Mechanical Modeling, Control and Simulation of a Quadrotor UAV

Hamid Hassani, Anass Mansouri, and Ali Ahaitouf

Abstract This paper presents the development of a quadrotor 3D-Model based on a new approach integrating both the flight controller and the quadrotor CAD-model (Computer-aided design). The quadrotor design is performed using CAD modelling environment, then imported to MATLAB Simscape for the design of the control scheme based on the PID (Proportional-Integral-derivative) controller. The effectiveness of the offered flight simulator system is tested using several predefined trajectories and the simulation results of each trajectory emphasize the accuracy of the proposed simulator.

Keywords Quadrotor · 3D-Model · Flight controller · MATLAB/simscape · PID

1 Introduction

Quadrotor or quadcopter, known also as drone, has become one of the most attractive research topics. Mainly due to their ability to achieve autonomously multitude tasks, even in cluttered places. Recently, the use of this robot has been widened to cover new missions involving autonomous delivery, mapping and image acquisition [1]. Generally, quadrotor is an unstable flying robot suffering from the non-controllability in the lateral motions (x, y), its rotational and translational dynamics are under-actuated and strongly coupled [2]. Several stabilization approaches have been introduced to deal with these limitations. Among them, the PID controller has taken more attention, it has proven by many researchers especially in the practical implementation [3]. Additionally, nonlinear techniques such as sliding mode and backstepping controllers have been successfully proved in both simulations and experiments [4].

H. Hassani (✉) · A. Ahaitouf
Laboratory of Intelligent Systems, Geo-Resources and Renewable Energies, Faculty of Sciences and Technology, Sidi Mohammed Ben Abdellah University, Fez, Morocco
e-mail: Hamid.hassani@usmba.ac.ma

A. Mansouri
Laboratory of Intelligent Systems, Geo-Resources and Renewable Energies, School of Applied Sciences, Sidi Mohammed Ben Abdellah University, Fez, Morocco

B. Hajji et al. (eds.), *Proceedings of the 2nd International Conference on Electronic Engineering and Renewable Energy Systems*, Lecture Notes in Electrical Engineering 681, https://doi.org/10.1007/978-981-15-6259-4_47

Recently, simulation platforms have been used in a great number of application domains such as, automotive systems, autonomous underwater robots and aerial applications [5]. There is a real need to visualize the quadrotor behavior in a virtual environment before performing physically the flight experimentation where the fail is forbidden. Herein we discuss the development of a quadrotor flight simulator based on a new approach that involves both the mechanical model and the flight controller. This simulator will be used as a virtual prototyping for drone model validation, it can be also adopted to simplify the challenging task when tuning the flight controller.

To scrutinize the contribution of this paper, we organize it as follows. In the second section, a general overview on the quadrotor system as well as the nonlinear dynamic model computation are discussed. The core of the flight simulator is presented in the third section. The simulator results are presented in section 5. Finally, the last section concludes the work and introduces some future directions of this study.

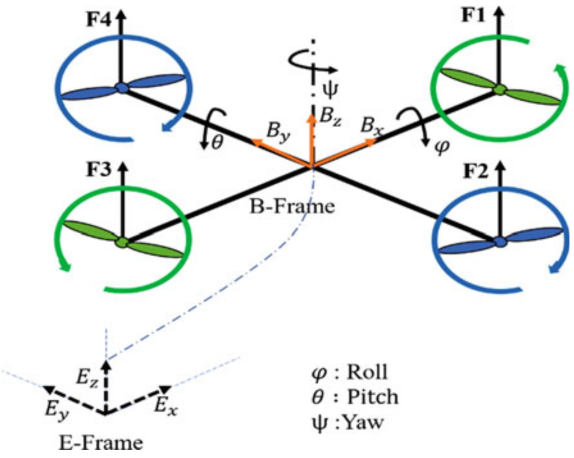
2 Quadrotor System

2.1 Quadrotor Description

Quadrotor is an aerial robot powered by four identical rotors arranged in a plus or cross configuration. It consists of two pair's propeller-rotor, as shown in Fig. 1. The first pair (M_1, M_3) spin in the counter clockwise direction, while the second pair (M_2, M_4) rotates in the clockwise direction. This configuration makes the reactive force produced by each propeller-Rotor being cancelled.

Quadrotors are controlled by the thrust forces produced by four identical rotors. However, the rotation around x axes called roll motion can be achieved by inversely changing the speed of the pair (M_2, M_4), also the pitch movement is obtained by

Fig. 1 Quadrotor configuration



creating an unbalance between the generated force by rotors M_1 and M_3. Yaw motion is screamed by changing the counter-torque between each pair propeller-rotor. The vertical flight is obtained by simultaneously varying the speed of all rotors.

2.2 Quadrotor Model

The obsession behind the quadrotor modeling studies is the design of a near-realistic model. This can be achieved using the Newton Euler methodology [6] or the Lagrange Euler formula. In this work, the quadrotor system is modeled using the Newtonian formalism, and the modeling phase is based on the following assumptions:

- The quadrotor body frame and the propellers are rigid.
- The quadrotor structure is symmetrical.
- The thrust and drag forces are proportional to the square of the rotor speed.

The quadrotor translational and rotational equations of motions are as follows:

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} \frac{U_1}{m} U_x \\ \frac{U_1}{m} U_y \\ \frac{U_1}{m} (cos\varphi \; cos\theta) - g \\ \dot{\theta}\dot{\psi}\left(\frac{I_y - I_z}{I_y}\right) - \frac{J_r}{I_x}\dot{\theta}\Omega + \frac{U_2}{I_x} \\ \dot{\varphi}\dot{\psi}\left(\frac{I_z - I_x}{I_y}\right) + \frac{J_r}{I_y}\dot{\varphi}\Omega + \frac{U_3}{I_y} \\ \dot{\varphi}\dot{\theta}\left(\frac{I_x - I_y}{I_z}\right) + \frac{U_4}{I_z} \end{pmatrix} \tag{1}$$

With

$$\begin{cases} U_1 = b\left(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2\right) \\ U_2 = bl\left(\omega_4^2 - \omega_2^2\right) \\ U_3 = bl\left(\omega_3^2 - \omega_1^2\right) \\ U_4 = d\left(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2\right) \end{cases} \tag{2}$$

$$\begin{cases} U_x = \cos\varphi \;\; \sin\theta \cos\psi + \sin\;\varphi\;\; \sin\psi \\ U_y = \cos\varphi \;\; \sin\theta \sin\;\psi - \sin\;\varphi\;\; cos\psi \\ \Omega = \omega_4 + \omega_3 - \omega_2 - \omega_1 \end{cases} \tag{3}$$

Where:

- *(x, y, z)* and *(φ, θ, ψ)* denote the position and orientation coordinates.
- $I\,(I_x, I_y, I_z)$ is the diagonal matrix of inertia and m is the quadrotor total mass.
- J_r is the rotor inertia, b and d symbolize the thrust and drag constant.
- ω_i and U_i are respectively the rotors speed and the input signals.

The vertical movement and the quadrotor orientation (φ, θ, ψ) are directly controlled by the control efforts U_1, U_2, U_3 and U_4. In contrast, the horizontal displacements (x, y) is indirectly controlled by selecting the appropriate roll and pitch angles.

Based on Eq. (3), the desired pitch and roll angles are calculated as:

$$\theta_d = \arctan\left(\frac{U_x\cos(\psi) + U_{\mathcal{Y}}\sin(\psi)}{U_z}\right);\ \varphi_d = \arctan\left(\cos(\theta_d)\left(\frac{U_x\sin(\psi) - U_y\cos(\psi)}{U_z}\right)\right) \tag{4}$$

With

$$U_z = (\cos(\varphi)\cos(\theta))U_1\Big/m \tag{5}$$

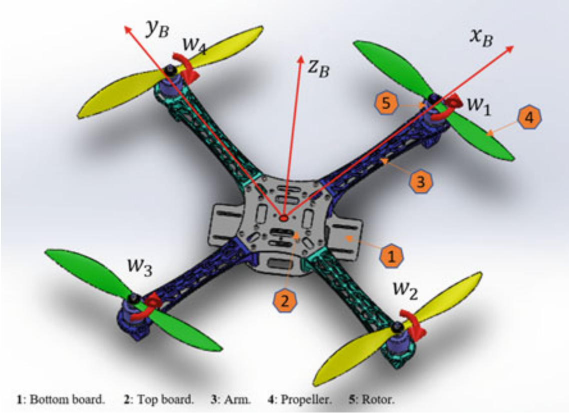
3 Proposed Flight Simulator

The quadrotor mechanical model is designed and validated using SolidWorks CAD. The design phase is done based on specifications listed in reference [7]. The designed components are imported to Simscape tools to generate the compatible file with MATLAB environment. Finally, the design of the control scheme is done using MATLAB/Simulink.

3.1 Quadrotor CAD-Models Design

The idea is to build each component of the quadrotor system in SolidWorks tools, this phase involves the 3D drawing, the visual appearance and the material specification. Next, the designed components must be correctly connected to shape the quadrotor assembly as shown in Fig. 2.

Fig. 2 Quadrotor SolidWorks assembled CAD-models


y_B
w_4
z_B
x_B
5
w_1
4
3
1
w_3
2
w_2
1: Bottom board. 2: Top board. 3: Arm. 4: Propeller. 5: Rotor.

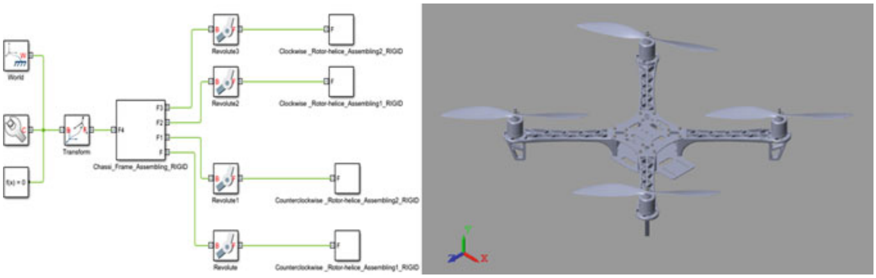

World
Transform
Chassi_Frame_Assembling_RIGID
F4
F3
F2
F1
F
f(x) = 0
Revolute3
Clockwise _Rotor-helice_Assembling2_RIGID
Revolute2
Clockwise _Rotor-helice_Assembling1_RIGID
Revolute1
Counterclockwise _Rotor-helice_Assembling2_RIGID
Revolute
Counterclockwise _Rotor-helice_Assembling1_RIGID
Y
X

Fig. 3 MATLAB simscape CAD-models

Table 1 Quadrotor flight controller parameters

Quadrotor motions	P	I	D
Attitude (φ; θ; ψ)	(1.5;1.1;6.5)	(0.005;0.008;0.9)	(0.04;0.037;0.025)
Position (X; Y)	(340; 350)	(60; 60)	(0.001; 0.001)
Altitude Z	10	15	1.9

Once the quadrotor model is properly designed, then correctly assembled and verified using the SolidWorks motion analysis. We use Simscape multibody link to generate the compatible files with MATLAB environment as illustrated in Fig. 3. The generated files involve the quadrotor mechanical model, the gravitational force and the system parameters (mass and inertial properties).

3.2 Control Strategy

The control strategy adopted in this work is a hierarchical controller having three loops, a full actuated inner loop that controls roll, pitch and yaw movements. An under actuated outer loop that controls the lateral position (x, y), and an altitude loop which controls the vertical flight. The PID controller is adopted for the control of the three loops using the configuration listed in Table 1. The synoptic scheme of the control strategy is shown in Fig. 4.

The near-optimal gains of the PID controller valid in our case, are selected based on empirical tests inspired by work [5], where a thoroughly description of the PID parameters is presented.

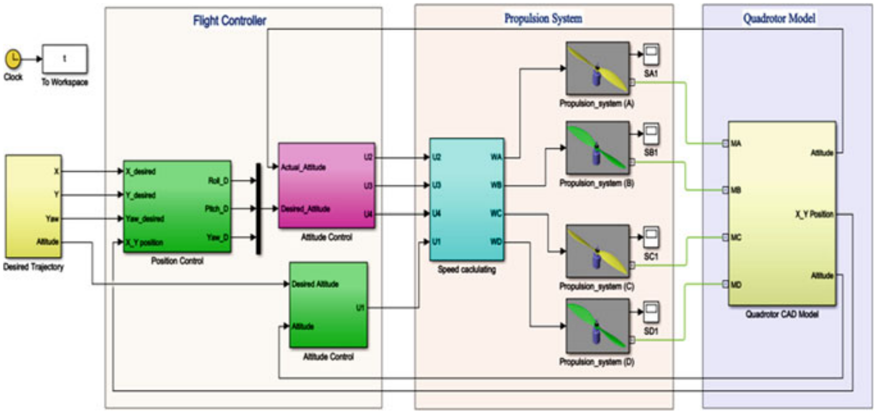


Fig. 4 Drone simulator structure

4 Simulator Environment

The main concept of the proposed flight simulator is illustrated in Fig. 4. However, the drone user defines the desired trajectory and then the control process begins. First, the position controller manages the quadrotor lateral behavior (x-y plane), and produces the desired roll and pitch angles which serve as a setpoint for the attitude controller Eq. (4). The altitude controller produces the sufficient thrust allowing the quadrotor to reach the desired z position. The controller's outputs (U_1, U_2, U_3 and U_4) are used to calculate the speed of each rotor based on the inverse transformation of Eq. (2). The propulsion system generates the needed forces to achieve the expected flight. The drone localization is obtained through six sensors, position (x, y, z) and orientation (φ, θ, ψ), added to mimic the role of the Inertial Measurement Unit (IMU).

5 Results and Discussions

Two different simulations were introduced to highlight the accuracy of the developed simulator. The fist, concerns the problem of stabilization, while the second simulation evaluates the ability of the quadrotor in path tracking missions.

Case 1: Hovering Mode

The simulation is carried out during 5 s which is sufficient to reach the desired altitude 1 m, with an initial configuration for the quadrotor attitude fixed at (0.2, 0.2, 0.2) rad.

As depicted in Fig. 5, the PID controller is able to push the quadrotor to achieve the desired altitude 1 m within less than 1 s. Also, the quadrotor orientation is maintained to zero rapidly. Furthermore, the offered simulator provides a 3D-visualization of the quadrotor motions. Figure 6 illustrates the Front, Top and Isometric views of the quadrotor system in the simulator environment when performing the hovering flight.

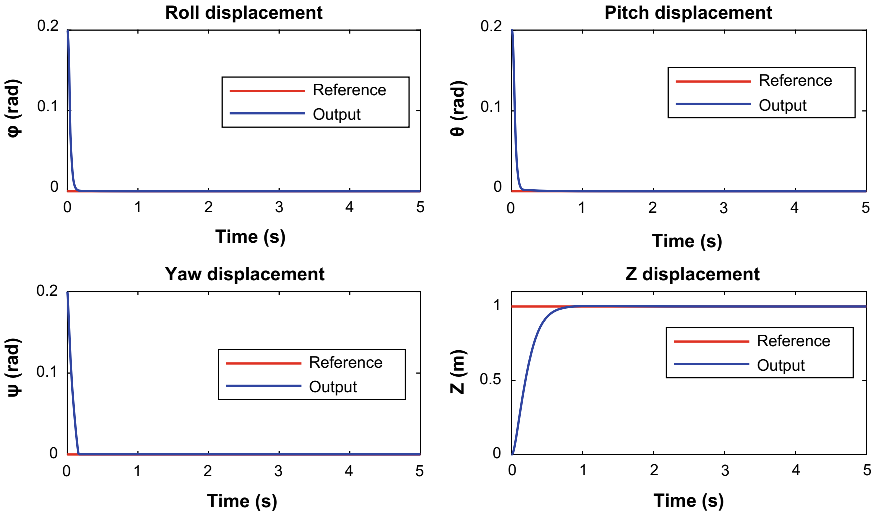


Fig. 5 Drone-simulator attitude and altitude stabilization

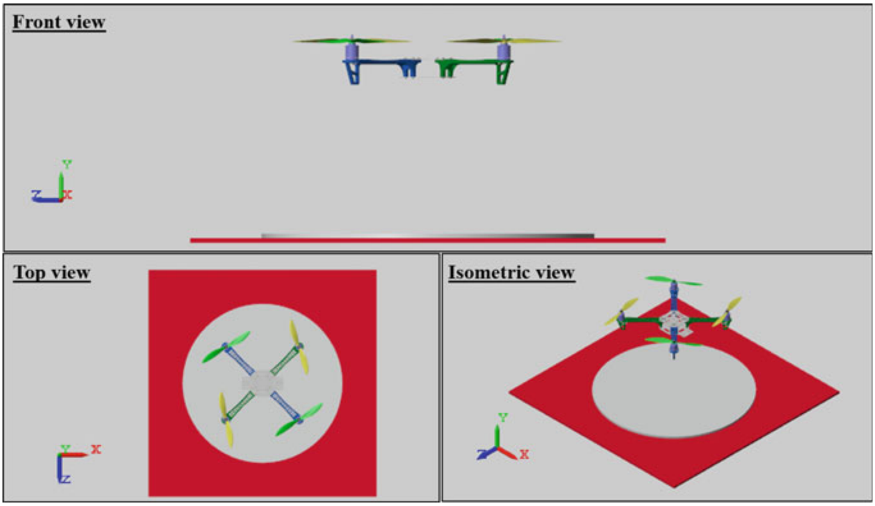


Fig. 6 Quadrotor views in MATLAB simscape interface

Case 2: Path Tracking In order to check the trajectory tracking ability of the offered simulator, four flight tests have been performed. Figure 7 shows the 3D-representation of the real and the desired trajectories in the simulator environment using square trajectory (a), cross trajectory (b), Lemniscate trajectory (c) and circular trajectory (d).

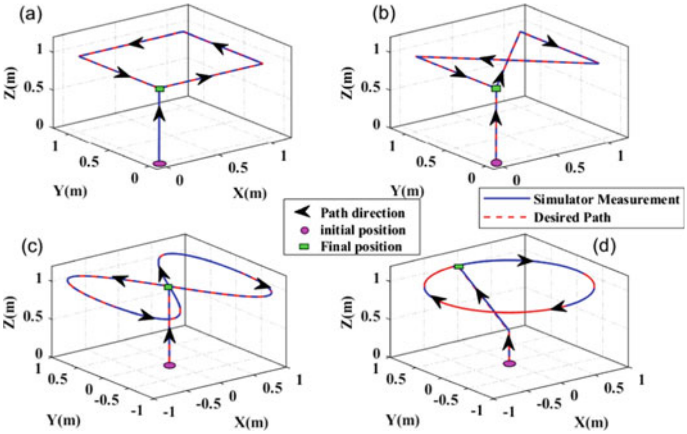


Fig. 7 Real and desired path for the square, cross, lemniscate and circular trajectory

As seen in the 3D-representation (Fig. 7), the quadrotor is well controlled and follows accurately the planned path. Even if the reference trajectory is suddenly changed, the quadrotor is able to accomplish the flight mission successfully. It should be mentioned that the adopted hierarchical methodology based on the PID controller has effectively controlled the quadrotor UAV in different flight tests.

6 Conclusion

In this paper, based on the quadrotor mechanical design a quadrotor simulator is developed. A hierarchical methodology based on the PID controller is adopted to solve the under-actuation and to steer the quadrotor behaviors in the simulator environment, contributing promising results in both stabilization and path following. The present simulator can be used as test platform to evaluate the quadrotor behaviors under several external conditions. It can be also used to configure the flight controller before moving on to the real time experimentation. In future work, robust adaptive controller will be used to enhance the flight ability in the presence of external perturbations.

References

1. Calì M, Ambu R (2018) Advanced 3D photogrammetric surface reconstruction of extensive objects by UAV camera image acquisition. Sensors 18:2815
2. Wang Y, Jiang B, Ningyun L, Pan J (2016) Hybrid modeling based double-granularity fault detection and diagnosis for quadrotor helicopter. Nonlinear Anal Hybrid Syst 21:22–36

3. Demir BE, Bayir R, Duran F (2016) Real-time trajectory tracking of an unmanned aerial vehicle using a self-tuning fuzzy proportional integral derivative controller. Int J Micro Air Veh 8(4):252–268
4. Mian AA, Daobo W (2008) Modeling and backstepping-based nonlinear control strategy for a 6 DOF quadrotor helicopter. Chin J Aeronaut 21:261–268
5. Yasmina B, Mansouri A, Ahaitouf A (2018) Quadrotor flight simulator modeling. In: International conference on advanced intelligent systems for sustainable development, pp 665–674. Springer, Heidelberg
6. Hassani H, Mansouri A, Ahaitouf A (2019) Control system of a quadrotor UAV with an optimized backstepping controller. In: 2019 International conference on intelligent systems and advanced computing sciences (ISACS), pp 1–7
7. Flame wheel F450 manual (2015). http://dl.djicdn.com/downloads/flamewheel/en/F450_User_Manual_v2.2_en.pdf