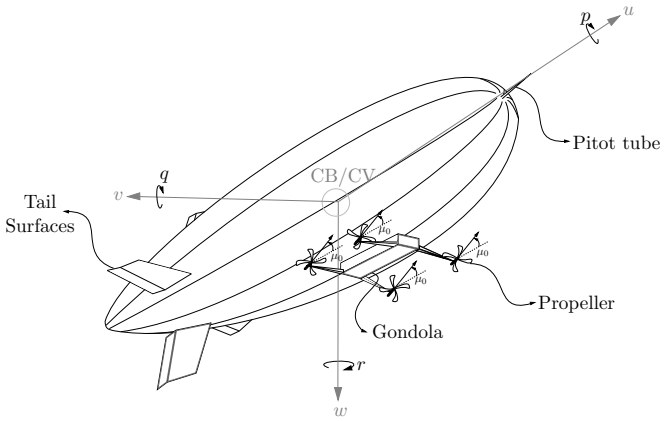

Figure 2. NOAMAY airship body diagram.

The dynamics are based on Newton-Euler equations considering the body frame centered in the Center of Buoyancy (CB), which is approximately equivalent to the Center of Volume (CV) [12]. Further, some nonlinear effects are included in the simulation, such as: atmospheric pressure variation, temperature variation, lifting gas pressure variation, Coriolis effect and actuator nonlinearities [7].

Finally, the linear and angular positions are updated through kinematic equations as follows:

$$g(\boldsymbol{\xi}, \mathbf{x}) = \begin{bmatrix} \mathbf{S_\Phi^T} & 0_{3\times3} \\ 0_{3\times3} & \mathbf{R_\Phi^T} \end{bmatrix} \begin{bmatrix} \mathbf{V_g} \\ \mathbf{\Omega} \end{bmatrix}, \tag{2}$$

where $0_{3\times3}$ is a null matrix of 3th order, $\mathbf{S_\Phi} \in \mathbf{R^{3\times3}}$ is the rotational matrix from airship body to NED frame, and $\mathbf{R_\Phi} \in \mathbf{R^{3\times3}}$ is the angular transformation matrix, both given by [7]:

$$\mathbf{S_\Phi} = \begin{bmatrix} c_\psi c_\theta & s_\psi c_\theta & -s_\theta \\ c_\psi s_\theta s_\phi - s_\psi c_\phi & s_\psi s_\theta s_\phi + c_\psi c_\phi & c_\theta s_\phi \\ c_\psi s_\theta c_\phi + s_\psi s_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi & c_\theta c_\phi \end{bmatrix},$$

$$\mathbf{R}_{\mathbf{\Phi}} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix},$$

where $c_x = \cos(x)$, $s_x = \sin(x)$ and $t_x = \tan(x)$.

3. Sensors Modeling

In this section, a summary of the sensors modeling is presented, including manufacturer specifications, such as bias and sample frequency. The NOAMAY robotic airship is instrumented with a set of sensors including a Pitot Tube, responsible for measuring the airspeed (V_{pitot}) and a multi-sensor Xsens Mti-G 700 which provides the following information [15]:

- absolute position $\mathbf{P_{NED}}$ [m] measured by GPS;
- absolute orientation in the world $\mathbf{\Phi}$ [rad], as result of a sensor fusion between Gyroscope and Magnetometer;
- inertial linear velocity ($\mathbf{V_g}$) in three-axis [m/s], estimated by the GPS;
- inertial angular velocity ($\mathbf{\Omega}$) in three-axis [rad/s], estimated by the IMU;
- inertial acceleration in three-axis ($\boldsymbol{a} = [a_x \; a_y \; a_z]^T$) [m/s^2], measured by the Accelerometer;
- atmospheric pressure P_h [hPa], by the barometer;
- and temperature T_h [K], measured by the thermometer.

The sample frequency of the sensors are shown in the Table 1. Further, a first order model for the Gauss Markov error of each sensor is given as [16]:

$$y = \hat{y} + b + w_y \tag{3a}$$

$$\dot{b} = -\frac{1}{\tau} b + w_b \tag{3b}$$

where b is the sensor bias with Gaussian noise w_b, w_y is a Gaussian noise for the measure as shown in Table 2, $\hat{y}$ is the true simulated value and y is the corresponding sensor output. For the sensors presenting bias, the parameter σ_{RMS} (The noise power is given by $(\sigma_{RMS})^2/$(sample frequency)) is specified in Table 3. In the case of Accelerometer, Barometer and Thermometer there are some special parameters shown below.

Table 1. Sample frequency specification.

Sensor	Frequency [Hz]	Sampled data
IMU	100	$\mathbf{\Phi}$, $\mathbf{\Omega}$, $\boldsymbol{a}$
GPS	4.00	$\mathbf{V_g}$, $\mathbf{P_{NED}}$
Barometer	50.0	P_h
Thermometer	1.00	T_h

Table 2. Sensor noise specification (w_y).

Data	σ_{RMS}	Data	σ_{RMS}
ϕ, θ [rad]	0.00520	$\boldsymbol{a}$ [m/s^2]	0.0040
ψ [rad]	0.10000	P_N, P_E [m]	2.5000
P_D [m]	5.00000	V_{pitot} [m/s]	0.0020
$\mathbf{\Omega}$ [rad/s]	0.00035	$\mathbf{V_g}$ [m/s]	0.4000
P_h [hPa]	0.01000	T_h [K]	1.0000

Table 3. Sensor bias specification (w_b).

Sensor	σ_{RMS}
Gyroscope [$^\circ/s$]	$1.3{\cdot}10^{-4}$
Accelerometer [m/s^2]	$2.3{\cdot}10^{-5}$
Barometer [hPa]	$1.4{\cdot}10^{-5}$

For the case of pose measurement a coordinate transformation is applied to the data (with translational and rotational components). The GPS output is modeled as following:

$$\mathbf{P_{gps_k}} = \mathbf{S_{xsens}}(\mathbf{P_{NED_k}} + \mathbf{P_{xsens}}) + \boldsymbol{\varphi}_{\mathbf{pos_k}}, \tag{4}$$

where: $\mathbf{S_{xsens}}$ is the rotational matrix from the CB angular pose to the Xsens angular pose; and $\mathbf{P_{xsens}}$ is the Xsens position in the body frame; $\boldsymbol{\varphi}_{pos_k}$ is a position Gaussian noise vector at instant k; and $\mathbf{P_{gps_k}}$ is the sampled position given by the GPS at instant k.

The orientation output is modeled as following:

$$\boldsymbol{\Phi}_{imu_k} = \boldsymbol{\Phi}_{xsens} + \boldsymbol{\Phi}_k + \boldsymbol{\varphi}_{ori_k}, \tag{5}$$

where: $\boldsymbol{\Phi}_k$ is the orientation given in simulation at instant k; $\boldsymbol{\Phi}_{xsens}$ is the Xsens angular pose in the body frame; $\boldsymbol{\varphi}_{ori_k}$ is an orientation Gaussian noise vector at instant k; and $\boldsymbol{\Phi}_{imu_k}$ is the sampled angular position given by IMU at instant k.

For the velocity measurement, we apply the coordinate transformation to the data (with rotational components).

The resulting linear velocity output is as follows:

$$\mathbf{V_{gps_k}} = \mathbf{S_{xsens}}(\mathbf{V_{g_k}} + \boldsymbol{\Omega}_\mathbf{k} \times \tilde{\mathbf{O}}_\mathbf{xsens}) + \boldsymbol{\varphi}_{\mathbf{vel_k}}, \tag{6}$$

where: $\boldsymbol{\Omega}_k$ is the angular velocity given in simulation at instant k; $\tilde{\mathbf{O}}_\mathbf{xsens}$ is the position vector from the CB to the Xsens position; $\mathbf{S_{xsens}}$ is the rotational matrix from the CB angular pose to the Xsens angular pose; and $\mathbf{P_{xsens}}$ is the Xsens position in the body frame; $\boldsymbol{\varphi}_{vel_k}$ is a position Gaussian noise vector at instant k; and $\mathbf{P_{gps_k}}$ is the sampled position given by the GPS at instant k.

The angular velocity output is modeled as following:

$$\boldsymbol{\Omega}_{imu_k} = \mathbf{S_{xsens}}\boldsymbol{\Omega}_\mathbf{k} + \boldsymbol{\varphi}_{\mathbf{rate_k}}, \tag{7}$$

where: $\boldsymbol{\varphi}_{rate_k}$ is an angular velocity Gaussian noise vector at instant k; and $\boldsymbol{\Omega}_{imu_k}$ is the sampled angular velocity given by the IMU at instant k.

The accelerometer signal includes the additional components of centripetal acceleration (a_c) and gravitational acceleration (a_g), as:

$$\boldsymbol{a} = \dot{\mathbf{V}}_g + \dot{\boldsymbol{\Omega}} \times \tilde{\mathbf{O}}_\mathbf{xsens} - \mathbf{S}_\boldsymbol{\Phi}\mathbf{a_g} + \boldsymbol{a_c}, \tag{8}$$

where $\tilde{\mathbf{O}}_\mathbf{xsens}$ is the vector from the CB to the sensor location, S_Φ is the rotation matrix from the body frame to the North-East-Down frame and $\boldsymbol{a}_c = \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \tilde{\mathbf{O}}_\mathbf{xsens})$.

In the case of the static pressure at a given altitude, we use the International Standard Atmosphere (ISA):

$$P_h = P_0\left(\frac{T_h}{T_h - 0.0065P_D}\right)^{5.257}, \tag{9}$$

where P_h is the static pressure, T_h is temperature in Kelvin, P_D is the negative of the altitude (in meters) in the inertial frame and P_0 is the pressure at zero altitude (at the sea level $P_0 = 101.325$ hPa).

The temperature is given by the classical model, as linearly varying with the altitude:

$$T_h = T_0 + 0.0065P_D, \tag{10}$$

where T_0 is the temperature at zero altitude, recalling that P_D is negative in our model.

4. Filtering and Estimation

We present here three possible solutions for the estimation of the state variables of an airship, with particular analysis for the NOAMAY airship. The first one is the classical second-order filter, and the other two are Kalman Filters (EKF and UKF). The performances of the three approaches are then analyzed and compared. The algorithms developed in C/C++ using the Robotic Operating System (ROS) [17], are available for consulting.

4.1. Low-Pass Filter

The first estimator is the classical second-order Low-pass filter (LPF), defined as:

$$\frac{\chi(s)}{Y(s)} = \frac{\omega_f^2}{s^2 + 2\omega_f \zeta s + \omega_f^2}, \tag{11}$$

where $\chi(s)$ is the estimated state, $Y(s)$ is the sensor measure, ω_f is the cut-off frequency and ζ is the damping coefficient. For each estimated state and corresponding sensor, the filter parameters used here are given in the Table 4. These values were chosen during simulation by manual tuning.

Table 4. Configuration for Low-pass filter.

State	Sensor	ω_f [rad/s]	ζ
$\mathbf{P_{NED}}$, $\mathbf{V_g}$	GPS	3.33	0.96
$\mathbf{\Phi}$, $\mathbf{\Omega}$	IMU	10	0.96

4.2. Extended Kalman Filter

The EKF algorithm is probably the most popular estimator used in the literature. In this work we use the EKF ROS package (Available https://wiki.ros.org/robot_localization accessed on 25 May 2022) known as `robot_localization`) [18]. This package is related to the configuration of a generic 6DOF mobile robot with 6-DOF, yielding the filtered estimation of positions, velocities and accelerations, both linear and angular.

We want to estimate here the complete airship state vector ($\mathbf{P_{NED}}$, $\mathbf{\Phi}$, $\mathbf{V_g}$, $\mathbf{\Omega}$, $\boldsymbol{a}$). Thus, the measurement update stage uses the sensors output, as described in Table 5, where 1 indicates the presence of the sensor in the corresponding measured variable while 0 is the opposite.

Table 5. Configuration for measurement update stage.

Data\Sensor	GPS	IMU	Barometer
P_{NED}	1	0	0
Φ	0	1	0
u, v	1	0	0
w	1	0	1
Ω	0	1	0
$\boldsymbol{a}$	0	1	0

One important feature of `robot_localization` package is the sensor-based frame reference in the East-North-Upper (ENU) frame, while we use the Norht-East-Down (NED) frame, requiring a conversion. Further, regarding the acceleration signal, we implement a cascaded approach in order to remove the centripetal acceleration component, as:

$$\hat{\dot{\mathbf{V}}}_{\mathbf{g}} = \boldsymbol{a} - \hat{\mathbf{\Omega}}_{\mathbf{0}} \times (\hat{\mathbf{\Omega}}_{\mathbf{0}} \times \tilde{\mathbf{O}}_{\mathbf{xsens}}), \tag{12}$$

The weights of the diagonal covariance matrix (15×15) of the process are shown in Table 6. The gains were chosen empirically, considering that the vehicle has slow dynamics and that the airship can be assumed to be a rigid body. The hypothesis of enough stiffness

of the hull is true if we consider that the envelope is under full operational volume and pressure [19–21]. The rigidity assumption of a conventional airship is broken for extremely high speeds or if the envelope is made of thin films to reduce the weight [22,23], which is not the case of the NOAMAY airship.

Table 6. Process covariance matrix.

State	Weight	State	Weight
Φ [rad]	0.010	$\boldsymbol{a}$ [m/s^2]	0.100
Ω [rad/s]	0.100	u, v[m/s]	0.050
P_N [m]	0.001	P_E [m]	0.001
P_D [m]	0.010	w [m/s]	0.001

4.3. State Estimation Results

We show here the simulation results comparing the three approaches, for the state estimation in a trajectory tracking control case, as developed in [3]. The sampling rate of the estimators is 32 Hz and the wind is blowing from the North. A visualization of the estimation results, made in Simulink/Matlab, are available in youtu.be/VL5dvCyOZwY (EKF) and youtu.be/B26xaKtAyWo (UKF) accessed on 25 May 2022.

Figure 3(above) shows the estimated trajectories for the different estimators, from which we can see that EKF appears to be smoother and with less errors. The signals of the longitudinal (u), vertical (w) and angular velocity in z-axis (r) appear in Figure 3(down). We can see that the classical LPF provides good filtering, but with a significant delay, while UKF is faster though with a bigger error. The EKF filter has a time response similar to the UKF case, while displaying lower errors.

Thus, for the airship model considered here, the EKF showed a better performance in comparison to UKF and LPF. For further details, including the RMS values of the position errors, please refer to [3]. These results show the natural compromise between noise attenuation versus estimation delay, found in an estimator design.

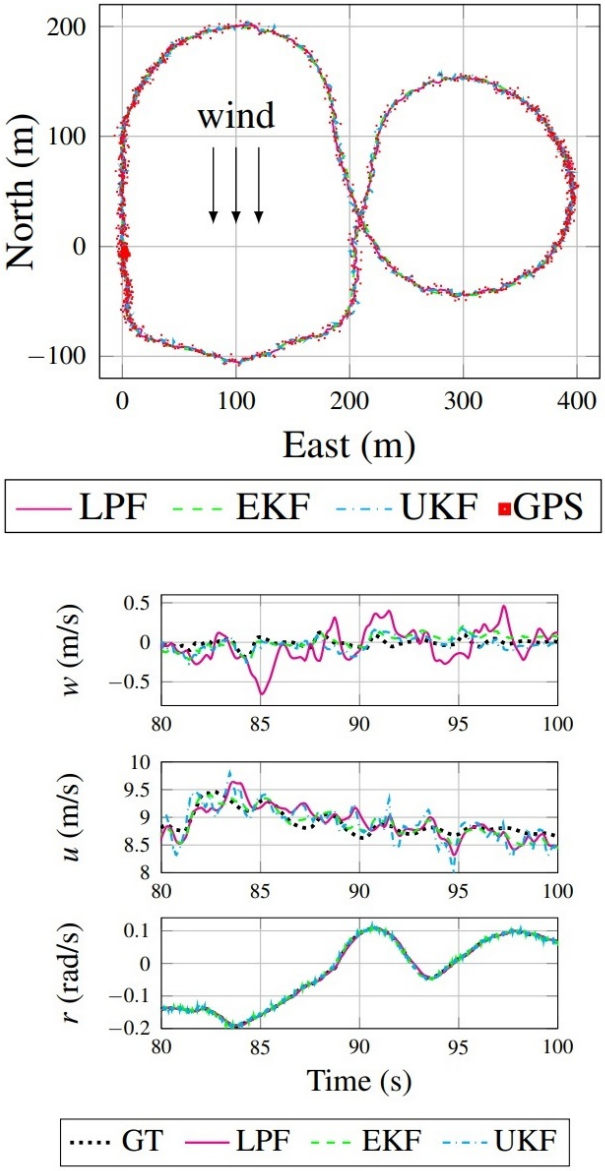

Figure 3. Estimated trajectory (**above**); Estimated vertical velocity (w), longitudinal velocity (u) and angular velocity (r) (**down**).

5. Wind Velocity Estimators

This section addresses the problem of wind velocity estimation. Firstly, we detail the equations related to the wind ($\mathbf{V_w}$) and inertial airship velocities ($\mathbf{V_g}$), as well as the airship orientation ($\mathbf{\Phi}$), which are related to the Pitot probe sensor.

Let us define the airship's relative velocity ($\mathbf{V_a}$), or airspeed, as:

$$\mathbf{V_a} = \mathbf{V_g} - \mathbf{V_w}, \tag{13}$$

Whose modulus is called *true airspeed* (V_t) and is given by:

$$V_t = ||\boldsymbol{V}_a||_2 = \sqrt{u_a^2 + v_a^2 + w_a^2}. \tag{14}$$

Let us recall also the aerodynamic angles, the sideslip β and the angle of attack α, as shown in Figure 4, defined by:

$$\beta = \sin^{-1} \frac{v_a}{V_t}, \tag{15}$$

$$\alpha = \tan^{-1} \frac{w_a}{u_a}. \tag{16}$$

Or, equivalently:

$$w_a = u_a \frac{\sin \alpha}{\cos \alpha} \quad \text{and} \quad v_a = V_t \sin \beta. \tag{17}$$

Finally, we obtain:

$$V_t = \frac{u_a}{\cos \alpha \cos \beta}. \tag{18}$$

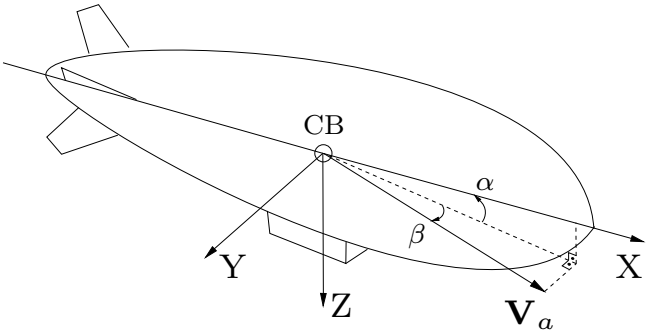

Figure 4. Sideslip angle (β) and angle of attack (α).

The Pitot tube is located at the airship nose (see Figure 2). Also, the airship has a pressure sensor connected to the Pitot Tube. Thus, as the airship moves forward, the longitudinal dynamic pressure ΔP will be measured by the pressure sensor. The relation between the measured pressure and the airship longitudinal airspeed can be derived from Bernoulli's equation as shown by [6], which can be rewritten as below:

$$\Delta P = \eta \left(u_a\right)^2, \tag{19}$$

where η is the calibrating factor related to the air density and pitot efficiency. Now, if we use the variable transformation below:

$$V_{pitot} = \sqrt{\Delta P}, \tag{20}$$

And using the statements (18) and (20), we can rewrite (19) as below:

$$V_t = \frac{V_{pitot}}{\sqrt{\eta} \cos \alpha \cos \beta}. \tag{21}$$

The unknown parameters in η together with the inaccurate angle values α and β are then estimated together in a single variable, or scale factor c_f, given by:

$$c_f = \sqrt{\eta} \cos \alpha \cos \beta, \tag{22}$$

And thus (21) is now given by:

$$V_t = \frac{1}{c_f} V_{pitot}. \tag{23}$$

Now, recalling that $\mathbf{V_g}$ is written in the body, as well as $\mathbf{V_a}$, if we take these vectors written in the inertial frame (NED), we can write the relative velocity equation (13) using the rotational matrix $\mathbf{S_\Phi}$ from airship body to NED frame, we have:

$$\mathbf{V_{NEDa}} = \mathbf{V_{NED}} - \mathbf{V_{NEDw}} = \mathbf{S_\Phi^T V_a}. \tag{24}$$

As the rotational operation does not change the vector modulus, and assuming a wind speed in the North-East plane, we have:

$$V_{pitot}^2 = c_f^2\left((V_N - V_{N_w})^2 + (V_E - V_{E_w})^2 + (V_D)^2\right). \tag{25}$$

Assuming initial conditions of $\alpha \approx \beta \approx 0$, we have $u_a = V_t$ and $v_a = w_a = 0$, and in the global frame we have:

$$V_{N_a} = u_a \cos\psi \cos\theta, \tag{26}$$

where ψ and θ are the yaw and pitch angles, respectively, which together with (13) give:

$$V_{E_a} = u_a \sin\psi \cos\theta, \tag{27}$$

$$V_N = \frac{V_{pitot}}{c_f} \cos\psi \cos\theta + V_{N_w}, \tag{28}$$

$$V_E = \frac{V_{pitot}}{c_f} \sin\psi \cos\theta + V_{E_w}. \tag{29}$$

The measures of V_{pitot}, V_N and V_E come from the Pitot tube plus the GPS. In the same way, the Euler angles (ϕ, θ and ψ) are obtained from the IMU. Therefore (25), (28) and (29) can be used as observation equations, while c_f, V_{N_w} and V_{E_w} are the estimated states in the EKF Kalman Filter. Although these equations are linear dependent, we are introducing redundant information of different sensors by adding IMU combined with GPS measurements, which may lead to faster convergence and better filtering of noise.

5.1. Extended Kalman Filter

In this section we propose an EKF filter to estimate the wind speed in the airship body. Supposing that the wind is blowing in the North-East plane, the objective here is to estimate the wind velocity in the horizontal plane as well as the Pitot scale factor. By introducing both (28) and (29) in the measurement update stage, we expect to improve the results, in contrast when using just (25), as in the aproach of [6].

We consider here that the wind is constant with an additional Gaussian variance component. In addition, as the uncertain and time varying factor c_f is estimated at the same time, we have the following model:

$$\boldsymbol{\chi}_{k+1} = \mathbf{F}\boldsymbol{\chi}_{\mathbf{k}} + \boldsymbol{\nu}_{\mathbf{k}}, \tag{30a}$$

$$\mathbf{z}_{\mathbf{k}} = \mathbf{h}(\boldsymbol{\chi}_{\mathbf{k}}) + \boldsymbol{v}_{\mathbf{k}} \tag{30b}$$

where: $\boldsymbol{\chi}_k = [V_{N_{wk}}\ V_{E_{wk}}\ c_{fk}]^T$ is the state vector in the instant $t = kt_s$; $\mathbf{z_k} = [V_{pitot_k}^2\ V_{N_k}\ V_{E_k}]^T$ is the system output in the instant $t = kt_s$; t_s is the sample time in seconds; $\boldsymbol{h}(\boldsymbol{\chi}_k)$ is the output function, computed through (25), (28) and (29), and from which the Jacobian matrix is derived. Further, $\boldsymbol{F} = I_3$, is the identity matrix, $\boldsymbol{\nu}_k \sim \mathbf{N}(\mathbf{0}, \mathbf{Q})$ is the process noise with Gaussian distribution and covariance $\mathbf{Q}$; and, finally, $\boldsymbol{v}_k \sim \mathbf{N}(\mathbf{0}, \mathbf{R})$ is the measurement noise also with Gaussian distribution and covariance $\mathbf{R}$. Given the model described in (30a), we can use the traditional EKF algorithm, as presented in [3].

5.2. Neural Network

This section presents the design of a Neural Network (NN) algorithm for the estimation of the wind speed and angle, supposing an horizontal wind profile. One of the advantages of the use of a Neural Network is that it is trained by measured data, and it is thus able to detect abrupt variations in the wind velocity. However, in contrast, it is usually more sensitive to measurement errors. We will show that combining NN with the classical estimators, using Kalman filters, we can benefit of the advantages of both.

As presented above in equations (30a)–(30b), the wind model output is composed by (25), (28) and (29), which are nonlinear equations in the model states, vehicle velocity, orientation and Pitot pressure. To obtain an easier and more tractable model, the measured data vector is transformed into 8 inputs given by:

$$\mathbf{z_{nn}} = \begin{bmatrix} V_{pitot}^2 \\ V_D^2 \\ V_N \\ V_E \\ V_E^2 \\ V_N^2 \\ V_{pitot} \cos\psi \cos\theta \\ V_{pitot} \sin\psi \cos\theta \end{bmatrix}.$$

Meanwhile, the output vector χ_{nn} of the Neural Network contains the estimated wind speed components and the scale factor c_f as:

$$\chi_{nn} = \begin{bmatrix} V_{N_w} \\ V_{E_w} \\ c_f \end{bmatrix}.$$

The proposed Neural Network consists of a three-layer fitting NN, which has three nonlinear hidden layers containing 24 neurons each and three linear outputs. The activation function of the nonlinear neurons is sigmoidal. It was designed with the help of the MATLAB Neural Network Toolbox^{TM}. For more details, the reader is referred to [3].

5.3. Hybrid Estimator

Following the developments in [3], we propose now a hybrid estimator that is a combination of both estimators, the EKF designed in Section 5.1 and the Neural Network designed in Section 5.2. The idea is to change the measure update stage of the EKF approach, when the output of the NN χ_{nn} is added to the measurement vector of the EKF as a redundant measure. Thus, resulting in the new measurement vector $\mathbf{z_{h_k}}$, updating function $\mathbf{h_{h_k}}(\chi_\mathbf{k})$ and its respective Jacobian $\mathbf{H_{h_k}}$ given below:

$$\mathbf{z_{h_k}} = \begin{bmatrix} \mathbf{z_k} \\ \chi_{nn} \end{bmatrix}, \mathbf{h_{h_k}}(\chi_\mathbf{k}) = \begin{bmatrix} \mathbf{h}(\chi_k) \\ \chi_k \end{bmatrix} \text{ and } \mathbf{H_{h_k}} = \begin{bmatrix} \mathbf{H} \\ \mathbf{I_3} \end{bmatrix},$$

where $\mathbf{I_3}$ is the identity matrix. Then the EKF standard algorithm is used by updating the matrices $\mathbf{C_k}$, $\mathbf{K_k}$ and $\mathbf{R}$, as shown in [3].

5.4. Wind Estimation Results

In this section, we run a simulation to evaluate the three approaches presented, establishing a comparison with the traditional model-based approach proposed by [6]. The sensors models in the simulation environment use sample frequency as follows: IMU (100 Hz), GPS (4 Hz) and Pitot tube (18 Hz). An online repository (https://github.com/leve-fem/airship_estimatoraccessed on 25 May 2022) is available containing all the approaches.

As shown in Figure 5, the simulation considers wind with absolute value $|\vec{V}_w| = 2m/s$ and heading $\psi_w = \frac{\pi}{2}$ rad (blowing from East to West). Then in the instant $t = 160s$ the wind is intensified to $|\vec{V}_w| = 3m/s$ and its heading is changed to $\psi_w = \pi$ (blowing from South to North). In the same Figure we highlighted five special moments with gray background in order to establish further comparisons with the results in Figure 6. Moreover, we introduced results using the estimator proposed by [6] as "Cho2011" for comparison.

From Figure 6, we can see that under an abrupt wind variation, as in instant (III), the classical estimators (EKF and UKF) do not converge immediately as they are model-based estimators depending directly on the Pitot information. Meanwhile the NN clearly, which is data driven, shows an instantaneous wind correction in this situation, although exhibiting

a biased value. The hybrid approach was also able to yield a correct estimation of the wind speed before the instants (IV) and (V), when the Model-based estimators finally converges for the acceptable values.

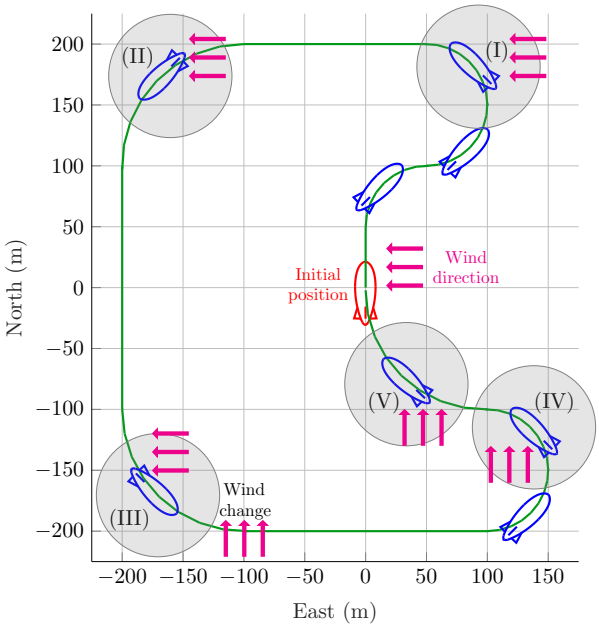

Figure 5. Simulation trajectory for wind estimation.

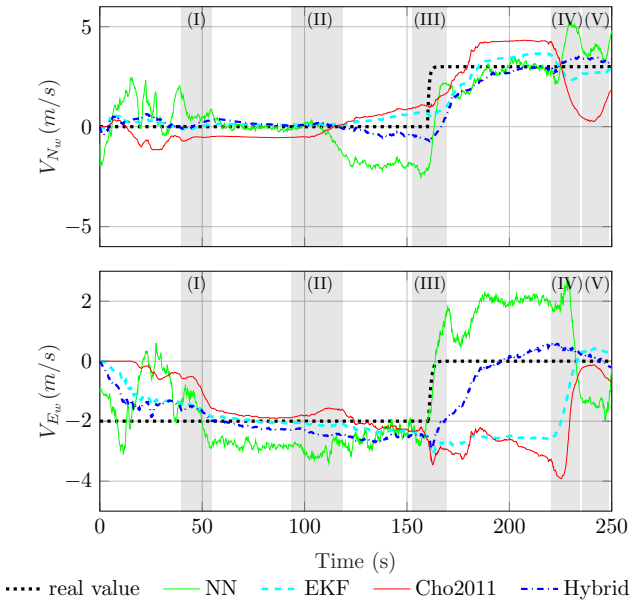

Figure 6. Simulation using the airship nonlinear model with realistic sensor noise: estimated wind velocity components in NE frame.

In conclusion, we can observe that the NN had a better estimation of V_{E_w} in comparison to the Model-based approaches. However, for the component V_{N_w}, the Model-based approaches presented a better performance. Meanwhile the "Hybrid" approach had the best performance in the estimation of V_{E_w} and an acceptable estimation for V_{N_w}.

6. Mitigating Filtering Design Issues

In this section, we present possible solutions to mitigate two common problems arriving in filtering implementation that are the control input redundancy and the delay

appearing in second-order filters. Input redundancy is related to an overactuated control system. A common issue is that the redundant inputs can cancel each other when the system achieves a stationary state condition which results in more energy consumption. One solution for this problem is to perform a filtering in the commanded redundant input which is less important for maintaining the equilibrium point, which is known as Washout Filter (WOF) [13].

Another critical point is the drawback that appears in the second-order filters used to estimate state derivatives. The natural dynamics delay produced by these filters may impose serious problems to the control system feedback. One possible solution to mitigate this delay is the so called Input Scaling Gain (ISG), that was first proposed by our research group as a scalar factor gain [14]. In this section, we generalize the input scale gain to the MIMO case. To illustrate these two design solutions we present briefly the concept of Incremental Controllers, that are used to show the benefits of these filtering techiques, although not limited to these kind of controllers.

6.1. Incremental Controllers

In order to illustrate the application of these two design techniques (WOF and ISG), we present first the concept of incremental dynamics (ID) and incremental controllers.

In the last two decades, researchers have studied the classical Nonlinear Dynamics Inversion (NDI), also known as feedback linearization, as a promising approach to unify the control scheme of an Unmanned Aerial Vehicles (UAV) during the different phases of a standard flight plan [24–26].

To cancel model nonlinearites, NDI controllers require a complete and precise model of the system. However, in real-world systems, realistic dynamic models with accurate parameters are almost impossible to be obtained. Firstly presented in [27], the incremental controlleres (ID) was used for designing a simplified version of Nonlinear Dynamic Inversion. Later this strategy was named as Incremental Nonlinear Dynamic Inversion (INDI) [28]. Since then, several works use Incremental Dynamics (ID) for designing nonlinear control laws. As some examples, in [29], the authors use ID for designing Incremental Backstepping (IBKS). In [30], ID is used for design the Incremental Sliding-Mode (ISM) control law.

As a first step, let us present the concept of incremental dynamics [28]. Consider a control affine nonlinear system in state space representation:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}, \tag{31a}$$

$$\mathbf{y} = h(\mathbf{x}). \tag{31b}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of state variables, $\mathbf{u} \in \mathbb{R}^m$ is the vector of control inputs, $\mathbf{y} \in \mathbb{R}^l$ is the output vector, and f, g, h are real analytic Lipschitz continuous functions.

The system dynamics (31a) can be approximated by its Taylor series expansion around $\mathbf{x} = \mathbf{x}_0$ and $\mathbf{u} = \mathbf{u}_0$:

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_0 + \mathbf{A}_0(\mathbf{x} - \mathbf{x}_0) + \mathbf{B}_0(\mathbf{u} - \mathbf{u}_0) + \mathcal{O}((t - t_0)^2), \tag{32}$$

where $(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u})$ and $(\mathbf{x}_0, \dot{\mathbf{x}}_0, \mathbf{u}_0)$ are respectively the state, the state derivative, and input at current time t and some previous time $t_0 < t$, $\mathcal{O}((t - t_0)^2)$ includes the higher order terms of the Taylor expansion, and

$$\mathbf{A}_0 = \frac{\partial}{\partial \mathbf{x}}[f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}_0]\bigg|_{\mathbf{x}_0}, \quad \mathbf{B}_0 = g(\mathbf{x}_0), \tag{33}$$

are state-dependent matrices that capture the linear system dynamics relationship with the state and input variables, respectively.

One of the key points in incremental controllers is that we assume that the time interval $t_s = t - t_0$ elapsed between $\mathbf{x}_0$ and $\mathbf{x}$ is sufficiently small, such that we can suppose $\mathbf{x}_0 \approx \mathbf{x}$, and thus the system dynamics (32) can be approximated by the so-called incremental dynamics formulation:

$$\dot{\mathbf{x}} \approx \dot{\mathbf{x}}_0 + \mathbf{B}_0(\mathbf{u} - \mathbf{u}_0), \tag{34}$$

yielding the current state derivative from the knowledge of its value at the previous time step and the input increment $\Delta\mathbf{u} = \mathbf{u} - \mathbf{u}_0$.

Incremental controllers such as IBKS [29], ISM [30] and INDI [14] are sensor-based controllers, taking advantage of the simplified dynamics (34), where the use of state dependent dynamics is replaced by the measurement of the previous time derivative $\dot{\mathbf{x}}_0$.

INDI is the equivalent of the well known NDI control applied to the incremental dynamics (34).

Let us impose a desired dynamics $\boldsymbol{\nu} \in \mathbb{R}^n$. Then, the incremental dynamic inversion results in the following control law:

$$\mathbf{u} = \mathbf{u}_0 + \mathbf{B}_0^+(\boldsymbol{\nu} - \dot{\mathbf{x}}_0), \tag{35}$$

where $\mathbf{B}_0\mathbf{B}_0^+ = \mathbf{I}_n$ is the identity matrix of order n.

Note that if the matrix inversion is perfect then, replacing the control law in the incremental dynamic equation gives

$$\dot{\mathbf{x}} \approx \dot{\mathbf{x}}_0 + \mathbf{B}_0\Delta\mathbf{u} = \dot{\mathbf{x}}_0 + \mathbf{B}_0\mathbf{B}_0^+(\boldsymbol{\nu} - \dot{\mathbf{x}}_0) = \boldsymbol{\nu} \tag{36}$$

which shows that:

- the system modes are decoupled, allowing the design of independent linear controllers for each of them.
- the state derivative tracks the dynamics imposed by $\boldsymbol{\nu}$;
- the previous state derivatives, and consequently their nonlinearities, are canceled (DI is also called feedback linearization);

Taking advantage of the above, we can define $\boldsymbol{\nu}$ as a pseudo-control signal, which is usually taken as a linear state feedback. Imposing a linear dynamics, the feedback gain places the closed loop poles, designing the desired response. As it is common sense in cascade control, a Time Scale Separation Principle (TSSP) must be respected, and the INDI loop must converge faster than the linear control loop. In addition, the implementation of incremental controller considers the following assumptions:

A6.1 The system is output controllable (34), and any internal dynamics are intrinsically stable in closed-loop.

A6.2 States are sampled at a sufficiently high frequency when compared with system dynamics.

A6.3 Fast control action in comparison to the system modes.

A6.4 The control signals and state references are measurable, continuous and bounded. Additionally, accurate information on the state derivatives and actuator variables is available.

A6.5 The input matrix B_0 has known coefficient signals, and it is non-singular around the region of interest.

Since all assumptions of Incremental controllers are satisfied we can define the INDI control law (35) for the system in (31a). Figure 7 shows a typical block diagram of a sensor based INDI control, where z^{-1} represents a delay of one sample time.

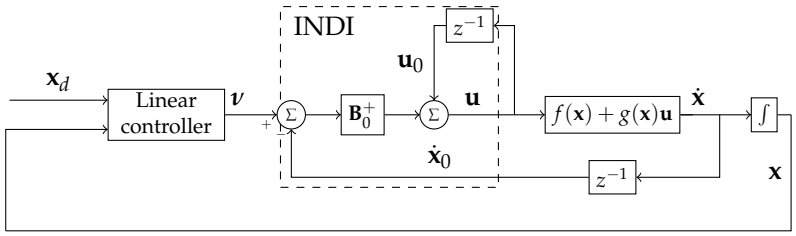

Figure 7. INDI closed loop block diagram.

One of the most important advantages of using incremental controllers, which makes it a successfull candidate for real applications, is the great robustness to model parameters. The most common control approaches are said to be model based, for their design being strongly dependent on the system dynamics model. This aspect requires the definition of an accurate mathematical representation, as well as a careful system identification process. Nevertheless, the resulting model will still be subject to noise, disturbances and remaining model uncertainties.

The approach based on the incremental dynamics on its hand requires only the identification of the input matrix, neglecting the parameters that depend exclusively on the internal states and it is expected to be robust to model uncertainties. Furthermore, the incremental formulation is very simple and intuitive, supported on well established mathematical fundamentals, with only a few parameters to set-up.

On the other hand, restrictions on the use of incremental controllers arise out of assumptions *A6.1* to *A6.5*. Initially, as a consequence of the model simplifications, all the needed information about the system states is obtained from measurements. Thus, to ensure quality in the measurements, the feedback states and state derivatives are to be updated at a sufficiently high sampling rate, with good quality sensors. Furthermore, *A6.3* implies that input control signal must have dominance on the system dynamics, demanding fast actuation when compared with system modes. These assumptions are satisfied in the most of UAVs since the control actuation has the greater influence over the dynamics.

In the following subsections we present and analyze the two techniques (WOF and ISG) proposed to mitigate the filtering design problems of input redundancy and second-order filtering delay. The results are illustrated through numerical examples considering an application for the INDI control.

6.2. Input Redundancy Treatment

If we consider that the input matrix $\mathbf{B}_0$ from (35) may not be square, we can expect an input redundancy to occur. Therefore, there are various possible solutions of inputs $\mathbf{u}_e$ that achieve a given equilibrium point $\mathbf{x}_e$. A common issue is that the redundant inputs can cancel each other when the system achieves a stationary state condition, resulting in more energy consumption.

As an example, consider a simplified vehicle dynamics with a single state variable x given by the longitudinal velocity and two inputs u_1 and u_2. This system dynamics can be represented by the following mathematical modeling:

$$\dot{x} = f(x) + g_1(x)u_1 + g_2(x)u_2. \tag{37}$$

Both control inputs u_1 and u_2 have influence over the vehicle velocity dynamics. Now consider an equilibrium point $x_e > 0$, thus $\dot{x}_e = 0$ and the following can be stated:

$$\begin{bmatrix} b_1 & -b_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = -f(x_e), \tag{38}$$

where $b_1 = g_1(x_e)$ and $b_2 = -g_2(x_e)$ are positive constants.

In this case, there are many choices for u_1 and u_2 resulting in the same constant longitudinal velocity x_e. However, this choice will impact in the energy consumption. The ideal solution is to use the minimal control effort in order to save energy.

One solution for this problem is to perform a filtering in the commanded redundant input which is less important for maintaining the equilibrium point. As a result we obtain a system dynamics described in the following form:

$$\dot{x} = f(x) + g_1(x)u_1 + g_2(x)\hat{u}_2, \tag{39a}$$

$$\dot{\hat{u}}_2 = \frac{-1}{\tau}\hat{u}_2 + u_2. \tag{39b}$$

where $\hat{u}_2$ is the filtered commanded input for the brake. By imposing this dynamics, u_2 will naturally converge to zero and u_1 will also reduce, once u_2 is no longer canceling it. Note that u_2 is still useful for the transient state condition when the vehicle needs to slowdown fast. This strategy is commonly referenced in the literature as Washout Filter (WOF) [13].

The solution can be extended for systems with Multiple Inputs and Multiple Outputs (MIMO). Consider the following generalized MIMO system dynamics:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}, \tag{40}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$ and $m \geq n$.

Thus, applying the WOF to the redundant inputs we obtain the following extended dynamics:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\hat{\mathbf{u}}}_s \end{bmatrix} = \begin{bmatrix} \mathbf{I}_n & g_s(\mathbf{x}) \\ \mathbf{0}_{l\times n} & \mathbf{T}_s \end{bmatrix} \begin{bmatrix} f(\mathbf{x}) \\ \hat{\mathbf{u}}_s \end{bmatrix} + \begin{bmatrix} g_m(\mathbf{x}) & \mathbf{0}_{n\times l} \\ \mathbf{0}_{l\times n} & \mathbf{I}_l \end{bmatrix} \begin{bmatrix} \mathbf{u}_m \\ \mathbf{u}_s \end{bmatrix} \tag{41}$$

where: $l = m - n$, $\mathbf{x} \in \mathbb{R}^n$ is the vector of states; $\mathbf{u}_m \in \mathbb{R}^n$ is the vector of main actuators; $\mathbf{u}_s \in \mathbb{R}^l$ is the vector of secondary (or redundant) actuators; $\hat{\mathbf{u}}_s \in \mathbb{R}^l$ is the vector of filtered input signals, $f(\mathbf{x})$ is the function of state dynamics; $g_m(\mathbf{x})$ is a function which describes the influence of $\mathbf{u}_m$ in the state dynamics; $g_s(\mathbf{x})$ is a function which describes the influence of $\mathbf{u}_s$ in the state dynamics; $\mathbf{T}_r = \text{diag}([-1/\tau_1; -1/\tau_2; -1/\tau_3; \cdots; -1/\tau_l])$ is a diagonal matrix with positive constants $\tau_1, \cdots, \tau_l$ chosen by the designer; $\mathbf{I}_n$ is the identity matrix of order n; and $\mathbf{0}_{n\times l}$ is a matrix full of zeros with n lines and l columns.

The vector of redundant actuators can be chosen by analyzing the input function $g(\mathbf{x})$ through a systematic procedure. First, the designer must linearize $g(\mathbf{x})$ in a chosen point $\mathbf{x}_0$, obtaining the following:

$$\mathbf{B}_0 = \frac{\partial}{\partial \mathbf{x}} g(\mathbf{x})\Big|_{x_0}. \tag{42}$$

By analyzing the matrix $\mathbf{B}_0 \in \mathbb{R}^{n\times m}$, the designer must identify the linearly dependent columns, which indicates the redundant inputs. After identifying the redundant actuators, the designer must choose and separate between main ($\mathbf{u}_m$) and secondary ($\mathbf{u}_s$) actuators, by also defining the functions $g_m(\mathbf{x})$ and $g_s(\mathbf{x})$. Then the extended dynamics (41) can be applied. This solution can be represented by a block diagram in a cascaded form as shown in Figure 8, where $\hat{\mathbf{u}} = [\mathbf{u}_m\ \hat{\mathbf{u}}_s]^T$ and $g(\mathbf{x}) = [g_m(\mathbf{x})\ g_s(\mathbf{x})]$.

Numerical Example

As an example consider the following linear system with two redundant inputs:

$$\dot{x} = x + 0.2u_1 - 0.8u_2 \tag{43}$$

where x is the state, u_1 is the first input and u_2 is the second input.

By applying INDI, we obtain the following control law:

$$\mathbf{u} = \mathbf{u}_0 + \mathbf{B}_0^+(\nu - \dot{x}_0) \tag{44}$$

where $\mathbf{u} = [u_1 \quad u_2]^T$, $\mathbf{B}_0 = [0.2 \quad -0.8]$, $\mathbf{u}_0$ is the previous input and $\nu = k(x_d - x)$ is a desired dynamics of first order. The block diagram in Figure 8 illustrates the closed loop system, where z^{-1} represents a delay of one sample time $t_s = 0.01$ seconds, $\hat{\mathbf{u}} = [u_1 \quad \hat{u}_2]^T$ is the control signal filtered by the WOF, $k = 20s^{-1}$ is a linear gain and $\tau = 5$ seconds is the time constant of the WOF. We suppose the presence of actuator saturation ($-0.5 \leq u_1 \leq 0.5$ and $-0.5 \leq u_2 \leq 0.5$,) which is treated with an anti-windup strategy [3].

Figures 9 and 10 show the results for a step in x_d using WOF denominated case (a) (or "INDI+WOF") and without WOF denominated case (b) (or "INDI"). For sake of comparison consider the following quadratic cost function:

$$J_u = \int_0^{t_f} u_1^2(t) + u_2^2(t)dt \tag{45}$$

where $t_f = 15$ seconds is the final time of simulation.

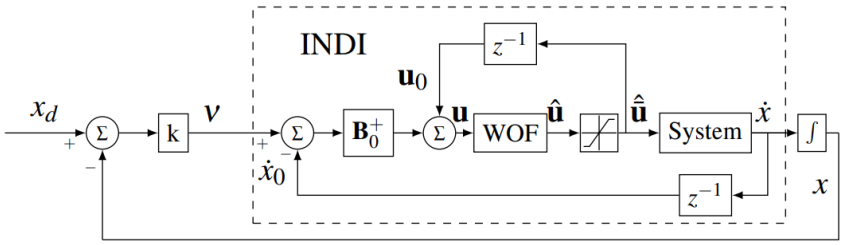

Figure 8. Numerical Example: Block diagram of the closed loop system of INDI+WOF.

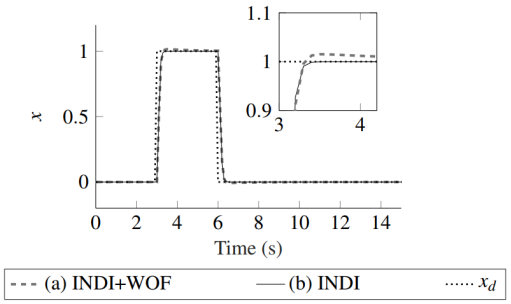

Figure 9. Numerical example: comparison of tracking performance between INDI+WOF and INDI

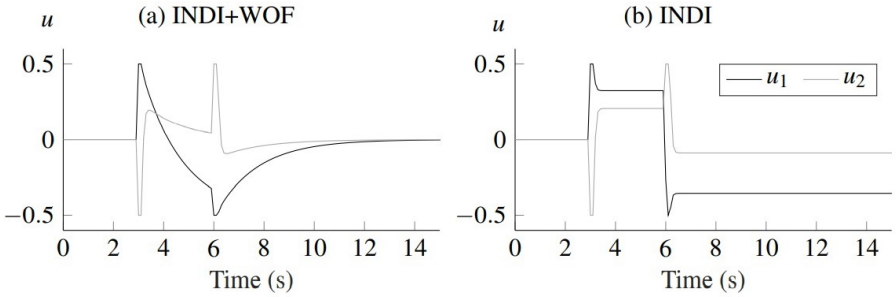

Figure 10. Numerical example: comparison of input signals between INDI+WOF and INDI

From the figures, we see that the tracking performance is similar for both cases, which does not happen for the control effort. In Figure 10, the saturation is achieved in both control inputs for both cases (a) and (b). In case (b) the commanded signals u_1 and u_2 converge to values different from zero after the transient state ($t > 6$ seconds). Consequently the system has higher energy consumption in case (b) with $J_u = 17.85$. In

case (a) the energy consumption is reduced to $J_u = 5.74$. Therefore, the WOF appears as a simple and advantageous solution for systems with redundant actuation with actuator saturation, such as aerial vehicles i.e. multirotor drones, aircrafts and airships.

6.3. The Second-Order Differentiator and the Estimation of Derivatives

One important problem in control design, not limited to incremental controllers, is the need for the derivative of a given state. They may be difficult to obtain or even impossible, as is the case of the derivative of angular rates. Further, direct numerical differentiation of the output signal may lead to noise amplification and abrupt variations.

A common solution to this problem is the use a Second-Order Differentiator (SOD) to obtain the derivative of the state. The SOD works as a low-pass filter for the state yielding a filtered output of the measure $\mathbf{y}$. While providing an estimation of the state derivative, the SOD simultaneously attenuates the high-frequency noise in the measure signal.

If we call the state derivative vector as $\mathbf{w} = \dot{\mathbf{x}}$, the estimation of the derivative $\hat{\mathbf{w}}$, using the second-order-derivative (SOD), is given by:

$$\hat{\mathbf{w}}(s) = \left[\frac{\omega_n^2 s}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right]\mathbf{y}(s)\,. \tag{46}$$

In the design procedure, the filter parameters like the natural frequency ω_n and the damping ratio ζ are used to adjust the passband mid-frequency, as well as the passband size.

6.4. Input Scaling Gain

A drawback that comes with the SOD filter is the natural delay produced by the second order dynamics, which may harm the control system feedback. Thus, it is important to mitigate this delayed estimation in order to robustify the controller.

One possible solution to this issue is the so called Input Scaling Gain (ISG), that was first proposed by our research group in 2015 [14], as a scalar factor. In this paper, we propose the generalization of the scale gain to MIMO systems, using a diagonal matrix $\mathbf{\Lambda} \in \mathbb{R}^{n\times n}$. In the case of the incremental controller design, the ISG is used to scale down the difference $(\boldsymbol{\nu} - \dot{\mathbf{x}}_0)$ in (35), reducing the bandwidth of the closed-loop system. The matrix diagonal elements λ_{ii} are real numbers in the interval $]0,\ 1]$, such that the modified INDI control law is now given as:

$$\mathbf{u} = \mathbf{u}_0 + \mathbf{B}_0^{+}\mathbf{\Lambda}(\boldsymbol{\nu} - \dot{\mathbf{x}}_0)\,, \tag{47}$$

Note that with $\Lambda = \mathbf{I}_n$ we have the traditional INDI control law (35).

Substituting (47) in the closed loop Incremental Dynamics (34) results in:

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_0 + \mathbf{\Lambda}(\boldsymbol{\nu} - \dot{\mathbf{x}}_0)\,. \tag{48}$$

Considering the discrete implementation, where the time interval between t_0 and t is $\Delta t = T_s$, we can rewrite (48) as:

$$\frac{\Delta\dot{\mathbf{x}}}{\Delta t} = \frac{\Delta\dot{\mathbf{x}}}{T_s} = \frac{1}{T_s}\mathbf{\Lambda}(\boldsymbol{\nu} - \dot{\mathbf{x}}_0)\,, \tag{49}$$

Assuming a fast sampling rate (small Δt), and denoting $\dot{\mathbf{x}}$ as $\mathbf{w}$, (49) approximates to:

$$\dot{\mathbf{w}} \simeq \frac{1}{T_s}\mathbf{\Lambda}(\boldsymbol{\nu} - \mathbf{w}_0)\,. \tag{50}$$

Considering $t_0 = t - T_s$, (50) can be rewritten as:

$$\dot{\mathbf{w}}(t) = \frac{1}{T_s}\mathbf{\Lambda}[\boldsymbol{\nu}(t) - \mathbf{w}(t - T_s)]\,. \tag{51}$$

Applying Laplace transform to (51), provides the transfer function between the state derivative and the pseudo-control as:

$$\begin{aligned}\mathcal{L}\left\{\dot{\mathbf{w}}(t)+\frac{1}{T_s}\mathbf{\Lambda}\mathbf{w}(t-T_s)\right\}&=\frac{1}{T_s}\mathbf{\Lambda}\mathcal{L}\{\boldsymbol{\nu}(t)\}\Leftrightarrow\\ \Leftrightarrow T_s\mathbf{\Lambda}^{-1}s\mathbf{w}(s)+e^{-sT_s}\mathbf{w}(s)&=\boldsymbol{\nu}(s)\Leftrightarrow\\ \Leftrightarrow \mathbf{w}(s)=\left(T_s\mathbf{\Lambda}^{-1}s+e^{-sT_s}\mathbf{I}\right)^{-1}\boldsymbol{\nu}(s)&=\operatorname{diag}\left\{\frac{1}{\frac{T_s}{\lambda_{ii}}s+e^{-sT_s}}\right\}\boldsymbol{\nu}(s)\,.\end{aligned}\tag{52}$$

Some important points to be remarked are:

- each component of ν_i of pseudo-control $\boldsymbol{\nu}$ has an independent time constant $\frac{T_s}{\lambda_{ii}}$, as the scale matrix $\mathbf{\Lambda}$ is a diagonal matrix.
- Under fast sampling, we can assume $e^{-sT_s}\simeq 1$ implying that (52) simplifies to a simple first order low-pass filter, attenuating high-frequency perturbations in the pseudo-control ν;
- as a possible drawback, the ISG reduces the bandwidth of the closed-loop dynamics, eventually decreasing the overall performance;

6.5. Combining ISG and SOD: Closed-Loop Analysis

In order to investigate the stability properties as well as the performance improvements against delays in the closed-loop system, we analyze here the proposed SOD+ISG solution for a first-order SISO system, as shown in Figure 11.

Although illustrated here for a SISO system, the approach can be extended to multiple-input-multiple-output (MIMO) systems as the diagonal structure of the ISG yields independent components z_i of the INDI-loop error $\mathbf{z}=\boldsymbol{\nu}-\mathbf{w}_0$, and the SOD is also independent for each state x_i.

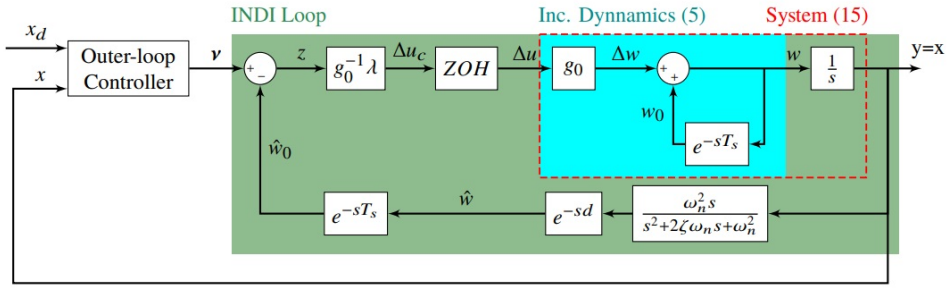

Figure 11. Block diagram of INDI controller applied to a first-order plant.

Firstly, to consider continuous systems, the discrete sample delays z^{-1} of INDI (see Figure 7) are substituted by time delay components e^{-sT_s}. Further, to simulate the discrete feature of INDI we add a Zero-Order Hold (ZOH) in the control loop, as well as an extra transport delay d to investigate the effects of additional unmodeled delays in the loop.

6.5.1. Analytical Formulation

The system to be controlled is defined as:

$$\begin{cases}\dot{x}=\dot{x}_0+g(x_0)\Delta u\\ y=x\end{cases},\tag{53}$$

where its dynamics is described using ID formulation (34), and its output is the state itself.

In this section we only analyze the local behavior of the system, where the control effectiveness function $g(x_0)$ is approximated by a constant g_0.

The transfer function $H_1(s)$ relating the state derivative w with the input increment Δu is given (34) by:

$$H_1(s) = \frac{w(s)}{\Delta u(s)} = g_0\left(\frac{1}{1-e^{-sT_s}}\right), \tag{54}$$

and, thus, the transfer function $H_2(s)$ is given (53) by:

$$H_2(s) = \frac{y(s)}{\Delta u(s)} = \frac{1}{s}H_1(s)\,. \tag{55}$$

One important feature of the INDI controller, as shown in (36), is that the state derivative w should follow a given pseudo-control ν. Therefore, it is important to analyze the transfer function from ν to w, denoted here as $H_3(s)$, shown in the block diagram of Figure 11, and given by:

$$\begin{aligned} H_3(s) = \frac{w(s)}{\nu(s)} &= \frac{g_0^{-1}\lambda Z(s)H_1(s)}{1+g_0^{-1}\lambda Z(s)H_2(s)\frac{\omega_n^2 s}{s^2+2\zeta\omega_n s+\omega_n^2}e^{-s(T_s+d)}} \\ &= \frac{g_0^{-1}\lambda Z(s)H_1(s)\left(s^2+2\zeta\omega_n s+\omega_n^2\right)}{s^2+2\zeta\omega_n s+\omega_n^2+g_0^{-1}\lambda\omega_n^2 e^{-s(T_s+d)}Z(s)H_2(s)s}, \end{aligned} \tag{56}$$

where $Z(s) = \frac{1-e^{-sT_s}}{sT_s}$ is the zero-order-holder.

Substituting (54), (55), and $Z(s)$ into (56), we finally come to the transfer function from ν to w as:

$$H_3(s) = \frac{\frac{\lambda}{T_s}\left(s^2+2\zeta\omega_n s+\omega_n^2\right)}{s^3+2\zeta\omega_n s^2+\omega_n^2 s+\frac{\lambda}{T_s}\omega_n^2 e^{-s(T_s+d)}}. \tag{57}$$

Note that (57) is a delayed system where the delay appears in the constant term of the characteristic equation, such that the simple first-order Padé approximation

$$e^{-sT} \approx \frac{1-sT/2}{1+sT/2}\,. \tag{58}$$

can be used to investigate the stability analysis of the closed-loop system.

The resulting transfer function $H_3(s)$ is finally approximated to:

$$H_3(s) = \frac{\frac{\lambda}{T_s}\left(1+s\frac{(T_s+d)}{2}\right)\left(s^2+2\zeta\omega_n s+\omega_n^2\right)}{\frac{(T_s+d)}{2}s^4+(\zeta\omega_n(T_s+d)+1)s^3+\left(2\zeta\omega_n+\frac{(T_s+d)}{2}\omega_n^2\right)s^2+\left(\frac{(2-\lambda)T_s-\lambda d}{2T_s}\omega_n^2\right)s+\frac{\lambda}{T_s}\omega_n^2}. \tag{59}$$

From which we can conclude that:

- As desired in the INDI approach, w will follow a pseudo control ν, due to the fact that the transfer function has unitary dc-gain;
- Considering the Routh–Hurwitz criterion, $H_3(s)$ will be stable for positive coefficients in the characteristic polynomial, which implies that $(2-\lambda)T_s - \lambda d > 0$;
- Further, the ISG is also effective to mitigate the effects of additional delays in the closed-loop. For example, without the use of ISG ($\lambda = 1$), the extra delay should never be greater than the sample time T_s.

6.5.2. Numerical Example

A numerical example is presented here to investigate the effect of the ISG gain and the SOD differentiator approach in the control loop. To investigate the role of the delay in the INDI control, let us analyze the scenario in which there is an extra delay $d > T_s$ in the INDI feedback loop. Figure 12 shows the results for the step inputs applied to $H_3(s)$ and the corresponding Bode plots, assuming a sampling frequency of 50 Hertz.

The simulations were done for three diferent conditions: a) Varying ISG scale gains for a given measured state derivative (note that lower λ implies lower natural frequency and bigger damping ratio); b) Varying ISG scale gains for a differentiator with a given natural frequency of $\frac{15}{T_s}$; and c) Varying the natural frequencies of the differentiators for a fixed ISG gain equal to 0.5.

The step response plots, without using the ISG (equivalent to $\lambda = 1$), shows that the system is unstable for this plant that has an extra delay d larger than the sample time T_s. However, the system becomes stable if a lower value of λ is considered. For the case where the bandwidth of the differentiator satisfies the relation $\omega_n T_s = 15$, the extra delay d added by the filter destabilizes the system even if we use an ISG gain of 0.8, while an ISG gain of 0.5 is sufficient to make it stable for different SOD frequencies.

From these results, we conclude that the combined approach SOD + ISG yields a useful and practical derivative estimator, since the ISG can mitigate the effects of the delay generated by the second-order differentiator. Additionally, we remark that it can even mitigate other possible delays appearing in the feedback loop.

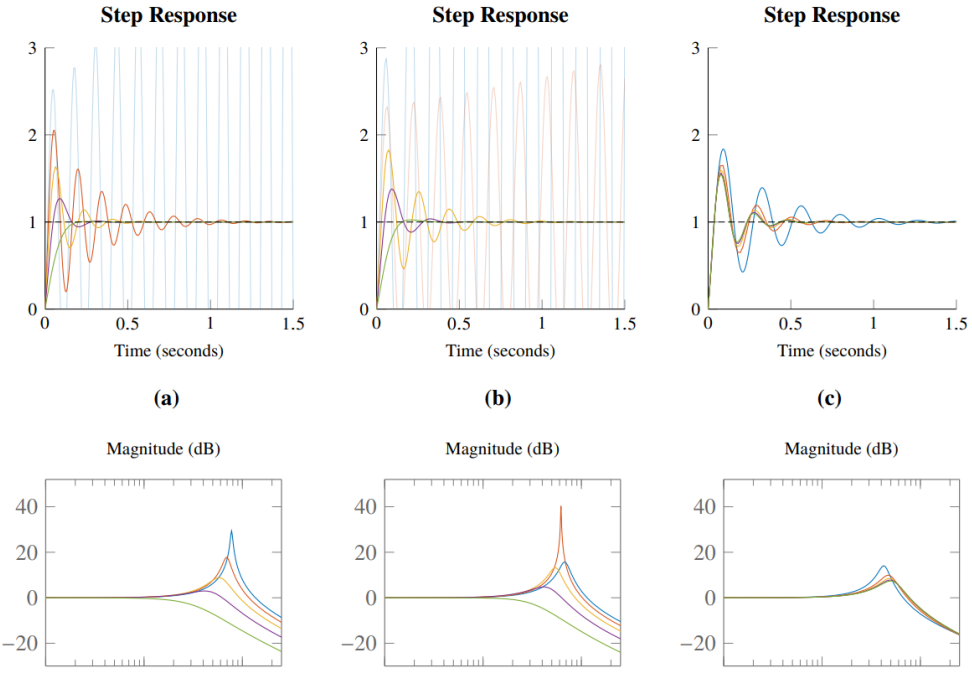

Figure 12. Step responses and bode plots of the INDI loop with exta loop delay $d = 1.1T_s$ $(22ms)$, comparing: (**a**) ISG values with ideal state feedback; (**b**) ISG values with $\omega_n T_s = 15$ in the SOD; (**c**) SOD frequencies with ISG of $\lambda = 0.5$.

7. Conclusions

In this paper we presented a comparative study evaluating different solutions for the state estimation of a robotic airship as well as its derivatives. For the state estimation we used a second order filter, an Extended Kalman Filter and an Unscented Kalman Filter. We performed all simulations using a high-fidelity airship dynamic model for Simulink platform. For the state estimation, the EKF showed a better performance when compared to the other estimation techniques.

In a second moment, we presented a Model-based (EKF) and a Data-driven (Neural Network) approaches for the estimation of the wind velocity. Also, we proposed a novel Hybrid approach, by performing a fusion between the designed Model-based and Data-driven approaches in a cascaded structure. While the NN approach alone tends to estimate the wind speed with an offset (bias), the EKF estimation has long delays for sudden wind

changes. However, the Hybrid approach tends to yield a faster response with small bias for the estimated wind components.

Finally, we presented possible solutions to mitigate two common problems arriving in control implementation that are the control input redundancy and the delay introduced by second-order filters. The first one is solved with the use of the Wash Out Filter (WOF). The second one is mitigated with the use of the Input Scaling Gain (ISG) which is generalized here for the MIMO case. To illustrate the design solutions we applied the techniques for the design of incremental controllers, although they can be used also for other nonlinear control approaches in the same situation.

Author Contributions: A.M.: formal analysis and investigation; J.A.: investigation and supervision; A.F.: supervision; E.P.: supervision, validation and writing—review & editing; J.C.: methodology and resource; R.C.: methodology and writing—review & editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Project INCT-SAC (CNPq 465755/2014-3;FAPESP 2014/508510).

Data Availability Statement: The estimation results are available for visualization as video at youtu.be/VL5dvCyOZwY (EKF) and youtu.be/B26xaKtAyWo (UKF) accessed on 25 May 2022.

Acknowledgments: The authors gratefully acknowledge the agencies FAPESP and CNPq for the financial support. From Fapesp INCT-SAC Project (2014/50851-0), co-funded by CNPq (465755/2014-3).

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

Abbreviations

The following abbreviations are used in this manuscript:

AURORA	Autonomous Unmanned Remote Monitoring Robotic Airship
CB	Center of Buoyancy
CV	Center of Volume
DIVA	Instrumented Airship for Air Surveillance
DRONI	Robotic Airship of Innovative Design
EKF	Extended Kalman Filter
ENU	East-North-Upper
ID	Incremental Dynamics
INDI	Incremental Nonlinear Dynamic Inversion
ISG	Input Scale Gain
GPS	Global Positioning System
IMU	Inertial measurement unit
InSAC	National Institute of Science and Technology for Cooperative Autonomous Systems
LPF	Low-pass filter
NDI	Nonlinear Dynamic Inversion
NED	North-East-Down
NN	Neural Network
ROS	Robotic Operating System
SOD	Second-Order Differentiator
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
WOF	Wash Out Filter

References

1. Liu, Y.; Pan, Z.; Stirling, D.; Naghdy, F. Control of autonomous airship. In Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO), Guilin, China, 13–19 December 2009 ; Institute of Electrical & Electronics Engineers (IEEE): Piscataway, NJ, USA, 2009.
2. Liu, S.Q.; Gong, S.J.; Li, Y.X.; Lu, Z.R. Vectorial backstepping method–based trajectory tracking control for an under-actuated stratospheric airship. *Aeronaut. J.* **2017**, *121*, 916–939. [CrossRef]
3. Marton, A.S. Control Architecture for the Navigation System of Robotic Airship using Incremental Controllers. Ph.D. Thesis, Universidade de Campinas, Campinas, Brazil, 2021.

4. Recoskie, S.; Lanteigne, E.; Gueaieb, W. A High-Fidelity Energy Efficient Path Planner for Unmanned Airships. *Robotics* **2017**, *6*, 28. [CrossRef]
5. Yang, Y.; Wu, J.; Zheng, W. Positioning Control for an Autonomous Airship. *J. Aircr.* **2016**, *53*, 1638–1646. [CrossRef]
6. Cho, A.; Kim, J.; Lee, S.; Kee, C. Wind Estimation and Airspeed Calibration using a UAV with a Single-Antenna GPS Receiver and Pitot Tube. *IEEE Trans. Aerosp. Electron. Syst.* **2011**, *47*, 109–117. [CrossRef]
7. Moutinho, A.; Azinheira, J.R.; de Paiva, E.C.; Bueno, S.S. Airship robust path-tracking: A tutorial on airship modelling and gain-scheduling control design. *Control. Eng. Pract.* **2016**, *50*, 22–36. [CrossRef]
8. Perry, J.; Mohamed, A.; Johnson, B.; Lind, R. Estimating Angle of Attack and Sideslip Under High Dynamics on Small UAVs. In Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008), Savannah, Georgia, 16–19 September 2008; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2008; pp. 1165–1173.
9. Johansen, T.A.; Cristofaro, A.; Sorensen, K.; Hansen, J.M.; Fossen, T.I. On estimation of wind velocity, angle-of-attack and sideslip angle of small UAVs using standard sensors. In Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015; Institute of Electrical & Electronics Engineers (IEEE): Piscataway, NJ, USA, 2015.
10. Shen, S.; Liu, L.; Huang, B.; Lin, X.; Lan, W.; Jin, H. Wind Speed Estimation and Station-Keeping Control for Stratospheric Airships with Extended Kalman Filter. In *Proceedings of the 2015 Chinese Intelligent Automation Conference*; Springer Science Business Media: Berlin/Heidelberg, Germany, 2015; pp. 145–157.
11. Rhudy, M.B.; Gu, Y.; Gross, J.N.; Chao, H. Onboard Wind Velocity Estimation Comparison for Unmanned Aircraft Systems. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 55–66. [CrossRef]
12. Vieira, H.S.; de Paiva, E.C.; Moriguchi, S.K.; Carvalho, J.R.H. Unified Backstepping Sliding Mode Framework for Airship Control Design. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 3246–3258. [CrossRef]
13. Hassouneh, M.; Lee, H.C.; Abed, E. Washout filters in feedback control: Benefits, limitations and extensions. In Proceedings of the 2004 American Control Conference, Boston, MA, USA, 30 June–2 July 2004; IEEE: Piscataway, NJ, USA, 2004.
14. Azinheira, J.; Moutinho, A.; Carvalho, J. Lateral Control of Airship with Uncertain Dynamics using Incremental Nonlinear Dynamics Inversion. *IFAC-PapersOnLine* **2015**, *48*, 69–74. [CrossRef]
15. Rueda, M.; Mirisola, L.; Nogueira, L.; Fonseca, G.; Ramos, J.; Koyama, M.; Azinheira, J.; Carvalho, R.; Bueno, S.; de Paiva, E. Uma Infraestrutura, de Hardware, Software e Comunicação para a Robotização de Plataformas Rádio-Controladas: Aplicação a um Dirigível Robótico. In Proceedings of the 2017 SBAI-XIII Simpósio Brasileiro de Automação Inteligente, Porto Alegre, Brazil, 1–4 October 2017.
16. Meyer, J.; Sendobry, A.; Kohlbrecher, S.; Klingauf, U.; von Stryk, O. Comprehensive Simulation of Quadrotor UAVs using ROS and Gazebo. In Proceedings of the 3rd International Conference on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR), Tsukuba, Japan, 5–8 November 2012.
17. Quigley, M.; Conley, K.; Gerkey, B.P.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009.
18. Moore, T.; Stouch, D. A Generalized Extended Kalman Filter Implementation for the Robot Operating System. In *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*; Springer: Berlin/Heidelberg, Germany, 2014.
19. Bestaoui, Y. *Lighter than Air Robots*; Springer: Berlin/Heidelberg, Germany, 2011.
20. Gomes, S. An Investigation of the Flight Dynamics of Airships with Application to the YEZ-2A. Ph.D. Thesis, Cranfield Institute of Technology, Cranfield, UK, 1990.
21. Li, Y.; Nahon, M.; Sharf, I. Airship dynamics modeling: A literature review. *Prog. Aerosp. Sci.* **2011**, *47*, 217–239. [CrossRef]
22. Li, Y. Dynamics Modeling and Simulation of Flexible Airships. Ph.D. Thesis, McGill University, Montreal, QC, Canada, 2008.
23. Azouz, N.; Bestaoui, Y.; Lemaitre, O. Dynamic analysis of airships with small deformations. In Proceedings of the Third International Workshop on Robot Motion and Control (RoMoCo '02), Bukowy Dworek, Poland, 11 November 2002; pp. 209–215. [CrossRef]
24. Lane, S.H.; Stengel, R.F. Flight control design using non-linear inverse dynamics. *Automatica* **1988**, *24*, 471–483. [CrossRef]
25. Benallegue, A.; Mokhtari, A.; Fridman, L. Feedback Linearization and High Order Sliding Mode Observer For A Quadrotor UAV. In Proceedings of the International Workshop on Variable Structure Systems, 2006. (VSS06), Alghero, Italy, 5–7 June 2006; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2006.
26. Voos, H. Nonlinear control of a quadrotor micro-UAV using feedback-linearization. In Proceedings of the 2009 IEEE International Conference on Mechatronics, Malaga, Spain, 14–17 April 2009; Institute of Electrical & Electronics Engineers (IEEE): Piscataway, NJ, USA, 2009.
27. Smith, P. A simplified approach to nonlinear dynamic inversion based flight control. In Proceedings of the 23rd Atmospheric Flight Mechanics Conference, Boston, MA, USA, 10–12 August 1998; American Institute of Aeronautics and Astronautics (AIAA): Reston, VA, USA, 1998.
28. Sieberling, S.; Chu, Q.P.; Mulder, J.A. Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction. *J. Guid. Control. Dyn.* **2010**, *33*, 1732–1742. [CrossRef]

29. Acquatella, P.; van Kampen, E.; Chu, Q.P. Incremental backstepping for robust nonlinear flight control. In Proceedings of the EuroGNC 2013, 2nd CEAS Specialist Conference on Guidance, Navigation & Control, Delft University of Technology, Delft, The Netherlands, 10–12 April 2013.
30. Wang, X.; Kampen, E.J.V.; Chu, Q.; Lu, P. Incremental Sliding-Mode Fault-Tolerant Flight Control. *J. Guid. Control. Dyn.* **2019**, *42*, 244–259. [CrossRef]