Trajectory Generation and Control for Quadrotors

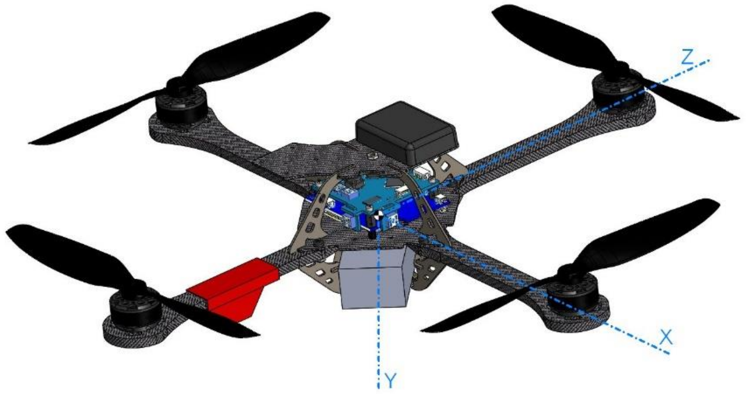
Sandesh Thapa

Department of Mechanical & Aerospace Engineering

Oklahoma State University

Introduction

- Quadrotors are popular research because of their agility and mobility in 3D space
- For a quadrotor maneuvering in tightly constraint environment, it is necessary to develop optimal trajectories to ensure safe passage through corridors and constraints [1].
- Develop flight plans that leverage the dynamics of system [1].



[1] Mellinger, Daniel, and Vijay Kumar. "Minimum snap trajectory generation and control for quadrotors." Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011.
Image source: http://wiki.asctec.de/display/AR/CAD+Models

Objective

- Review of control and trajectory generation for quadrotors [1].
- Understand and implement differential flatness for trajectory generation
- Examples code given at [2, 3].
- Implement minimum snap trajectory and control for quadrotors based on flatness [1, 3].

[1] Mellinger, Daniel, and Vijay Kumar. "Minimum snap trajectory generation and control for quadrotors." Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011.
[2] https://github.com/WenjinTao/Robotics--Coursera/tree/master/Robotics-Aerial_Robotics
[3] https://www.coursera.org/learn/robotics-flight

Dynamic Model

r

$$q = \begin{bmatrix} x \\ y \\ z \\ \varphi \\ \theta \\ \psi \end{bmatrix}$$

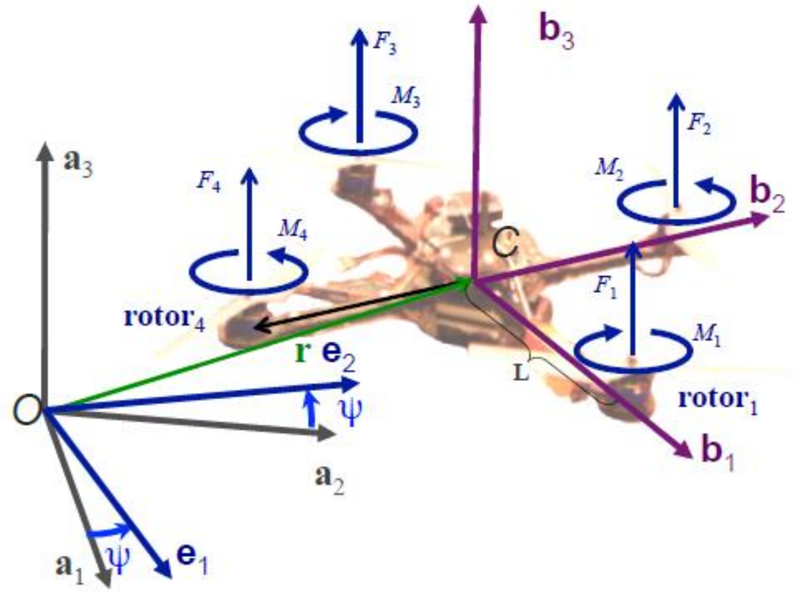
Roll

Pitch

Yaw

Angular velocity in body frame

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$



$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R_{BA} \begin{bmatrix} 0 \\ 0 \\ \boxed{F_1 + F_2 + F_3 + F_4} \end{bmatrix}$$

u_1

Newton's Equation of motion

u_2

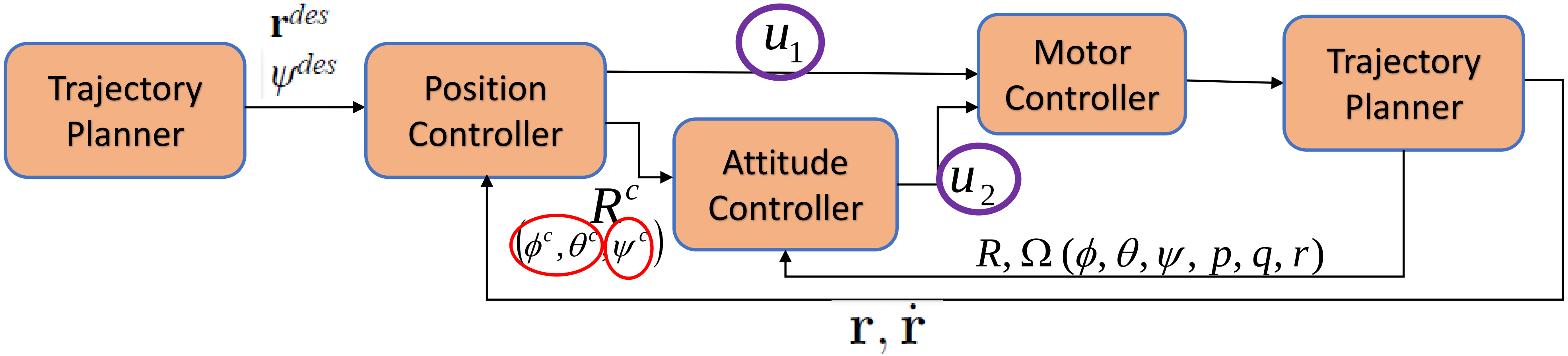
$$I \begin{bmatrix} \dot{q} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \boxed{\begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix}} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Euler's Equation of motion

$$R_{BA} = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix}$$

Source: N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, The GRASP Multiple Micro-UAV Testbed," IEEE Robotics and Automation Magazine, 2010.

Control



commanded

actual (feedback)

$$(\ddot{r}_{i,des} - \ddot{r}_{i,c}) + k_{d,i}(\dot{r}_{i,des} - \dot{r}_i) + k_{p,i}(r_{i,des} - r_i) = 0$$

specified

$$\boxed{u_1 = m(g + \ddot{r}_{3,c})}$$

$$\phi_c = \frac{1}{g}(\ddot{r}_{1,c}\sin\psi_{des} - \ddot{r}_{2,c}\cos\psi_{des})$$

$$\theta_c = \frac{1}{g}(\ddot{r}_{1,c}\cos\psi_{des} - \ddot{r}_{2,c}\sin\psi_{des})$$

$$\psi_c = \psi^{des}$$

$$u_2 = \begin{bmatrix} k_{p,\phi}(\phi_c - \phi) + k_{d,\phi}(p_c - p) \\ k_{p,\theta}(\theta_c - \theta) + k_{d,\theta}(q_c - q) \\ k_{p,\psi}(\psi_c - \psi) + k_{d,\psi}(r_c - r) \end{bmatrix}$$

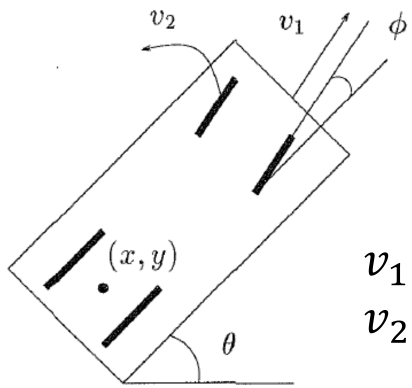
Source: N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, The GRASP Multiple Micro-UAV Testbed," IEEE Robotics and Automation Magazine, 2010.

Differential Flatness

- **Definition:** A non linear system is *differentially flat* if we can find a set of outputs such that we can express all states and inputs in terms of those outputs and their derivatives [1].
- **Example:** Kinematic car

Equations of Motion

$$\boxed{\begin{aligned} \dot{x} &= \cos\theta \cos\phi \ v_1 \\ \dot{y} &= \sin\theta \cos\phi \ v_2 \\ \dot{\theta} &= \frac{1}{l}\sin\phi \ v_1 \\ \dot{\phi} &= v_2 \end{aligned}}$$



v_1 : forward velocity
v_2 : steering velocity

- If output is (*x*, *y*), ϕ can be expressed in terms of flat outputs as

$$\phi = \arctan\left(\frac{1}{l}(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}, \dot{x}\ddot{x} - \dot{y}\ddot{y}\right)$$

[1] M. J. Van Nieuwstadt and R. M. Murray, "Real-time trajectory generation for differentially flat systems," International Journal of Robust and Nonlinear Control, vol. 8, pp. 995–1020, 1998

3D Trajectory Generation

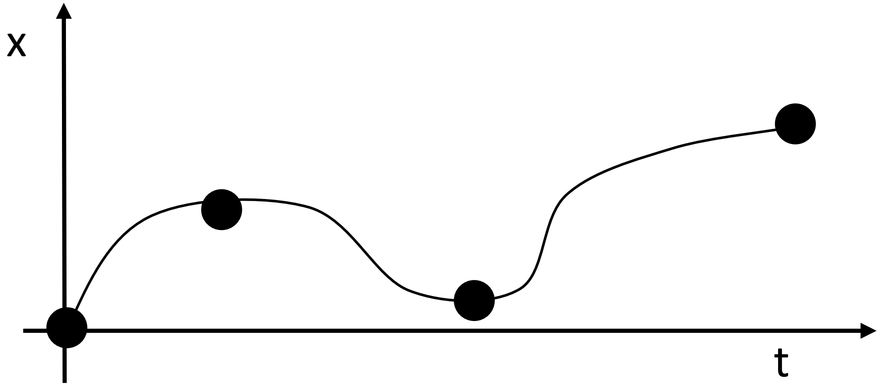
- Optimal Trajectory:

$$x^*(t) = \underset{x(t)}{\arg\min} \int_0^T L(\dot{x}, x, t)\,dt \qquad L : functional$$

- Smooth Trajectories:

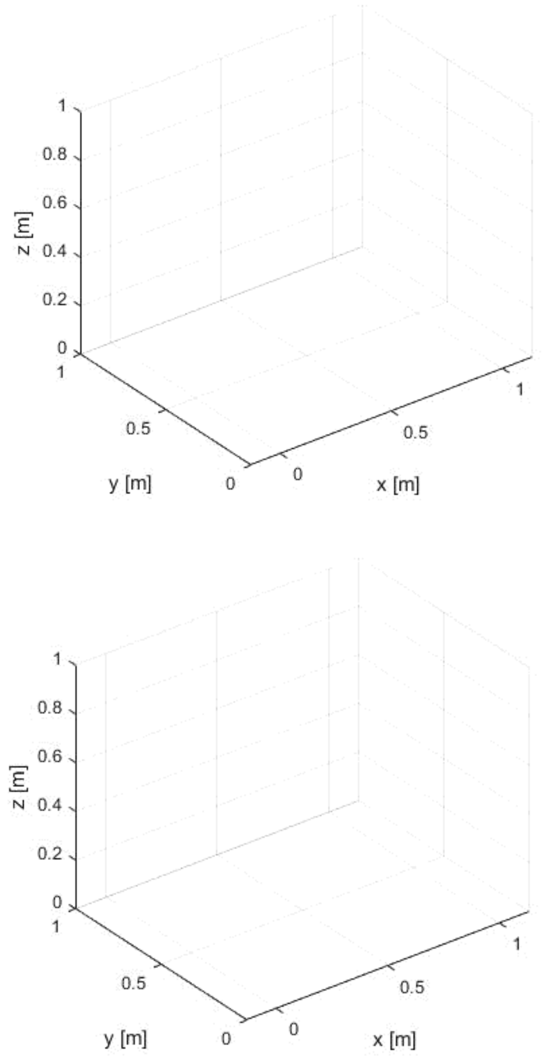
$$x^*(t) = \underset{x(t)}{\arg\min} \int_0^T \left(x^{(n)}\right)^2 dt$$

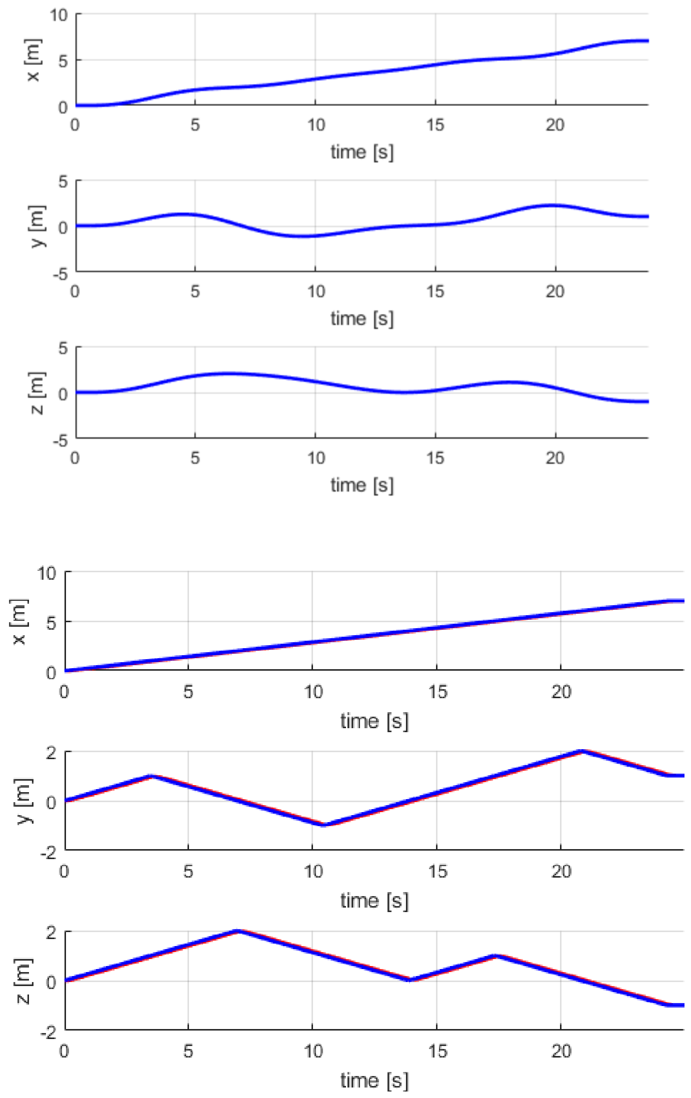
  - n = 1, shortest distance (velocity)
  - n = 2, minimum acceleration
  - n = 3, minimum jerk
  - n = 4, minimum snap
- Constraints:
  - Must pass through way points:
  - Velocity, Acceleration, Jerk are zero at the end points
  - Velocity, acceleration, 3rd to 6th derivatives are continuous
  - Convert the equation into matrix for $A\alpha = b$ and solve for a



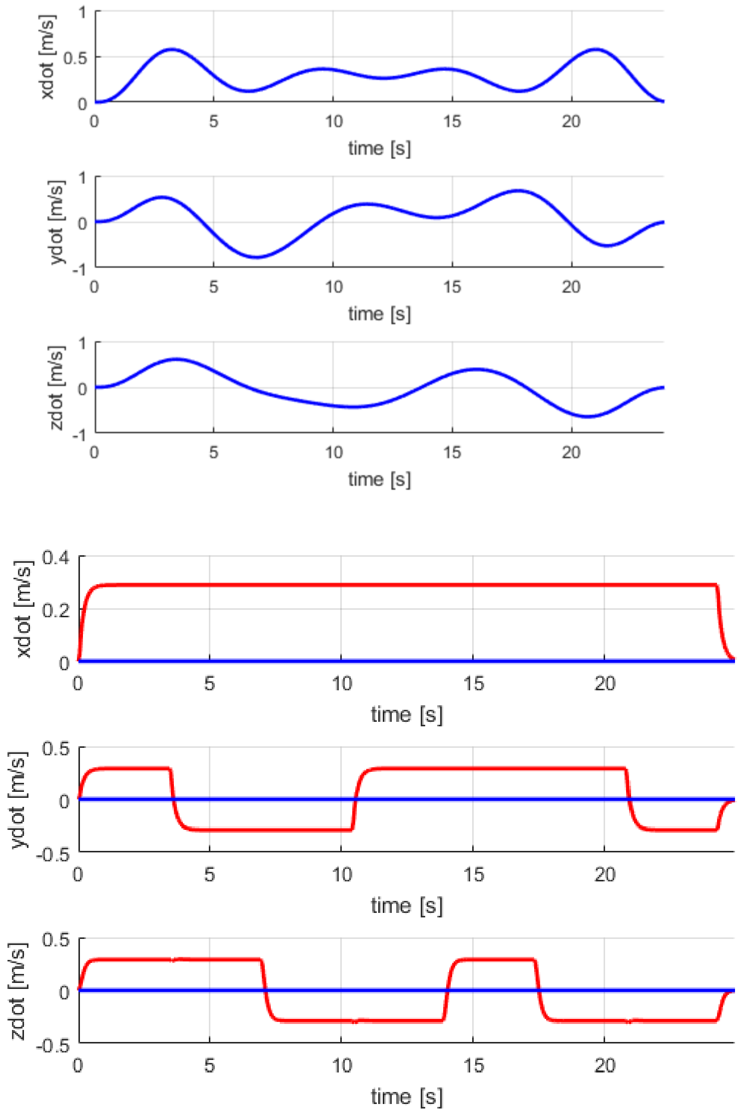
$$x(t) = C_0 + C_1 t + C_2 t^2 + C t^3 + C_4 t^4 + C_5 t^5 + C_6 t^5 + C_7 t^7$$

Simulation Results





Quad Position



Quad Velocity

Conclusion & Future Work

- Implemented minimum snap trajectory and control for quadrotors
- The trajectories are optimal
- Can navigate based on ways points with smooth trajectories

- Can be extended to motion planning for precise and aggressive maneuvers
- Implement in experimental platform
- If possible extend to multi-agents

Questions?