Development of a Control Algorithm for a Two-Axis Gimbal for Object Tracking on UAV

Done by Student:
Zorian Kulyk
Thesis Supervisor:
Andrew Kurochkin

Kyiv, 2026

Content

1. Motivation and Objectives
2. Technology Stack and System Design
3. Coordinate systems
4. Control algorithm
5. Limitations
6. Evaluation

Motivation

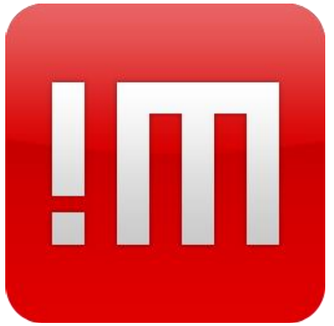
1. Computational budget
2. A little work addresses a complete, low-cost gimbal tracking system

Objective

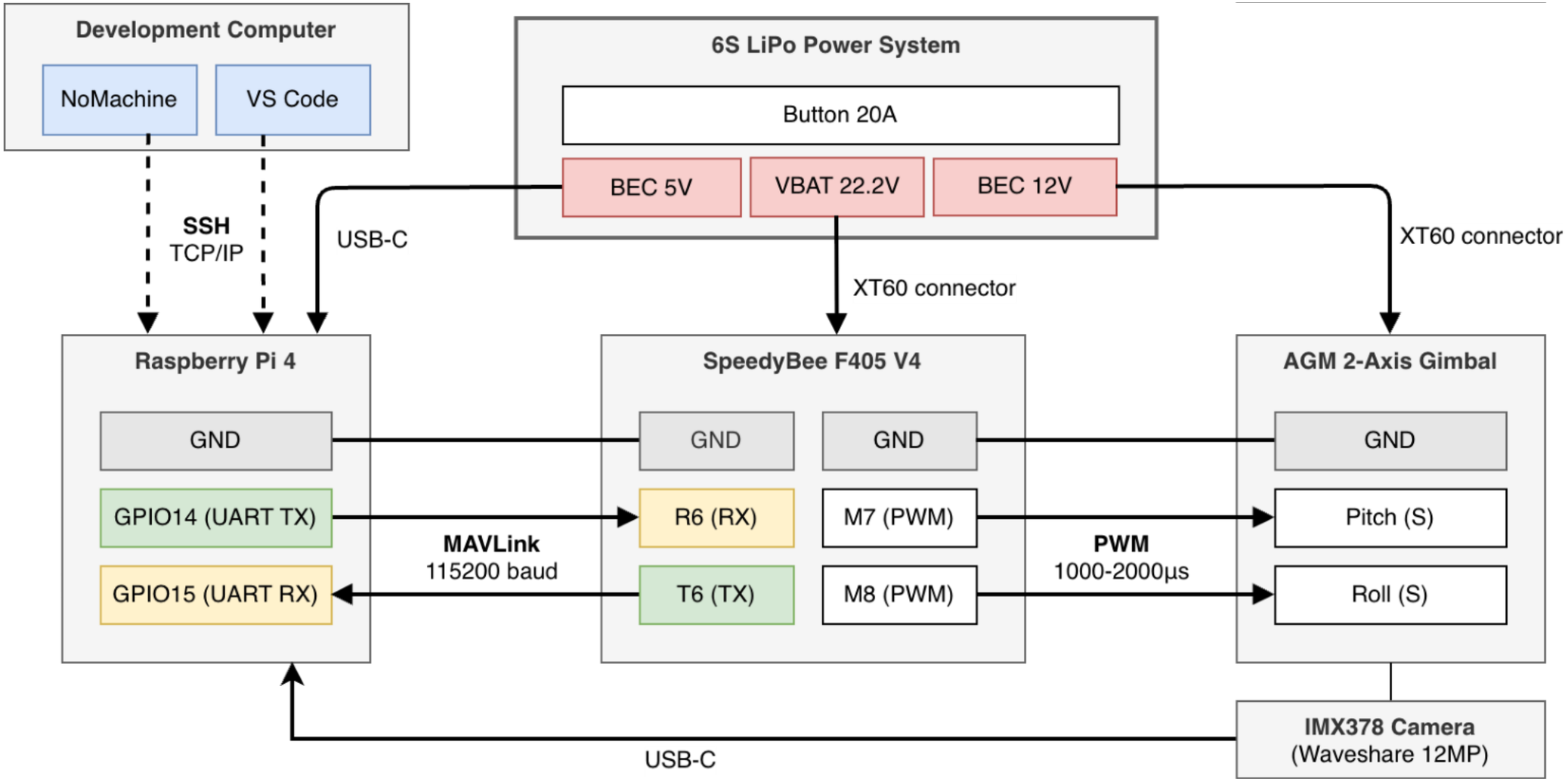
1. Develop a control algorithm under hardware constraints
2. Validate system capability in realistic scenarios
3. Demonstrate the overall feasibility of the approach

Technology Stack

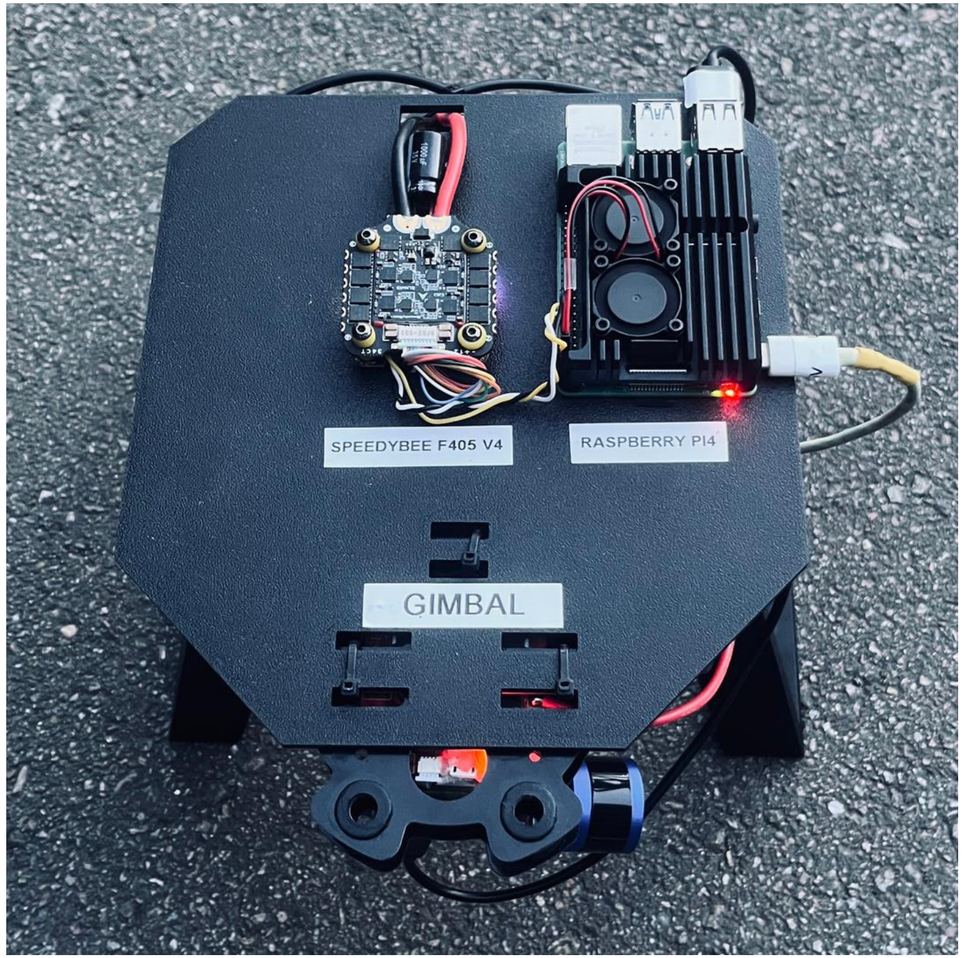
- Remote Desktop: NoMachine
- Development: VS Code IDE, VS Code Server
- Tracking Foundations: Python 3.12, OpenCV

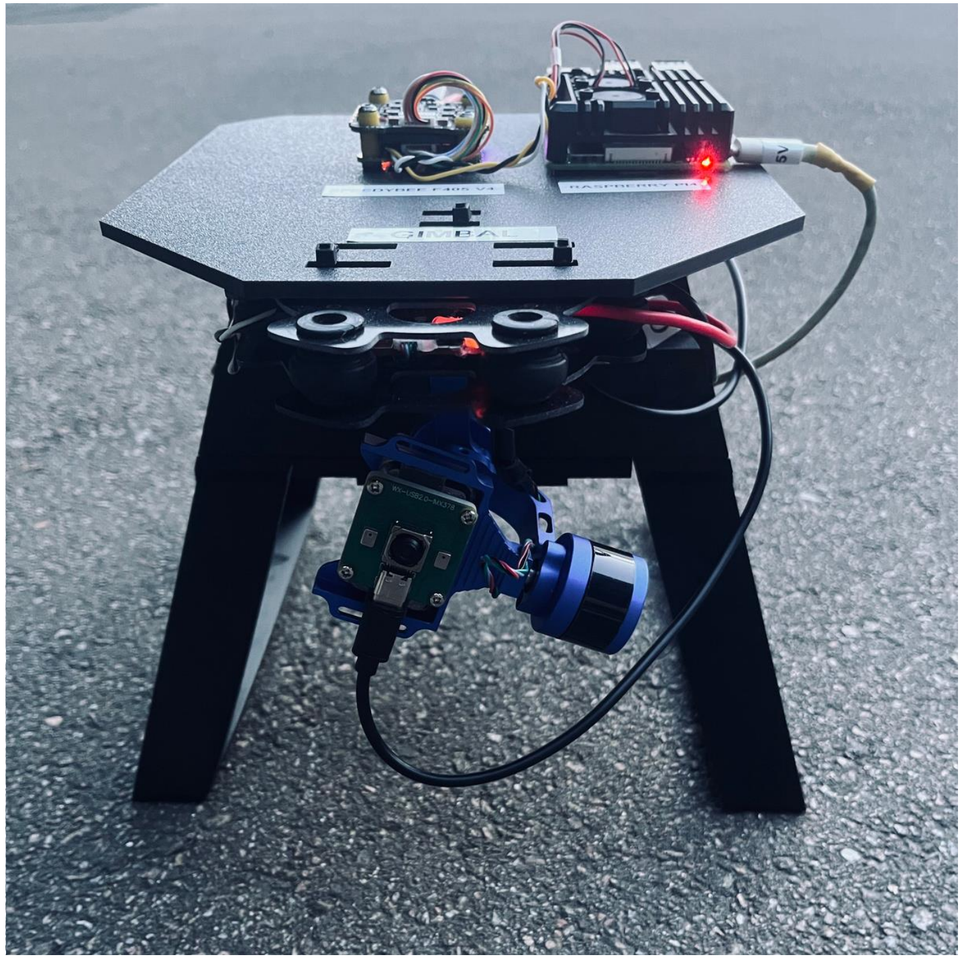


System Design



Physical Layout





Coordinate System of Gimbal

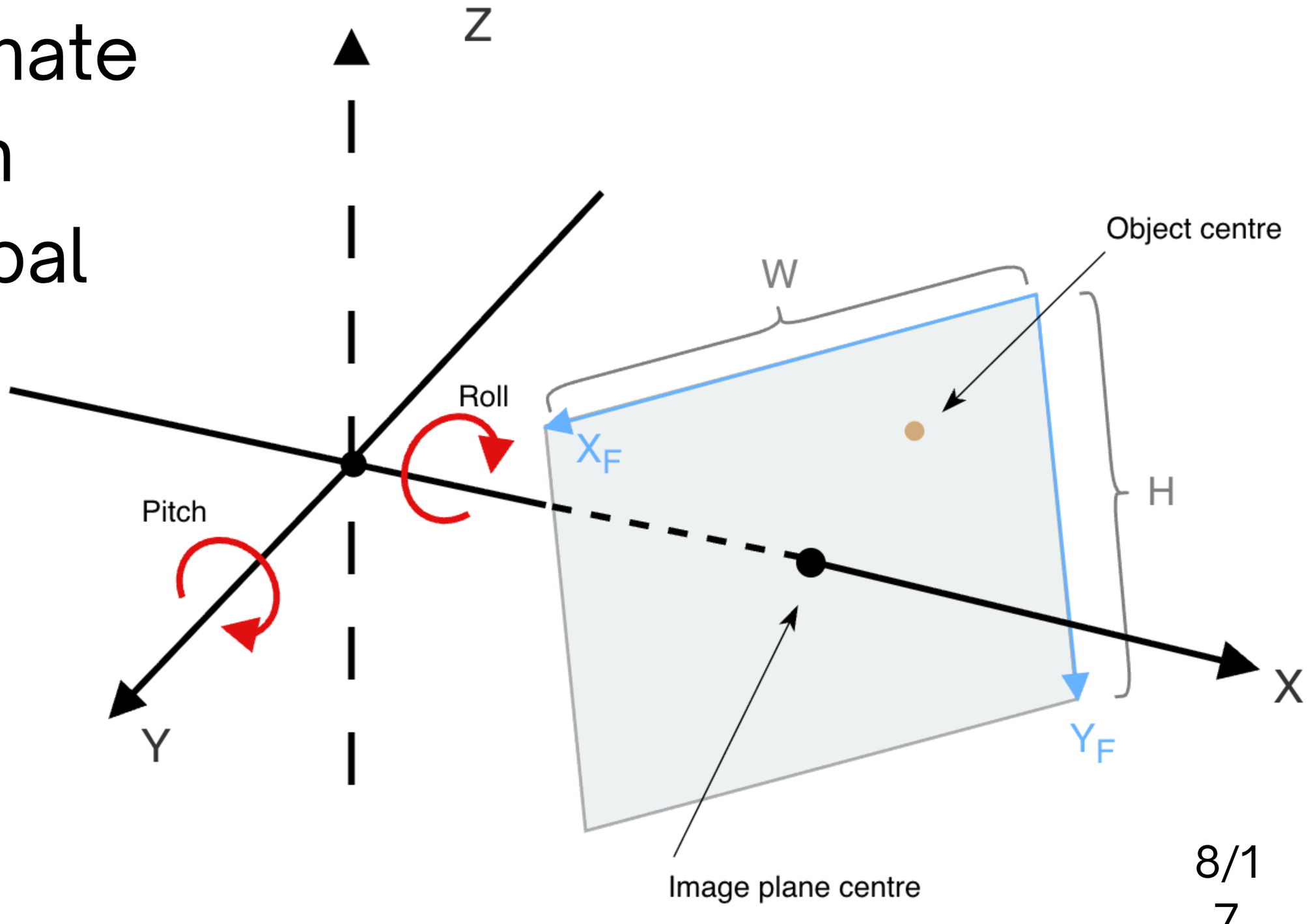
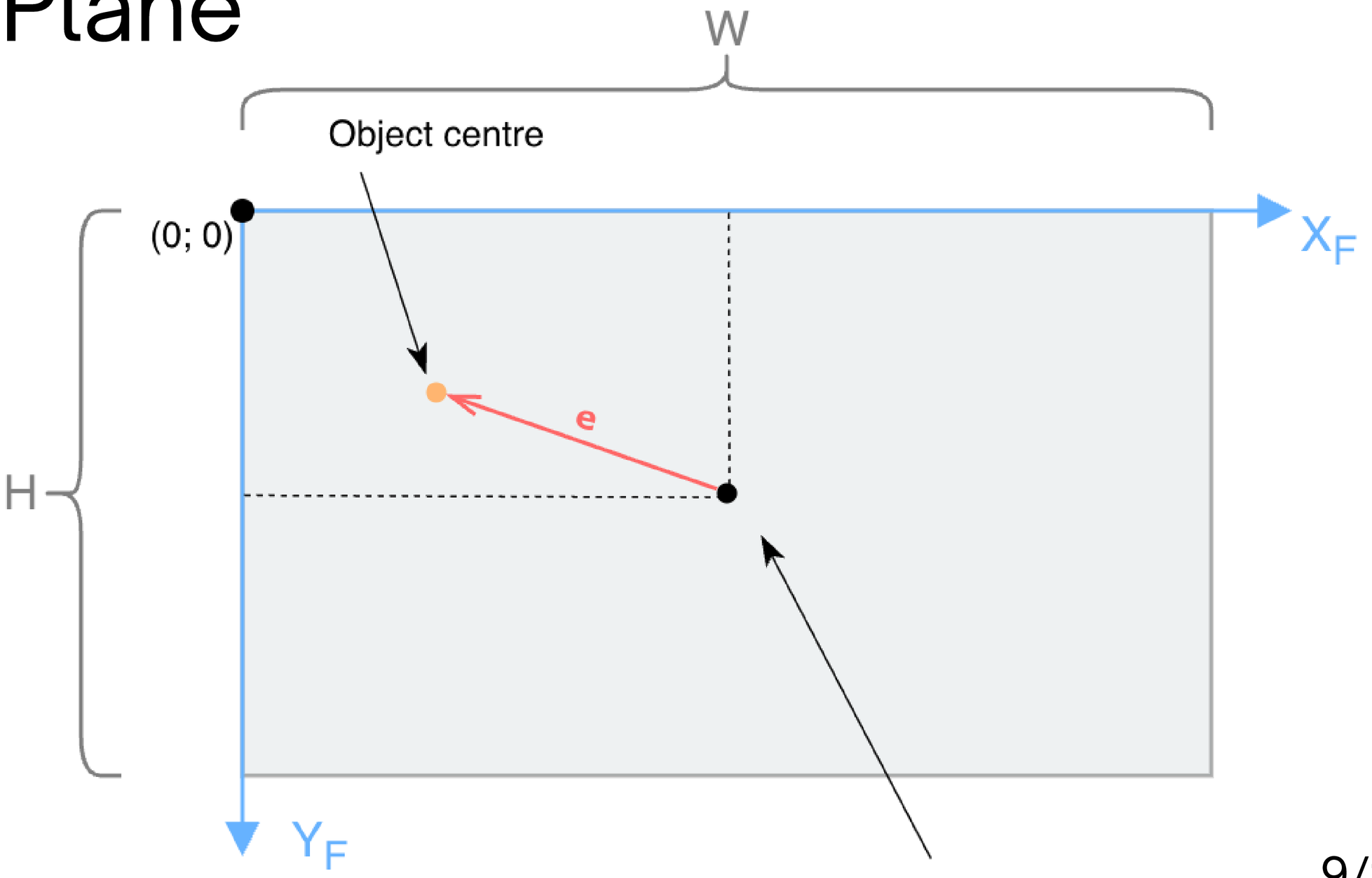


Image Plane



For a 3-axis gimbal, the full rotation from the base frame to the camera frame is obtained by multiplying all three matrices in the yaw-roll-pitch order [2]:

$$\mathrm{R}_3 = \mathrm{R}(\theta_1) \cdot \mathrm{R}(\theta_2) \cdot \mathrm{R}(\theta_3) \tag{2.4}$$

Expanding (2.4):

$$\mathrm{R}_3 = \begin{bmatrix} c_1c_3 - s_1s_2s_3 & -c_2s_1 & c_1s_3 + c_3s_1s_2 \\ c_3s_1 + c_1s_2s_3 & c_1c_2 & s_1s_3 - c_1c_3s_2 \\ -c_2s_3 & s_2 & c_2c_3 \end{bmatrix} \tag{2.5}$$

where the shorthand notation $c_i = \cos\theta_i$ and $s_i = \sin\theta_i$ for $i = 1, 2, 3$ is used for compactness.

For the 2-axis gimbal, the yaw angle is absent ($\theta_1 = 0$). Substituting into (2.5):

$$\mathrm{R}_2 = \mathrm{R}_2(\theta_2, \theta_3) = \mathrm{R}_3\big|_{\theta_1=0} = \mathrm{R}(\theta_2) \cdot \mathrm{R}(\theta_3) = \begin{bmatrix} c_3 & 0 & s_3 \\ s_2s_3 & c_2 & -c_3s_2 \\ -c_2s_3 & s_2 & c_2c_3 \end{bmatrix} \tag{2.6}$$

Forward Kinematics

Control Algorithm

The motor position command is defined by an iterative update of the previous PWM signal:

$$p_{i+1} = \begin{cases} p_i, & \text{if } |e_i| \le \delta \\ \max\left\{1000,\ \min\left\{2000,\ p_i + SR \cdot \left(c_i - \frac{N}{2}\right)\right\}\right\}, & \text{otherwise} \end{cases} \tag{2.17}$$

Proposal 1. *Let $\Delta_{\max}$ be the full control angle range of a gimbal axis, $\Delta p_{\max}$ be the full PWM command span, and γ and N denote the FOV and pixel extent of the image along that same axis. Under the pinhole camera model (2.18), the spatial resolution coefficient that drives the pixel tracking error to zero in a single step is*

$$SR = \frac{2\tan(\gamma/2) \cdot \Delta p_{\max}}{N \cdot \Delta_{\max}} \tag{2.22}$$

where $\Delta_{\max}$ is expressed in radians.

Limitations

1. Only one rotational degree of freedom (1 rotational d.o.f. + 1 orientation d.o.f.)
2. High total delay per tracking cycle
3. Low gimbal rotation speed
4. Static physical layout

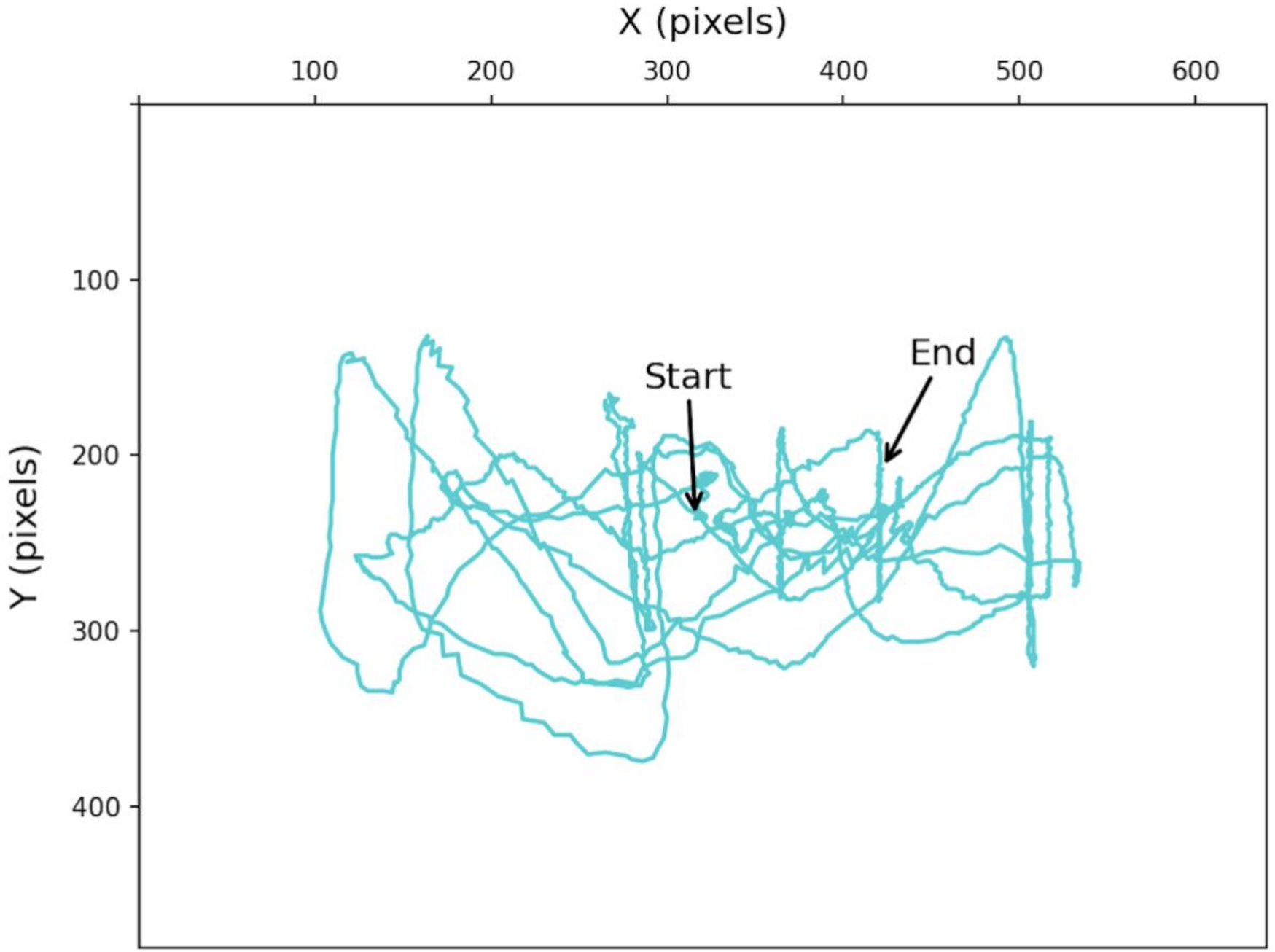
Evaluation

An experiment setup:

- Mount elevation: 2 m
- Experimental area: ≈200 m^2
- Duration: 5 min

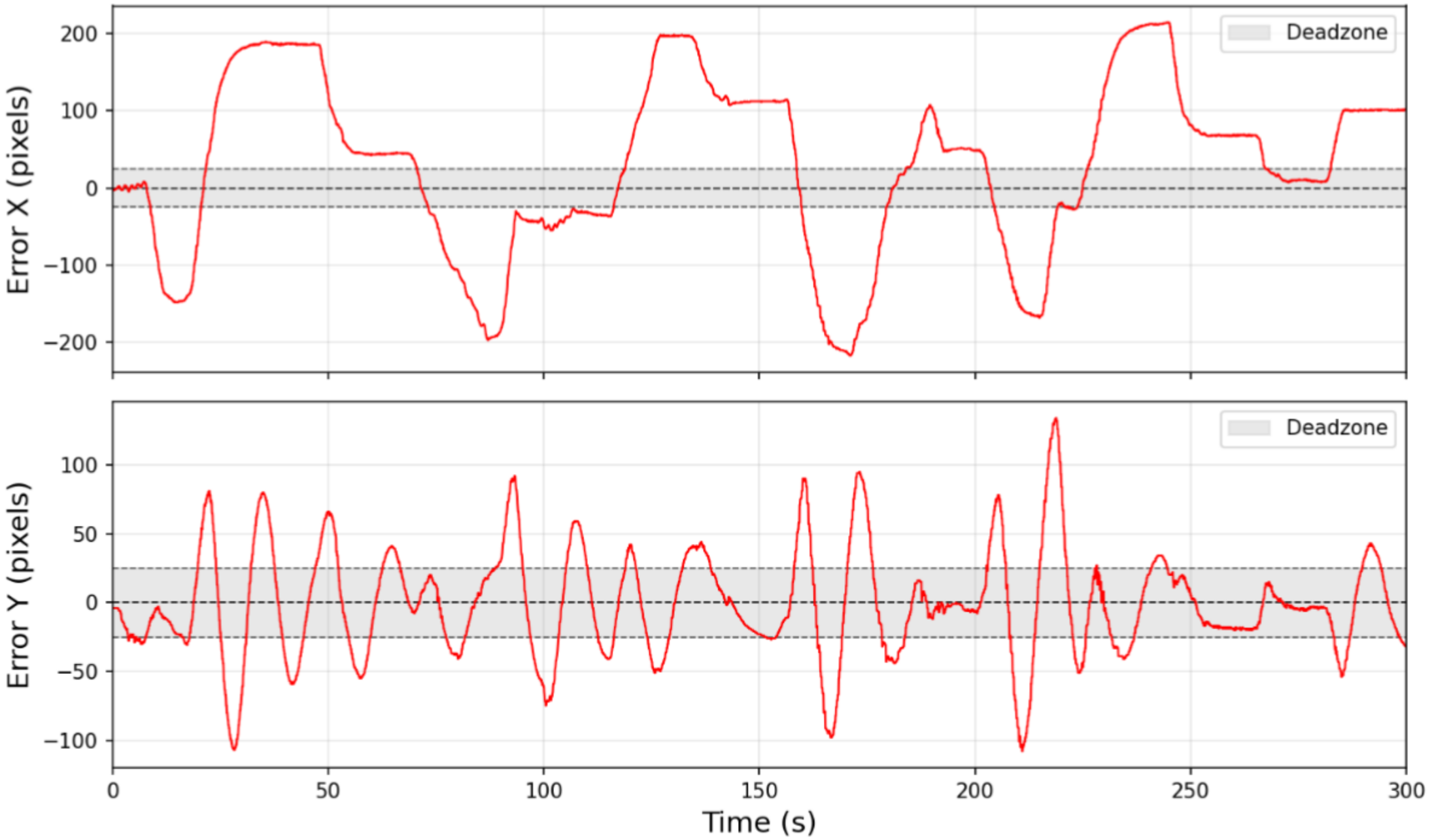
System parameters setup:

- FPS = 20
- SR pitch = 0.1
- SR roll = 1.33
- Tracker: CSRT
- Resolution: 640x480

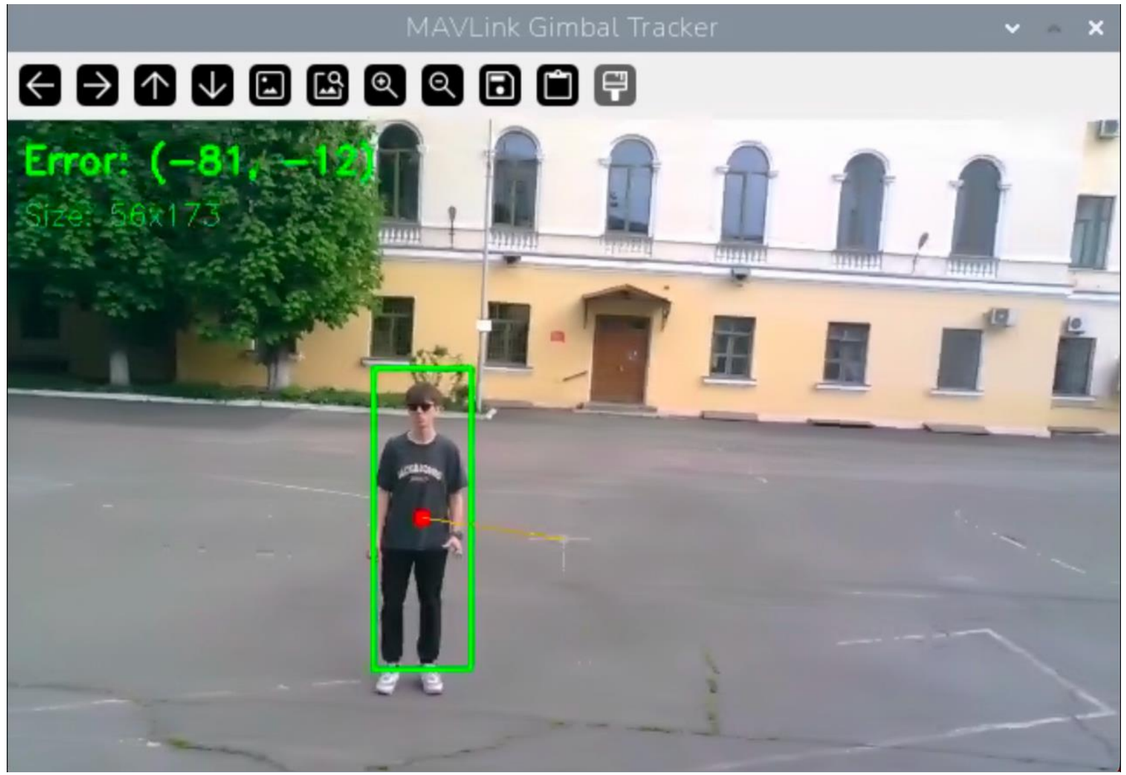


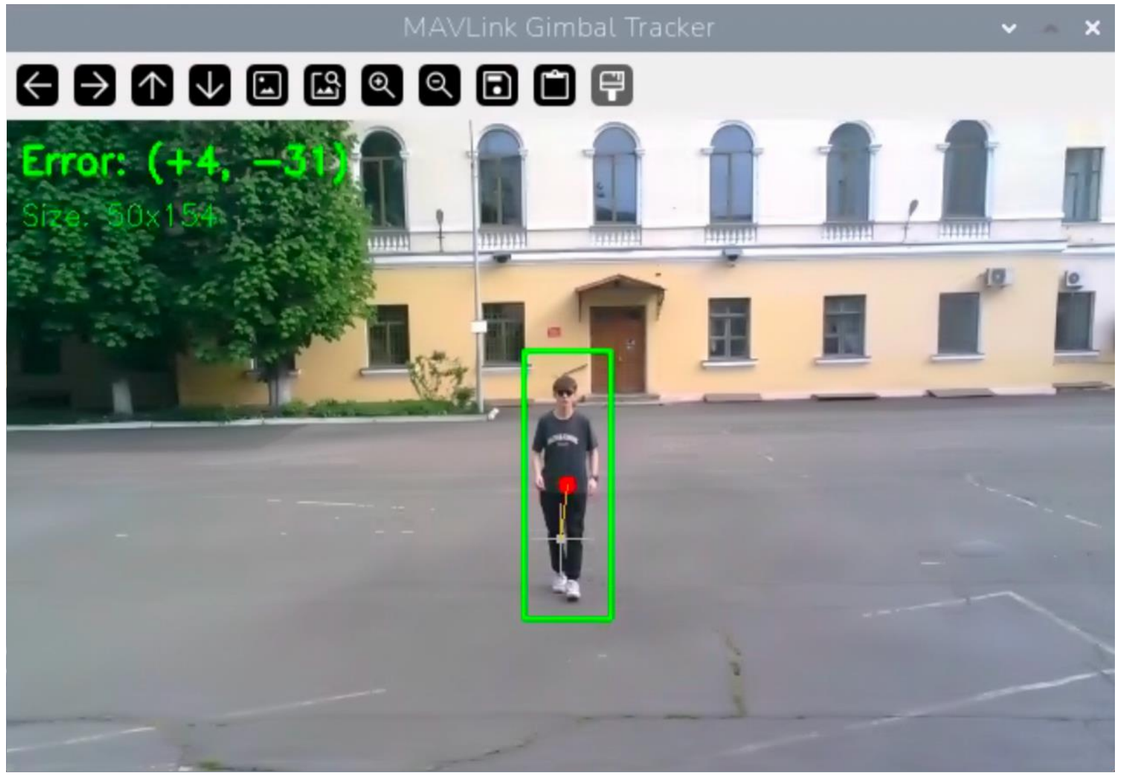
Object movement trajectory for CSRT

Object tracking error on both axes for CSRT



Real-Time Tracking





demo link:

https://youtu.be/uLFLAKk6mI8

Conclusions and Further Work

It is feasible to develop a low-cost gimbal-tracking system using the proposed control algorithm, despite significant hardware constraints.

Future work must address the current limitations:

1. Rotation degrees of freedom
2. Tracking cycle delay
3. Gimbal speed
4. Static physical layout