UCLA Samueli School of Engineering

Quaternion based trajectory tracking quadrotor controller

Nikunj Sanghai and Shubham Kiran Wani

MAE 271D: Special Topics in Dynamics Systems Control

Date: 29th Nov, 2022

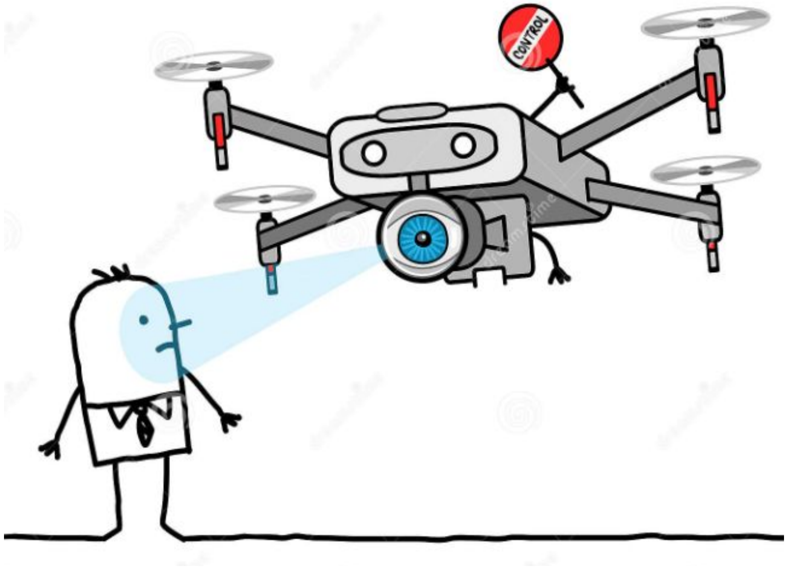
UCLA Mechanical and Aerospace Engineering

Contents

Introduction/Problem Setup

Motivation:

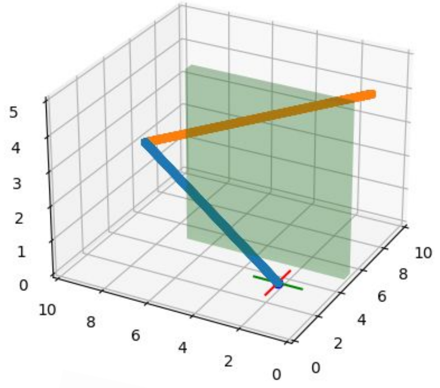
- Navigate autonomous drone in a structured environment such as warehouse to read barcodes on boxes in shelves
- Static obstacle avoidance using RRT* path planning(not implemented)
- Dynamic obstacle avoidance is beyond the scope of this controller.



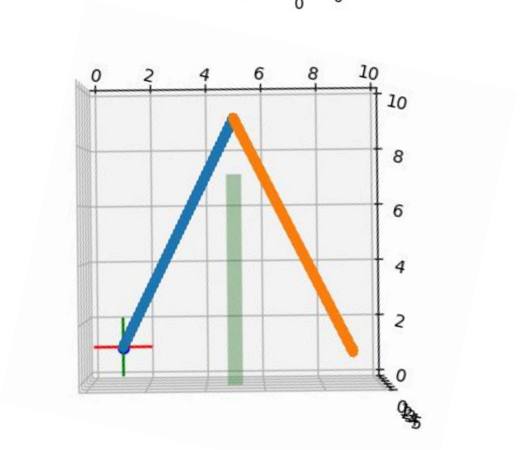
Trajectory Generation

Details of the Warehouse environment:

- The setup is 10x10x5 [m]
- Start Point: (1,1,1) [m]
- Goal Point: (9,1,4) [m]
- Static Obstacle: Warehouse stack 1x8x5 [m]



Goal: Quadcopter should be able to travel from start to end point with minimal control input for following the specified trajectory.



X_start → X_waypoint → X_goal

Static object collision avoidance is achieved

ASSUMPTION: The environment is well known and all data is available. The uncertainty in the system is minimal.

Piecewise Polynomial, Min Jerk Trajectory

Minimum Jerk Trajectory

Design a trajectory $x(t)$ such that $x(0) = a$, $x(T) = b$

$$x^\star(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T \mathcal{L}(\dddot{x}, \ddot{x}, \dot{x}, x, t)\, dt$$

$$\mathcal{L} = (\dddot{x})^2$$

Euler-Lagrange:

$$\boxed{\frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) + \frac{d^2}{dt^2}\left(\frac{\partial \mathcal{L}}{\partial \ddot{x}}\right)}^{0} \boxed{- \frac{d^3}{dt^3}\left(\frac{\partial \mathcal{L}}{\partial x^{(3)}}\right)} = 0$$

$$x^{(6)} = 0$$

$$x = c_5 t^5 + c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0$$

Penn Engineering

Solving for Coefficients

$$x = c_5 t^5 + c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0$$

Boundary conditions:

	Position	Velocity	Acceleration
$t = 0$	a	0	0
$t = T$	b	0	0

Solve:

$$\begin{bmatrix} a \\ b \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ T^5 & T^4 & T^3 & T^2 & T & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5T^4 & 4T^3 & 3T^2 & 2T & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 20T^3 & 12T^2 & 6T & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix}$$

Penn Engineering

<u>2 polynomials, one for each segment,</u>

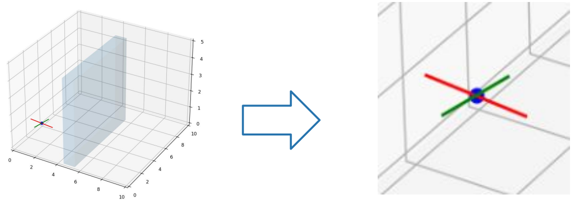
Simulation Approach

Platform - Python with matplotlib

1. Linear algebra with numpy arrays
2. Easy 3D visualizations for drone state and orientation

arenaviz.py

1. Creates point mass for COM(X)
2. Cross (X) to show orientation(q)



quatfunc.py

1. Quaternion is a (4,) ndarray
2. Implements product, conjugate and norm operations

$$q \in \mathbb{H}; \bar{q} \in \mathbb{R}^3; q_0 \in \mathbb{R}$$
$$r \in \mathbb{H}; \bar{r} \in \mathbb{R}^3; r_0 \in \mathbb{R}$$
$$q \otimes r = (q_0 r_0 - \bar{q}.\bar{r}) + (r_0 \bar{q} + q_0 \bar{r} + \bar{q} \times \bar{r})$$

Control Analysis

Known Trajectory

$$\ddot{\vec{r}}_d = \begin{bmatrix} \sum a_i * t_i \\ \sum b_i * t_i \\ \sum c_i * t_i \end{bmatrix} \qquad \psi_d(t) = 0 \forall t$$

$$T_d = m||\ddot{\vec{r}}_d + g|| \Rightarrow \vec{T}_d = \begin{bmatrix} 0 \\ 0 \\ m||\ddot{\vec{r}}_d + g|| \end{bmatrix}$$

Desired thrust and orientation

$$q_d = \frac{1}{\sqrt{2(1 + \hat{T}_d^T \hat{F}_I))}} \begin{bmatrix} 1 + \hat{T}_d^T \hat{F}_I \\ \hat{T}_d \times \hat{F}_I \end{bmatrix} \quad \text{where } \vec{F}_I = m(\ddot{\vec{r}}_d + g)$$

$$\begin{bmatrix} \omega_{yd} \\ -\omega_{xd} \\ 0 \end{bmatrix} = \frac{m}{T_d} g_d^* \otimes \vec{r}_d^{\,3} \otimes q_d - \frac{\dot{\vec{T}}_d}{T_d}$$

Control Analysis

Error Calculation

$$q_e = q_d^* \otimes q$$

$$\vec{\omega_e} = \vec{\omega} - q_e^* \otimes \vec{\omega_d} \otimes q_e$$

$$x_e = x - x_d$$

Moment Calculation

$$\text{If } \vec{s} = \vec{\omega_e} + \lambda \text{sgn}(\dot{q_e})\vec{q_e} \text{ Then}$$

$$\vec{M_B} = \vec{\omega} \times J\vec{\omega} + J\dot{\vec{\omega_r}} - \lambda J \text{sgn}(\dot{(q_e)})\dot{\vec{q_e}} - KJ\vec{s}$$

Thrust Calculation

$$\text{If we set } g \otimes \vec{T} \otimes g^* = m\vec{u}$$

$$m\ddot{\vec{r}} = \vec{F_r} + m\vec{u} \Rightarrow \ddot{\vec{r}} = \vec{a_I} + \vec{u}$$

$$\vec{u} = \ddot{\vec{r}} - \vec{a_I} - K_p\vec{r_e} - K_d\dot{\vec{r_e}}$$

Control Analysis

System Dynamics

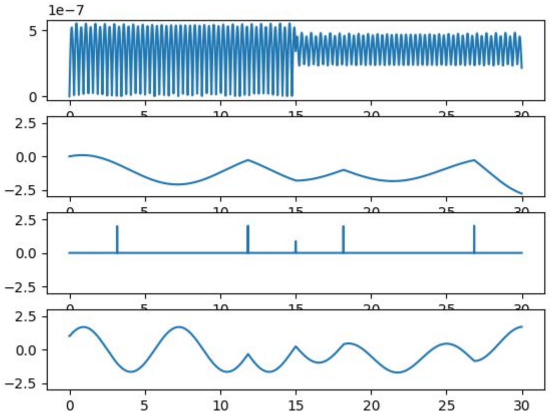
$$\dot{x} = \frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ q \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{x} \\ q \otimes \frac{T}{m} \otimes q^* + \bar{g} \\ \frac{1}{2} q \otimes \omega \\ J^{-1}(M_b - \omega \times J\omega) \end{bmatrix}$$

Update state variables using Euler's method

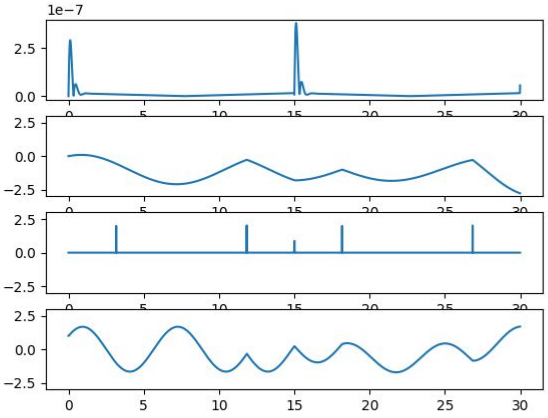
$$Y_n = Y_{n-1} + hF\,(X_{n-1}, Y_{n-1})$$

Controller

Ideal System: No uncertainty



Kp=100, K_d=0, lambda=0, K=0



Kp=100, K_d=10, lambda=0, K=0

a) Position error norm [m]

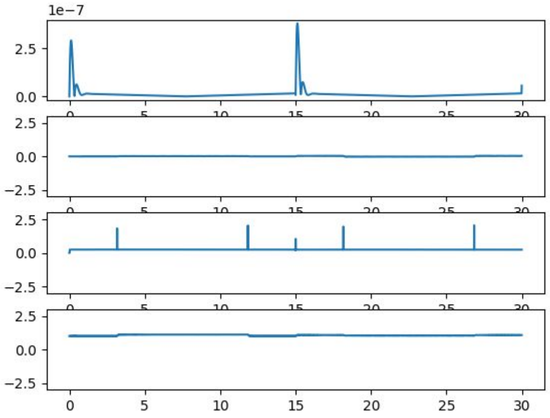
b) Quaternion error - 1

c) Applied Moment [Nm]

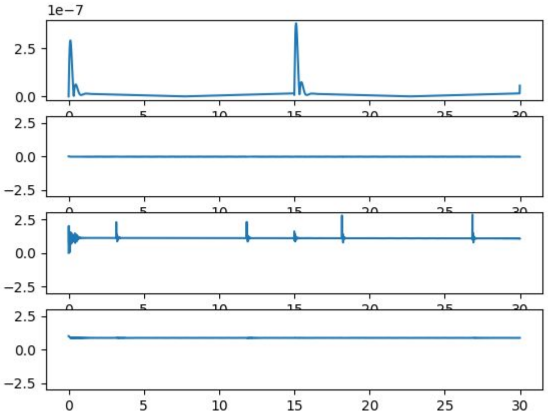
d) Applied Thrust [N]

Controller

Ideal System: No uncertainty



Kp=100, K_d=10, lambda=5, K=0



Kp=100, K_d=10, lambda=20, K=20

a) Position error norm [m]

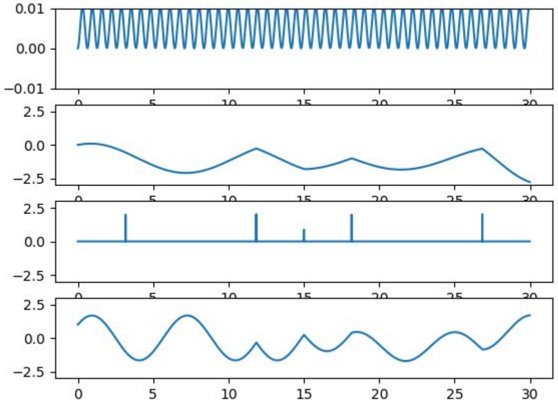
b) Quaternion error - 1

c) Applied Moment [Nm]

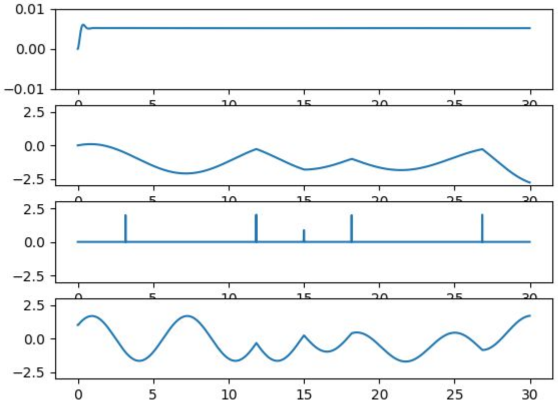
d) Applied Thrust [N]

Controller

Non-ideal System with 5% error



Kp=100, K_d=0, lambda=0, K=0



Kp=100, K_d=10, lambda=0, K=0

a) Position error norm [m]

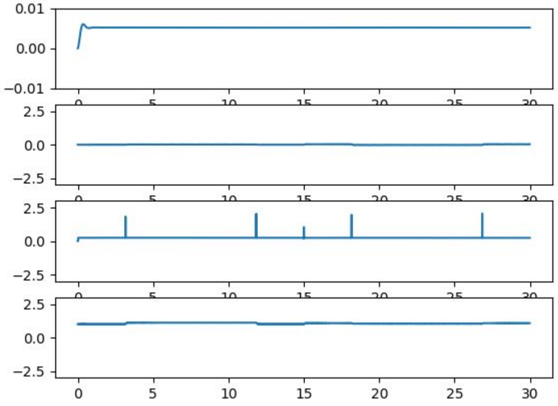
b) Quaternion error - 1

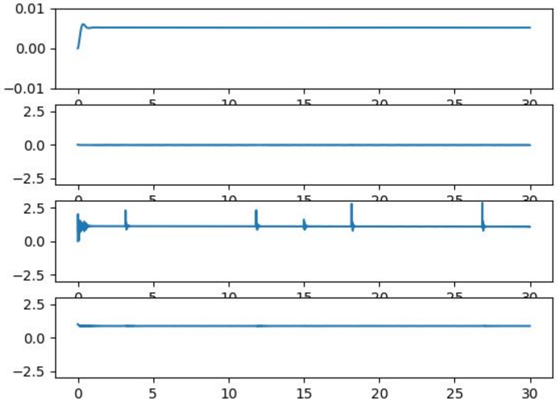
c) Applied Moment [Nm]

d) Applied Thrust [N]

Controller

Non-ideal System with 5% error





a) Position error norm [m]

b) Quaternion error - 1

c) Applied Moment [Nm]

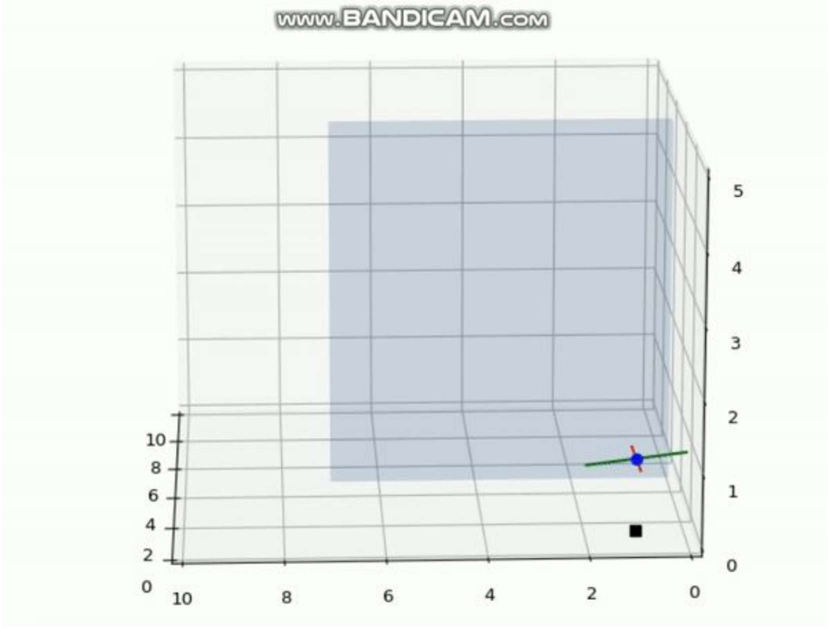
d) Applied Thrust [N]

Kp=100, K_d=10, lambda=5, K=0

Kp=100, K_d=10, lambda=20, K=20

Demonstration

https://github.com/shubhamwani376/MPC_Quadcopter



Future Work

Trajectory Generation:

- RRT*/ Dijkstra's algorithm to calculate near optimal trajectory for the quadrotor.
- Comparative study with MPC created trajectory and RRT* /Dijkstra's algorithm to analyze the difference in minimal control input.
- Dynamic obstacle avoidance with potential fields MPC

Control:

- Minimal control input MPC implementation as a controller.

We plan to make progress on this project after the quarter, collaborators are invited.

References

1. Brett. T. Lopez, Fall 2022 MAE271D, Class Notes.
2. D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 2520-2525, doi: 10.1109/ICRA.2011.5980409
3. K. Choutri, M. Lagha, L. Dala and M. Lipatov, "Quadrotors trajectory tracking using a differential flatness-quaternion based approach," 2017 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO), 2017, pp. 1-5, doi: 10.1109/ICMSAO.2017.7934901.
4. Parwana, Hardik, Jay S. Patrikar, and Mangal Kothari. "A novel fully quaternion based nonlinear attitude and position controller." 2018 AIAA Guidance, Navigation, and Control Conference. 2018.

UCLA Samueli School of Engineering

Q&A