

Article

Fault-Tolerant Control of a Variable-Pitch Quadrotor under Actuator Loss of Effectiveness and Wind Perturbations

Alessandro Baldini [†], Riccardo Felicetti [†], Alessandro Freddi ^{*,†} and Andrea Monteriù [†]

Department of Information Engineering, Università Politecnica delle Marche, 60131 Ancona, Italy; a.baldini@univpm.it (A.B.); r.felicetti@univpm.it (R.F.); a.monteriù@univpm.it (A.M.)

* Correspondence: a.freddi@univpm.it; Tel.: +39-071-220-4598

† These authors contributed equally to this work.

Abstract: The actuator fault-tolerant control problem for a variable-pitch quadrotor is addressed under uncertain conditions. Following a model-based approach, the plant nonlinear dynamics are faced with a disturbance observer-based control and a sequential quadratic programming control allocation, where only kinematic data of the onboard inertial measurement unit are required for the fault-tolerant control, i.e., it does not require the measurement of the motor speed nor the current drawn by the actuators. In the case of almost horizontal wind, a single observer handles both faults and the external disturbance. The estimation of the wind is fed forward by the controller, while the actuator fault estimation is exploited in the control allocation layer, which copes with the variable-pitch nonlinear dynamics, thrust saturation, and rate limits. Numerical simulations in the presence of measurement noise show the capability of the scheme to handle multiple actuator faults in a windy environment.

Keywords: variable-pitch quadrotors; fault-tolerant control; fault diagnosis; disturbance observer-based control; loss of effectiveness



Citation: Baldini, A.; Felicetti, R.; Freddi, A.; Monteriù, A. Fault-Tolerant Control of a Variable-Pitch Quadrotor under Actuator Loss of Effectiveness and Wind Perturbations. *Sensors* **2023**, *23*, 4907. <https://doi.org/10.3390/s23104907>

Academic Editor: Baoping Cai

Received: 29 March 2023

Revised: 5 May 2023

Accepted: 16 May 2023

Published: 19 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Multicopter helicopters are a class of Unmanned Aerial Vehicle (UAV) that are employed in a wide range of applications, resulting in one of the most appealing classes of vehicles for both research and industry. The standard quadrotor represents the most common configuration, which is equipped with four fixed-pitch blades connected to independent motors. Variable-Pitch Quadrotors (VPQs) can instead vary both the rotation speed of each motor and the blade pitch of each propeller, theoretically doubling the degrees of freedom of each actuator. VPQs are also known as *collective pitch quadcopters* or *heliquads*.

This results in many advantages, from which it is possible to list higher thrust rate of change, reverse thrust, reverse flight capabilities, and scaling well with size [1–3]. Moreover, the popular ArduPilot control suite supports VPQs using the *Heliquad* configuration. Despite this, VPQs still represent a market niche, where few commercial devices (e.g., the Assault Reaper 500, the Stingray 500, and the WLtoys Assassin V383, for recreational use) are available. In most of the scientific literature, however, custom prototypes (e.g., [4–6]) are developed to test algorithms and control strategies. Fixed-pitch quadrotors and VPQs have similar dynamics, while relevant differences arise in the control effectiveness matrix. This allows for identical control laws for both configurations, but VPQs require a more complex control allocation algorithm. The additional degrees of freedom, provided by the pitch angles, can be exploited to satisfy additional constraints, such as minimizing energy consumption, handling the presence of faults and failures (as defined in [7]), etc.

Specifically, the control allocation algorithm, which manages the redundancy while respecting constraints, strictly depends on two key factors: the thrust model and the choice of the control inputs.

As regards the thrust model, several mathematical models have been proposed in the literature to catch the relation between motor speed, pitch angle, as well as propeller size and shape [3,5,6,8,9].

As for the choice of the control inputs, a VPQ instead has eight inputs available in general (i.e., four motor speeds and four pitch angles), a relevant input redundancy which is perfectly suited to provide actuator fault tolerance while satisfying actuation constraints. Two issues must be faced in this case. The increased number of actuators entails higher manufacturing costs and smaller payloads due to the extra weight. Moreover, the non-linearity of the control allocation problem makes it harder to solve online. In any case, the solution in which both the pitch angle and the motor speed can independently concur in providing the requested thrusts and torques is investigated in the literature, as carried out in [9,10] and, partially, in [6,11].

Alternatively, some simplifications have been proposed in the literature. The simpler solution is to assume constant motor speeds and use pitch angles as control inputs, as proposed in [1,3,8,12,13]; in this case, four inputs are available, so the allocation problem is very simple and there is no redundancy at all. Employing a single central motor to drive the four propellers makes a decrease in costs, weight, and inertia possible. However, assuming the rotational speed is constant is just a choice to simplify the control scheme, and employing a time-varying speed, even with a single central motor, is feasible (see, for example, [12,14]). Taking into account a variable centralized motor speed allows for some input redundancy (five inputs) and thus enables for fault-tolerant control, which is the focus of this paper.

Any fault that can affect a conventional quadrotor can affect VPQs as well, and they can involve any part of the rotor system (e.g., ESC, motor, propeller). The most common class of actuator fault is a Loss of Effectiveness (LOE), i.e., the produced thrust is lower than the expected one. In fact, LOE may be the consequence of many issues: battery voltage drop [15], increased drag due to the blade pitch [5], and of course any physical damage to the blades [16]. However, as for the specific case of fault diagnosis and tolerant control for VPQs, few works can be found in the literature. In [17], the presence of a pitch lock-in-place is considered for a VPQ, and both fault diagnosis and fault-tolerant control are developed. The authors of [14] focus on the effects of a lock-in-place in a centrally powered VPQ, but no fault diagnosis or fault-tolerant control are proposed. Finally, in [18], a fault detection and fault-tolerant control strategy to face both pitch lock-in-place and motor LOE is proposed.

A major issue in control of multirotors is the presence of wind. In the UAV literature, both active and passive fault-tolerant control schemes have been employed to cope with the presence of wind. In [19], the authors model the aerodynamic effects, define the upper bounds, and then deal with the wind by means of a robust sliding mode control; in [20], extended state observers are instead employed to mitigate the effects of wind. The presence of external disturbances is also detrimental to the fault diagnosis task, as the effects of both the disturbances and faults may trigger a fault detection algorithm. In this paper, we show that horizontal wind can be managed for both the purposes of fault diagnosis and disturbance rejection by means of a single observer that estimates the actuator faults and the effects of the wind.

The main contribution of this paper is proposing a novel, complete fault-tolerant control scheme for a VPQ under actuator faults and wind, which consists of three main tasks: fault and disturbance estimation, non-linear control with disturbance rejection, and fault-tolerant control allocation. The proposed control strategy fills a gap in the literature because, to the best of the authors' knowledge, there are no fault-tolerant control schemes for VPQs that take into account common external disturbances such as the wind. With respect to the authors' previous works [17,18], in this paper, we consider the effects of wind that introduce significant complexity in the first two tasks, i.e., fault estimation and control. As an additional contribution, we propose a generalized strategy that is well-suited for both centrally powered VPQs and VPQs with four independent motors. In fact, the fault and disturbance estimation strategy, as well as the non-linear control law, does not depend

on the choice of the control inputs (i.e., centralized or independent propeller speed). On the other hand, the control allocation algorithm must exploit redundancy to achieve fault tolerance, so the choice of the control inputs is relevant; however, the proposed control allocation algorithm is designed such that it suffices to enable an equality constraint to deal with a centrally powered VPQ (instead of a conventional one with four separate motors).

The paper is structured as follows. In Section 2, the mathematical model of the vehicle is detailed, including the characterization of variable-pitch propellers and the model of actuator faults to be faced. Section 3 is devoted to the design of a Non-linear Disturbance Observer (NDO) to estimate faults and disturbances. In Section 4, a double-loop controller to track position and orientation, despite the presence of external disturbances, is introduced. In Section 5, a control allocation strategy that redistributes the control effort by taking into account the presence of actuator faults is detailed. Simulation results to show the effectiveness of the overall fault-tolerant control strategy are proposed in Section 6. Finally, conclusions and future works end the paper.

2. Mathematical Model

The overall kinematics and dynamics of a generic multirotor, thus including variable-pitch quadrotors, can be approximated as that of a unique rigid body with six degrees of freedom [21]. Let us consider an earth-fixed frame $R_E = (O_E, x_E, y_E, z_E)$, which is assumed to be inertial, and a body-fixed frame $R_B = (O_B, x_B, y_B, z_B)$, which is placed in the center of mass of the vehicle. For simplicity, we assume the center of mass coincides with the geometric center of the quadrotor, as depicted in Figure 1.

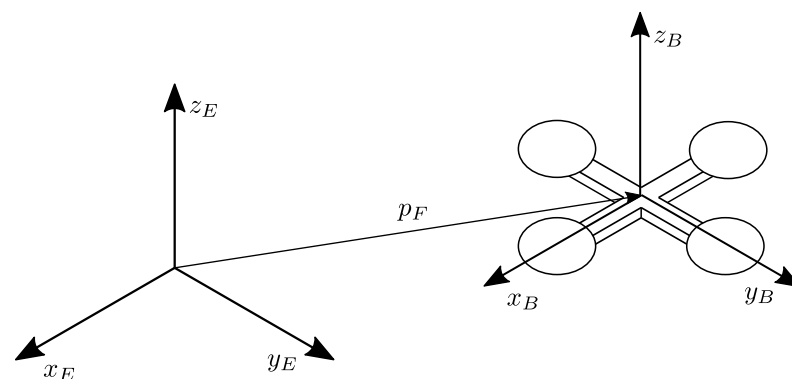


Figure 1. Quadrotor configuration and frames.

From now on, the standard basis vectors of \mathbb{R}^3 are denoted with e_1, e_2, e_3 . Let us denote with $p_F = \text{col}(x_F, y_F, z_F)$ the position of the center of mass with respect to R_E , and with $\omega = \text{col}(p, q, r)$ the angular velocity of the vehicle. Neglecting the second-order effects (i.e., rotor dynamics, blade flapping, gyroscopic and inertial effects due to the rotors), the quadrotor kinematics and dynamics can be expressed as [21]

$$\begin{aligned} m\ddot{p}_F &= -k_t\dot{p}_F - mge_3 + RF_m^B + F_w^E \\ J\dot{\omega} &= -k_r\omega - \omega \times J\omega + \tau_m^B \\ \dot{\eta} &= T(\eta)\omega, \end{aligned} \quad (1)$$

where m is the total system mass, $J = \text{diag}(J_x, J_y, J_z)$ is the tensor of inertia along the x_B, y_B , and z_B axes, k_t and k_r are the linear and angular friction coefficients, $\eta = \text{col}(\varphi, \theta, \psi)$ is the vector of the attitude angles (roll, pitch, and yaw, respectively) that characterize the rotation from R_B to R_E .

$$R = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\theta - c_\varphi s_\psi & s_\varphi s_\psi + c_\varphi c_\psi s_\theta \\ c_\theta s_\psi & c_\varphi c_\psi + s_\varphi s_\psi s_\theta & c_\varphi s_\psi s_\theta - c_\psi s_\varphi \\ -s_\theta & c_\theta s_\varphi & c_\varphi c_\theta \end{bmatrix}, \quad (2)$$

where $\cos(\cdot) = c(\cdot)$ and $\sin(\cdot) = s(\cdot)$ are introduced for the sake of brevity, F_m^B and τ_m^B are the overall force and torque due to the motors (decomposed in R_B), F_w^E is the force due to the wind (decomposed in R_E), and $T(\eta)$ is the kinematic coordinate transformation related to the adopted roll–pitch–yaw rotation, that is [22]:

$$T(\eta) = \begin{bmatrix} 1 & \sin(\varphi) \tan(\theta) & \cos(\varphi) \tan(\theta) \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \frac{\sin(\varphi)}{\cos(\theta)} & \frac{\cos(\varphi)}{\cos(\theta)} \end{bmatrix}. \quad (3)$$

Let us denote with $f_i \in \mathbb{R}$ the actual lift thrust produced by the i -th actuator. In VPQs, as in conventional quadrotors, all the actuators are oriented upwards and they produce collinear lift forces only, as depicted in Figure 2.

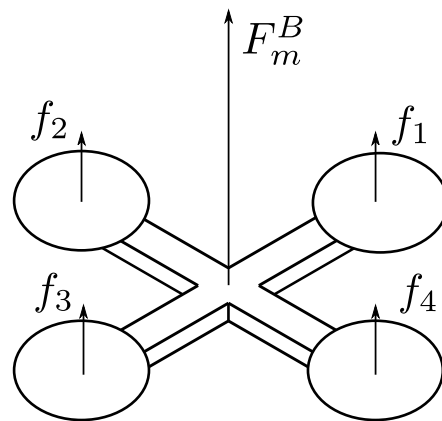


Figure 2. Propeller lift thrusts of the variable-pitch quadrotor.

Denoting with u_z the overall lift force, given by $u_z = f_1 + f_2 + f_3 + f_4$, we have

$$\begin{aligned} F_m^B &= \text{col}(0, 0, u_z) \\ \tau_m^B &= \text{col}(\tau_p, \tau_q, \tau_r). \end{aligned} \quad (4)$$

In other words, the overall lift thrust is along z_B only, because none of the actuators can generate a horizontal force in R_B . Then, substituting (4) in the first row of (1), we obtain the following model:

$$\begin{aligned} m\ddot{p}_F &= -k_i \dot{p}_F - mge_3 + u_z Re_3 + F_w^E \\ J\dot{\omega} &= -k_r \omega - \omega \times J\omega + \tau_m^B \\ \dot{\eta} &= T(\eta)\omega. \end{aligned} \quad (5)$$

The four-dimensional vector $\text{col}(u_z, \tau_m^B) = \text{col}(u_z, \tau_p, \tau_q, \tau_r)$ is the so called virtual input vector.

2.1. Input Mapping

Consider the i -th motor. We denote with $f_i \in \mathbb{R}$ the lift thrust generated by the propeller attached to the motor, with $d_i \in \mathbb{R}$ the generated drag, and with $\Omega_i \in \mathbb{R}$ the angular rate of the propeller, taken to be positive when the rotation of the motor matches that of the propeller (clockwise or counter-clockwise). Let us consider the case of rectangular, untwisted blades (which are suitable for variable-pitch drones) with uniform inflow. Let us assume the blade pitch is limited in order to avoid propeller stalling, which is an undesired

effect to be avoided in practice. Then, both the lift force and drag of each rotor can be expressed as a function of the rotational speed [3]:

$$f_i = c_{L_i} \Omega_i^2 \tag{6}$$

$$d_i = c_{D_i} \Omega_i^2, \tag{7}$$

where c_{L_i} and c_{D_i} are the lift and drag coefficients that depend on the blade geometry. In the case of fixed-pitch propellers, c_{L_i} and c_{D_i} are constant. In a variable-pitch propeller, however, as the pitch angle $\alpha_i \in \mathbb{R}$ changes, the lift and drag coefficients change as well. Two main approaches to characterize the dependency of c_{L_i} and c_{D_i} on α_i can be found in [9], and they can be categorized into grey-box (e.g., [3]) and black-box (e.g., [5]) approaches. In this work, we follow the grey-box approach of [3], which makes use of the blade element theory and the momentum theory to obtain

$$c_{L_i} = c_{T_i}(\alpha_i) \rho \pi R_a^4 \tag{8}$$

$$c_{D_i} = c_{Q_i}(\alpha_i) \rho \pi R_a^5, \tag{9}$$

where R_a is the propeller radius, ρ is the air density, while $c_{T_i}(\cdot)$ and $c_{Q_i}(\cdot)$ are dimensionless lift and drag coefficients, which are functions of α_i . Following [9,23], which assume steady-state linearized blade aerodynamics, we consider the implicit relations

$$\alpha_i = k_1 c_{T_i} + k_2 \sqrt{|c_{T_i}|} \tag{10}$$

$$c_{Q_i} = k_3 \sqrt{|c_{T_i}|^3} + k_4, \tag{11}$$

where $k_1 = \frac{6}{\sigma C_{l_{\alpha_i}}}$, $k_2 = \frac{3}{2\sqrt{2}}$, $k_3 = \frac{1}{\sqrt{2}}$, and $k_4 = \frac{1}{8} \sigma C_{d0_i}$. In detail, $\sigma = \frac{N_b c}{\pi R_a}$ is the blade solidity, $N_b = 2$ is the number of blades, c is the chord length, $C_{l_{\alpha_i}}$ is the 2D lift-curve-slope of the airfoil section, comprising the rotor, and finally, C_{d0_i} is the zero-lift drag coefficient (see [9] for further details). Without loss of generality, we consider a '+' quadrotor configuration (see Figure 1), so the overall input mapping is

$$\begin{bmatrix} u_z \\ \tau_p \\ \tau_q \\ \tau_r \end{bmatrix} = \begin{bmatrix} c_{L_1}(\alpha_1) \Omega_1^2 + c_{L_2}(\alpha_2) \Omega_2^2 + c_{L_3}(\alpha_3) \Omega_3^2 + c_{L_4}(\alpha_4) \Omega_4^2 \\ -l c_{L_2}(\alpha_2) \Omega_2^2 + l c_{L_4}(\alpha_4) \Omega_4^2 \\ -l c_{L_1}(\alpha_1) \Omega_1^2 + l c_{L_3}(\alpha_3) \Omega_3^2 \\ -c_{D_1}(\alpha_1) \Omega_1^2 + c_{D_2}(\alpha_2) \Omega_2^2 - c_{D_3}(\alpha_3) \Omega_3^2 + c_{D_4}(\alpha_4) \Omega_4^2 \end{bmatrix}, \tag{12}$$

where l is the arm length. The mapping (12) is nonlinear in α_i and Ω_i , which are the actual control inputs for each motor i , for a total of eight inputs.

2.2. Explicit Lift and Drag Functions

Equations (10) and (11) are implicit functions of α_i , so they are not practical for online computation. In the literature, such functions are solved iteratively by means of numerical approaches [9]. In this section, instead, we introduce two equivalent equations that determine the coefficients c_{T_i} and c_{Q_i} in terms of explicit functions of α_i .

The lift function $c_{T_i}(\alpha_i)$ in (10) is expected to be an odd function of α_i [9]. In fact, from (8), we obtain

$$c_{T_i}(\alpha_i) = \frac{1}{k_1} \alpha_i + \frac{1}{2} \text{sign}(\alpha_i) \left(\frac{k_2^2}{k_1^2} - \frac{k_2}{k_1^2} \sqrt{k_2^2 + 4k_1 |\alpha_i|} \right), \tag{13}$$

which is an explicit odd function of α_i . The drag function $c_{Q_i}(\alpha_i)$ in (11) is expected to be an even function of α_i [9]. Then, noting that $c_{T_i}(|\alpha_i|) = \text{sign}(\alpha_i) c_{T_i}(\alpha_i)$, (11) can be rewritten as

$$c_{Q_i}(\alpha_i) = k_3 \sqrt{\text{sign}(\alpha_i) c_{T_i}(\alpha_i)^3} + k_4, \tag{14}$$

which is an explicit even function of α_i . We also calculate in advance the derivatives of (13) and (14) with respect to α_i , as they are going to be employed in the remainder of the paper. We have

$$\begin{aligned} c'_{T_i}(\alpha_i) &= \frac{1}{k_1} - \frac{k_2}{k_1} \frac{1}{\sqrt{k_2^2 + 4k_1|\alpha_i|}} \\ c'_{Q_i}(\alpha_i) &= \frac{3}{2} k_3 c'_{T_i}(\alpha_i) \text{sign}(\alpha_i) \sqrt{\text{sign}(\alpha_i) c_{T_i}(\alpha_i)} \\ c'_{L_i}(\alpha_i) &= c'_{T_i}(\alpha_i) \rho \pi R_a^4 \\ c'_{D_i}(\alpha_i) &= c'_{Q_i}(\alpha_i) \rho \pi R_a^5, \end{aligned} \quad (15)$$

where the second row is obtained with the use of

$$\frac{dc_{T_i}(|\alpha_i|)}{d\alpha_i} = \text{sign}(\alpha_i) \frac{dc_{T_i}(\alpha_i)}{d\alpha_i}. \quad (16)$$

2.3. Actuator Faults and Control Input Definition

A total of eight control inputs are considered, which are the pitch angles α_i and the squares of the desired rotor speeds, denoted as u_i , for $i = 1, \dots, 4$. All the control inputs are collapsed in the vectors

$$\begin{aligned} \alpha &= \text{col}(\alpha_1, \alpha_2, \alpha_3, \alpha_4), \\ u &= \text{col}(u_1, u_2, u_3, u_4). \end{aligned} \quad (17)$$

In the absence of faults, the relation $\Omega_i^2 = u_i$ holds. For each actuator i , a possible LOE is considered, so that the resulting angular speed may be lower than the desired one, i.e.,

$$\Omega_i^2 = w_i u_i. \quad (18)$$

Each w_i , for $i = 1, \dots, 4$, is an unknown and time-varying quantity which models the fault magnitude on the i -th actuator. It is constrained in $w_i \in [0, 1]$, where $w_i = 1$ corresponds to the faultless scenario. For the sake of brevity, the fault magnitudes are collected in the vector

$$w = \text{col}(w_1, w_2, w_3, w_4). \quad (19)$$

3. Wind and Fault Estimation

We assume the wind acts on the $x_E - y_E$ plane only, that is, $F_w^E = f_{wx}e_1 + f_{wy}e_2$. In other words, we neglect the updraft, and we claim that the horizontal wind does not generate any significant torque due to the symmetry of the vehicle. Thus, six target independent variables need to be estimated; these are f_{wx} , f_{wy} and w_1, w_2, w_3, w_4 , and they are summarized in the extended uncertainty vector $\bar{w} = \text{col}(f_{wx}, f_{wy}, w_1, w_2, w_3, w_4)$. By handling the joint vector \bar{w} as an external disturbance, a disturbance observer for the estimation \hat{w} of \bar{w} is then designed, which in our purpose can be defined as a dynamical system in the form

$$\dot{\hat{s}} = F(\hat{s}, \bar{x}, \bar{u}) \quad (20)$$

$$\hat{w} = H(\hat{s}, \bar{x}, \bar{u}), \quad (21)$$

for which $\|\hat{w} - \bar{w}\| \rightarrow 0$ as $t \rightarrow \infty$, and where \hat{s} is the observer state, $\bar{x} = \text{col}(p_F, \dot{p}_F, \omega, \eta)$ and $\bar{u} = \text{col}(u, \alpha)$ compactly refers to the quadrotor state and the control inputs, and where $F(\cdot)$ and $H(\cdot)$ are smooth function to design. Several disturbance observers have been proposed in the literature. Here, we follow the NDO proposed in [24], which can be summarized as follows.

1. Define an auxiliary variable s which directly depends on the estimation target \bar{w} . Usually, the auxiliary variable $s = \bar{w} - \lambda(\bar{x}, \bar{u})$ is chosen, for some smooth function $\lambda(\cdot)$.

2. Find the dynamics \dot{s} of the auxiliary variable s . In this step, the disturbance observer design problem is turned into a state observer design problem.
3. Based on the auxiliary dynamics \dot{s} , design a state observer for the estimation \hat{s} of s .
4. Define \hat{w} by using the inverse function of the chosen auxiliary variable definition.

For design purposes, the model is conveniently rewritten in Section 3.1. Then, the design of the NDO is proposed in Section 3.2.

Remark 1. The estimation of the actuator faults and the external force generated by the wind is obtained by the disturbance observer, where only kinematic data are required, i.e., without the need of measuring motor speeds, motor currents, etc.

Remark 2. The design of $\lambda(\cdot)$ involves the solution of a set of partial differential equations.

Remark 3. If the dynamics of \bar{w} is uncertain, the ultimate boundedness of $\|\hat{w} - \bar{w}\|$ is demanded instead of convergence.

3.1. Model Manipulation

Note that the input mapping (12) and the action of the fault (18) can be rewritten as

$$\begin{bmatrix} u_z \\ \tau_p \\ \tau_q \\ \tau_r \end{bmatrix} = \begin{bmatrix} c_{L_1}(\alpha_1) & c_{L_2}(\alpha_2) & c_{L_3}(\alpha_3) & c_{L_4}(\alpha_4) \\ 0 & -lc_{L_2}(\alpha_2) & 0 & lc_{L_4}(\alpha_4) \\ -lc_{L_1}(\alpha_1) & 0 & lc_{L_3}(\alpha_3) & 0 \\ -c_{D_1}(\alpha_1) & c_{D_2}(\alpha_2) & -c_{D_3}(\alpha_3) & c_{D_4}(\alpha_4) \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \triangleq B_c(\alpha) \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (22)$$

and

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} w_1 & 0 & 0 & 0 \\ 0 & w_2 & 0 & 0 \\ 0 & 0 & w_3 & 0 \\ 0 & 0 & 0 & w_4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \text{diag}(w)u = \text{diag}(u)w, \quad (23)$$

resulting in the overall relation $\text{col}(u_z, \tau_m^B) = B_c(\alpha)\text{diag}(u)w$.

The quadrotor kinematics and dynamics in (5) can be equivalently expressed as

$$\frac{d}{dt} \begin{bmatrix} p_F \\ \eta \end{bmatrix} = \begin{bmatrix} \dot{p}_F \\ T(\eta)\omega \end{bmatrix} \quad (24)$$

$$\frac{d}{dt} \begin{bmatrix} \dot{p}_F \\ \omega \end{bmatrix} = \begin{bmatrix} -\frac{k_t}{m}\dot{p}_F - ge_3 \\ -k_r J^{-1}\omega - J^{-1}(\omega \times J\omega) \end{bmatrix} + \frac{1}{m} \begin{bmatrix} e_1 & e_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f_{wx} \\ f_{wy} \end{bmatrix} + \begin{bmatrix} \frac{1}{m}Re_3 & 0 \\ 0 & J^{-1} \end{bmatrix} \begin{bmatrix} u_z \\ \tau_m^B \end{bmatrix}. \quad (25)$$

Then, defining $z_1 = \text{col}(p_F, \eta)$ and $z_2 = \text{col}(\dot{p}_F, \omega)$, we can rewrite the overall quadrotor model in order to show that the non-linear model is affine with respect to the uncertainty \bar{w} . In fact, we have

$$\begin{aligned} \dot{z}_1 &= h_1(z_1, z_2) \\ \dot{z}_2 &= h_2(z_2) + P(z_1, u, \alpha)\bar{w}, \end{aligned} \quad (26)$$

where

$$h_1(z_1, z_2) = \begin{bmatrix} \dot{p}_F \\ T(\eta)\omega \end{bmatrix}, \quad h_2(z_2) = \begin{bmatrix} -\frac{k_t}{m}\dot{p}_F - ge_3 \\ -k_r J^{-1}\omega - J^{-1}(\omega \times J\omega) \end{bmatrix}, \quad (27)$$

and $P(z_1, u, \alpha) = [P_1 \quad P_2(z_1, u, \alpha)] \in \mathbb{R}^{6 \times 6}$, with

$$P_1 = \frac{1}{m} \begin{bmatrix} e_1 & e_2 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{6 \times 2}, \quad P_2(z_1, u, \alpha) = \begin{bmatrix} \frac{1}{m}Re_3 & 0 \\ 0 & J^{-1} \end{bmatrix} B_c(\alpha)\text{diag}(u) \in \mathbb{R}^{6 \times 4}. \quad (28)$$

Finally, P is almost everywhere nonsingular, since

$$\begin{aligned}\det(P) &= \frac{1}{m^2} \det\left(\begin{bmatrix} \frac{1}{m} \cos(\varphi) \cos(\theta) & 0 \\ 0 & J^{-1} \end{bmatrix} B_c(\alpha) \text{diag}(u)\right) \\ &= u_1 u_2 u_3 u_4 \frac{\cos(\varphi) \cos(\theta)}{m^3 J_x J_y J_z} \det(B_c(\alpha)),\end{aligned}\quad (29)$$

where $\det(B_c(\alpha)) \neq 0$ almost everywhere (assuming $l > 0$). Noting that each u_i is positive in real flight conditions, $\cos(\varphi) \cos(\theta)$ is different from zero in non-aerobatic flight, and $\det(B_c(\alpha)) \neq 0$, then the inverse of P is well-defined in practical flight conditions.

3.2. Disturbance Observer Design

The NDO is designed on model (26). Indeed, the NDO design in DOBC is well-known for control affine systems [25], so the idea is actually based on the affine uncertainty, which is \bar{w} in our problem. Consider the auxiliary variable $s = \bar{w} - \lambda(z_1, z_2, u, \alpha)$, where $\lambda(\cdot)$ is a function to design. For the sake of brevity, we denote $\lambda = \lambda(z_1, z_2, u, \alpha)$. By differentiation, we have

$$\begin{aligned}\dot{s} &= \dot{\bar{w}} - \frac{\partial \lambda}{\partial z_1} h_1(z_1, z_2) - \frac{\partial \lambda}{\partial z_2} (h_2(z_2) + P(z_1, u, \alpha) \bar{w}) - \frac{\partial \lambda}{\partial u} \dot{u} - \frac{\partial \lambda}{\partial \alpha} \dot{\alpha} \\ &= \dot{\bar{w}} - \frac{\partial \lambda}{\partial z_1} h_1(z_1, z_2) - \frac{\partial \lambda}{\partial z_2} (h_2(z_2) + P(z_1, u, \alpha) (s + \lambda)) - \frac{\partial \lambda}{\partial u} \dot{u} - \frac{\partial \lambda}{\partial \alpha} \dot{\alpha} \\ &= -\frac{\partial \lambda}{\partial z_2} P(z_1, u, \alpha) s + R_s + \dot{\bar{w}},\end{aligned}\quad (30)$$

where R_s is a known term given by

$$R_s = -\frac{\partial \lambda}{\partial z_1} h_1(z_1, z_2) - \frac{\partial \lambda}{\partial z_2} (h_2(z_2) + P(z_1, u, \alpha) \lambda) - \frac{\partial \lambda}{\partial u} \dot{u} - \frac{\partial \lambda}{\partial \alpha} \dot{\alpha}.\quad (31)$$

The disturbance observer is then designed as

$$\dot{\hat{s}} = -\frac{\partial \lambda}{\partial z_2} P(z_1, u, \alpha) \hat{s} + R_s\quad (32)$$

$$\hat{\bar{w}} = \hat{s} + \lambda,\quad (33)$$

where \hat{s} and $\hat{\bar{w}}$ are estimations for s and \bar{w} , respectively. The function $\lambda(\cdot)$ is investigated through the estimation error dynamics. On this purpose, let us consider the error variables $\tilde{s} = s - \hat{s}$ and $\tilde{\bar{w}} = \bar{w} - \hat{\bar{w}}$. The error dynamics is

$$\dot{\tilde{s}} = -\frac{\partial \lambda}{\partial z_2} P(z_1, u, \alpha) \tilde{s} + \dot{\tilde{\bar{w}}}\quad (34)$$

$$\dot{\tilde{\bar{w}}} = \tilde{s}.\quad (35)$$

With the choice $\lambda(z_1, z_2, u, \alpha) = HP(z_1, u, \alpha)^{-1} z_2$, where $-H$ is Hurwitz, the overall estimation error boils down to

$$\dot{\tilde{\bar{w}}} = -H \tilde{\bar{w}} + \dot{\tilde{\bar{w}}},\quad (36)$$

which is an asymptotically stable linear time-invariant system, perturbed by the unknown input $\dot{\tilde{\bar{w}}}$.

4. Control Law Design

A classical inner/outer loop approach is employed to calculate the desired virtual inputs [26,27], where the small angle approximation is adopted ($\varphi \approx 0$ and $\theta \approx 0$). Moreover, in order to avoid a conflict between the inner and the outer loop, a spectral separation is required; to achieve this, in practice, a higher rate is employed for the inner loop with respect to the outer loop [28].

4.1. Outer Loop Design

Given the time-varying position references x_{Fr} and y_{Fr} (for x_F and y_F , respectively), the outer loop is designed to calculate the references φ_r and θ_r (for the angles φ and θ , respectively) in order to track the position reference. Exploiting the small angle approximation [26,27], that is, $\varphi \approx 0$ and $\theta \approx 0$, we have

$$\begin{bmatrix} e_1^T \\ e_2^T \end{bmatrix} Re_3 \approx \begin{bmatrix} \sin(\psi) & \cos(\psi) \\ -\cos(\psi) & \sin(\psi) \end{bmatrix} \begin{bmatrix} \varphi \\ \theta \end{bmatrix} \triangleq R_2(\psi) \begin{bmatrix} \varphi \\ \theta \end{bmatrix}. \quad (37)$$

Substituting (37) in (5), it follows that

$$\begin{aligned} \begin{bmatrix} \ddot{x}_F \\ \ddot{y}_F \end{bmatrix} &= \begin{bmatrix} e_1^T \\ e_2^T \end{bmatrix} \ddot{p}_F = -\frac{k_t}{m} \begin{bmatrix} \dot{x}_F \\ \dot{y}_F \end{bmatrix} + \frac{u_z}{m} \begin{bmatrix} e_1^T \\ e_2^T \end{bmatrix} Re_3 + \frac{1}{m} \begin{bmatrix} f_{wx} \\ f_{wy} \end{bmatrix} \\ &\approx -\frac{k_t}{m} \begin{bmatrix} \dot{x}_F \\ \dot{y}_F \end{bmatrix} + \frac{u_z}{m} R_2(\psi) \begin{bmatrix} \varphi \\ \theta \end{bmatrix} + \frac{1}{m} \begin{bmatrix} f_{wx} \\ f_{wy} \end{bmatrix}. \end{aligned} \quad (38)$$

As the inner loop runs at a higher rate with respect to the outer loop, we can assume the inner loop is fast enough to track φ_r and θ_r [26]. Hence, by solving (38) for the variables φ and θ , we obtain the ideal references φ_r and θ_r for $u_z \neq 0$, that is,

$$\begin{bmatrix} \varphi_r \\ \theta_r \end{bmatrix} = \frac{m}{u_z} R_2^T(\varphi) \left(\frac{k_t}{m} \begin{bmatrix} \dot{x}_F \\ \dot{y}_F \end{bmatrix} - \frac{1}{m} \begin{bmatrix} \hat{f}_{wx} \\ \hat{f}_{wy} \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \end{bmatrix} \right), \quad (39)$$

where v_x and v_y are injection inputs, defined as

$$v_x = \dot{x}_{Fr} - \alpha_{x1}(\dot{x}_F - \dot{x}_{Fr}) - \alpha_{x0}(x_F - x_{Fr}) \quad (40)$$

$$v_y = \dot{y}_{Fr} - \alpha_{y1}(\dot{y}_F - \dot{y}_{Fr}) - \alpha_{y0}(y_F - y_{Fr}). \quad (41)$$

The parameters $\alpha_{x1}, \alpha_{x0}, \alpha_{y1}, \alpha_{y0}$ are chosen according to the desired pole placement problem. In fact, if the angles φ and θ follow references (39)–(41), we can show by substitution in (5) that the outer loop tracking error dynamics becomes:

$$\ddot{e}_x + \alpha_{x1}\dot{e}_x + \alpha_{x0}e_x = \frac{1}{m}\tilde{f}_{wx} \quad (42)$$

$$\ddot{e}_y + \alpha_{y1}\dot{e}_y + \alpha_{y0}e_y = \frac{1}{m}\tilde{f}_{wy}, \quad (43)$$

where $e_x = x_F - x_{Fr}$, $e_y = y_F - y_{Fr}$, $\tilde{f}_{wx} = f_{wx} - \hat{f}_{wx}$ and $\tilde{f}_{wy} = f_{wy} - \hat{f}_{wy}$.

4.2. Inner Loop Design

Let z_{Fr} and ψ_r be the reference signals for z_F and ψ that we want to track, and let φ_r and θ_r be the reference signals for φ and θ (provided by the outer loop in (39)). From (5), it follows that

$$\ddot{z}_F = -\frac{k_t}{m}\dot{z}_F - g + \frac{u_z}{m} \cos(\varphi) \cos(\theta) \quad (44)$$

$$\dot{\eta} = \dot{T}(\eta)\omega - k_r T(\eta)J^{-1}\omega - T(\eta)J^{-1}(\omega \times J\omega) + T(\eta)J^{-1}\tau_m^B, \quad (45)$$

so we can set the reference values $u_{z,r}$ and $\tau_{m,r}^B$ (for u_z and τ_m^B , respectively) as

$$u_{z,r} = \frac{m}{\cos(\varphi) \cos(\theta)} \left(\frac{k_t}{m}\dot{z}_F + g + v_z \right) \quad (46)$$

$$\tau_{m,r}^B = JT(\eta)^{-1} \left(-\dot{T}(\eta)\omega + k_r T(\eta)J^{-1}\omega + T(\eta)J^{-1}(\omega \times J\omega) + v_\eta \right).$$

The terms $v_z \in \mathbb{R}$ and $v_\eta \in \mathbb{R}^3$ are auxiliary inputs, and they are set as

$$v_z = \ddot{z}_{Fr} - \alpha_{z1}(\dot{z}_F - \dot{z}_{Fr}) - \alpha_{z0}(z_F - z_{Fr}) \quad (47)$$

$$v_\eta = \ddot{\eta}_r - \alpha_{\eta1}(T(\eta)\omega - \dot{\eta}_r) - \alpha_{\eta0}(\eta - \eta_r), \quad (48)$$

where the scalars α_{z1}, α_{z0} and the matrices $\alpha_{\eta1}, \alpha_{\eta0}$ are set according to the desired pole placement. If the variables u_z and τ_m^B follow references (46)–(48), we can show by substitution in (5) that the inner loop tracking error dynamics becomes

$$\ddot{e}_z + \alpha_{z1}\dot{e}_z + \alpha_{z0}e_z = 0 \quad (49)$$

$$\ddot{e}_\eta + \alpha_{\eta1}\dot{e}_\eta + \alpha_{\eta0}e_\eta = 0, \quad (50)$$

where $e_z = z_F - z_{Fr}$ and $e_\eta = \eta - \eta_r$.

5. Control Allocation

In this section, we show how the virtual input (i.e., u_z and τ_m^B) can be distributed among the motors (i.e., u) and the pitch servos (i.e., α). Let us consider a non-linear optimization problem in the optimization variables α and u :

$$\min_{\alpha, u} J(\alpha, u) \quad (51)$$

$$s.t. \quad h(\alpha, u) = 0 \quad (52)$$

$$g(\alpha, u) \leq 0, \quad (53)$$

where $J(\alpha, u)$ in (51) is a cost function to be determined, while (52) and (53) represent generic non-linear and nonconvex constraints. In the literature, the control allocation problem in multirotors is usually tackled with lightweight methods, such as pseudo-inverse and redistributed pseudo-inverse [29,30]. Such methods are motivated by the need for high rates, but they are not effective in the presence of constraints and rate limits. Moreover, they need a significant simplification of the problem formulation (i.e., linear or quadratic cost functions, linear equality constraints, potentially also removing inequality constraints). In the following, we set up the control allocation problem and we detail a method to solve it efficiently by means of a locally convex Quadratic Programming (QP) reformulation (as carried out, for example, in [31] for a marine vessel).

The method that is detailed in this section to solve the non-linear allocation problem (51)–(53) represents a simplified version of the generic Sequential Quadratic Programming (SQP) strategy (see, for example, [32]) to solve non-linear optimization problems. The main differences are

- SQP employs the Hessian of the Lagrangian in the cost function, while we employ the Hessian of the original cost function. In other words, we always set the Lagrange multipliers' initial guess to zero for simplicity, so that the Hessian of the Lagrangian coincides with the Hessian of the original cost function.
- Once the search direction is found, a full step is performed in the proposed method, and no merit function is computed.
- The stopping criterion in the proposed method boils down to performing just a single iteration (i.e., a single QP is solved as in Real-Time Iteration [33], instead of a sequence of QP problems as done in SQP), and so no convergence criteria are needed.

These modifications introduce some loss of accuracy in the solution, but they reduce the number of evaluations, in order to perform the allocation faster in view of online implementation.

5.1. Equality Constraints

The equality constraints (52) originate from (22)–(23), namely

$$h(\alpha, u) = B \begin{bmatrix} \text{diag}(w) \text{diag}(\bar{c}_L(\alpha)) u \\ \text{diag}(w) \text{diag}(\bar{c}_D(\alpha)) u \end{bmatrix} - \begin{bmatrix} u_z \\ \tau_m^B \end{bmatrix} \quad (54)$$

$$= B \begin{bmatrix} \text{diag}(w) \text{diag}(u) \bar{c}_L(\alpha) \\ \text{diag}(w) \text{diag}(u) \bar{c}_D(\alpha) \end{bmatrix} - \begin{bmatrix} u_z \\ \tau_m^B \end{bmatrix} = 0, \quad (55)$$

where

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -l & 0 & l & 0 & 0 & 0 & 0 \\ -l & 0 & l & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & -1 & 1 \end{bmatrix} \quad (56)$$

and

$$\begin{aligned} \bar{c}_L(\alpha) &= \text{col}(c_L(\alpha_1), c_L(\alpha_2), c_L(\alpha_3), c_L(\alpha_4)) \\ \bar{c}_D(\alpha) &= \text{col}(c_D(\alpha_1), c_D(\alpha_2), c_D(\alpha_3), c_D(\alpha_4)) \\ \bar{c}'_L(\alpha) &= \text{col}(c'_L(\alpha_1), c'_L(\alpha_2), c'_L(\alpha_3), c'_L(\alpha_4)) \\ \bar{c}'_D(\alpha) &= \text{col}(c'_D(\alpha_1), c'_D(\alpha_2), c'_D(\alpha_3), c'_D(\alpha_4)). \end{aligned} \quad (57)$$

Please note that $c_L(\alpha_i)$ and $c_D(\alpha_i)$ have been defined in (8), while $c'_L(\alpha_i)$ and $c'_D(\alpha_i)$ have been defined in (15). The time-varying numeric values for u_z , τ_m^B , and w in (54) are given by (46) ($u_{z,r}$, $\tau_{m,r}^B$) and (33) (\hat{w}), respectively; in other words, the actual lift force and torques must satisfy those required by the inner loop, also taking into account the actual fault estimation.

Please note that the equality constraints are non-linear and nonconvex with respect to the optimization variables α and u . Non-linear optimization problems are hard to solve online due to computational limits, as well as because the existence of the solution is not guaranteed in general. To solve efficiently the problem online, the equality constraints (52) are linearized, as in SQP, so that a QP problem can be obtained. Using the first-order Taylor approximation around (α_0, u_0) , we have

$$h(\alpha, u) \approx h(\alpha_0, u_0) + \frac{\partial h(\alpha_0, u_0)}{\partial \alpha} (\alpha - \alpha_0) + \frac{\partial h(\alpha_0, u_0)}{\partial u} (u - u_0), \quad (58)$$

where α_0 and u_0 represent the current value of α and u , respectively. Then, (54) can be approximated by

$$\underbrace{\begin{bmatrix} \frac{\partial h(\alpha_0, u_0)}{\partial \alpha} & \frac{\partial h(\alpha_0, u_0)}{\partial u} \end{bmatrix}}_{A_{eq}} \underbrace{\begin{bmatrix} \Delta \alpha \\ \Delta u \end{bmatrix}}_x = \underbrace{-h(\alpha_0, u_0)}_{b_{eq}}, \quad (59)$$

where $\Delta \alpha = \alpha - \alpha_0$, $\Delta u = u - u_0$, and

$$\begin{aligned} \frac{\partial h(\alpha_0, u_0)}{\partial \alpha} &= B \begin{bmatrix} \text{diag}(w) \text{diag}(u_0) \text{diag}(c'_L) \\ \text{diag}(w) \text{diag}(u_0) \text{diag}(c'_D) \end{bmatrix} \\ \frac{\partial h(\alpha_0, u_0)}{\partial u} &= B \begin{bmatrix} \text{diag}(w) \text{diag}(\bar{c}_L(\alpha_0)) \\ \text{diag}(w) \text{diag}(\bar{c}_D(\alpha_0)) \end{bmatrix} \\ h(\alpha_0, u_0) &= B \begin{bmatrix} \text{diag}(w) \text{diag}(\bar{c}_L(\alpha_0)) u_0 \\ \text{diag}(w) \text{diag}(\bar{c}_D(\alpha_0)) u_0 \end{bmatrix} - \begin{bmatrix} u_z \\ \tau_m^B \end{bmatrix}. \end{aligned} \quad (60)$$

As w is not available, to build the control allocation problem online we replace it in (60) with its estimation contained in \hat{w} .

Remark 4. In the case of a centralized motor, it suffices to add the constraint $\Delta u_1 = \Delta u_2 = \Delta u_3 = \Delta u_4$; provided that the same initial value is employed for u , then $u_1 = u_2 = u_3 = u_4$ holds. In other words, (59) is replaced by

$$\underbrace{\begin{bmatrix} \frac{\partial h(\alpha_0, u_0)}{\partial \alpha} & & \frac{\partial h(\alpha_0, u_0)}{\partial u} \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}}_{A_{eq}} \underbrace{\begin{bmatrix} \Delta \alpha \\ \Delta u \end{bmatrix}}_x = \underbrace{\begin{bmatrix} -h(\alpha_0, u_0) \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{b_{eq}}. \quad (61)$$

5.2. Inequality Constraints

The inequality constraints of interest to be included in (53), in practice, are saturation constraints and rate limits, i.e.,

$$\begin{aligned} \underline{\alpha} &\leq \alpha \leq \bar{\alpha} \\ \underline{u} &\leq u \leq \bar{u} \\ \underline{\Delta \alpha} &\leq \alpha - \alpha_0 \leq \bar{\Delta \alpha} \\ \underline{\Delta u} &\leq u - u_0 \leq \bar{\Delta u}. \end{aligned} \quad (62)$$

Such inequality constraints are affine with respect to the optimization variables α and u , so they are tractable in online optimization.

Replacing (52) with its approximation (59) (or (61), for a single centralized motor) and expressing the constraints (62) in the variable $x = \text{col}(\Delta \alpha, \Delta u)$, we obtain an optimization problem in the form

$$\begin{aligned} \min_x \quad & J(x) \\ \text{s.t.} \quad & A_{eq}x = b_{eq} \\ & \underline{x} \leq x \leq \bar{x}, \end{aligned} \quad (63)$$

where

$$\begin{aligned} \underline{x} &= \text{col}(\max(\underline{\alpha} - \alpha_0, \underline{\Delta \alpha}), \max(\underline{u} - u_0, \underline{\Delta u})) \\ \bar{x} &= \text{col}(\min(\bar{\alpha} - \alpha_0, \bar{\Delta \alpha}), \min(\bar{u} - u_0, \bar{\Delta u})) \end{aligned} \quad (64)$$

are defined to express the inequality constraints (62) in a more compact way. Please note that (63) is a QP problem by construction, provided that the cost function $J(x)$ is quadratic.

5.3. Cost Function

The main term to be considered in a cost function for UAVs, especially in the case of multirotors, is the energy consumption $E(x)$. A quadratic form for $E(x)$ is often assumed in the UAV literature [29,30]; such a choice is mainly motivated by simplicity, as it usually leads to a QP problem. Motivated by the need for autonomy in UAVs, more detailed representations, coming from physically accurate consumption models, have been proposed in the literature as well. The most employed models are second-order polynomials [9] and irrational functions [34], as well as more generic non-linear functions [6]. All of the previous choices fit well with the proposed framework, as it is always possible to locally approximate the cost function with a quadratic cost function, as done for example in [31]. For the purpose of this work, the steady-state thruster consumption from [34] is employed. Given a generic matrix $M = [m_{ij}] \in \mathbb{R}^{m \times n}$ and $k \in \mathbb{R}$, let us introduce, with a slight abuse of notation, the short $|M|^k = [|m_{ij}|^k]$ (i.e., the k -th power of the absolute values of the entries of M , in an element-wise fashion). Then, the steady-state thruster consumption for a quadrotor can be rewritten as

$$E(x) = \sum_{i=1}^4 q_{u,i} |u_i|^{3/2} = |u^T|^{3/4} Q_u |u|^{3/4}, \quad (65)$$

where $|u|^{3/4}$ and $|u^T|^{3/4}$ should be intended element-wise as already anticipated. Furthermore, $q_{u,i}$ is assumed to be constant for simplicity (see [34] for further details) and

$$Q_u = \text{diag}(q_{u,1}, \dots, q_{u,4}), \tag{66}$$

with $q_{u,i} > 0$ for $i = 1, \dots, 4$. Hence, we can define the cost function

$$J(x) = E(x) + \Delta\alpha^T Q_{\Delta\alpha} \Delta\alpha + \Delta u^T Q_{\Delta u} \Delta u + (\alpha - \alpha_p)^T Q_\alpha (\alpha - \alpha_p), \tag{67}$$

where $Q_{\Delta\alpha}$, $Q_{\Delta u}$, Q_α , Q_u are symmetric positive definite matrices. The first term of (67), which takes into account energy consumption, is detailed in (65). The second and the third quadratic terms of (67) discourage variations in the control inputs to minimize actuator wear and tear. The last term of (67) defines preferred values α_p for the pitch angles α . This term becomes necessary in order to compensate for the first term (i.e., the one that discourages energy consumption), because the actuator is more energy-efficient when the pitch angle is large in magnitude. Thus, in the absence of the fourth term, the pitch angle is often stuck to its saturation limit to minimize energy consumption. In such a case, the only way to increase the lift further is to increase the motor speed, as the pitch angle cannot be increased in magnitude, so the maneuverability of the vehicle worsens.

Considering the second-order Taylor approximation for $E(x)$, i.e.,

$$E(x) \approx |u_0^T|^{3/4} Q_u |u_0|^{3/4} + \frac{3}{2} |u_0^T|^{1/2} Q_u \Delta u + \frac{3}{8} \Delta u^T Q_u \text{diag}(|u_0|^{-1/2}) \Delta u, \tag{68}$$

and performing some manipulation, (67) becomes

$$J(x) = \frac{1}{2} x^T \underbrace{\begin{bmatrix} 2(Q_\alpha + Q_{\Delta\alpha}) & & & 0_{4 \times 4} \\ & 0_{4 \times 4} & & 2Q_{\Delta u} + \frac{3}{4} Q_u \text{diag}(|u_0|^{-1/2}) \end{bmatrix}}_H x + \underbrace{\left[2(\alpha_0 - \alpha_p)^T Q_\alpha \quad \frac{3}{2} (|u_0|^{1/2})^T Q_u \right]}_{f^T} x + c, \tag{69}$$

where c is a constant term (i.e., a term that does not depend on x), which is irrelevant for the solution of the optimization problem, so it can be neglected.

5.4. Scaling and Infeasibility

The problem formulation according to (59) (or (61), for a single centralized motor), (64), (69), that is,

$$\begin{aligned} \min_x \quad & \frac{1}{2} x^T H x + f^T x \\ \text{s.t.} \quad & A_{eq} x = b_{eq} \\ & \underline{x} \leq x \leq \bar{x}, \end{aligned} \tag{70}$$

shows two drawbacks. First of all, the existence of a feasible solution is not guaranteed, so problem infeasibility must be dealt with. Moreover, the problem is ill-conditioned, because the magnitude of $\Delta\alpha$ and Δu is very different; according to (17)–(18), α is typically in the range of decimals (in radians), while u is typically in the range of tens of thousands (in radians per second squared), and the same occurs for $\Delta\alpha$ and Δu .

To improve the problem conditioning, diagonal scaling is performed [32]. We define a new optimization variable

$$z = P^{-1} x, \tag{71}$$

where $P = \text{diag}(\max(|\underline{x}|, |\bar{x}|))$ is a diagonal matrix such that the components of z share the same magnitude, as they become constrained in $[-1, 1]$ according to the rate limits in (62) (please note that the box constraint estimation $[-1, 1]$ is conservative; tighter constraints may apply).

In order to deal with infeasible problems, we soften the equality constraint by adding a vector of unconstrained artificial variables ζ into the equality constraint, while minimizing the norm of the artificial variable. The overall problem (70), after the change of variables and the softening of equality constraints, becomes

$$\begin{aligned} \min_{z, \zeta} \quad & \frac{1}{2} z^T P^T H P z + f^T P z + \zeta^T Q_\zeta \zeta \\ \text{s.t.} \quad & A_{eq} P z + \zeta = b_{eq} \\ & P^{-1} \underline{x} \leq z \leq P^{-1} \bar{x}, \end{aligned} \tag{72}$$

which can be finally rewritten in the classical form that is usually required by QP solvers, i.e.,

$$\begin{aligned} y^* = \arg \min_y \quad & \frac{1}{2} y^T \check{H} y + \check{f}^T y \\ \text{s.t.} \quad & \check{A}_{eq} y = \check{b}_{eq} \\ & \check{A} y \leq \check{b} \\ & \underline{y} \leq y \leq \bar{y}, \end{aligned} \tag{73}$$

where

$$\begin{aligned} y = \begin{bmatrix} z \\ \zeta \end{bmatrix} \quad \check{H} = \begin{bmatrix} P^T H P & 0_{8 \times 4} \\ 0_{4 \times 8} & Q_\zeta \end{bmatrix} \quad \check{f}^T = [f^T P \quad 0_{1 \times 4}] \\ \check{A}_{eq} = [A_{eq} P \quad I_4] \quad \check{b}_{eq} = b_{eq} \quad \underline{y} = \begin{bmatrix} P^{-1} \underline{x} \\ \underline{\zeta} \end{bmatrix} \quad \bar{y} = \begin{bmatrix} P^{-1} \bar{x} \\ \bar{\zeta} \end{bmatrix}, \end{aligned} \tag{74}$$

$\underline{\zeta}$ and $\bar{\zeta}$ are large enough to guarantee feasibility (e.g., $\bar{\zeta} = -\underline{\zeta} = M, M \rightarrow \infty$), while \check{A} and \check{b} are empty matrices, because no inequality constraints are left, with an exception made for the upper and lower bounds that are usually introduced separately.

Once the solution y^* of (72)–(74) is calculated, the solution to the control allocation problem is

$$\begin{aligned} x^* &= \begin{bmatrix} \Delta \alpha^* \\ \Delta u^* \end{bmatrix} = [P \quad 0_{8 \times 4}] y^* \\ \alpha^* &= \alpha_0 + \Delta \alpha^* \\ u^* &= u_0 + \Delta u^*. \end{aligned} \tag{75}$$

Remark 5. The choice of a quadratic penalty for ζ is made for simplicity, as it introduces a small number of additional variables and it makes the cost function smooth and hence easier to solve online. This strategy is similar to the quadratic penalty method in [32] and, more precisely, to the weighted least squares in [35]. As a drawback, such a penalty term is not exact, i.e., the solution of the penalized problem is not equivalent to the solution of the original problem [32,36]. In particular, the higher the penalty through Q_ζ , the more the solutions of the two problems coincide, but also the worse is the conditioning. Thus, Q_ζ is a trade-off between ideal solution quality and acceptable conditioning. We claim that this issue is not crucial, because several approximations are introduced to simplify the problem, and so the solution is not exact anyway. Alternatively, it is possible to introduce additional positively constrained slack variables and to employ exact penalty functions, such as ℓ_1 and ℓ_∞ penalties, at the price of dealing with the nonsmoothness of the cost function (see [32] for further details).

6. Numerical Simulation

A numerical simulation is performed using MATLAB, where the inner loop and the outer loop run at 400 Hz (i.e., ArduPilot's loop rate) and 10 Hz, respectively. A zero-order hold is performed between consecutive samples. The quadrotor parameters are set according to [18] and they are reported in Table 1. Different from [18], the friction is neglected and the thrust-to-weight ratio is slightly increased (from 2.5 to 3.131) in order to face severe faults.

Table 1. Variable-pitch quadrotor parameters.

Parameter	Variable	Value	Unit
Total system mass	m	1.37	[kg]
Inertia about x_B, y_B	J_x, J_y	7.5×10^{-3}	[kg·m ²]
Inertia about z_B	J_z	1.3×10^{-2}	[kg·m ²]
Arm length	l	0.3	[m]
Lift curve slope	$c_{l\alpha}$	5.23	[−]
Zero lift drag coefficient	c_{d0}	0.01	[−]
Propeller radius	R_a	0.18	[m]
Propeller chord	c	0.03	[m]
Rotor solidity	σ	0.106	[−]
Gravitational acceleration	g	9.81	[m/s ²]
Air density	ρ	1.225	[kg/m ³]
Linear friction coefficient	k_t	0	[N·s/m]
Angular friction coefficient	k_r	0	[N·s·m]
Minimum pitch angle	α_{min}	0.05	[deg]
Maximum pitch angle	α_{max}	15	[deg]
Pitch angle rate limit	$\Delta\alpha_{max}$	60	[deg/s]
Minimum desired square rotor speed	u_{min}	0	[rad ² /s ²]
Maximum desired square rotor speed	u_{max}	2×10^5	[rad ² /s ²]
Motor square speed rate limit	Δu_{max}	1.6×10^5	[rad ² /s ³]

The control parameters were set empirically through a coarse grid search (as such, better performances could be achieved with a finer tuning). The closed-loop poles for the inner loop are placed in -2 and -2.1 for z_F , in -10 and -11 for φ and θ , and finally in -5 and -5.1 for ψ . The outer loop poles are instead placed in -1 and -2 for both x_F and y_F . A properly scaled white Gaussian sensor noise is simulated according to an MPU-9250 IMU TDK InvenSense [37], which is a common device in quadrotor flight controllers (i.e., in Pixhawk 4).

6.1. Independent Motors' Speed

Let us first consider a VPQ equipped with four independent motors and four independent pitch servos. The simulation is performed for a total of 30 s under the external wind force reported in Figure 3, which also reports the correspondent estimations provided by the observer, where the gain matrix has been set as $H = 5I$, with I being the identity matrix.

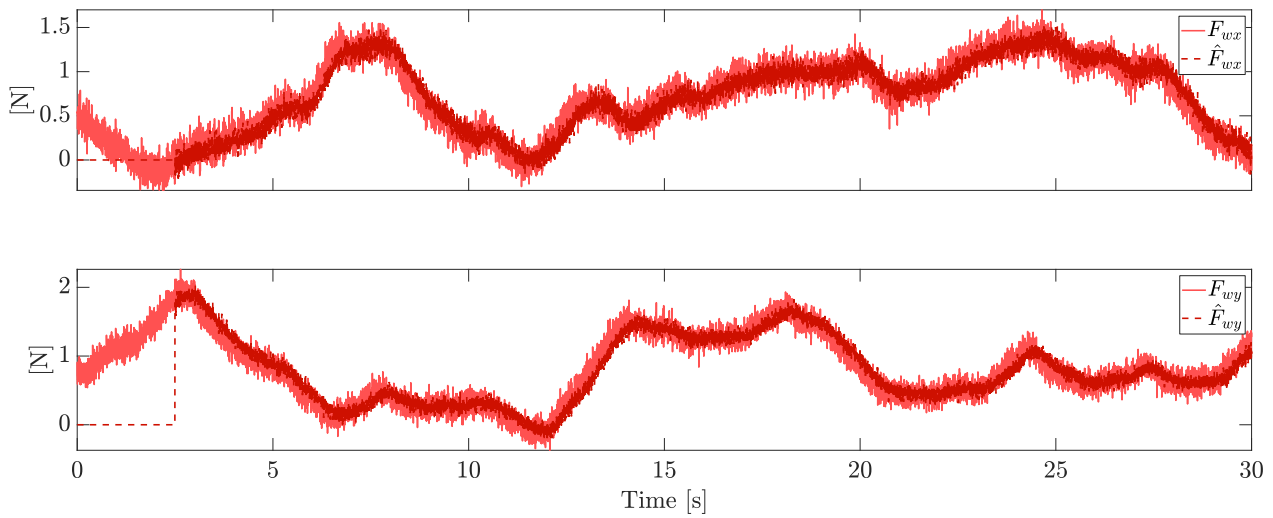


Figure 3. Horizontal wind estimation (independent motors' speed).

Due to the stochastic nature of the wind, achieving the boundedness of the estimation error replaces the ideal goal of a perfect convergence, which is here achieved in the case of bounded derivatives of the wind force. The estimation of the faults and the wind is practically started after a finite time from the beginning of the flight (i.e., after 1 s of simulation in this example), in order to make the initial conditions of the observer practically irrelevant.

Faults components and their estimations are reported in Figure 4.

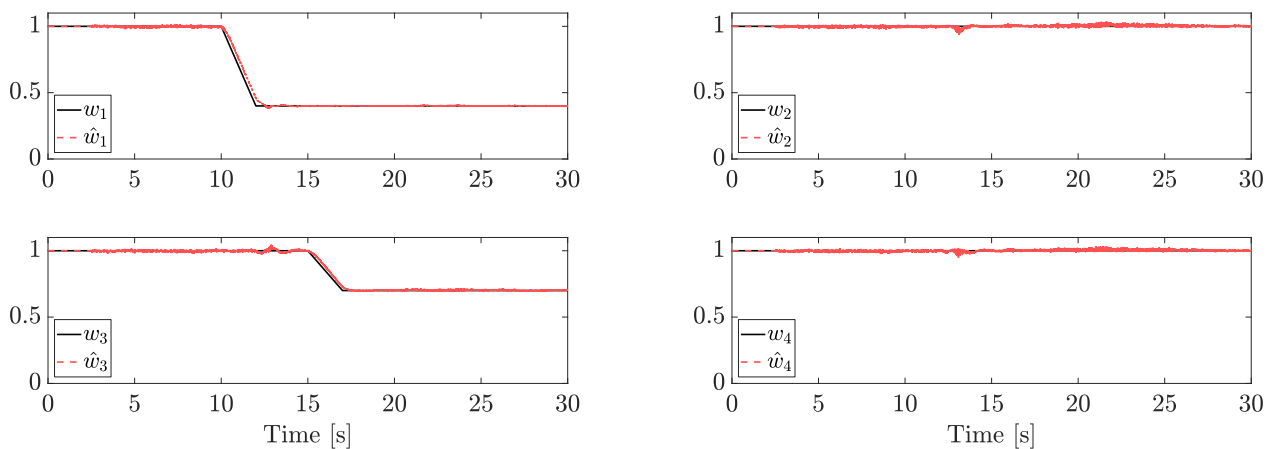


Figure 4. Fault estimation (independent motors' speed).

Two trapezoidal faults are injected. The first one affects motor 1 after 10 s (the LOE magnitude is 60%, i.e., $w_1 = 0.4$) and a second one affects motor 3 after 15 s (the LOE magnitude is 30%, i.e., $w_3 = 0.7$). In the range $t \in [0, 1]$, each \hat{w}_i is set equal to 1, matching the reasonable assumption of no faults at the initial time (or, without loss of generality, we assume that we know the fault status at the beginning of the flight). Note that, since each estimation is provided independently from the NDO, the same results can be achieved in the case of simultaneous faults.

The tracking reference components and the related state variables are reported in Figure 5.

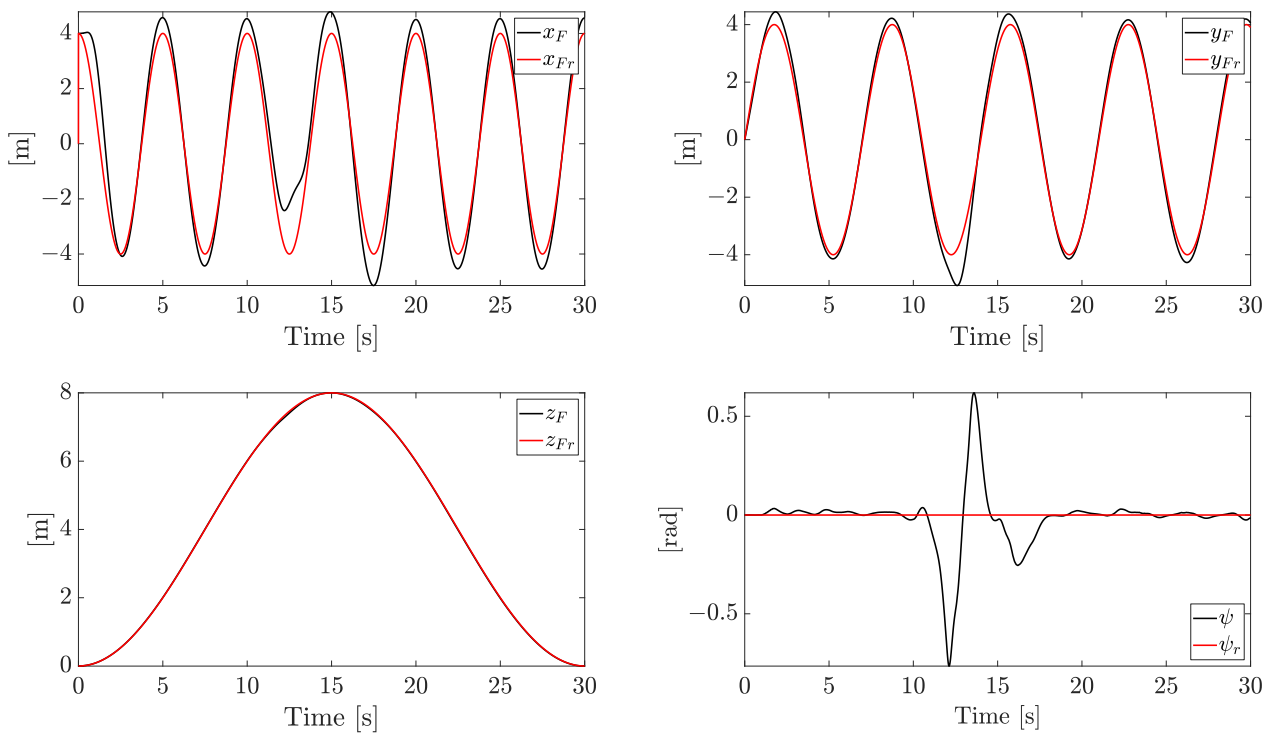


Figure 5. Tracking performances (independent motors' speed).

The references on x_F and y_F are sinusoidal signals, the reference on z_F forces the quadrotor to face both ascending and descending profiles, while the yaw angle is kept at zero. The system is forced to change the velocity constantly in both magnitude and direction, with a linear speed magnitude up to 7 m/s. All the references are smoothly tracked with an acceptable error (the plotted variables are the actual state variables instead of the noisy ones), with the exception of the yaw component (ψ), which exhibits a temporary increasing error when the fault worsens.

The references φ_r and θ_r generated by the outer loop are reported in Figure 6.

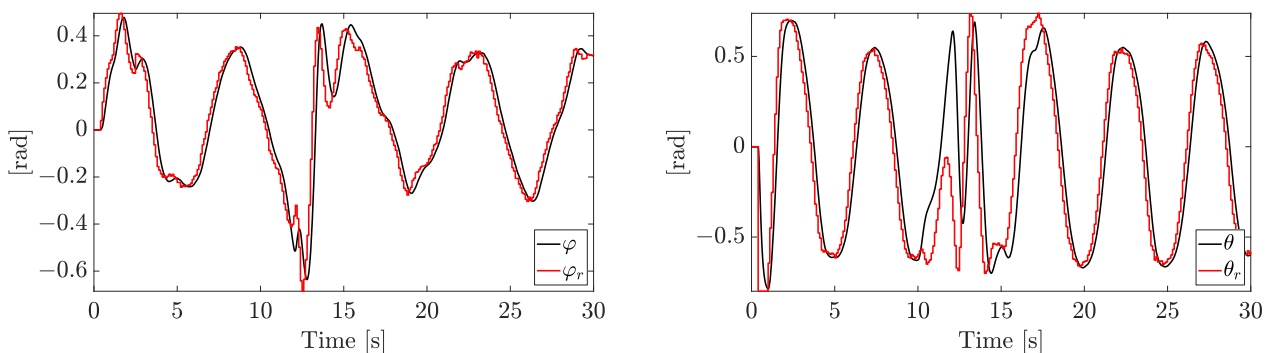


Figure 6. Inner loop tracking performances on desired angles (independent motors' speed).

Please note that, when the fault is rapidly worsening, the tracking capabilities in the inner loop are affected. The inner/outer loop approach, which employs a higher frequency of the inner loop and a zero-order hold, allows us to track the references, despite the presence of noise, faults, and disturbances.

The control inputs u and α are reported in Figure 7.

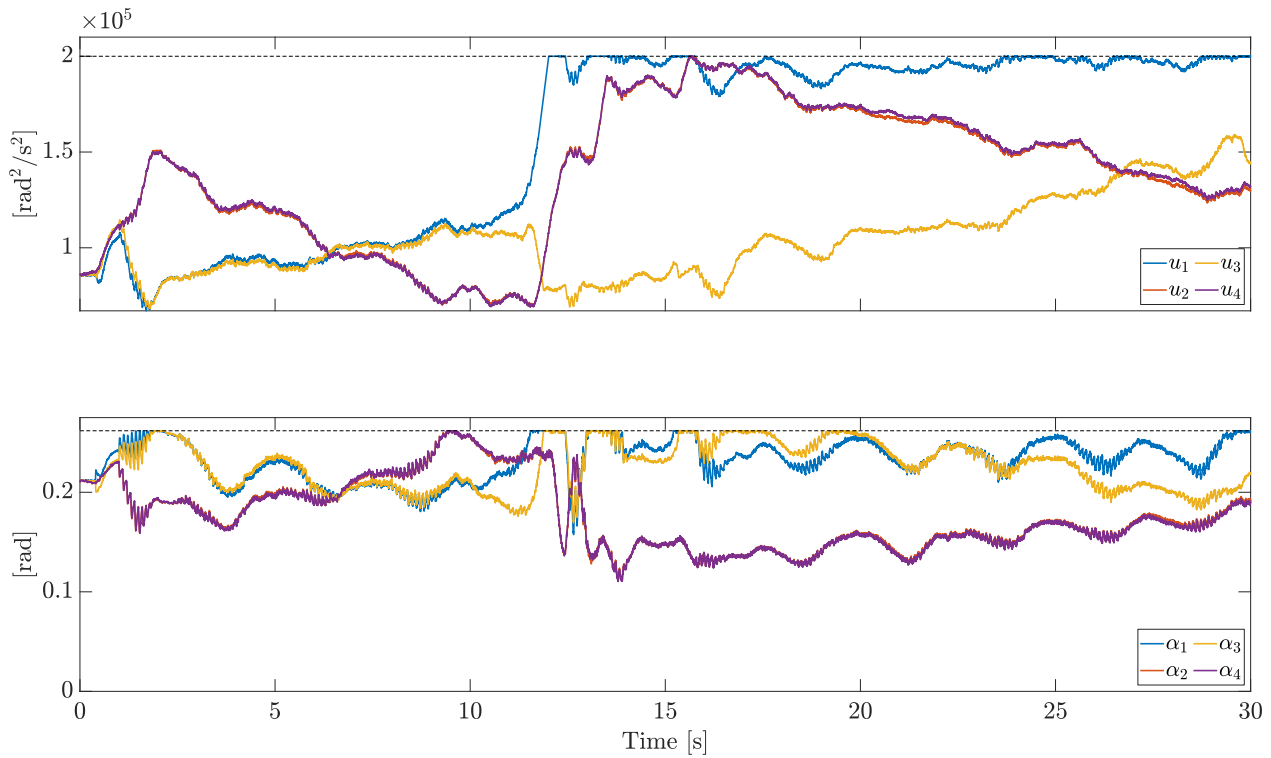


Figure 7. Control inputs u and α (independent motors' speed).

All the actuators are subject to the same saturation constraints. This allows us to plot the bounds with horizontal dashed lines in Figure 7 (upper bounds in this case). In particular, the saturation limits have been set as

$$\begin{aligned}
 \underline{\alpha} &= \text{col}(\alpha_{min}, \alpha_{min}, \alpha_{min}, \alpha_{min}) \\
 \bar{\alpha} &= \text{col}(\alpha_{max}, \alpha_{max}, \alpha_{max}, \alpha_{max}) \\
 \underline{u} &= \text{col}(u_{min}, u_{min}, u_{min}, u_{min}) \\
 \bar{u} &= \text{col}(u_{max}, u_{max}, u_{max}, u_{max}) \\
 \underline{\Delta\alpha} &= -\underline{\alpha} = \text{col}(\Delta\alpha_{max}, \Delta\alpha_{max}, \Delta\alpha_{max}, \Delta\alpha_{max}) \\
 \underline{\Delta u} &= -\underline{u} = \text{col}(\Delta u_{max}, \Delta u_{max}, \Delta u_{max}, \Delta u_{max}),
 \end{aligned} \tag{76}$$

where u_{min} , u_{max} , α_{min} , and α_{max} are detailed in Table 1. Considering $\alpha = \bar{\alpha}$, \bar{u} corresponds to a thrust-to-weight ratio equal to 3.131.

At the beginning, when no fault is present, the opposite actuators (1 and 3, 2 and 4) show similar motor speeds and pitch angles, due to symmetry. As soon as the fault on motor 1 occurs, the opposite actuators 1 and 3 behave differently; in response to the severe fault acting on motor 1, the control allocation algorithm demands a high speed and a large pitch angle to compensate for the missing lift force.

6.2. Centralized Motor

In this section, we consider a VPQ equipped with a single centralized motor and four independent pitch servos. The simulation conditions and the wind components are the same as the previous simulation, including the same random seed. Figure 8 reports the wind effect and its estimations provided by the observer; no noticeable differences can be noted, as expected, because the fault estimation does not directly depend on the actual inputs.

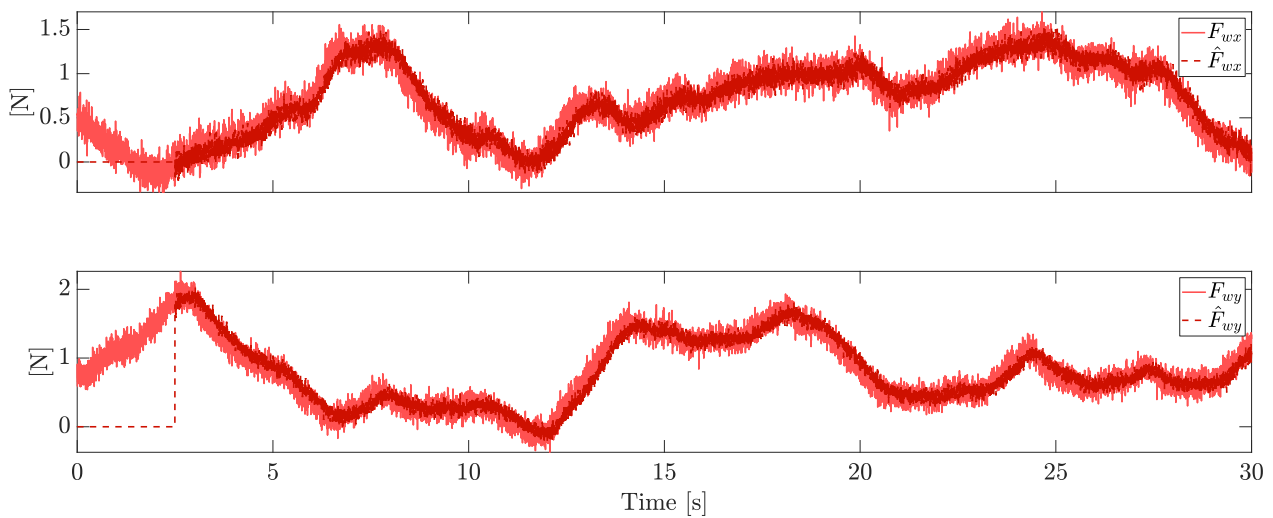


Figure 8. Horizontal wind estimation (centralized motor).

The same consideration holds for the faults and their estimations (Figure 9).

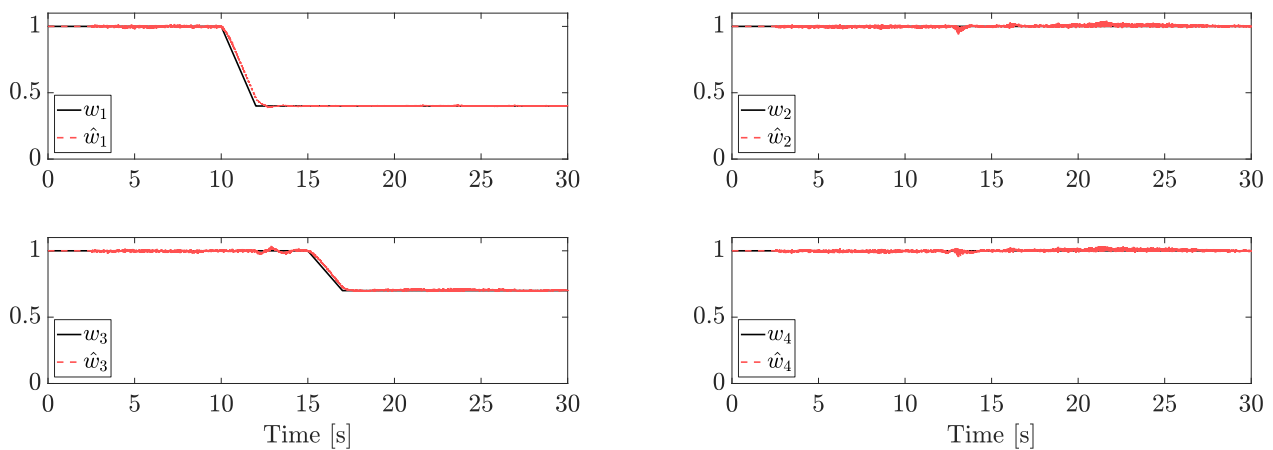


Figure 9. Fault estimation (centralized motor).

The tracking reference components and the related state variables are reported in Figures 10 and 11. The performances are similar; minor differences in the outer loop tracking capabilities can be detected.

Finally, the control inputs u and α are reported in Figure 12.

First of all, please note that the motor speeds coincide, as expected. Furthermore, in response to the fault acting on motor 1, the control allocation algorithm demands a high motor speed to compensate for the missing lift force. As there is only one centralized motor, the speed of every propeller must increase. Analogously, please note that, the larger the fault, the higher the average value of the pitch angle.

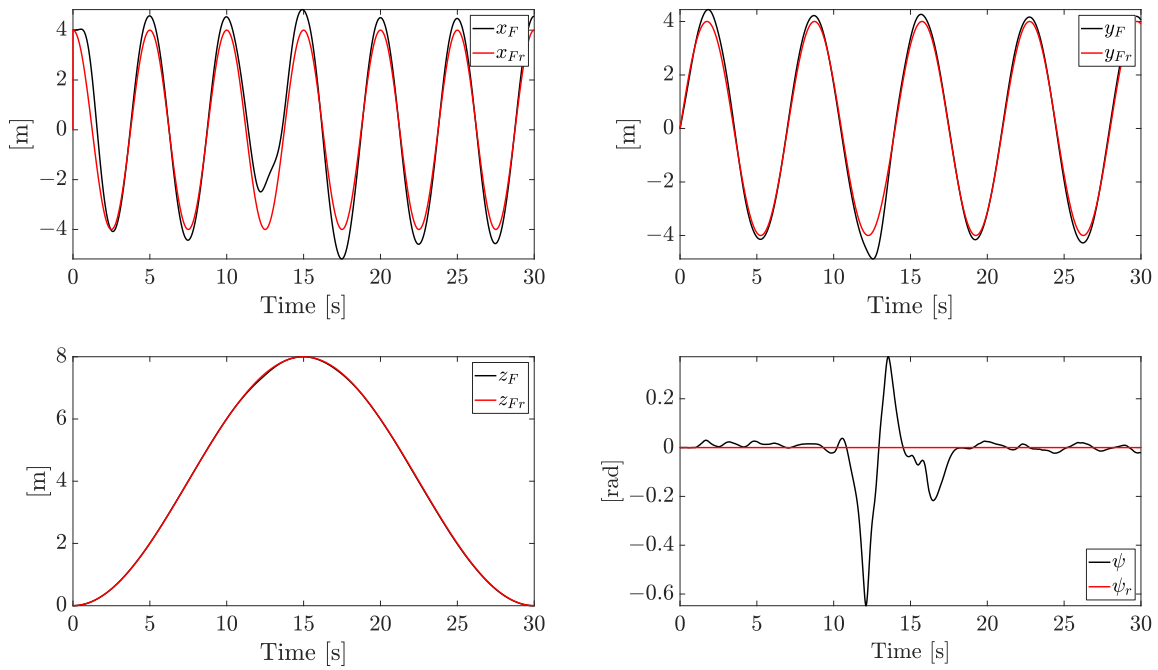


Figure 10. Tracking performances (centralized motor).

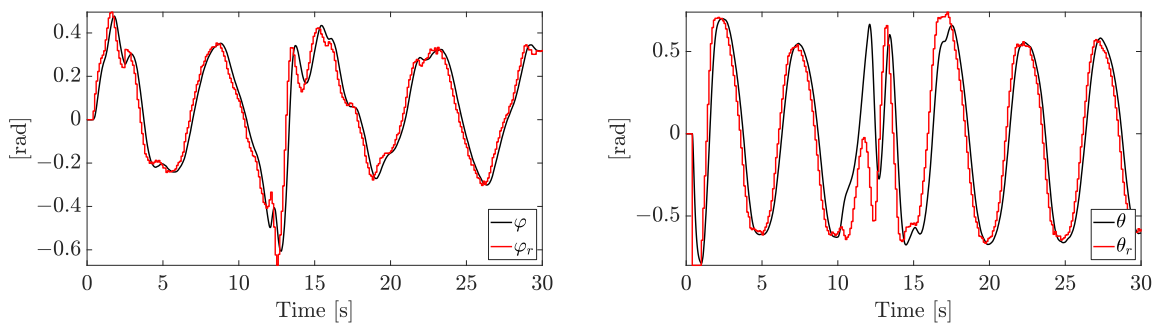


Figure 11. Inner loop tracking performances on desired angles (centralized motor).

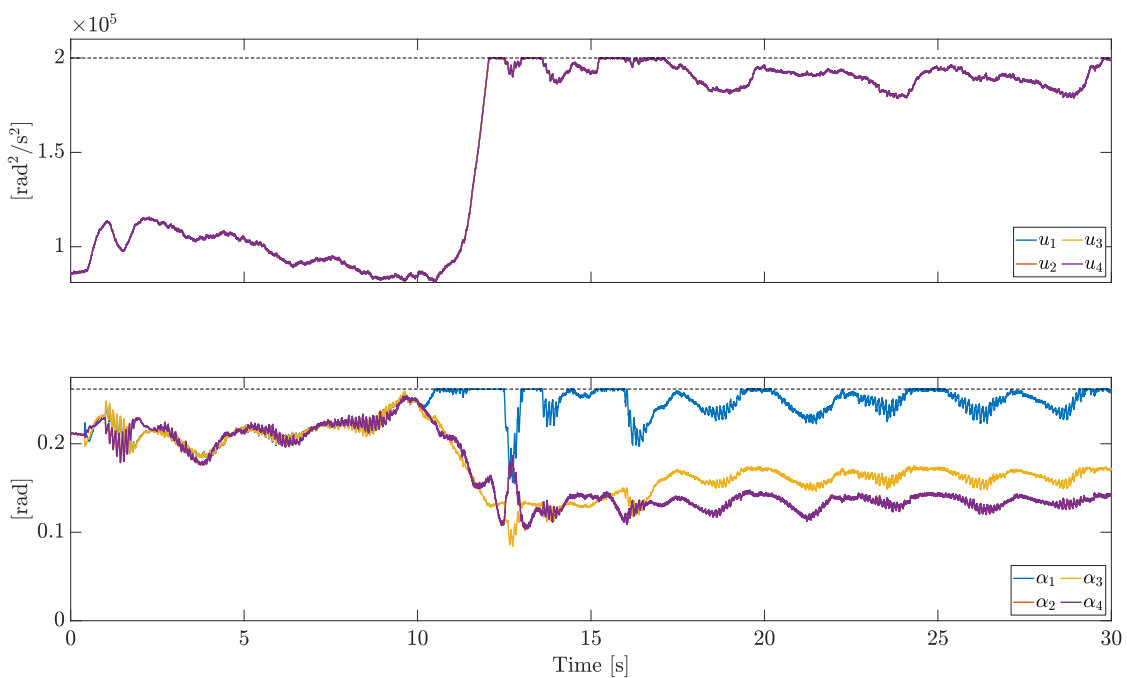


Figure 12. Control inputs u and α (centralized motor).

As the motor speed is higher on average in the case of the centralized motor (Figures 7 and 12), the energy consumption is higher. This is unavoidable, because in the case of the centralized motor, the input redundancy is lower than in the case of independent motors. In Table 2, we summarize the overall normalized (i.e., $Q_u = I$) consumption during the flight, as proposed in (65).

Table 2. Consumptions: independent motors' speed vs. centralized motor.

Scenario	Consumption ($ u ^{3/2}$)
Independent motors	6.144×10^9
Centralized motor	7.551×10^9

7. Conclusions

Unlike standard quadrotors that are equipped with four fixed-pitch blades and four independent motors, VPQs can vary both the rotation speed and the blade pitch of each propeller, thus increasing the degrees of freedom for control purposes. In addition to higher thrust rate of change, reverse thrust and reverse flight capabilities, and scaling well with size, VPQs enable for fault-tolerant control strategies thanks to their inherent input redundancy, whether they are equipped with four independent motors or a centralized one. We have shown that a single observer can estimate both actuator loss of effectiveness and the external disturbance given by horizontal wind. According to the disturbance observer-based control framework, the estimation of the wind is fed forward in the outer (position) control loop. The estimation of the actuator loss of effectiveness is instead employed in the control allocation algorithm. It manages each propeller's pitch and motor speed to generate the force and the torques commanded by the inner (attitude and altitude) control loop. Simulation results show that the estimation of both the disturbance and the actuator loss of effectiveness is practically feasible in the presence of conventional measurement noise in commercial devices. We remark that the fault estimation is performed from kinematic data of the onboard inertial measurement unit only, without the need to measure the motor speed nor the current drawn by the actuators.

Comparing centralized and independent motors, the former is less efficient in terms of energy consumption, especially in the case of faults, because all the propellers must spin faster. Clearly, many other factors should be considered to evaluate the energy consumption properly, such as possible weight and cost reduction. In any case, both centralized and independent motors can tolerate actuator faults using the proposed scheme. The proposed control scheme could also deal with pitch lock-in-place, provided that reliable information on the stuck fault is available (e.g., measuring the current pitch angle or estimating it through a fault detection algorithm). Practically, it is sufficient to constrain the faulty pitch servo to the current value, and the control allocation algorithm can accommodate for the stuck fault using the remaining control inputs. However, in the case of a single centralized motor, the reduced amount of redundancy makes it less robust to lock-in-place faults and severe loss of effectiveness.

Author Contributions: Conceptualization, A.B., R.F., A.F., and A.M.; methodology, A.B. and R.F.; software, A.B. and R.F.; validation, A.B. and R.F.; formal analysis, A.B., R.F., A.F., and A.M.; writing—original draft preparation, A.B. and R.F.; writing—review and editing, A.F. and A.M.; visualization, A.B. and R.F.; supervision, A.F. and A.M.; project administration, A.F. and A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

LOE	Loss Of Effectiveness
NDO	Non-linear Disturbance Observer
QP	Quadratic Programming
SQP	Sequential Quadratic Programming
UAV	Unmanned Aerial Vehicle
VPQ	Variable-Pitch Quadrotor

References

- Cutler, M.; Ure, N.K.; Michini, B.; How, J. Comparison of fixed and variable pitch actuators for agile quadrotors. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Portland, OR, USA, 8–11 August 2011; pp. 1–17.
- Cutler, M.; How, J. Actuator constrained trajectory generation and control for variable-pitch quadrotors. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Minneapolis, MN, USA, 13–16 August 2012; pp. 1–15.
- Gupta, N.; Kothari, M.; Abhishek. Flight dynamics and nonlinear control design for variable-pitch quadrotors. In Proceedings of the 2016 American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 3150–3155.
- Panizza, P.; Invernizzi, D.; Riccardi, F.; Formentin, S.; Lovera, M. Data-driven attitude control law design for a variable-pitch quadrotor. In Proceedings of the 2016 American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 4434–4439.
- Cutler, M.; How, J.P. Analysis and control of a variable-pitch quadrotor for agile flight. *J. Dyn. Syst. Meas. Control.* **2015**, *137*, 101002. [[CrossRef](#)]
- Fresk, E.; Nikolakopoulos, G. Experimental model derivation and control of a variable pitch propeller equipped quadrotor. In Proceedings of the IEEE Conference on Control Applications (CCA), Juan Les Antibes, France, 8–10 October 2014; pp. 723–729.
- Isermann, R. *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*; Springer: Berlin/Heidelberg, Germany, 2005.
- Pretorius, A.; Boje, E. Design and modelling of a quadrotor helicopter with variable pitch rotors for aggressive manoeuvres. *IFAC Proc. Vol.* **2014**, *47*, 12208–12213. [[CrossRef](#)]
- Arellano-Quintana, V.M.; Merchán-Cruz, E.A.; Franchi, A. A Novel Experimental Model and a Drag-Optimal Allocation Method for Variable-Pitch Propellers in Multirotors. *IEEE Access* **2018**, *6*, 68155–68168. [[CrossRef](#)]
- Sheng, S.; Sun, C. Control and optimization of a variable-pitch quadrotor with minimum power consumption. *Energies* **2016**, *9*, 232. [[CrossRef](#)]
- Porter, R.; Shirinzadeh, B.; Choi, M.H. Experimental analysis of variable collective-pitch rotor systems for multirotor helicopter applications. *J. Intell. Robot. Syst.* **2016**, *83*, 271–288. [[CrossRef](#)]
- Pang, T.; Peng, K.; Lin, F.; Chen, B.M. Towards long-endurance flight: Design and implementation of a variable-pitch gasoline-engine quadrotor. In Proceedings of the 2016 12th IEEE International Conference on Control and Automation (ICCA), Kathmandu, Nepal, 1–3 June 2016; pp. 767–772.
- Abhishek, A.; Duhoon, A.; Kothari, M.; Kadukar, S.; Rane, L.; Suryavanshi, G. Design, development, and closed-loop flight-testing of a single power plant variable pitch quadrotor unmanned air vehicle. In Proceedings of the 73rd American Helicopter Society Annual Forum, Fort Worth, TX, USA, 9–11 May 2017; pp. 205–218.
- Wang, Z.; Groß, R.; Zhao, S. Control of centrally-powered variable pitch propeller quadcopters subject to propeller faults. *Aerosp. Sci. Technol.* **2022**, *120*, 107245. [[CrossRef](#)]
- Podhradský, M.; Bone, J.; Coopmans, C.; Jensen, A. Battery model-based thrust controller for a small, low cost multirotor Unmanned Aerial Vehicles. In Proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 28–31 May 2013; pp. 105–113.
- Yang, P.; Wen, C.; Geng, H.; Liu, P. Intelligent fault diagnosis method for blade damage of quad-rotor UAV based on stacked pruning sparse denoising autoencoder and convolutional neural network. *Machines* **2021**, *9*, 360. [[CrossRef](#)]
- Felicetti, R.; Baldini, A.; Freddi, A.; Longhi, S.; Monteriù, A. Fault Tolerant Control of a Variable Pitch Quadrotor. In Proceedings of the 15th European Workshop on Advanced Control and Diagnosis, Bologna, Italy, 21–22 November 2019; pp. 235–255.
- Baldini, A.; Felicetti, R.; Freddi, A.; Longhi, S.; Monteriù, A. Actuator fault tolerant control of variable pitch quadrotor vehicles. *IFAC-PapersOnLine* **2020**, *53*, 4095–4102. [[CrossRef](#)]
- Perozzi, G.; Efimov, D.; Biannic, J.M.; Planckaert, L. Trajectory tracking for a quadrotor under wind perturbations: Sliding mode control with state-dependent gains. *J. Frankl. Inst.* **2018**, *355*, 4809–4838. [[CrossRef](#)]
- Guo, K.; Jia, J.; Yu, X.; Guo, L.; Xie, L. Multiple observers based anti-disturbance control for a quadrotor UAV against payload and wind disturbances. *Control. Eng. Pract.* **2020**, *102*, 104560. [[CrossRef](#)]
- Michieletto, G.; Ryll, M.; Franchi, A. Fundamental actuation properties of multirotors: Force–moment decoupling and fail–safe robustness. *IEEE Trans. Robot.* **2018**, *34*, 702–715. [[CrossRef](#)]
- Fossen, T. *Guidance and Control of Ocean Vehicles*; Wiley: Hoboken, NJ, USA, 1994.

23. Leishman, G.J. *Principles of Helicopter Aerodynamics with CD Extra*; Cambridge University Press: Cambridge, UK, 2006.
24. Li, S.; Yang, J.; Chen, W.; Chen, X. *Disturbance Observer-Based Control: Methods and Applications*; Taylor & Francis: Abingdon, UK, 2014.
25. Yang, J.; Chen, W.H.; Li, S. Non-linear disturbance observer-based robust control for systems with mismatched disturbances/uncertainties. *IET Control. Theory Appl.* **2011**, *5*, 2053–2062. [[CrossRef](#)]
26. Nagaty, A.; Saeedi, S.; Thibault, C.; Seto, M.; Li, H. Control and navigation framework for quadrotor helicopters. *J. Intell. Robot. Syst.* **2013**, *70*, 1–12. [[CrossRef](#)]
27. Madani, T.; Benallegue, A. Backstepping control for a quadrotor helicopter. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 3255–3260.
28. Bouabdallah, S.; Siegwart, R. Full control of a quadrotor. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 153–158.
29. Marks, A.; Whidborne, J.F.; Yamamoto, I. Control allocation for fault tolerant control of a VTOL octorotor. In Proceedings of the 2012 UKACC International Conference on Control, Cardiff, UK, 3–5 September 2012; pp. 357–362.
30. Merheb, A.; Nourra, H.; Bateman, F. Active fault tolerant control of octorotor uav using dynamic control allocation. In Proceedings of the International Conference on Intelligent Unmanned Systems, Montreal, QC, Canada, 29 September–1 October 2014.
31. Johansen, T.A.; Fossen, T.I.; Berge, S.P. Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming. *IEEE Trans. Control. Syst. Technol.* **2004**, *12*, 211–216. [[CrossRef](#)]
32. Wright, S.; Nocedal, J. *Numerical Optimization*; Springer Series in Operations Research and Financial Engineering; Springer: New York, NY, USA, 1999; Volume 35, p. 7.
33. Gros, S.; Zanon, M.; Quirynen, R.; Bemporad, A.; Diehl, M. From linear to nonlinear MPC: Bridging the gap via the real-time iteration. *Int. J. Control.* **2020**, *93*, 62–80. [[CrossRef](#)]
34. Dyer, E.; Sirouspour, S.; Jafarinasab, M. Energy optimal control allocation in a redundantly actuated omnidirectional UAV. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 5316–5322.
35. Harkegard, O. Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation. In Proceedings of the 41st IEEE Conference on Decision and Control, Piscataway, NJ, USA, 10–13 December 2002; Volume 2, pp. 1295–1300.
36. Kerrigan, E.C.; Maciejowski, J.M. Soft constraints and exact penalty functions in model predictive control. In Proceedings of the Control 2000 Conference, Cambridge, MA, USA, 12–15 November 2000; pp. 2319–2327.
37. TDK InvenSense. MPU-9250, Nine-Axis (Gyro+ Accelerometer+ Compass) MEMS MotionTracking™ Device. 2016. Available online: <https://invensense.tdk.com/download-pdf/mpu-9250-datasheet/> (accessed on 5 May 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.