Person following with obstacle avoidance

Elective in Robotics – A.A. 2025-2026

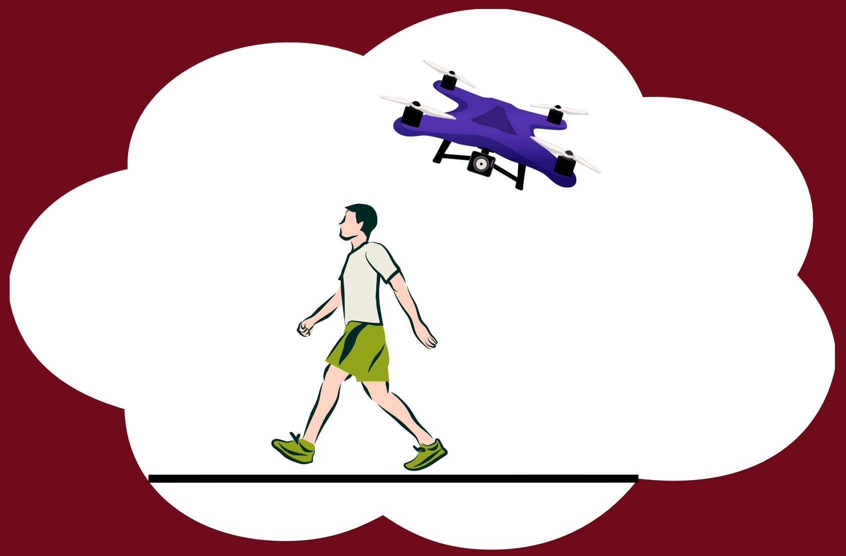
Francesco Barbetta
Giorgio De Santis
Roberto Passante
Marco Antonio Iossa

SAPIENZA
UNIVERSITÀ DI ROMA

PROBLEM STATEMENT & REQUIREMENTS

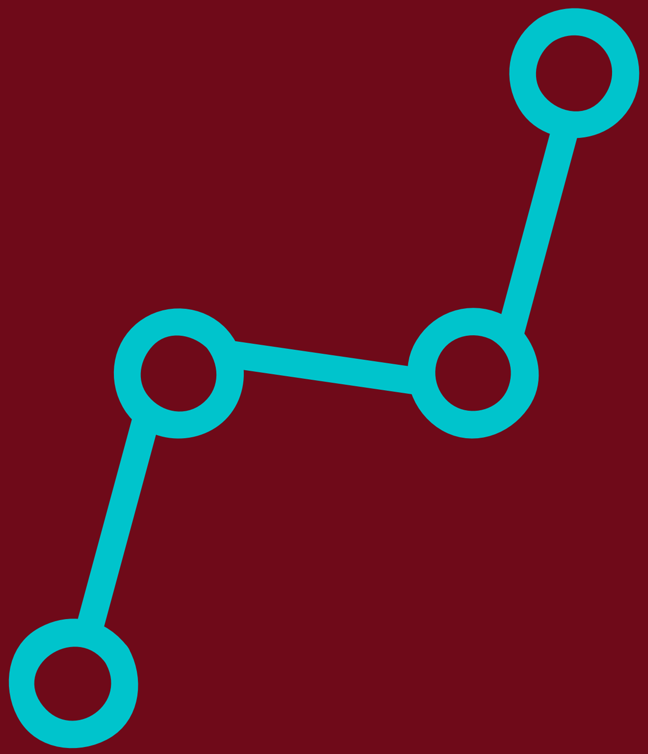
Problem

- Goal: track a moving person (Bill) in 3D while keeping a desired relative pose (distance + altitude + yaw alignment)
- Context: simulated quadrotor in a cluttered environment (static obstacles)



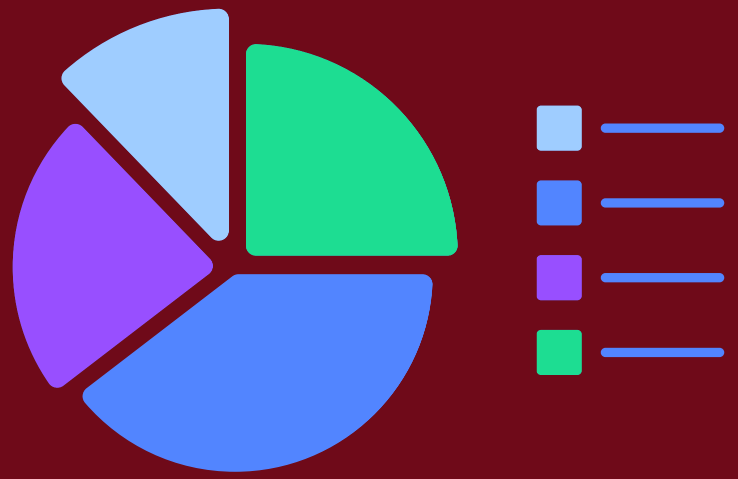
Key requirements

- Safety: maintain separation from obstacles (no collisions)
- Tracking quality: bounded relative-position error (behind/offset)
- Stability: avoid oscillations / chattering when avoidance activates
- Real-time & reactive: no global map; decisions from onboard sensing



Measurements

- Mean / max tracking distance error
- Minimum obstacle distance (and time below threshold)
- Control smoothness (optional): peaks / oscillations in roll–pitch commands



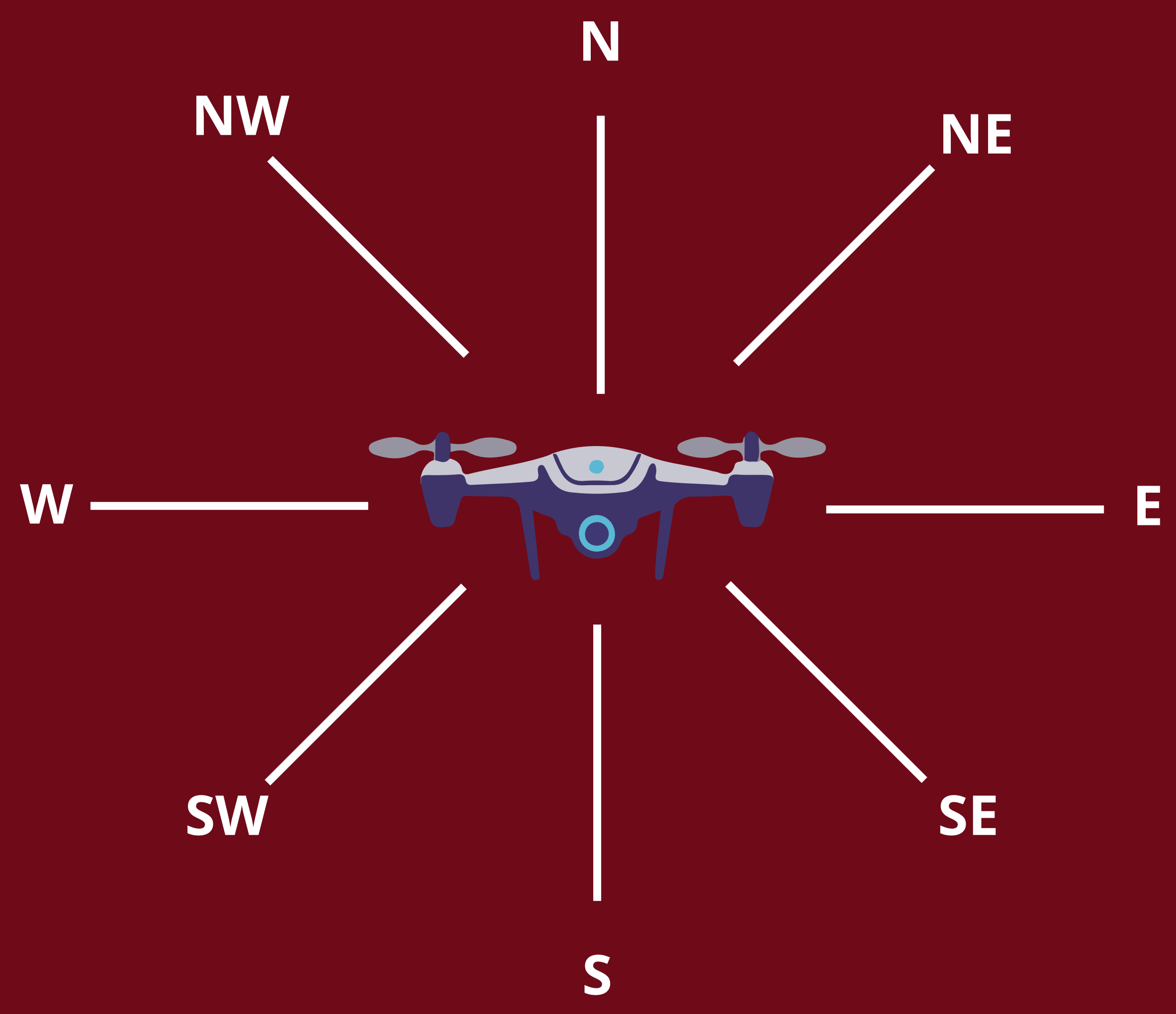
SAPIENZA
UNIVERSITÀ DI ROMA

SCENARIO & SIMULATION SETUP

PERCEPTIONS

Environment (CoppeliaSim)

- **Quadrotor in 3D workspace with static obstacles**
- **Moving target: Bill (planned trajectory)**



Logging / Data

- **Drone pose + target pose**
- **Sensor detections + avoidance activation**
- **Tracking error + min obstacle distance**

SYSTEM MODEL

State and Frames

- Position: $\mathbf{p} = [x,\ y,\ z]^T$
- Velocity: $\dot{\mathbf{p}} = [\dot{x},\ \dot{y},\ \dot{z}]^T$
- Attitude: Euler angles $\boldsymbol{\eta} = [\phi,\ \theta,\ \psi]^T + \boldsymbol{\omega}$

Inputs (control)

- Collective Thrust
- Body torques $\boldsymbol{\tau} = [\,\tau_\phi,\ \tau_\theta,\ \tau_\psi\,]^T$

$$\mathbf{F}_T = T\,\mathbf{R}(\phi,\theta,\psi)\,\mathbf{e}_3, \qquad \mathbf{e}_3 = [0,\ 0,\ 1]$$
$$m\,\ddot{\mathbf{p}} = \mathbf{F}_T + m\,\mathbf{g}, \qquad \mathbf{g} = [0,\ 0,\ -g]$$

Key idea

- Horizontal motion obtained by tilting thrust (roll/pitch coupling)
- Vertical motion depends on thrust magnitude vs gravity

SAPIENZA
UNIVERSITÀ DI ROMA

ACTUATION: THRUST GENERATION & MOTOR MIXING

Rotor thrust model

- Each rotor generates an upward thrust proportional to the square of its angular speed
- Total thrust is the sum of the four rotor thrusts

$$T_i = k_f\,\omega_i^2 \qquad (i = 1, 2, 3, 4)$$

$$T = \sum_{i=1}^{4} T_i$$

Control torques from differential thrust

- Roll and pitch torques are produced by unbalanced thrust across opposite rotors
- Yaw torque comes from reaction torques (alternating rotor spin directions)

$$\boldsymbol{\tau} = [\,\tau_\phi,\ \tau_\theta,\ \tau_\psi\,]$$

$$\tau_\phi \approx l\,(T_2 - T_4), \qquad \tau_\theta \approx l\,(T_3 - T_1)$$

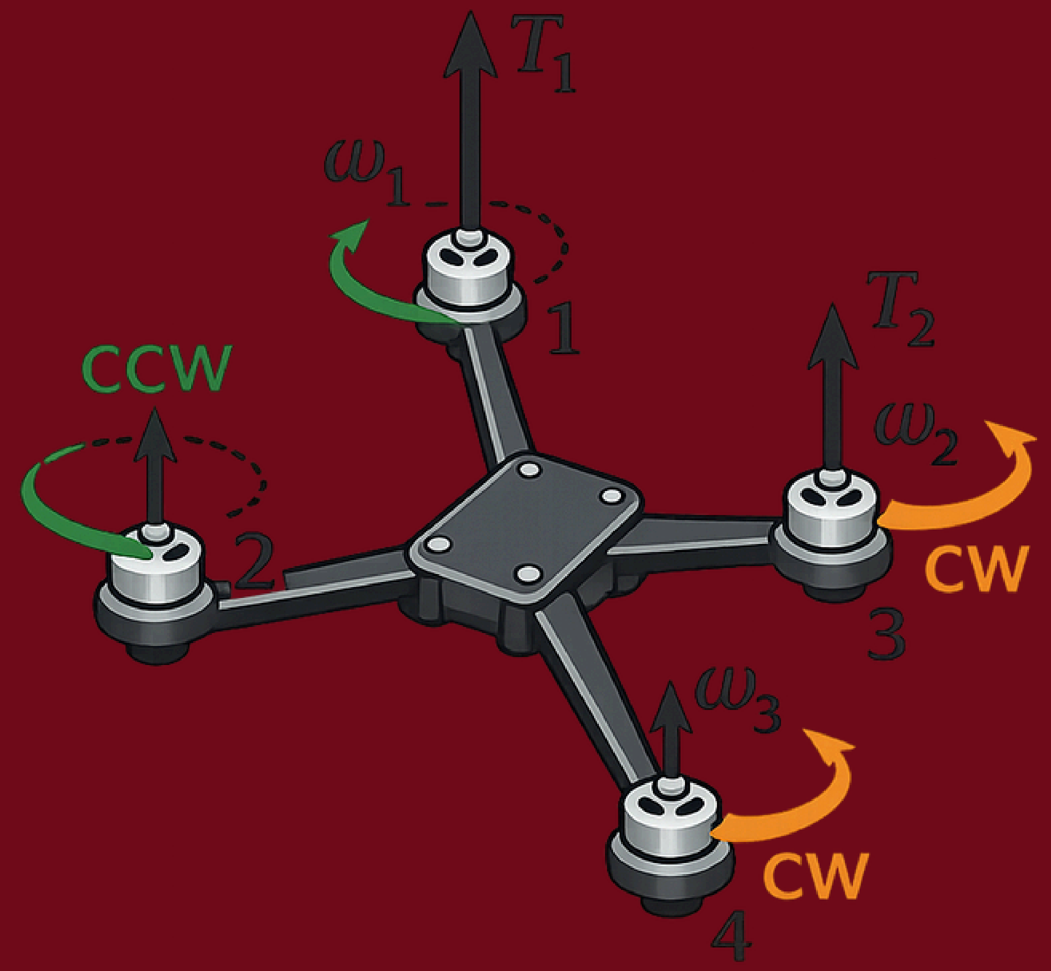
$$\tau_\psi \approx k_m(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)$$

Mixing

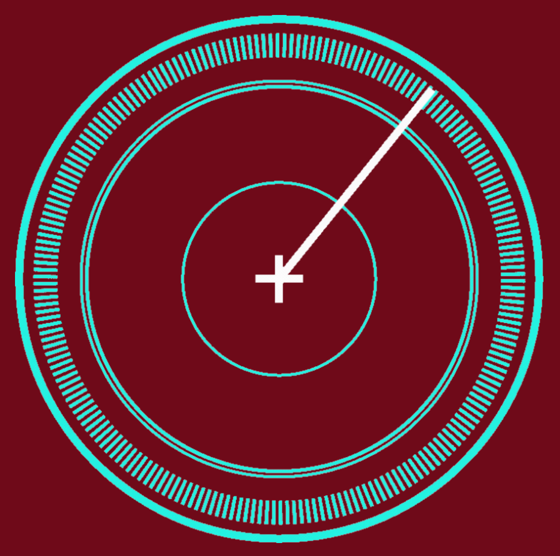
- High-level controller outputs: T and attitude corrections (roll/pitch/yaw)
- A mixing matrix maps $(T, \tau_\phi, \tau_\theta, \tau_\psi)$ to the four motor thrusts

$$[\,T,\ \tau_\phi,\ \tau_\theta,\ \tau_\psi\,] = [\,T_1,\ T_2,\ T_3,\ T_4\,]\,\mathbf{A}^\top$$

$$[\,T_1,\ T_2,\ T_3,\ T_4\,] = k_f[\,\omega_1^2,\ \omega_2^2,\ \omega_3^2,\ \omega_4^2\,]$$



SENSORS

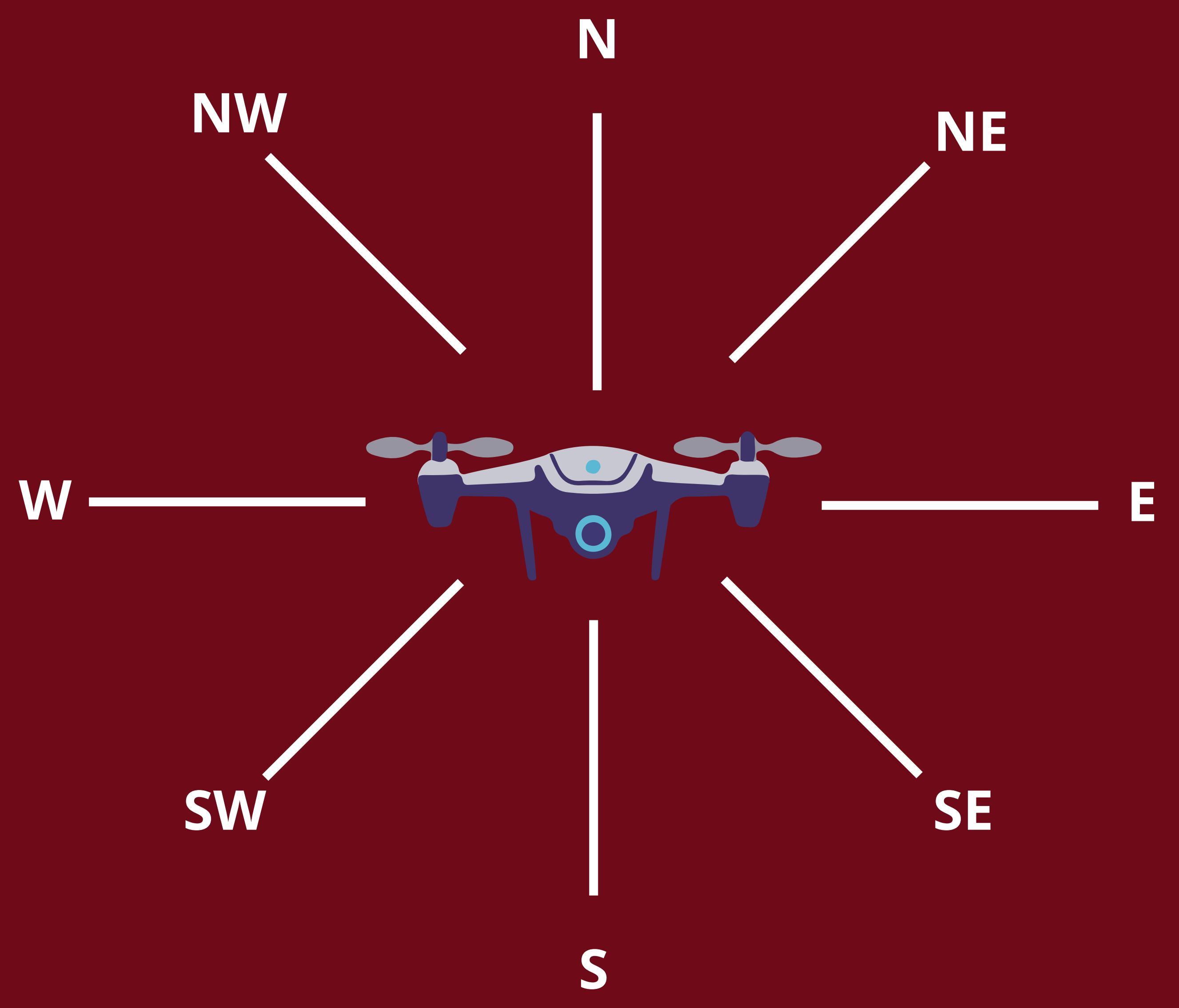


8 proximity sensors (N, S, E, W, diagonals)
Each sensor detects the closest obstacle point along its ray

Detected point is expressed in world frame
Self-filtering: ignore detections belonging to the UAV body

Use the minimum distance among all detections:

$$d_{\min} = \min_{s \in \{N,S,E,W,NE,NW,SE,SW\}} d_s$$



Assumptions :

- Sensors are ideal (no noise, negligible delay)
- Obstacles are static; target motion is repeatable (seeded planning)

CONTROL ARCHITECTURE OVERVIEW

The quadrotor is controlled using a hierarchical cascaded control architecture, where high-level position errors are converted into low-level attitude and thrust commands.
The overall structure is composed of an outer position loop and an inner attitude loop, with the physical dynamics fully handled by the CoppeliaSim simulator.

The control system is organized into three main components:

- Altitude control (outer loop, vertical axis)
- Horizontal position control (outer loop → attitude setpoints)
- Attitude and heading control (inner loop)

ALTITUDE CONTROL (HEIGHT CONTROL)

- Vertical motion controlled via PID controller
- Objective: maintain desired altitude
- Controller output: total thrust command

$$T = g + K_p e_z + K_i \int e_z \, dt + K_d v_z$$

- Feedforward gravity compensation ensures steady hover
- The thrust command is distributed to the four propellers through a mixer.

CASCADE CONTROLLER

- Horizontal motion is not directly controlled
- Lateral displacement is achieved indirectly through attitude modulation
- Position error is expressed in the body (local) reference frame
- We use a combination of position controller + attitude controller

e represents the position error between the drone and the target, expressed in the drone's body frame

$$e = R_{WD}^{T} \cdot ({p_t}^{W} - p_D^{W})$$

POSITION CONTROL

Using a linear small-angle approximation, a PD position controller is applied to the horizontal position error to compute the desired roll and pitch angles.

$$\theta_{des} = k_p \cdot e[2] + k_d \cdot \dot{e}\,[2]$$

$$\varphi_{des} = k_p \cdot e[1] + k_d \cdot \dot{e}\,[1]$$

Sapienza
Università di Roma

ATTITUDE CONTROL (ROLL & PITCH)

Attitude control stabilizes the drone around the desired roll and pitch angles generated by the outer position loop.

$$e_\phi = \phi_{des} - \phi \qquad e_\theta = \theta_{des} - \theta$$

The controller computes:

$$\alpha_{corr} = k_{\theta p} \cdot e_\theta + k_{\theta d} \cdot \dot{e}_\theta$$
$$\beta_{corr} = k_{\varphi p} \cdot e_\varphi + k_{\varphi d} \cdot \dot{e}_\varphi$$

HEADING CONTROL

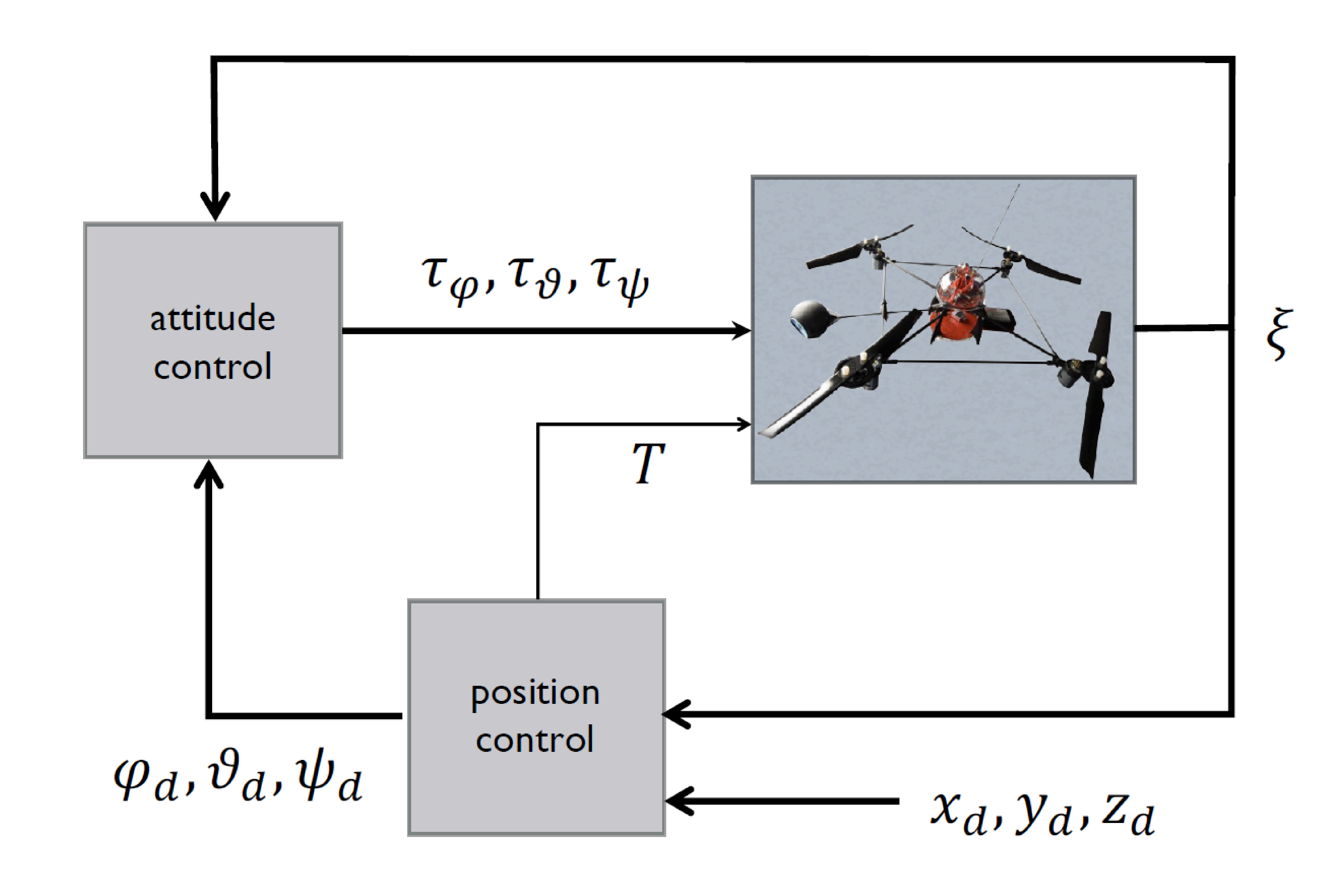
- Yaw orientation controlled via PD controller
- Objective: align UAV heading with target orientation
- Controller output: yaw correction term (rotCorr)
- Yaw error computed relative to target orientation

$$e_\psi = \psi_d - \psi$$

$$\mathrm{rotCorr} = K_p e_\psi + K_d \dot{e}_\psi$$

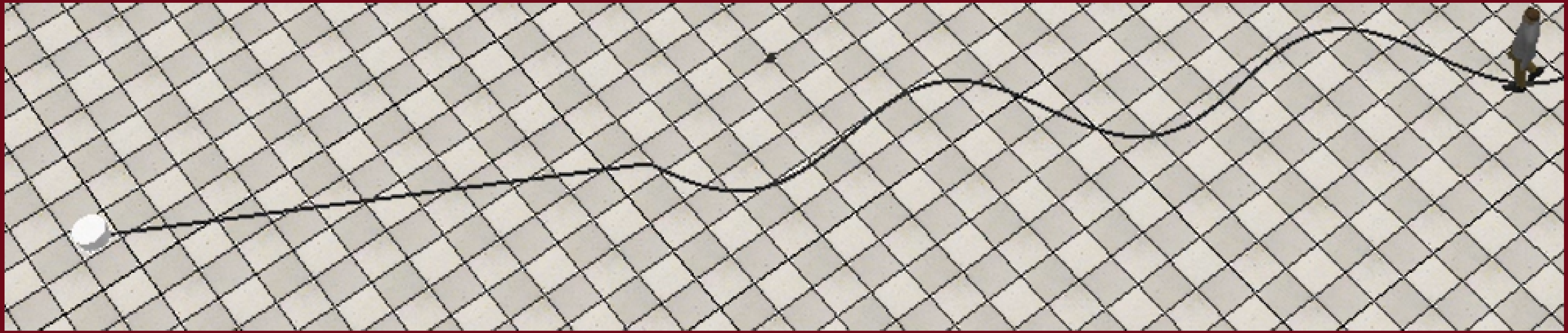
- Proportional term reduces heading error
- Derivative term provides angular damping and prevents overshoot

HIERARCHICAL CONTROL SCHEME FOR A QUADROTOR



TARGET TRAJECTORY GENERATION

- Bill’s motion is generated through a motion planning process based on the Open Motion Planning Library (OMPL).
- In particular, the RRT-Connect algorithm is used to compute a collision-free path between a start and a goal configuration
- Quadratic Bézier interpolation is used to smooth the planned path, making it fluid and realistic



TARGET KINEMATICS AND FOLLOW STRATEGY

The target's orientation is continuously updated to remain consistent with the direction of motion, aligning the yaw angle with the tangent to the path.

Rather than directly following the target position, the quadrotor tracks a virtual setpoint defined at a constant distance and height relative to the target.

```
local ex = math.cos(yaw)
local ey = math.sin(yaw)
local ex_lat = -math.sin(yaw)
local ey_lat =  math.cos(yaw)

local virtualTargetPos = {
    targetPos[1] - follow_dist * ex + lateral_offset * ex_lat,
    targetPos[2] - follow_dist * ey + lateral_offset * ey_lat,
    desiredZ
}
```

OBSTACLE AVOIDANCE — CONCEPT OVERVIEW

Obstacle avoidance is implemented as a reference modulation strategy that acts only when obstacles are detected within a predefined influence region.

Differently from classical Artificial Potential Field approaches, no attractive force toward the goal is used.

$$\cancel{F_{att}}$$

The quadrotor follows the nominal reference trajectory, while obstacle avoidance is handled exclusively through repulsive forces.

VORTEX FIELD AND HEURISTICS

Define:

- **quadrirotor position:** $p \in \mathbb{R}^3$
- **obstacle position:** $p_0 \in \mathbb{R}^3$
- **influence radius:** d_0
- **distance:** $d\left(p, p_0\right) = \|p - p_0\|$

Activation condition

$$d\left(p, p_0\right) \leqslant d_0$$

The total repulsive force is defined as:

$$F_r = F_{rt} + F_{rr}$$

$$F_r = \underbrace{F_{rt}}_{\text{translational component}} + \underbrace{F_{rr}}_{\text{rotational component}}$$

translational component

rotational component

TRANSLATIONAL REPULSIVE COMPONENT F_{rt}

$$\mathbf{F}_{rt}(p) = \begin{cases} k_{rt}\left(\frac{1}{d(p,p_o)} - \frac{1}{d_0}\right)\frac{1}{d(p,p_o)^3}(p - p_o), & d \leq d_0 \\ 0, & d > d_0 \end{cases}$$

where k_{rt} is a translational gain

The force is not null only when distance is less than influence radius.

Since the distance from the obstacle is small, a cubic factor in the equation denominatoris needed, in this way the smaller the distance between the robot and the obstacle, the greater the force will be

ROTATIONAL REPULSIVE COMPONENT F_{rr}

$$\mathbf{F}_{rt}(p) = \begin{cases} k_{rr}\left(\frac{1}{d(p,p_o)} - \frac{1}{d_0}\right)\frac{1}{d(p,p_o)^3}R(p-p_o), & d \leq d_0 \\ 0, & d > d_0 \end{cases}$$

where:

$$R = \begin{cases} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, & \theta \geq 0 \\ \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, & \theta < 0 \end{cases}$$

- rotational gain k_{rr}
- planar rotation matrix R
- $\theta = \varphi - \rho$ is the relative angle between the quadrotor velocity direction φ, and the direction toward the obstacle centroid ρ

This outcomes the limitations of purely repulsive field, preventing the obstacle simply gets back

OBSTACLE REPULSIVE FORCE

To prevent collisions, each obstacle generate a repulsive force acting on the quadrotor whenever the distance from the obstacle falls within a predefined influence region.

The obstacle-induced force acting on the quadrirotor is obtained from the combination of repulsive and vortex (orthogonal to repulsive) force

$$F_{obs} = F_{rep} + F_{vor}$$

REPULSIVE AND VORTEX COMPONENTS

The repulsive component acts along the normal direction away from the obstacle, guaranteeing collision avoidance by increasing the separation distance.

$$F_{rep} = -k_{rep} \cdot \sigma(d)^3 \cdot u_{obs}$$

$$\sigma(d) = \frac{d_0 - d}{d_0 - d_{min}}$$

$$\sigma(d) \in [0, 1]$$

SMOOTH ACTIVATION FUNCTION

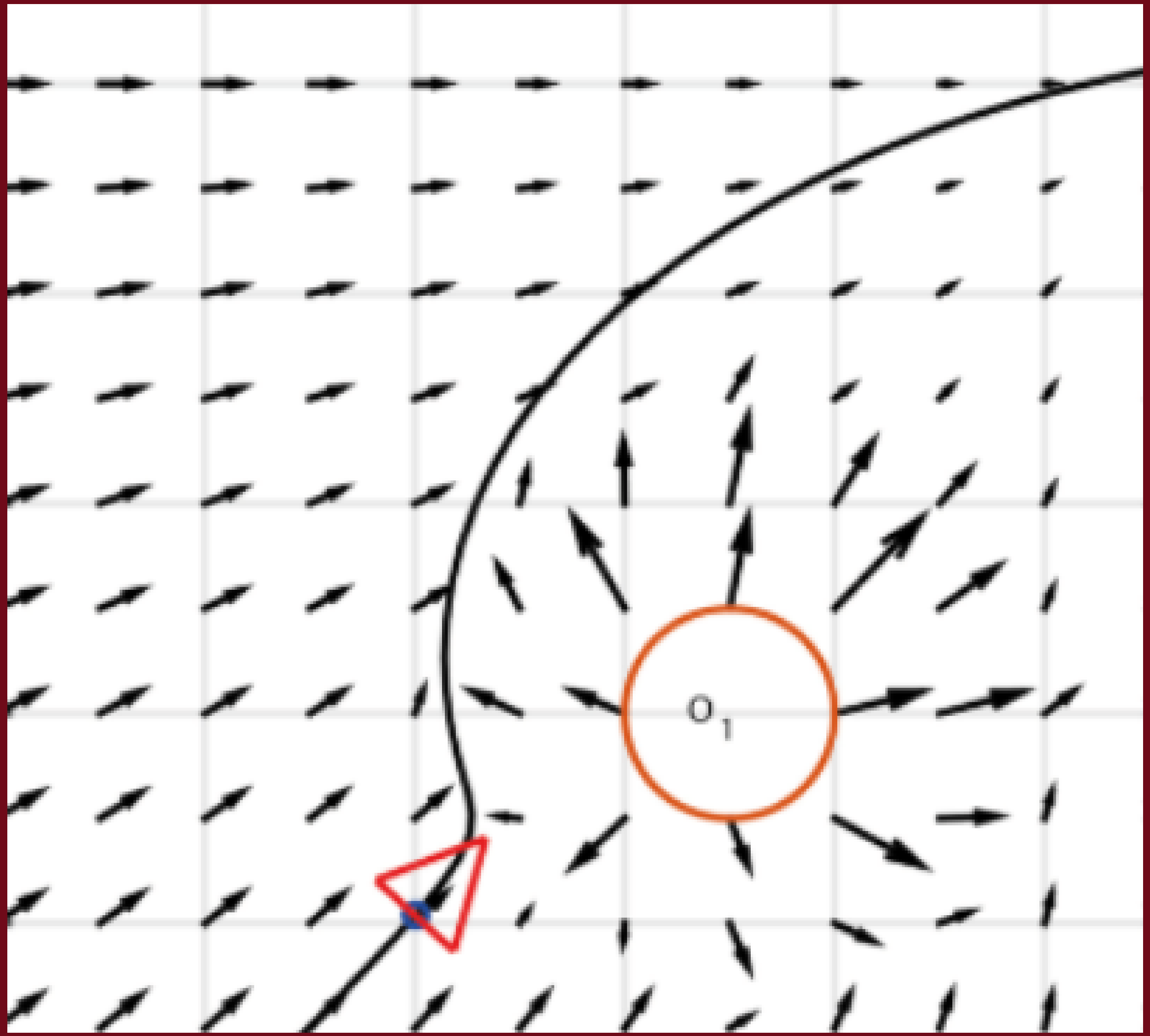
The tangential component acts orthogonally to the repulsive direction and induces a smooth circular motion around the obstacle, enabling stable obstacle circumnavigation.

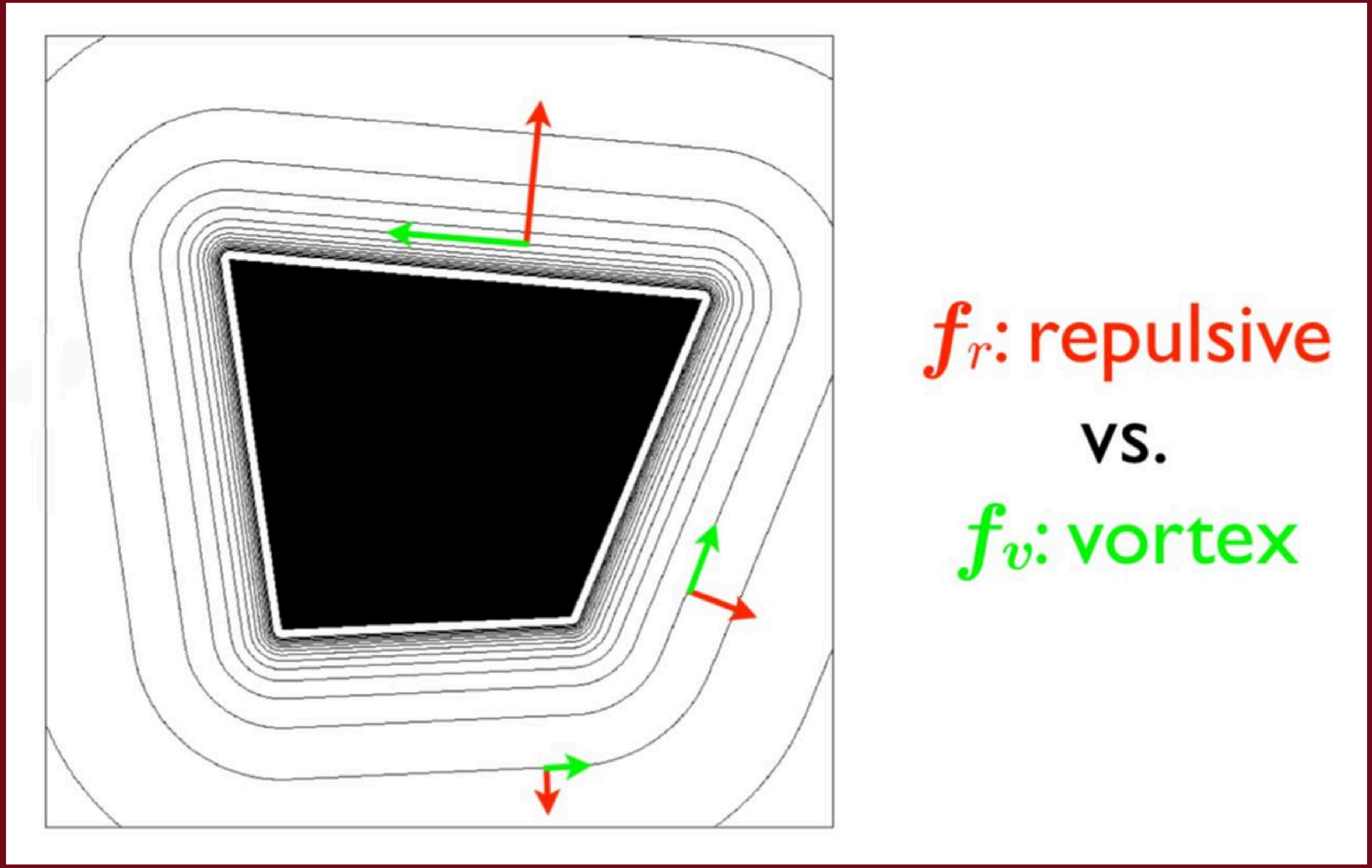
$$F_{vor} = k_{vor} \cdot \sigma(d)^3 \cdot \rho \cdot u_{tan}$$

$$\rho \in +1, -1$$

DIRECTION OF ROTATION

REPULSIVE AND VORTEX FIELD





SMOOTH ACTIVATION AND BLENDING

The vortex field is not activated abruptly, but a continuous blending mechanism is adopted, in order to prevent discontinuities in the commanded forces and significantly reduce oscillatory behaviors.

BLENDING FACTOR

$$w_{raw} = \frac{d_{off} - (d_{min} - d_{soft})}{d_{off} - d_{on}}$$

- **activation distance** d_{on}
- **deactivation distance** d_{off}
- **artificial buffer** d_{soft}
- **real minimum distance** d_{min}

BLENDING COEFFICIENT

$$w = clamp\left(w_{raw}, 0, 1\right) \qquad w \in [0, 1]$$

Apply also a cubic shaping, to further smooth the activation: $w \leftarrow w^3$

SMOOTH ACTIVATION AND BLENDING OF THE VORTEX FIELD

PROBLEM → vortex fields may induce large lateral accelerations and oscillatory motion if not properly regulated.

SOLUTIONS →

- Force smoothing
- Force saturation
- Hysteresis logic

STABILIZATION MECHANISMS

Force smoothing

Filtering of lateral forces generated by the vortex field using exponential smoothing scheme, as a discrete-time low-pass filter

$$F_k = (1-\alpha)F_{k-1} + \alpha F_k^{raw}$$

- reduction of sensor noise

Force saturation

Filtered forces limited to a maximum admissible magnitude, ensuring bounded control inputs

- prevention of excessive accelerations
- improvement of overall stability

$$F \in [-F_{max}, F_{max}]$$

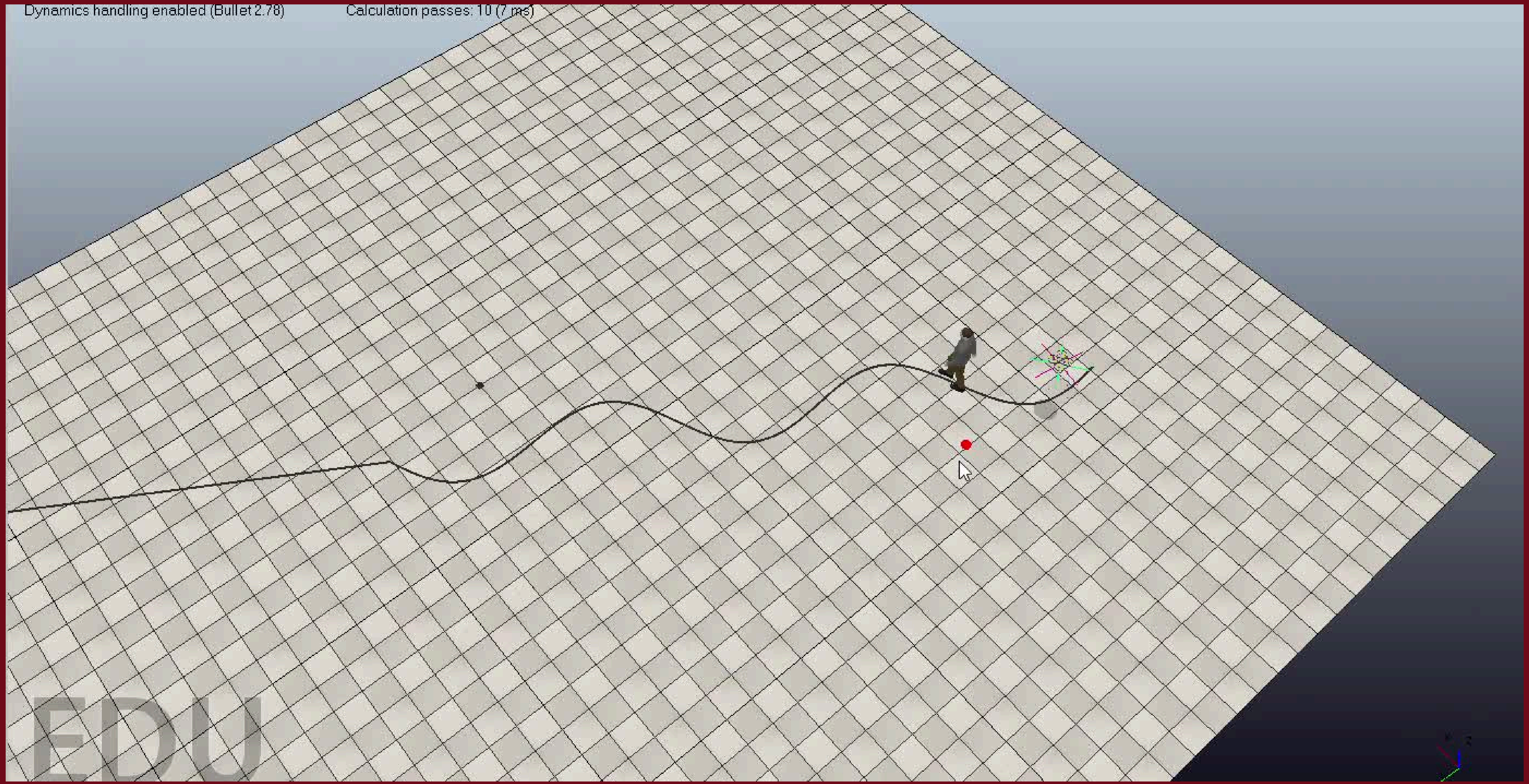
Hysteresis

Avoids frequent switching of the avoidance mode when near influence radius

Avoidance mode $\longrightarrow$ ACTIVATED when the obstacle distance falls below d_{on}
$\longrightarrow$ DEACTIVATED when the distance exceeds d_{off}

- significant reduction of chattering effects

SIMULATION RESULTS AND DISCUSSION



PATH WITH NO OBSTACLES

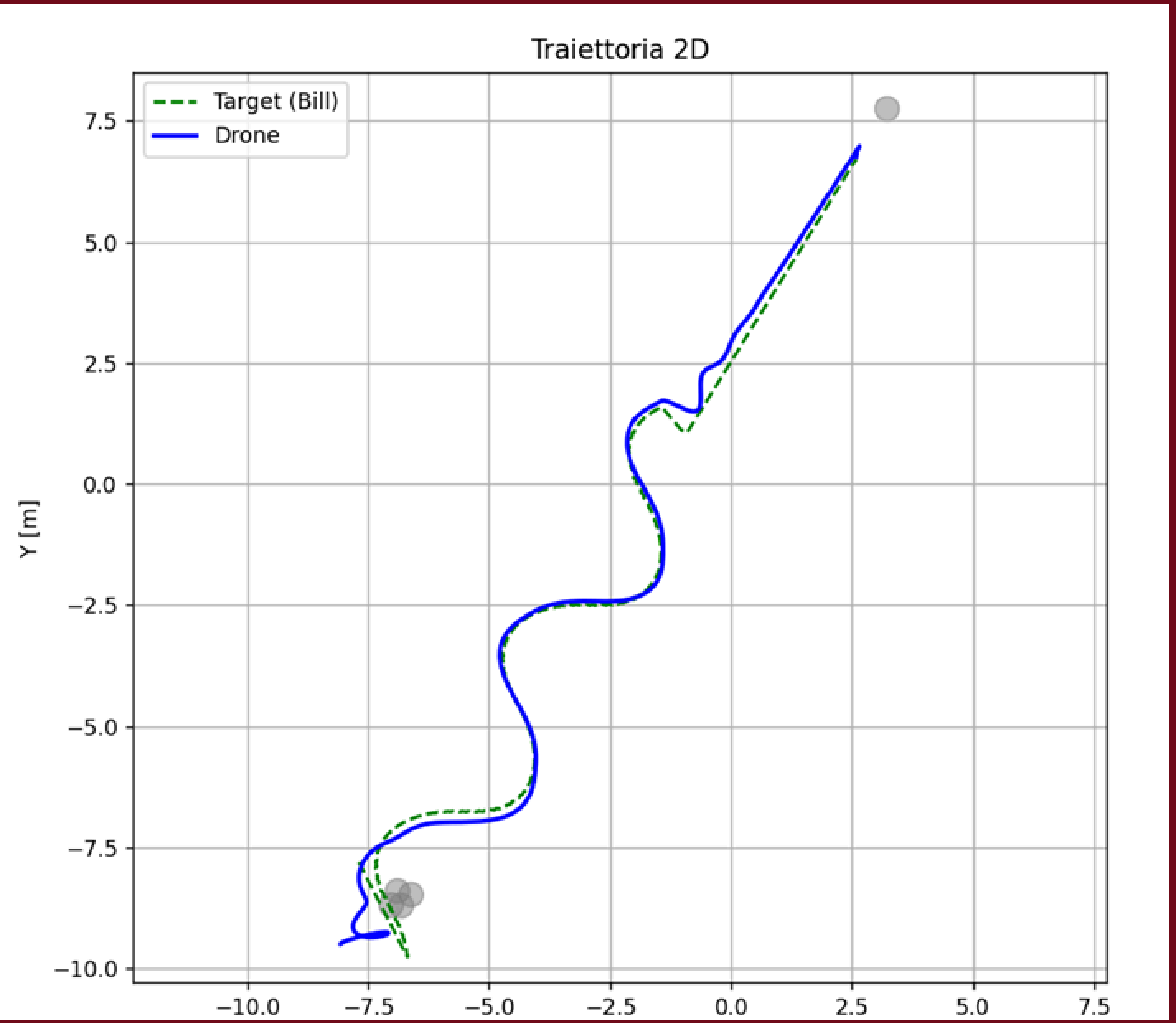
starting configuration to test the path model working

- **no conflict between Bill and the robot**
- **good working of the repulsive force**
- **self-generated path with fixed seed**

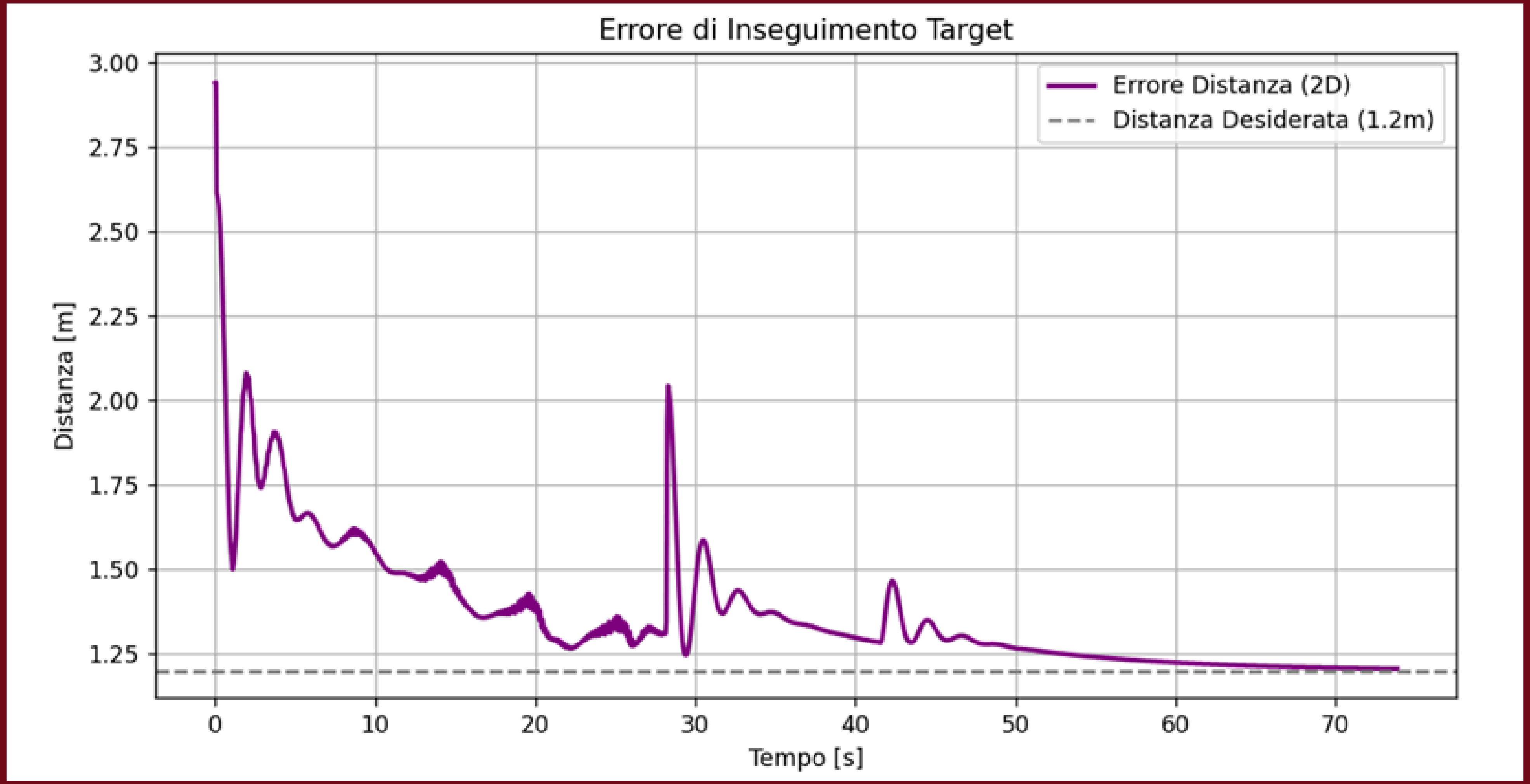
```
local fixedSeed = 5678
math.randomseed(fixedSeed)
```

SIMULATION RESULTS AND DISCUSSION

PATH WITH NO OBSTACLES



- **2D trajectory**: the UAV path (blue) almost overlaps the target path (green). After a short initial transient (due to initial offset), tracking remains very accurate.

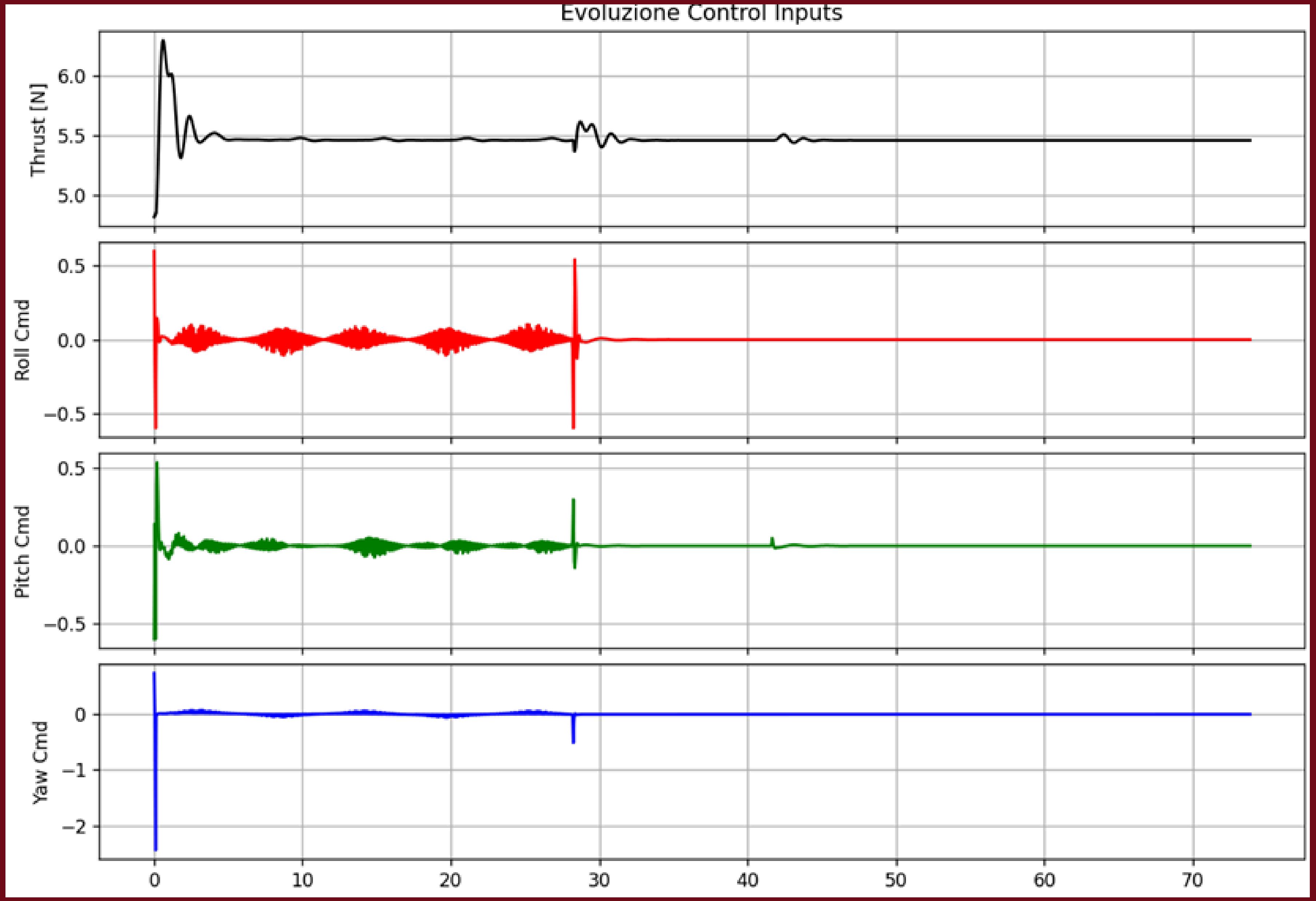


- **Distance error (2D)**: the error converges smoothly to the desired following distance (1.2 m), with only small residual oscillations mainly caused by target path curvature changes (not controller instability).

SAPIENZA
UNIVERSITÀ DI ROMA

SIMULATION RESULTS AND DISCUSSION

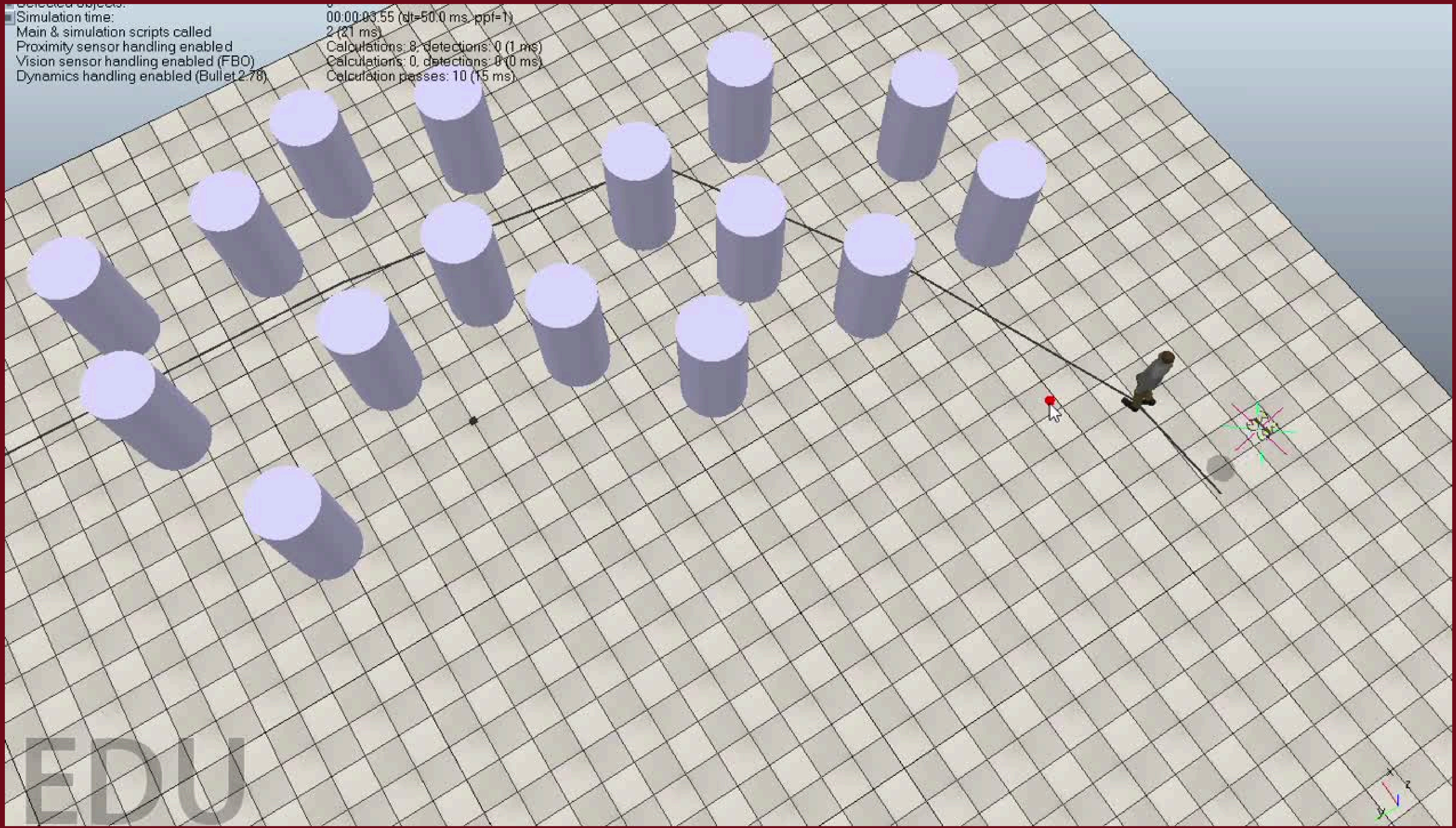
PATH WITH NO OBSTACLES



altitude control parameters
pParam = 10.0
iParam = 0.2
dParam = 0.4
horizontal control parameters
k_phi_p = 0.25
k_phi_d = 2.5
k_theta_p = 0.25
k_theta_d = 2.5
heading control parameters
k_psi_p = 0.1
k_psi_d = 1.0
position control parameters
k_pos_p = 0.01
k_pos_d = 0.5

- Thrust: Initial peak ---> Stable Hovering (Zero steady-state error).
- Roll/Pitch: Smooth & Bounded ---> Safe Dynamics (No saturation).
- Yaw: Fast convergence ---> Solid Heading (0° error).

SIMULATION RESULTS AND DISCUSSION



OBSTACLES
(NO HEURISTIC METHODS)

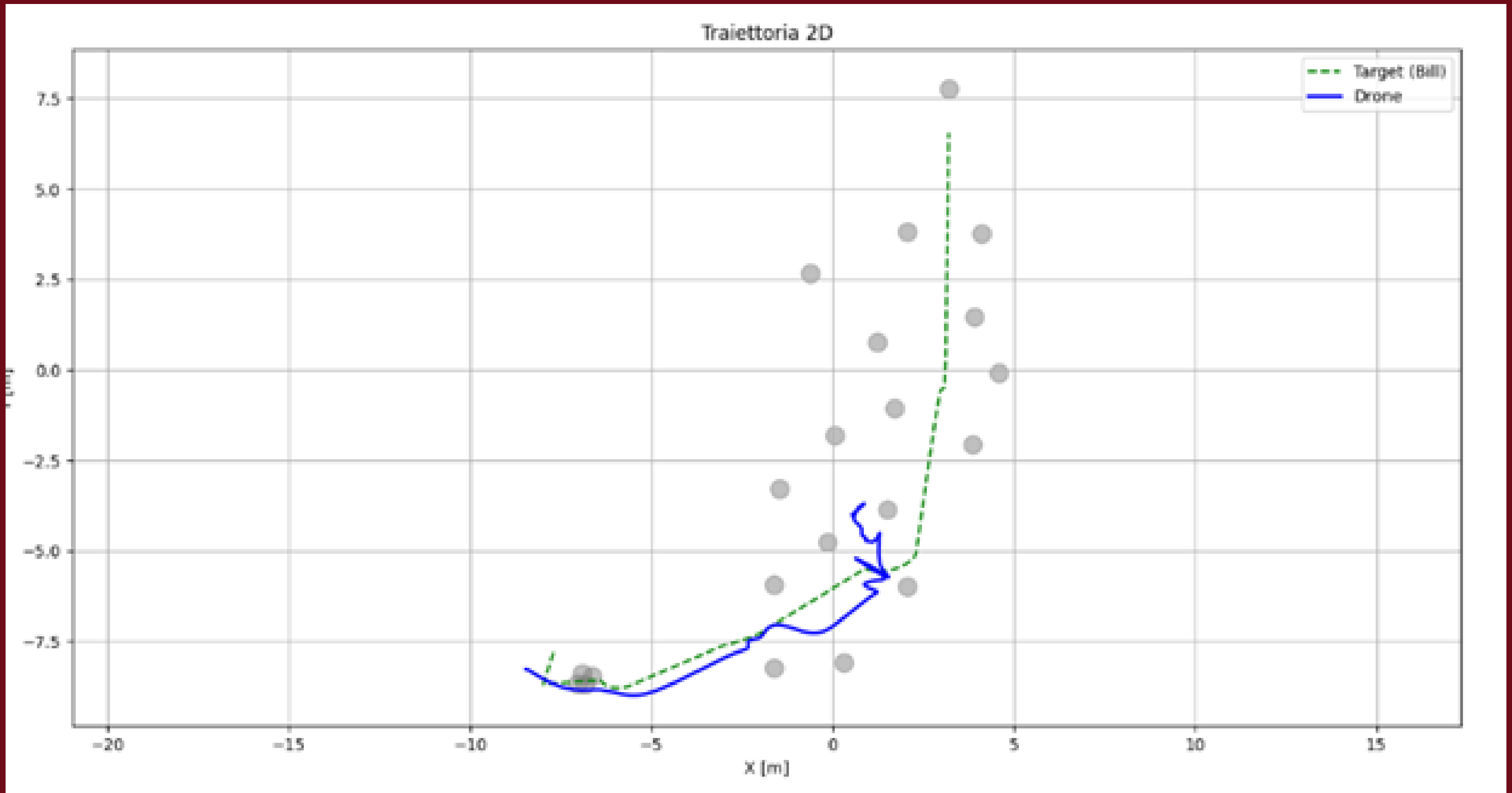
- the drone reacts only locally to repulsive forces, without any priority logic or memory
- no control on robot angles

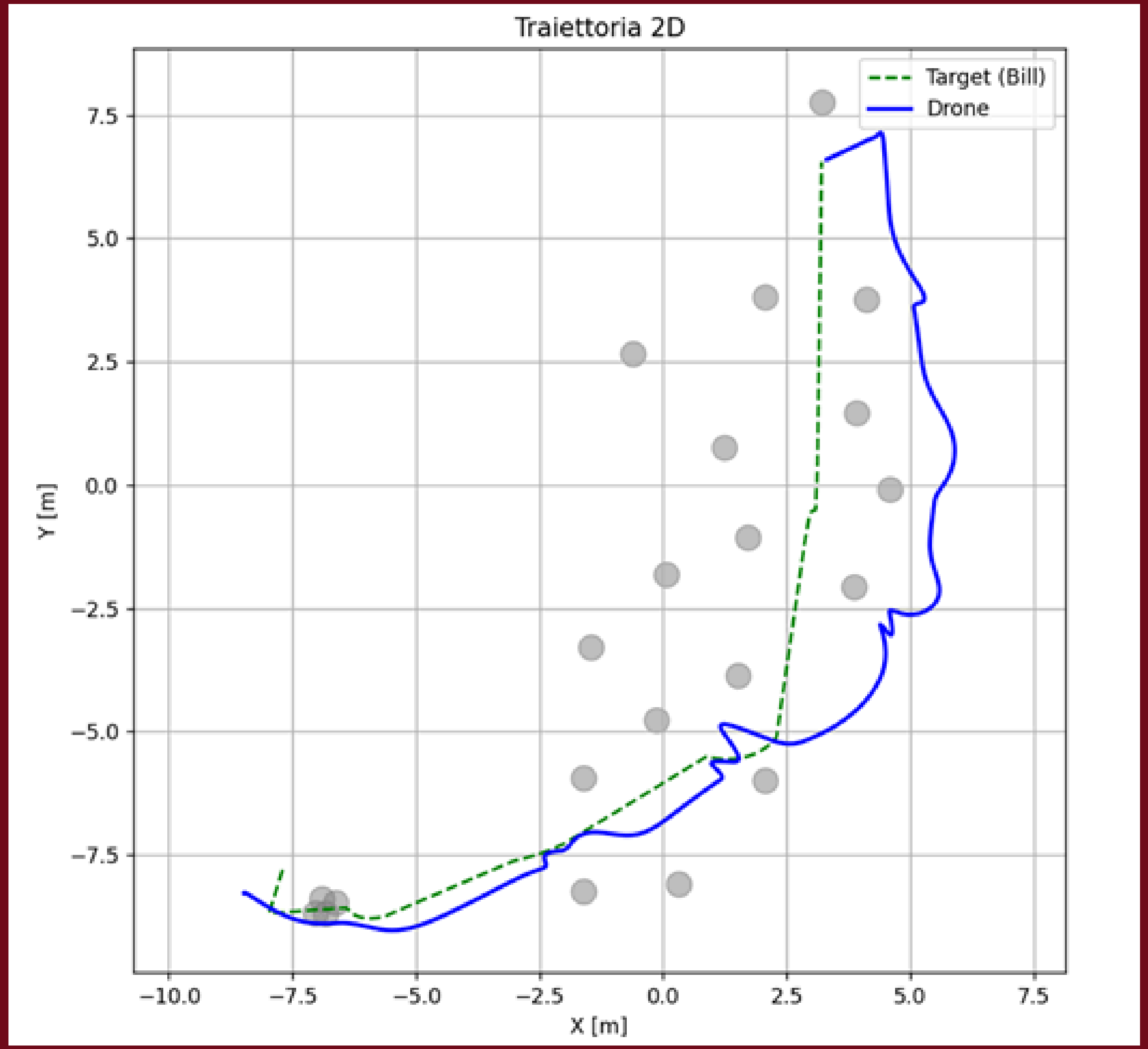
↓

unstable behavior: oscillations, hesitations, and loss of tracking

SIMULATION RESULTS AND DISCUSSION

OBSTACLES (NO HEURISTIC METHODS)



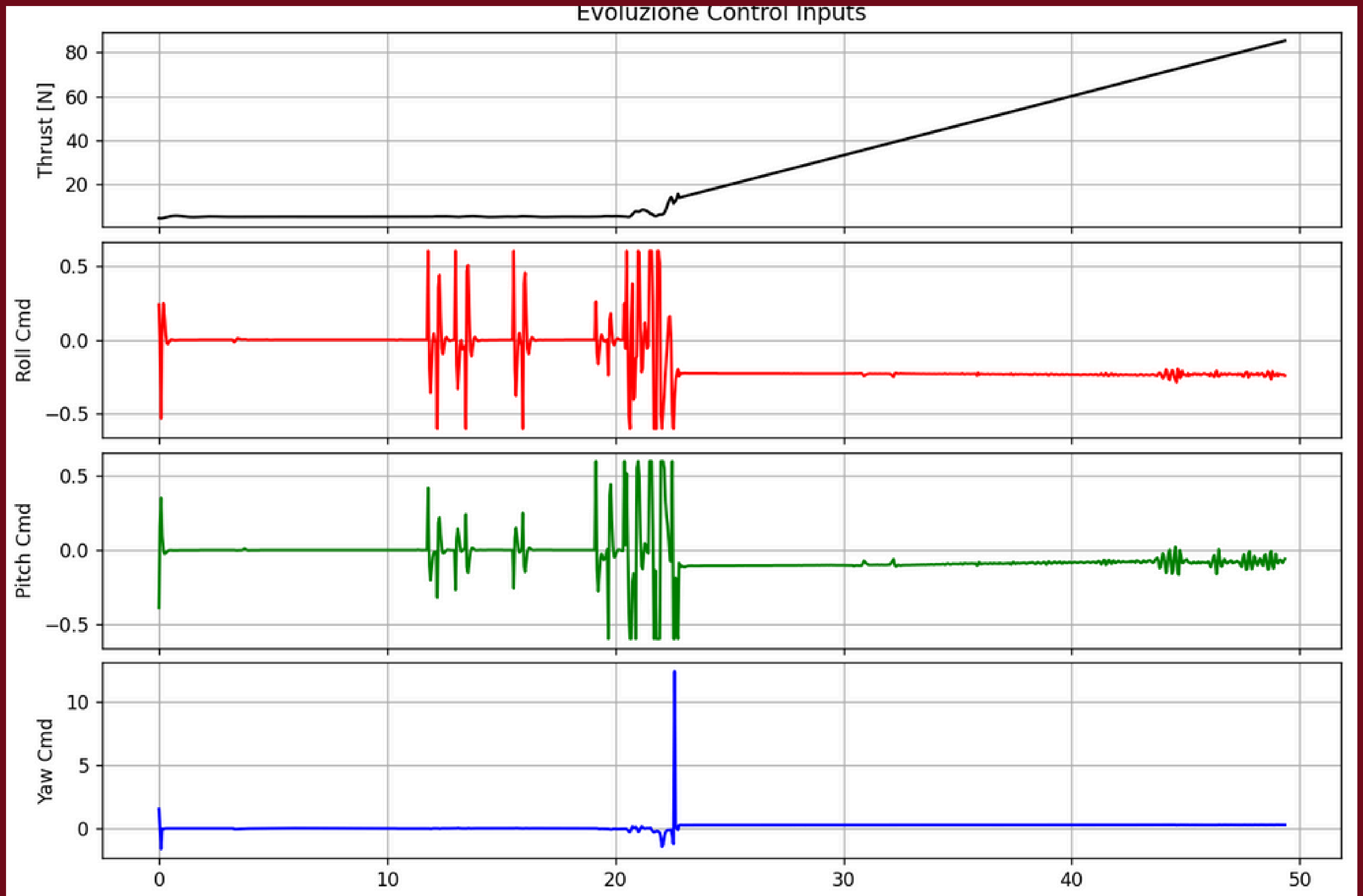


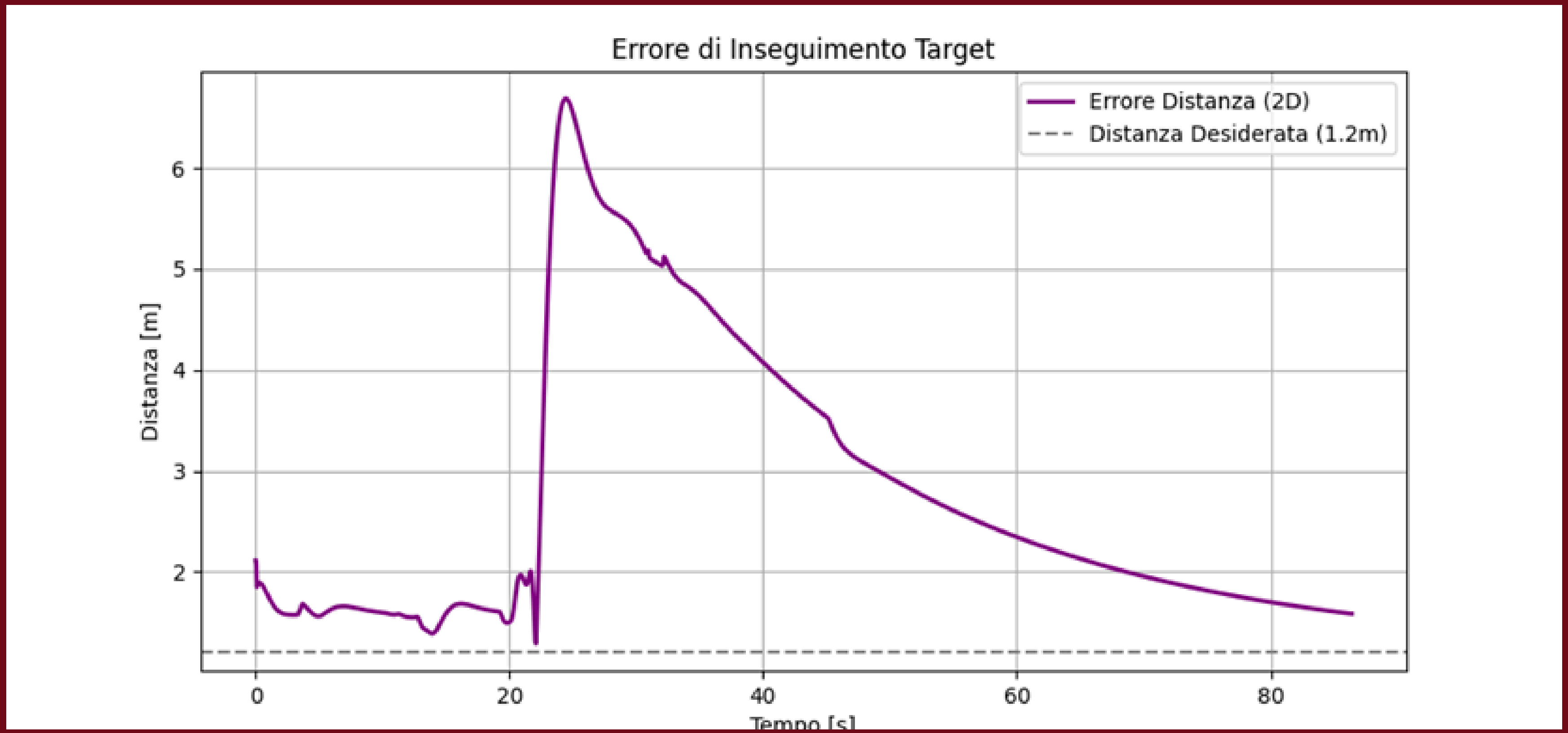
- ***Lack of repeatability***: even with the same setup and parameters, different runs generate different 2D UAV paths, highlighting sensitivity to the exact timing/geometry of obstacle interactions.
- ***Abrupt deviations near obstacles***: when entering the obstacle influence region, the UAV performs sharp detours instead of a smooth bypass, temporarily breaking the nominal person-following behavior.
- ***Tracking is not recovered immediately***: after the detour, the UAV does not quickly realign to the target path / desired following configuration, which anticipates the error increase seen in the plots.

SIMULATION RESULTS AND DISCUSSION

OBSTACLES (NO HEURISTIC METHODS)

- **THRUST DIVERGENCE leeds to instability: the drone can’t achiave the desired altitude**
- **Target-Following Error:**
  - **Large Deviation: Significant loss of reference distance (1.2m) during avoidance.**
  - **Slow Recovery: Post-interaction error does not converge rapidly Poor Robustness (without heuristics).**



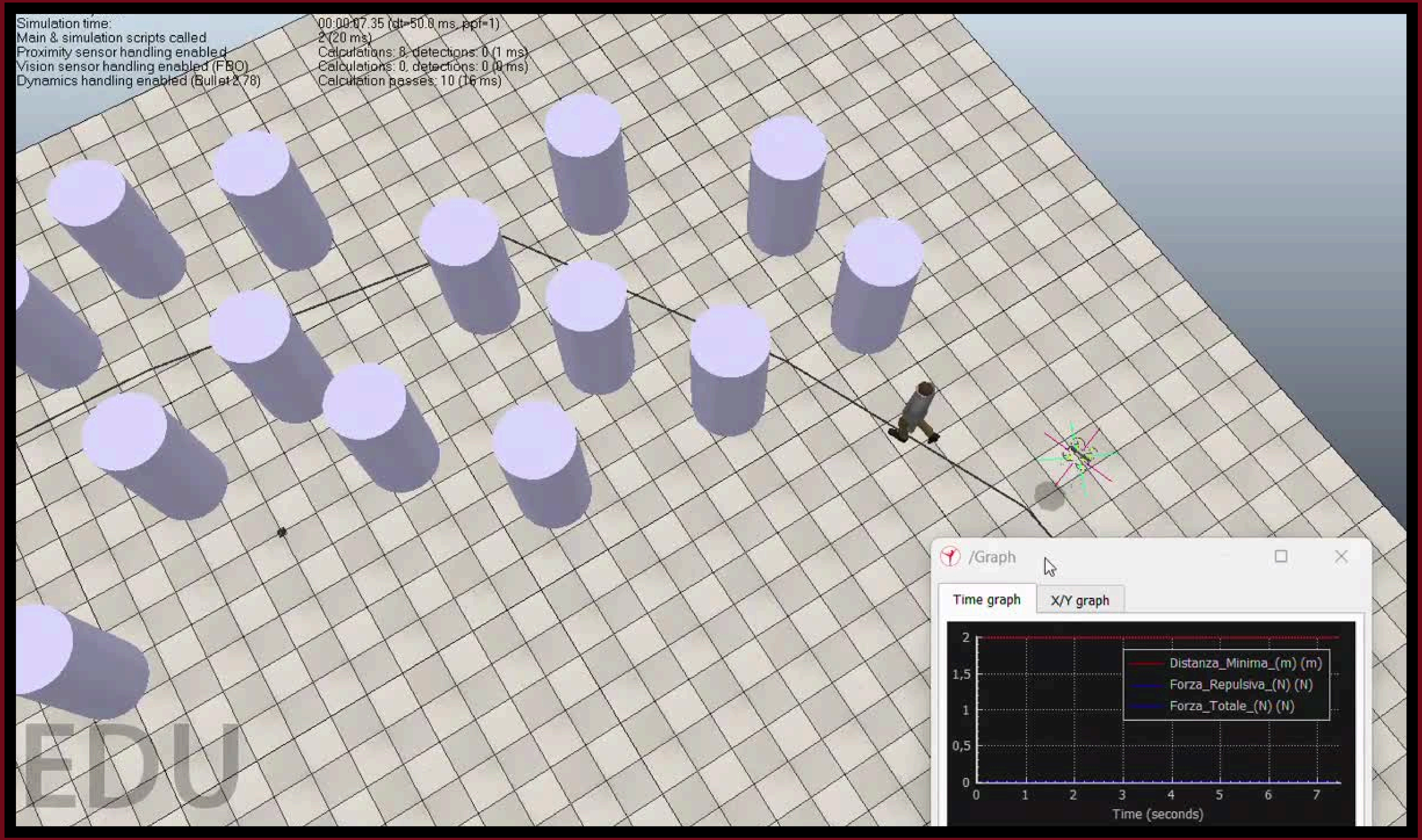


APF + VF parameters
k_rep = 0.4
k_vortex = 0.25

The control parameters are the same of the first simulation

SAPIENZA
UNIVERSITÀ DI ROMA

SIMULATION RESULTS AND DISCUSSION



HEURISTIC METHODS INTRODUCTION

- correction factors on the robot
- priority to security (avoid collisions)
- use of additional forces

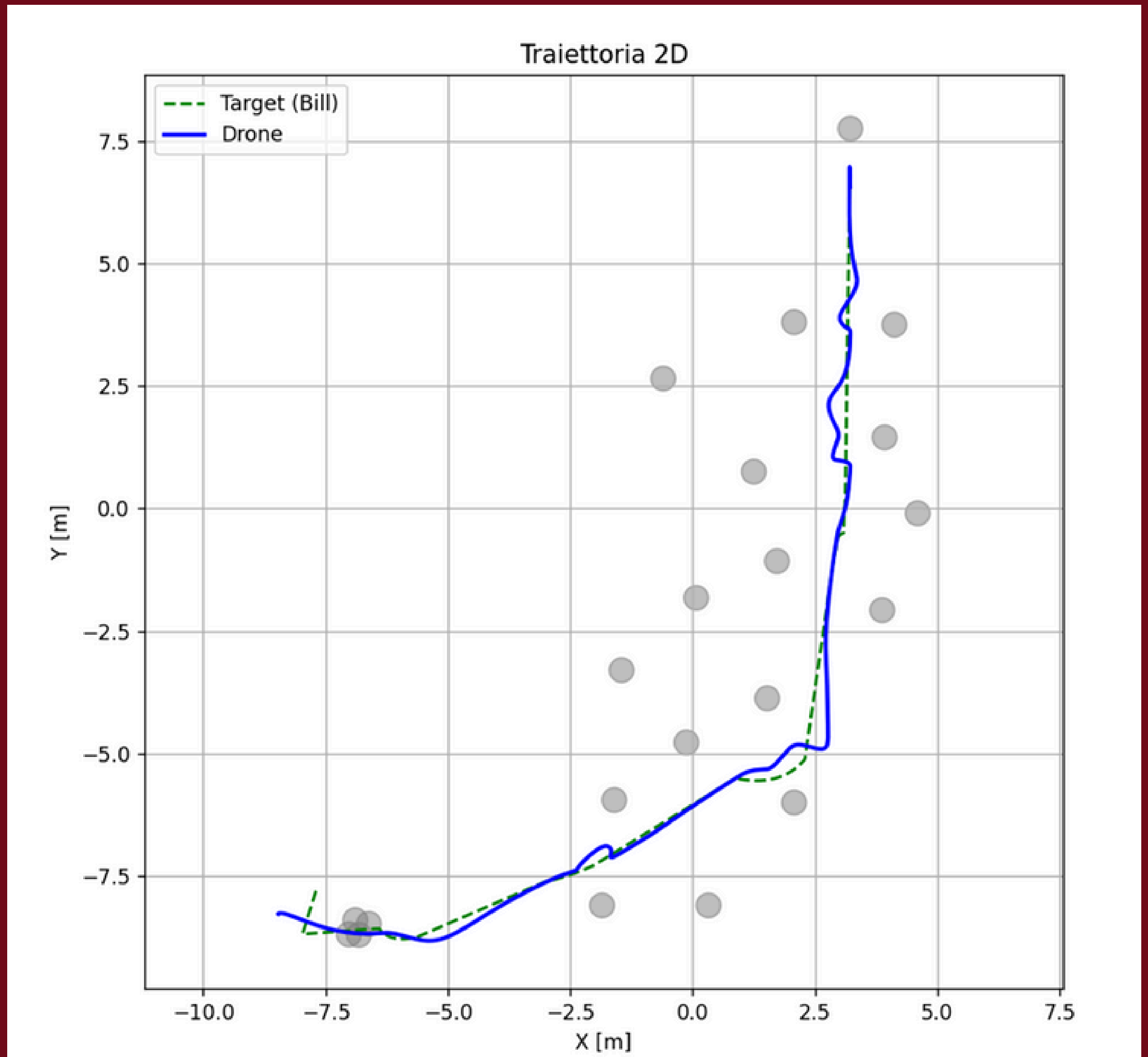
↓

1. stable behavior
2. no blocks in narrow points

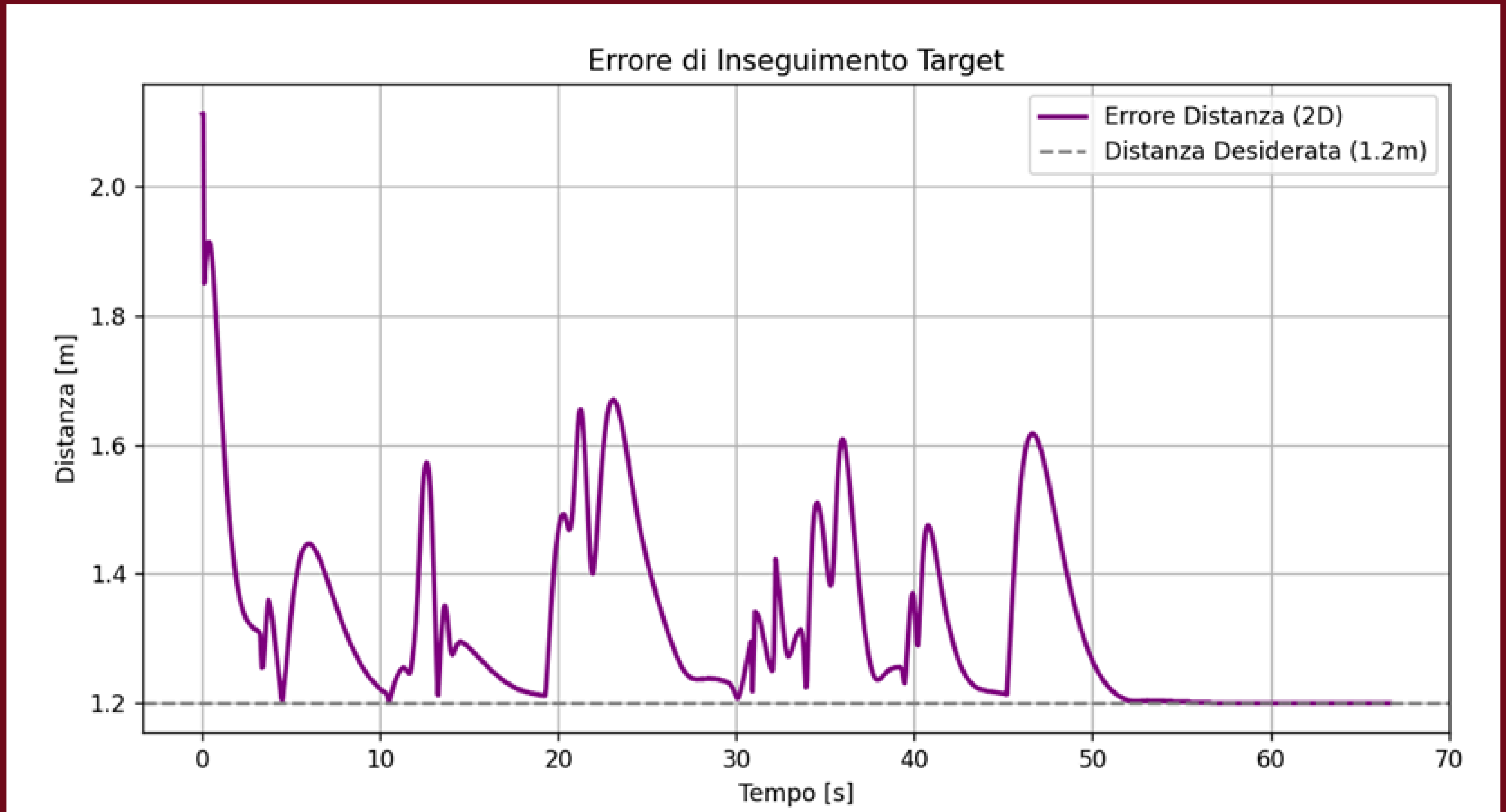
SAPIENZA
UNIVERSITÀ DI ROMA

SIMULATION RESULTS AND DISCUSSION

HEURISTIC METHODS INTRODUCTION



- **2D trajectory**: the UAV successfully follows the target path while navigating through the obstacle field. Near the obstacles, the drone performs smooth with limited lateral deviations, effectively bypassing them without generating large oscillations or trajectory loops. Once the obstacle region is cleared, the UAV naturally re-aligns behind the target.

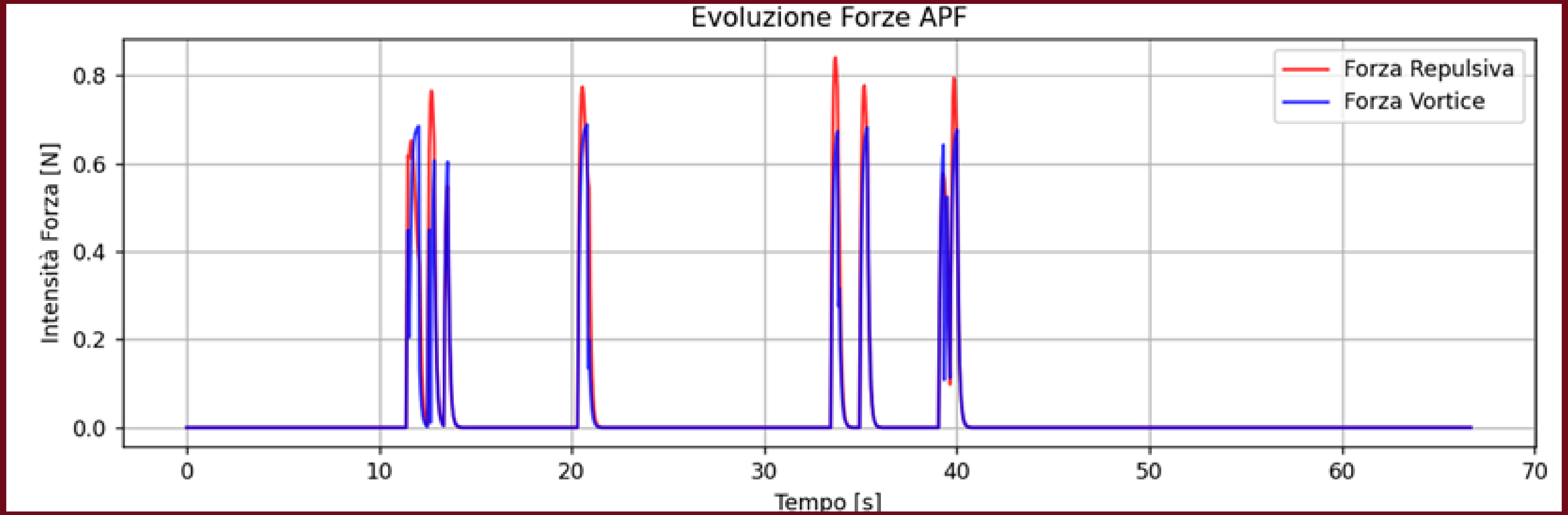


Distance error (2D): after a short initial transient, the error remains close to the desired reference distance of 1.2 m. Temporary peaks appear during obstacle interactions, but the error always remains bounded and converges back to the nominal value, demonstrating stable closed-loop behavior.

SAPIENZA
UNIVERSITÀ DI ROMA

SIMULATION RESULTS AND DISCUSSION

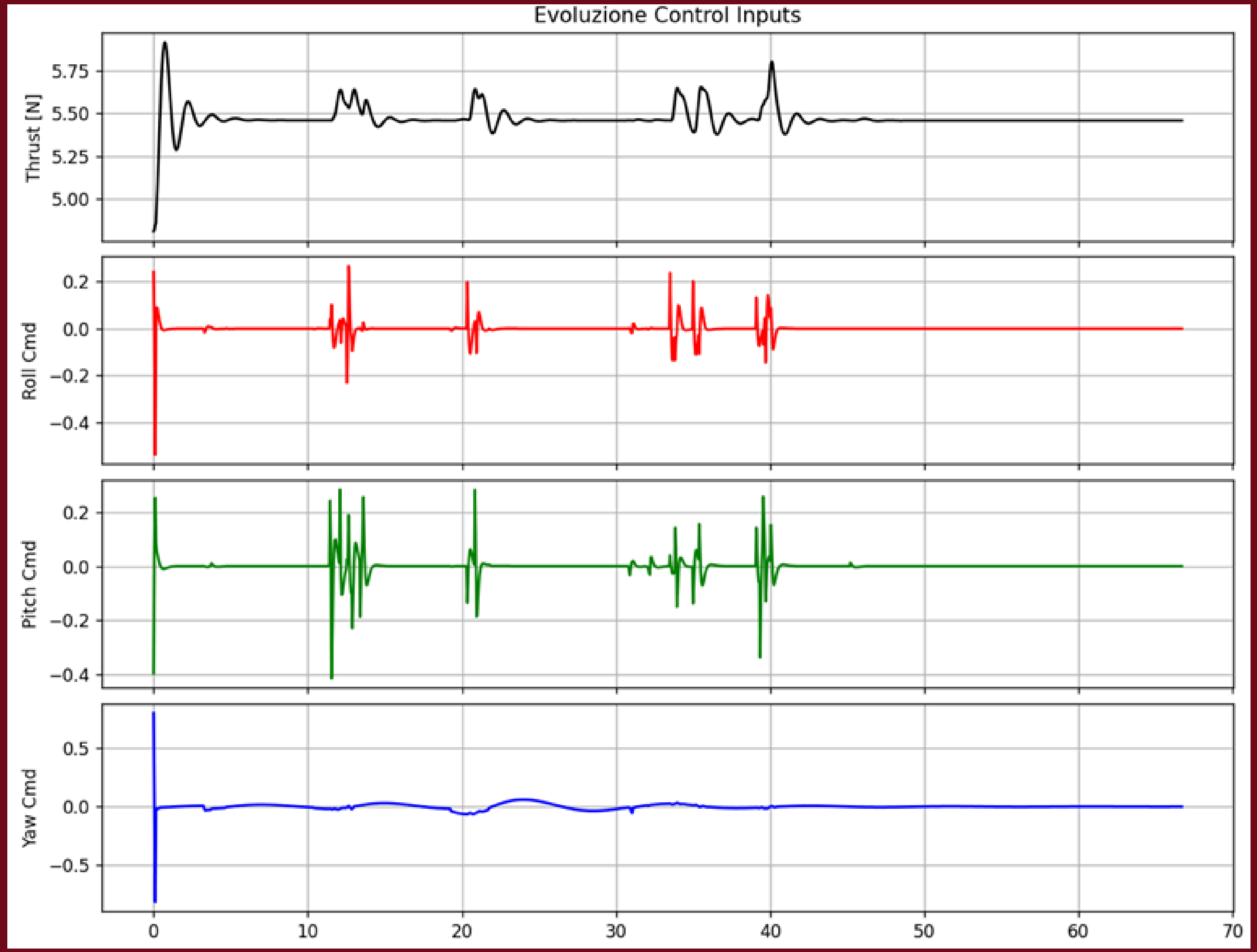
HEURISTIC METHODS INTRODUCTION



The heuristic additions (force shaping, smoothing, and blending) effectively stabilize the interaction between tracking and obstacle avoidance.

K_REP = 0.7
K_VORTEX = 0.7
D_ON = 0.3
D_OFF = 1.0
SMOOTHING_FACTOR = 0.4

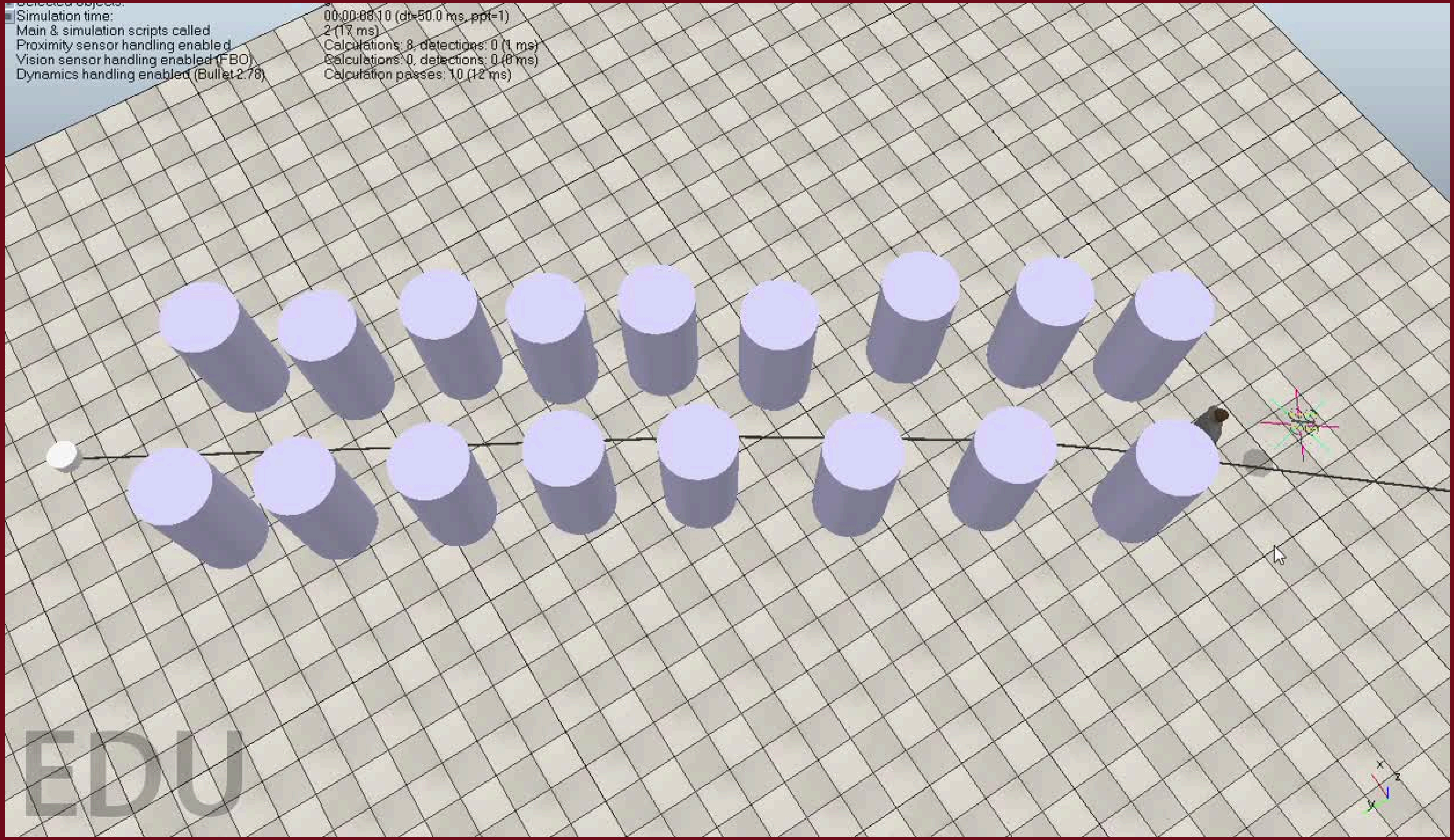
SAME VALUES OF THE PARAMETERS OF CONTROL



Control inputs

- Roll and pitch exhibit short bursts during obstacle avoidance without prolonged saturation
- Thrust stays close to its nominal value

SIMULATION RESULTS AND DISCUSSION

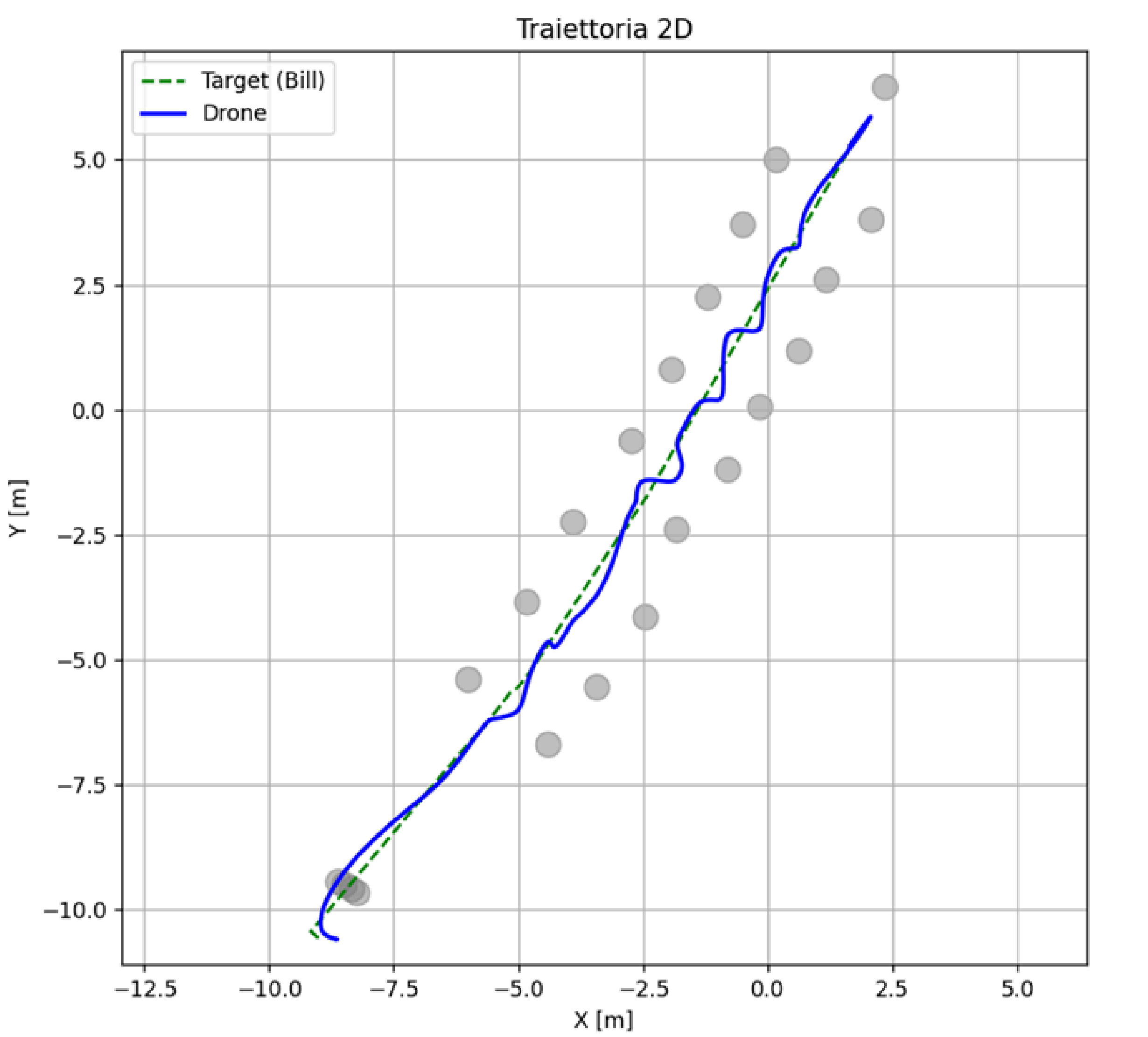


NARROW CORRIDORS

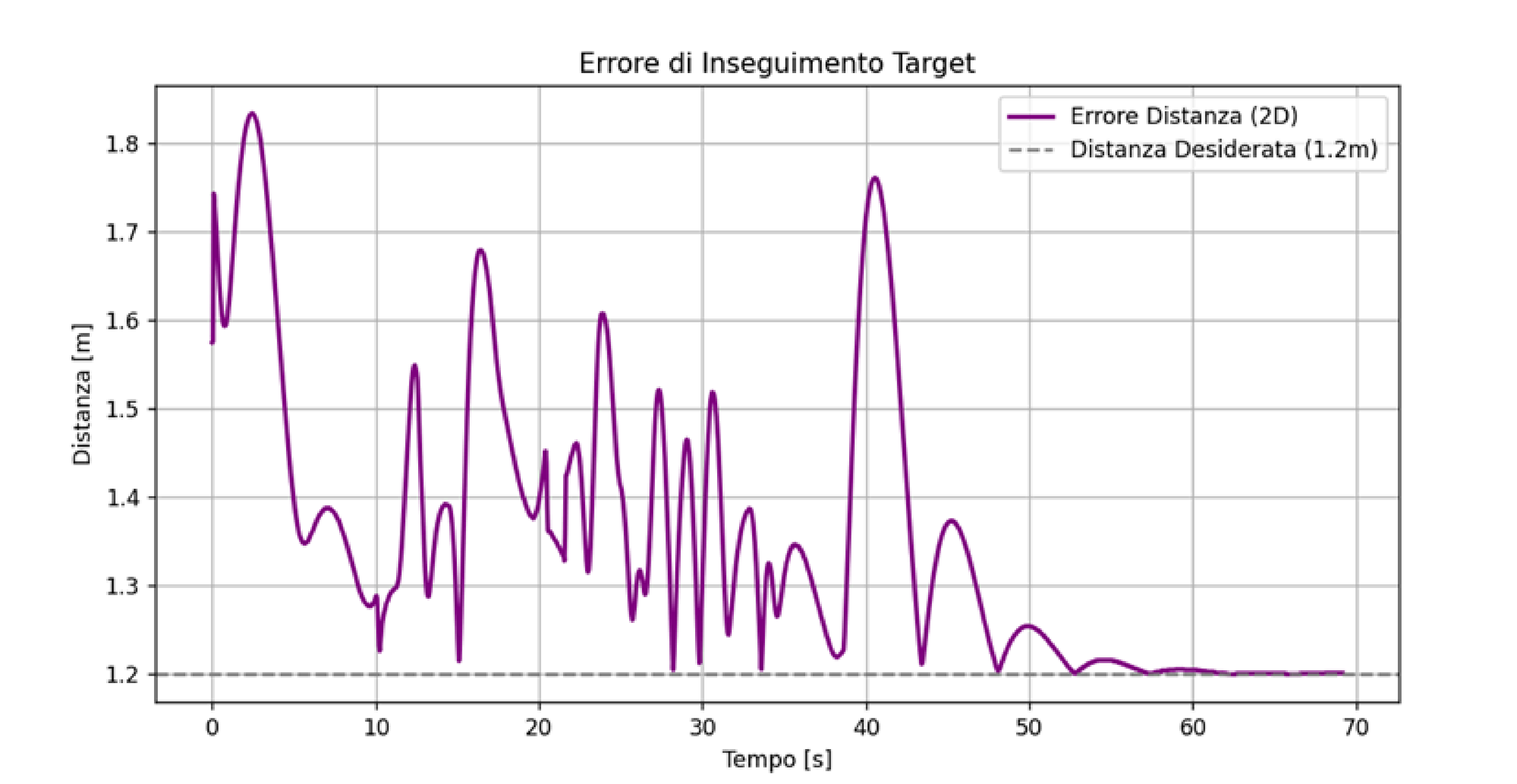
new heuristic method implementation, in order limit the robot swings, due to a bigger amount of computation in a small time, risking the quadcopter destabilization

SIMULATION RESULTS AND DISCUSSION

NARROW CORRIDORS



2D trajectory: UAV remains confined within the free space of the corridor while safely following the target.
Lateral deviations are strongly constrained, and the drone performs continuous small corrections rather than wide detours.
The resulting trajectory is smooth and free of divergence or large oscillations, indicating stable behavior even in a highly constrained environment.

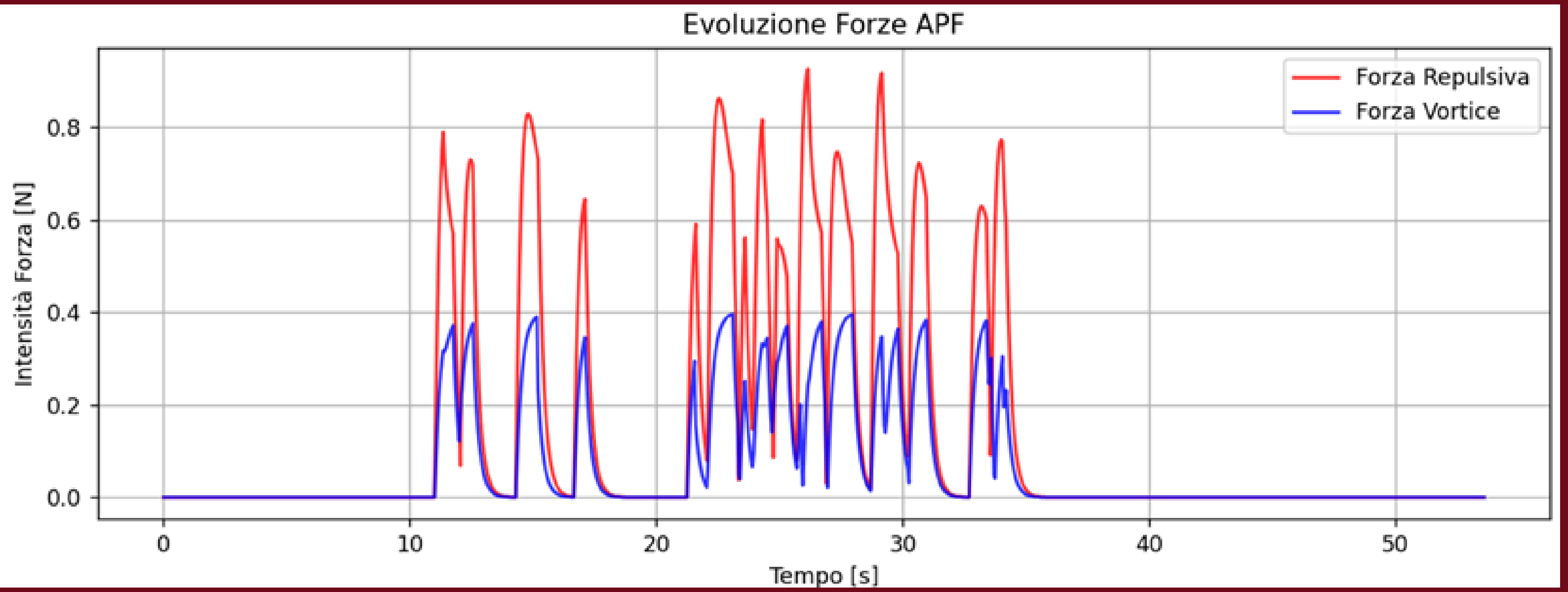


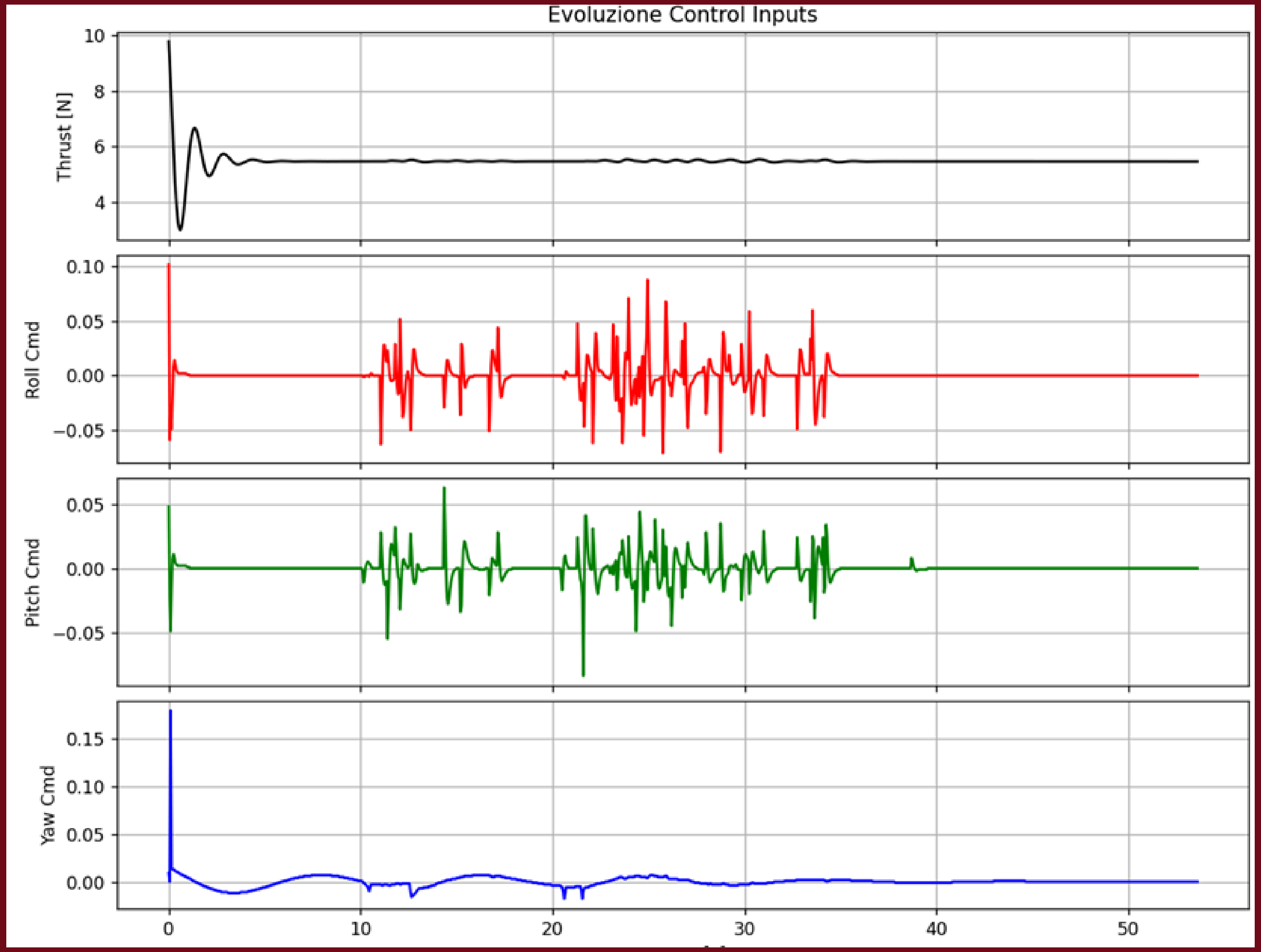
Distance error (2D): stronger interaction between tracking and avoidance in the corridor. During the traversal, the error oscillates around the reference value, with more frequent and slightly higher peaks than in the heuristic obstacle scenario.
Nevertheless, the error remains bounded and, once the corridor is cleared, it converges back toward the desired distance, showing the system’s ability to recover the nominal following configuration.

SAPIENZA
UNIVERSITÀ DI ROMA

SIMULATION RESULTS AND DISCUSSION

NARROW CORRIDORS





Control Inputs

- Roll/Pitch: Small magnitude (No aggressive maneuvers).
- Thrust: Fast stabilization Altitude

APF Forces Analysis

- Repulsive: High-frequency activation due to dense obstacle configuration.
- Vortex: Promotes flow along the corridor, preventing "push-back".

SAME PARAMETERS FOR THE HEURISTIC STRATEGY

K_POS_P = 0.01
K_POS_D = 0.3
K_ATT_P = 0.25
K_ATT_D = 2.0
K_YAW_P = 0.1
K_YAW_D = 1.0

CONCLUSIONS

- The proposed architecture achieves stable and accurate person-following in nominal conditions: the UAV converges to the desired following distance (1.2 m) with smooth, bounded control inputs.
- With obstacles and without stabilization heuristics, the closed-loop behavior becomes sensitive and non-robust: trajectories are not repeatable, avoidance forces generate impulsive peaks, and roll/pitch can reach saturation, leading to tracking degradation.
- The main contribution is the reactive vortex-based avoidance: tangential components enable the UAV to slide around obstacles, reducing typical issues of purely repulsive fields (e.g., local minima / stop near obstacles).
- **Signal conditioning is essential for stability:** continuous blending, hysteresis, and force saturation mitigate switching effects and prevent chattering near safety boundaries, yielding a computationally efficient and reliable solution in simulation.

FUTURE WORKS

- **State Estimation and Sensor Noise:** The current simulation assumes ideal, noise-free sensing. Future work should integrate a state estimator, such as an EKF (Extended Kalman Filter), to handle measurement uncertainty and sensor delays, which are unavoidable in physical hardware

- **Dynamic Obstacle Avoidance:** While the current system effectively handles static environments, extending the vortex logic to account for the velocity vector of moving obstacles would significantly improve safety in dynamic crowds

- **Vision-Based Target Tracking:** Replacing the ground-truth position data provided by the simulator with an onboard computer vision module (e.g., using YOLO or similar algorithms) would render the system fully autonomous and deployable without external motion capture systems

THANK YOU !