Study and Development of Robust Control Systems for Educational Drones

Maria Letizia Corradini, Gianluca Ippoliti, Giuseppe Orlando,
and Simone Terramani

Abstract This paper considers the problem of attitude and altitude control of quadrotors using the sliding mode control theory. The mathematical model of the quadrotor is derived using the Euler-Newton formalism. The sliding-mode is applied to the Parrot Mambo minidrone, which is a strong example of bringing educational robotics to formal (MATLAB, Python, JavaScript), non-formal (Tynker, Blockly, Swift Playground) and informal education. The control considered shows good performance and enhanced robustness.

Keywords Robust Control · Drones · Sliding-Mode Control · Education · Robotics

1 Introduction

The Parrot Mambo minidrone is an example of a new way of bringing the teaching of STEM (science, technology, engineering, mathematics) subjects into the classroom [1]. It offers a variety of tools and approaches that make it suitable for students from primary school to PhD level. In fact, it supports the best coding platforms, providing the end user with plenty of choices for tackling heterogeneous STEM problems. In particular, for primary schools it offers block coding with platforms like Tynker and

M. L. Corradini (✉)

School of Science and Technology, University of Camerino, Via Madonna delle Carceri, 62032
Camerino, MC, Italy
e-mail: letizia.corradini@unicam.it

G. Ippoliti · G. Orlando · S. Terramani

Dipartimento di Ingegneria dell'Informazione, Università Politecnica Delle Marche, Via Brecce
Bianche, 60131 Ancona, Italy
e-mail: gianluca.ippoliti@univpm.it

G. Orlando

e-mail: giuseppe.orlando@univpm.it

S. Terramani

e-mail: S1085358@studenti.univpm.it

© The Author(s) 2021

D. Scaradozzi et al. (eds.), *Makers at School, Educational Robotics and Innovative Learning Environments*, Lecture Notes in Networks and Systems 240,
https://doi.org/10.1007/978-3-030-77040-2_40

301

Blocky; for secondary schools and universities it offers text coding with JavaScript and Python and also compatibility with MATLAB & Simulink [2]. Simulink is the environment used to implement a robust nonlinear control algorithm (sliding mode control), in order to demonstrate the flexibility and the versatility of this novel educational ecosystem: from block coding to state-of-the-art prototyping, implementation and testing of experimental control approaches. This paper considers a mathematical model of the Parrot Mambo, which was used to modify the classical sliding mode control algorithm in order to get enhanced robustness against disturbances.

2 Quadrotor Mathematical Model

As shown in Fig. 1, the quadrotor contains four motors placed in a cross-configuration (front and rear propeller rotate counterclockwise, while the right and the left ones rotate clockwise) [3]. Each rotor is linked to a propeller. The quadrotor is an underactuated system: in fact, it has 6 *DOF* (degree of freedom), three Cartesian coordinates (X, Y, Z) and three attitude angles, ϕ, θ and ψ (roll, pitch and yaw, respectively) but only four rotors/motors. However, it is possible to choose four controls to set the altitude (vertical distance from the ground) and the attitude (angular position) of the drone, that is, *Throttle* U_z : vertical force obtained by increasing or decreasing the speed of the four propellers by the same amount; *Roll* U_ϕ : torque with respect to the x axis of the body frame, obtained by increasing or decreasing the speed of either the left or the right propellers; *Pitch* U_θ : torque with respect to the y axis of the body frame, obtained by increasing or decreasing the speed of either the front or the rear propellers; *Yaw* U_ψ : torque with respect to the z axis of the body frame, obtained by increasing (or decreasing) the speed of both the front and the rear propellers, while decreasing (or increasing) both the left and the right propellers. Two coordinate frames are used to describe the motion of the drone: the earth-centered inertial frame and the body-fixed frame. The earth frame ($O_e x_e y_e z_e$) is used to define the

Fig. 1 Quadrotor structure representation

position vector of the quadrotor $\xi = [X \ Y \ Z]^T$ expressed in meters and the vector of Euler angles $\sigma = [\phi \ \theta \ \psi]^T$ expressed in radians. The body frame $(O_b \ x_b \ y_b \ z_b)$ defines the vector of linear velocities $\mu = [u \ v \ w]^T$ expressed in meters per seconds, the vector of angular velocities $\omega = [P \ Q \ R]^T$ expressed in radians per seconds, the forces F expressed in newton and the torques τ expressed in newton meters. The dynamic model was derived using the Newton–Euler formalism. As seen in [4], the dynamics of the 6 DOF rigid body under the effect of external forces and torques are:

$$\begin{bmatrix} mI_{3,3} & 0 \\ 0 & J \end{bmatrix} \begin{bmatrix} \dot{\mu} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \omega \times m\omega \\ \omega \times J\omega \end{bmatrix} = \begin{bmatrix} F \\ \tau \end{bmatrix} \quad (1)$$

In order to develop Eq. (1), the different forces and torques that affect the quadrotor need to be determined. The main forces that affect the system are:

- *Input forces and torques*: directly produced by the four rotors. As a result of rotation, each motor exhibits an aerodynamic moment M_i and a lift force F_i given by [5]:

$$F_i = C_T \Omega_i^2, \quad M_i = C_q \Omega_i^2 \quad (2)$$

where C_T and C_q are the thrust and the torque coefficient respectively, Ω_i is the velocity of the i^{th} rotor. As reported in [6], the resultant input torque vector is:

$$\tau_{in} = \begin{bmatrix} U_\phi \\ U_\theta \\ U_\psi \end{bmatrix} = \begin{bmatrix} C_T d(\Omega_4^2 - \Omega_2^2) \\ C_T d(\Omega_3^2 - \Omega_1^2) \\ C_q(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{bmatrix}. \quad (3)$$

The vector of input forces with respect to the body frame is:

$$F_{in}^B = \begin{bmatrix} 0 \\ 0 \\ U_Z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ C_T(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{bmatrix}. \quad (4)$$

- *Gyroscopic effect*: this is produced by rotation of the propeller and it is given as:

$$\tau_g = \left[-\sum_{i=1}^4 J_{tp} \left[\omega \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right] - (-1)^i \Omega_i \right] = \begin{bmatrix} -Q\Omega J_{tp} \\ P\Omega J_{tp} \\ 0 \end{bmatrix} \quad (5)$$

where J_{tp} is the total rotational moment of inertia around the propeller axis and Ω is the sum ($\Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$) of the four propeller speeds expressed in radians per seconds.

- *Gravitational force*:

$$F_G = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (6)$$

where g is the gravitational acceleration and m the mass of the drone.

Replacing Eqs. (2)–(6) into Eq. (1), and expressing the rotational subsystem with respect to the earth frame, the dynamics equations become [6]:

$$\begin{cases} \ddot{X} = (S_\phi S_\psi + C_\psi S_\theta C_\phi) \frac{U_z}{m} \\ \ddot{Y} = -(S_\phi C_\psi + S_\psi S_\theta C_\phi) \frac{U_z}{m} \\ \ddot{Z} = -g + C_\theta C_\phi \frac{U_z}{m} \\ \ddot{\phi} = \frac{1}{J_{xx}} (U_\phi + (J_{yy} - J_{zz}) \dot{\theta} \dot{\psi} + \dot{\theta} \Omega J_{tp}) \\ \ddot{\theta} = \frac{1}{J_{yy}} (U_\theta + (J_{zz} - J_{xx}) \dot{\phi} \dot{\psi} - \dot{\phi} \Omega J_{tp}) \\ \ddot{\psi} = \frac{1}{J_{zz}} (U_\psi + (J_{xx} - J_{yy}) \dot{\phi} \dot{\theta}) \end{cases} \quad (7)$$

3 Sliding Mode Controller

The sliding mode control technique consists of a high-speed switching control law that forces the state trajectories to follow a manifold called switching surface, and keeps the trajectories on this surface [7]. Sliding mode control design is a two-step procedure: sliding surface synthesis to provide the desired performance, and control design to ensure that all motions slide along the desired surfaces. The sliding mode control (SMC) is composed of equivalent control and corrective control [8]. The former ensures that the trajectories stay on the surface. The latter ensures that the trajectories reach the surface and compensates for any variations around the surface, whether due to external disturbances or to unmodeled dynamics. The sliding control for a given state is:

$$U_i = U_i^{eq} + U_i^C \quad (8)$$

where U_i^{eq} is the equivalent control and U_i^C is the corrective control. As in [9], let the sliding surface be of the form:

$$S_i = \dot{e}_i + \alpha_i e_i \quad (9)$$

where S_i is the sliding surface for a given state, e_i is the error between the actual state and the desired value, and α_i is a positive proportional gain. As for most first order systems, the chosen Lyapunov function is:

$$V_i = \frac{1}{2} S_i^2. \quad (10)$$

To ensure system stability, the time derivative of the Lyapunov function ($\dot{V}_i = S_i \dot{S}_i$) must strictly be negative. To ensure that, we can set [6]:

$$\dot{S}_i = -\beta_i S_i - \lambda_i \text{sign}(S_i). \quad (11)$$

by the corrective control. As is well known, sliding mode control is robust against matched bounded disturbances. However, as proposed in [6], a more realistic assumption on unmodeled dynamics and external disturbances can be represented as:

$$d_i = \rho_i |S_i|^{p_i} |U_i|^{v_i} + d_{0i} \quad (12)$$

where $0 < v_i < 1$, $0 \leq \rho_i \leq 1$, p_i is a positive real number and d_{0i} is a bounded disturbance. With corrective control the surface dynamics take the form [6]:

$$\dot{S}_i = (-\beta_i |S_i|^{m_i} - \lambda_i) \text{sign}(S_i) + d_i \quad (13)$$

where m_i is a positive integer. To respect the condition of a strictly negative derivative of Eq. (10) we can take [6]:

$$m_i = \frac{p_i + 1 - v_i}{1 - v_i}. \quad (14)$$

and the control algorithm is [6]:

$$\left\{ \begin{array}{l} U_z = \frac{m}{\cos \theta \cos \phi} \left(\ddot{Z}^d + g + \alpha_z \ddot{e}_z + \beta_z |S_z|^{\frac{p_z+1-v_z}{1-v_z}} \text{sign}(S_z) + \lambda_z \text{sign}(S_z) \right) \\ U_\phi = J_{xx} \left(\ddot{\phi}^d + \beta_\phi |S_\phi|^{\frac{p_\phi+1-v_\phi}{1-v_\phi}} \text{sign}(S_\phi) + \lambda_\phi \text{sign}(S_\phi) + \alpha_\phi \dot{e}_\phi \right) \\ \quad - (J_{yy} - J_{zz}) \dot{\theta} \dot{\psi} - \dot{\theta} \Omega J_{Ip} \\ U_\theta = J_{yy} \left(\ddot{\theta}^d + \beta_\theta |S_\theta|^{\frac{p_\theta+1-v_\theta}{1-v_\theta}} \text{sign}(S_\theta) + \lambda_\theta \text{sign}(S_\theta) + \alpha_\theta \dot{e}_\theta \right) \\ \quad - (J_{zz} - J_{xx}) \dot{\phi} \dot{\psi} + \dot{\phi} \Omega J_{Ip} \\ U_\psi = J_{zz} \left(\ddot{\psi}^d + \beta_\psi |S_\psi|^{\frac{p_\psi+1-v_\psi}{1-v_\psi}} \text{sign}(S_\psi) + \lambda_\psi \text{sign}(S_\psi) + \alpha_\psi \dot{e}_\psi \right) \\ \quad - (J_{xx} - J_{yy}) \dot{\phi} \dot{\theta} \end{array} \right. \quad (15)$$

4 Experimental Results

The performance of the control algorithm deployed on the Parrot Mambo drone are shown below. The reference trajectory is shown in red (dashed line) and the actual trajectory is shown in blue (continuous line). In Fig. 2, the weight of the drone has been modified (its weight has been increased by 16% compared with the nominal mass) and we can see that the control algorithm, with surface dynamics of Eq. (11), cannot compensate for this weight increase, since there is an offset of about 10 cm in altitude (see Fig. 2a). On the other hand, the control algorithm with surface dynamics of Eq. (13), after a transient, is able to reach the desired altitude (see Fig. 2b). In Fig. 3, the yaw angle is shown, and the drone is in an environment with disturbances that are created artificially (wind fan). In Fig. 3a, spikes and an oscillatory behavior are seen for the control algorithm with surface dynamics of Eq. (11). In Fig. 3b, the control algorithm, with surface dynamics of Eq. (13), eliminates the spikes, and the behavior of the drone does not differ more than 0.1 radians with respect to the reference signal.

(a) Control algorithm with surface dynamics of Eq. (11)

(b) Control algorithm with surface dynamics of Eq. (13)

Fig. 2 Altitude control

(a) Control algorithm with surface dynamics of Eq. (11)

(b) Control algorithm with surface dynamics of Eq. (13)

Fig. 3 Yaw control

5 Conclusions

This paper has proposed a sliding mode control to control the altitude and the attitude of the quadrotor. The derived quadrotor model and the controller have been tested on a real drone, and the results demonstrate the robustness and the effectiveness of the control algorithm. What has been shown so far also demonstrates the flexibility and versatility of this new educational ecosystem: from block programming (via Simulink) to prototyping, to implement and test cutting-edge experimental and research control approaches.

References

1. Parrot Educational.: (2019). <https://edu.parrot.com/>
2. Parrot Minidrones Support from Simulink.: (2019). <https://it.mathworks.com/hardware-support/parrot-minidrones.html>

3. Fum, W.Z.: Implementation of Simulink Controller Design on Iris+ Quadrotor. Naval Postgraduate School (2015). <https://calhoun.nps.edu/handle/10945/47258>
4. Murray, R.M., Sastry, S.S., Zexiang, L.: A Mathematical Introduction to Robotic Manipulation, 1st edn. CRC Press Inc., Boca Raton, FL, USA (1994)
5. Azzam, A., Wang, X.: Quad rotor arial robot dynamic modeling and configuration stabilization. In: 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010), vol. 1, pp. 438–444 (March 2010)
6. Abci, B., Zheng, G., Efimov, D., El Najjar, M.E.B.: Robust altitude and attitude sliding mode controllers for quadrotors. IFAC-PapersOnLine **50**(1), 2720–2725 (2017). <https://doi.org/10.1016/j.ifacol.2017.08.576>, <http://www.sciencedirect.com/science/article/pii/S2405896317309473>, 20th IFAC World Congress
7. Khalil, H.K.: Nonlinear Systems. Pearson, 3 edn (2001)
8. Singh, G.K., Hole, K.E.: Guaranteed performance in reaching mode of sliding mode controlled systems. Sadhana **29**(1), 129–141 (Feb 2004). <https://doi.org/10.1007/BF02707005>
9. Khalid, K.M.: Design and Application of Second Order Sliding Mode Control Algorithms. Ph.D. thesis. University of Leicester (2003). <http://hdl.handle.net/2381/30209>
10. Bresciani, T.: Modelling, Identification and Control of a Quadrotor Helicopter. Master's thesis. Lund University Libraries (2008). <https://lup.lub.lu.se/student-papers/search/publication/8847641>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.