

Flight Control of a Quadrotor by Backstepping/PID

Mohamed Amine SABRI ¹, Abdelkader BEKRI ², Abdeldjebar HAZZAB ³

¹⁾Science and Technology department, Tahri Mohamed University of Bechar Algeria

²⁾CAOSEE laboratory, Tahri Mohamed University of Bechar, BP, 417 Bechar (08000), Algeria

³⁾CR2ie, Cégep de Sept-Îles, Québec, Canada

Email of the corresponding author: sabriamine91@gmail.com

Received: March 16, 2026

Accepted for publication: June 20, 2026

Published: July 12, 2026

Abstract--- The quadrotor is an unmanned aerial vehicle (UAV) with four rotors, able to carry out a multitude of missions. This paper consists of establishing nonlinear laws of command to control the movement of this engine. Firstly, a dynamic model, taking care of the all the phenomenon which govern the movement of the helicopter, has been developed. Secondly two approaches of control one classic, the other uses the backstepping. Thereafter, a new strategy has been proposed to improve the performance of control. The control law of this technique is developed using backstepping based on the PID. The results of simulation show the performance of this command.

Keywords—Quadrotor; Dynamic model; Control; backstepping; PID

I. INTRODUCTION

The number of industrialists and academics interested in drones continues to rise, due to their multiple civilian and military applications. Indeed, these devices have the advantage of not exposing people to dangers while covering a wide range of missions [1]. Thus, unmanned aerial vehicle (UAV) could identify areas contaminated with radioactive, chemical, or biological agents without endangering people's lives. The objective of this paper is to contribute to the modeling and control of autonomous heavier-than-air flying vehicles, unmanned, quad-rotor type. From a theoretical standpoint, the difficulty of the problem is due to the complexity of highly nonlinear models with time-varying unknown parameters. Moreover, and unlike manipulator robots, drones are non-holonomic, under-actuated systems (the number of control inputs is less than the number of degrees of freedom of the system) with a state vector that is not fully accessible. These constraints make controlling quadrotors difficult; therefore, it is essential and imperative to develop controls capable of correcting inaccuracies and measurement errors, as well as the limitations imposed by real actuators [2],[3],[4]

The work consists of synthesizing control laws to enable the stabilization and control of this four-rotor flying platform.

A. Quadrotor Configuration

It's a special helicopter, small and agile, as its name suggests, contains four rotors located at the front, rear, left,

and right ends through a frame. It is capable of flying in vertical take-off and landing (VTOL), and also performing maneuvers in all directions thanks to the control of its four engines, as shown in the figure (1) [5].

B. Presentation of the X4 stationnary flyer (XSF)

The X4 stationnary flyer consists of four fixed-pitch two-blade propellers coupled to motors mounted at the ends of a rigid cross-shaped body. These motors operate on direct current supplied by a battery fixed at the center of the cross. In the same box located at the center, there are also sensors ensuring good stability and electronic boards that are lighter and more efficient [6].

The XSF is a quadcopter with overall dimensions of 68×68 cm that can carry a load of 2 kg with a wingspan of 50 cm and has a maximum flight time of 30 minutes. it is designed in a cross shape made of carbon fiber. Each end of the cross has a rotor comprising an electric motor and a cowl-mounted quad-blade propeller [6].

Rotors 1 and 2 rotate in the opposite direction to rotors 3 and 4 in order to eliminate the anti-torques between the propellers and allow yaw stability (figure 2). At equilibrium, the drone is either in a stationary position or in vertical motion (ascending or descending). The XSF compared to the existing quadrotor is the pivoting of the supports that carry rotors 1 and 3 around the pitch axis. This pivoting ensures either horizontal straight-line movement or rotational movement around the yaw axis, or a combination of these two movements (the turn) [6].



Fig. 1. Quadrotor [2]

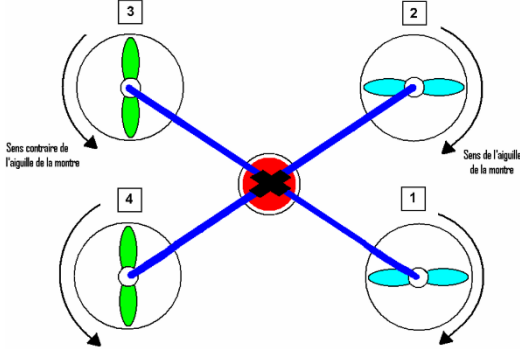


Fig. 2. Rotation direction of the XSF rotors

II. QUADROTOR DYNAMIC MODELLING

The first step, dedicated to the mathematical modeling of the bidirectional quadrotor, takes into account the various effects that manifest on it. To do this, we will begin with a dynamic study of the quadrotor using the fundamental law of classical dynamics (Newton's laws of translation and rotation) after identifying the various forces acting on the platform, namely aerodynamic drag, gyroscopic effect, gravitational force, drag force, and the lift force of each rotor, the determination of kinetic and potential energies. For this, we will begin by presenting the different trihedra used in flight mechanics, as well as the transformation matrices between them [4] [5].

A. Quadrotor frame R_m

It is also convenient to define an equivalent trihedron (same orientation as that of R_0) to the previous trihedron and whose origin is linked to the quadrotor. It is denoted as (O_1, X_1, Y_1, Z_1) . The point O_1 is a fixed point of the quadrotor and coincides with its center of gravity G . This triad is called the terrestrial nominal triad. The vector $O_0 O_1 = (x, y, z)^T$ defines the position of the center of gravity of the quadrotor.

It is a trihedron associated with the presumed undeformable quadrotor. It is denoted (O_1, X_2, Y_2, Z_2) . The axes X_2, Y_2, Z_2 are the principal axes of inertia of the quadrotor.

The orientations of R_2 relative to R_1 (or R_0) are characterized by the roll-pitch-yaw (RPY) angles, also known as aeronautical angles, Cardan angles, or Tait-Bryan angles (Figure 3). These angles are defined as follows:

- Roll (θ): rotation around X_0
- Pitch (ϕ): rotation around Y_0
- Yaw (ψ): rotation around Z_0

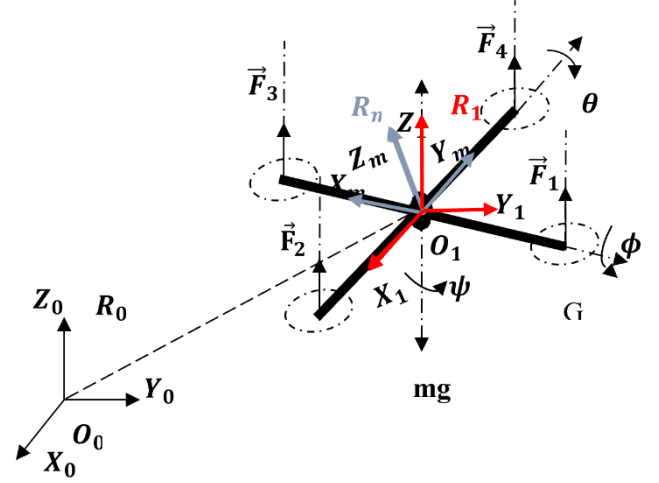


Fig. 3. The inertia, body and vehicle frames of reference.

The relationship between the XSF frame denoted R_m and the inertial frame denoted R_0 can be given by an orthogonal rotation matrix of dimension 3×3 . The orthogonality constraint reduces the number of parameters needed to present this matrix to three parameters [4].

$$R_0 \rightarrow \mathcal{R}_{(x_1, y_1, z_0)} \rightarrow \mathcal{R}_{(x_m, y_1, z_1)} \rightarrow \mathcal{R}_{(x_m, y_m, z_m)} \quad (1)$$

Thus, the transformation that allows moving from the local frame R_m to the global frame R_0 is given by:

$$R(\theta, \phi, \psi) = R_{z_0\psi} \cdot R_{y_1\phi} \cdot R_{x_m\theta} \quad (2)$$

Such as:

$$R(\theta, \phi, \psi) = \begin{bmatrix} C_\psi C_\theta & C_\theta S_\psi & -S_\theta \\ S_\phi C_\psi S_\theta - S_\psi C_\phi & S_\phi S_\psi S_\theta + C_\psi C_\phi & C_\theta S_\phi \\ S_\theta C_\psi C_\phi + S_\psi S_\phi & C_\phi S_\theta S_\psi - C_\psi S_\phi & C_\theta C_\phi \end{bmatrix} \quad (3)$$

Where: C is cosinus, S is sinus, R: is an orthogonal rotation matrix, with a unit determinant, whose inverse is its transpose.

Thanks to this rotation matrix R , we can establish a relationship between the speed of the mini-helicopter expressed in its own moving frame and its speed expressed in the inertial frame [5].

B. Summary of the forces and moments applied to the quadrotor

To perform these calculations, we resort to the classical definitions of moments and products of inertia. We will limit ourselves to recalling the moments I_{xx} and I_{xy} given by:

$$\begin{cases} I_{xx} = \int_0^d r^2 dm = \int_0^d r^2 \rho dv = \int_0^d r^2 \frac{m_p}{d} dr \\ I_{xy} = \int_0^d xy dm = \int_0^d xy \rho dv = \int_0^d xy \frac{m_p}{d} dr \end{cases} \quad (4)$$

m_p is the mass of the blade

r is the radial distance from the center of rotation to the differential blade element dr .

1) The moments of inertia of the two bars

$$I_B = I_{B_1} + I_{B_2} = \begin{pmatrix} \frac{m_B}{12}(2l)^2 & 0 & 0 \\ 0 & \frac{m_B}{12}(2l)^2 & 0 \\ 0 & 0 & \frac{m_B}{6}(2l)^2 \end{pmatrix} \quad (5)$$

2) Kinetic and potential energies of the bidirectional quadrotor XSF

The dynamic model is a set of equations linking the forces acting on the quadrotor and its kinematic parameters. This model is developed based on the following assumptions [4][5]:

- The inertia matrix is assumed to be diagonal (perfectly symmetric structure), this assumption allows for simple equations;
- The quadrotor moves at low speeds;
- The atmospheric conditions are standard pressure and temperature conditions (no wind);
- The quadrotor has a fairing around its rotors.

We present the generalized force vector $Q = (R\hat{Q}, \tau)^T$ associated with the XSF. The translational force vector applied to the rotors, expressed in the local frame (e_1, e_2, e_3) by the frame (E_1^g, E_2^g, E_3^g) is:

$$\hat{Q} = f_1 e'_1 + f_2 e'_2 + f_3 e'_3 + f_4 e'_4 \quad (6)$$

$$\text{Where: } e'_1 = \begin{pmatrix} 0 \\ \sin(\xi_1) \\ \cos(\xi_1) \end{pmatrix}; e'_2 = \begin{pmatrix} 0 \\ \sin(\xi_3) \\ \cos(\xi_3) \end{pmatrix}; e'_3 = e'_4 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

The sum of the moments of the forces applied respectively on the four rotors of the XSF is denoted $\tau = (\tau_\theta, \tau_\phi, \tau_\psi)^T$ given by:

$$\begin{cases} \tau = M_1 + M_2 + M_3 + M_4 \\ M_1 = \begin{pmatrix} l \\ 0 \\ 0 \end{pmatrix} \wedge \begin{pmatrix} 0 \\ K_M \omega_1^2 + K_T \omega_1^2 \sin(\xi_1) \\ K_T \omega_1^2 \cos(\xi_1) \end{pmatrix} \\ M_2 = \begin{pmatrix} 0 \\ l \\ 0 \end{pmatrix} \wedge \begin{pmatrix} -K_M \omega_2^2 \\ 0 \\ K_T \omega_2^2 \end{pmatrix} \\ M_3 = \begin{pmatrix} -l \\ 0 \\ 0 \end{pmatrix} \wedge \begin{pmatrix} 0 \\ K_M \omega_3^2 + K_T \omega_3^2 \sin(\xi_3) \\ K_T \omega_3^2 \cos(\xi_3) \end{pmatrix} \\ M_4 = \begin{pmatrix} 0 \\ -l \\ 0 \end{pmatrix} \wedge \begin{pmatrix} -K_M \omega_4^2 \\ 0 \\ K_T \omega_4^2 \end{pmatrix} \end{cases} \quad (7)$$

From where:

$$\begin{cases} \tau_\theta = l(f_2 - f_4) \\ \tau_\phi = l(f_3 \cos(\xi_3) - f_1 \cos(\xi_1)) \\ \tau_\psi = l[f_1 \sin(\xi_1) - f_3 \sin(\xi_3) + \frac{K_M}{K_T}(f_2 + f_1 - f_4 - f_3)] \end{cases} \quad (8)$$

We determine the dynamic model of the XSF, let L be the Lagrangian function of the XSF given:

$$L = \frac{1}{2} m \dot{\eta}_1^T \dot{\eta}_1 + \frac{1}{2} v_2^T I_B v_2 + \frac{1}{2} (v_2 + \omega_1 \cos(\xi_1) e_3 + \omega_1 \sin(\xi_1) e_2)^T I_{r1} (v_2 + \omega_1 \cos(\xi_1) e_3 + \omega_1 \sin(\xi_1) e_2) + \frac{1}{2} (v_2 + \omega_2 e_3)^T I_{r2} (v_2 + \omega_2 e_3) +$$

$$\frac{1}{2} (v_2 + \omega_3 \cos(\xi_3) e_3 + \omega_3 \sin(\xi_3) e_2)^T I_{r3} (v_2 + \omega_3 \cos(\xi_3) e_3 + \omega_3 \sin(\xi_3) e_2) + \frac{1}{2} (v_2 + \omega_4 e_3)^T I_{r4} (v_2 + \omega_4 e_3) - g m \eta_1 e_3 - 4k(a_{0i} - a_{1s_i} \cos(\gamma) - b_{1s_i} \sin(\gamma)) \quad (9)$$

The dynamics of translation are as follows:

$$\begin{cases} m\ddot{x} = S_\psi C_\theta u_5 - S_\theta u_1 \\ m\ddot{y} = (S_\theta S_\psi C_\phi + C_\psi C_\phi) u_5 + C_\theta S_\phi u_1 \\ m\ddot{z} = (S_\theta S_\psi S_\phi - C_\psi S_\phi) u_5 + C_\phi C_\theta u_1 - mg \end{cases} \quad (10)$$

Where: C is cosinus, S is sinus

By applying the Euler-Lagrange formalism, we obtain:

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}_i} \right] - \frac{\partial L}{\partial q_i} = \begin{pmatrix} I_M \ddot{\eta}_1 + m g e_3 \\ \mathbb{J}^{-1T}(\eta_2) (I_B + I_r) \mathbb{J}^{-1}(\eta_2) \ddot{\eta}_2 + C(\eta_2, \dot{\eta}_2) \dot{\eta}_2 \end{pmatrix} \quad (11)$$

$C(\eta_2, \dot{\eta}_2) \dot{\eta}_2$: is a coefficient containing the Coriolis term and the gyroscopic and centrifugal terms. This coefficient is expressed as:

$$C(\eta_2, \dot{\eta}_2) \dot{\eta}_2 = \frac{d}{dt} [\mathbb{J}^{-1T}(\eta_2) K] + \frac{d}{dt} [\mathbb{J}^{-1T}(\eta_2) (I_B + I_r) \mathbb{J}^{-1}(\eta_2)] \dot{\eta}_2 - \frac{1}{2} \frac{\partial}{\partial \eta_2} [\dot{\eta}_2^T \mathbb{J}^{-1T}(\eta_2) (I_B + I_r) \mathbb{J}^{-1}(\eta_2) \dot{\eta}_2] - \frac{\partial}{\partial \eta_2} [\dot{\eta}_2^T \mathbb{J}^{-1T}(\eta_2) K] \quad (12)$$

Where K is defined by:

$$K = (I_{r1}(\omega_1 \cos(\xi_1) e_3 + \omega_1 \sin(\xi_1) e_2) + (I_{r3}(\omega_3 \cos(\xi_3) e_3 + \omega_3 \sin(\xi_3) e_2) + (I_{r4} \omega_4 + I_{r2} \omega_2) e_3$$

By performing an input/output linearization using the following decoupled control laws:

$$\tilde{\tau} = [\mathbb{J}^{-1T}(\eta_2) (I_B + I_r) \mathbb{J}^{-1}(\eta_2)]^{-1} [\tau - C(\eta_2, \dot{\eta}_2) \dot{\eta}_2] \quad (13)$$

The rotation dynamics are:

$$\begin{cases} \tau_\theta = \frac{1}{I_{xx} C_\phi} (\tau_\theta + I_{xx} S_\phi \dot{\phi} \dot{\theta}) \\ \tau_\phi = \frac{1}{I_{yy} C_\phi C_\theta} (\tau_\phi + I_{yy} S_\theta C_\phi \dot{\phi} \dot{\theta} + I_{yy} S_\phi C_\theta \dot{\phi}^2) \\ \tau_\psi = \frac{1}{I_{zz}} \tau_\psi \end{cases} \quad (14)$$

Finally, the dynamics of the XSF can be expressed in the following simple form [4] [5]:

$$\begin{cases} m\ddot{x} = \cos(\theta) \sin(\psi) u_5 - \sin(\theta) u_1 \\ m\ddot{y} = \begin{bmatrix} \sin(\theta) \sin(\phi) \sin(\psi) \\ + \cos(\theta) \sin(\phi) \sin(\psi) \end{bmatrix} u_5 \\ m\ddot{z} = \begin{bmatrix} \sin(\theta) \cos(\phi) \sin(\psi) \\ - \cos(\theta) \cos(\phi) \sin(\psi) \end{bmatrix} u_5 \\ \ddot{\theta} = \tilde{\tau}_\theta \\ \ddot{\phi} = \tilde{\tau}_\phi \\ \ddot{\psi} = \tilde{\tau}_\psi \end{cases} \quad (15)$$

With:

$$\begin{cases} u_5 = f_1 \sin(\xi_1) + f_3 \sin(\xi_3) \\ u_1 = f_1 \cos(\xi_1) + f_3 \cos(\xi_3) + f_2 + f_4 \end{cases}$$

Where: u_5 controls the horizontal displacement generated by the rotations of rotors 1 and 3 around their lift axes; u_1 controls the vertical displacement of the XSF; $\tilde{\tau}_\psi$ controls the rotation of the XSF around the Z axis; $\tilde{\tau}_\theta$ controls the pitch angle; $\tilde{\tau}_\phi$ controls the roll angle.

We note that this dynamic model is highly nonlinear and underactuated (the number of controls is less than the number of state variables).

The vector $(u_1, u_5, \tilde{\tau}_\psi, \tilde{\tau}_\theta, \tilde{\tau}_\phi)^T$ presents the control vector of the model

III. CONTROL STRATEGY DESIGN

The backstepping technique combines the notion of Lyapunov control function with a recursive design procedure. This allows overcoming the dimensional obstacle and exploiting the design flexibility in the scalar case to solve control problems for very high-order systems. Not necessarily relying on linearization, backstepping allows, when present, to retain useful nonlinearities which often help maintain finite values of the state vector. Backstepping is based on the second Lyapunov method, which combines the choice of the Lyapunov control function with that of the control laws. This allows it, in addition to the task for which the controller is designed (tracking and/or regulation), to ensure, at all times, the global stability of the compensated system [7].

The idea is to calculate a control law to ensure that the derivative of a certain function (Lyapunov) is positive definite and that its derivative is always negative. The method consists of fragmenting the system into a set of nested subsystems in decreasing order. The calculation of the Lyapunov function is then carried out recursively, starting from the innermost loop. At each step, the order of the system is increased and the part that was not stabilized in the previous step is addressed. At the final stage, the control law is determined. This ensures, at all times, the global stability of the compensated system while working in pursuit and regulation [7].

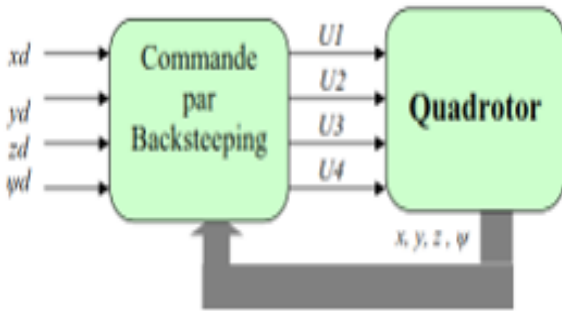


Fig. 4. Synoptic diagram of backstepping control

Unlike most other methods, backstepping has no constraints regarding the type of non-linearity. However, the system must be in the form of what is called pure parametric. The equations of such a system are given by:

$$\begin{cases} \dot{x}_1 = \varphi_1^T(x_1)\vartheta + \psi_1(x_1)x_2 \\ \dot{x}_2 = \varphi_2^T(x_1, x_2)\vartheta + \psi_2(x_1, x_2)x_3 \\ \vdots \\ \dot{x}_{n-1} = \varphi_{n-1}^T(x_1, x_2, \dots, x_{n-1})\vartheta + \psi_{n-1}(x_1, x_2, \dots, x_{n-1})x_n \\ \dot{x}_n = \varphi_n^T(x_1, x_2, \dots, x_n)\vartheta + \psi_n(x_1, x_2, \dots, x_n)u \\ y = x_1 \end{cases} \quad (16)$$

IV. APPLICATION OF BACKSTEPPING/PID CONTROL TO A XSF MODEL

A. Movement control along the y and z axes

$$u_{5x} = \frac{c_\phi}{c_\psi} m \ddot{y} - \frac{s_\phi}{c_\psi} m(\ddot{z} + g) \quad (17)$$

$$u_1 = \frac{c_\psi s_\phi c_\phi - c_\phi s_\psi s_\theta}{c_\psi c_\theta} m \ddot{y} + \frac{s_\phi s_\theta s_\psi + c_\psi c_\phi}{c_\psi c_\theta} m(\ddot{z} + g) \quad (18)$$

$$\ddot{y} = k_p^y(y_d - y) + k_d^y(\dot{y}_d - \dot{y}) + \ddot{y}_d \quad (19)$$

$$\ddot{z} = k_p^z(z_d - z) + k_d^z(\dot{z}_d - \dot{z}) + \ddot{z}_d \quad (20)$$

The closed-loop dynamics of (y, z) are decoupled and exponentially asymptotically stable with the appropriate choice of gains k_i^y and k_i^z with $i = (p, o, d)$.

B. Pitch and roll control

$$\theta_d = \arcsin\left(\frac{\sin(\psi) u_{5x} - m \ddot{x}}{u_1}\right) \quad (21)$$

$$\phi_d = \arcsin\left(\frac{1}{u_1}(m \ddot{y} - \cos(\psi) u_5)\right) \quad (22)$$

Pitch and roll control The backstepping/PID regulation technique of the subsystem is designed for rotation, in which the control inputs U_2 and U_3 control the quadrotor during pivoting [4]

Let the roll tracking error be defined as:

$$e = \theta - \theta_d \quad (23)$$

The first error considered in the design of backstepping is:

$$z_1 = k_1 e + k_2 \int e dt \quad (24)$$

Where k_1 and k_2 are positive tuning parameters, and $\int e dt$ is the integral of the roll error. Lyapunov's theory is used during the use of the Lyapunov function z_1 , which is defined as positive, and its derivative is defined as negative at the same time [7]:

$$V_1 = \frac{1}{2} z_1^2 \quad (25)$$

$$\dot{V}_1 = z_1 \dot{z}_1 = z_1(k_1 \dot{\theta} - k_1 \dot{\theta}_d + k_2 e) \quad (26)$$

There is no control input in equation (26). Setting $\dot{\theta}$ as a virtual control, the desired virtual control $(\dot{\theta})_d$ is defined as follows:

$$(\dot{\theta})_d = \dot{\theta} + \frac{k_2 e}{k_1} - \frac{c_1 \dot{z}_1}{k_1} \quad (27)$$

Where c_1 is a positive constant to increase the convergence speed of the roll tracking loop.

Now, the virtual control θ is the roll rate of a quadrotor with its own error:

$$z_2 = \dot{\theta} - (\dot{\theta})_d = \frac{1}{k_1}(\dot{z}_1 + c_1 z_1) \quad (28)$$

The augmented Lyapunov function for the second step is given by:

$$V_2 = \frac{1}{2} z_1^2 + \frac{1}{2} z_2^2 \quad (29)$$

$$\dot{V}_2 = z_1 \dot{z}_1 + z_2 \dot{z}_2 \quad (30)$$

$$\begin{aligned} \dot{V}_2 = z_2 \left[e \left(k_1^2 + \frac{c_1 k_2}{k_1} \right) + \int edt (k_1 k_2) + \dot{e} \left(\frac{k_2}{k_1} + c_1 \right) + \right. \\ \left. \phi \dot{\psi} \left(\frac{I_{yy} - I_{zz}}{I_{xx}} \right) + \frac{I_{u_2}}{I_{xx}} - \dot{\theta}_d \right] - z_1 [c_1 k_1 e + c_1 k_2 \int e dt] \end{aligned} \quad (31)$$

The desired dynamics are:

$$\dot{V}_2 = -\dot{e}(c_2) - e \left(\frac{c_2 k_2}{k_1} + c_1 c_2 \right) - \int edt \left(\frac{c_1 c_2 k_2}{k_1} \right) \quad (32)$$

The desired dynamics ensure a defined-negative position tracking error, its integration, and a speed tracking error.

$$\begin{aligned} u_2 = \ddot{\theta} = \frac{I_x}{l} \left[-e \left(\frac{c_2 k_2}{k_1} + c_1 c_2 + k_1^2 + \frac{c_1 k_2}{k_1} \right) - \right. \\ \left. \int edt \left(\frac{c_1 c_2 k_2}{k_1} + k_1 k_2 \right) - \dot{e} \left(c_2 + \frac{k_2}{k_1} + c_1 \right) + \right. \\ \left. \dot{\theta}_d - \phi \dot{\psi} \left(\frac{I_{yy} - I_{zz}}{I_{xx}} \right) \right] \end{aligned} \quad (33)$$

As a rule, equation (33) is a PID, where the gains of each mode are given by:

$$\begin{aligned} P &= \frac{c_2 k_2}{k_1} + c_1 c_2 + k_1^2 + \frac{c_1 k_2}{k_1} \\ I &= \frac{c_1 c_2 k_2}{k_1} + k_1 k_2 \\ D &= c_2 + \frac{k_2}{k_1} + c_1 \end{aligned}$$

Let's choose high values of c_1 and c_2 , which makes the Lyapunov function more negative, resulting in faster and more precise regulation dynamics.

As for the roll subsystem, the backstepping-based PID control is designed for the pitch subsystem to obtain u_3 as follows:

$$\begin{aligned} u_3 = \ddot{\phi} = \frac{I_y}{l} \left[-e \left(\frac{c_4 k_4}{k_3} + c_3 c_4 + k_3^2 + \frac{c_3 k_4}{k_3} \right) - \right. \\ \left. \int edt \left(\frac{c_3 c_4 k_4}{k_3} + k_3 k_4 \right) - \dot{e} \left(c_4 + \frac{k_4}{k_3} + \right. \right. \\ \left. \left. c_3 \right) + \ddot{\phi}_d - \dot{\theta} \dot{\psi} \left(\frac{I_{zz} - I_{xx}}{I_{yy}} \right) \right] \end{aligned} \quad (34)$$

C. The new control of movement along the z and x axes

$$u_1 = (S_\psi C_\phi - S_\phi C_\psi S_\theta)^{-1} (-m v_x (C_\phi S_\psi S_\theta - C_\psi S_\phi) + m(v_z + g) S_\psi C_\theta) \quad (35)$$

$$u_{5y} = (S_\psi C_\phi - S_\phi C_\psi S_\theta)^{-1} (m v_x C_\phi C_\theta + m(v_z + g) S_\theta) \quad (36)$$

$$v_x = \ddot{x} = k_p^x (x_d - x) + k_d^x (\dot{x}_d - \dot{x}) + \ddot{x}_d \quad (37)$$

$$v_z = \ddot{z} = k_p^z (z_d - z) + k_d^z (\dot{z}_d - \dot{z}) + \ddot{z}_d \quad (38)$$

Note:

- ✓ **u_{5x} is the control of the two motors 1 and 3 in the (z, y) plane for $\psi = \left] -\frac{\pi}{2}, \frac{\pi}{2} \right]$.**
- ✓ **u_{5y} is the control of the two motors 1 and 3 in the (z, x) plane for $\psi = \pm \frac{\pi}{2}$.**

V. SIMULATION AND RESULTS

The geometric and inertial parameters of the quadcopter are shown in the following table:

TABLE I. PARAMETERS USED IN THE SIMULATION.

Ixx [kg.m ²]	0.0224931
Iyy [kg.m ²]	0.0222611
Izz [kg.m ²]	0.0325130
l [m]	0.23
m [kg]	2
K _M [N.m.s ²]	9E-6
K _T [N.s ²]	1E-5
g [m.s ⁻²]	9.81
b [kg.m.rd ⁻²]	6e-6
d [kg.m ² .rd ⁻²]	1e-4

The simulations that follow are all carried out using the parameters of Table I, they correspond to simple flight plans.

We work with a straight-line trajectory along the z, x, and y axes and perform arc-shaped or semi-circular trajectories. Point-by-point stabilization is used with motion planning. The drone must fly in the z direction (vertical flight), followed by a movement along the x axis, and then a movement along the y axis.

The reference trajectory for vertical flight is parameterized by the following equation:

$$z_d(t) = h_d \frac{t^5}{t^5 + (T_f^1 - t)^5} \quad (39)$$

With h_d the desired altitude of 10m and T_f^1 is the final time.

Flight Control of a Quadrotor by Backstepping/PID

Given that the final value of a flight phase is the initial value of the following movement, we can write the reference trajectory along x as:

$$x_d(t) = h_d \frac{(t-T_f^1)^5}{(t-T_f^1)^5 + (T_f^1 - (t-T_f^1))^5} \quad (40)$$

We can write the movement along y using the following formula:

$$y_d(t) = h_d \frac{(t-T_f^2)^5}{(t-T_f^2)^5 + (T_f^2 - (t-T_f^2))^5} \quad (41)$$

Where: T_f^1 is the final time of movement along the z-axis; T_f^2 is the final time of movement along the x-axis; T_f^3 is the

final time of movement along the y-axis; h_d : Desired altitude of 10 m.

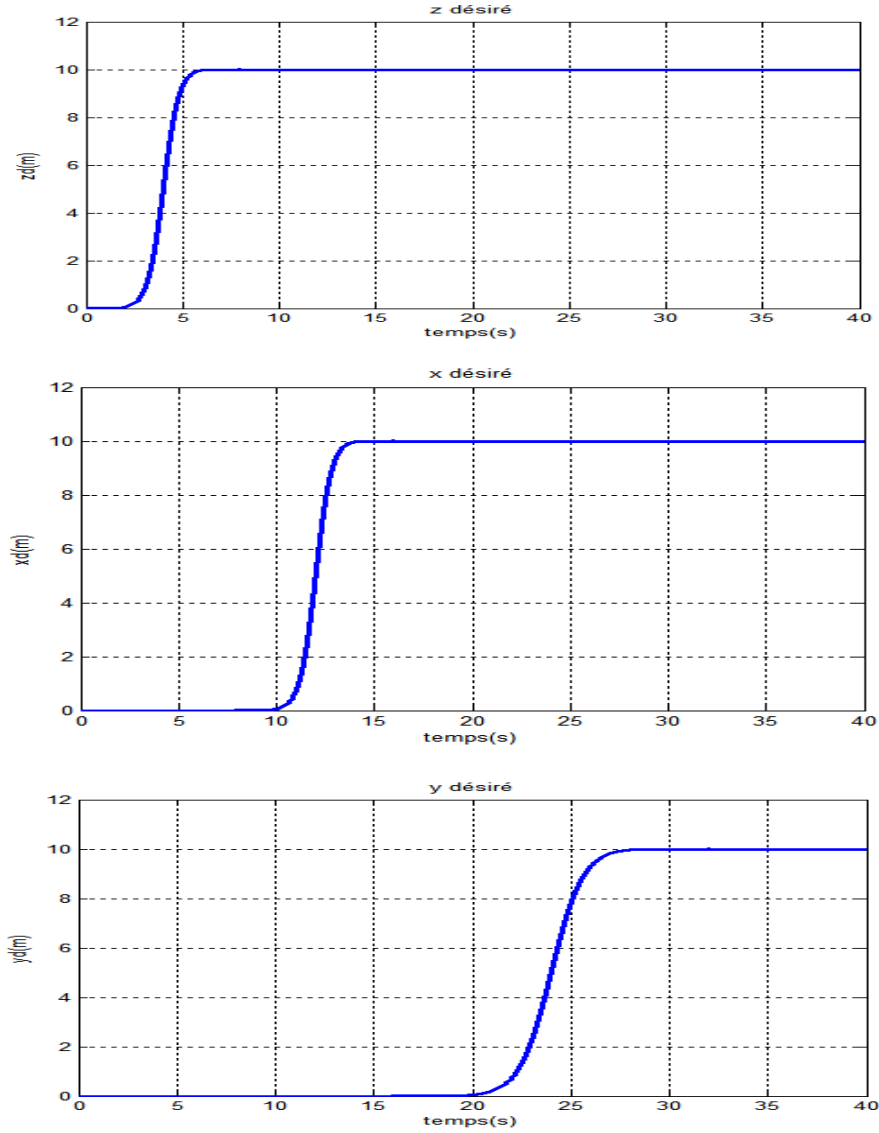
These trajectories are subject to performance constraints which are:

$$z_d(0) = x_d(T_f^1) = y_d(T_f^2) = 0 \quad (42)$$

$$z_d(T_f^1) = x_d(T_f^2) = y_d(T_f^3) = h_d \quad (43)$$

At $t=0$: the initial configuration is: $(x, y, z, \Phi, \theta, \psi) = (0, 0, 0, 0, 0, 0)$.

Figure (5) shows Desired trajectories on the axis x, y, z and ψ . Figure (6) presents Desired trajectory of the quadrotor. Figure (7) illustrates the variation of the Euler angles Φ and θ .



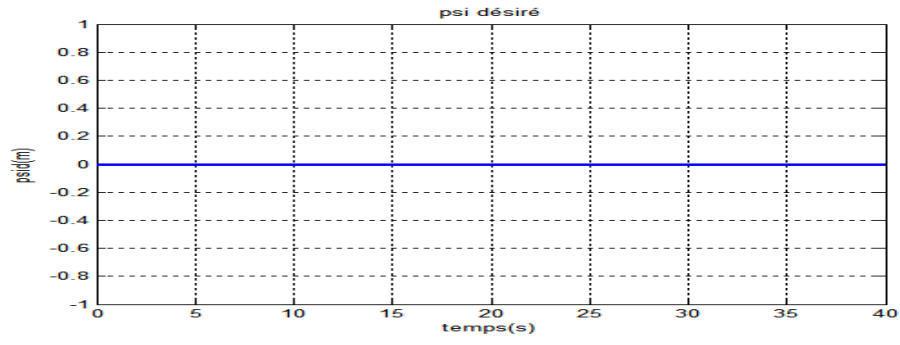


Fig. 5. Desired trajectories x, y, z, ψ

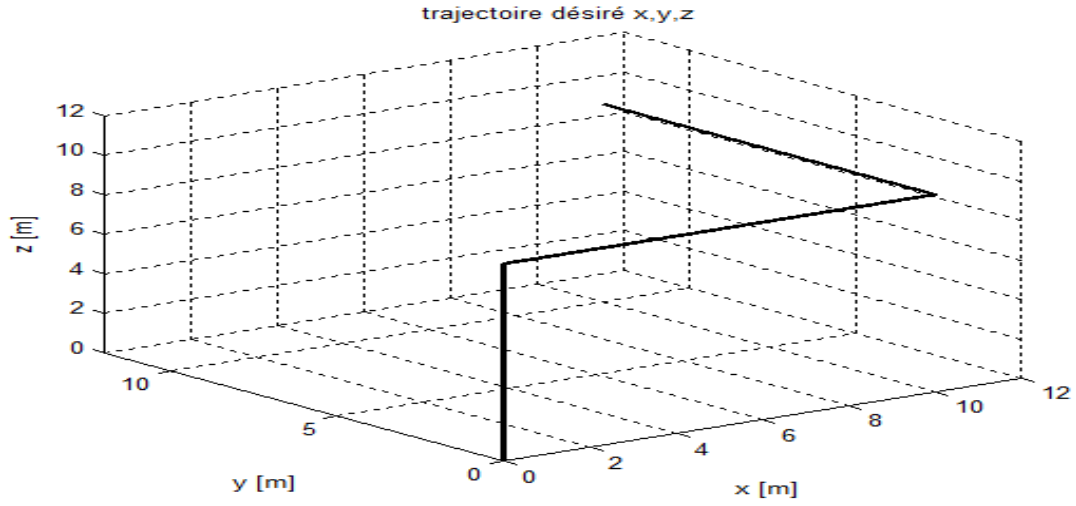
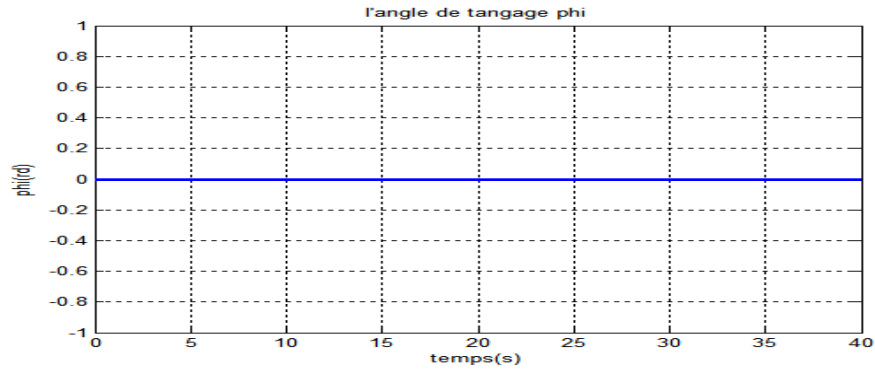
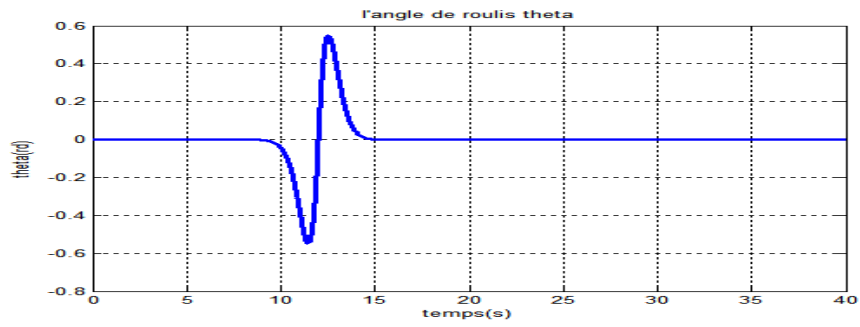


Fig. 6. Desired trajectory of the quadrotor



-a-



-b-

Fig. 7. Variation of the Euler angles: a) angle Φ , b) angle θ

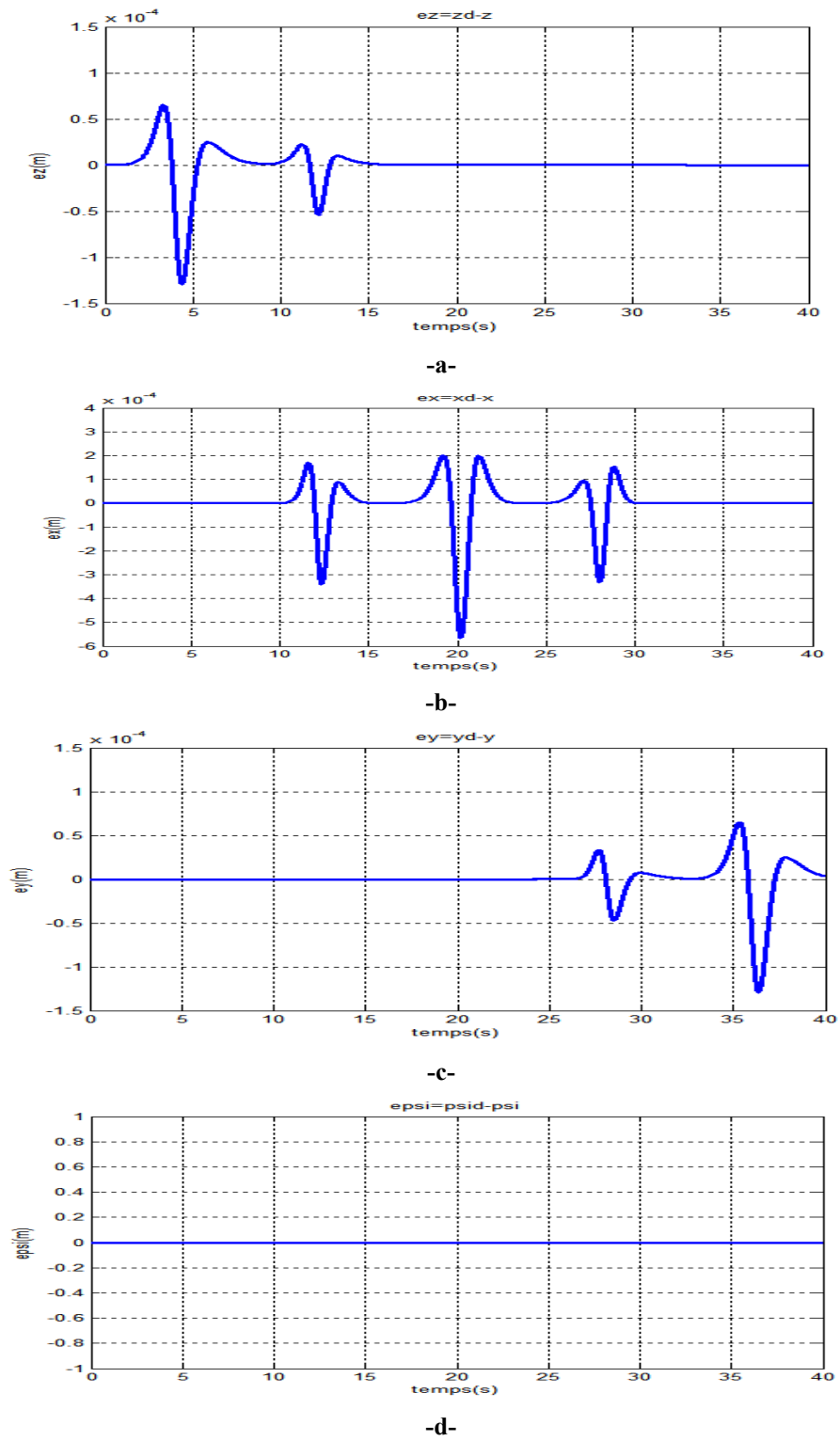


Fig. 8. Errors in: a) tracking x, b) tracking y, c) tracking z, d) tracking ψ

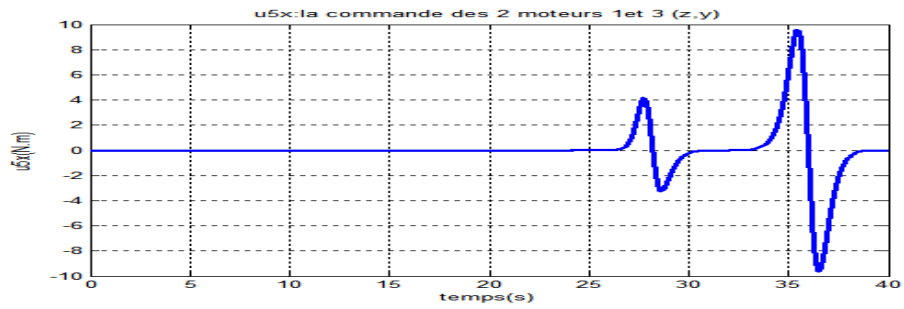


-a-

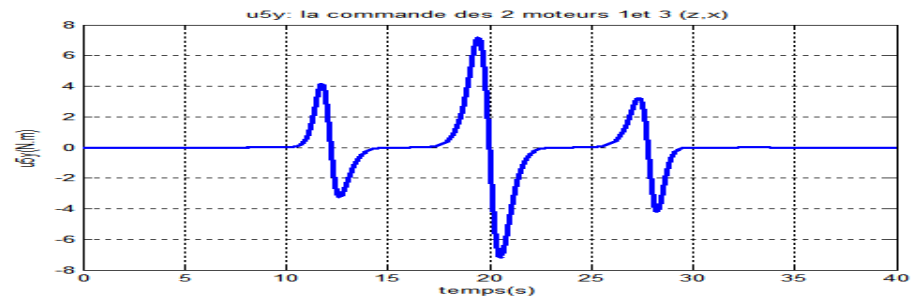


-b-

Fig. 9. Motor control voltages : a) Motor 1, b) Motor 2.



-a-



-b-

Fig. 10. Behavior of is the control of the two motors 1 and 3: a) u_{5x} in the (z, y) plane; b) u_{5y} in the (z, x) plane.

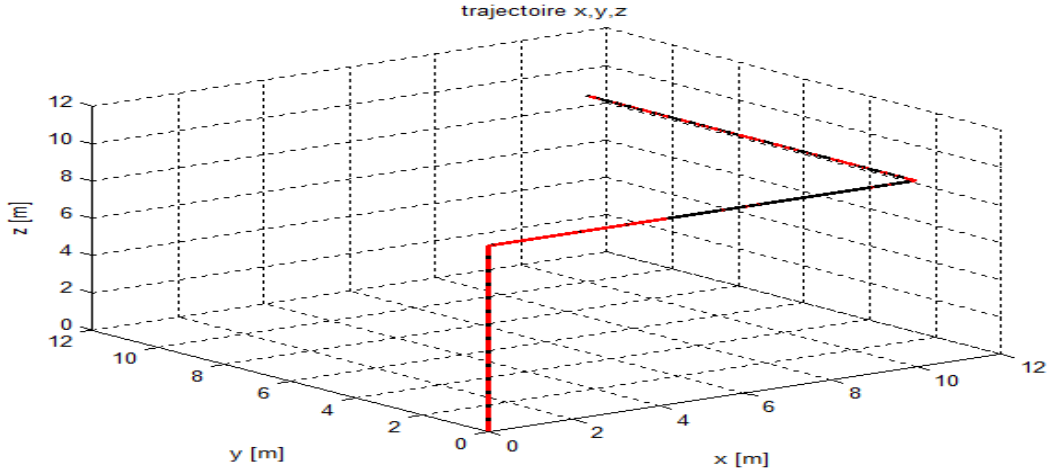
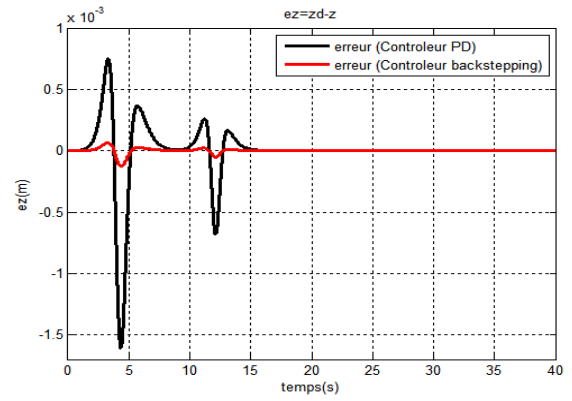


Fig. 11. Desired trajectory and quadrotor trajectory

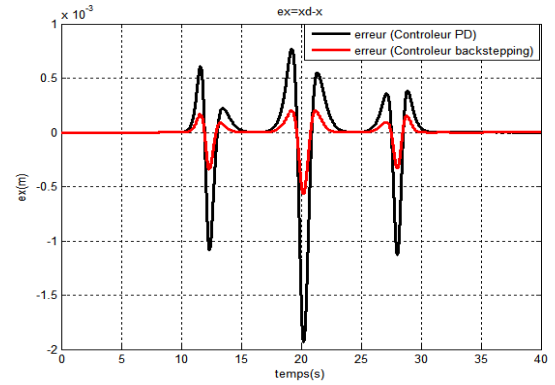
Figure (10) presents the behavior of is the control of the two motors 1 and 3 in the (z, y) plane; band in the (z, x) plane.

Observing the results obtained, we can draw the following conclusions:

- ✓ The tracking errors shown in Figure (8) reflect the performance of the chosen control law.
- ✓ Thanks to the appropriate choice of the PD-BACKSTEPPING controller's K_p , K_i , and K_d values, and the tracking errors that cancel out in a short time, the controller uses minimal energy to perform the desired task (Fig.9).
- ✓ The desired trajectory tracking is perfect (Fig. 11). This is explained by the fact that the control generated by the backstepping/PID controller ensures asymptotic convergence of the state vector.
- ✓ According to the implementation results, this method of extracting trajectory primitives yielded better results in terms of speed and accuracy, which is necessary for real-time applications, compared to conventional control (Fig.12).
- ✓ Analysis of the simulation curves suggests good performance; tracking and stability conditions are met. We observe that tracking errors are significantly improved, and Euler angles tend towards zero, implying that the device stabilizes at its desired position with good performance compared to previous controllers.



-a-



-b-

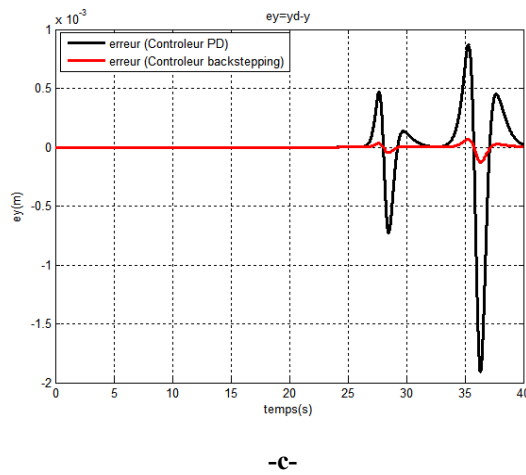


Fig. 12. Comparison of tracking errors for the two applied commands: a) for x, b) for y, c) for z

VI. CONCLUSION

The backstepping control technique should not be viewed as a rigid design process, but rather as a design philosophy that can be bent and twisted to meet the specific needs of the system being controlled. In the particular example of the drone helicopter, we can exploit the flexibility of backstepping with regard to the choice of virtual controls, initial stabilizing functions, and Lyapunov functions. We conducted a stability study using the total Lyapunov function of the chosen quadcopter model. We showed that the correct choice of control parameters helps to mitigate the error (between the desired and actual trajectory) affecting the helicopter's tracking. Furthermore, the results obtained demonstrated the effectiveness of using this type of control on the quadcopter, as evidenced by the good control performance and the sustained tracking. Also, the simulation results show that the derivative of the total Lyapunov function is always negative. From a practical standpoint, the PID control is the easiest to implement, while the backstepping control is the most difficult because it requires significant computation to arrive at the final version of the control.

We can conclude that both proposed control algorithms produce satisfactory dynamic performance. The backstepping and backstepping-PID methods demonstrate better tracking and stabilization under the prescribed performance conditions. We have shown the efficiency of each control based on the minimum error.

Generally, we notice that during our study, the XSF model is best suited for fairly complicated trajectories and external disturbances. Its two new degrees of freedom or joints give it

good flexibility in movement and also allow it to maintain a horizontal position despite the trajectory constraints (arc or semicircle movement).

The results obtained from the simulation demonstrate the effectiveness of the various control strategies developed, and thus were very satisfactory.

REFERENCES

- [1] E. Colle and G. Damm, "UAV control system", [Système de commande pour drone], Laboratoire Systèmes Complexes (LSC), Université d'Évry Val d'Essonne, Évry, France, Tech. Rep. FRE 2494, 2004.
- [2] P. Castillo, A. Dzul and R. Lozano, "Real-time stabilization and tracking of a four-rotor mini rotorcraft," in *IEEE Transactions on Control Systems Technology*, vol. 12, no. 4, pp. 510-516, July 2004, <https://doi.org/10.1109/TCST.2004.825052>
- [3] A. Mokhtari, "Robust control of an airborne quadrotor", [Commande robuste d'un engin Quadrotor aéroporté], Ph.D. dissertation, Dept. Electr. Eng., Université des Sciences et de la Technologie d'Oran (USTO), Oran, Algeria, 2005.
- [4] K. M. Zemalache, "Commande d'un système sous actionné Application à un drone à Quatre Hélices" (Control of an underactuated system: Application to a four-rotor drone), Ph.D. dissertation, University of Evry Val d'Essonne, France, 20 décembre 2006.
- [5] R. Slim, "Contribution à la planification et à la commande d'un drone hélicoptère a quatre rotors", (Contribution to the planning and control of a four-rotor helicopter drone), Ph.D. dissertation, University of Evry Val d'Essonne, France, November 2006.
- [6] Lotfi Beji and Azgal Abichou, « Streamlined Rotors Mini Rotorcraft Trajectory Generation and Tracking », *International Journal of Control, Automation, and Systems* vol.3, no. 1, pp. 87-99, march 2005.
- [7] L.N. Paquin, « Application du backstepping à une colonne de flottation », ("Applying backstepping to a flotation column"), Ph.D. dissertation, University of Laval, France, July 2000.
- [8] Abiyev, R.H., Akkaya, N., Aytac, E., Abizada, S. (2022). Fuzzy Gain Scheduling Controller for Quadrotor. In: Aliev, R.A., Kacprzyk, J., Pedrycz, W., Jamshidi, M., Babanli, M., Sadikoglu, F.M. (eds) 11th International Conference on Theory and Application of Soft Computing, Computing with Words and Perceptions and Artificial Intelligence - ICSCCW-2021. ICSCCW 2021. Lecture Notes in Networks and Systems, vol 362. Springer, Cham. https://doi.org/10.1007/978-3-030-92127-9_52
- [9] J. Kim, S. A. Gadsden and S. A. Wilkerson, "A Comprehensive Survey of Control Strategies for Autonomous Quadrotors," *Canadian Journal of Electrical and Computer Engineering*, vol. 43, no. 1, pp. 3-16, winter 2020, <https://doi.org/10.1109/CJECE.2019.2920938>
- [10] R. S. Wijaya, S. Prayoga, R. A. Fatekha, and M. T. Mubarak, "A Real-Time Hand Gesture Control of a Quadcopter Swarm Implemented in the Gazebo Simulation Environment", *Journal of Applied Informatics and Computing (JAIC)*, Vol. 9, no. 3, pp. 979-988, Jun. 2025. <https://doi.org/10.30871/jaic.v9i3.9578>
- [11] M. Aly, B. Ali, F. Y. Mekonnen, M. Elhesasy, M. Wang, M. M. Kamra, & T. N. Dief, "Real-Flight-Path Tracking Control Design for Quadrotor UAVs: A Precision-Guided Approach". *Automation*, Vol. 6, N° 4, Art. N°93, 2025. <https://doi.org/10.3390/automation6040093>