



Article

An Efficient Design and Implementation of a Quadrotor Unmanned Aerial Vehicle Using Quaternion-Based Estimator

Eva H. Dulf ^{1,2} , Mihnea Saila ³, Cristina I. Muresan ^{1,*}  and Liviu C. Miclea ¹

¹ Department of Automation, Faculty of Automation and Computer Science, Technical University of Cluj-Napoca, Memorandumului Str. 28, 400014 Cluj-Napoca, Romania; Eva.Dulf@aut.utcluj.ro (E.H.D.); Liviu.Miclea@aut.utcluj.ro (L.C.M.)

² Physiological Controls Research Center, Óbuda University, H-1034 Budapest, Hungary

³ Huisman Equipment BV, Admiraal Trompstraat, 3115HH Schiedam, The Netherlands; msaila@huisman-nl.com

* Correspondence: Cristina.Muresan@aut.utcluj.ro

Received: 21 August 2020; Accepted: 14 October 2020; Published: 18 October 2020



Abstract: The main goal of the research is to design a low-cost, performing quadrotor unmaned aerial vehicle (UAV) system. Because of low cost limits, the performance must be ensured by other ways. The present proposal is a quaternion-based estimator used in the control loop. In order to make the proposed solution easy to be reproduced by the reader, step-by-step instructions are given, including component choices, design, and implementation. Throughout the article, detailed description of the system model is given. The efficacy of the suggested quaternion-based predictive control is evaluated by extended experimental results.

Keywords: unmanned aerial vehicle (UAV); quaternion-based estimator; low-cost design

1. Introduction

Unmanned aerial vehicles (UAV)s have fascinated many researchers and engineers, as they turned out to be accessible in a large variety of applications, not just for costly military operations. Nowadays, UAVs have a broad range of applications, such as: image capturing, aerial recording, military operations, operations in hard-to-reach areas, etc., [1–7]. Along with the development of wireless communications, the control of UAVs has become extremely precise, robust, and even predictive. New research results in the design of UAVs and new application areas include advanced and complex control techniques like robust and adaptive control, algorithms for different flight conditions, fault tolerance, disturbance rejection, etc., [8–13]. All these methods increase the complexity and the cost of the UAV. Because of the extremely alert technological progress registered in the past two decades, the global industrialization and the minimization of the costs of electronic components, countless researchers have shown a high interest in the development of various devices helpful for the society.

One key issue regarding the control of UAVs resides in the estimation of their position. Various methods have been proposed. An efficient method is presented in [14], both from the point of view of the algorithm's performance and from the point of view of using the processing capacity of a microcontroller. Several estimation algorithms are compared in [15], with the results showing that the extended Kalman algorithm is slower in terms of processing time than Madgwick algorithm [15].

The approach of estimating the pitch and roll coordinates presented in [16] constitutes a reference that fits perfectly in the context of the present paper. For the application of the algorithm proposed in the paper, a method of fusion of the data received from an accelerometer, a gyroscope, and a magnetometer was used to estimate, accurately, the position of a flight apparatus. A combination

of the extension of the classical Kalman filtering algorithm and the sequential geometric correction is proposed, completely eliminating the magnetic distortions captured by the sensor. In addition, the paper offers a clear and concise comparison between certain popular approaches to the problem of estimating the coordinates of a flight apparatus and the method proposed by the author. Both the improvements and the problems that arise in the implementation of this method are presented.

From all the knowledge resulting from this state-of-art, it can be concluded that UAVs can be made at a relatively lower cost. Simple transducers can be used, as long as this is compensated by a high performance and optimized estimation algorithm. The authors already designed a cheap and easy to use two-rotor equipment, in order to be multiplied for laboratory works [17,18].

Quaternion framework is widely used today to avoid locks and to ensure better computational efficiency [19,20]. The field of application is large, ranging from mechanical systems [21] and medical robots [22] to neural networks [23] and human activities and postures recognition [24], all research papers reporting remarkable results. Quaternions are also used in UAV control with great success. In [25], the authors developed a nonlinear state space model using the quaternion and angular velocity as state variables, which simplifies the system dynamics. The main focus of the research is directed toward the feedback linearization of the model. The simulation results are presented solely for the attitude stabilization task of the quadcopter. A quaternion representation of the attitude of a quadrotor is also used in [26], where various control methods are discussed and compared, such as the PD, LQR, and backstepping methods. Various case scenarios are discussed including noisy data, actuator restrictions, external disturbances. The attitude control of a quadrotor is designed in a quaternion framework in [27], to avoid gimbal lock and for better computational efficiency. The controllers are tuned based on third-order sliding mode control, with a low-pass filter to reduce chattering and a disturbance observer to cover disturbance estimation problems. To ensure the robustness, a disturbance compensation term is also included in the control law. The simulation results show that the proposed method is efficient. In [28], two variants of adaptive state space controllers for attitude stabilization and self-tuning of a quadrotor are proposed. The effectiveness of the approach is demonstrated through simulations that use a quaternion-based nonlinear dynamic model of a quadrotor. A quaternion representation of the attitude of a quadrotor is also used in [29], where a quaternion-energy-based control law is defined as a Lyapunov function, with the control laws described with unit quaternions and their axis-angle representation. Various simulation and experimental results are presented. Unit quaternions are also used in [30] to describe a simple yet complete dynamic model for the rotational and translational dynamics of unmanned aerial vehicles, whereas dual quaternions are explained and used for robotic systems with multiple rotations and translations. An unmanned aerial vehicle described with unit quaternions is presented in [31]. In this case, a quaternion-passivity-based control is derived. The experimental results and numerical simulations validate the results. Intermediary quaternions are used in the design of a backstepping control technique with integral properties in [32]. Compared to classical quaternions, the proposed approach has also the advantage that one specific orientation corresponds to only one intermediary quaternion, which helps coping with the unwinding phenomenon. Numerical simulations, as well as experimental tests, are presented. The robustness of the algorithm is also tested during the numerical simulations only. In [33], a quaternion-based guidance law is proposed which feeds into an attitude control system based on a PD+ control law. A quaternion control scheme for a quadrotor is also proposed in [34]. An attitude control algorithm is developed to stabilize the vehicle's heading and an additional position control law for stabilization of the vehicle in all states. In this case also, numerical and experimental results are presented to validate the approach. An advanced control scheme, also based on quaternions, is presented in [35] for the attitude control of a quadrotor. Here, both the model and the proportional squared control algorithm are implemented in the quaternion space. Extended simulation results are included to demonstrate the efficacy of the suggested novel approach. Quaternions for attitude control are also used in [36], where a quaternion multiplicative formula is proposed to obtain the change of the attitude angle of a quadrotor. Only some practical solutions are presented.

Other recent significant results in UAV control includes more complex structures or calculus. In [37], a control structure based on a hierarchical scheme is proposed, consisting of an energy-based control to stabilize the vehicle translational dynamics and to attenuate the payload oscillation and a nonlinear state feedback controller based on a linear matrix inequality (LMI) to control the quadrotor rotational dynamics. The authors of [38] propose a neuroadaptive integral robust controller, while [39] discusses the dynamic motion planning and control of an UAV using Direct and the Second Method of Lyapunov. An interesting, but complex solution is proposed in [40], where the dynamic system is divided in two subsystems driven by the translational and the rotational dynamics, based on a linear parameter-varying model.

All these approaches have in common the use of advanced control algorithms, with the major drawback of requiring expensive hardware for implementation purposes. Thus, the main objective of the present work is to design and implement a low-cost, easy to use quadrotor UAV, accessible for any user. A quaternion-based estimator is proposed, similarly to existing research studies. However, in terms of the proposed control strategy, the classical PID controller is used, instead of advanced control algorithms. In this way, the implementation of the control strategy is simplified, which triggers the possibility of using low-cost devices for measurement and control. The final control structure includes four controllers, one for each direction of movement. Step by step design and implementation details are presented in order to be easily reproduced by the reader. Using simulation and experimental data, the proposed method is validated. The results show that similar closed loop performance can be achieved using our proposed approach, compared to other more advanced control strategies. The major advantage is that using our proposed method, these results are achieved using a low-cost UAV with a simple, yet efficient control strategy. The novelty of this work consists, thus, in a quaternion-based estimator and classical Proportional-Integral-Derivative (PID) control strategy, implemented using low-cost microcontroller and sensors. For the proposed remote control, performances are imposed in terms of rejecting a moderate range of disturbances and filtering sensor noisy signals.

The rest of this paper is organized as follows. The materials and methods used are presented in the next section. The resulting quadrotor UAV prototype, along with experimental data, is detailed in Section 3. Finally, conclusions are presented in Section 4.

2. Materials and Methods

From construction point of view, the system includes the following elements: plastic skeleton for the flight apparatus; support for the electrical circuit of the remote control; electrical circuits; ATmega32U4 and ATmega 328 microcontrollers; four DC motors; four electronic velocity controllers; four propellers; two wireless remote communication modules and a position detection module.

The main aspects of the flight apparatus described in this work are defined by: the number of engines, the position of the support arms, the mass and the center of gravity of the whole assembly. The arms are mounted in "X," to allow easy change of direction, and the center of gravity is fixed at the intersection of the axes of the arms. The change of direction is facilitated by the control of the angular velocity of the engines. The motors are positioned as follows: two motors on one diagonal are rotated in the same direction, while the remaining two motors on the other diagonal rotate in the opposite direction. Viewed as a whole, the system is composed of a four-arm flight apparatus mounted in "X" and a remote control that provides references to the control circuit located on the quadrotor. They communicate via the UART protocol, using two RF transmission and reception modules. Two-way data exchanges are made between the quadrotor and the remote control, so both items send data and await receipt.

Regarding the mechanical design of the system, a variety of computer-aided design environments could be used to create 3D drawings and model the parts necessary for the physical realization of the system. In the present work AutoCAD and SolidWorks were adopted. In addition, Ultimaker Cura—a G code generator and a 3D printer that could correctly interpret the generated code—was operated to create the remote control.

After choosing the components, measuring their dimensions and making the connections, an electrical scheme could be conceived. The present practice used a CAD/CAM environment provided by Autodesk, called Eagle.

The device is designed in such way that the center of gravity coincides with the geometric one, also serving as the center of the coordinate system attached to the quadrotor. This coordinate system describes the relative movements of the flight apparatus to a fixed coordinate system, with an axis perpendicular to the earth's surface. The other two axes of the fixed coordinate system can be chosen so as to coincide with cardinal points whose axes are perpendicular (for example north-east or south-west).

Like any aerial vehicle, this system has also six degrees of freedom, meaning three movements of translation and three of rotation. All these movement possibilities are strongly dependent on the velocity and implicitly the angular velocity of the four engines. Therefore, depending on these aspects, the following kinetic forces and moments developed and applied to the quadrotor can be distinguished: the altitude advance, the gyroscopic effect, the yaw moment, the pitch moment, the roll moment and, of course, the force of gravitational attraction. The increase or decrease of altitude is possible by simultaneously increasing or decreasing the velocity of all engines. In order to maintain a constant altitude it is necessary to drive the engines at the same velocity, each developing the same angular velocity. Unlike the altitude movement, the kinetic yaw, pitch, and roll momenta are obtained by differentiating the engine velocity. The yaw moment, or rotation around the vertical axis, is obtained by simultaneously increasing the velocity of two motors rotating in the same direction. Depending on the chosen engine group, the flight apparatus will rotate clockwise or trigonometrically.

2.1. Quaternion-Based Estimator

In order to obtain the orientation angles and to facilitate the calculus, two representations can be used, namely: Euler angles and quaternions.

Quaternions are used to express the orientation of a coordinate system to a reference system [41]. Given an angle of rotation Ψ about the axis \hat{r} , an orientation of the coordinate system B can be represented with respect to the system A as follows [41]:

$${}^A_B\hat{q} = [q_1 q_2 q_3 q_4] = \left[\cos \frac{\Psi}{2} - \hat{r}_x \sin \frac{\Psi}{2} - \hat{r}_y \sin \frac{\Psi}{2} - \hat{r}_z \sin \frac{\Psi}{2} \right] \tag{1}$$

The terms $\hat{r}_x, \hat{r}_y, \hat{r}_z$ represent the components of the unity vector \hat{r} of the reference system A.

A very important advantage presented by this angle expression method is that the product of two quaternions ${}^C_D\hat{q}$ and ${}^D_E\hat{q}$ represents the orientation of the system E with respect to the reference system C.

Moreover, the orientation described by a quaternion ${}^A_B\hat{q} = [q_1 \ q_2 \ q_3 \ q_4]$ can be expressed by the rotation matrix A_BR , representing the rotation of the coordinate system B with respect to the reference system A. The dependence between the quaternion terms and the rotation matrix is presented in Equation (2) [16,20].

$${}^A_BR = \begin{bmatrix} 2q_1^2 + 2q_2^2 - 1 & 2(q_1q_4 + q_2q_3) & 2(q_2q_4 - q_1q_3) \\ 2(q_2q_3 - q_1q_4) & 2q_1^2 + 2q_3^2 - 1 & 2(q_1q_2 + q_3q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_3q_4 - q_1q_2) & 2q_1^2 + 2q_4^2 - 1 \end{bmatrix} \tag{2}$$

Although, from a computational point of view, obtaining orientation using quaternions is more efficient, they are hard to interpret physically. Thus, in order to have a clear picture of the real movement, the orientations expressed by quaternions are transformed into representations using Euler angles. To carry out these transformations, Equations (3)–(5) could be used [21].

$$\Psi = \text{atan}(2q_2q_3 - 2q_1q_4, 2q_1^2 + 2q_2^2 - 1) \tag{3}$$

$$\theta = \arcsin(2q_1q_3 + 2q_2q_4) \tag{4}$$

$$\Phi = \text{atan}(2q_3q_4 - 2q_1q_2, 2q_1^2 + 2q_4^2 - 1) \tag{5}$$

In order to obtain the real values of the angles, a sensor with 9 degrees of freedom was used, consisting of an accelerometer, a gyroscope, and a magnetometer. The sensor used is from the MPU9250 family and communicates with the microcontroller via the I2C interface, at a frequency of 400 kHz. The I2C protocol is a very popular data transmission protocol, due to the multitude of advantages it presents [42]. For data filtering and estimating the orientation of the aerial vehicle, the quaternion representation described above was used. With a physical interpretation much closer to reality, the data provided by the gyroscope are filtered and estimated easily. Thus, the angular positions on the X, Y, and Z axes are arranged in a vector W as described in Equation (6). In addition to these three elements, on the first position in the vector is inserted the term 0 in order to be able to perform quaternion products.

$$W = [0 \ w_x \ w_y \ w_z] \tag{6}$$

$$\overset{\text{Ref}}{\text{Sensor}} \dot{q} = \overset{\text{Ref}}{\text{Sensor}} q \otimes W \tag{7}$$

With the angular position arranged in the vector W it is possible to compute the orientation change of the coordinate system given by the earth to the coordinate system attached to the UAV. This calculus is represented in Equation (7).

Where the term $\overset{\text{Ref}}{\text{Sensor}} q$ represents the current orientation of the coordinate system given by the earth to the quadrotor coordinate system.

In order to obtain an orientation of the coordinate system attached to the quadrotor with respect to the reference one, at a time t it is necessary to perform the mathematical operations detailed in Equations (8) and (9).

$$\overset{\text{Sensor}}{\text{Ref}} \dot{q}_{\text{gyro},k} = \frac{1}{2} \overset{\text{Sensor}}{\text{Ref}} q_{\text{est}, k-1} \otimes W \tag{8}$$

$$\overset{\text{Sensor}}{\text{Ref}} q_{\text{gyro},k} = \overset{\text{Sensor}}{\text{Ref}} q_{\text{est}, k-1} + \overset{\text{Sensor}}{\text{Ref}} \dot{q}_{\text{gyro},k} T_s \tag{9}$$

where T_s represents the sampling time, and $t = k \cdot T_s$.

Because of the nature of the data from the accelerometer, an optimization problem can be formulated in which the orientation of the sensor $\overset{\text{Sensor}}{\text{Ref}} \hat{q}$ and, implicitly of the flight system, is given by minimizing the difference between the orientation of the reference system of the earth d_{ref} and that of the sensor, d_{sensor} . The objective function to be minimized of is described by Equations (10) and (11), with the components detailed in (12)–(14).

$$\text{of}(\overset{\text{Sensor}}{\text{Ref}} \hat{q}, d_{\text{ref}}, d_{\text{sensor}}) = \overset{\text{Sensor}}{\text{Ref}} \hat{q}^* \otimes d_{\text{ref}} \otimes \overset{\text{Sensor}}{\text{Ref}} \hat{q} - d_{\text{sensor}} \tag{10}$$

$$\text{of}(\overset{\text{Sensor}}{\text{Ref}} \hat{q}, d_{\text{ref}}, d_{\text{sensor}}) = \begin{bmatrix} 2d_{rx}(\frac{1}{2} - q_3^2 - q_4^2) + 2d_{ry}(q_1q_4 + q_2q_3) + 2d_{rz}(q_2q_4 - q_1q_3) - d_{sx} \\ 2d_{rx}(q_2q_3 - q_1q_4) + 2d_{ry}(\frac{1}{2} - q_2^2 - q_4^2) + 2d_{rz}(q_1q_2 + q_3q_4) - d_{sy} \\ 2d_{rx}(q_1q_3 + q_2q_4) + 2d_{ry}(q_3q_4 - q_1q_2) + 2d_{rz}(\frac{1}{2} - q_2^2 - q_3^2) - d_{sz} \end{bmatrix} \tag{11}$$

$$\overset{\text{Sensor}}{\text{Ref}} \hat{q} = [q_1 \ q_2 \ q_3 \ q_4] \tag{12}$$

$$d_{\text{ref}} = [0 \ d_{rx} \ d_{ry} \ d_{rz}] \tag{13}$$

$$d_{\text{sensor}} = [0 \ d_{sx} \ d_{sy} \ d_{sz}] \tag{14}$$

In order to solve this optimization problem, the conjugate gradient method is used, a simple, efficient method that requires a relatively low computing power [43]. However, the conjugate gradient

method presents a number of disadvantages related to the algorithm step, μ and the initial point $\hat{q}_{Ref}^{Sensor} \hat{q}_0$. Equations (15) and (16) describe the estimation of future orientation $\hat{q}_{Ref}^{Sensor} \hat{q}_{k+1}$.

$$\hat{q}_{Ref}^{Sensor} \hat{q}_{k+1} = \hat{q}_{Ref}^{Sensor} \hat{q}_k - \mu \frac{F}{\|F\|} \tag{15}$$

$$F = \frac{\partial f(\hat{q}_{Ref}^{Sensor}, d_{ref}, d_{sensor})}{\partial (\hat{q}_{Ref}^{Sensor}, d_{ref})} of(\hat{q}_{Ref}^{Sensor}, d_{ref}, d_{sensor}) \tag{16}$$

The general cost function of given in (10) can be simplified to be easy to implement even in a low-cost microcontroller. Because of the fact that by convention gravitational acceleration determines only the Z axis of the reference system, this objective function can be expressed as in Equation (17), while the vectors d_{ref} and d_{sensor} are given in (18) and (19).

$$of(\hat{q}_{Ref}^{Sensor}, d_{ref}, d_{sensor}) = \begin{bmatrix} 2(q_2q_4 - q_1q_3) - d_x \\ 2(q_1q_2 + q_3q_4) - d_y \\ 2(\frac{1}{2} - q_2^2 - q_3^2) - d_z \end{bmatrix} \tag{17}$$

$$d_{ref} = [0 \ 0 \ 0 \ 1] \tag{18}$$

$$d_{sensor} = [0 \ d_x \ d_y \ d_z] \tag{19}$$

The data obtained from the magnetometer will be processed in the same way as the data obtained from the accelerometer, but with a more laborious processing given by the decomposition of the earth's magnetic field in both a component on the X axis and one on the Z axis. To obtain the next orientations $\hat{q}_{Ref}^{Sensor} \hat{q}_{k+1}$, the same conjugate gradient algorithm will be used. Equation (20) describes the objective function, with the terms detailed in (21) and (22), while Equation (23) presents the gradient of the objective function.

$$of(\hat{q}_{Ref}^{Sensor}, m_{ref}, m_{sensor}) = \begin{bmatrix} 2m_{rx}(\frac{1}{2} - q_3^2 - q_4^2) + 2m_{rz}(q_2q_4 - q_1q_3) - m_{sx} \\ 2m_{rx}(q_2q_3 - q_1q_4) + 2m_{rz}(q_1q_2 + q_3q_4) - m_{sy} \\ 2m_{rx}(q_1q_3 + q_2q_4) + 2m_{rz}(\frac{1}{2} - q_2^2 - q_3^2) - m_{sz} \end{bmatrix} \tag{20}$$

$$m_{ref} = [0 \ m_{rx} \ 0 \ m_{rz}] \tag{21}$$

$$m_{sensor} = [0 \ m_{sx} \ m_{sy} \ m_{sz}] \tag{22}$$

$$F = \frac{\partial of(\hat{q}_{Ref}^{Sensor}, m_{ref}, m_{sensor})}{\partial (\hat{q}_{Ref}^{Sensor}, m_{ref})} of(\hat{q}_{Ref}^{Sensor}, m_{ref}, m_{sensor}) \tag{23}$$

In order to obtain both a measurement and an accurate estimation of the orientation of the quadrotor, it is necessary to compose the two objective functions presented in Equations (10) (or the simplified form in (17)) and (20). Also, the gradient of both functions will be used to implement the conjugate gradient algorithm for the combination of functions. The composition will be noted with f_{com} and the gradient of this compound function will be noted by F_{com} . In addition, to make the algorithm more efficient, the step μ will be variable and recomputed at each iteration, as shown in Equation (24). The algorithm and the gradient of the new objective function are presented in Equations (25) and (26):

$$\mu_t = \alpha \| \hat{q}_{Ref}^{Actual} - \hat{q}_{gyro,k} \|_{T_S} \tag{24}$$

$$\hat{q}_{Ref}^{Sensor} q_{com,k} = \hat{q}_{Ref}^{Sensor} \hat{q}_{est,k-1} - \mu_t \frac{F_{com}}{\|F_{com}\|} \tag{25}$$

$$F_{com} = \begin{bmatrix} \frac{\partial f(\text{Sensor}_{Ref} \hat{q}_{est,k-1}, d_{\text{Sensor}})}{\partial (\text{Sensor}_{Ref} \hat{q}_{est,k-1})} f(\text{Sensor}_{Ref} \hat{q}_{est,k-1}, d_{\text{Sensor}}) \\ \frac{\partial f(\text{Sensor}_{Ref} \hat{q}_{est,k-1}, m_{ref}, m_{\text{Sensor}})}{\partial (\text{Sensor}_{Ref} \hat{q}_{est,k-1}, m_{ref})} f(\text{Sensor}_{Ref} \hat{q}_{est,k-1}, m_{ref}, m_{\text{Sensor}}) \end{bmatrix} \tag{26}$$

where α is a constant chosen experimentally to minimize the measurements noise from the accelerometer and magnetometer, T_S is the sampling period, $\hat{q}_{gyro,k}^{Actual}$ represents the orientation given by the gyroscope, computed using Equation (8).

Because of the fusion of measurements from the gyroscope, $\text{Sensor}_{Ref} q_{gyro,k}$ and those from the accelerometer and magnetometer $\text{Sensor}_{Ref} q_{com,k}$, a weighted, very accurate estimate is obtained, as presented in Equation (27). The weight P_k will be computed at each iteration based on the step μ_t , a control constant β , and the sampling period T_S , as in (28).

$$\text{Sensor}_{Ref} \hat{q}_{est,k} = P_k \text{Sensor}_{Ref} q_{com,k} + (1 - P_k) \text{Sensor}_{Ref} q_{gyro,k} \tag{27}$$

$$P_k = \frac{\beta}{\frac{\mu_t}{T_S} + \beta} \tag{28}$$

The proposed filter in (27) and (28) ensures an accurate estimation such that $\text{Sensor}_{Ref} \hat{q}_{est,k} \rightarrow \text{Sensor}_{Ref} q_k$ as $k \rightarrow \infty$. This can be easily proved using the classical Lyapunov function.

At each iteration, after obtaining the current estimate, Equations (3)–(5) are used to express the Euler angle orientation, which gives a much easier to understand perspective on the movement of the quadrotor.

After obtaining the orientation angles and converting them from quaternions to Euler angles, at each iteration the rotation matrices $R_x(\Phi)$, $R_y(\theta)$, and $R_z(\Psi)$ will be constructed. With these matrices, the rotation matrix of the entire system $R_{xyz}(\Psi, \theta, \Phi)$ is computed, as described in Equations (29)–(32).

$$R_x(\Phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\Phi) & -s(\Phi) \\ 0 & s(\Phi) & c(\Phi) \end{bmatrix} \tag{29}$$

$$R_y(\theta) = \begin{bmatrix} c(\theta) & 0 & s(\theta) \\ 0 & 1 & 0 \\ -s(\theta) & 0 & c(\theta) \end{bmatrix} \tag{30}$$

$$R_z(\Psi) = \begin{bmatrix} c(\Psi) & -s(\Psi) & 0 \\ s(\Psi) & c(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{31}$$

$$R_{xyz}(\Phi, \theta, \Psi) = R_x(\Psi) \cdot R_y(\theta) \cdot R_z(\Phi) = \begin{bmatrix} c(\theta)c(\Psi) & s(\Phi)s(\theta)c(\Psi) - c(\Phi)s(\Psi) & c(\Phi)s(\theta)c(\Psi) + s(\Phi)s(\Psi) \\ c(\theta)s(\Psi) & s(\Phi)s(\theta)s(\Psi) + c(\Phi)c(\Psi) & c(\Phi)s(\theta)s(\Psi) - s(\Phi)c(\Psi) \\ -s(\theta) & s(\Phi)c(\theta) & c(\Phi)c(\theta) \end{bmatrix} \tag{32}$$

where $c(\Psi) = \cos(\Psi)$, $s(\Psi) = \sin(\Psi)$, $c(\theta) = \cos(\theta)$, $s(\theta) = \sin(\theta)$, $c(\Phi) = \cos(\Phi)$, $s(\Phi) = \sin(\Phi)$.

2.2. Quadrotor Kinematic and Dynamic Model

In order to establish an efficient mathematical model, as close as possible to the reality, which ensures greater system controllability, it is necessary to use the Equations of Newton classical mechanics and of Euler for angular motions. It is also necessary to take into account both the relative movements of the fixed coordinate system (in this case, the earth), as well as the relative dynamics of the coordinate system attached to the quadrotor. Thus, two vectors, P_p and P_a , will be used, described

by Equations (33) and (34). P_p is the vector of the linear and angular positions of the flight system relative to earth, while P_a is the vector of the linear and angular velocities of the quadrotor.

$$P_p = [x \ y \ z \ \Phi \ \theta \ \Psi]^T \tag{33}$$

$$P_a = [u \ v \ w \ p \ q \ r]^T \tag{34}$$

To link these two vectors, the rotation matrix $R_{xyz}(\Phi, \theta, \Psi)$ and a matrix of angular velocity transformations, $T_V(\Phi, \theta)$ is used, derived from the inverse of the derivative of the Euler angle change rate. Thus in Equations (35)–(40) the dependencies between the vectors P_p and P_a are detailed.

$$v_p = [\dot{x} \ \dot{y} \ \dot{z}]^T \tag{35}$$

$$\omega_p = [\dot{\Phi} \ \dot{\theta} \ \dot{\Psi}]^T \tag{36}$$

$$v_a = [u \ v \ w]^T \tag{37}$$

$$\omega_a = [p \ q \ r]^T \tag{38}$$

$$v_p = R_{xyz}(\Phi, \theta, \Psi) \cdot v_a \tag{39}$$

$$\omega_p = T_V(\Phi, \theta) \cdot \omega_a \tag{40}$$

The vectors v_p and ω_p are the derivatives of the linear and angular positions of P_p , while v_a and ω_a are the linear and angular velocities of the vector P_a . The matrix of angular velocity transformations $T_V(\Phi, \theta)$ is constructed as described by Equation (41).

$$T_V(\Phi, \theta) = \begin{bmatrix} 1 & c(\Phi) \tan(\theta) & c(\Phi) \tan(\theta) \\ 0 & c(\Phi) & -s(\Phi) \\ 0 & \frac{s(\Phi)}{c(\theta)} & \frac{c(\Phi)}{c(\theta)} \end{bmatrix} \tag{41}$$

Performing the multiplications leads to the kinematic model:

$$\begin{cases} \dot{x} = uc(\Psi)c(\theta) - v[c(\Phi)s(\Psi) - c(\Psi)s(\Phi)s(\theta)] + w[s(\Phi)s(\Psi) + c(\Phi)s(\Psi)s(\theta)] \\ \dot{y} = uc(\theta)c(\Psi) + v[c(\Phi)c(\Psi) + s(\Phi)s(\theta)s(\Psi)] - w[c(\Psi)s(\Phi) - c(\Phi)s(\Psi)s(\theta)] \\ \dot{z} = -uc(\theta) + vc(\theta)s(\Phi) + wc(\Phi)c(\theta) \\ \dot{\Phi} = p + qs(\Phi)t(\theta) + rc(\Phi)t(\theta) \\ \dot{\theta} = qc(\Phi) - rs(\Phi) \\ \dot{\Psi} = q\frac{s(\Phi)}{c(\theta)} + r\frac{c(\Phi)}{c(\theta)} \end{cases} \tag{42}$$

From Newton’s laws, the forces acting on the quadrotor can be determined. These will be denoted with vector F_a and calculated as described in Equations (43) and (44).

$$F_a = m_q(\omega_a \times v_a + \dot{v}_a) \tag{43}$$

$$F_a = [f_x \ f_y \ f_z]^T \tag{44}$$

where m_q denotes the mass of the quadrotor, “ \times ” is the vector product of the linear and angular velocity relative to the quadrotor coordinate system, while \dot{v}_a is the linear acceleration.

Similar to the computation of the force, the angular velocity applied to the quadrotor will also be determined from Euler’s Equation. These velocities will be noted with M_a , and are strongly dependent on the inertia matrix I , as it is presented in Equations (45)–(47).

$$M_a = I \cdot \dot{\omega}_a + \omega_a \times (I \cdot \omega_a) \tag{45}$$

$$M_a = \begin{bmatrix} m_x & m_y & m_z \end{bmatrix}^T \tag{46}$$

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{47}$$

Combining Equations (44) and (46), the dynamic model of the quadrotor relative to its own coordinate system can be expressed as:

$$\begin{cases} f_x = m(\dot{u} + qw - rv) \\ f_y = m(\dot{v} - pw + ru) \\ f_z = m(\dot{w} + pv - qu) \\ m_x = \dot{p}I_x - qrI_y + qrI_z \\ m_y = prI_x + \dot{q}I_y - prI_z \\ m_z = -pqI_x + pqI_y + \dot{r}I_z \end{cases} \tag{48}$$

The forces and velocities described above can also be expressed by Equations (49) and (50).

$$F_a = m_q g R_{xyz}(\Phi, \theta, \Psi)^T \cdot \hat{e}_z - f_p \cdot \hat{e}_3 + f_v \tag{49}$$

$$M_a = \tau_a - g_a + \tau_v \tag{50}$$

In the above expression m_q means the total mass of the quadrotor, g is the gravitational acceleration, \hat{e}_z and \hat{e}_3 are the unit vectors on the Z axis of the reference coordinate system, respectively of the coordinate system attached to the quadrotor. The element f_p represents the total propulsion force developed by the engines, and $f_v = \begin{bmatrix} f_{vx} & f_{vy} & f_{vz} \end{bmatrix}^T$ represents the disturbances or forces that are opposed to the rotation of each engine, caused by air currents. τ_a represents the angular velocity generated by the velocity differences of the four motors, while τ_v stands for the angular velocities produced by air currents on each motor, detailed in Equations (51) and (52). g_a are the gyroscope moments caused by the combined velocities of the four motors. Given the fact that the inertia of the motors is negligible compared to the developed force, the gyroscopic moments may be neglected from Equation (50).

$$\tau_a = \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^T \tag{51}$$

$$\tau_v = \begin{bmatrix} \tau_{vx} & \tau_{vy} & \tau_{vz} \end{bmatrix}^T \tag{52}$$

Replacing these new Equations for forces and velocities, a new dynamic model is obtained:

$$\begin{cases} -m_q g s(\theta) + f_{vx} = m_q(\dot{u} + qw - rv) \\ m_q g c(\theta) s(\Phi) + f_{vy} = m_q(\dot{v} - pw + ru) \\ m_q g c(\theta) c(\Phi) + f_{vz} - f_p = m_q(\dot{w} + pv - qu) \\ \tau_x + \tau_{vx} = \dot{p}I_x - qrI_y + qrI_z \\ \tau_y + \tau_{vy} = prI_x + \dot{q}I_y - prI_z \\ \tau_z + \tau_{vz} = -pqI_x + pqI_y + \dot{r}I_z \end{cases} \tag{53}$$

In order to control the quadrotor, the dependence between the propulsion force f_p , velocity τ_a , and the motor's angular velocities $\Omega_a = [\Omega_1 \ \Omega_2 \ \Omega_3 \ \Omega_4]$ needs to be introduced in the model, using Equation (54).

$$\begin{cases} f_p = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ \tau_x = b \cdot l(\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2) \\ \tau_y = b \cdot l(\Omega_1^2 + \Omega_4^2 - \Omega_2^2 - \Omega_3^2) \\ \tau_z = d(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2) \end{cases} \quad (54)$$

where b is a propulsion coefficient and d is the aerodynamic resistance coefficient. The term l represents the distance from the center of gravity of the quadrotor to the center of rotation of the engine. This term is equal for all four arms of the quadrotor. In addition, replacing the terms obtained from Equation (54) in (53) leads to a new expression of the dynamic model, given by:

$$\begin{cases} -m_q g s(\theta) + f_{vx} = m_q(\dot{u} + qw - rv) \\ m_q g c(\theta)s(\Phi) + f_{vy} = m_q(\dot{v} - pw + ru) \\ m_q g c(\theta)c(\Phi) + f_{vz} - b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) = m_q(\dot{w} + pv - qu) \\ b \cdot l(\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2) + \tau_{vx} = \dot{p}I_x - qrI_y + qrI_z \\ b \cdot l(\Omega_1^2 + \Omega_4^2 - \Omega_2^2 - \Omega_3^2) + \tau_{vy} = prI_x + \dot{q}I_y - prI_z \\ d(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2) + \tau_{vz} = -pqI_x + pqI_y + \dot{r}I_z \end{cases} \quad (55)$$

This model will be used as predictor in the control structure.

2.3. Quadrotor State Space Model Used for Controller Design

The next step consists in the model design in a state space form, in order to easily apply the controller design methods. Therefore, the state vector \mathbb{X} , the input vector u , and the output vector y will be chosen as it is presented below:

$$\mathbb{X} = [\Phi \ \theta \ \Psi \ p \ q \ r \ u \ v \ w \ x \ y \ z]^T \quad (56)$$

$$u = [f_p \ \tau_x \ \tau_y \ \tau_z]^T \quad (57)$$

$$y = [\Phi \ \theta \ \Psi \ f_p]^T \quad (58)$$

Using this state vector \mathbb{X} and Equations (42) and (55), one can determine the derivative of this state vector, $\dot{\mathbb{X}}$:

$$\begin{cases} \dot{\Phi} = p + qs(\Phi)t(\theta) + rc(\Phi)t(\theta) \\ \dot{\theta} = qc(\Phi) - rs(\Phi) \\ \dot{\Psi} = q \frac{s(\Phi)}{c(\theta)} + r \frac{c(\Phi)}{c(\theta)} \\ \dot{p} = \frac{I_y - I_z}{I_x} qr + \frac{\tau_x + \tau_{vx}}{I_x} \\ \dot{q} = \frac{I_z - I_x}{I_y} pr + \frac{\tau_y + \tau_{vy}}{I_y} \\ \dot{r} = \frac{I_x - I_y}{I_z} pq + \frac{\tau_z + \tau_{vz}}{I_z} \\ \dot{u} = rv - qw - gs(\theta) + \frac{f_{vx}}{m} \\ \dot{v} = pw - ru + gs(\Phi)c(\theta) + \frac{f_{vy}}{m} \\ \dot{w} = qu - pv + gc(\Phi)c(\theta) + \frac{f_{vz} - f_p}{m} \\ \dot{x} = uc(\Psi)c(\theta) - v[c(\Phi)s(\Psi) - c(\Psi)s(\Phi)s(\theta)] + w[s(\Phi)s(\Psi) + c(\Phi)c(\Psi)s(\theta)] \\ \dot{y} = uc(\theta)s(\Psi) + v[c(\Phi)c(\Psi) + s(\Phi)s(\theta)s(\Psi)] - w[c(\Psi)s(\Phi) - c(\Phi)s(\Psi)s(\theta)] \\ \dot{z} = -us(\theta) + vc(\theta)s(\Phi) + wc(\Phi)c(\theta) \end{cases} \quad (59)$$

As can be seen from Equation (59), the system is strongly nonlinear, presenting major problems in the design of a control system based on models. In order to be linearized, a Jacobian matrix is used, at certain chosen equilibrium points. Given that it is desired that in the absence of a command the system be maintained at a fixed point at a predetermined altitude, the equilibrium points are chosen as described below.

$$\mathbb{X}_e = \left[0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ x_e \ y_e \ z_e \right]^T \tag{60}$$

$$\mathbf{u}_e = \left[m_q \cdot g \ 0 \ 0 \ 0 \right]^T \tag{61}$$

where $g = 9.8 \frac{m}{s^2}$ is the gravitational acceleration and m_q is the total mass of the quadrotor.

Also, since the trigonometric dependencies between the system states do not disappear even after the linearization by the Jacobian method, a preliminary simplification is made. Thus, in order to eliminate the trigonometric functions from the system model, all the values of the sine functions are approximated with their argument, respectively the cosine functions with 1. The approximate model, resulting from the simplification, has the form as described in (62).

$$\begin{cases} \dot{\Phi} = p + q\Phi\theta + r\theta \\ \dot{\theta} = q - r\Phi \\ \dot{\Psi} = q\Phi + r \\ \dot{p} = \frac{I_y - I_z}{I_x} qr + \frac{\tau_x + \tau_{vx}}{I_x} \\ \dot{q} = \frac{I_z - I_x}{I_y} pr + \frac{\tau_y + \tau_{vy}}{I_y} \\ \dot{r} = \frac{I_x - I_y}{I_z} pq + \frac{\tau_z + \tau_{vz}}{I_z} \\ \dot{u} = rv - qw - g\theta + \frac{f_{vx}}{m_q} \\ \dot{v} = pw - ru + g\Phi + \frac{f_{vy}}{m_q} \\ \dot{w} = qu - pv + g + \frac{f_{vz} - f_p}{m_q} \\ \dot{x} = u - v(\Psi - \Phi\theta) + w(\Phi\Psi + \theta) \\ \dot{y} = u\Psi + v(1 + \Phi\theta\Psi) - w(\Phi - \Psi\theta) \\ \dot{z} = -u\theta + v\Phi + w \end{cases} \tag{62}$$

In the state space form, the system is $\dot{\mathbb{X}} = h(\mathbb{X}, \mathbf{u})$. Applying the linearization by the Jacobian matrix method and using the equilibrium points expressed in (60) and (61), the linearized state space system became:

$$\begin{cases} \dot{\mathbb{X}} = A_e \cdot \mathbb{X} + B_e \cdot \mathbf{u} \\ y = C \cdot \mathbb{X} \end{cases} \tag{63}$$

with

$$A_e = \left. \frac{\partial h(\mathbb{X}, \mathbf{u})}{\partial \mathbb{X}} \right|_{\substack{\mathbb{X} = \mathbb{X}_e \\ \mathbf{u} = \mathbf{u}_e}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{64}$$

$$B_e = \frac{\partial h(\mathbb{X}, \mathbf{u})}{\partial \mathbf{u}} \Bigg|_{\substack{\mathbb{X} = \mathbb{X}_e \\ \mathbf{u} = \mathbf{u}_e}} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m_q} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{65}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/m_q & 0 & 0 & 0 \end{bmatrix} \tag{66}$$

The same model can be described as a system of Equations as indicated in (67).

$$\begin{cases} \dot{\Phi} = p \\ \dot{\theta} = q \\ \dot{\Psi} = r \\ \dot{p} = \frac{\tau_x + \tau_{vx}}{I_x} \\ \dot{q} = \frac{\tau_y + \tau_{vy}}{I_y} \\ \dot{r} = \frac{\tau_z + \tau_{vz}}{I_z} \\ \dot{u} = -g\theta + \frac{f_{vx}}{m_q} \\ \dot{v} = g\Phi + \frac{f_{vy}}{m_q} \\ \dot{w} = \frac{f_{vz} - f_p}{m_q} \\ \dot{x} = u \\ \dot{y} = v \\ \dot{z} = w \end{cases} \tag{67}$$

2.4. Controller Design

The controller design method uses the linear model of the system (63). Considering that references for the orientation of the quadrotor and for the altitude of flight will be transmitted from the remote controller, and the system inputs depend on the angular velocities of the four engines, a number of four controllers will be implemented for each direction of movement. Each controller can be designed with any tuning algorithm, ensuring the cancellation of the steady state error and a short settling time. An interesting choice is presented for example in [44]. If advanced controller tuning methods are used, the performances could be increased. The idea of the present work is to implement a very low-cost quadrotor, with the simplest control algorithm, but with results comparable with advanced control methods. With this regard a simple PID controller is designed for each rotor, using the classical root locus method [45]. For this method, given the characteristic polynomial of the closed-loop system, the parameters of controller are chosen depending on the location of the poles of the system. Overshoot, settling time and steady state error cancellation are imposed for each controller. Figure 1 illustrates the block diagram of the control strategy chosen for this quadrotor, with the PID blocks detailed in Figure 2. The proposed feedback control requires feedback signals and disturbance identification. To obtain these signals, a sensor with 9 degrees of freedom, consisting of an accelerometer, a gyroscope,

and a magnetometer is used. Signals from this sensor must be processed because they suffer from noise disturbance and other drawbacks. For example the gyroscope has a flowing bias. This inconvenient is mitigated by the estimator. Both data filtering and estimating the orientation of the aerial vehicle are realized with the quaternion representation of the estimator (27,28). The nonlinear dynamic model (55) is used as predictor in the control structure.

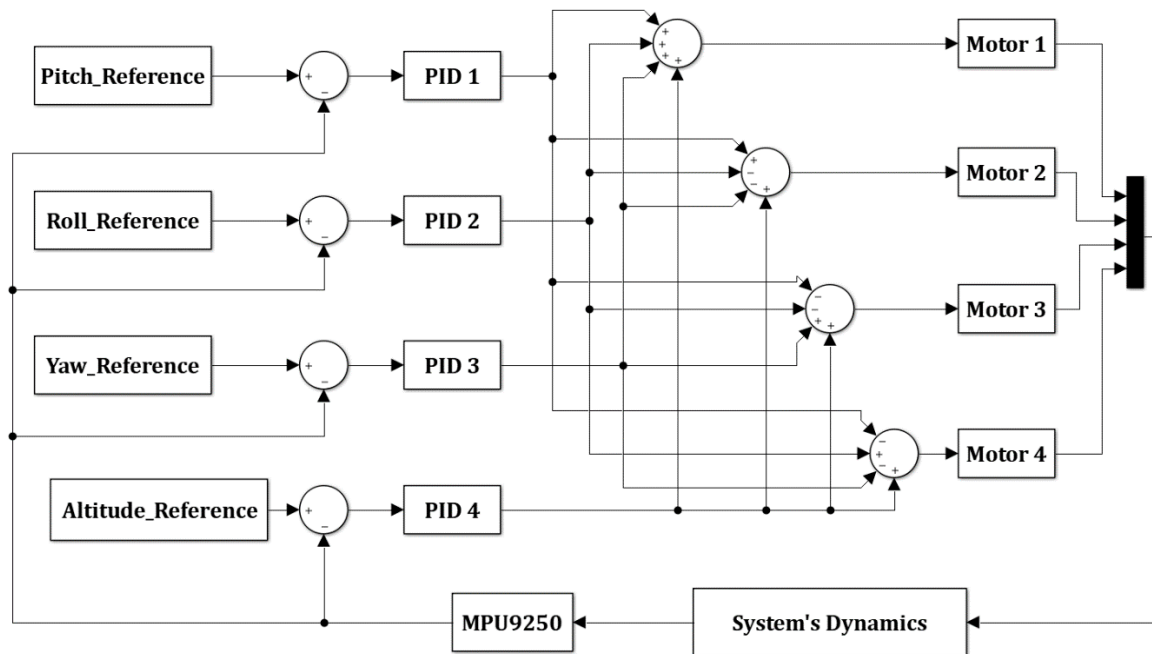


Figure 1. Block diagram of the three-dimensional space orientation control.

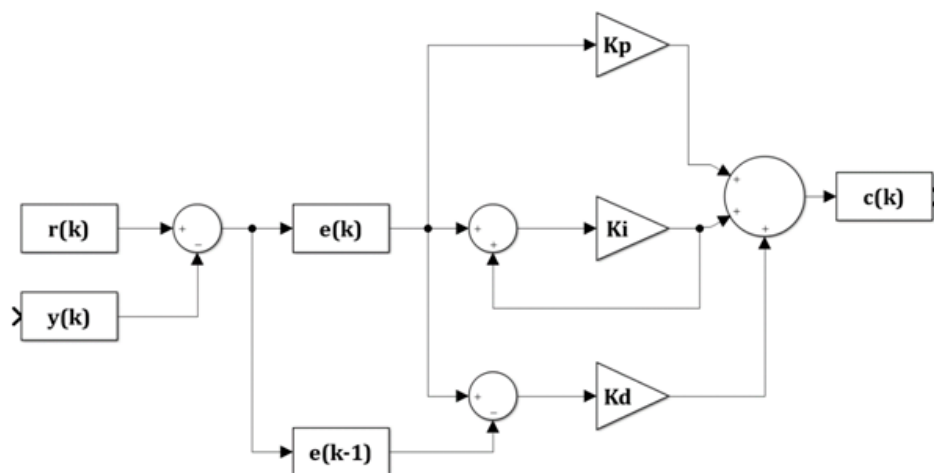


Figure 2. Block diagram of each PID controller.

In Figure 2 the signals are denoted as follows: $r(k)$ is the reference signal at current iteration k ; $c(k)$ represents the control signal at this current iteration k ; $y(k)$ is the output of the system, measured by sensors at iteration k ; $e(k)$ is the error signal at iteration k ; K_p , K_i , and K_d represent the proportionality, integration, and derivation constant, respectively. Regarding the angular velocities of the motors, it is necessary to ensure that they behave according to the control signals received from the angular position and altitude controllers. In view of the microcontroller’s processing capacity and the relatively large dimensions of the program used to obtain the inclination angles and the control law previously determined, four electronic speed control modules (ESCs) will be used to control the angular velocities of the motors.

The verification of the designed controllers was first performed through a numerical simulation. The non-linear model from Equation (55) was used to carry out all simulations. Several simulation scenarios were adjusted in order to set the simulation closer to reality. Furthermore, some restrictions related to the actuators were applied based on real data measurements. The delay of the actuators was implemented because of the use of the electronic speed controller (ESC). Moreover, sensor noise was implemented to the measured feedback signals. The evaluation of the designed controller was done both in disturbance free, constant disturbance, and real disturbance conditions. In each case the quadrotor has to follow the same trajectory, including takes off, flying from point A to B, and rotation around the Z axis. As quality indicators chosen to discuss the efficiency of the proposed algorithm are the steady state position error, overshoot, and settling time. In all cases the proposed simple control structure exhibit very similar behavior to advanced, expensive solutions.

3. Results

Figure 3 presents the resulted low-cost quadrotor UAV. It has four motors controlled by electronic speed controllers rotating as described in Figure 4. Each motor is mounted on a plastic arm, which in turn is attached to the carbon fiber central structure. All pieces were chosen so that the assembly has the lowest weight and, at the same time, to maintain the condition of the center of gravity described in the previous section.



Figure 3. The unmanned aerial vehicle (UAV) prototype.

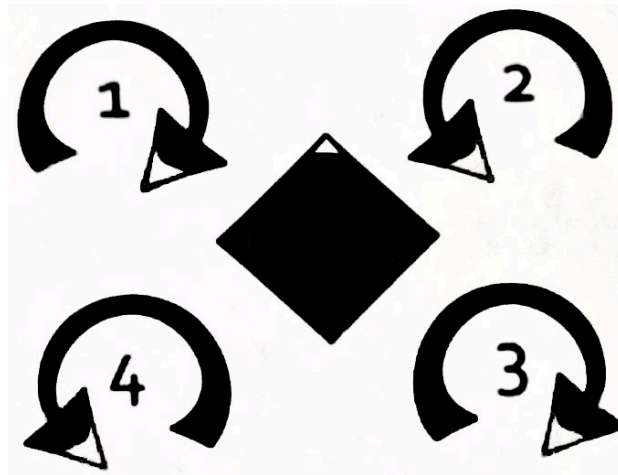


Figure 4. Rotational directions.

In each step of the design, the aim was to understand the functionality of each component of the system and to describe the relationships between them using block diagrams. In this regard Figures 5 and 6 detail the block diagrams of each subsystem, highlighting the type of data provided by/for each element.

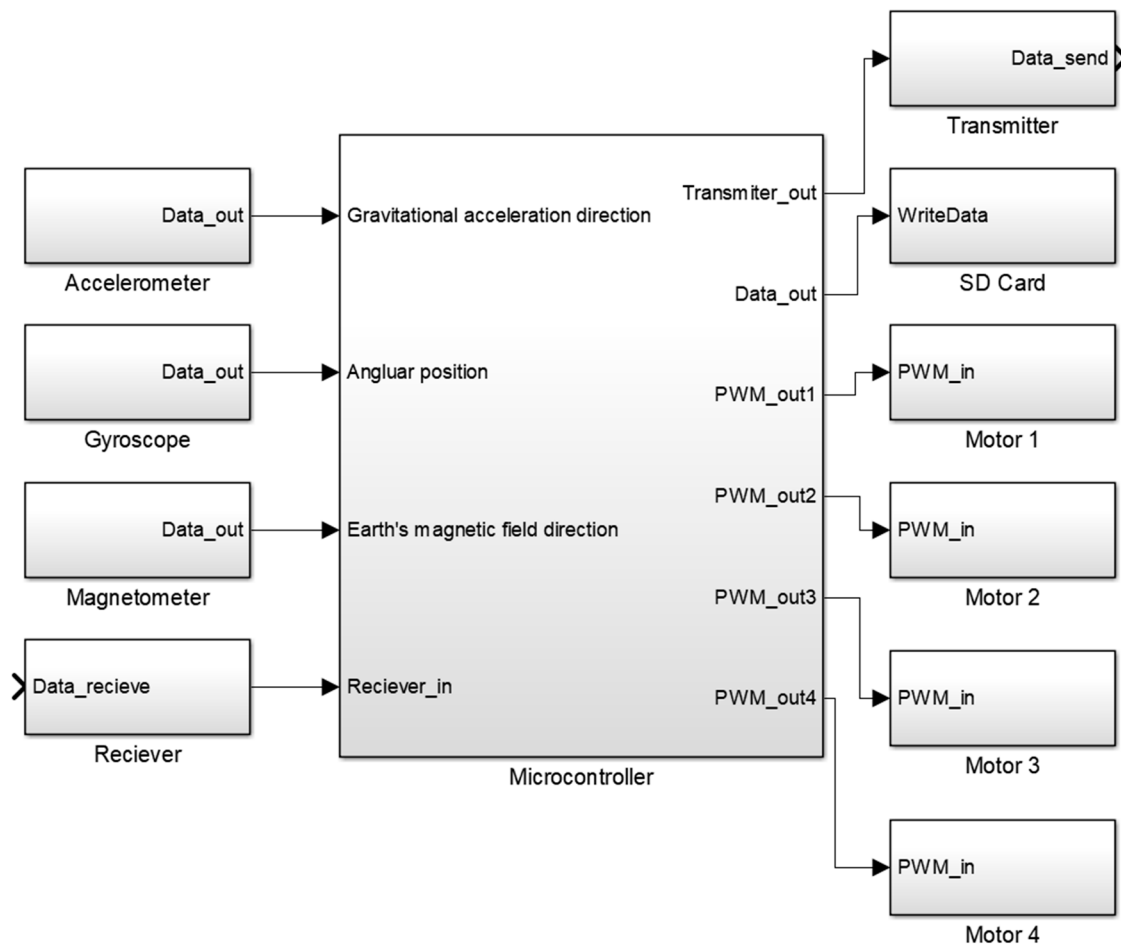


Figure 5. Block scheme of the quadrotor UAV.

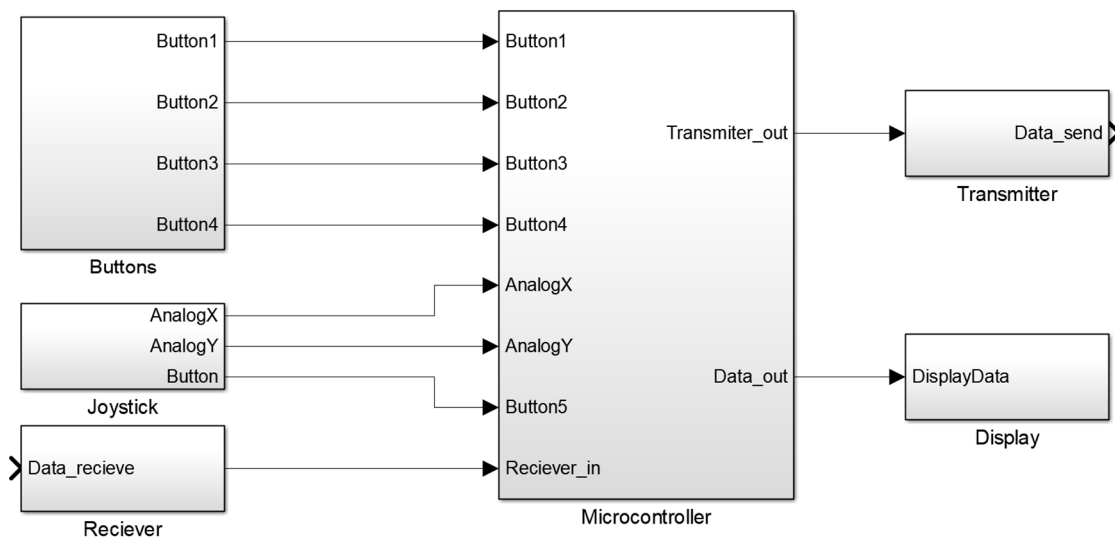


Figure 6. Block scheme of the remote controller.

In accordance with these block schemes and the dimensions imposed by the mechanical elements, a series of electrical components were chosen. These have been selected so that they can achieve the specifications of the desired control, allow flexibility in resolving errors and have low cost. Also, from the point of view of the processing capacity and the number of input and output signals, respectively, a microcontroller was chosen that satisfies these conditions.

The wiring diagram and the implemented UAV system are presented in Figure 7. The corresponding remote controller schemes are in Figure 8, where (a) represents the wiring diagram designed in Eagle, while (b) is the implemented circuit.

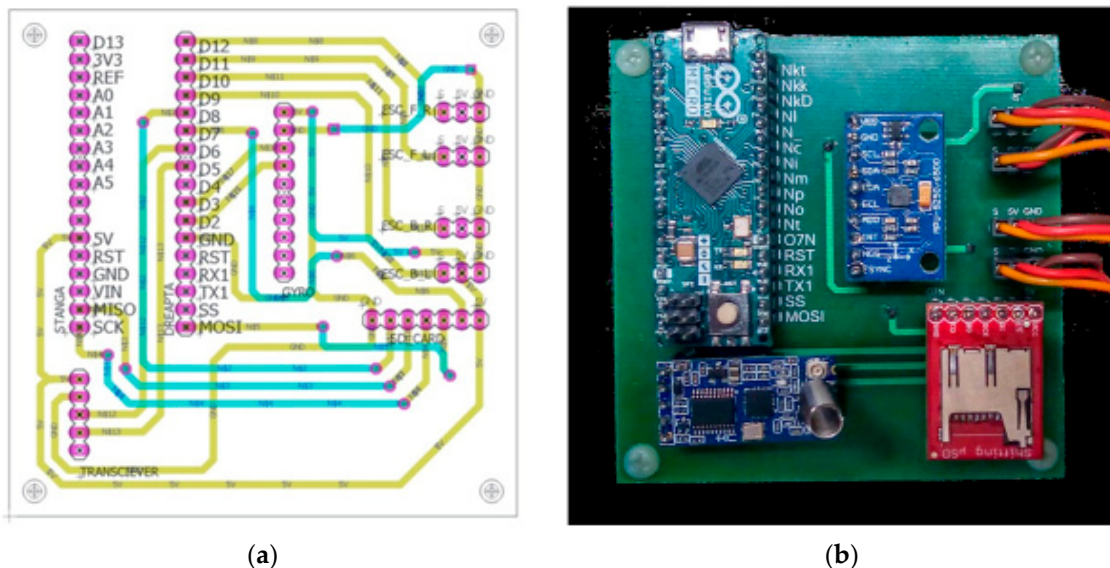
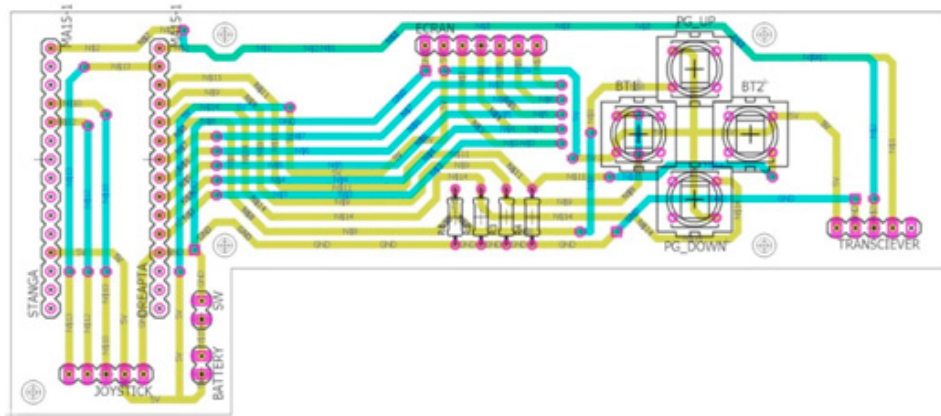
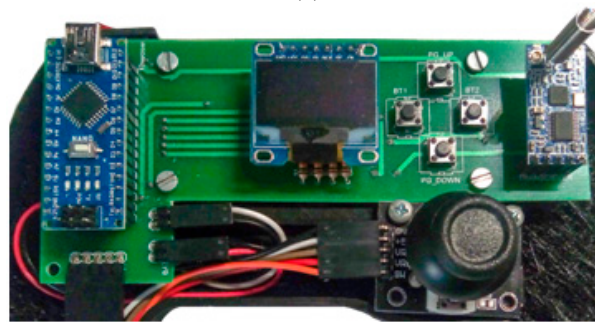


Figure 7. The designed (a) and implemented (b) electronic circuit of UAV.

Measurements were realized without using the developed estimator. Figure 9 presents the raw results of the gyroscope, accelerometer and magnetometer for a linear movement. It can be concluded that in the case of noisy signals, such an approach is not usable in a feedback control structure. It is obvious the necessity of the estimator.



(a)



(b)

Figure 8. The remote controller: (a) the designed and (b) the implemented electronic circuit.

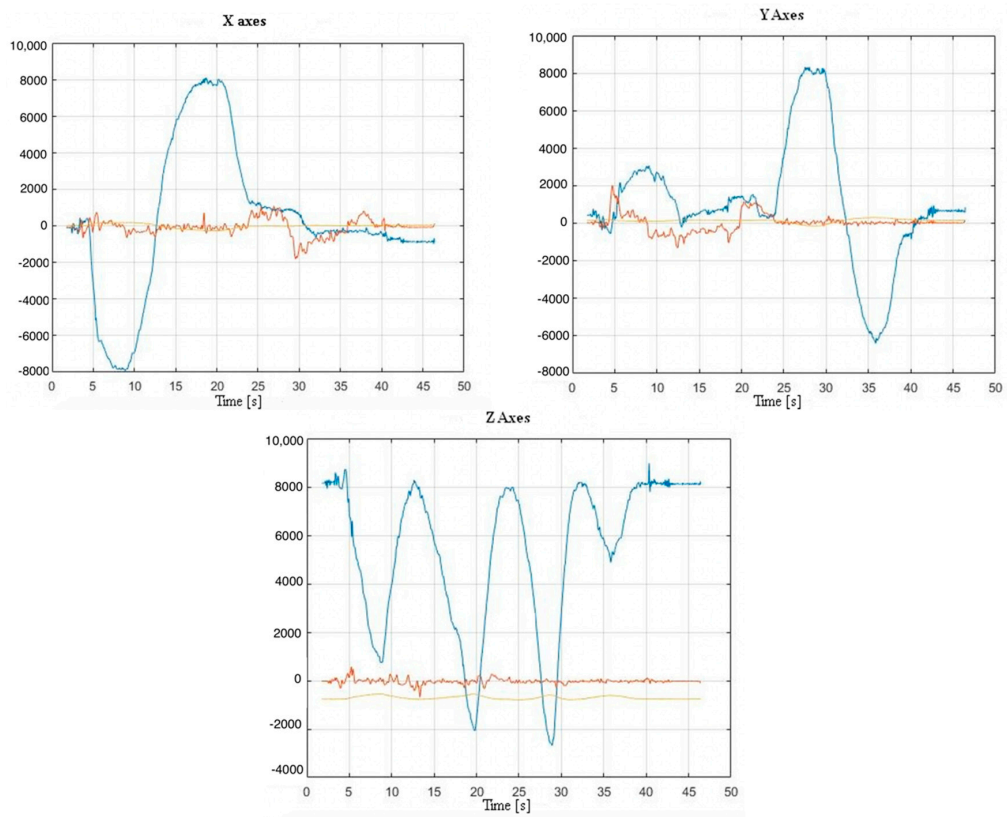


Figure 9. Raw measured output for the three degree of freedom.

The quaternion-based estimation algorithm described in the previous section was implemented on the microcontroller. In order to test the obtained system, a reference sequence of the form: 0, maximum value to the right, maximum value to the left was applied. The obtained results are plotted in Figure 10.

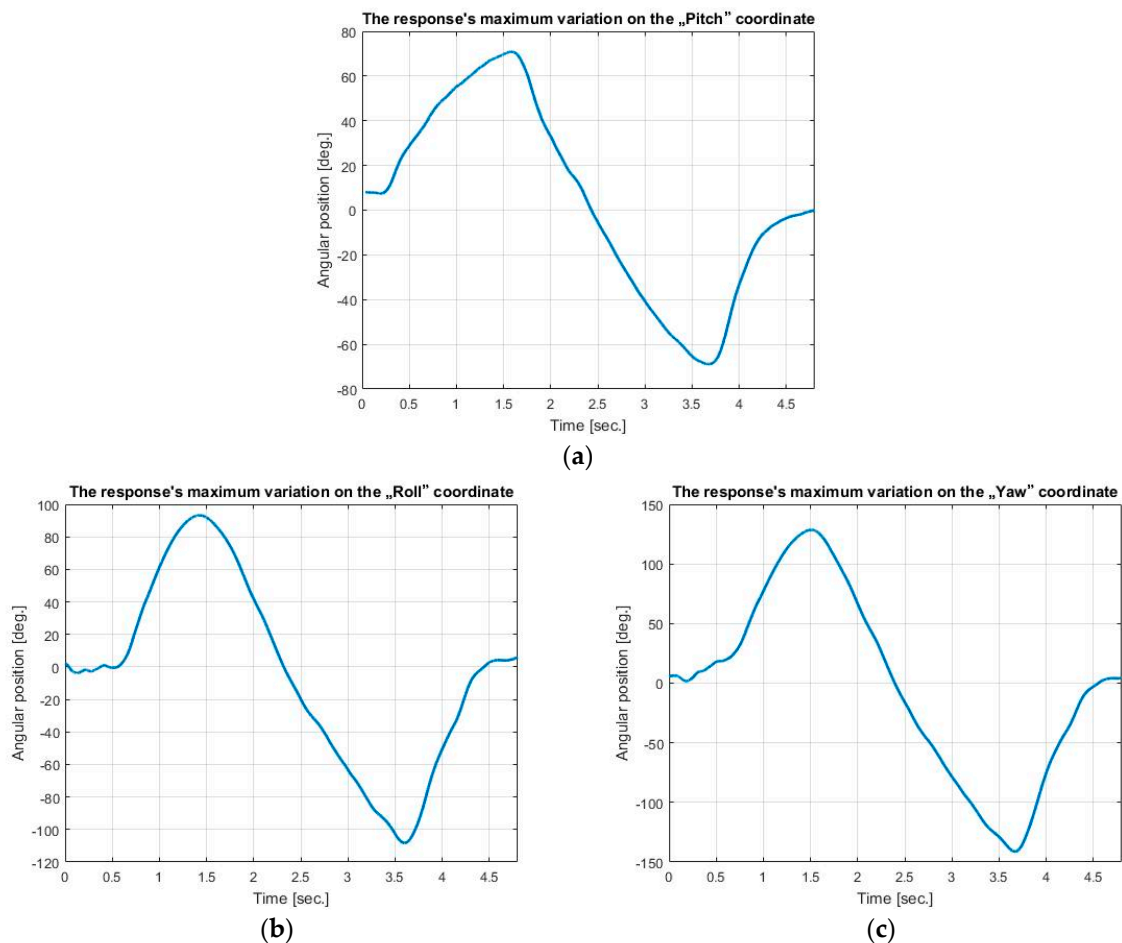


Figure 10. Measured output for the three degree of freedom: (a) pitch; (b) roll; (c) yaw.

The designed nonlinear model was tested, obtaining the results from Figure 11.

The designed control algorithms were also implemented on the microcontroller. The obtained results in the worst-case scenario, windy conditions, are plotted on Figure 12, presenting the response of the closed loop system to a step reference on each of the four directions of movement.

The performance was measured for different operation scenarios, including different step inputs on each axis, wind-free and windy conditions. The results for one of the “classical” scenarios— 16° step input for angular position on X and Y axis and 45° on Z axis, 5 m altitude, with relatively high wind speed—are presented in Table 1, highlighting good performances. All these results are comparable with the results of advanced control algorithms in [25–36], without needing expensive hardware equipment. In [26], where the studied quadrotor is similar with our prototype, a LQR controller is used for altitude and a PD controller for position, resulting in a settling time for a step input between 2 and 3.7 s and overshoot 13–20%. Using a backstepping controller combined with the PD, the settling time varies between 2.10 and 3.70 s and the overshoot between 12 and 14%. The LQR controller used both for altitude and position, the settling time are 2.7–3.35 s, while the overshoot is 19–25%. The combination of the backstepping controller with LQR leads to values varying between 2.7–3.3 s, overshoot 15–25%. In our experiments the overshoot does not exceed 13.75% neither in worst case and the largest settling time is 1.2 s. The model identification adaptive control (MIAC) used in [28] leads to settling time between 0.8 and 1.4 s, while with the Model Reference Adaptive Control (MRAC) from the same

study, the achieved settling time is of 0.75–2 s, very close to our values. The main advantages of these two (MIAC and MRAC) controllers are the overshoot cancellation, but the cost is the control effort. Analyzing the active disturbance rejection controller designed in [11], the presented settling times are 0.85–1.5 s for a 20° step input, overshoot is 7–25%, comparable with our results. The advantage of the high-order sliding mode-based fixed-time active disturbance rejection control from [11] is that it tracks the unknown disturbances in about 3 s.

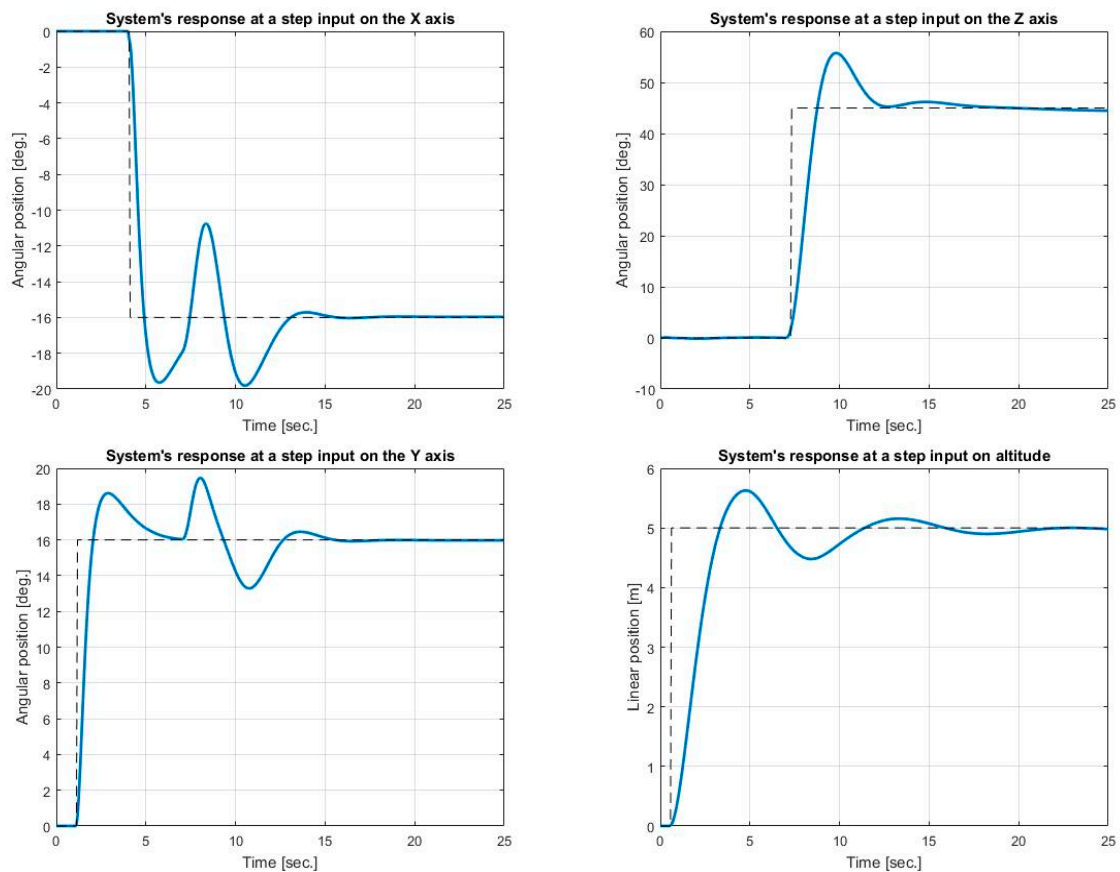


Figure 11. The nonlinear model output for step input on each axis.

Table 1. The obtained performance measures.

Movement	Overshoot [%]	Settling Time [sec]	Steady State Position Error
Front-back	13.75	0.65	0
Left-right	12.5	0.51	0
Rotation around the central axis	0	1.2	0
Up-down	2	0.8	0

Comparing our results with the results of the bioinspired controller from [10], the present results are still competitive. Moreover, imposing different settling time and overshoot in the design stage, it is possible to set a different transient response. Reducing the overshoot will increase the settling time and vice versa. Obviously, designing an advanced controller could increase the performances, but the idea of present work was to analyze the most simple algorithm, a PID controller.

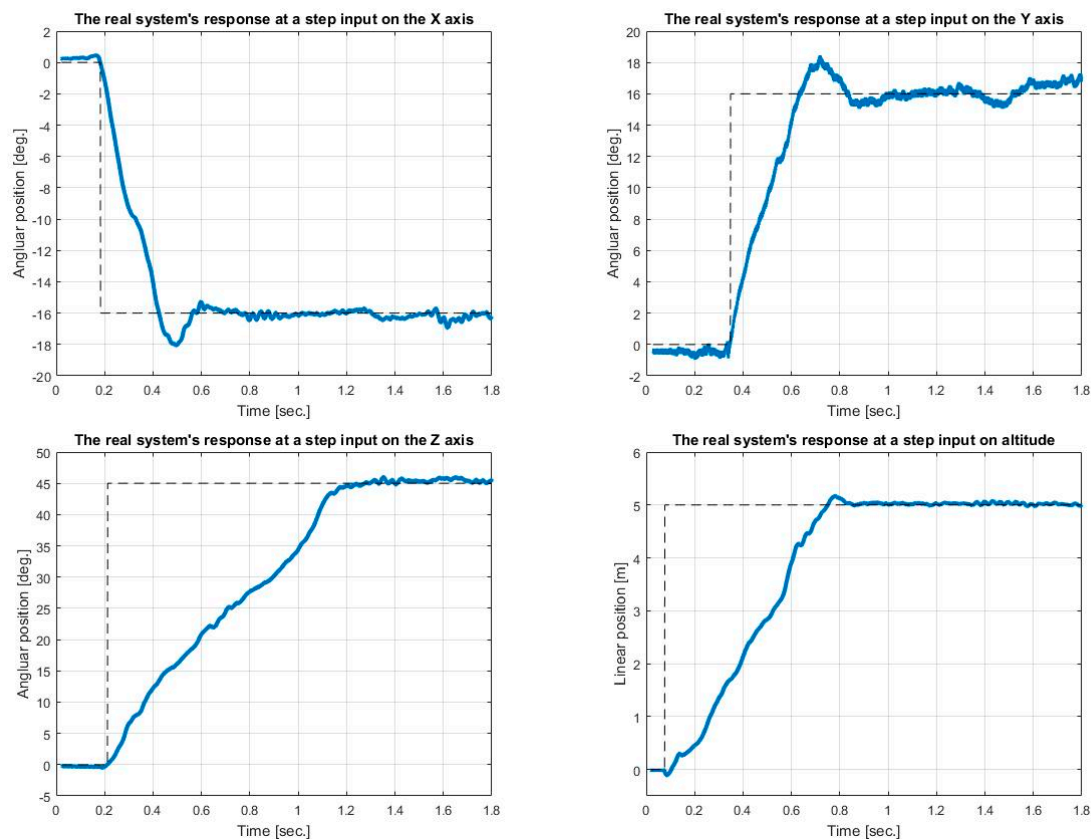


Figure 12. The closed loop system output for step input on each axis.

4. Conclusions

The present research is focused on a low-cost, but performing UAV system design. Taken as a whole, such a flight system presents great difficulties in obtaining positioning data, in particular due to their complex determination or estimation algorithms. In addition to the high complexity of the estimation algorithms, the problem of measurement errors and the resolution of the sensors must often be taken into consideration, so that the control structures are provided with the most accurate data. The offered solution is the quaternion-based estimation. In addition, the tuning of the proportional, integrative, and derivative terms of the control laws is another major problem of the UAV system. Also, the nonlinearities present in such a system introduce challenging problems.

The prototype described in the previous sections offers solution for all these problems.

As future works a global positioning system (GPS) would be added to the equipment model in order to acquire more functionalities.

Author Contributions: Conceptualization and methodology, M.S.; validation, E.H.D.; formal analysis, C.I.M. and L.C.M.; writing—review and editing, C.I.M. and E.H.D.; supervision, E.H.D.; funding acquisition, L.C.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI-UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0734/ROBIN—“Roboții și Societatea: Sisteme Cognitive pentru Roboți Personali și Vehicule Autonome,” within PNCDDI III. E.H.D. was funded by Hungarian Academy of Science, Janos Bolyai Grant (BO/00313/17) and the ÚNKP-19-4-OE-64 New National Excellence Program of the Ministry for Innovation and Technology. Cristina Muresan is financed by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI-UEFISCDI, project number PN-III-P1-1.1-TE-2019-0745.

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

Abbreviations

Notations Used

A_e, B_e, C	matrices of the state space representation
b	propulsion coefficient
$c(k)$	the control signal at this current iteration k
$c(\Psi)$	$\cos(\Psi)$
$s(\Psi)$	$\sin(\Psi)$
$c(\theta)$	$\cos(\theta)$
$s(\theta)$	$\sin(\theta)$
$c(\Phi)$	$\cos(\Phi)$
$s(\Phi)$	$\sin(\Phi)$
d	aerodynamic resistance coefficient
d_{sensor}	orientation of the sensor
d_{ref}	orientation of the reference system
$e(k)$	the error signal at iteration k
\hat{e}_z	the unit vector on the Z axis of the reference coordinate system
\hat{e}_3	unit vector on the Z axis of the coordinate system attached to the quadrotor
f_{com}	composition of functions
f_x, f_y, f_z	forces acting on the quadrotor on the X, Y and Z axes
F	gradient of function f
F_a	vector of forces acting on the quadrotor
f_p	total propulsion force developed by the engines
$f_v = [f_{vx} \quad f_{vy} \quad f_{vz}]^T$	disturbances or forces that are opposed to the rotation of each engine on the X, Y and Z axes
g	the gravitational acceleration
g_a	the gyroscope moments caused by the combined velocities of the four motors
I	inertia matrix, with components I_x, I_y, I_z on each axis
K_p, K_i, K_d	the proportionality, integration, and derivative constant
l	the distance from the center of gravity of the quadrotor to the center of rotation of the engine
m	magnetic field
M_a	angular velocities vector applied to the quadrotor
m_q	mass of the quadrotor
of	objective function
P_a	vector of the linear and angular velocities of the quadrotor
P_k	weighting factor
P_p	vector of the linear and angular positions of the quadrotor
R	rotation matrix
\hat{q}	quaternion
${}^D_E \hat{q}$	quaternion of system E with respect to the reference system D
${}^A_B R$	rotation matrix of the coordinate system B with respect to the reference system A
\hat{q}_0	initial point
\hat{q}_{k+1}	future orientation
$q_1 \quad q_2 \quad q_3 \quad q_4$	components of the quaternion q
$r(k)$	the reference signal at current iteration k
$\hat{r}_x, \hat{r}_y, \hat{r}_z$	components of the unity vector \hat{r}
T_s	sampling period
$T_v(\Phi, \theta)$	matrix of angular velocities transformations
u	input vector
u_e	input vector at equilibrium point
v_p	vector of the derivatives of the linear positions of P_p
v_a	vector of the linear velocities of the vector P_a
\dot{v}_a	linear acceleration
$w_x \quad w_y \quad w_z$	angular positions on X, Y and Z axes
W	angular positions vector
X	state vector
X_e	state vector of equilibrium point

x_e, y_e, z_e	equilibrium points on the X, Y and Z axes
y	output vector
y(k)	the output of the system, measured by sensors at iteration k
Greek Letters	
α, β	constants
Ψ, θ, Φ	Euler angles
τ_a	vector of angular velocities generated by the velocity differences of the four motors
$\tau_x \tau_y \tau_z$	angular velocities on the X, Y and Z axes generated by the velocity differences of the four motors
τ_v	vector of angular velocities produced by air currents on each motor
$\tau_{vx} \tau_{vy} \tau_{vz}$	angular velocities produced by air currents on each motor on the X, Y and Z axes
μ	algorithm step
μ_t	variable step at time t
ω_a	vector of the angular velocities of the vector P_a
ω_p	vector of the derivatives of the angular positions of P_p
$\Omega_a = \begin{bmatrix} \Omega_1 & \Omega_2 & \Omega_3 & \Omega_4 \end{bmatrix}$	vector of angular velocities of the four motors

References

- Shafi, U.; Mumtaz, R.; García-Nieto, J.; Hassan, S.A.; Zaidi, S.A.R.; Iqbal, N. Precision Agriculture Techniques and Practices: From Considerations to Applications. *Sensors* **2019**, *19*, 3796. [CrossRef]
- Avanzato, R.; Beritelli, F. An Innovative Technique for Identification of Missing Persons in Natural Disaster Based on Drone-Femtocell Systems. *Sensors* **2019**, *19*, 4547. [CrossRef] [PubMed]
- Rosa, R.; Wehrmeister, M.A.; Brito, T.; Lima, J.L.; Pereira, A.I.P.N. Honeycomb Map: A Bioinspired Topological Map for Indoor Search and Rescue Unmanned Aerial Vehicles. *Sensors* **2020**, *20*, 907. [CrossRef] [PubMed]
- Hinas, A.; Ragel, R.; Roberts, J.M.; Gonzalez, F. A Framework for Multiple Ground Target Finding and Inspection Using a Multirotor UAS. *Sensors* **2020**, *20*, 272. [CrossRef]
- Hu, Z.; Wan, K.; Gao, X.; Zhai, Y.; Wang, Q. Deep Reinforcement Learning Approach with Multiple Experience Pools for UAV's Autonomous Motion Planning in Complex Unknown Environments. *Sensors* **2020**, *20*, 1890. [CrossRef] [PubMed]
- Fabra, F.; Zamora, W.; Sanguesa, J.A.; Calafate, C.T.; Cano, J.C.; Manzoni, P. A Distributed Approach for Collision Avoidance between Multirotor UAVs Following Planned Missions. *Sensors* **2019**, *19*, 2404. [CrossRef] [PubMed]
- González-Rocha, J.; De Wekker, S.F.J.; Ross, S.D.; Woolsey, C.A. Wind Profiling in the Lower Atmosphere from Wind-Induced Perturbations to Multirotor UAS. *Sensors* **2020**, *20*, 1341. [CrossRef] [PubMed]
- Zhang, R.; Zhang, J.; Yu, H. Review of modeling and control in UAV autonomous maneuvering flight. In Proceedings of the 2018 IEEE International Conference on Mechatronics and Automation (ICMA), Changchun, China, 5–8 August 2018; pp. 1920–1925.
- García-Nieto, S.; Velasco-Carrau, J.; Paredes-Valles, F.; Salcedo, J.V.; Simarro, R. Motion Equations and Attitude Control in the Vertical Flight of a VTOL Bi-Rotor UAV. *Electronics* **2019**, *8*, 208. [CrossRef]
- Armendariz, S.; Becerra, V.; Bausch, N. Bio-inspired Autonomous Visual Vertical and Horizontal Control of a Quadrotor Unmanned Aerial Vehicle. *Electronics* **2019**, *8*, 184. [CrossRef]
- Song, C.; Wei, C.; Yang, F.; Cui, N. High-Order Sliding Mode-Based Fixed-Time Active Disturbance Rejection Control for Quadrotor Attitude System. *Electronics* **2018**, *7*, 357. [CrossRef]
- Ferrero, A. Control of a Supersonic Inlet in Off-Design Conditions with Plasma Actuators and Bleed. *Aerospace* **2020**, *7*, 32. [CrossRef]
- Qin, Z.; Dong, C.; Wang, H.; Li, A.; Dai, H.; Sun, W.S.; Xu, Z. Trajectory Planning for Data Collection of Energy-Constrained Heterogeneous UAVs. *Sensors* **2019**, *19*, 4884. [CrossRef] [PubMed]
- Madgwick, S.O.H. An Efficient Orientation Filter for Inertial and Inertial/Magnetic Sensor Arrays, Internal_Report. 2010. Available online: https://www.samba.org/tridge/UAV/madgwick_internal_report.pdf (accessed on 15 March 2020).
- Cavallo, A.; Cirillo, A.; Cirillo, P.; De Maria, G.; Falco, P.; Natale, C.; Pirozzi, S. Experimental Comparison of Sensor Fusion Algorithms for Attitude Estimation. In *IFAC World Congress*; IFAC: Cape Town, South Africa, 2014; Volume 19, pp. 7585–7591.

16. Feng, K.; Li, J.; Zhang, X.; Shen, C.; Bi, Y.; Zheng, T.; Liu, J. A New Quaternion-Based Kalman Filter for Real-Time Attitude Estimation Using the Two-Step Geometrically-Intuitive Correction Algorithm. *Sensors* **2017**, *17*, 2146. [[CrossRef](#)]
17. Danku, A.; Kovari, A.; Miclea, L.C.; Dulf, E.-H. Intelligent Control of an Aerodynamical System. In Proceedings of the 2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 5–7 September 2019; pp. 49–52.
18. Dulf, E.-H.; Timis, D.D.; Szekely, L.; Miclea, L.C. Adaptive Fractional Order Control Applied to a Multi-Rotor System. In Proceedings of the 2019 22nd International Conference on Control Systems and Computer Science (CSCS), Bucharest, Romania, 28–30 May 2019; pp. 696–699.
19. Humphries, U.; Rajchakit, G.; Kaewmesri, P.; Chanthorn, P.; Sriraman, R.; Samidurai, R.; Lim, C.P. Stochastic Memristive Quaternion-Valued Neural Networks with Time Delays: An Analysis on Mean Square Exponential Input-to-State Stability. *Mathematics* **2020**, *8*, 815. [[CrossRef](#)]
20. Kizilates, C.; Catarino, P.; Tuglu, N. On the Bicomplex Generalized Tribonacci Quaternions. *Mathematics* **2019**, *7*, 80. [[CrossRef](#)]
21. Cabarbaye, A.; Lozano, R.; Estrada, M.B. Adaptive quaternion control of a 3-DOF inertial stabilised platforms. *Int. J. Control* **2018**, *93*, 473–482. [[CrossRef](#)]
22. Birlescu, I.; Husty, M.; Calin, V.; Gherman, B.; Tucan, P.; Pisla, D. Joint-Space Characterization of a Medical Parallel Robot Based on a Dual Quaternion Representation of SE(3). *Mathematics* **2020**, *8*, 1086. [[CrossRef](#)]
23. Wei, R.; Cao, J. Synchronization control of quaternion-valued memristive neural networks with and without event-triggered scheme. *Cogn. Neurodynamics* **2019**, *13*, 489–502. [[CrossRef](#)]
24. Zmitri, M.; Fourati, H.; Vuillerme, N. Human Activities and Postures Recognition: From Inertial Measurements to Quaternion-Based Approaches. *Sensors* **2019**, *19*, 4058. [[CrossRef](#)]
25. Long, Y.; Lyttle, S.; Pagano, N.; Cappelleri, D.J. Design and Quaternion-Based Attitude Control of the Omnicopter MAV Using Feedback Linearization. In Proceedings of the ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2012, Chicago, IL, USA, 12–15 August 2012.
26. Chovancová, A.; Fico, T.; Hubinský, P.; Duchoň, F. Comparison of various quaternion-based control methods applied to quadrotor with disturbance observer and position estimator. *Robot. Auton. Syst.* **2016**, *79*, 87–98. [[CrossRef](#)]
27. Sanwale, J.; Trivedi, P.; Kothari, M.; Malagaudanavar, A. Quaternion-based position control of a quadrotor unmanned aerial vehicle using robust nonlinear third-order sliding mode control with disturbance cancellation. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2019**, *234*, 997–1013. [[CrossRef](#)]
28. Schreier, M. Quaternion-based adaptive attitude control schemes for quadrotor systems. *Int. J. Mechatron. Autom.* **2013**, *3*, 217. [[CrossRef](#)]
29. Sanchez, M.E.G.; Abaunza, H.; Castillo, P.; Lozano, R.; García-Beltrán, C.D. Quadrotor Energy-Based Control Laws: A Unit-Quaternion Approach. *J. Intell. Robot. Syst.* **2017**, *88*, 347–377. [[CrossRef](#)]
30. Abaunza, H.; Castillo, P.; Lozano, R. Quaternion Modeling and Control Approaches. In *Handbook of Unmanned Aerial Vehicles*; Vachtsevanos, G., Ed.; Springer: Cham, Switzerland, 2018; pp. 1–29.
31. Sanchez, M.E.G.; Abaunza, H.; Castillo, P.; Lozano, R.; García-Beltrán, C.; Rodriguez-Palacios, A. Passivity-Based Control for a Micro Air Vehicle Using Unit Quaternions. *Appl. Sci.* **2016**, *7*, 13. [[CrossRef](#)]
32. Colmenares-Vazquez, J.; Marchand, N.; Castillo, P.; Gomez-Balderas, J.E. An intermediary quaternion-based control for trajectory following using a quadrotor. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 5965–5970.
33. Andersen, T.S.; Kristiansen, R. Quaternion guidance and control of quadrotor. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 1567–1601.
34. Carino, J.; Abaunza, H.; Castillo, P. Quadrotor quaternion control. In Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015; pp. 825–831.
35. Fresk, E.; Nikolakopoulos, G. Full quaternion based attitude control for a quadrotor. In Proceedings of the 2013 European Control Conference (ECC), Zurich, Switzerland, 17–19 July 2013; pp. 3864–3869.

36. Ji, X. Partial study of quadrotor based on quaternions. In Proceedings of the 6th International Conference on Computer-Aided Design, Manufacturing, Modeling and Simulation (CDMMS 2018), Busan, South Korea, 14–15 April 2018.
37. Sanchez, M.E.G.; Gonzalez, O.H.; Lozano, R.; García-Beltrán, C.; Valencia-Palomo, G.; López-Estrada, F.-R. Energy-Based Control and LMI-Based Control for a Quadrotor Transporting a Payload. *Mathematics* **2019**, *7*, 1090. [[CrossRef](#)]
38. Shao, X.; Liu, N.; Wang, Z.; Zhang, W.; Yang, W. Neuroadaptive integral robust control of visual quadrotor for tracking a moving object. *Mech. Syst. Signal Process.* **2020**, *136*, 106513. [[CrossRef](#)]
39. Raj, J.; Raghunwaiya, K.S.; Vanualailai, J. Novel Lyapunov-Based Autonomous Controllers for Quadrotors. *IEEE Access* **2020**, *8*, 47393–47406. [[CrossRef](#)]
40. Guzmán-Rabasa, J.A.; López-Estrada, F.R.; González-Contreras, B.M.; Valencia-Palomo, G.; Chadli, M.; Pérez-Patricio, M. Actuator fault detection and isolation on a quadrotor unmanned aerial vehicle modeled as a linear parameter-varying system. *Meas. Control.* **2019**, *52*, 1228–1239. [[CrossRef](#)]
41. Chiella, A.C.B.; Teixeira, B.O.S.; Pereira, G.A.S. Quaternion-Based Robust Attitude Estimation Using an Adaptive Unscented Kalman Filter. *Sensors* **2019**, *19*, 2372. [[CrossRef](#)]
42. Naharro, R.J.; Gómez-Bravo, F.; Garcia, J.M.; Sánchez-Raya, M.; Gómez-Galán, J.A. A Smart Sensor for Defending against Clock Glitching Attacks on the I2C Protocol in Robotic Applications. *Sensors* **2017**, *17*, 677. [[CrossRef](#)]
43. Abubakar, A.B.; Kumam, P.; Mohammad, H.; Awwal, A.M. An Efficient Conjugate Gradient Method for Convex Constrained Monotone Nonlinear Equations with Applications. *Mathematics* **2019**, *7*, 767. [[CrossRef](#)]
44. Waliszkievicz, M.; Wojtowicz, K.; Rochala, Z.; Balestrieri, E. The Design and Implementation of a Custom Platform for the Experimental Tuning of a Quadcopter Controller. *Sensors* **2020**, *20*, 1940. [[CrossRef](#)] [[PubMed](#)]
45. Bavafa-Toosi, Y. *Introduction to Linear Control Systems*; Elsevier: Amsterdam, The Netherlands, 2019. [[CrossRef](#)]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).