

Backstepping Control of Drone †

Ali Saibi ¹, Razika Boushaki ² and Hadjira Belaidi ^{1,*} 

¹ Signals and Systems Laboratory, Institute of Electrical and Electronic Engineering, University M'hamed Bougara of Boumerdes, Boumerdes 35000, Algeria; a.saibi@univ-boumerdes.dz

² Laboratoire d'Automatique Appliquée, Institute of Electrical and Electronic Engineering, University M'hamed Bougara of Boumerdes, Boumerdes 35000, Algeria; r.boushaki@univ-boumerdes.dz

* Correspondence: ha.belaidi@univ-boumerdes.dz or hadjira983@yahoo.fr

† Presented at the 1st International Conference on Computational Engineering and Intelligent Systems, Online, 10–12 December 2021.

Abstract: This work derives the models which can be used to design and implement control laws for six degrees-of-freedom (DOF) quadrotor stability. The first part of this paper deals with the presentation of the background of quadrotor modeling; the second part describes the direct control of the drone using the backstepping control principal. This principal is based on the division of the system into several sub-systems in a cascade, which makes the control laws generated on each subsystem, in a decreasing manner, until a global control law for the whole system is generated. The simulation results for the sm controller are generated on the MATLAB/Simulink platform; the results show a good performance in both the transient and steady-state operations.

Keywords: quadrotor; dynamic model; backstepping



Citation: Saibi, A.; Boushaki, R.; Belaidi, H. Backstepping Control of Drone. *Eng. Proc.* **2022**, *14*, 4. <https://doi.org/10.3390/engproc2022014004>

Academic Editors: Abdelmadjid Recioui, Hamid Bentarzi and Fatma Zohra Dekhandji

Published: 24 January 2022

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



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Drones are used as a means of monitoring and following important events such as forest fires and political demonstrations; they can also be used to rescue people in earthquakes in the civilian field. Additionally, they can be used as tools for supervision and fault diagnostic in smart grid systems. Drones can also act as aerial base stations (BSs) to deliver communication services (both uplink and downlink) for the subscribers on the ground [1]. In the military field, drones reduce human losses and material; they are able to closely monitor the enemy and reveal their location without exposing individuals to danger, and can direct precise strikes, like helicopters and aircrafts.

Many control approaches were developed for unmanned aerial vehicles (UAVS) in the literature. Hence, a detailed drone description model is described in [2–4]. The sliding mode control strategy based on backstepping control is widely used as it can produce high performances and a faster response for drone systems in general [5–9] and in indoor micro-quadrotors in particular [10]. Modeling- and backstepping-based nonlinear control for a six-DOF quadrotor helicopter is proposed by [11], in addition to the proposition of an adaptive sliding mode control for a quadrotor helicopter in [12].

Quadrotor mathematical modeling is very complicated [13]. It presents non-linearity due to having six degrees of freedom (translational and rotational motion) with only four control inputs. To preserve the equilibrium or the desired attitude of the drone, a traditional PID controller is commonly used; however, it does not ensure the robustness of the quadrotor, whatever the controlling target Euler angle or angular rate is. Hence, new approaches have proposed the use of a cascade PID algorithm to provide better performance and motion stabilization [14,15]. However, the implemented system still needs to tackle the drone system's non-linearity.

Hence, to control such a system, a backstepping method is the best choice [16,17]. Backstepping control is based on the Lyapunov stability principle of dynamic systems, and

it is robust to parametric variation; thus, it ensures the stability of the system and gives good performances results.

This work describes the direct control of the drone using the backstepping control principal, where the quadrotor is supposed to track the desired trajectories with an acceptable dynamic. This paper is treated in three mean parts, organized as follows: first, the mathematical model of the quadrotor is developed; in the second step, the algorithm of backstepping control is presented; in the last section, the simulation results and their interpretations are presented. Finally, the conclusion and possible future developments of the work are presented.

2. Quadrotor Dynamic Modeling

The quadrotor consists of four rotors in a cross-configuration, as shown in Figure 1. The four-rotor design allows the quadrotor to be relatively simple in design yet highly reliable and maneuverable. The dynamic equation of movement of the attitude could be deduced from the Euler equation. The quadrotor mathematical and state-space models are explained in the following subsections.

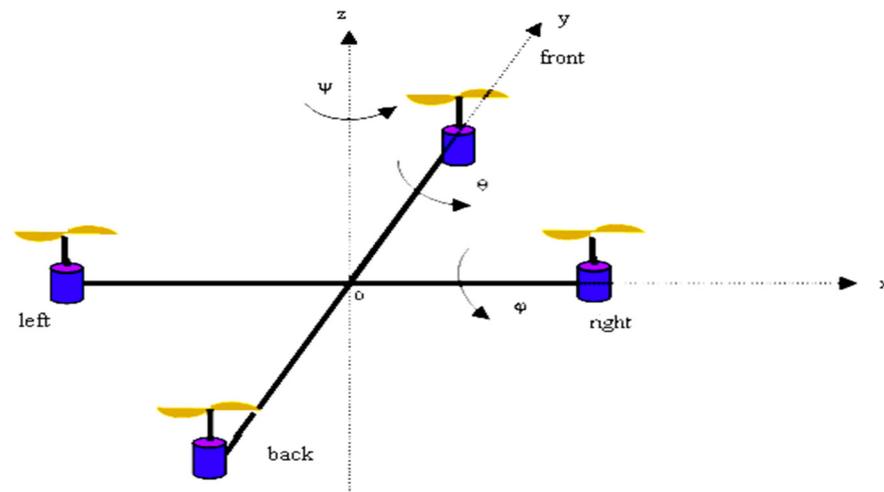


Figure 1. Quadrotor configuration.

2.1. Drone Dynamic Model

The dynamic model of the quadrotor can be defined in terms of the position vector and force expressions as given in Equations (1)–(3).

$$\begin{cases} x'' = -\frac{T}{m}[\sin(\varphi) \sin(\psi) + \cos(\varphi) \cos(\theta) \cos(\psi)] \\ y'' = -\frac{T}{m}[\cos(\varphi) \sin(\theta) \sin(\psi) - \sin(\psi) \cos(\theta)] \\ z'' = g - \frac{T}{m}[\cos(\varphi) \cos(\theta)] \end{cases} \quad (1)$$

Such that (x'', y'', z'') represents the second derivative of position vector, T is the torque, and m denotes the mass of the drone.

The moment equations can be expressed in terms of the orientation angles (φ, θ, ψ) —roll, pitch, and yaw, respectively—as given in Equations (2) and (3).

$$\begin{cases} p' = \frac{I_z - I_y}{I_x}qr - \frac{I_r}{I_x}\Omega q + \frac{1}{I_x}\tau_\varphi \\ q' = \frac{I_z - I_x}{I_y}pr - \frac{I_r}{I_y}\Omega p + \frac{1}{I_y}\tau_\theta \\ r' = \frac{I_y - I_x}{I_z}pq + \frac{1}{I_z}\tau_\psi \end{cases} \quad (2)$$

$$\begin{cases} \varphi' = p + q \sin(\varphi)tg(\theta) + r \cos(\varphi)tg(\theta) \\ \theta' = q \cos(\varphi) - r \sin(\varphi) \\ \psi' = q \frac{\sin(\varphi)}{\cos(\theta)} + r \frac{\cos(\varphi)}{\cos(\theta)} \end{cases} \quad (3)$$

τ_φ , τ_θ , and τ_ψ represent the total rolling, pitching, and yawing torques, while p , q , and r represent the angular velocities in the body frame.

2.2. State-Space Model

A state-space representation is a mathematical model of a physical system as a set of inputs, outputs, and state variables related by first-order differential equations. "State space" refers to the space whose axes are the state variables. The state of the system can be represented as a vector within that space.

In this work, the state-space model of the quadrotor in the inertial frame is developed. Thus, the dynamic model of the quadrotor in the inertial frame can be expressed by the system referred to as Equation (4):

$$\begin{cases} x'_1 = x_2 \\ x'_2 = a_1 x_4 x_6 + a_3 \Omega x_4 + b_1 U_2 \\ x'_3 = x_4 \\ x'_4 = a_4 x_2 x_6 + a_6 \Omega x_2 + b_2 U_3 \\ x'_5 = x_6 \\ x'_6 = a_7 x_2 x_4 + b_3 U_4 \\ x'_7 = x_8 \\ x'_8 = \frac{\cos(x_1) \cos(x_2)}{m} U_1 - g \\ x'_9 = x_{10} \\ x'_{10} = U_y \frac{U_1}{m} \\ x'_{11} = x_{12} \\ x'_{12} = U_x \frac{U_1}{m} \end{cases} \quad (4)$$

The parameters a_1, a_4, a_7, b_1, b_2 , and b_3 can be calculated as follows:

$$a_1 = \frac{I_y - I_z}{I_x}, \quad a_3 = \frac{J_r}{I_x}, \quad a_4 = \frac{I_z - I_x}{I_y}, \quad a_6 = \frac{J_r}{I_y}, \quad a_7 = \frac{I_x - I_y}{I_z}, \\ b_1 = \frac{d}{I_x}, \quad b_2 = \frac{d}{I_y}, \quad b_3 = \frac{d}{I_z}$$

I_x, I_y , and I_z denote the inertias of the x -, y -, and z -axis of the quadrotor, respectively; J_r denotes the z -axis inertia of the propellers' rotors.

$$\text{whereas } U_x = \cos(x_1) \cos(x_3) \cos(x_5) + \sin(x_1) \sin(x_5) \\ U_y = \cos(x_1) \sin(x_3) \sin(x_5) - \sin(x_1) \cos(x_5)$$

To solve the given system, a backstepping control scheme is used as detailed in the following section.

3. Backstepping Control of Drone

The principle of backstepping is to divide the system into several sub-systems in a cascade. The control laws are then made for each subsystem, in a decreasing manner, until a global control law for the whole system is generated.

3.1. Control of the Angle φ

Considering the first subsystem mentioned below:

$$\begin{cases} x'_1 = x_2 \\ x'_2 = x_4 x_6 a_1 - x_4 \Omega a_2 + b_1 U_2 \end{cases} \quad (5)$$

step1

The error ε_1 between the desired and actual roll angle is expressed as follows: $\varepsilon_1 = x_1^d - x_1$. Consider the Lyapunov function $V_1 = \frac{1}{2} \varepsilon_1^2$, where the derivate of V_1 along x_1 trajectory, V' , is computed as follows:

$$V'_1 = \varepsilon_1 \varepsilon'_1 \text{ with : } \varepsilon'_1 = x^{d'}_1 - x'_1 = x^{d'}_1 - x_2$$

Choosing $\varepsilon'_1 = -K_1\varepsilon_1$ (knowing that: $K_1\varepsilon_1$ is positive definite function). Thus, the desired x_2^d is extracted as: $x_2^d = x_1^{d'} + K_1\varepsilon_1$

step2

Denoting ε_2 the error between desired and actual roll angle rate, so that: $\varepsilon_2 = x_2^d - x_2$. Using $V_2 = V_1 + \frac{1}{2}\varepsilon_2^2$ as a candidate Lyapunov function, we obtain:

$$\begin{aligned} V'_2 &= V'_1 + \varepsilon_2\varepsilon'_2 = \varepsilon_1\varepsilon'_1 + \varepsilon_2\varepsilon'_2 = \varepsilon_2(x_1^{d'} - x_2) + \varepsilon_2(x_2^d - x_2) \\ &= \varepsilon_1(x_1^d - (x_2^{d'} - \varepsilon_2)) + \varepsilon_2(x_2^d - (x_4x_6a_1 - x_4\Omega a_2 + b_1U)) \\ &= \varepsilon_1(x_1^d - x_2^{d'}) + \varepsilon_1\varepsilon_2 + \varepsilon_2(x_2^d - (x_4x_6a_1 - x_4\Omega a_2 + b_1U)) \\ &= -K_1\varepsilon_1^2 + \varepsilon_2(\varepsilon_1 + x_2^d - x_4x_6a_1 + x_4\Omega a_2 - b_1U) \\ \varepsilon_1 + x_2^{d'} - x_4x_6a_1 + x_4\Omega a_2 - b_1U &= -K_2\varepsilon_2 \end{aligned}$$

where K_2 is a positive constant and $x_2^{d'} = a'_1 = [K_1(x_1^d - x_1) + x_1^{d'}] = -K_1(x_2)$.

Thus, the control law is expressed by: $U_2 = \frac{1}{b_1}[\varepsilon_1 - K_1x_2 - x_4x_6a_1 + x_4\Omega a_2 + K_2\varepsilon_2]$

3.2. Control of the Angle θ

Considering the second subsystem mentioned below:

$$\begin{cases} x'_3 = x_4 \\ x'_4 = x_2x_6a_4 + x_2\Omega a_6 + b_2U_3 \end{cases} \quad (6)$$

step1

Considering ε_3 is the error between the desired and actual angle θ and can be found by:

$$\varepsilon_3 = x_3^d - x_3 \Rightarrow \varepsilon'_3 = x_3^{d'} - x'_3$$

using Lyapunov stability by choosing: $V(\varepsilon_3) = \frac{1}{2}\varepsilon_3^2$. Therefore, if V' is negative, then the system trajectory is ensured to verify this condition:

$$v'(\varepsilon) = \varepsilon_3\varepsilon'_3 = \varepsilon_3(x_3^{d'} - x_4) < 0, \text{ then : } x_3^{d'} - x_4 = -K_3\varepsilon_3 \Rightarrow x_4^d = x_3^{d'} + K_3\varepsilon_3$$

step2

The error $\varepsilon_4 = x_4^d - x_4 \Rightarrow x_4 = x_4^d - \varepsilon_4 \Rightarrow \varepsilon'_4 = x_4^{d'} - x'_4$

$$\begin{aligned} V_4 &= V_3 + \frac{1}{2}\varepsilon_4^2 \Rightarrow V'_4 = \varepsilon_3\varepsilon'_3 + \varepsilon_4\varepsilon'_4 \\ \varepsilon'_3 &= x_3^{d'} - x_4^d + \varepsilon_4 \Rightarrow \varepsilon_3\varepsilon'_3 = \varepsilon_3(x_3^{d'} - x_4^d + \varepsilon_4) + \varepsilon_4(\varepsilon'_4) \\ \varepsilon_3\varepsilon'_3 &= \varepsilon_3(x_3^{d'} - x_4^d) + \varepsilon_3\varepsilon_4 + \varepsilon_4(\varepsilon'_4) = \varepsilon_4(\varepsilon_3 + x_4^{d'} - (a_3x_2x_6 + a_4\Omega x_2 + b_2U_3)) \\ \varepsilon_3 + x_4^{d'} - a_3x_2x_6 - a_4\Omega x_2 - b_2U_3 &= -K_4\varepsilon_4 \\ b_2U_3 &= \varepsilon_3 + x_4^{d'} - a_3x_2x_6 - a_4\Omega x_2 + K_4\varepsilon_4 \\ U_3 &= \frac{1}{b_2}[\varepsilon_3 - a_3x_2x_6 - a_4\Omega x_2 + K_4\varepsilon_4 - K_3x_4] \end{aligned}$$

3.3. Control of the Angle ψ

Now, consider the third subsystem mentioned below:

$$\begin{cases} x'_5 = x_6 \\ x'_6 = a_7x_2x_4 + b_3U_4 \end{cases} \quad (7)$$

step1

Let us name ε_5 the error between the desired and actual angle ψ . Thus:

$$\varepsilon_5 = x_5^d - x_5 \Rightarrow \varepsilon'_5 = x_5^{d'} - x'_5$$

With Lyapunov function being $V(\varepsilon_5) = \frac{1}{2}\varepsilon_5^2$, such that $v'(\varepsilon) = \varepsilon_5\varepsilon'_5 = \varepsilon_5(x_5^{d'} - x_5) < 0$

Thus, $x_5^{d'} - x'_5 = -K_5 \varepsilon_5 \Rightarrow x_6^d = x_5^{d'} + K_5 \varepsilon_5$

The error $\varepsilon_6 = x_6^d - x_6 \Rightarrow x_6 = x_6^d - \varepsilon_6 \Rightarrow \varepsilon'_6 = x_6^{d'} - x'_6$

$$\begin{aligned} V_6 &= V_5 + \frac{1}{2} \varepsilon_6^2 \Rightarrow V'_6 = \varepsilon_5 \varepsilon'_5 + \varepsilon_6 \varepsilon'_6 \\ \varepsilon'_5 &= x_5^{d'} - x_5^d + \varepsilon_6 \Rightarrow V'_6 = \varepsilon_5 (x_5^{d'} - x_5^d + \varepsilon_6) + \varepsilon_6 (\varepsilon'_6) \\ &= \varepsilon_5 (x_5^{d'} - x_5^d) + \varepsilon_5 \varepsilon_6 + \varepsilon_6 (\varepsilon'_6) \\ &\Rightarrow \varepsilon_5 + x_6^{d'} - a_5 x_2 x_4 - b_3 U_4 = -K_6 \varepsilon_6 \\ &= \varepsilon_6 (\varepsilon_5 + x_6^{d'} - (a_5 x_2 x_4 + b_3 U_4)) \\ b_3 U_4 &= \varepsilon_5 + x_6^{d'} - a_5 x_2 x_4 + K_6 \varepsilon_6 \Rightarrow U_4 = \frac{1}{b_3} [\varepsilon_5 - a_5 x_2 x_4 - K_6 \varepsilon_6 - K_5 x_6] \end{aligned}$$

3.4. Control of the Position z

Equation (8) represents the fourth subsystem:

$$\begin{cases} x'_7 = x_8 \\ x'_8 = \frac{\cos(x_1) \cos(x_2)}{m} U_1 - g \end{cases} \quad (8)$$

step1

ε_7 is the error between the desired and actual position z, such that:

$$\varepsilon_7 = x_7^d - x_7 ; \text{ thus,}$$

The Lyapunov function is $V(\varepsilon_7) = \frac{1}{2} \varepsilon_7^2 \Rightarrow v'(\varepsilon) = \varepsilon_7 \varepsilon'_7 = \varepsilon_7 (x_7^{d'} - x_8) < 0$ Then,
 $x_7^{d'} - x_8 = -K_7 \varepsilon_7 \Rightarrow x_8^d = x_7^{d'} + K_7 \varepsilon_7$

step2

The error $\varepsilon_8 = x_8^d - x_8 \Rightarrow x_8 = x_8^d - \varepsilon_8$ thus, $\varepsilon'_8 = x_8^{d'} - x'_8$

$$\begin{aligned} V_8 &= V_7 + \frac{1}{2} \varepsilon_8^2 \Rightarrow V'_8 = \varepsilon_7 \varepsilon'_7 + \varepsilon_8 \varepsilon'_8 \\ \varepsilon'_7 &= x_7^{d'} - x_8^d + \varepsilon_8 \Rightarrow \varepsilon_7 \varepsilon'_7 = \varepsilon_7 (x_7^{d'} - x_8^d + \varepsilon_8) + \varepsilon_8 (\varepsilon'_8) \\ \varepsilon_7 \varepsilon'_7 &= \varepsilon_7 (x_7^{d'} - x_8^d) + \varepsilon_7 \varepsilon_8 + \varepsilon_8 (\varepsilon'_8) = \varepsilon_8 (\varepsilon_7 + x_8^{d'} - (g - \frac{U_1}{m} \cos(x_1) \cos(x_3))) \\ \varepsilon_7 + x_8^{d'} - g + \frac{U_1}{m} \cos(x_1) \cos(x_3) &= -K_8 \varepsilon_8 \\ \frac{U_1}{m} \cos(x_1) \cos(x_3) &= -\varepsilon_7 - x_8^{d'} + g - K_8 \varepsilon_8 \\ U_1 &= \frac{m}{\cos(x_1) \cos(x_3)} [-\varepsilon_7 + g - K_8 \varepsilon_8 - K_7 x_8] \end{aligned}$$

3.5. Control of the Position y

Equation (9) represents the fifth subsystem:

$$\begin{cases} x'_9 = x_{10} \\ x'_{10} = U_y \frac{U_1}{m} \end{cases} \quad (9)$$

step1

Name ε_9 the error between the desired and actual position y

$$\varepsilon_9 = x_9^d - x_9 \Rightarrow \varepsilon'_9 = x_9^{d'} - x'_9$$

The Lyapunov function is $V(\varepsilon_9) = \frac{1}{2} \varepsilon_9^2 \Rightarrow V'(\varepsilon) = \varepsilon_9 \varepsilon'_9 = \varepsilon_9 (x_9^{d'} - x_{10}) < 0$ Then,
 $x_9^{d'} - x_{10} = -K_9 \varepsilon_9 \Rightarrow x_{10}^d = x_9^{d'} + K_9 \varepsilon_9$

step2

The error will be: $\varepsilon_{10} = x_{10}^d - x_{10} \Rightarrow x_{10} = x_{10}^d - \varepsilon_{10} \Rightarrow \varepsilon'_{10} = x_{10}^{d'} - x'_{10}$

$$\begin{aligned} V_{10} &= V_9 + \frac{1}{2} \varepsilon_{10}^2 \Rightarrow V' = \varepsilon_9 \varepsilon'_9 + \varepsilon_{10} \varepsilon'_{10} \\ \varepsilon'_9 &= x_9^{d'} - x_{10}^d + \varepsilon_{10} \Rightarrow \varepsilon_9 \varepsilon'_9 = \varepsilon_9 (x_9^{d'} - x_{10}^d + \varepsilon_{10}) + \varepsilon_{10} (\varepsilon'_{10}) \end{aligned}$$

$$= \varepsilon_9 \left(x_9^{d'} - x_{10}^d \right) + \varepsilon_9 \varepsilon_{10} + \varepsilon_{10} \left(\varepsilon'_{10} \right) = \varepsilon_{10} \left(\varepsilon_9 + x_{10}^{d'} - \left(-\frac{U_1}{m} U_y \right) \right)$$

$$\varepsilon_9 + x_{10}^{d'} + \frac{U_1}{m} U_y = -K_{10} \varepsilon_{10} \Rightarrow \frac{U_1}{m} U_y = -\varepsilon_9 - x_{10}^{d'} - K_{10} \varepsilon_{10}$$

$$U_y = \frac{m}{U_1} \left[-\varepsilon_9 - K_{10} \varepsilon_{10} - K_9 x_{10} \right]$$

3.6. Control of the Position x

The last subsystem is represented by Equation (10) below:

$$\begin{cases} x'_{11} = x_{12} \\ x'_{12} = U_x \frac{U_1}{m} \end{cases} \tag{10}$$

step1

Name ε_{11} the error between the desired and actual position x , such that

$$\varepsilon_{11} = x_{11}^d - x_{11} \Rightarrow \varepsilon'_{11} = x_{11}^{d'} - x'_{11} \tag{11}$$

The Lyapunov function is $V(\varepsilon_{11}) = \frac{1}{2} \varepsilon_{11}^2 \Rightarrow v'(\varepsilon) = \varepsilon_{11} \varepsilon'_{11} = \varepsilon_{11} (x_{11}^{d'} - x_{12}) < 0$

Thus, $x_{11}^{d'} - x_{12} = -K_{11} \varepsilon_{11} \Rightarrow x_{12}^d = x_{11}^{d'} + K_{11} \varepsilon_{11}$

step2

The error is defined by: $\varepsilon_{12} = x_{12}^d - x_{12} \Rightarrow x_{12} = x_{12}^d - \varepsilon_{12}$ its derivative is $\varepsilon'_{12} = x_{12}^{d'} - x'_{12}$

$$V_{12} = V_{11} + \frac{1}{2} \varepsilon_{12}^2 \Rightarrow V' = \varepsilon'_{11} + \varepsilon_{12} \varepsilon'_{12}$$

$$\varepsilon'_{11} = x_{11}^{d'} - x_{12}^d + \varepsilon_{12} \Rightarrow \varepsilon_{11} \varepsilon'_{11} = \varepsilon_{11} (x_{11}^{d'} - x_{12}^d + \varepsilon_{12}) + \varepsilon_{12} \varepsilon'_{12}$$

$$\varepsilon_{11} + x_{12}^d + \frac{U_1}{m} U_x = -K_{12} \varepsilon_{12}$$

$$\frac{U_1}{m} U_x = -\varepsilon_{11} - x_{12}^d - K_{12} \varepsilon_{12}$$

$$U_x = \frac{m}{U_1} \left[-\varepsilon_{11} - K_{12} \varepsilon_{12} - K_{11} x_{12} \right]$$

All the previous steps of the backstepping control, used to generate a global control law for the whole system, are summarized in the block diagram shown in Figure 2.

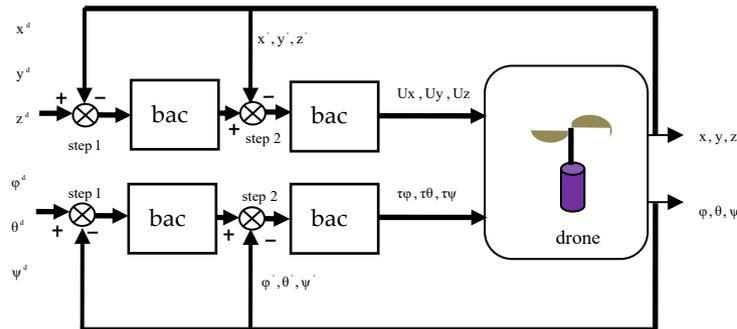


Figure 2. System block diagram.

4. Results and Discussions

In order to validate our proposed control solution, the model is simulated under Matlab Simulink software. For that purpose, the results are obtained based on the application of the real parameters summarized on Table 1 [2].

Table 1. Quadrotor parameter used in our simulation [2].

Gravitational Acceleration	g	9.8 m/s^2
Mass of quadrotor	m	0.4794 Kg
Length of wings	l	0.225 m
Rotational inertia	J_x	0.0086 Kg m^2
	J_y	0.0086 Kg m^2
	J_z	0.0172 Kg m^2
Residual inertia	J_r	$3.740 \times 10^{-5} \text{ Kg m}^2$

In this scenario, it is desired to follow a circular trajectory on the XY plane, centered in the origin. The height z increases uniformly from zero to 15 m (as shown in Figure 3b), where the drone stabilizes.

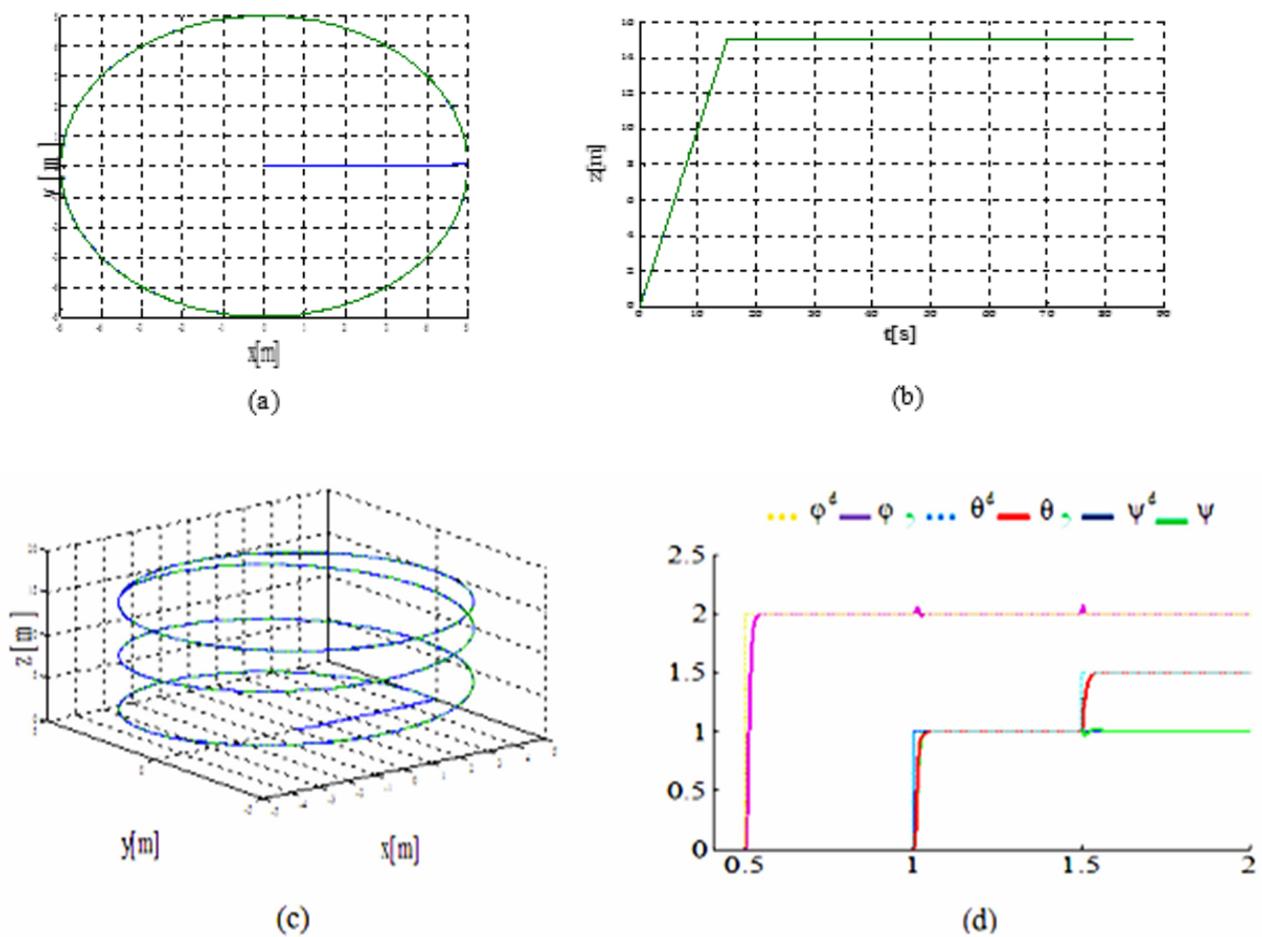


Figure 3. Simulation results. (a) xy drone control. (b) Uniform increase in z height from zero to 15 m. (c) Real and the desired positions exactly meet each other. (d) Orientation angles' response.

Figure 3a shows the response of the drone controller to the xy desired value; where we see that the estimated value follows the set-point perfectly. Figure 3c illustrates that the real and the desired positions exactly meet each other in three-dimensional space.

Figure 3d illustrates the response of orientation angles (roll, pitch, and yaw), where the dotted lines denote the desired values and continues-lines shows the estimated values. It is clearly demonstrated that the estimated values track the desired trajectories with an acceptable dynamic.

5. Conclusions

In this paper, a backstepping control is used to provide the dynamic control of the quadrotor. The models derived in this paper are used to design and implement control laws for six-DOF quadrotor stability. For this purpose, firstly, the mathematical model of the quadrotor was developed. Secondly, the backstepping control strategy was devised to control the position and orientation of the quadrotor subsystem. Several scenarios were performed to examine the performance of the backstepping strategy, and we noticed that the simulation results showed the effectiveness of the proposed control. For further work, this approach will be implemented on a quadrotor that will be applied for monitoring and fault diagnostic on a multi-agent-based smart grid [18].

Author Contributions: Conceptualization, A.S. and H.B.; methodology, H.B. and A.S.; software, A.S.; validation, A.S., H.B. and R.B.; formal analysis, A.S.; investigation, A.S.; resources, A.S.; data curation, A.S.; writing—original draft preparation, A.S.; writing—review and editing, A.S., R.B. and H.B.; visualization, A.S.; supervision, R.B.; project administration, H.B.; funding acquisition, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the “la Direction Générale de la Recherche Scientifique et du Développement Technologique (DGRSDT)” for its financial support.

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

References

1. Alsamhi, S.H.; Ma, O.; Ansari, M.S.; Almalki, F.A. Survey on Collaborative Smart Drones and Internet of Things for Improving Smartness of Smart Cities. *IEEE Access* **2019**, *7*, 128125–128152. [[CrossRef](#)]
2. Shahid, F.; Kadri, B.; Jumani, N.A.; Pirwani, Z. Dynamical Modeling and Control of quadrotor. *Trans. Mach. Des.* **2016**, *4*, 50–63.
3. Tuan, L.L.; Won, S. PID based sliding mode controller design for the micro quadrotor. In Proceedings of the 2013 13th International Conference on Control, Automation and Systems (ICCAS 2013), Gwangju, Korea, 20–23 October 2013; pp. 1860–1865. [[CrossRef](#)]
4. Erginer, B.; Altug, E. Modeling and PD Control of a Quadrotor VTOL Vehicle. In Proceedings of the 2007 IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, 13–15 June 2007; pp. 894–899. [[CrossRef](#)]
5. Bouadi, H.; Bouchoucha, M.; Tadjine, M. Sliding Mode Control based on Backstepping Approach for an UAV Type-Quadrotor. *World Acad. Sci. Eng. Technol. Int. J. Mech. Mechatron. Eng.* **2007**, *1*, 2. [[CrossRef](#)]
6. Zahran, S.; Moussa, A.; El-Sheimy, N. Enhanced Drone Navigation in GNSS Denied Environment Using VDM and Hall Effect Sensor. *Int. J. Geo Inf.* **2019**, *8*, 169. [[CrossRef](#)]
7. Labbadi, M.; Cherkaoui, M.; Houm, Y.E.; Guisser, M. Modeling and Robust Integral Sliding Mode Control for a Quadrotor Unmanned Aerial Vehicle. In Proceedings of the 2018 6th International Renewable and Sustainable Energy Conference (IRSEC), Rabat, Morocco, 5–8 December 2018; pp. 1–6. [[CrossRef](#)]
8. Mohamed, H.A.F.; Yang, S.S.; Moghavvemi, M. Sliding mode controller design for a flying quadrotor with simplified action planner. In Proceedings of the 2009 ICCAS-SICE, Fukuoka, Japan, 18–21 August 2009; pp. 1279–1283.
9. Bouadi, H.; Cunha, S.S.; Drouin, A.; Mora-Camino, F. Adaptive sliding mode control for quadrotor attitude stabilization and altitude tracking. In Proceedings of the 2011 IEEE 12th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, 21–22 November 2011; pp. 449–455. [[CrossRef](#)]
10. Bouabdallah, S.; Siegwart, R. Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 2247–2252. [[CrossRef](#)]
11. Mian, A.; Daobo, W. Modeling and backstepping-based nonlinear control strategy for a 6 dof quadrotor helicopter. *Chin. J. Aeronaut.* **2008**, *21*, 261–268. [[CrossRef](#)]
12. Lee, D.; Kim, H.J.; Sastry, S. Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter. *Int. J. Control. Autom. Syst.* **2009**, *7*, 419–428. [[CrossRef](#)]
13. McKerrow, P. Modelling the Draganflyer four-rotor helicopter. In Proceedings of the IEEE International Conference on Robotics and Automation, 2004 Proceedings. ICRA '04. 2004, New Orleans, LA, USA, 26 April–1 May 2004; Volume 4, pp. 3596–3601. [[CrossRef](#)]

14. Hanani, N.; Syazwanadira, F.; Fakharulrazi, N.A.; Yakub, F.; Rasid, Z.A.; Sarip, S. Full Control of Quadrotor Unmanned Aerial Vehicle using Multivariable Proportional Integral Derivative Controller. In Proceedings of the 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET), Shah Alam, Malaysia, 7 October 2019; pp. 447–452. [[CrossRef](#)]
15. Wang, P.; Man, Z.; Cao, Z.; Zheng, J.; Zhao, Y. Dynamics modelling and linear control of quadcopter. In Proceedings of the 2016 International Conference on Advanced Mechatronic Systems (ICAMechS), Melbourne, VIC, Australia, 30 November–3 December 2016; pp. 498–503. [[CrossRef](#)]
16. Glida, H.E.; Abdou, L.; Chelihi, A.; Sentouh, C.; Hasseni, S.-E.-I. Optimal model-free backstepping control for a quadrotor helicopter. *Nonlinear Dyn.* **2020**, *100*, 3449–3468. [[CrossRef](#)]
17. Basri, M.A.M.; Noordin, A. Optimal backstepping control of quadrotor UAV using gravitational search optimization algorithm. *Bull. Electr. Eng. Inform.* **2020**, *9*, 1819–1826. [[CrossRef](#)]
18. Hadjira, B.; Rabiai, Z. Decentralized Energy Management System Enhancement for Smart Grid. In *Optimizing and Measuring Smart Grid Operation and Control*; Reoui, A., Bentarzi, H., Eds.; IGI Global: Hershey, PA, USA, 2021; pp. 156–169. [[CrossRef](#)]