Chapter 1

Control Techniques to Deal with the Damage of a Quadrotor Propeller

Fabio Ruggiero, Diana Serra,
Vincenzo Lippiello, Bruno Siciliano[1]

1.1 Introduction

Civilians, servicemen, mass media, and researchers paid close attention to unmanned aerial vehicles (UAVs) in the last decade. The growth of applications in which vertical take-off and landing (VToL) UAVs are becoming present in every-day life is incredible. Professional photographers and film-makers are now always accompanied by certified UAV pilots to see the scene from different points of view. One of the most prominent electronic commerce websites around the world is planning to deliver packages to customers hands in thirty minutes, or less, using hexacopters [1]. The White House and the National Science Foundation accelerated the use of UAVs for civilian uses in monitoring and inspection of physical infrastructures, quick response to disasters, agricultural and meteorological domains with a considerable amount of funding [2]. Some companies started to think about a sort of personal UAV equipped with a camera to record self-movies [3], particularly indicated for a sportsman. Oil and gas facilities see the nondestructive measurement tests as a prosperous application for the UAVs in refineries since they are currently performed by human operators climbing huge and costly scaffolding. Several robotic commercial solutions are currently (or are ready-to-be) available like the APPELLIX drone [www.apellix.com], the Texo Drone Survey and Inspection platform [www.texodroneservices.co.uk/blog/56], and the Ronik Inspectioneering UT device [www.inspectioneering.eu].

Researchers see in the UAVs an unstable dynamic system that perfectly fits the challenge of implementing safe and robust control and mechatronic designs to accomplish every application mentioned above and go further beyond. As a matter of fact, the UAVs can be both employed for passive tasks like inspection and surveillance, and active tasks like grasping and aerial manipulation. This scenario requires the introduction not only of rules and regulation but also of safe controllers. An overview about control applied to UAVs, and the related applications in aerial ma-

[1] Authors are with PRISMA Lab, Department of Electrical Engineering and Information Technology, Università degli Studi di Napoli Federico II, via Claudio 21, 80125, Naples, Italy. Corresponding author's email fabio.ruggiero@unina.it

nipulation, can be found in [4]. The design of controllers for safety-critical systems is thus required: fault detection, diagnosis, and tolerance approaches become essential. Fault tolerance methods try to maintain the same functionalities of the system, allowing reduced performance when damage appears [5]. Passive fault-tolerant control systems (PFTCS) do not alter the structure of the controller, while the active fault tolerant control systems (AFTCS) reconfigure the control actions to guarantee stability and acceptable performance of the system [6].

This chapter wants to be a tutorial to guide the reader towards the implementation of an AFCTS dealing with the damage of a UAV propeller. In particular, the quadrotor with fixed propellers is the aerial device taken into account. The quadrotor is a versatile and agile device with four rotors lying on the same plane, with the adjacent propellers rotating in the opposite directions [https://www.youtube.com/watch?v=w2itwFJCgFQ]. This chapter resumes the results provided in [7] and [8] in a tutorial form, thus neglecting the stability proofs, to leave room to a fast and concise description of the implementation procedures.

The outline of the chapter is as follows. The problem statement and the literature review about it are provided in the next section. The notation and mathematical modeling are introduced in Section 1.3. The control methodologies are described in Section 1.4 with the related control schemes. The comparison between the proposed methodologies is carried out through numerical simulation within Section 1.5. Section 1.6 concludes the chapter.

1.2 Problem statement

In this chapter, the damage of a quadrotor propeller is addressed. To cope with such a situation, the adopted solution is to turn off the damaged motor completely. Besides, the motor aligned on the same quadrotor axis of the broken propeller is turned off as well. This configuration is a birotor with fixed propellers. Such a solution also includes the case of two broken propellers provided that they are aligned on the same axis. Other case studies are not taken into account.

It is assumed that the system already detected the failure: it is out the scope of this work the implementation of a fault-detection procedure. Besides, the control methodologies addressed in this chapter starts from a birotor configuration with a given initial condition of the state. The switch between the nominal quadrotor working condition into the birotor emergency condition is out of this scope of this chapter. Therefore, it is possible to provide the following problem formulation.

- **Problem**. *Control the position of the birotor from an initial position to the desired one in the Cartesian space, along a desired trajectory.*

The next sections solve the problem above by providing a tutorial extracting the solutions from [7] and [8].

In the literature, several alternative solutions can be found solving similar problems, or considering the parts that are neglected in this chapter.

Regarding the fault-detection, a method was proposed in [9]. A Luenberger observer was instead employed in [10] along with a sliding mode controller. When the loss in the efficiency is verified in each of the four propellers, the gain scheduling approach proposed in [11] may be employed.

A thorough literature review reveals that there exist many methods addressing the control of a UAV in case of one or more motor partial failures. Most of them belong to the AFTCS class. Instead, a PFTCS for a for a coaxial counter-rotating octorotor based on the super-twisting algorithm was designed in [12]. The proposed second-order sliding mode technique ensured robustness with respect to uncertainties and disturbances, and it could also deal directly with faults and failures by compensating for the actuator loss in the system without prior knowledge on the fault, its location, and its severity. About AFTCS, a method was tackled in [13] to estimate the aerial vehicle model after the failure detection (50% loss in the efficiency of a quadrotor propeller), guaranteeing the stability of the UAV. A backstepping approach was proposed in [14] but only 25% performance loss in the motors was considered. A comparison among different methodologies was compared in [15] for a 50% loss in propellers performance.

On the other hand, additional methods consider the complete failure of a quadrotor propeller. Despite the impossibility to control the yaw angle, a feedback linearization with a proportional-derivative (PD) based controller was employed in [16] to control a quadrotor with a wholly broken motor. However, no stability analysis and coupling effects between inner and outer loops were considered as instead in [7]. A controller for an equidistant trirotor was designed in [17], but the formulation is available only for spiral motions. An H-infinity loop shaping technique was adopted in [18] for the safe landing of a quadrotor with a propeller failure. Periodic solutions were exploited in [19] together with an LQR to control the quadrotor in case of single, two opposing, or three propellers failure. A reconfigurability analysis of UAVs with four and six rotors was investigated in [20]. The hovering control in the presence of the blockage or the complete loss of a rotor was pursued. The quadrotor is shown to be not reconfigurable, while the hexrotors can handle a blockage without difficulties. Furthermore, it is shown that whenever a UAV is stabilisable, it is also possible to recover the system's behavior accurately. A three-loop hybrid nonlinear controller was instead designed in [21] to achieve high-speed flight of a quadrotor with a complete loss of a single rotor. Robustness against complex aerodynamic effects brought by both fast translational and fast spinning motion of a damaged quadrotor was shown. As a result, the UAV can continue high-speed missions instead of having an emergency landing immediately. A morphing quadrotor was shown in [22] that can stay in hover after rotor failure. Similarly to [7] and [8], an emergency fault-tolerant controller was developed in [23] employing a biritor through a bounded control law. Instead of using a birotor, an emergency controller based on quadrotor to trirotor conversion maneuver was applied experimentally in [24] on the AR Drone 2 platform to recover from a total failure of one rotor. The proposed controller was based on control re-allocation, where the infected actuator was exempted from the control effect, and control efforts are redistributed among the remaining actuators. Hence, the emergency controller transformed the infected quadrotor into a trirotor, using a simple proportional-integral-derivative (PID) based controller. To control the attitude of a quadrotor and to increase robustness against model uncer-

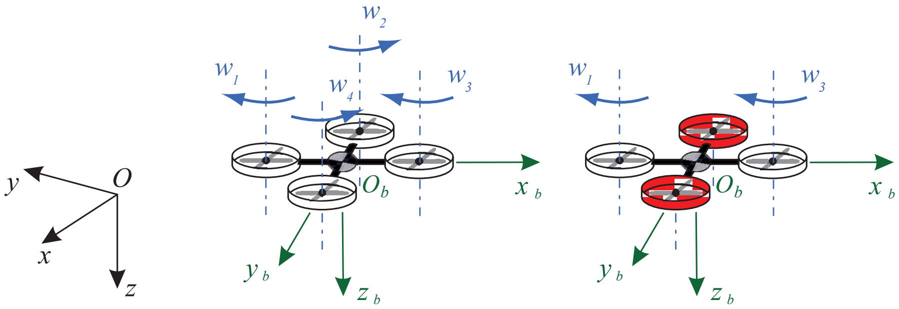


Figure 1.1 *Left. The quadrotor and related frames: black, the inertial frame Σ_i; green, the body frame Σ_b; blue, the speed and label of each motor. Right. The birotor configuration with in red the turned off propellers.*

tainties and actuator faults, a sliding mode controller was applied in [25]. Besides, an adaptive fuzzy system was employed to compensate for the estimation error of nonlinear functions and faulty parts. In order to avoid instability and to increase the robustness of the closed-loop system, a new parallel fuzzy system was proposed along with a main fuzzy system. The adaptation rules of the main and parallel fuzzy systems were extracted from Lyapunov's stability theory.

A complete AFCTS architecture, including error detection, fault isolation, and system recovery, was presented in [26]. The diagnosis system was based on the motor speeds and currents measurements. Once the motor failure or the rotor loss is diagnosed, a recovery algorithm was applied using the pseudo-inverse control allocation approach to redistribute the control efforts among the remaining actuators. An approach coping with not only fault detection and isolation but also fault-tolerant control was proposed in [27]. An incremental nonlinear dynamic inversion approach was introduced to design the fault-tolerant controller for the quadrotor in the presence of the fault. The complete active fault-tolerant control system enabled the quadrotor to achieve any position even after the complete loss of one rotor.

Recently, multicopter (more than four or six) and tilting propellers can provide novel solutions and control approaches, and also achieve an actuator redundancy in the system in case of failure of one or more motors.

1.3 Modelling

In this section, the mathematical models of both the quadrotor and the birotor are introduced.

1.3.1 Quadrotor

Let $\Sigma_i - \{x_i, y_i, z_i\}$ and $\Sigma_b - \{x_b, y_b, z_b\}$ be the inertial world-fixed frame and the body frame placed at the center of the UAV, respectively. To clarify the notation, the main terms are resumed in the following.

$R_b \in SO(3)$	rotation matrix from Σ_b to Σ_i
$\eta_b = \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}^T \in \mathbb{R}^3$	roll, pitch, and yaw angles, respectively, representing a minimal orientation representation
$\omega_b^b \in \mathbb{R}^3$	angular velocity of Σ_b with respect to Σ_i expressed in Σ_b
$m \in \mathbb{R}$	mass of the vehicle
$I_b = \mathrm{diag}\{I_x, I_y, I_z\} \in \mathbb{R}^{3\times 3}$	inertia matrix with respect to Σ_b
$g \in \mathbb{R}$	gravity acceleration
$p_b = \begin{bmatrix}x & y & z\end{bmatrix}^T \in \mathbb{R}^3$	position of Σ_b in Σ_i
$u \in \mathbb{R}^+$	the total thrust perpendicular to the propellers rotation plane
$\tau_b^b = \begin{bmatrix}\tau_\phi & \tau_\theta & \tau_\psi\end{bmatrix}^T \in \mathbb{R}^3$	torque around the axes of Σ_b expressed in the same frame
$I_n \in \mathbb{R}^{n\times n}$	identity matrix of proper dimensions
$e_3 = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T \in \mathbb{R}^3$	definition of the vertical axis vector in Σ_i
$F_p = \mathrm{diag}\{F_{px}, F_{py}, F_{pz}\} \in \mathbb{R}^{3\times 3}$	linear velocity air drag matrix
$F_o = \mathrm{diag}\{F_{ox}, F_{oy}, F_{oz}\} \in \mathbb{R}^{3\times 3}$	angular velocity air drag matrix

Neglecting the gyroscopic torques, due to the combination of the UAV rotation and the propellers, the equations of motion for a UAV can be written as

$$m\ddot{p}_b = mg - uRe_3 - F_p\dot{p}_b, \tag{1.1a}$$

$$I_b\dot{\omega}_b^b = -\omega_b^b \times I_b\omega_b^b - F_o\omega_b + \tau_b^b, \tag{1.1b}$$

$$\dot{R}_b = R_bS(\omega_b^b), \tag{1.1c}$$

where × represents the cross-product operator, $S(\cdot) \in \mathbb{R}^{3\times 3}$ is the skew-symmetric matrix. The configuration of the UAV is thus defined by the position p_b and the attitude expressed by the rotation matrix R_b, that lies within $SO(3) = \{R \in \mathbb{R}^{3\times 3} : R^TR = I_3, \det(R) = 1\}$.

The dynamic model (1.1) has the linear configuration expressed with respect to Σ_i, while the angular configuration is expressed with respect to Σ_b. For control purposes, neglecting air drag, the equations of motion can be both expressed in the inertial frame as

$$m\ddot{p}_b = mg - uRe_3, \tag{1.2a}$$

$$M\ddot{\eta}_b = -C\dot{\eta}_b + Q^T\tau_b^b, \tag{1.2b}$$

with $M = Q^TI_bQ \in \mathbb{R}^{3\times 3}$ the positive defnite mass matrix, $C = Q^TS(Q\dot{\eta}_b)I_bQ + Q^TI_B\dot{Q} \in \mathbb{R}^{3\times 3}$ the Coriolis matrix, and

$$Q = \begin{bmatrix}1 & 0 & -s_\theta \\ 0 & c_\phi & c_\theta s_\phi \\ 0 & -s_\phi & c_\theta c_\phi\end{bmatrix} \in \mathbb{R}^{3\times 3},$$

a suitable transformation matrix such that $\omega_b^b = Q\dot{\eta}_b$. Notice the use of notation s_α and c_α for the $\sin(\alpha)$ and $\cos(\alpha)$, respectively.

The dynamic models (1.1) and (1.2) can be referred to most of the available UAVs. The control inputs are generalized as the total thrust u and the torques τ_b^b in the body frame. In this chapter, a particular UAV like the quadrotor is addressed. Looking at Fig. 1.1, this vehicle has four rotors and propellers located in the same plane in a cross or X configuration symmetrical to the center of mass. Neglecting the dynamics of the rotors and the propellers, it is assumed that the torque generated by each propeller is directly proportional to the thrust. Notice that the rotation clockwise of propellers 1 and 3, and counter-clockwise of 2 and 4, generate a thrust always directed in the opposite direction of z_b-axis, but along it. These common assumptions yield the following equations between the control input and the propeller speed $\omega_i \in \mathbb{R}$, with $i = 1, \ldots, 4$,

$$u = \rho_u(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2), \tag{1.3a}$$

$$\tau_\phi = l\rho_u(\omega_2^2 - \omega_4^2), \tag{1.3b}$$

$$\tau_\theta = l\rho_u(\omega_3^2 - \omega_1^2), \tag{1.3c}$$

$$\tau_\psi = \rho_c(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2), \tag{1.3d}$$

with $l > 0$ the distance between each propeller and the quadrotor center of mass, $\rho_u > 0$ and $\rho_c > 0$ two aerodynamic parameters.

1.3.2 Birotor

This chapter proposes the following technique to deal with motor failure. Suppose that a motor is either completely broken or losing power. The choice carried out here is to completely turn off both such a damaged motor and the symmetric one to the center of mass, even though if this last is correctly working. Without loss of generality, with reference to Fig. 1.1, suppose that the motor 2 is damaged. Then, it is decided to turn off also motor 4. Therefore, $\omega_2 = \omega_4 = 0$ yields $\tau_\phi = 0$ from (1.3b). The resulting configuration is a birotor with fixed propellers. The equations of motion are again either (1.1) or (1.2), while the computation of the propellers speed in (1.3) changes into

$$u = \rho_u(\omega_1^2 + \omega_3^2), \tag{1.4a}$$

$$\tau_\phi = 0, \tag{1.4b}$$

$$\tau_\theta = l\rho_u(\omega_3^2 - \omega_1^2), \tag{1.4c}$$

$$\tau_\psi = \rho_c(\omega_1^2 + \omega_3^2). \tag{1.4d}$$

If motor 4 is damaged, the situation is identical by turning off motor 2. Instead, suppose that either motor 1 or 3 is damaged, then motor 3 or 1 is turned off, respectively. In this case, the computation of the propellers speed in (1.3) changes into

$$\begin{aligned}
u &= \rho_u(\omega_2^2+\omega_4^2),\\
\tau_\phi &= l\rho_u(\omega_2^2-\omega_4^2),\\
\tau_\theta &= 0,\\
\tau_\psi &= -\rho_c(\omega_2^2+\omega_4^2),
\end{aligned}$$

Without loss of generality, in the remainder of the chapter, only the case with motor 2 and 4 turned is considered. Notice that the case of two defective motors but not aligned on the same quadrotor axis is out of the scope of this work.

An analysis of (1.4) reveals that τ_ψ cannot be freely controlled since it is impossible to change its sign. On the other hand, it is possible to control the total thrust u and the pitch torque τ_θ independently. Folding (1.4a) into (1.4d) yields

$$\tau_\psi = \bar{\tau}_\psi = \frac{\rho_c}{\rho_u}u, \tag{1.5}$$

which is the spinning torque of the birotor around the z_b-axis of Σ_b and depending on the current total thrust scaled by some aerodynamic parameters. The effect is a continuous rotation of the birotor around its vertical axis.

1.4 Control design

This section addresses two control solutions to deal with birotor control. In both of them is demonstrated that, apart from the yaw angle that is uncontrollable as seen above, the birotor can reach any position in the Cartesian space. Despite the uncontrollability of the yaw angle, the device can be safely landed in a predetermined location following a desired emergency landing trajectory.

The two control techniques are namely a PID-based approach and a backstepping procedure. Since this chapter is a tutorial to guide the reader step-by-step in the implementation of the control laws, the proofs of stability of each of the control schemes mentioned above are not here reported. The interested reader can found them in [7] and [8], respectively.

In order to implement the control schemes, the following assumptions must be introduced

- **Assumption 1.** The birotor configuration is always within the configuration space defined by $\mathscr{Q} = \{p_b \in \mathbb{R}^3, \eta_b \in \mathbb{R}^3 : \theta \neq \pi/2 + k\pi, \phi \neq \pi/2 + k\pi, k \in \mathbb{Z}\}$.
- **Assumption 2.** The planned linear acceleration $\ddot{p}_d$ is bounded.

While the second assumption can be easily guaranteed by a suitable planner, the first assumption usually depends on initial conditions of the birotor once the controller is activated. Notice that the desired quantities are denoted with the subscript d.

1.4.1 PID control scheme

The design of the PID-based control scheme starts from the following control law

$$\tau_\theta = \left(\frac{I_y c_\phi + I_y s_\phi^2}{c_\phi} \right) \bar{\tau}_\theta + \alpha_1, \tag{1.6}$$

where $\bar{\tau}_\theta \in \mathbb{R}$ is a virtual control input, and

$$\begin{aligned}
\alpha_1 &= I_y s_\phi \frac{\bar{\tau}_\psi - \alpha_2}{I_z c_\phi} + \alpha_3, \\
\alpha_2 &= -I_x \dot{\phi} \dot{\theta} c_\phi + I_y \dot{\phi} \dot{\theta} c_\phi - I_z \dot{\phi} \dot{\theta} c_\phi + I_x \dot{\psi}^2 c_\theta s_\phi s_\theta \\
&\quad - I_y \dot{\psi}^2 c_\theta s_\phi s_\theta - I_x \dot{\phi} \dot{\psi} c_\theta s_\phi + I_y \dot{\phi} \dot{\psi} c_\theta s_\phi \\
&\quad - I_z \dot{\phi} \dot{\psi} c_\theta s_\phi + I_x \dot{\psi} \dot{\theta} c_\phi s_\theta - I_y \dot{\psi} \dot{\theta} c_\phi s_\theta - I_z \dot{\psi} \dot{\theta} c_\phi s_\theta, \\
\alpha_3 &= -I_x \dot{\phi} \dot{\theta} s_\phi - I_y \dot{\phi} \dot{\theta} s_\phi + I_z \dot{\phi} \dot{\theta} s_\phi - I_x \dot{\psi}^2 c_\phi c_\theta s_\theta \\
&\quad + I_z \dot{\psi}^2 c_\phi c_\theta s_\theta + I_x \dot{\phi} \dot{\psi} c_\phi c_\theta + I_y \dot{\phi} \dot{\psi} c_\phi c_\theta \\
&\quad - I_z \dot{\phi} \dot{\psi} c_\phi c_\theta + I_x \dot{\psi} \dot{\theta} s_\phi s_\theta - I_y \dot{\psi} \dot{\theta} s_\phi s_\theta - I_z \dot{\psi} \dot{\theta} s_\phi s_\theta .
\end{aligned}$$

Substituting (1.6) into (1.2b) and considering (1.4b) and (1.5) yield

$$\ddot{\theta} = \bar{\tau}_\theta, \tag{1.7}$$

meaning that it is possible to entirely control the pitch angle through $\bar{\tau}_\theta$. Let $\theta_d \in \mathbb{R}$ the desired pitch angle. A simple PD controller can be designed for the pitch angle as follows

$$\bar{\tau}_\theta = \ddot{\theta}_d + k_{d,\theta} \dot{e}_\theta + k_{p,\theta} e_\theta, \tag{1.8}$$

with $e_\theta = \theta_d - \theta \in \mathbb{R}$, $\dot{e}_\theta = \dot{\theta}_d - \dot{\theta} \in \mathbb{R}$, $\ddot{e}_\theta = \ddot{\theta}_d - \ddot{\theta} \in \mathbb{R}$, and $k_{p,\theta} > 0$ and $k_{d,\theta} > 0$ two gains.

Let $\mu = \begin{bmatrix} \mu_x & \mu_y & \mu_z \end{bmatrix}^T \in \mathbb{R}^3$ be the acceleration of the birotor, expressed in Σ_i, where the magnitude is the desired thrust u_d produced by the remaining propellers, while the orientation $R_d \in SO(3)$ is given by the desired pitch and the current measured roll and yaw as follows

$$R_d = \begin{bmatrix} c_{\theta_d} c_\psi & s_\phi s_{\theta_d} c_\psi - c_\phi s_\psi & c_\phi s_{\theta_d} c_\psi + s_\phi s_\psi \\ c_{\theta_d} s_\psi & s_p his_{\theta_d} s_\psi + c_\phi c_\psi & c_\phi s_{\theta_d} s_\psi - s_\phi c_\psi \\ -s_{\theta_d} & s_\phi c_{\theta_d} & c_\phi c_{\theta_d} \end{bmatrix}.$$

The definition of such an acceleration is thus given by

$$\mu = \frac{u_d}{m} R_d e_3 + g e_3. \tag{1.9}$$

Substituting $\theta = \theta_d - e_\theta$ into (1.2a) and taking into account the definition in (1.9) yield

$$\ddot{p}_b = \mu + \frac{u}{m} \delta, \tag{1.10}$$

where $\delta = \begin{bmatrix}\delta_x & \delta_y & \delta_z\end{bmatrix}^T \in \mathbb{R}^3$ is the following interconnection vector

$$\delta_x = 2\cos(\phi)\cos(\psi)\sin(e_\theta/2)\cos(\theta_d - e_\theta/2), \tag{1.11a}$$

$$\delta_y = 2\cos(\phi)\sin(\psi)\sin(e_\theta/2)\cos(\theta_d - e_\theta/2), \tag{1.11b}$$

$$\delta_z = -2\cos(\phi)\sin(e_\theta/2)\sin(\theta_d - e_\theta/2). \tag{1.11c}$$

The acceleration μ can be designed as follows

$$\mu = \ddot{p}_d + K_p e_p + K_d \dot{e}_p, \tag{1.12}$$

where $K_p, K_d \in \mathbb{R}^{3\times 3}$ are positive definite gain matrices, $p_d, \dot{p}_d, \ddot{p}_d \in \mathbb{R}^3$ represent the desired position trajectory for the birotor, and $e_p = p_d - p_b$, $\dot{e}_p = \dot{p}_d - \dot{p}_b$, $\ddot{e}_p = \ddot{p}_d - \ddot{p}_b \in \mathbb{R}^3$ are the related tracking errors. Substituting (1.12) in (1.10) and (1.8) in (1.7) yields the following closed-loop equations

$$\ddot{e}_p + K_d \dot{e}_p + K_p e_p = -\frac{u}{m}\delta, \tag{1.13a}$$

$$\ddot{e}_\theta + k_{d,\theta}\dot{e}_\theta + k_{p,\theta} e_\theta = 0. \tag{1.13b}$$

It remains to derive how to retrieve the desired total thrust u_d of the birotor and the desired pitch angle θ_d. Once the errors e_p and $\dot{e}_p$ are computed, knowing the planned acceleration $\ddot{p}_d$, the virtual control input μ can be computed from (1.12). The desired total thrust and the desired pitch angle can be computed by inverting (1.9) as follows

$$u_d = m\sqrt{\mu_x^2 + \mu_y^2 + (\mu_z - g)^2}, \tag{1.14a}$$

$$\theta_d = \tan^{-1}\left(\frac{\mu_x c_\psi + \mu_y s_\psi}{\mu_z - g}\right), \tag{1.14b}$$

where ψ is measured from the IMU. A second-order low-pass digital filter can be employed to reduce noise and compute both first and second derivatives of θ_d, to compute the pitch tracking errors e_θ and $\dot{e}_\theta$ in turn. The control input τ_θ is then computed from (1.6), with $\bar{\tau}_\theta$ obtained as from (1.8). Finally, the propeller inputs of the birotor can be computed inverting (1.4a) and (1.4c). The proposed controller is drawn in Figure 1.2.

Notice that two PD controller are employed in (1.8) and (1.12). As noticed in [28], an integral action can be added without destroying the stability properties. The PD gains play like programmable stiffness and damping parameters, giving a physical interpretation to their choice.

Similar considerations can be made in case of damage to motor 1 and/or motor 3. In such a case, the desired angle becomes the roll that can be computed by inverting $\mu = -\frac{u}{m}R_{dr}e_3 + ge_3$, with $R_c \in SO(3)$ the equivalent of R_d by substituting θ_d with the measured θ and ϕ with the desired $\phi_d = \sin^{-1}\left(\frac{m}{u_d}(\mu_y c_\psi - \mu_x s_\psi)\right)$.

1.4.2 Backstepping control scheme

The derivation of the proposed backstepping control scheme starts from the definition of the virtual acceleration μ as in (1.9). The desired total thrust and the desired

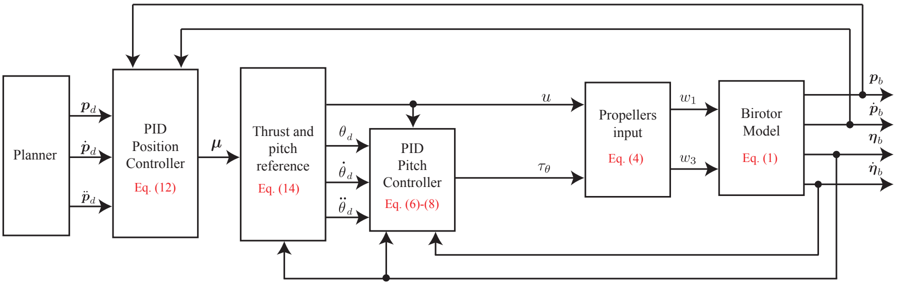


Figure 1.2 Block scheme of the PID-based control architecture. In red, the corresponding equations in the paper related to each block.

pitch can be then retrieved from (1.14). Again, the goal is to design the vector μ suitably. The approach is different from the previous control scheme. First, an altitude control is derived. Afterward, the planar position control is designed. Finally, the control to achieve the desired pitch is addressed.

The following PD controller can be employed to control the altitude of the UAV

$$\mu_z = \ddot{z}_d + k_{d,z}\dot{e}_z + k_{p,z}e_z, \tag{1.15}$$

with $z_d \in \mathbb{R}$ the desired altitude in Σ_i, $e_z = z_d - z$, $\dot{e}_z = \dot{z}_d - \dot{z}$, $k_{p,z}, k_{d,z} > 0$ some gains.

For the planar control, it is worth noticing that the considered birotor, since it is an underactuated system continuously spinning around the vertical axis of Σ_b, can only rotate around the y_b-axis of Σ_b. Therefore, the projection of the birotor vertical axis z_b into the x_iy_i-plane of Σ_i is a rotating vector with rate $\dot{\psi}$. Figure 1.3 sketches the above concept. The following kinematic constraint can be thus introduced

$$\begin{bmatrix}\dot{x} & \dot{y}\end{bmatrix}^T = \beta\upsilon, \tag{1.16}$$

with $\beta = \begin{bmatrix}\cos\psi & \sin\psi\end{bmatrix}^T$, and $\upsilon \in \mathbb{R}$ the magnitude of the projection vector. The term υ is employed as a virtual input to design the desired planar velocities in Σ_i. The desired planar accelerations of the vector μ are obtained by differentiating (1.16) with respect to time as follows

$$\begin{bmatrix}\mu_x & \mu_y\end{bmatrix}^T = \dot{\beta}\upsilon + \beta\alpha, \tag{1.17}$$

with $\alpha = \dot{\upsilon} \in \mathbb{R}$. The virtual inputs υ and α must be now designed to zero the error in the horizontal plane of Σ_i. Consider a regulation problem with $\dot{x}_d = \dot{y}_d = 0$. The planar errors are defined as $e_x = x_d - x$ and $e_y = y_d - y$. By deriving these two quantities and taking into account (1.16) yield

$$\begin{bmatrix}\dot{e}_x & \dot{e}_y\end{bmatrix}^T = -\beta\upsilon, \tag{1.18a}$$

$$\dot{\upsilon} = \alpha. \tag{1.18b}$$

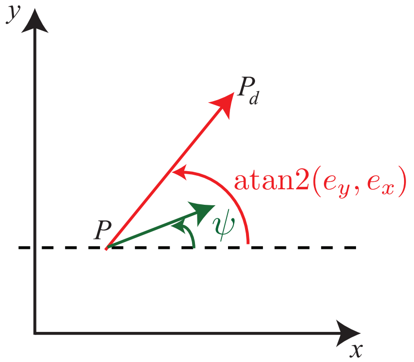


Figure 1.3 *The $x_i y_i$-plane of Σ_i is here represented. The point P represents the current position p_b of the birotor projected in such a plane. The point P_d represents the desired position p_d of the birotor on the same plane. The green vector is the current heading vector of the birotor in the $x_i y_i$-plane of Σ_i, which continuously rotates as the yaw angle. The red vector is the planar error creating an angle of* $\mathrm{atan2}(e_y, e_x)$ *with respect to the x_i-axis.*

Through a backstepping approach [8, 29], the following virtual input can be designed to provide at least marginal stability in zeroing the planar errors

$$\upsilon = k_\upsilon \sqrt{e_x^2 + e_y^2} \cos(\mathrm{atan2}(e_y, e_x) - \psi), \tag{1.19}$$

with $k_\upsilon > 0$, and

$$\begin{aligned}\alpha = &(\dot{e}_x(k_\upsilon + k_\alpha) + e_x)c_\psi + (\dot{e}_y(k_\upsilon + k_\alpha) + e_y)s_\psi \\ &+ k_\upsilon k_\alpha \sqrt{e_x^2 + e_y^2} \cos(\mathrm{atan2}(e_y, e_x) - \psi(t)),\end{aligned} \tag{1.20}$$

with $k_\alpha > 0$.

Finally, the low-level attitude control for the birtor is designed as for the PID-based controller. Then, control law (1.6) is firstly introduced to get (1.7). The virtual control input $\bar{\tau}_\theta$ can be designed as in (1.8) to entirely control the pitch angle.

The derived backstepping controller is represented in Fig. 1.4. To recap, as first, the position errors components e_x, e_y, e_z are computed, as well as the related time derivatives $\dot{e}_x, \dot{e}_y, \dot{e}_z$. Knowing the feedforward acceleration $\ddot{z}_d$, it is possible to compute the control input μ_z from (1.15). Taking into account both (1.19) and (1.20), the other two components of the virtual control input μ are retrieved from (1.17). The desired total thrust u_d and the pitch angle θ_d are then computed as in (1.14). A second-order low-pass digital filter is employed to reduce noise and compute both first and second derivatives of θ_d. Afterwards, the pitch tracking errors e_θ and $\dot{e}_\theta$ are computed. The control input τ_θ is then computed from (1.6), with $\overline{\tau}_\theta$ obtained from (1.8). Finally, the propellers speeds for the birotor are given by (1.4a) and (1.4c). An inte-

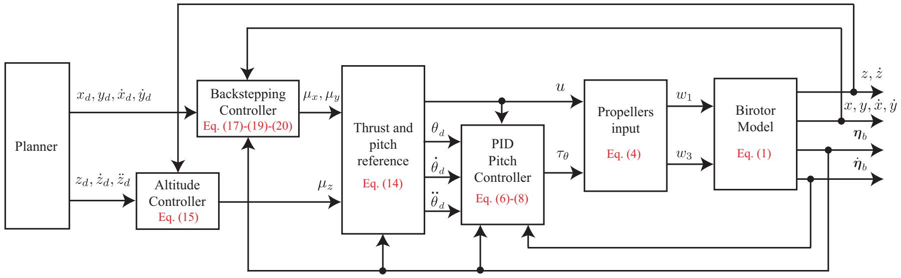


Figure 1.4 Block scheme of the backstepping-based control architecture. In red, the corresponding equations in the paper related to each block.

gral action might be added in (1.15) and (1.8) to increase tracking accuracy without destroying stability properties.

1.5 Numerical simulations

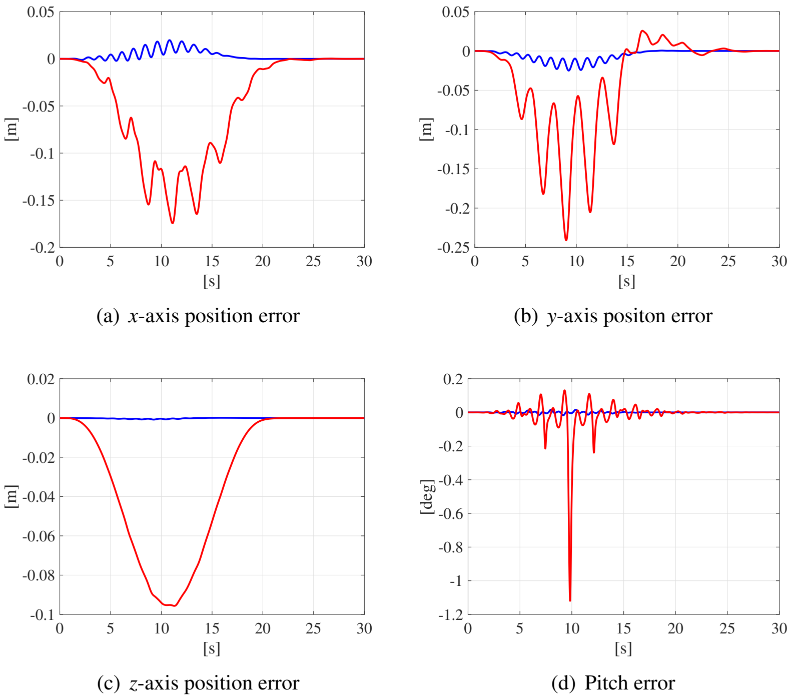
In this section, the two above described controllers are compared in a case study simulating an emergency landing. As stated in Section 1.2, the employed controllers do not consider the switch from a nominal operating condition, in which all the motors are active, to the emergency landing procedure. Only the control of the birotor is evaluated through numerical simulations. The comparison between the results obtained with the two described control methodologies is thus compared. The next subsection introduces the parameters employed in the two controllers and the choices made for the simulation as well as the software environment. Afterward, the case study is presented and analyzed.

1.5.1 Description

The proposed control methodologies are tested in a Matlab/Simulink environment running on a standard PC (2,6 GHz Intel Core i5, 8 GB 1600 MHz DDR3). The two schemes depicted in Fig. 1.2 and Fig. 1.4 have been implemented.

To simulate the dynamic of the birotor, the more accurate dynamic model (1.1) including air drag is implemented. The dynamic parameters of the employed UAV are taken from a real Asctech Pelican quadrotor [30]. In detail, the following parameters are considered: $m = 1.25$ kg, $I_b = \text{diag}\{3.4, 3.4, 4.7\}$ kgm^2, $l = 0.21$ m, $\rho_u = 1.8 \cdot 10^{-5}$ Ns^2/rad^2, $\rho_c = 8 \cdot 10^{-7}$ Nms^2/rad^2. Besides, to consider saturations of the actuators, a maximum speed of about $\omega_i = 940$ rad/s (about 6000 rpm) is considered. It was verified in practice that the birotor at steady-state has a constant rotation of about 7 rad/s around its z_b-axis. Therefore, the friction coefficients in (1.1) are set to $F_p = \text{diag}\{1.25, 1.25, 5\}$ and $F_o = \text{diag}\{1, 1, 2\}$. Finally, the controllers are discretized at 100 Hz, to further test the robustness of the proposed controllers.

The gains for the PID-based controller are experimentally tuned as $K_p = \text{diag}\{6.25, 16, 25\}$, $K_d = \text{diag}\{6, 8, 10\}$, $k_{p,\theta} = 900$, and $k_{d,\theta} = 60$.



(a) x-axis position error

(b) y-axis positon error

(c) z-axis position error

(d) Pitch error

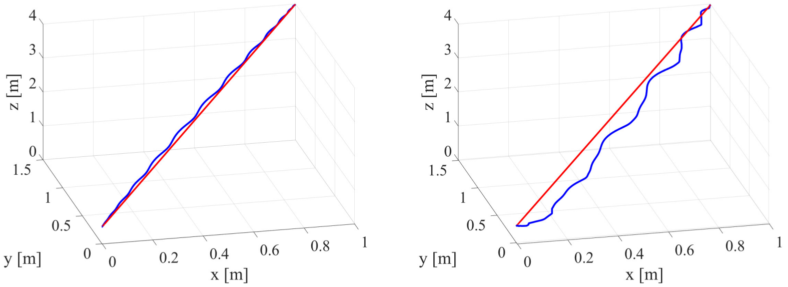
Figure 1.5 Position and pitch error for the considered case study. In blue, the results obtained with the PID-based controller. In red, the results obtained with the backstepping controller.

The gains for the backstepping controller are experimentally tuned as $k_v = 2$, $k_\alpha = 0.1$, $k_{p,\theta} = 144$, $k_{d,\theta} = 26.4$, $k_{p,z} = 16$, and $k_{d,z} = 8$.

In the following, a comparative case study is addressed. Given the desired trajectory, the two controllers are compared through a numerical simulation starting from the same initial conditions.

1.5.2 Case study

The considered case study is the tracking of a possible landing trajectory. The birotor starts from the position $p_b = \begin{bmatrix} 1 & 1.5 & 4 \end{bmatrix}^T$ m, which is the same desired initial position for the landing trajectory at $t = 0$ s, with $t > 0$ the time variable, and an initial yaw velocity of 3 rad/s. The desired goal position for the birotor is $\begin{bmatrix} 0 & 0 & 0.5 \end{bmatrix}^T$ m with respect to Σ_i at $t = 20$ s. The initial and final linear velocities and accelerations are zero. A seventh-order polynomial is employed to generate the desired trajectory and guarantee the conditions defined above. The birotor desired trajectory stays for other 10 s in a steady-state condition equal to the final one.



(a) PID-based

(b) Backstepping

Figure 1.6 Tridimensional Cartesian path followed by the birotor with the two implemented controllers. On the left, the path followed by employing the PID-based controller. On the right, the path followed by employing the backstepping controller. In red, the desired path. In blu, the path followed by the birotor for the given controller.

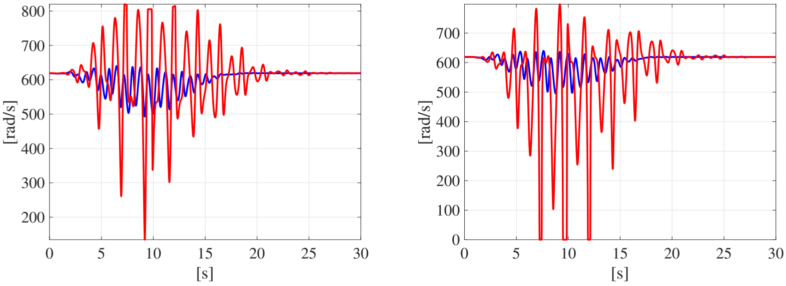
The time histories of the position error are depicted in Fig. 1.5. The pictures show the position error for each of the Σ_i axes for the two implemented controllers. Both approaches successfully achieve the sought task. However, the backstepping approach needs more time to converge to the desired trajectory. The backstepping approach has a maximum error of about 0.15 m along the x-axis, 0.25 m along the y-axis, and 0.09 m along the vertical axis of Σ_i. The tridimensional Cartesian path followed by the birotor in both cases is depicted in Fig. 1.6. The pitch error of the inner loop of both controllers is depicted in Fig. 1.5(d). Also in this case, the backstepping controller shows a more significant error of around 1 deg, which worsens the overall performance of the task.

The time histories of the commanded velocities for the birotor active propellers are illustrated in Fig. 1.7. The propeller speeds do not saturate in both cases. Nevertheless, the behavior related to the backstepping controller appears overwrought.

It is worth recalling that the results show in any case robustness of both the control designs. The friction is indeed not included in the mathematical model employed for the control designs, even though it brings benefits to the stability analysis of the closed-loop system. Besides, the controllers are discretized at 100 Hz as in the common practice. The final idea is that both the controllers fulfill the sought task. However, at least for the analysed case study, the PID-based control seems to be more robust and employing less power for the motors.

1.6 Conclusion

This chapter provided a tutorial solution, based on [7] and [8], to solve the tracking control problem for a birotor UAV. Such a birotor configuration could be employed in those cases where a quadrotor UAV has one damaged propeller. Two control



(a) Motor 1

(b) Motor 3

Figure 1.7 *Commanded velocities of the active birotor propellers. In blue, the results obtained with the PID-based controller. In red, the results obtained with the backstepping controller.*

schemes have been proposed to deal with the contemplated problem: namely, a PID-based control scheme and a backstepping approach. It was shown that, with both approaches, the birotor could reach any position in the Cartesian space, and thus it could follow a safe emergency trajectory.

Acknowledgement

The research leading to these results has been partially supported by the HYFLIERS project (Horizon 2020 Grant Agreement No. 779411) and the AERIAL-CORE project (Horizon 2020 Grant Agreement No. 871479). The authors are the sole responsible of its contents.

References

[1] Amazon Prime Air;. http://www.amazon.com/b?node=8037720011.

[2] The White House;. https://obamawhitehouse.archives.gov/the-press-office/2016/08/02/fact-sheet-new-commitments-accelerate-safe-integration-unmanned-aircraf

[3] IEEE Spectrum;. http://spectrum.ieee.org/aerospace/aviation/flying-selfie-bots-tagalong-video-drones-are-here/.

[4] Ruggiero F, Lippiello V, Ollero A. Aerial manipulation: A literature review. IEEE Robotics and Automation Letters. 20018;3(3):1957–1964.

[5] Blanke M, Staroswiecki M, Wu NE. Concepts and methods in fault-tolerant control. In: 2001 American Control Conference. vol. 4. Arlington, VA; 2001. p. 2606–2620.

[6] Zhang Y, Jiang J. Bibliographical review on reconfigurable fault-tolerant control systems. Annual Reviews in Control. 2008;32(2):229–252.

[7] Lippiello V, Ruggiero F, Serra D. Emergency landing for a quadrotor in case of a propeller failure: A PID approach. In: 2014 IEEE RSJ International

Conference on Robotics and Automation. Chicago, IL, USA; 2014. p. 4782–4788.

[8] Lippiello V, Ruggiero F, Serra D. Emergency landing for a quadrotor in case of a propeller failure: A backstepping approach. In: 12th IEEE International Symposium on Safety, Security and Rescue Robots. Toyako-Cho, Hokkaido, J; 2014. .

[9] Freddi A, Longhi S, Monteriu A. Actuator fault detection system for a mini-quadrotor. In: 2010 IEEE International Symposium on Industrial Electronics. Bari, I; 2010. p. 2055–2060.

[10] Sharifi F, Mirzaei M, Gordon BW, et al. Fault tolerant control of a quadrotor UAV using sliding mode control. In: 2010 Conference on Control and Fault-Tolerant Systems. Nice, F; 2010. p. 239–244.

[11] Sadeghzadeh I, Mehta A, Chamseddine A, et al. Active Fault Tolerant Control of a quadrotor UAV based on gain scheduled PID control. In: 25th IEEE Canadian Conference on Electrical and Computer Engineering. Montreal, QC; 2012. p. 1–4.

[12] Saied M, Lussier B, Fantoni I, et al. Passive Fault-Tolerant Control of an Octorotor using Super-Twisting Algorithm: Theory and Experiments. In: 2016 3rd Conference on Control and Fault-Tolerant Systems. Barcelona, S; 2016. p. 361–366.

[13] Ranjbaran M, Khorasani K. Fault recovery of an under-actuated quadrotor aerial vehicle. In: 49th IEEE Conference on Decision and Control. Atlanta, GA; 2010. p. 4385–4392.

[14] Khebbache H, Sait B, Yacef F, et al. Robust Stabilization of a Quadrotor Aerial Vehicle in Presence of Actuator Faults. International Journal of Information Technology, Control and Automation. 2012;2(2):1–13.

[15] Zhang Y, Chamseddine A. Fault Tolerant Flight Control Techniques with Application to a Quadrotor UAV Testbed. In: Lombaerts T, editor. Automatic Flight Control Systems - Latest Developments. InTech; 2012. p. 119–150.

[16] Freddi A, Lanzon A, Longhi S. A feedback linearization approach to fault tolerance in quadrotor vehicles. In: 18th IFAC World Congress. Milano, I; 2011. p. 5413–5418.

[17] Kataoka Y, Sekiguchi K, Sampei M. Nonlinear control and model analysis of trirotor UAV model. In: 18th IFAC World Congress. Milano, I; 2011. p. 10391–10396.

[18] Lanzon A, Freddi A, Longhi S. Flight Control of a Quadrotor Vehicle Subsequent to a Rotor Failure. Journal of Guidance, Control, and Dynamics. 2014;37(2):580–591.

[19] Mueller MW, D'Andrea R. Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers. In: 2014 IEEE International Conference on Robotics and Automation. Hong Kong, C; 2014. p. 45–52.

[20] Vey D, Lunze J. Structural reconfigurability analysis of multirotor UAVs after actuator failures. In: 2015 IEEE 54th Annual Conference on Decision and Control. Osaka, J; 2015. p. 5097–5104.

[21] Sun S, Sijbers L, Wang X, et al. High-Speed Flight of Quadrotor Despite Loss of Single Rotor. IEEE Robotics and Automation Letters. 2018;4(4):3201–3207.

[22] Avant T, Lee U, Katona B, et al. Dynamics, Hover Configurations, and Rotor Failure Restabilization of a Morphing Quadrotor. In: 2018 Annual American Control Conference. Wisconsin Center, Milwaukee, USA; 2018. p. 4855–4862.

[23] Morozov YV. Emergency Control of a Quadrocopter in Case of Failure of Two Symmetric Propellers. Automation and Remote Control. 2018;79(3):92–110.

[24] Merheb AR, Noura H, Bateman F. Emergency Control of AR Drone Quadrotor UAV Suffering a Total Loss of One Rotor. IEEE/ASME Transactions on Mechatronics. 2017;22(2):961–971.

[25] Barghandan S, Badamchizadeh MA, Jahed-Motlagh MR. Improved adaptive fuzzy sliding mode controller for robust fault tolerant of a Quadrotor. International Journal of Control, Automation and Systems. 2017;15(1):427–441.

[26] Saied M, Lussier B, Fantoni I, et al. Fault Diagnosis and Fault-Tolerant Control of an Octorotor UAV using motors speeds measurements. IFAC-PapersOnLine. 2017;50(1):5263–5268.

[27] Lu P, van Kampen EJ. Active fault-tolerant control for quadrotors subjected to a complete rotor failure. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems. Hamburg, D; 2015. p. 4698–4703.

[28] Nonami K, Kendoul F, Suzuki S, et al. Autonomous Flying Robots. Unmanned Aerial Vehicles and Micro Aerial Vehicles. Berlin Heidelberg, D: Springer-Verlag; 2010.

[29] Khalil HK. Nonlinear systems. Upper Saddle River, NJ: Prentice Hall; 2002.

[30] Ruggiero F, Cacace J, Sadeghian H, et al. Impedance control of VTOL UAVs with a momentum-based external generalized forces estimator. In: 2014 IEEE International Conference on Robotics and Automation. Hong Kong, C; 2014. p. 2093–2099.