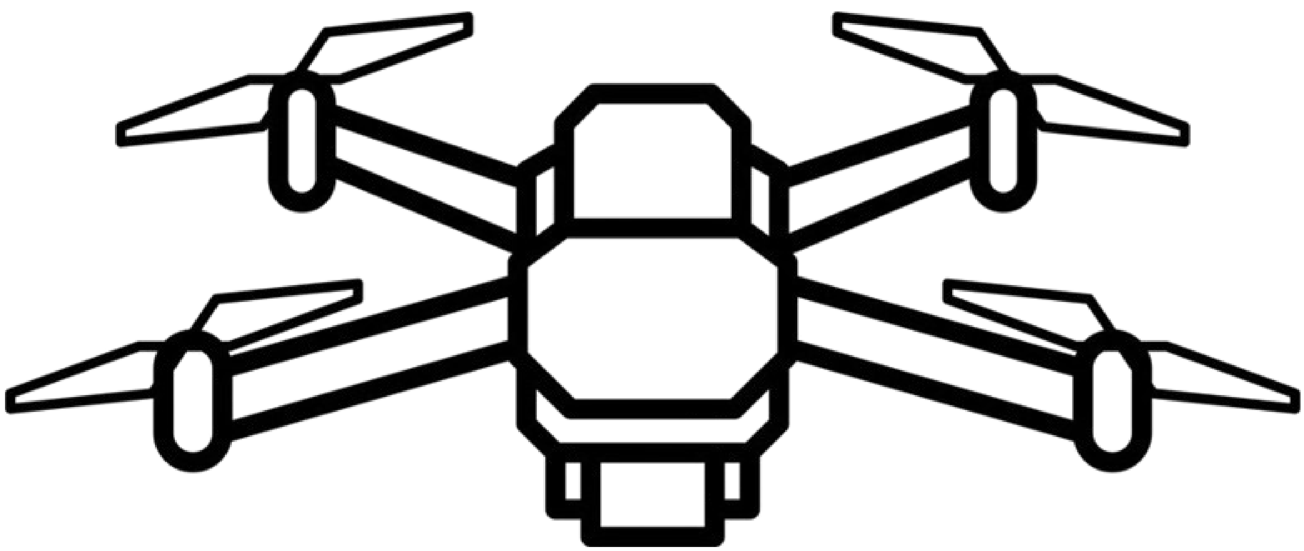
Université de Technologie
de Compiègne

Master Automation and Robotics in Intelligent Systems (ARS)

ARS5 Project Report

Quadcopter Control Using Backstepping and Newton-Euler Modeling



Prepared by :

Yassine Mathlouthi
Fedi Ben Ali
Adnane Sallili

Supervisor: Prof. Dr. Predro CASTILLO

2024-2025

Contents

1 Introduction and objectives

A quadcopter is essentially a drone that's powered by four rotor blades. Because it can hover and take off vertically and execute agile maneuvers , the quadcopter can be used for a variety of purposes including military missions, delivering, and providing surveillance... However, controlling a quadcopter is challenging due to its nonlinear and complex dynamics.

Figure 1: S109 RC Aircraft Quadcopter

So, the focal point of this project is to construct a quadcopter control system that is based on modeling approaches of Newton-Euler and a backstepping process controls the quadcopter. Newton-Euler approach provides an alternative understanding of the quadcopter dynamics which is crucial for the development of the quadcopter system.
This approach, known as Backstepping, is effective in addressing issues surrounding a quadcopter including nonlinearity. This is because the design combines robust control with good control models. Overall, this project endeavors to create a quadcopter that can hover steadily at a specific point and can follow given movement patterns.
The following sections outline the modeling process, the design of the backstepping controller, the t numerical simulation and the evaluation of the system's performance.

2 Mathematical Model

In this section, we will study the mathematical model of the rotorcraft based on Newton-Euler approach.
To specify the attitude of quadcopters in space, two frames have to be introduced which are inertial frame F_i and body frame F_b.
Let the position vector of the quadcopter be defined as $\xi = \begin{bmatrix} x & y & z \end{bmatrix}^T \in F_i$, and its orientation vector described by Euler angle $\eta = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T \in F_i$ in terms of roll, pitch and yaw angles; then, the linear velocity $\vec{V} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T \in F_b$ and angular velocity $\vec{\Omega} = \begin{bmatrix} p & q & r \end{bmatrix}^T \in F_b$ of the quadcopter have the following relationship with position and orientation vectors:

$$\begin{aligned} \dot{\vec{\xi}} &= \mathbf{R}\vec{V} \\ \dot{\vec{\eta}} &= \mathbf{W}_\eta^{-1}\vec{\Omega} \end{aligned} \tag{1}$$

Where $\mathbf{R}$ is the rotation matrix from the body frame to the inertial frame and $\mathbf{W}_\eta^{-1}$ is The transformation matrix for angular velocities from the body frame to the inertial frame, C, S and T represent cosine, sine and tangent functions respectively:

$$R = \begin{bmatrix} C_\theta C_\psi & C_\theta S_\psi & -S_\theta \\ C_\psi S_\theta S_\phi - S_\psi C_\phi & S_\psi S_\theta S_\phi + C_\psi C_\phi & C_\theta S_\psi \\ C_\psi S_\theta S_\phi + S_\psi C_\phi & S_\psi S_\theta S_\phi - C_\psi C_\phi & C_\theta C_\psi \end{bmatrix}$$

(2)

$$\mathbf{W}_\eta = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta C_\phi \end{bmatrix} \mathbf{W}_\eta^{-1} = \begin{bmatrix} 1 & S_\phi T_\theta & C_\phi T_\theta \\ 0 & C_\phi & -S_\phi \\ 0 & \frac{S_\phi}{C_\theta} & \frac{C_\phi}{C_\theta} \end{bmatrix}$$

(3)

The matrix W_n is invertible if $\theta \neq (2k-1)\pi/2$, $(k \in Z)$.

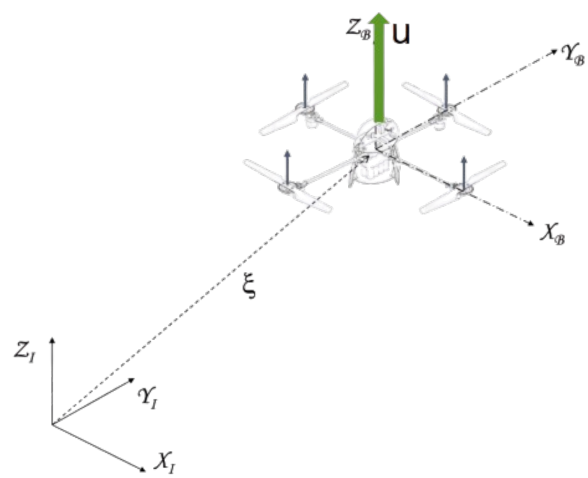


Figure 2: Quadcopter frame system with a vehicle frame and inertial frame

2.1 Dynamics of quadcopter using Newton-Euler approach

The dynamics model is obtained based on the following assumptions:

- The body frame of the quadcopter is rigid and symmetric.
- The center of the body frame coincides with the center of gravity.
- The aerodynamics effects are neglected.

The angular velocity of rotor i, denoted with ω_i, creates force f_i in the direction of the rotor axis. The angular velocity and acceleration of the rotor also create torque τ_{M_i} around the rotor axis.

$$\begin{cases} f_i = k\omega_i^2 \\ \tau_{M_i} = d\omega_i^2 + I_M \dot{\omega}_i \end{cases} \tag{4}$$

in which the lift constant is k, the drag constant is d and the inertia moment of the rotor is I_M. Usually the effect of $\dot{\omega}_i$ is considered small and thus it is omitted.

The combined forces of rotors create thrust T in the direction of the body z-axis. Torque τ^B consists of the torques τ_ϕ, τ_θ and τ_ψ in the direction of the corresponding body frame angles:

$$T = \sum_{i=1}^{4} f_i = k \sum_{i=1}^{4} \Omega_i^2, \quad T^B = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \tag{5}$$

$$\tau^B = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} lk(\Omega_2^2 - \Omega_4^2) \\ lk(\Omega_3^2 - \Omega_1^2) \\ d(\Omega_3^2 - \Omega_1^2 + \Omega_2^2 - \Omega_4^2) \end{bmatrix}. \tag{6}$$

in which l is the distance between the rotor and the center of mass of the quadcopter. Pitch movement is obtained by increasing (reducing) the 3^{rd} rotor velocity while reducing (increasing) the 1^{st} rotor velocity. The roll movement is obtained by increasing (reducing) the 2^{nd} rotor velocity while reducing (increasing) the 4^{th} rotor velocity. The Yaw movement is acquired by increasing the the angular velocities of two opposite rotors and decreasing the velocities of the other two.
Define U as input of the quadcopter:

$$U = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix} = \begin{bmatrix} T & \tau_\phi & \tau_\theta & \tau_\psi \end{bmatrix}^T \tag{7}$$

By using $Newton - Euler$ method:

$$\mathbf{m}\ddot{x} = \sum \mathbf{F} \tag{8}$$

Where m is the quadrotor total mass, $\mathbf{I}$ is inertia matrix, $\sum \mathbf{F}$ are the external forces acting on the quadrotor system.
In an inertial frame, the centrifugal force is effectively eliminated. Consequently, only the gravitational force and the magnitude and direction of the thrust are contributing in the acceleration of the quadcopter

$$m\ddot{\xi} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \mathbf{R} \cdot \mathbf{T}^B \tag{9}$$

$$\mathbf{T}^B = \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix} \tag{10}$$

where F_B is the total forces acting on the quadcopter in body frame.
By substituting (10) into (9), the following equations can be obtained:

$$\ddot{x} = \frac{-u_1}{m} \sin\theta \tag{11}$$

$$\ddot{y} = \frac{u_1}{m} \cos\theta \sin\phi \tag{12}$$

$$\ddot{z} = \frac{u_1}{m} \cos\theta \cos\phi - g \tag{13}$$

Now By applying Newton's law to rotational motion, we have:

$$J\dot{\omega} + \omega \times J\omega + \omega \times \begin{bmatrix} 0 \\ 0 \\ J_r\Omega_r \end{bmatrix} = \tau^B \tag{14}$$

where J is the Quadrotor's diagonal inertia matrix, ω is the angular body rates, J_r is the rotors' inertia, Ω_r is the rotors' relative speed $\Omega_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$, and $\tau^B = (\tau_\phi, \tau_\theta, \tau_\psi)^T$ represents the applied torques around the x, y, and z axes, in the body frame.
The first two terms of the previous equation $J\dot{\omega}$ and $\omega \times J\omega$ represent the rate of change of angular momentum in the body frame. The third term $\omega \times \begin{bmatrix} 0 \\ 0 \\ J_r\Omega_r \end{bmatrix}$ represents gyroscopic moments due to rotors' inertia J_r.

The inertia matrix for the Quadrotor is a diagonal matrix; the off-diagonal elements, which are the product of inertia, are zero due to the symmetry of the Quadrotor.

$$J = \begin{pmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{pmatrix}. \tag{15}$$

Finally we obtain:

$$\begin{pmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} \dot{\theta}\dot{\psi}\left(\frac{J_y - J_z}{J_x}\right) - \frac{J_r}{J_x}\dot{\theta}\Omega + \frac{U_2}{J_x} \\ \dot{\phi}\dot{\psi}\left(\frac{J_z - J_x}{I_y}\right) - \frac{J_r}{J_y}\dot{\phi}\Omega + \frac{U_3}{J_y} \\ \dot{\phi}\dot{\theta}\left(\frac{J_x - J_y}{J_z}\right) + \frac{U_4}{J_z} \end{pmatrix} \tag{16}$$

2.2 State Space Representation

Below is a state space representation that details the entire mathematical model of the quadrotor:

$$X = \begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_{12} \end{bmatrix}^T = \begin{bmatrix} x_1 = \phi \\ x_2 = \dot{\phi} \\ x_3 = \theta \\ x_4 = \dot{\theta} \\ x_5 = \psi \\ x_6 = \dot{\psi} \\ x_7 = z \\ x_8 = \dot{z} \\ x_9 = x \\ x_{10} = \dot{x} \\ x_{11} = y \\ x_{12} = \dot{y} \end{bmatrix}$$

(17)

$$\dot{X} = f(X, U) = \begin{bmatrix} x_2 \\ x_4 x_6 \left(\frac{I_y - I_z}{I_x}\right) - \frac{J_r}{I_x} x_4 \Omega + \frac{U_2}{I_x} \\ x_4 \\ x_2 x_6 \left(\frac{I_z - I_x}{I_y}\right) - \frac{J_r}{I_y} x_2 \Omega + \frac{U_3}{I_y} \\ x_6 \\ x_2 x_4 \dot{\theta} \left(\frac{I_x - I_y}{I_z}\right) + \frac{U_4}{I_z} \\ x_8 \\ \frac{u_1}{m}(\cos x_1 \cos x_3) - g \\ x_{10} \\ -\frac{u_1}{m}\sin x_3 \\ \\ x_{12} \\ \frac{u_1}{m}\cos x_3 \sin x_1 \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \\ \dot{z} \\ \ddot{z} \\ \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \end{bmatrix}$$

(18)

to simplify, the coriolis terms are neglected we get these much simpler set of equations; :

$$\dot{X} = f(X, U) = \begin{bmatrix} x_2 \\ x_4 x_6 a_1 + U_2 b1 \\ x_4 \\ x_2 x_6 a_2 + U_3 b_2 \\ x_6 \\ x_2 x_4 a_3 + U_4 b_3 \\ x_8 \\ \frac{u_1}{m}(\cos x_1 \cos x_3) - g \\ x_{10} \\ \frac{u_1}{m} U_x \\ \\ \frac{u_1}{m} U_y \end{bmatrix} \; where: \begin{bmatrix} U_x & = -\sin x_3, & \\ U_y & = \cos x_3 \sin x_1, & \\ a_1 & = \frac{I_y - I_z}{I_x}, & b_1 = \frac{l}{J_x}, \\ a_2 & = \frac{I_z - I_x}{I_y}, & b_2 = \frac{l}{J_y}, \\ a_3 & = \frac{I_x - I_y}{I_z}, & b_3 = \frac{l}{J_z}. \end{bmatrix}$$

$$\begin{bmatrix} u_\phi \\ u_\theta \\ u_\psi \\ u \end{bmatrix} = \begin{bmatrix} d(F_2 - F_4) \\ d(F_3 - F_1) \\ c(F_1 - F_2 + F_3 - F_4) \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix}$$

Finally we arrive to this system model

$$\begin{aligned}
\ddot{x} &= -\frac{1}{m}\sin(\theta)u, \\
\ddot{y} &= \frac{1}{m}\cos(\theta)\sin(\phi)u, \\
\ddot{z} &= \frac{1}{m}\cos(\theta)\cos(\phi)u - g, \\
\ddot{\phi} &= \dot{\theta}\dot{\psi}\left(\frac{J_y - J_z}{J_x}\right) - \frac{J_r}{J_x}\dot{\theta}\Omega + \frac{1}{J_x}u_\phi, \\
\ddot{\theta} &= \dot{\phi}\dot{\psi}\left(\frac{J_z - J_x}{J_y}\right) - \frac{J_r}{J_y}\dot{\phi}\Omega + \frac{1}{J_y}u_\theta, \\
\ddot{\psi} &= \dot{\theta}\dot{\phi}\left(\frac{J_x - J_y}{J_z}\right) + \frac{1}{J_z}u_\psi.
\end{aligned}$$

3 BACKSTEPPING CONTROL

3.1 Introduction

The main objective of this section is to synthesize the control law U, including stability analysis, based on the quadcopter's system model. This is achieved using the Backstepping control algorithm, which is a control design method particularly suited for strict-feedback systems, also known as "lower triangular" systems.

The Backstepping approach recursively designs virtual control inputs for each subsystem of the quadcopter, ensuring stability at each step using the Lyapunov function. This method is particularly useful for nonlinear systems, where the dynamics are complex and require step-by-step stabilization.

The controller aims to ensure that the position $\{x(t), y(t), z(t), \psi(t)\}$ asymptotically tracks the desired trajectory $\{x_d(t), y_d(t), z_d(t), \psi_d(t)\}$. To achieve this, the system is rewritten into three subsystems, each representing different aspects of the quadcopter's dynamics.

- **Underactuated Subsystem (S_1)**: This subsystem encapsulates both translational and partial rotational dynamics. It includes the horizontal plane motion (position x, y and linear velocities $\dot{x}, \dot{y}$) as well as the roll (ϕ) and pitch (θ) angles and their corresponding angular velocities.
- **Fully-Actuated Subsystem (S_2)**: This subsystem deals with the yaw (ψ) and vertical (z) dynamics, including their angular and linear velocities. The yaw angle and vertical position are directly influenced by the control inputs, allowing precise stabilization in these directions.
- **Propeller Subsystem (S_3)**: This subsystem represents the quadcopter's actuator dynamics, namely the propellers. It directly maps the control inputs ($U = [F_1, F_2, F_3, F_4]$) to the thrust and torques acting on the system. The objective is to generate the forces and torques required to stabilize and control the dynamics in the other two subsystems (S_1 and S_2).

3.2 Rewrite the system model

We rewrite the quadcopter's dynamic model in matrix form:

- **Underactuated Subsystem (S_1)**:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \frac{u}{m} \begin{bmatrix} -\sin(\theta) \\ \cos(\theta)\sin(\phi) \end{bmatrix} = \delta_0.\phi_0(\theta, \phi) \tag{20}$$

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{\theta}\dot{\psi}\left(\frac{J_y - J_z}{J_x}\right) - \frac{J_r}{J_x}\dot{\theta}\Omega \\ \dot{\phi}\dot{\psi}\left(\frac{J_z - J_x}{J_y}\right) - \frac{J_r}{J_y}\dot{\phi}\Omega \end{bmatrix} + \begin{bmatrix} \frac{1}{J_x} & 0 \\ 0 & \frac{1}{J_y} \end{bmatrix} \begin{bmatrix} u_\phi \\ u_\theta \end{bmatrix} = f_{\phi\theta} + \delta_1.\phi_1(U) \tag{21}$$

- **Fully-Actuated Subsystem (S_2)**:

$$\begin{bmatrix} \ddot{\psi} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} \dot{\theta}\dot{\phi}\left(\frac{J_x - J_y}{J_z}\right) \\ -g \end{bmatrix} + \begin{bmatrix} \frac{1}{J_z} & 0 \\ 0 & \frac{1}{m}\cos(\theta)\cos(\phi) \end{bmatrix} \begin{bmatrix} u_\psi \\ u \end{bmatrix} = f_{\psi z} + \delta_2.\phi_2(U) \tag{22}$$

- **Propeller Subsystem (S_3)**:

$$\begin{bmatrix} U \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{23}$$

Then we represent the quadcopter's dynamic model in state-space form to facilitate the backstepping control design :

$$(S_1) : \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \delta_0.\phi_0(x_3) \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_{\phi\theta} + \delta_1.\phi_1(x_7) \end{cases}$$

$$(S_2) : \begin{cases} \dot{x}_5 = x_6 \\ \dot{x}_6 = f_{\psi z} + \delta_2.\phi_2(x_7) \end{cases}$$

$$(S_3) : \left\{ \dot{x}_7 = u \right. \tag{24}$$

With :

$$x_1 = \begin{bmatrix} x \\ y \end{bmatrix}, x_2 = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}, x_3 = \begin{bmatrix} \phi \\ \theta \end{bmatrix}, x_4 = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \end{bmatrix}, x_5 = \begin{bmatrix} \psi \\ z \end{bmatrix}, x_6 = \begin{bmatrix} \dot{\psi} \\ \dot{z} \end{bmatrix}, x_7 = \begin{bmatrix} F_1 F_2 F_3 F_4 \end{bmatrix}$$

Each subsystem is analyzed and controlled step-by-step using Backstepping, ensuring stability at every stage. The coupling between these subsystems is addressed through the recursive design process, resulting in a unified control law U.

3.3 Synthesize the control law

Using the backstepping methodology, we will synthesize the control law U in seven steps. We will start by stabilizing the **Underactuated Subsystem** (S_1).

- **Step 1**: for the first step, we define error :

$$e_1 = x_1 - x_{1d} \tag{25}$$

The time derivative of error e_1 is :

$$\dot{e}_1 = \dot{x}_1 - \dot{x}_{1d} = x_2 - x_{2d} \tag{26}$$

with : $\quad x_{2d} = \dot{x}_{1d}$

Propose the Lyapunov function :

$$V_1 = \frac{1}{2} e_1^T e_1 \tag{27}$$

Its time derivative is :

$$\dot{V}_1 = e_1^T \dot{e}_1 = e_1^T \left(x_2 - x_{2d}\right) \tag{28}$$

The stabilization of e_1 can be obtained by introducing the virtual control law :

$$\alpha_1(e_1) = -k_1 e_1 + x_{2d} \tag{29}$$

with $k_1 \in R^{2\times 2}$ a positive definite matrix.

so, when :

$$x_2 \to \alpha_1(e_1)$$

The time derivative of the Lyapunov function is then :

$$\dot{V}_1 = -e_1^T k_1 e_1 < 0 \tag{30}$$

- **Step 2**: we define error :

$$e_2 = x_2 - \alpha_1(e_1) = x_2 - x_{2d} + k_1 e_1 = \dot{e}_1 + k_1 e_1 \tag{31}$$

Implies that :

$$\dot{e}_1 = e_2 - k_1 e_1 \tag{32}$$

The time derivative of error e_2 is :

$$\dot{e}_2 = \dot{x}_2 - \dot{\alpha}_1(e_1) = \delta_0.\phi_0(x_3) - \dot{\alpha}_1(e_1) \tag{33}$$

Propose the Lyapunov function :

$$V_2 = \frac{1}{2} \sum_{i=1}^{2} e_i^T e_i \tag{34}$$

Its time derivative is :

$$\dot{V}_2 = \sum_{i=1}^{2} e_i^T \dot{e}_i = -e_1^T k_1 e_1 + e_2^T \left(e_1 + \delta_0.\phi_0(x_3) - \dot{\alpha}_1(e_1)\right) \tag{35}$$

The stabilization of e_2 can be obtained by introducing the virtual control law :

$$\alpha_2\left(e_1, e_2\right) = \delta_0^{-1}.\left(-k_2 e_2 - e_1 + \dot{\alpha}_1(e_1)\right) \tag{36}$$

with $k_2 \in R^{2\times 2}$ a positive definite matrix.
Also,

$$\delta_0 = \frac{u}{m} \Rightarrow \delta_0^{-1} = \frac{m}{u}, u \neq 0$$

so, when :

$$\phi_0(x_3) \rightarrow \alpha_2\left(e_1, e_2\right)$$

The time derivative of the Lyapunov function is then :

$$\dot{V}_2 = -\sum_{i=1}^{2} e_i^T k_i e_i < 0 \tag{37}$$

- **Step 3**: we define error :

$$e_3 = \phi_0(x_3) - \alpha_2\left(e_1, e_2\right) = \phi_0(x_3) - \delta_0^{-1}.\left(-k_2 e_2 - e_1 + \dot{\alpha}_1(e_1)\right) \tag{38}$$

$$\Rightarrow \delta_0 e_3 = \delta_0 \phi_0(x_3) - \dot{\alpha}_1(e_1) + k_2 e_2 + e_1 = \dot{e}_2 + k_2 e_2 + e_1 \tag{39}$$

Implies that :

$$\dot{e}_2 = \delta_0 e_3 - k_2 e_2 - e_1 \tag{40}$$

The time derivative of error e_3 is :

$$\dot{e}_3 = \dot{\phi}_0(x_3) - \dot{\alpha}_2\left(e_1, e_2\right) = J_0.\dot{x}_3 - \dot{\alpha}_2\left(e_1, e_2\right) = J_0.x_4 - \dot{\alpha}_2\left(e_1, e_2\right) \tag{41}$$

with :

$$\dot{\phi}_0(x_3) = J_0.\dot{x}_3$$

and, J_0 is the jacobian matrix of $\phi_0(x_3)$ such as :

$$J_0 = \frac{\partial \varphi_0(x_3)}{\partial x_3} = \begin{bmatrix} 0 & -\cos(\theta) \\ \cos(\theta)\cos(\phi) & -\sin(\theta)\sin(\phi) \end{bmatrix} \tag{42}$$

Also, the determinant of the jacobian matrix J_0 : $\cos(\theta)^2\cos(\phi) \neq 0 \Leftrightarrow \left(-\frac{\pi}{2} < \phi < \frac{\pi}{2}\right)$ and $\left(-\frac{\pi}{2} < \theta < \frac{\pi}{2}\right)$
Therefore, J_0 is invertible.

Propose the Lyapunov function :

$$V_3 = \frac{1}{2}\sum_{i=1}^{3} e_i^T e_i \tag{43}$$

Its time derivative is :

$$\dot{V}_3 = \sum_{i=1}^{3} e_i^T \dot{e}_i = -\sum_{i=1}^{2} e_i^T k_i e_i + e_3^T\left(\delta_0^{-1}.e_2 + J_0.x_4 - \dot{\alpha}_2\left(e_1, e_2\right)\right) \tag{44}$$

The stabilization of e_3 can be obtained by introducing the virtual control law :

$$\alpha_3\left(e_1, e_2, e_3\right) = J_0^{-1}.\left(\dot{\alpha}_2\left(e_1, e_2\right) - \delta_0^{-1}.e_2 - k_3 e_3\right) \tag{45}$$

with $k_3 \in R^{2\times 2}$ a positive definite matrix.
so, when :

$$x_4 \rightarrow \alpha_3\left(e_1, e_2, e_3\right)$$

The time derivative of the Lyapunov function is then :

$$\dot{V}_3 = -\sum_{i=1}^{3} e_i^T k_i e_i < 0 \tag{46}$$

- **Step 4**: we define error :

$$e_4 = x_4 - \alpha_3\left(e_1, e_2, e_3\right) = x_4 - J_0^{-1}.\left(\dot{\alpha}_2\left(e_1, e_2\right) - \delta_0^{-1} e_2 - k_3 e_3\right) \tag{47}$$

$$\Rightarrow J_0 e_4 = J_0 x_4 - \dot{\alpha}_2\left(e_1, e_2\right) + \delta_0^{-1} e_2 + k_3 e_3 = \dot{e}_3 + \delta_0^{-1} e_2 + k_3 e_3 \tag{48}$$

Implies that :

$$\dot{e}_3 = J_0 e_4 - \delta_0^{-1} e_2 - k_3 e_3 \tag{49}$$

The time derivative of error e_4 is :

$$\dot{e}_4 = \dot{x}_4 - \dot{\alpha}_3\left(e_1, e_2, e_3\right) = f_{\phi\theta} + \delta_1.\phi_1(x_7) - \dot{\alpha}_3\left(e_1, e_2, e_3\right) \tag{50}$$

Propose the Lyapunov function :

$$V_4 = \frac{1}{2}\sum_{i=1}^{4} e_i^T e_i \tag{51}$$

Its time derivative is :

$$\dot{V}_4 = \sum_{i=1}^{4} e_i^T \dot{e}_i = -\sum_{i=1}^{3} e_i^T k_i e_i + e_4^T\left(J_0^{-1}.e_3 + f_{\phi\theta} + \delta_1.\phi_1(x_7) - \dot{\alpha}_3\left(e_1, e_2, e_3\right)\right) \tag{52}$$

The stabilization of e_4 can be obtained by introducing the virtual control law :

$$\alpha_4\left(e_1, e_2, e_3, e_4\right) = \delta_1^{-1}.\left(-k_4 e_4 - J_0^{-1}.e_3 - f_{\phi\theta} + \dot{\alpha}_3\left(e_1, e_2, e_3\right)\right) \tag{53}$$

with $k_4 \in R^{2\times 2}$ a positive definite matrix.
Also, the determinant of δ_1 : $\frac{1}{J_x J_y} \neq 0$. Therefore, δ_1 is invertible, iff $J_x \neq 0$ and $J_y \neq 0$.
so, when :

$$\phi_1(x_7) \rightarrow \alpha_4\left(e_1, e_2, e_3, e_4\right)$$

The time derivative of the Lyapunov function is then :

$$\dot{V}_4 = -\sum_{i=1}^{4} e_i^T k_i e_i < 0 \tag{54}$$

Consequently, the **Underactuated Subsystem (S_1)** is asymptotically stable with the virtual control laws : $\alpha_1\left(e_1\right), \alpha_2\left(e_1, e_2\right), \alpha_3\left(e_1, e_2, e_3\right)$, and $\alpha_4\left(e_1, e_2, e_3, e_4\right)$

- **Step 5**: We move to stabilize the **Fully-Actuated Subsystem (S_2)**, we define error :

$$e_5 = x_5 - x_{5d} \tag{55}$$

The time derivative of error e_5 is :

$$\dot{e}_5 = \dot{x}_5 - \dot{x}_{5d} = x_6 - x_{6d} \tag{56}$$

with : $x_{6d} = \dot{x}_{5d}$

Propose the Lyapunov function :

$$V_5 = \frac{1}{2}{e_5}^T e_5 \tag{57}$$

Its time derivative is :

$$\dot{V}_5 = {e_5}^T \dot{e}_5 = {e_5}^T . (x_6 - x_{6d}) \tag{58}$$

The stabilization of e_5 can be obtained by introducing the virtual control law :

$$\alpha_5(e_5) = -k_5 e_5 + x_{6d} \tag{59}$$

with $k_5 \in R^{2\times 2}$ a positive definite matrix.

so, when :

$$x_6 \to \alpha_5(e_5)$$

The time derivative of the Lyapunov function is then :

$$\dot{V}_5 = -e_5^T k_5 e_5 < 0 \tag{60}$$

- **Step 6**: we define error :

$$e_6 = x_6 - \alpha_5(e_5) = x_6 - x_{6d} + k_5 e_5 = \dot{e}_5 + k_5 e_5 \tag{61}$$

Implies that :

$$\dot{e}_5 = e_6 - k_5 e_5 \tag{62}$$

The time derivative of error e_6 is :

$$\dot{e}_6 = \dot{x}_6 - \dot{\alpha}_5(e_5) = f_{\psi z} + \delta_2 . \phi_2(x_7) - \dot{\alpha}_5(e_5) \tag{63}$$

Propose the Lyapunov function :

$$V_6 = \frac{1}{2} \sum_{i=5}^{6} e_i^T e_i \tag{64}$$

Its time derivative is :

$$\dot{V}_6 = \sum_{i=5}^{6} e_i^T \dot{e}_i = -e_5^T k_5 e_5 + e_6^T \left(e_5 + f_{\psi z} + \delta_2 . \phi_2(x_7) - \dot{\alpha}_5(e_5)\right) \tag{65}$$

The stabilization of e_6 can be obtained by introducing the virtual control law :

$$\alpha_6\left(e_5, e_6\right) = \delta_2^{-1} . \left(-k_6 e_6 - e_5 - f_{\psi z} + \dot{\alpha}_5(e_5)\right) \tag{66}$$

with $k_6 \in R^{2\times 2}$ a positive definite matrix.
Also, the determinant of δ_2 : $\frac{1}{J_z m} \cos\theta \cos\phi \neq 0$. Therefore, δ_2 is invertible, iff $J_z m \neq 0$ and $\left(-\frac{\pi}{2} < \phi < \frac{\pi}{2}\right)$ and $\left(-\frac{\pi}{2} < \theta < \frac{\pi}{2}\right)$.

so, when :

$$\phi_2(x_7) \to \alpha_6\left(e_5, e_6\right)$$

The time derivative of the Lyapunov function is then :

$$\dot{V}_6 = -\sum_{i=5}^{6} e_i^T k_i e_i < 0 \tag{67}$$

Consequently, the **Fully-Actuated Subsystem (S_2)** is asymptotically stable with the virtual control laws : $\alpha_5\left(e_5\right)$, and $\alpha_6\left(e_5, e_6\right)$

- **Step 7**: we define error :

$$e_7 = \begin{bmatrix} \phi_1(x_7) - \alpha_4\left(e_1, e_2, e_3, e_4\right) \\ \phi_2(x_7) - \alpha_6\left(e_5, e_6\right) \end{bmatrix} = \begin{bmatrix} \phi_1(x_7) - \delta_1^{-1}.\left(-k_4 e_4 - J_0^{-1}.e_3 - f_{\phi\theta} + \dot{\alpha}_3\left(e_1, e_2, e_3\right)\right) \\ \phi_2(x_7) - \delta_2^{-1}.\left(-k_6 e_6 - e_5 - f_{\psi z} + \dot{\alpha}_5(e_5)\right) \end{bmatrix} \tag{68}$$

$$= \begin{bmatrix} \delta_1^{-1}.\left(\dot{e}_4 + k_4 e_4 + J_0^{-1}.e_3\right) \\ \delta_2^{-1}.\left(\dot{e}_6 + k_6 e_6 + e_5\right) \end{bmatrix} = \begin{bmatrix} \delta_1{}^{-1} & 0_{2\times 2} \\ 0_{2\times 2} & \delta_2{}^{-1} \end{bmatrix} \begin{bmatrix} \dot{e}_4 + k_4 e_4 + J_0^{-1} e_3 \\ \dot{e}_6 + k_6 e_6 + e_5 \end{bmatrix}$$

Implies that :

$$\begin{bmatrix} \dot{e}_4 \\ \dot{e}_6 \end{bmatrix} = \begin{bmatrix} \delta_1 & 0_{2\times 2} \\ 0_{2\times 2} & \delta_2 \end{bmatrix} e_7 - \begin{bmatrix} k_4 e_4 + J_0^{-1}.e_3 \\ k_6 e_6 + e_5 \end{bmatrix} = \begin{bmatrix} \delta_1^* e_7 - k_4 e_4 - J_0^{-1} e_3 \\ \delta_2^* e_7 - k_6 e_6 - e_5 \end{bmatrix} \tag{69}$$

The time derivative of error e_7 is :

$$\dot{e}_7 = \begin{bmatrix} \dot{\phi}_1(x_7) - \dot{\alpha}_4\left(e_1, e_2, e_3, e_4\right) \\ \dot{\phi}_2(x_7) - \dot{\alpha}_6\left(e_5, e_6\right) \end{bmatrix} = \begin{bmatrix} J_1 \dot{x}_7 - \dot{\alpha}_4\left(e_1, e_2, e_3, e_4\right) \\ J_2 \dot{x}_7 - \dot{\alpha}_6\left(e_5, e_6\right) \end{bmatrix} = \begin{bmatrix} J_1 \\ J_2 \end{bmatrix} u - \begin{bmatrix} \dot{\alpha}_4\left(e_1, e_2, e_3, e_4\right) \\ \dot{\alpha}_6\left(e_5, e_6\right) \end{bmatrix} \tag{70}$$

where $J_1 = \frac{\partial \varphi_1(x_7)}{\partial x_7}$ and $J_2 = \frac{\partial \varphi_2(x_7)}{\partial x_7}$ are the *Jacobian* matrices of φ_1 and φ_2 such as:

$$J_1 = \begin{bmatrix} 0 & d & 0 & -d \\ -d & 0 & d & 0 \end{bmatrix}, \quad J_2 = \begin{bmatrix} c & -c & c & -c \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{71}$$

Propose the Lyapunov function :

$$V_7 = \frac{1}{2} \sum_{i=1}^{7} e_i^T e_i \tag{72}$$

Its time derivative is :

$$\dot{V}_7 = \sum_{i=1}^{7} e_i^T \dot{e}_i = -\sum_{i=1}^{7} e_i^T k_i e_i + e_7^T \left(\begin{bmatrix} \delta_1 & 0_{2\times 2} \\ 0_{2\times 2} & \delta_2 \end{bmatrix}^T \begin{bmatrix} e_4 \\ e_6 \end{bmatrix} + \begin{bmatrix} J_1 \\ J_2 \end{bmatrix} u - \begin{bmatrix} \dot{\alpha}_4\left(e_1, e_2, e_3, e_4\right) \\ \dot{\alpha}_6\left(e_5, e_6\right) \end{bmatrix} \right) \tag{73}$$

Therefore, the stabilization of the whole system can be obtained by introducing the following control law:

$$u = \begin{bmatrix} J_1 \\ J_2 \end{bmatrix}^{-1} \left(-k_7 e_7 - \begin{bmatrix} \delta_1 & 0_{2\times 2} \\ 0_{2\times 2} & \delta_2 \end{bmatrix}^T \begin{bmatrix} e_4 \\ e_6 \end{bmatrix} + \begin{bmatrix} \dot{\alpha}_4\left(e_1, e_2, e_3, e_4\right) \\ \dot{\alpha}_6\left(e_5, e_6\right) \end{bmatrix} \right) \tag{74}$$

with $k_7 \in R^{4\times 4}$ is a positive definite matrix. Also, the determinant of $\begin{bmatrix} J_1 \\ J_2 \end{bmatrix}$: $8cd^2 \neq 0$. Therefore, the matrix is invertible, iff $c \neq 0$, and $d \neq 0$.
The time derivative of the Lyapunov function is then :

$$\dot{V}_7 = -\sum_{i=1}^{7} e_i^T k_i e_i < 0 \tag{75}$$

Consequently, the whole system is asymptotically stable with the following control law:

$$(S_1):\begin{cases}\alpha_1(e_1) = -k_1e_1 + x_{2d} \\ \alpha_2\left(e_1,e_2\right) = \delta_0^{-1}.\left(-k_2e_2 - e_1 + \dot{\alpha}_1(e_1)\right) \\ \alpha_3\left(e_1,e_2,e_3\right) = J_0^{-1}.\left(\dot{\alpha}_2\left(e_1,e_2\right) - \delta_0^{-1}.e_2 - k_3e_3\right) \\ \alpha_4\left(e_1,e_2,e_3,e_4\right) = \delta_1^{-1}.\left(-k_4e_4 - J_0^{-1}.e_3 - f_{\phi\theta} + \dot{\alpha}_3\left(e_1,e_2,e_3\right)\right)\end{cases}$$

$$(S_2):\begin{cases}\alpha_5(e_5) = -k_5e_5 + x_{6d} \\ \alpha_6\left(e_5,e_6\right) = \delta_2^{-1}.\left(-k_6e_6 - e_5 - f_{\psi z} + \dot{\alpha}_5(e_5)\right)\end{cases}$$

$$(S_3):\begin{cases}u = \begin{bmatrix}J_1\\J_2\end{bmatrix}^{-1}\left(-k_7e_7 - \begin{bmatrix}\delta_1 & 0_{2\times2}\\0_{2\times2} & \delta_2\end{bmatrix}^T\begin{bmatrix}e_4\\e_6\end{bmatrix} + \begin{bmatrix}\dot{\alpha}_4\left(e_1,e_2,e_3,e_4\right)\\ \dot{\alpha}_6\left(e_5,e_6\right)\end{bmatrix}\right)\end{cases} \tag{76}$$

Let's rewrite the system model :

$$(S_1):\begin{cases}\dot{e}_1 = e_2 - k_1e_1 \\ \dot{e}_2 = \delta_0e_3 - k_2e_2 - e_1 \\ \dot{e}_3 = J_0e_4 - \delta_0^{-1}e_2 - k_3e_3 \\ \dot{e}_4 = \delta_1^*e_7 - k_4e_4 - J_0^{-1}e_3\end{cases}$$

$$(S_2):\begin{cases}\dot{e}_5 = e_6 - k_5e_5 \\ \dot{e}_6 = \delta_2^*e_7 - k_6e_6 - e_5\end{cases}$$

$$(S_3):\begin{cases}\dot{e}_7 = \begin{bmatrix}\phi_1(x_7) - \delta_1^{-1}.\left(-k_4e_4 - J_0^{-1}.e_3 - f_{\phi\theta} + \dot{\alpha}_3\left(e_1,e_2,e_3\right)\right) \\ \phi_2(x_7) - \delta_2^{-1}.\left(-k_6e_6 - e_5 - f_{\psi z} + \dot{\alpha}_5(e_5)\right)\end{bmatrix}\end{cases} \tag{77}$$

4 BACKSTEPPING 2

During the development process, two backstepping approaches were explored. The first approach, although theoretically sound, proved challenging to implement due to the complexity of the derived equations. As a result, a second, more practical backstepping formulation was adopted, which simplified the implementation while retaining robust performance.

4.1 ϕ Control and Stability

We start by defining the tracking error e_1:

$$e_1 = x_{1d} - x_1, \quad \dot{e}_1 = \dot{x}_{1d} - x_2 \tag{78}$$

Next, we introduce a Lyapunov function V_1:

$$V_1 = \frac{1}{2}e_1^2, \quad \dot{V}_1 = e_1\dot{e}_1 = e_1(\dot{x}_{1d} - x_2) \tag{79}$$

To ensure $\dot{V}_1 < 0$, we select a virtual input x_2^v:

$$x_2^v = \dot{x}_{1d} + K_1e_1, \quad \text{with } K_1 > 0 \tag{80}$$

We then define a new error e_2 between x_2 and x_2^v:

$$e_2 = x_2 - x_2^v = x_2 - \dot{x}_{1d} - K_1e_1 \tag{81}$$

$$\dot{e}_2 = \dot{x}_2 - \ddot{x}_{1d} - K_1\dot{e}_1 \tag{82}$$

Rewriting $\dot{V}_1$ with these definitions yields:

$$\dot{V}_1 = -e_1e_2 - K_1e_1^2 \tag{83}$$

Considering an augmented Lyapunov function $V_2 = V_1 + \frac{1}{2}e_2^2$, we have:

$$\dot{V}_2 = -e_1e_2 - K_1e_1^2 + e_2(\dot{x}_2 - \ddot{x}_{1d} - K_1\dot{e}_1) \tag{84}$$

$$\dot{V}_2 = -K_1e_1^2 + e_2(x_4x_6a_1 + U_2b_1 - \ddot{x}_{1d} - K_1e_1 - e_1) \tag{85}$$

To ensure $\dot{V}_2 < 0$, we choose control input U_2 as:

$$U_2 = \frac{1}{b_1}(-x_4x_6a_1 + \ddot{x}_{1d} + K_1e_1 + e_1 - K_2e_2) \tag{86}$$

Ultimately, this results in:

$$\dot{V}_2 = -K_1e_1^2 - K_2e_2^2 < 0 \tag{87}$$

4.2 θ Control and Stability

We start by defining the tracking error e_3:

$$e_3 = x_{3d} - x_3, \quad \dot{e}_3 = \dot{x}_{3d} - x_4 \tag{88}$$

Next, we introduce a Lyapunov function V_3:

$$V_3 = \frac{1}{2}e_3^2, \quad \dot{V}_3 = e_3\dot{e}_3 = e_3(\dot{x}_{3d} - x_4) \tag{89}$$

To ensure $\dot{V}_3 < 0$, we select a virtual input x_4^v:

$$x_4^v = \dot{x}_{3d} + K_3e_3, \quad \text{with } K_3 > 0 \tag{90}$$

We then define a new error e_4 between x_4 and x_4^v:

$$e_4 = x_4 - x_4^v = x_4 - \dot{x}_{3d} - K_3e_3 \tag{91}$$

$$\dot{e}_4 = \dot{x}_4 - \ddot{x}_{3d} - K_3\dot{e}_3 \tag{92}$$

Rewriting $\dot{V}_1$ with these definitions yields:

$$\dot{V}_1 = -e_1e_2 - K_1e_1^2 \tag{93}$$

Considering an augmented Lyapunov function $V_4 = V_3 + \frac{1}{2}e_4^2$, we have:

$$\dot{V}_2 = -e_1e_2 - K_1e_1^2 + e_2(\dot{x}_2 - \ddot{x}_{1d} - K_1\dot{e}_1) \tag{94}$$

$$\dot{V}_4 = -K_3e_3^2 + e_4(x_2x_6a_2 + U_3b_2 - \ddot{x}_{3d} - K_3\dot{e}_3 - e_3) \tag{95}$$

To ensure $\dot{V}_2 < 0$, we choose control input U_3 as:

$$U_3 = \frac{1}{b_2}(-x_2x_6a_2 + \ddot{x}_{3d} + K_1\dot{e}_3 + e_3) \tag{96}$$

Ultimately, this results in:

$$\dot{V}_4 = -K_3e_3^2 - K_4e_4^2 < 0 \tag{97}$$

4.3 ψ Control and Stability

We start by defining the tracking error e_5:

$$e_5 = x_{5d} - x_5, \quad \dot{e}_5 = \dot{x}_{5d} - x_6 \tag{98}$$

Next, we introduce a Lyapunov function V_5:

$$V_5 = \frac{1}{2}e_5^2, \quad \dot{V}_5 = e_5\dot{e}_5 = e_5(\dot{x}_{5d} - x_6) \tag{99}$$

To ensure $\dot{V}_5 < 0$, we select a virtual input x_6^v:

$$x_6^v = \dot{x}_{5d} + K_5 e_5, \quad \text{with } K_5 > 0 \tag{100}$$

We then define a new error e_6 between x_6 and x_6^v:

$$e_6 = x_6 - x_6^v = x_6 - \dot{x}_{5d} - K_5 e_5 \tag{101}$$

$$\dot{e}_6 = \dot{x}_6 - \ddot{x}_{5d} - K_5 \dot{e}_5 \tag{102}$$

Rewriting $\dot{V}_5$ with these definitions yields:

$$\dot{V}_5 = -e_5 e_6 - K_5 e_5^2 \tag{103}$$

Considering an augmented Lyapunov function $V_6 = V_5 + \frac{1}{2}e_6^2$, we have:

$$\dot{V}_6 = -e_5 e_6 - K_5 e_5^2 + e_6(\dot{x}_6 - \ddot{x}_{5d} - K_5 \dot{e}_5) \tag{104}$$

$$\dot{V}_6 = -K_5 e_5^2 + e_6(x_2 x_4 a_3 + U_4 b_3 - \ddot{x}_{5d} - K_5 \dot{e}_5 - e_5) \tag{105}$$

To ensure $\dot{V}_6 < 0$, we choose control input U_4 as:

$$U_4 = \frac{1}{b_3}(-x_2 x_4 a_3 + \ddot{x}_{5d} + K_5 \dot{e}_5 + e_5 - K_6 e_6) \tag{106}$$

Ultimately, this results in:

$$\dot{V}_6 = -K_5 e_5^2 - K_6 e_6^2 < 0 \tag{107}$$

4.4 Z Control and Stability

We start by defining the tracking error e_7:

$$e_7 = x_{7d} - x_7, \quad \dot{e}_7 = \dot{x}_{7d} - x_8 \tag{108}$$

Next, we introduce a Lyapunov function V_7:

$$V_7 = \frac{1}{2}e_7^2, \quad \dot{V}_7 = e_7\dot{e}_7 = e_7(\dot{x}_{7d} - x_8) \tag{109}$$

To ensure $\dot{V}_7 < 0$, we select a virtual input x_8^v:

$$x_8^v = \dot{x}_{7d} + K_7 e_7, \quad \text{with } K_7 > 0 \tag{110}$$

We then define a new error e_8 between x_8 and x_8^v:

$$e_8 = x_8 - x_8^v = x_7 - \dot{x}_{7d} - K_7 e_7 \tag{111}$$

$$\dot{e}_8 = \dot{x}_8 - \ddot{x}_{7d} - K_7 \dot{e}_7 \tag{112}$$

Rewriting $\dot{V}_7$ with these definitions yields:

$$\dot{V}_7 = -e_7 e_8 - K_7 e_7^2 \tag{113}$$

Considering an augmented Lyapunov function $V_8 = V_7 + \frac{1}{2}e_8^2$, we have:

$$\dot{V}_8 = -e_7 e_8 - K_7 e_7^2 + e_8(\dot{x}_8 - \ddot{x}_{7d} - K_7 \dot{e}_7) \quad (114)$$

$$\dot{V}_8 = -K_7 e_7^2 + e_8(\frac{U_1}{m}(cos(x_1)cos(x_3) - g - \ddot{x}_{7d} - K_7 \dot{e}_5 - e_7) \quad (115)$$

To ensure $\dot{V}_8 < 0$, we choose control input U_1 as:

$$U_1 = \frac{m}{cos(x_1)cos(x_3)}(g + \ddot{x}_{7d} + K_7 \dot{e}_7 + e_7 - K_8 e_8) \quad (116)$$

Ultimately, this results in:

$$\dot{V}_8 = -K_7 e_7^2 - K_8 e_8^2 < 0 \quad (117)$$

4.5 X Control and Stability

We start by defining the tracking error e_9:

$$e_9 = x_{9d} - x_9, \quad \dot{e}_9 = \dot{x}_{9d} - x_1 0 \quad (118)$$

Next, we introduce a Lyapunov function V_9:

$$V_9 = \frac{1}{2}e_9^2, \quad \dot{V}_9 = e_9 \dot{e}_9 = e_9(\dot{x}_{9d} - x_1 0) \quad (119)$$

To ensure $\dot{V}_9 < 0$, we select a virtual input $x_1 0^v$:

$$x_1 0^v = \dot{x}_{9d} + K_9 e_9, \quad \text{with } K_9 > 0 \quad (120)$$

We then define a new error $e_1 0$ between $x_1 0$ and $x_1 0^v$:

$$e_1 0 = x_1 0 - x_1 0^v = x_1 0 - \dot{x}_{9d} - K_9 e_9 \quad (121)$$

$$\dot{e}_1 0 = \dot{x}_1 0 - \ddot{x}_{9d} - K_9 \dot{e}_9 \quad (122)$$

Rewriting $\dot{V}_9$ with these definitions yields:

$$\dot{V}_9 = -e_9 e_1 0 - K_9 e_9^2 \quad (123)$$

Considering an augmented Lyapunov function $V_1 0 = V_9 + \frac{1}{2}e_1 0^2$, we have:

$$\dot{V}_1 0 = -e_9 e_1 0 - K_9 e_9^2 + e_1 0(\dot{x}_1 0 - \ddot{x}_{9d} - K_9 \dot{e}_9) \quad (124)$$

$$\dot{V}_1 0 = -K_9 e_9^2 + e_1 0(\frac{U_1}{m} U_x - \ddot{x}_{9d} - K_9 \dot{e}_9 - e_9) \quad (125)$$

To ensure $\dot{V}_1 0 < 0$, we choose control input U_x as:

$$U_x = \frac{m}{U_1}(\ddot{x}_{9d} + K_9 \dot{e}_9 + e_9 - K_{10} e_{10}) \quad (126)$$

Ultimately, this results in:

$$\dot{V}_{10} = -K_9 e_9^2 - K_{10} e_{10}^2 < 0 \quad (127)$$

4.6 Y Control and Stability

We start by defining the tracking error $e_{11]}$:

$$e_{11} = x_{11d} - x_{11}, \quad \dot{e}_{11} = \dot{x}_{11d} - x_{12} \tag{128}$$

Next, we introduce a Lyapunov function V_{11}:

$$V_{11} = \frac{1}{2}e_{11}^2, \quad \dot{V}_{11} = e_{11}\dot{e}_{11} = e_{11}(\dot{x}_{11d} - x_{12}) \tag{129}$$

To ensure $\dot{V}_{11} < 0$, we select a virtual input x_{12}^v:

$$x_{12}^v = \dot{x}_{11d} + K_{11}e_{11}, \quad \text{with } K_{11} > 0 \tag{130}$$

We then define a new error e_{12} between x_{12} and x_{12}^v:

$$e_{12} = x_{12} - x_{12}^v = x_{12} - \dot{x}_{11d} - K_{11}e_{11} \tag{131}$$

$$\dot{e}_{12} = \dot{x}_{12} - \ddot{x}_{11d} - K_{11}\dot{e}_{11} \tag{132}$$

Rewriting $\dot{V}_{11}$ with these definitions yields:

$$\dot{V}_{11} = -e_{11}e_{12} - K_{11}e_{11} \tag{133}$$

Considering an augmented Lyapunov function $V_1 2 we have: \dot{V}_{12} = -e_{11}e_{12} - K_{11}e_{11} + e_{12}\dot{e}_{12} (134)$

$$\dot{V}_{12} = -K_{11}e_{11} + e_{12}(\frac{U_1}{m}U_y - \ddot{x}_{11d} - K_{11}\dot{e}_{11} - e_{11}) \tag{135}$$

To ensure $\dot{V}_{12} < 0$, we choose control input U_y as:

$$U_y = \frac{m}{U_1}(\ddot{x}_{11d} + K_{11}\dot{e}_{11} + e_{11} - K_{12}e_{12}) \tag{136}$$

Ultimately, this results in:

$$\dot{V}_{12} = -K_{11}e_{11}^2 - K_{12}e_{12}^2 < 0 \tag{137}$$

5 Numerical Simulation

In our study, we devised a control strategy to effectively manage the trajectory of a quadcopter, with a detailed simulation framework developed in MATLAB. Our approach was grounded in the dynamic modeling of the quadcopter using Euler Newton equations, which allowed us to simulate flight dynamics under various conditions with precision.

5.1 Experimental Setup

Specific parameters reflective of common quadcopter configurations were selected to ensure realism and applicability of our findings:

- **Mass:** The quadcopter was assigned a mass of 1.63 kg to mimic a typical small UAV.
- **Gravitational Acceleration:** A standard gravitational pull of 10 N/kg was used to model the quadcopter's weight.
- **Moments of Inertia:** The moments of inertia were crucial for accurate simulation of rotational dynamics, set at $I_{xx} = 0.0151\,\text{kg.m}^2$, $I_{yy} = 0.0092\,\text{kg.m}^2$, and $I_{zz} = 0.0093\,\text{kg.m}^2$.

5.2 Control Gains

The control gains were meticulously chosen to optimize the response of the quadcopter to command inputs and disturbances. The following settings were implemented to fine-tune the control system:

Control Gain	**Value**
k_1, k_2	10, 0.5
k_3, k_4	2, 10
k_5, k_6	12, 10
k_7, k_8	10, 0.5
k_9, k_{10}	10, 15
k_{11}, k_{12}	8, 10

Table 1: Simulation Control Gains

These gains facilitated the quadcopter's ability to follow various trajectories, such as circular, spiral, and custom paths, which were defined using parametric equations. This control scheme allowed for dynamic adjustments based on the instantaneous position and velocity errors relative to the desired trajectory.

5.3 Cylinder Trajectory

We implemented a cylindrical trajectory in our quadcopter simulation to evaluate the system's stability and control effectiveness in both horizontal and vertical planes. This trajectory combines rotational movement around a central axis with a fixed vertical position, providing a consistent test of the quadcopter's ability to maintain a stable path.

The trajectory is defined by the following parametric equations, which describe a circular path with a constant radius of 10 units at an altitude of 4 units:

$$x(t) = 10\cos(0.2t), \tag{138}$$

$$y(t) = 10\sin(0.2t), \tag{139}$$

$$z(t) = 4. \tag{140}$$

The angular velocity is controlled by the factor 0.2 in the cosine and sine functions, dictating the rotation speed around the central axis.

The simulation began with the quadcopter positioned at the origin $(0, 0, 0)$, with all angles and angular velocities set to zero, except for the yaw angle which was initialized to $\frac{\pi}{3}$. This initial setup was chosen to test the system's responsiveness to angular displacement and its ability to align with the designated path effectively.

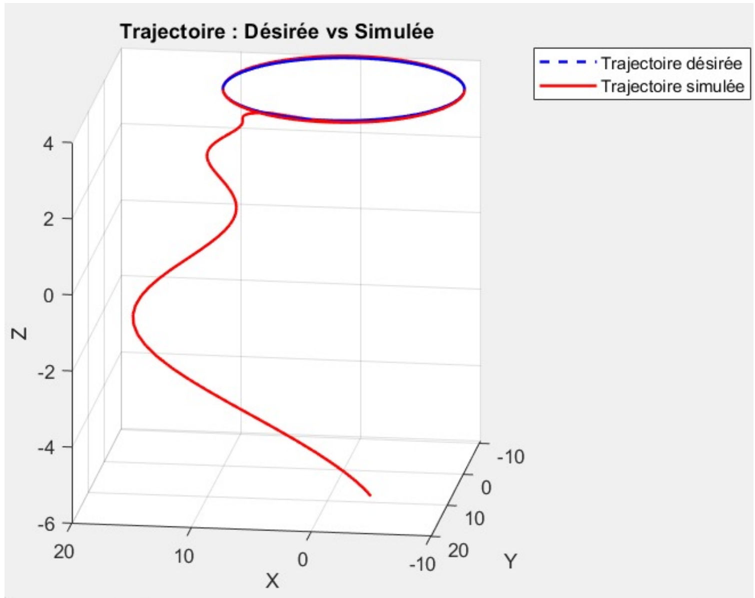


Figure 3: Cylinder trajectory

5.4 Reference Trajectory

In our quadcopter simulation, we implemented a linear or "reference" trajectory as a fundamental test to assess the system's tracking capabilities under straightforward and controlled conditions. This trajectory serves as a benchmark to evaluate basic stability and control effectiveness of the quadcopter.

Trajectory Definition The reference trajectory is defined by the following linear equations, which describe a straightforward path with constant velocities in all three spatial dimensions:

$$x(t) = x_0 + v_x t, \quad \text{where } x_0 = 50, v_x = 0.5, \tag{141}$$

$$y(t) = y_0 + v_y t, \quad \text{where } y_0 = 55, v_y = 0.5, \tag{142}$$

$$z(t) = z_0 + v_z t, \quad \text{where } z_0 = 20, v_z = 0.2. \tag{143}$$

These equations ensure the quadcopter follows a predetermined straight path, providing a clear metric for evaluating the accuracy and efficiency of the control algorithms.

Initial Conditions and Simulation Parameters The simulation starts with the quadcopter at the origin $(0, 0, 0)$ with all angular velocities set to zero, except for the yaw angle which was initialized to $\frac{\pi}{3}$. This initial condition tests the quadcopter's ability to correct its path from an angular displacement and align itself with the trajectory effectively.

This comprehensive approach in the simulation of the reference trajectory highlights the fundamental capabilities of the quadcopter, setting a baseline for further experiments and enhancements in more complex flight scenarios.

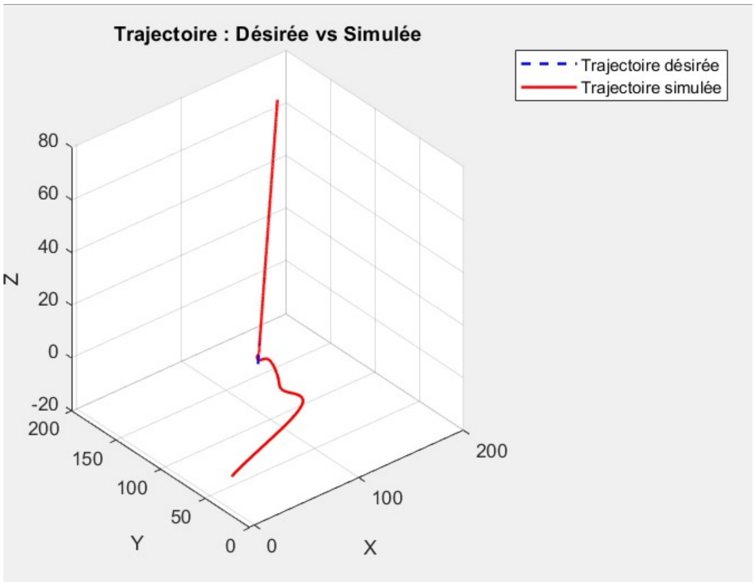


Figure 4: Reference trajectory

5.5 Spiral Trajectory

In our quadcopter simulation, we implemented a dynamically expanding spiral trajectory to test the control system's ability to maintain stability and tracking accuracy as the trajectory complexity increases over time.

The spiral trajectory is mathematically defined by the following parametric equations:

$$x(t) = r(t)\cos(\omega t), \tag{144}$$

$$y(t) = r(t)\sin(\omega t), \tag{145}$$

$$z(t) = z_0 + v_z t, \tag{146}$$

where $r(t) = r_0 + \alpha t$ describes the radius, which increases linearly with time. The parameters used in the simulation are:

- $r_0 = 2$ (initial radius),
- $\alpha = 0.1$ (rate of radius expansion),
- $\omega = 0.2$ (angular velocity),
- $z_0 = 0$ (initial altitude),
- $v_z = 0.1$ (vertical speed).

The simulation began with the quadcopter positioned at $(2, 1, 0)$ with all angles and angular velocities set to zero, except for the yaw angle, which was initialized to $\frac{\pi}{3}$. This initial condition was chosen to evaluate the control system's response to an angular offset and its ability to track the increasingly complex trajectory effectively.

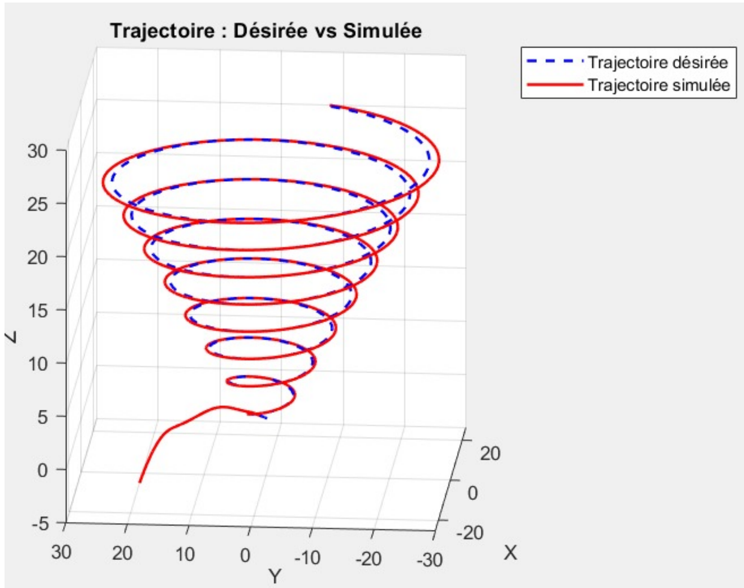


Figure 5: Spirale trajectory

6 Conclusion

In this project, we proposed a stabilization control algorithm for a quadcopter equipped with four rotors. The dynamic model of the quadcopter was derived using the Newton-Euler approach, providing a comprehensive representation of its physical behavior. The proposed control algorithm was developed based on the backstepping method, a robust approach for handling nonlinear systems. During the development process, two backstepping approaches were explored. The first approach, although theoretically sound, proved challenging to implement due to the complexity of the derived equations. As a result, a second, more practical backstepping formulation was adopted, which simplified the implementation while retaining robust performance.