ARS5 Final Project

Presented to:

Prof. Pedro CASTILLO GARCIA

Dr. Alessandro CORREA-VICTORINO

Jossué ESCOBAR

Presented by:

Stefano MUSSO PIANTELLI

Elias Charbel SALAMEH

utc
Université de Technologie
Compiègne

Problem Statement

Modeling of a Quadcopter System and Backstepping Control
to provide the stability analysis

Project Goals
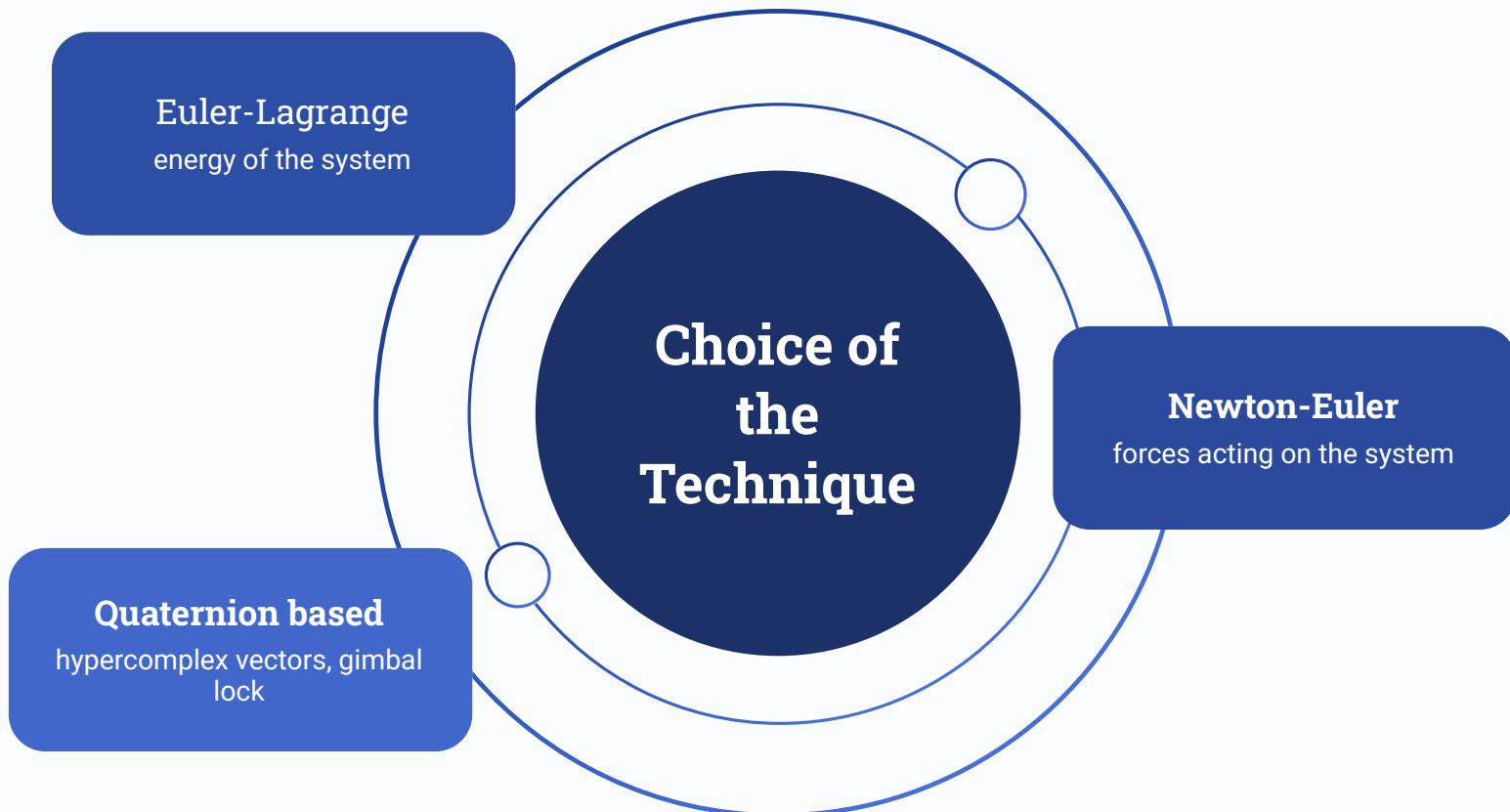
Develop a **complete model** robust to most of the forces that act on the drone

Avoid singularities

Have a stable system after conducting a thorough **stability analysis**

Convergence to fixed targets and **Follow** trajectories

utc
Université de Technologie
Compiègne

Modeling

Euler-Lagrange
energy of the system

Newton-Euler
forces acting on the system

Choice of the Technique

Quaternion based
hypercomplex vectors, gimbal lock

Modeling of the Quadcopter

State vector

$$\mathbb{X} = \begin{bmatrix} \xi^T & \dot{\xi}^T & \mathbf{q}^T & \Omega^T \end{bmatrix}^T$$

Modeling (cont'd)

Translational part

$$\dot{\mathbb{X}}_{pos} = \begin{bmatrix} \dot{\xi} \\ \ddot{\xi} \end{bmatrix} = \begin{bmatrix} \dot{\xi} \\ \mathbf{q} \otimes \frac{F_u}{m} \otimes \mathbf{q}^* + F_{ext} \end{bmatrix}$$

Modeling (cont'd)

Rotational part

$$\dot{\mathbb{X}}_{rot} = \begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\mathbf{q} \otimes \Omega \\ I^{-1}(\tau - \Omega \times I\Omega) \end{bmatrix}$$

Modeling (cont'd)

Overall base model

$$\dot{\mathbb{X}} = \begin{bmatrix} \dot{\xi} \\ \ddot{\xi} \\ \dot{\mathbf{q}} \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} \dot{\xi} \\ \mathbf{q} \otimes \frac{F_u}{m} \otimes \mathbf{q}^* + F_{ext} \\ \frac{1}{2}\mathbf{q} \otimes \Omega \\ I^{-1}(\tau - \Omega \times I\Omega) \end{bmatrix}$$

# Backstepping Control

---

Introduction
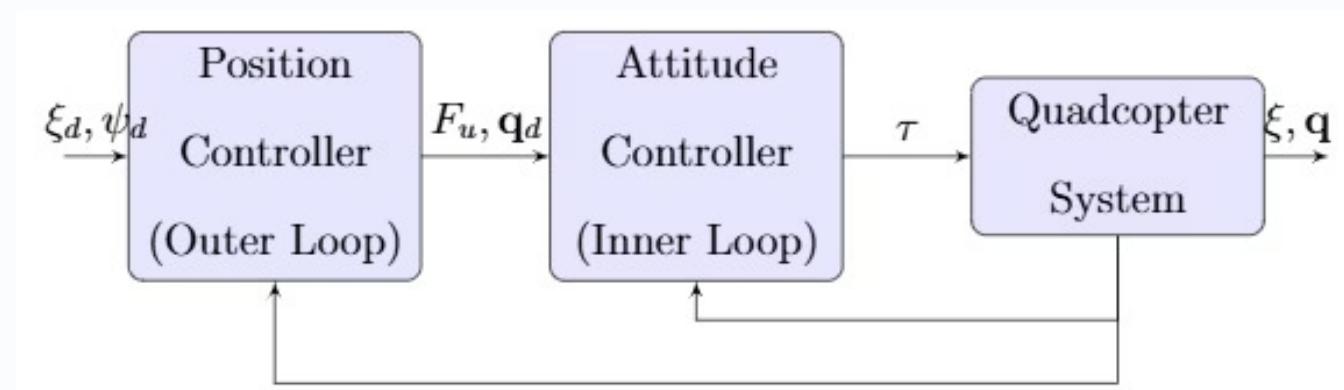
Non-Linear Control Methodology

Effective with 2+ integrator systems

Start with the translational part and follow up with the rotational part

Backstepping Control (cont'd)

Control Flow

Backstepping Control (cont'd)

Translational Part

Tracking error between the agent and the trajectory:

$$\mathbf{e}_{\xi,1} = -[\mathbf{a} * (\xi - \xi_{target}) - \mathbf{d}]$$

We propose:

$$V_1(\mathbf{e}_{\xi,1}) = \frac{\mathbf{e}_{\xi,1}^T \mathbf{e}_{\xi,1}}{2}$$

$$\dot{V}_1(\mathbf{e}_{\xi,1}) = \mathbf{e}_{\xi,1}^T \dot{\mathbf{e}}_{\xi,1} = \mathbf{e}_{\xi,1}^T * [-\mathbf{a}(\dot{\xi} - \dot{\xi}_{target})]$$

$$\dot{\xi}^v = \frac{k_1 \mathbf{e}_{\xi,1}}{a} + \dot{\xi}_{target}$$

$$\mathbf{e}_{\xi,2} = \dot{\xi} - \dot{\xi}^v$$

Our new system

$$\begin{cases} \dot{\mathbf{e}}_{\xi,1} = -\mathbf{a}.\mathbf{e}_{\xi,2} - k_1 \mathbf{e}_{\xi,1} \\ \dot{\mathbf{e}}_{\xi,2} = \mathbf{q} \otimes \frac{F_u}{m} \otimes \mathbf{q}^* + F_{ext} - \left(\frac{k_1 \dot{\mathbf{e}}_{\xi,1}}{a} + \ddot{\xi}_{target} \right) \end{cases}$$

$$V(\mathbf{e}_{\xi,1}, \mathbf{e}_{\xi,2}) = \frac{\mathbf{e}_{\xi,1}^T \mathbf{e}_{\xi,1}}{2} + \frac{\mathbf{e}_{\xi,2}^T \mathbf{e}_{\xi,2}}{2}$$

$$\dot{V}(\mathbf{e}_{\xi,1}, \mathbf{e}_{\xi,2}) = -k_1 \mathbf{e}_{\xi,1}^T \mathbf{e}_{\xi,1} + \mathbf{e}_{\xi,2}^T.[-\mathbf{a}.\mathbf{e}_{\xi,1} + \mathbf{q} \otimes \frac{F_u}{m} \otimes \mathbf{q}^* + F_{ext} - \left(\frac{k_1 \dot{\mathbf{e}}_{\xi,1}}{a} + \ddot{\xi}_{target} \right)]$$

Control input

$$F_u = \mathbf{q}^{-1} \otimes [m.(\mathbf{a}.\mathbf{e}_{\xi,1} - F_{ext} + \left(\frac{k_1 \dot{\mathbf{e}}_{\xi,1}}{a} + \ddot{\xi}_{target} \right) - k_2 \mathbf{e}_{\xi,2})] \otimes \mathbf{q}^{*-1}$$

Backstepping Control (cont'd)

Rotational Part

We propose: $V_1(\mathbf{q}_{\bar{e}}) = \frac{1}{2}\mathbf{q}_e^T\mathbf{q}_e = \frac{1}{2}\mathbf{q}_e^T M(\mathbf{q}_e)\Omega$

$$\Omega^v = -2k_3 M^T\mathbf{q}_e$$

$$\dot{\Omega}^v = -2k_3(M^T(\dot{\mathbf{q}}_{\bar{e}})\mathbf{q}_e + M^T\dot{\mathbf{q}}_e)$$

we define the error: $\alpha_2 = \Omega - \Omega^v$

we write the new system

$$\begin{cases} \dot{\mathbf{q}}_e = \frac{1}{2}M(\mathbf{q}_{\bar{e}})(\Omega^v + \alpha_2) \\ I\dot{\alpha}_2 = -\Omega \times I\Omega + \tau - I\dot{\Omega}^v \end{cases}$$

We propose

$$V(\mathbf{q}_e, \alpha_2) = \frac{1}{2}\mathbf{q}_e^T\mathbf{q}_e + \frac{1}{2}\alpha_2^T I\alpha_2 = \frac{1}{2}\mathbf{q}_e^T M(\mathbf{q}_{\bar{e}})(\Omega^v + \alpha_2) + \alpha_2^T(-\Omega \times I\Omega + \tau - I\dot{\Omega}^v)$$

control input is $\tau = \Omega \times I\Omega + I\dot{\Omega}^v - k_4\alpha_2 - \frac{1}{2}M(\mathbf{q}_{\bar{e}})\mathbf{q}_e^T$

Backstepping Control (cont'd)

New Equivalent System

$$\dot{\eta} = \begin{bmatrix} \dot{\mathbf{e}}_{\xi,1} \\ \dot{\mathbf{e}}_{\xi,2} \\ \dot{\mathbf{q}}_e \\ \dot{\alpha}_2 \end{bmatrix} = \begin{bmatrix} -\mathbf{a}.\mathbf{e}_{\xi,2} - k_1\mathbf{e}_{\xi,1} \\ \mathbf{q} \otimes \frac{F_u}{m} \otimes \mathbf{q}^* + F_{ext} - \left(\frac{k_1\dot{\mathbf{e}}_{\xi,1}}{a} + \ddot{\xi}_{target} \right) \\ \frac{1}{2}M(\mathbf{q}_e)(\Omega^v + \alpha_2) \\ I^{-1}(-\Omega \times I\Omega + \tau) - \dot{\Omega}^v \end{bmatrix}$$

Backstepping Control (cont'd)

Final Check on the Stability Analysis
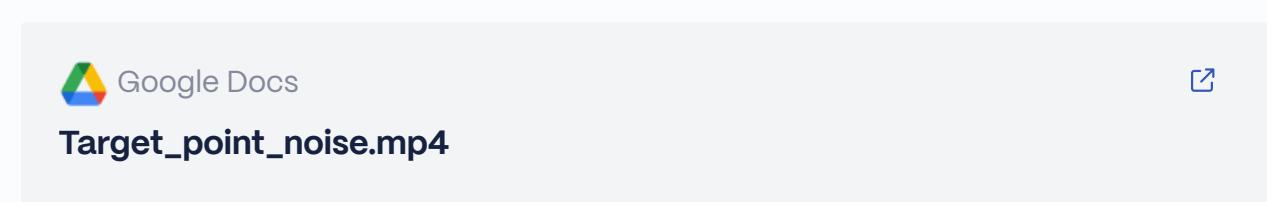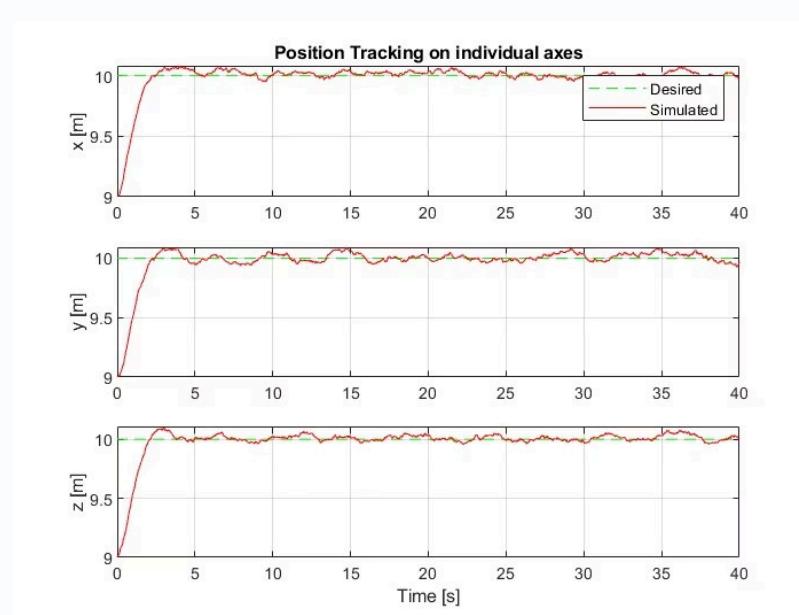
$$\ddot{\xi} \longrightarrow \mathbf{e}_\xi(\xi, m) \longrightarrow \dot{\xi}^v \longrightarrow F_u(u, \mathbf{q}_d)$$

$$\mathbf{q} \longrightarrow \mathbf{q}_e(\mathbf{q}, \mathbf{q}_d) \longrightarrow \Omega^v \longrightarrow \tau$$
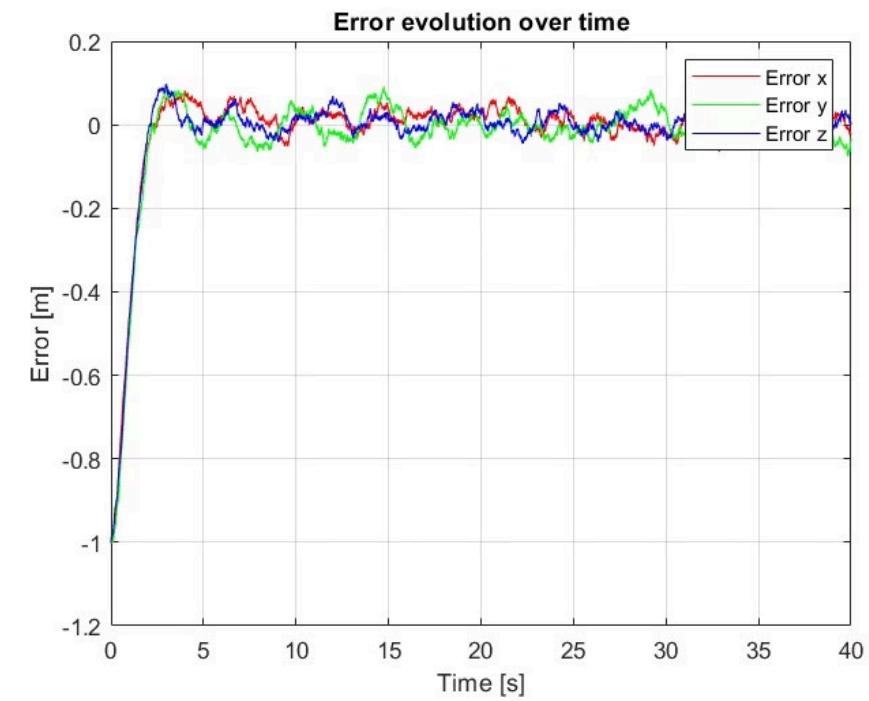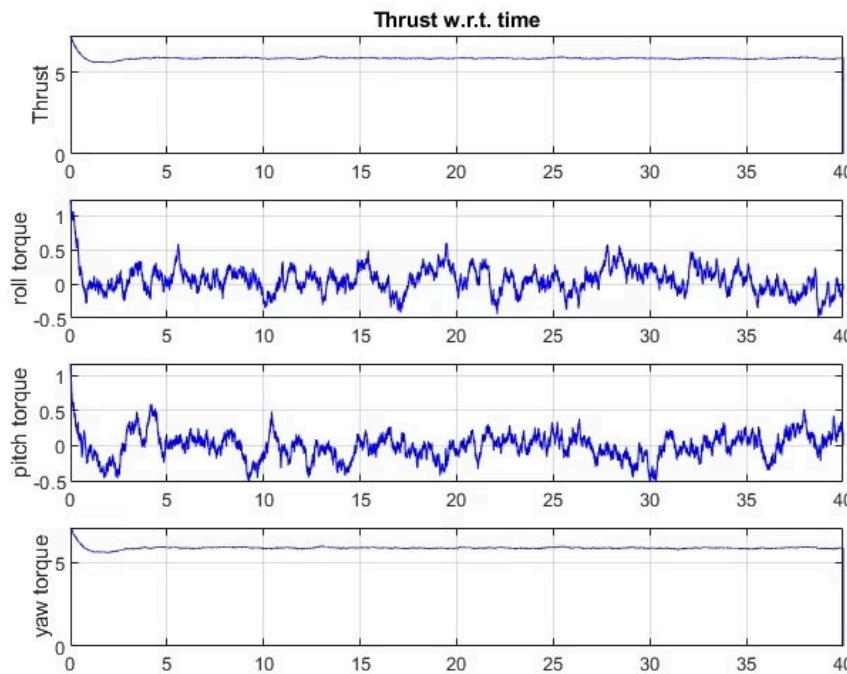
Numerical Simulations

- Constant reference point

- Ascending spiral trajectory

- Lemniscate trajectory in 3D

- Cylinder trajectory

Converting to a fixed desired point

Position Tracking on individual axes

Google Docs

Target_point_noise.mp4

Converging to a fixed desired point (cont'd)
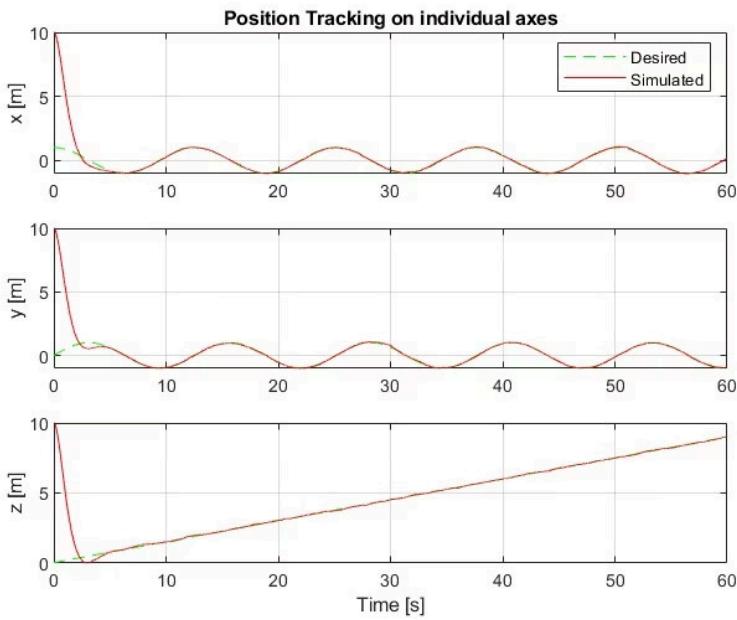
3D Trajectories

Cylindrical Trajectory

$$\mathbf{x}_d(t) = \begin{bmatrix} R\cos(\omega t) \\ R\sin(\omega t) \\ z_0 + v_z t \end{bmatrix}, \qquad t \in [0, T]$$

Cylindrical Trajectory (cont'd)

Position Tracking on individual axes
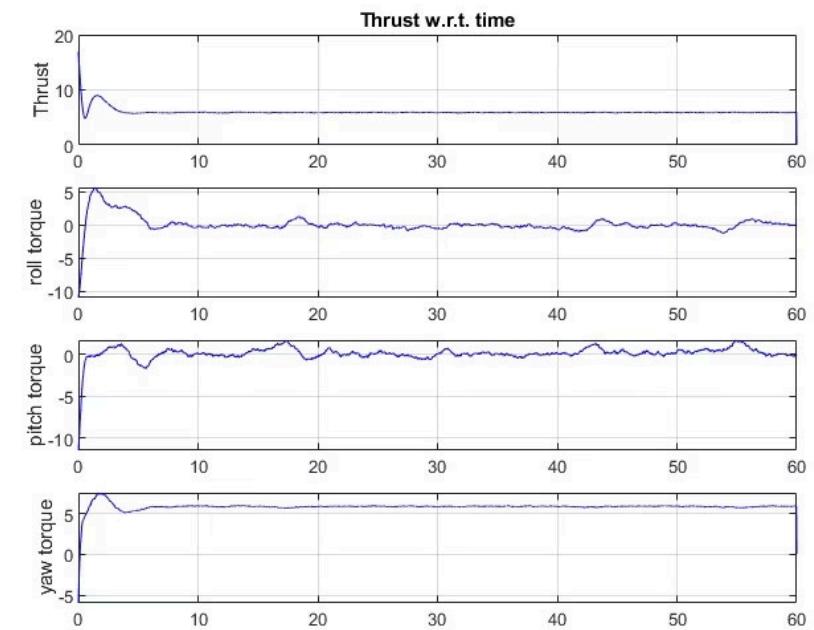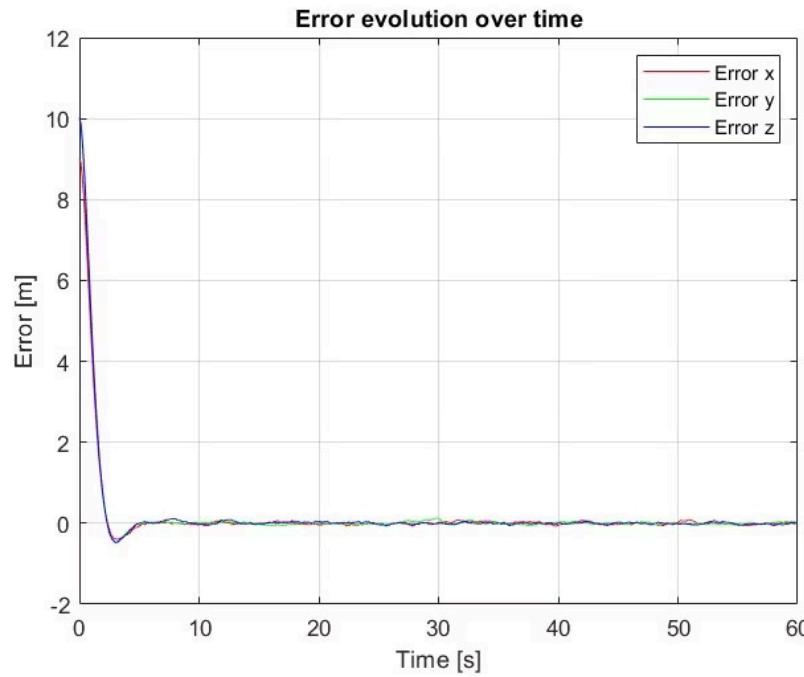
Google Docs

Quadcopter_Animation.mp4
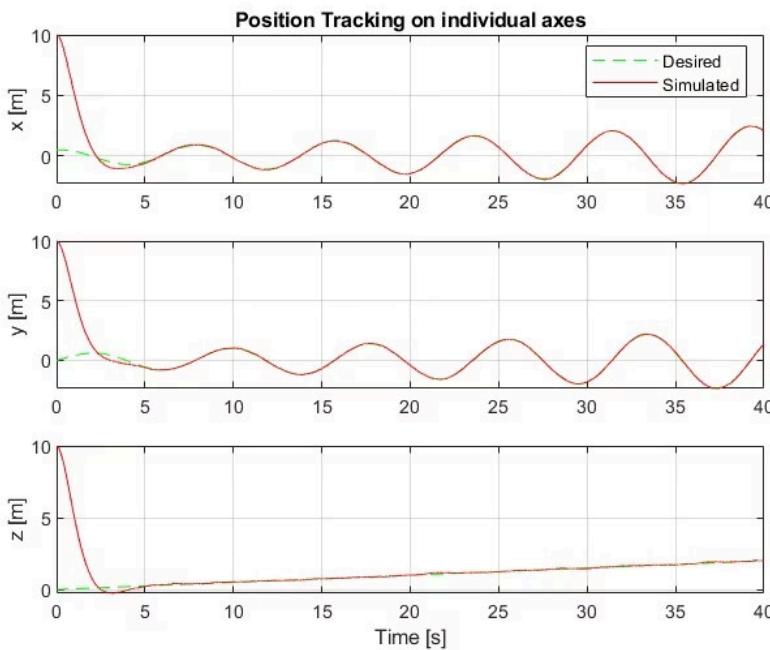
Cylindrical Trajectory (cont'd)

Spiral Trajectory
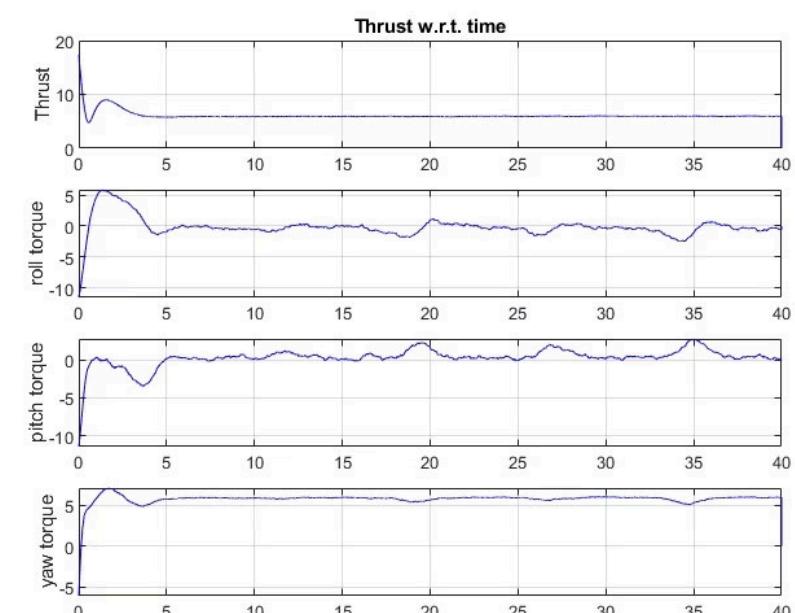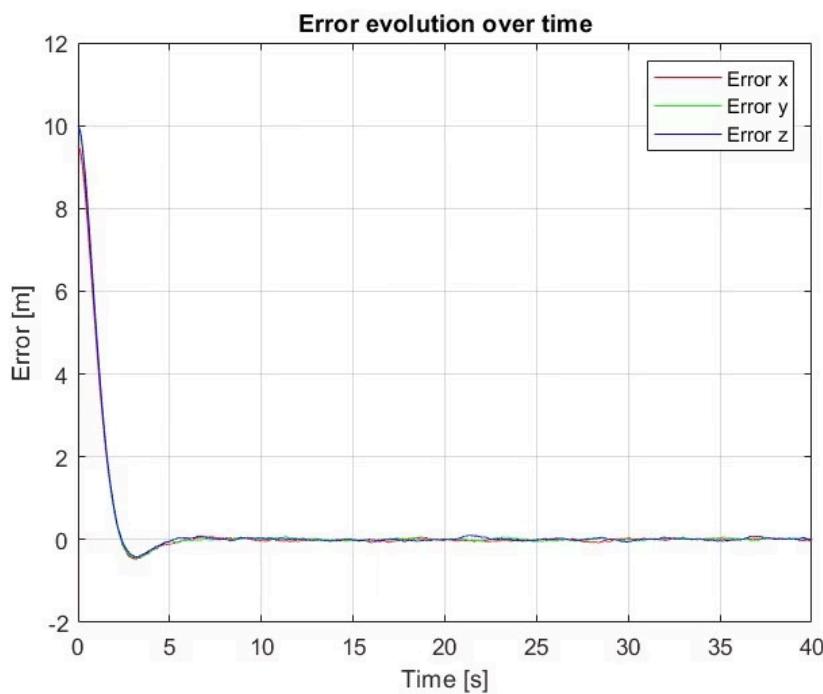
$$\mathbf{x}_d(t) = \begin{bmatrix} (R_0 + v_r\, t)\cos(\omega t) \\ (R_0 + v_r\, t)\sin(\omega t) \\ z_0 + v_z\, t \end{bmatrix}, \qquad t \in [0, T]$$

Spiral Trajectory (cont'd)


Google Docs

Quadcopter_Animation.mp4
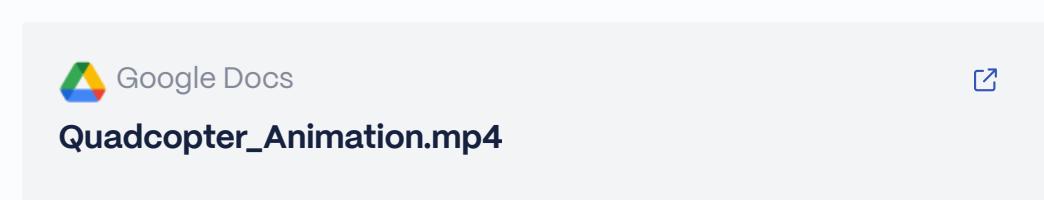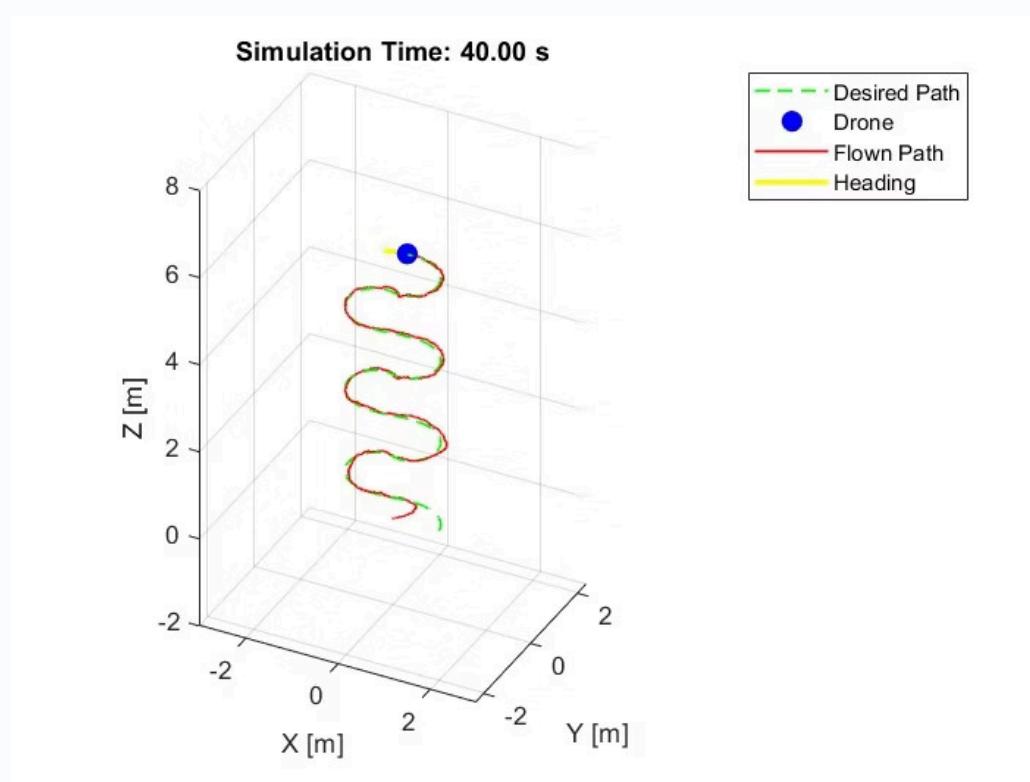
Spiral Trajectory (cont'd)
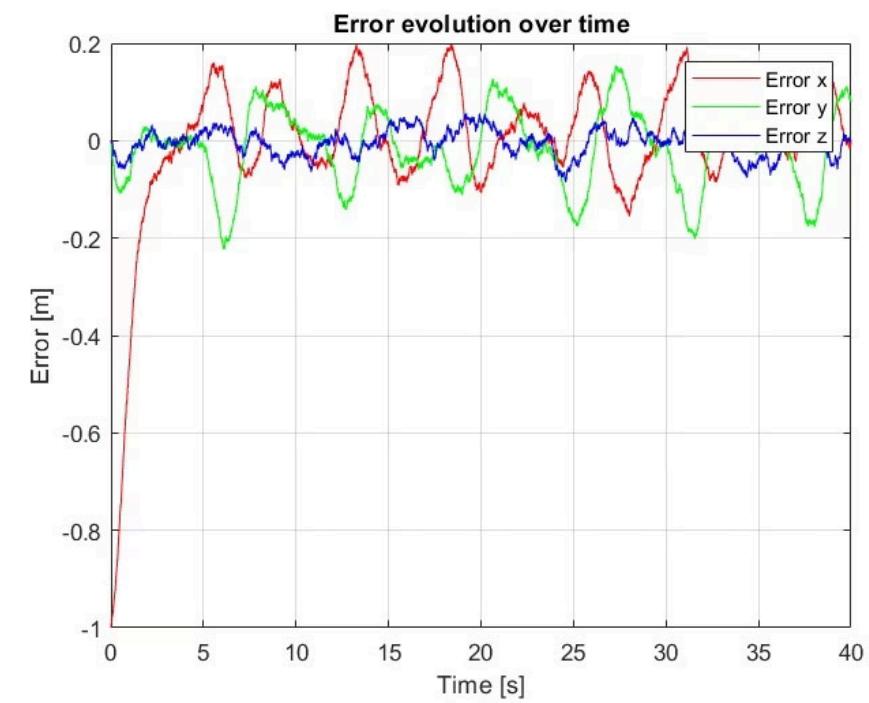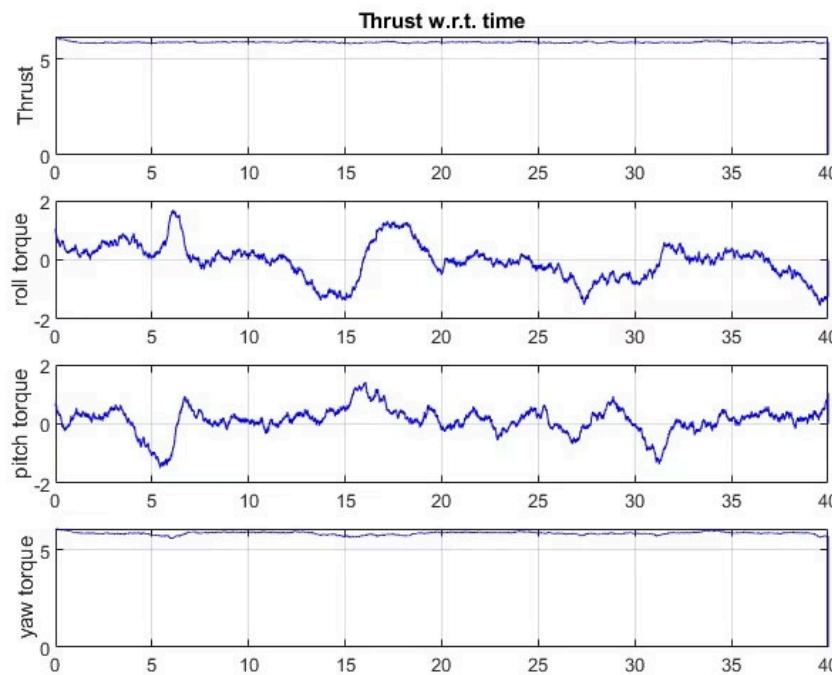
Lemniscate Trajectory

$$\mathbf{x}_d(t) = \begin{bmatrix} \dfrac{A\cos(\omega t)}{1 + \sin^2(\omega t)} \\ \dfrac{A\sin(\omega t)\cos(\omega t)}{1 + \sin^2(\omega t)} \\ z_0 + v_z t \end{bmatrix}, \qquad t \in [0, T]$$

Lemniscate Trajectory (cont'd)

Google Docs

Quadcopter_Animation.mp4

Lemniscate Trajectory (cont'd)

Multi Robot Coordination
case of 3 robots

Google Docs

Platooning_Animation.mp4

Discussion

- The results align well with the stability study conducted in the control-theory part.

- The simulation results validate our knowledge about the backstepping control and quadcopter modeling using the quaternion method.

Future Work & Enhancements

- Sensor-Based Control

- Fl-air and ROS implementation

- Hardware choice and integration

- Comparison of backstepping control with other control methods (NLMPC,AI, etc.)

eliascharbelsalameh/
Quadcopter_Backsteppi...

2
Contributors

0
Issues

0
Stars

0
Forks

GitHub

GitHub – eliascharbelsalameh/Quadcopter_Backstepping_Control

Contribute to eliascharbelsalameh/Quadcopter_Backstepping_Control development by creating an account on GitHub.

Thank You

Google Docs
ARS5_Project.pdf