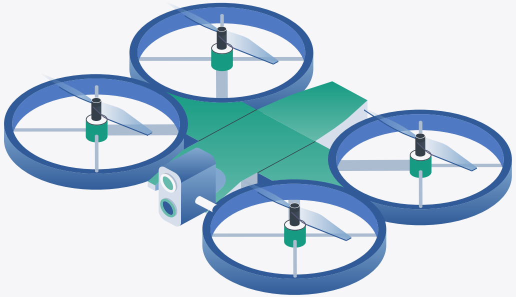


DRONE NAVIGATION THROUGH CONSTRAINED GATE ENVIRONMENTS

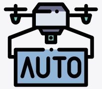
TEAM 44: 206A GROUP

Alejandro Municio, Ben Finch, Michael Howo, Rohin Shanker

01

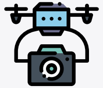
INITIAL GOALS

Develop an autonomous drone capable of finding optimal paths to fly through defined obstacles using:



Optimal Control

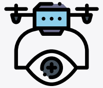
Using LQR (linear-quadratic regulator) Algorithm



Onboard Camera

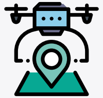
Drone is only equipped with a single onboard camera

Build a software pipeline for:



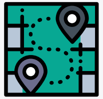
Perception

Detect gates and obstacles



Localization

Localizing drone using solely visual inputs



Trajectory Planning

Control using cost function & constraints

Optimal Control Algorithm

Why we chose to use a Discrete-Time Receding Horizon LQR controller

- Well documented, computes fast
- Can be used to adjust flight characteristics
- Reactive to environmental changes
- Horizon can be chosen to balance compute time and reactivity

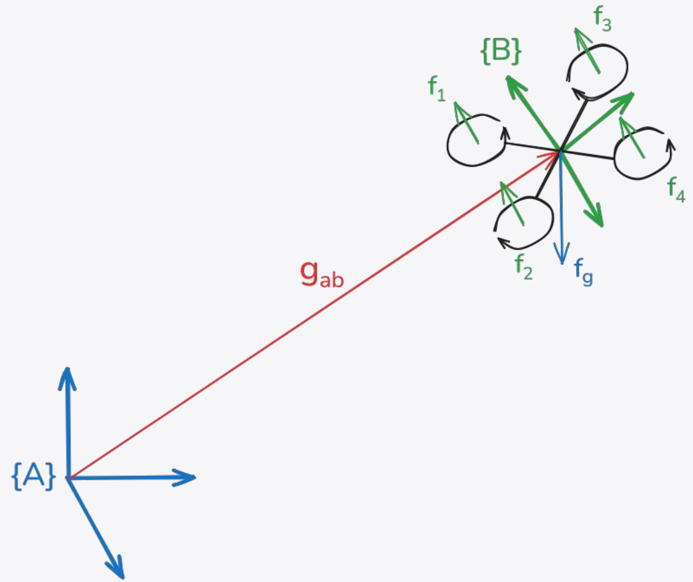
02

CALCULATIONS

Dynamics

Forces:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = R_{ab}(\phi, \theta, \psi) \begin{bmatrix} 0 \\ 0 \\ f_{total} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$

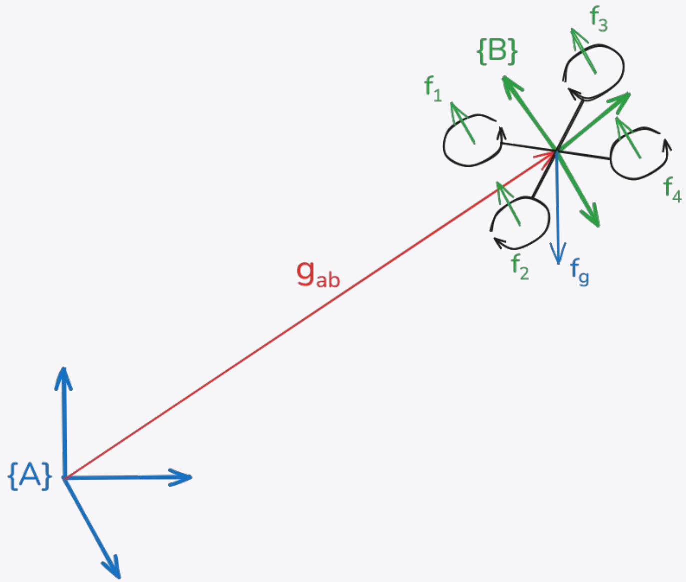


Dynamics

Linearization: Many Small Angle Approximations

$$R_{ab}(\phi,\theta,\psi) = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix}$$

$$\approx \begin{bmatrix} 1 & -\psi & \theta \\ \psi & 1 & -\phi \\ -\theta & \phi & 1 \end{bmatrix}$$



Dynamics

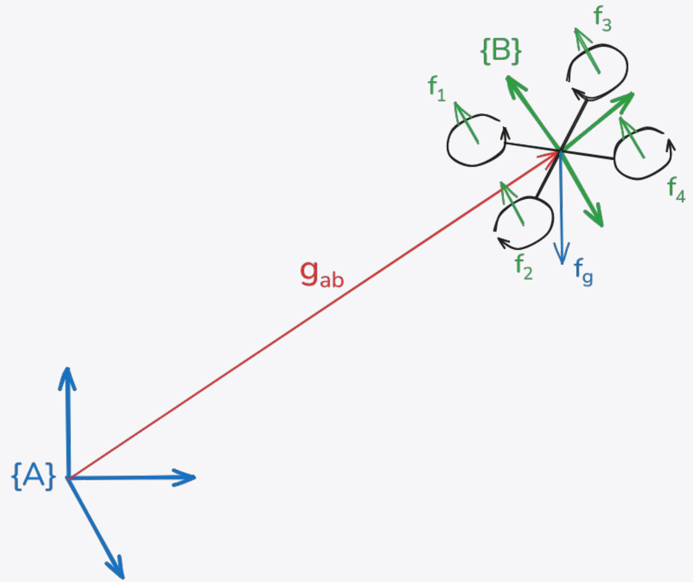
Forces:

$$m\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = R_{ab}(\phi,\theta,\psi)\begin{bmatrix}0\\0\\f_{total}\end{bmatrix} + \begin{bmatrix}0\\0\\-g\end{bmatrix}$$

$$m\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} \approx \begin{bmatrix}1 & -\psi & \theta\\ \psi & 1 & -\phi\\ -\theta & \phi & 1\end{bmatrix}\begin{bmatrix}0\\0\\f_{total}\end{bmatrix} + \begin{bmatrix}0\\0\\-g\end{bmatrix}$$

$$f_{total} \approx \Delta f_{total} + mg$$

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} \approx \begin{bmatrix}\theta g\\ -\phi g\\ \Delta f_{total}/m\end{bmatrix}$$



LINEARIZATION

More Small Angle Approximations

$$\begin{aligned} p &= \dot{\phi} \\ q &= \dot{\theta} \\ r &= \dot{\psi} \end{aligned} \quad \Rightarrow \quad \begin{bmatrix} I_{xx}\,\dot{p} \\ I_{yy}\,\dot{q} \\ I_{zz}\,\dot{r} \end{bmatrix} = \begin{bmatrix} (I_{yy} - I_{zz})\,qr \\ (I_{zz} - I_{xx})\,pr \\ (I_{xx} - I_{yy})\,pq \end{bmatrix} + \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}$$

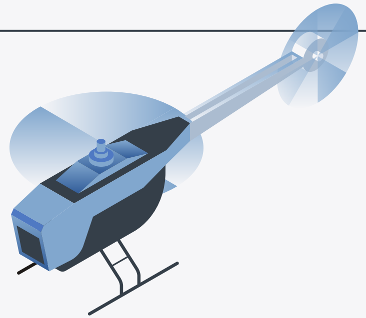
$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \approx \begin{bmatrix} \frac{1}{I_{xx}}\tau_x \\ \frac{1}{I_{yy}}\tau_y \\ \frac{1}{I_{zz}}\tau_z \end{bmatrix}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & \theta \\ 0 & 1 & -\phi \\ 0 & \phi & 1 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \approx \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

FULLY LINEARIZED MODEL

$$\frac{d}{dt}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\\ \dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\\ \ddot{x}\\ \ddot{y}\\ \ddot{z}\\ \ddot{\phi}\\ \ddot{\theta}\\ \ddot{\psi}\end{bmatrix} = \begin{bmatrix}0&0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&0&0&1\\ 0&0&0&0&g&0&0&0&0&0&0&0\\ 0&0&0&-g&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}x\\ y\\ z\\ \phi\\ \theta\\ \psi\\ \dot{x}\\ \dot{y}\\ \dot{z}\\ \dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} + \begin{bmatrix}0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 1/m&0&0&0\\ 0&1/I_x&0&0\\ 0&0&1/I_y&0\\ 0&0&0&1/I_z\end{bmatrix}\begin{bmatrix}F_{\text{total}}\\ \tau_x\\ \tau_y\\ \tau_z\end{bmatrix}$$

MIXER MATRIX



Thrust from each propeller to torque abt each body axis

$$\begin{bmatrix} f_{total} \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \ell & -\ell & -\ell & \ell \\ -\ell & -\ell & \ell & \ell \\ k & -k & k & -k \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \begin{matrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{matrix}$$

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & \ell^{-1} & -\ell^{-1} & k^{-1} \\ 1 & -\ell^{-1} & -\ell^{-1} & -k^{-1} \\ 1 & -\ell^{-1} & \ell^{-1} & k^{-1} \\ 1 & \ell^{-1} & \ell^{-1} & -k^{-1} \end{bmatrix} \begin{bmatrix} f_{total} \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}$$

LQR OPTIMIZATION

$$\min_{x_1, x_2, ..., x_N, u_0, u_1, ..., u_{N-1}} \sum_{k=0}^{N-1} \left(x_k^\top Q \, x_k + u_k^\top R \, u_k \right) + x_N^\top P \, x_N$$

$$\text{subj. to} \quad \left.\begin{aligned} & x_{k+1} = Ax_k + Bu_k, \quad k = 0, \ldots, N-1 \\ & x_k \in \mathcal{X}, \; u_k \in \mathcal{U}, \quad k = 0, \ldots, N-1 \\ & x_o = x_o \end{aligned}\right\} \text{System Dynamics Constraints}$$

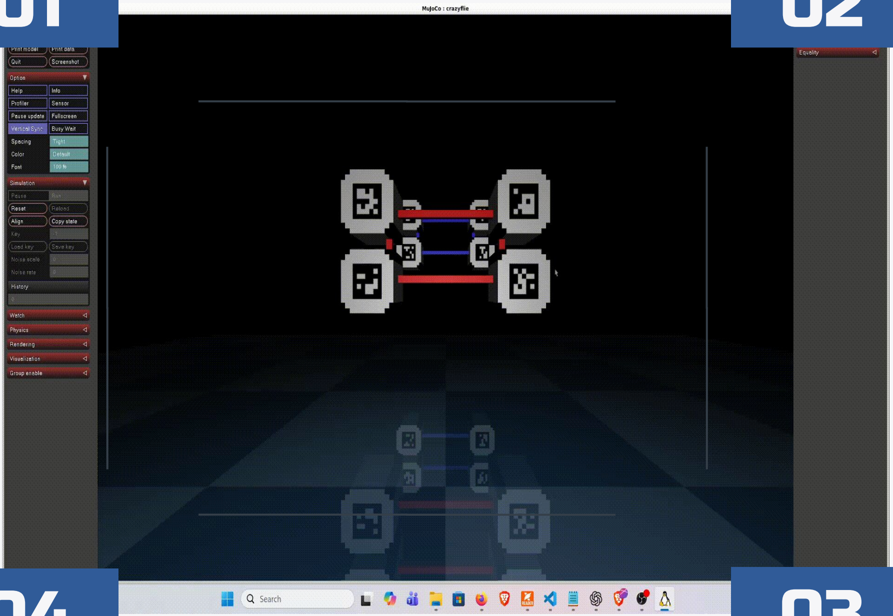
$$\left. u_{min} \leq u_k \leq u_{max} \right\} \text{Control Input Constraints}$$

03

SIMULATOR

MuJoCo Simulator

What we modeled:



01

Drone Dynamics

Symmetric Inertia Tensor, Limited Thrust & Torque Range

02

LQR Controller

Actuates thrust commands to each propeller

03

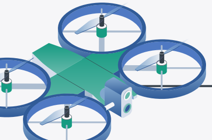
Camera

Simulated FPV Camera

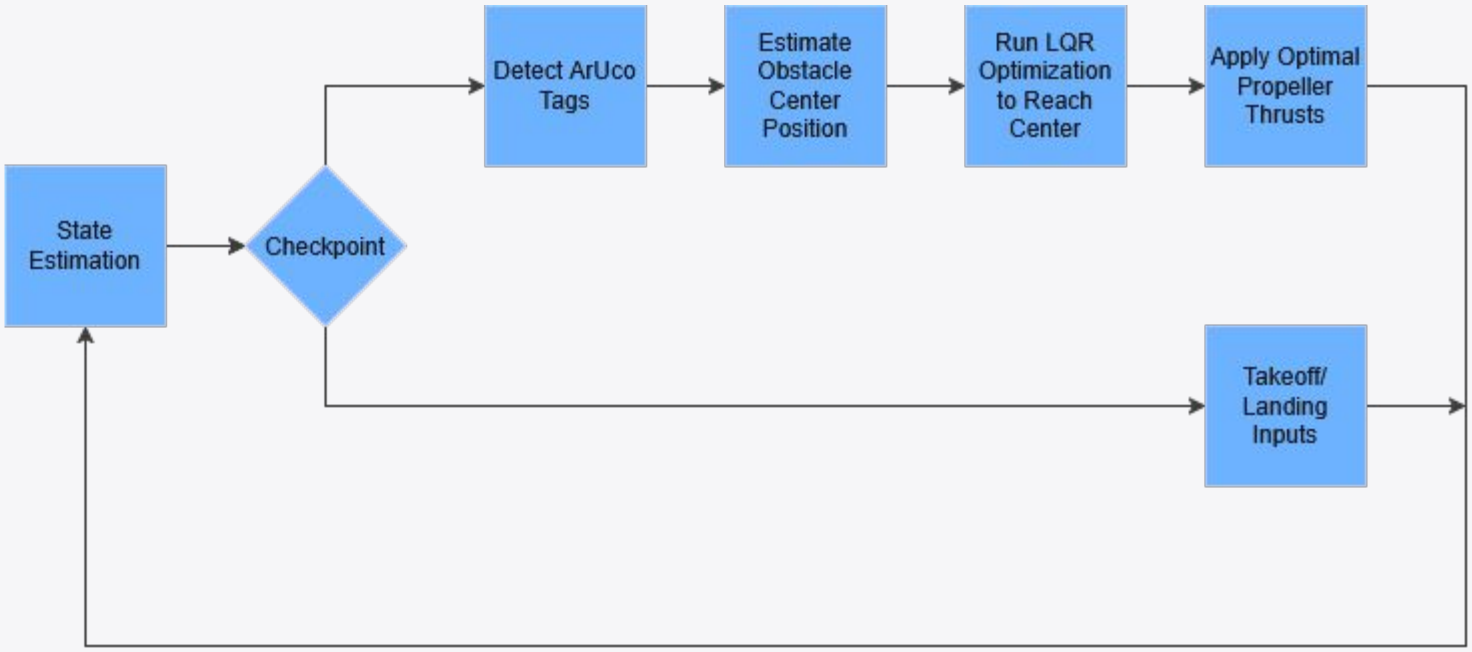
04

Physical Features

Gates marked with Aruco Tags



Autonomous Pipeline



Experimental Environment: Obstacle Setup

Experimental Conditions & Simplifications:

01

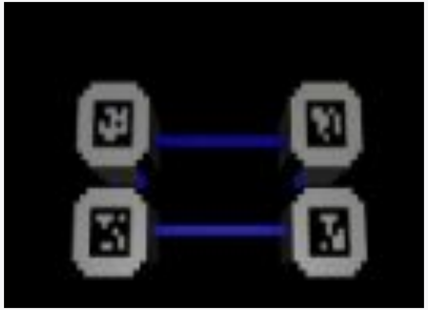
STATE ESTIMATION

Simulation: Ground truth state is always given
Testing: Assume SDK estimation is correct

02

GATEWAY AREA

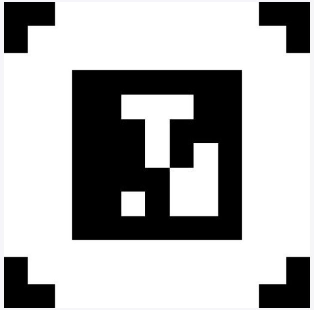
Gateway dimensions are **0.95 x 0.45 m** with an area of approximately **0.475 m^2**



03

ARUCO IN VIEW

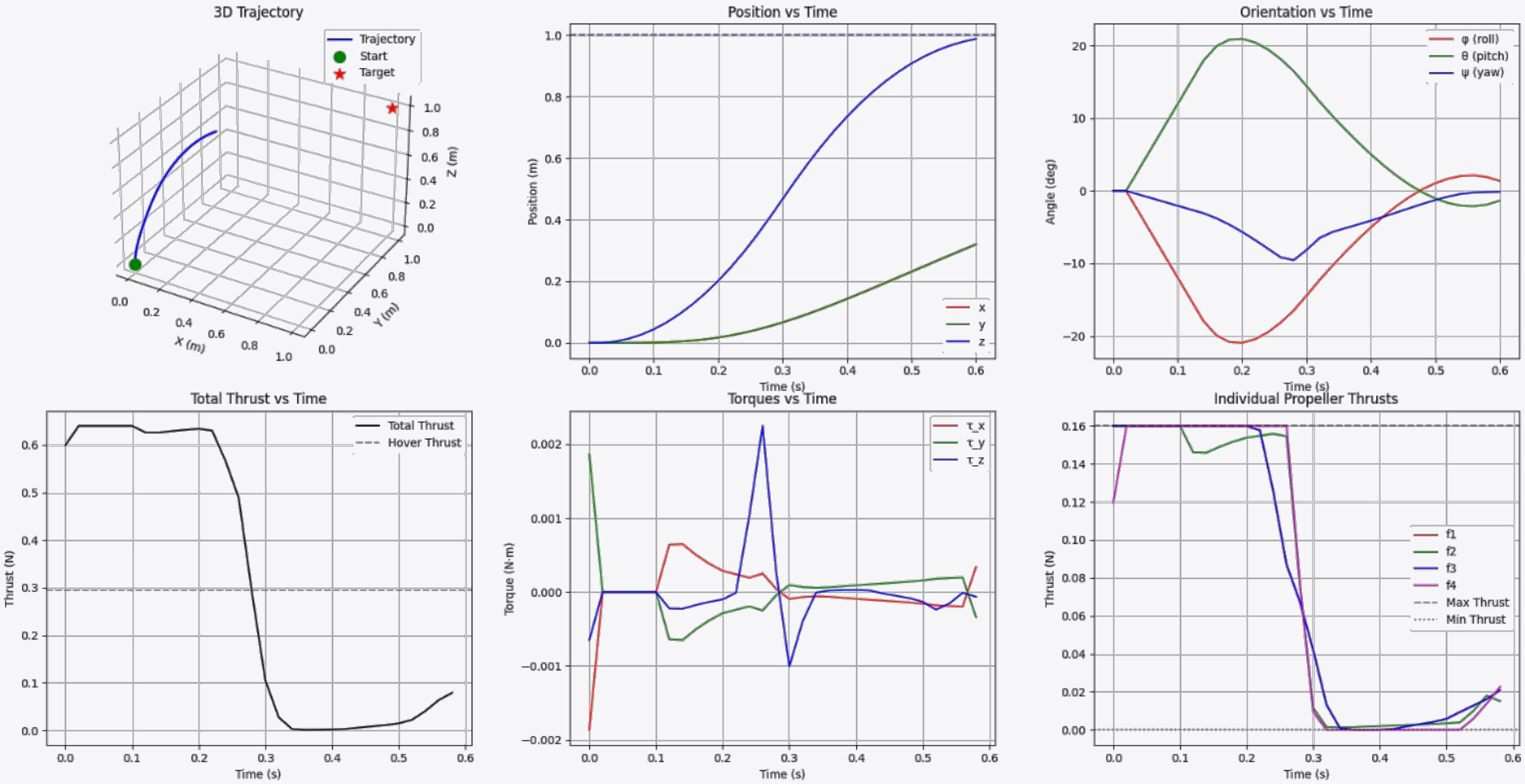
Takeoff Sequence starts the drone in a location where the tags are detectable



04

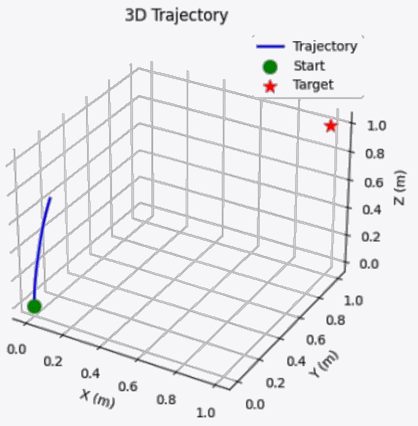
OPTIMIZATION

SIMULATION OPTIMIZATION



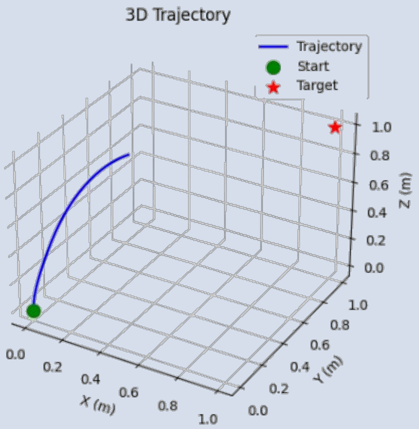
SIMULATION OPTIMIZATION

Parameter: Horizon Length



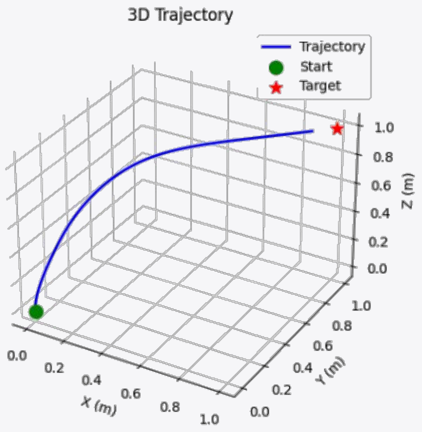
Horizon = 20 Steps

0.4 Seconds



Horizon = 30 Steps

0.6 Seconds



Horizon = 60 Steps

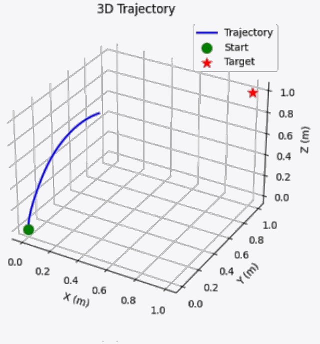
1.2 Seconds

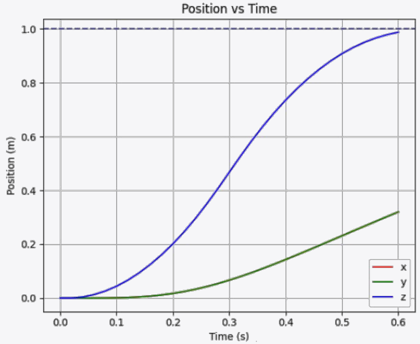
SIMULATION OPTIMIZATION

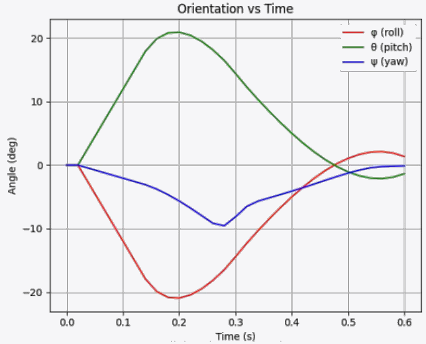
Parameter: QRP Weights

Q Cost Values:

X position = **100**
Y position = **100**
Z position = **1000**

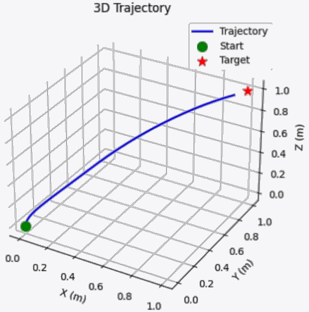


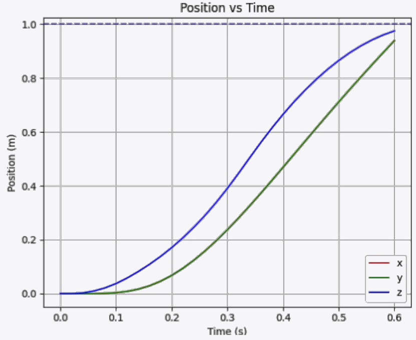


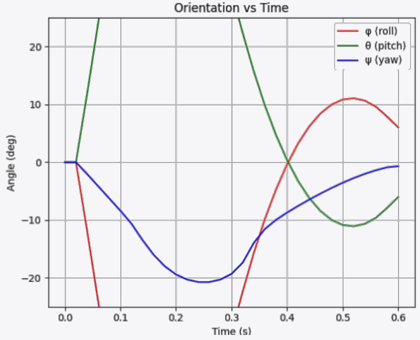


Q Cost Values:

X position = **1000**
Y position = **1000**
Z position = **1000**







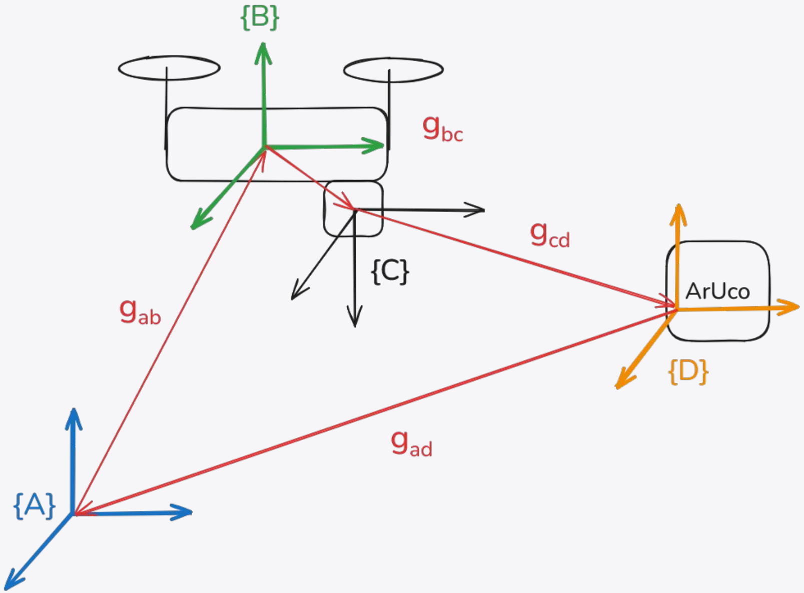
Computer Vision and Mapping

Objective:

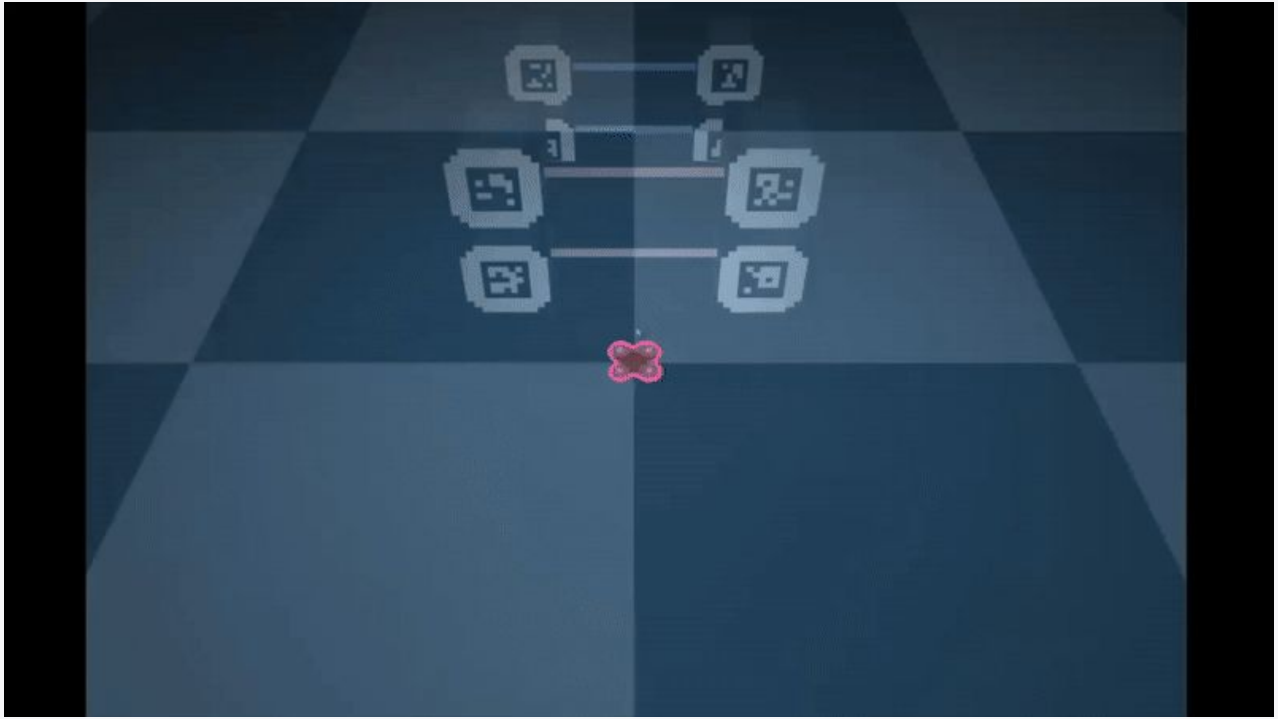
- Find gate position in world frame using FPV camera

Approach: openCV ArUco marker Detection

- PnP solver determines camera-aruco transform
- Coordinates combined to yield gate center in camera frame
- Transformed to world frame via g_{cb} and g_{ab}



MuJoCo Simulator Results



05

HARDWARE SELECTION

Drone 1: JY08 Pro

Drone Controls

Drone controlled through wifi commands sent over mobile app

- Pipeline commands could be sent with an Android emulator
- However takeoff sequence could not be emulated properly

WireShark Interception

Tried using WireShark to intercept and replicate control packages, but the process was time-consuming and fell far out of scope

Drone 2: DJI Tello

SDK Control

Chosen because the official Tello Software Development Kit (SDK) at least allows us to autonomously control the drone to some degree.

Limitations

The Tello SDK provides only high-level commands, not low-level motor control

PROPELLER

No access to individual propeller thrusts

COMMANDS

Can only give translational and rotational yaw commands, e.g. “move forward one meter” or “turn 90° clockwise”

Adapting our Controller

Workaround

- Directly send the trajectory commands to Tello's SDK
- Modify optimization to plan an easy path for the Tello drone to follow.
- Approximate optimal trajectory into simple commands that can be send as SDK commands

Adjusting Cost Function Weights

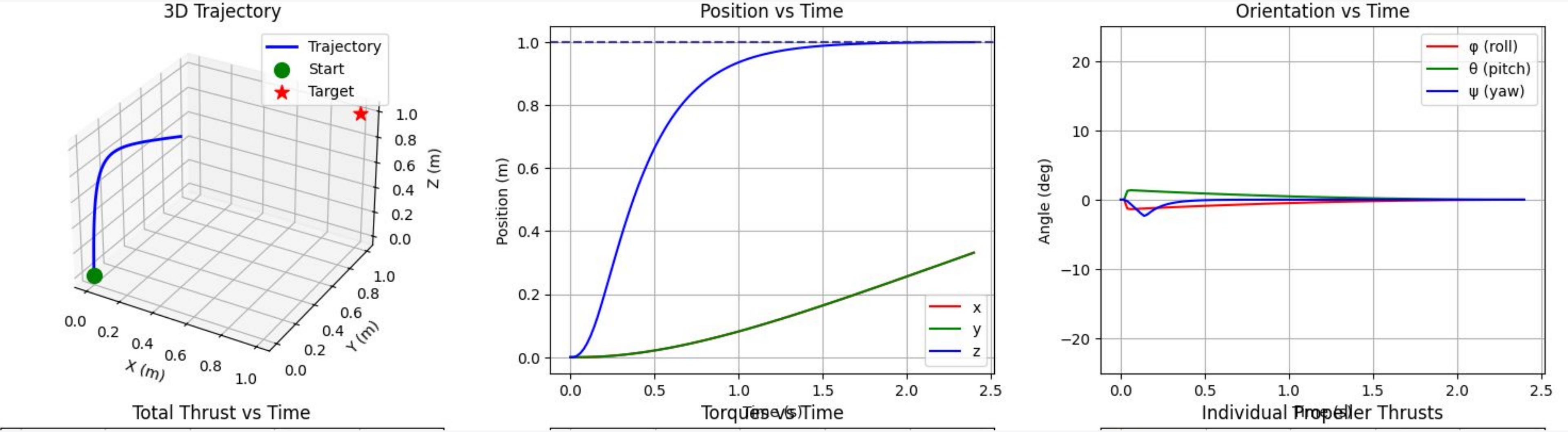
For Tello Hardware

Penalize roll and pitch angles heavily by increasing its cost to favor straight, translational motion:

- Reduces aggressive banking that the Tello cannot control
- Produces smoother, less curved trajectories that align with position-only commands

Slower, but as fast as we can go with allowed Tello functions

Adjusting Hardware QRP Weights



Video Demo

Next Steps for the Project

Obstacle detection without ArUco tags

Move toward fully vision-based gate and tunnel recognition.

Upgrade to hardware with direct motor control

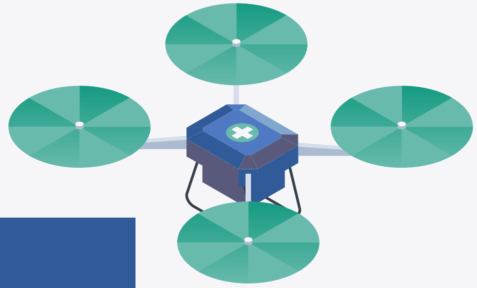
Enable thrust-level MPC instead of position-only commands.

Design + Test More Challenging Race Environments

Curved gates, tunnels, tighter constraints, dynamic elements.

Integrate SLAM

Integrate SLAM for advanced localization and mapping



Thank You!

Questions/Comments