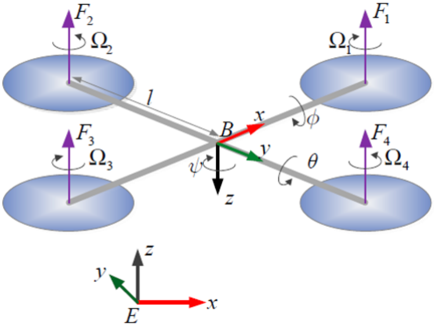
Multi-UAVs reference tracking using MPC

Mohammad Alikhani
Universitat Politècnica de Catalunya

April 23, 2025

Contents

UAV nonlinear Model

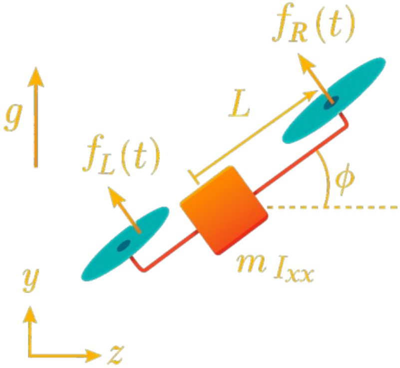


$$\begin{bmatrix} \ddot{\phi}_m \\ \ddot{\theta}_m \\ \ddot{\psi}_m \\ \ddot{z}_m \\ \ddot{x}_m \\ \ddot{y}_m \end{bmatrix} = \begin{bmatrix} l\kappa_{xx}\Omega_m^2 + \frac{1}{I_{xx}}U_2 \\ l\kappa_{yy}\Omega_m^2 + \frac{1}{I_{yy}}U_3 \\ l\kappa_{zz}\Omega_m^2 + \frac{1}{I_{zz}}U_4 \\ \frac{1}{m}(\cos\phi_m \cos\theta_m)U_1 - g \\ \frac{U_1}{m}(\cos\phi_m \sin\theta_m \cos\psi_m + \sin\phi_m \sin\psi_m) \\ \frac{U_1}{m}(\cos\phi_m \sin\theta_m \sin\psi_m - \sin\phi_m \cos\psi_m) \end{bmatrix}$$

SOM: 2D Quadrotor model

- **System Overview:**
  - 6 states: $x = \left[x, \dot{x}, y, \dot{y}, \theta, \dot{\theta}\right]^T$
  - 2 inputs: $u = \left[u_1, u_2\right]^T$
    left and right motor thrusts
- **Nonlinear Dynamics:**

$$
\begin{aligned}
\dot{x}_1 &= x_2, & \dot{x}_2 &= \frac{1}{m}(u_1 + u_2)\sin(\theta) \\
\dot{x}_3 &= x_4, & \dot{x}_4 &= \frac{1}{m}(u_1 + u_2)\cos(\theta) - g \\
\dot{x}_5 &= x_6, & \dot{x}_6 &= \frac{l}{I}(u_1 - u_2)
\end{aligned}
$$



SOM: 2D Quadrotor model

- **State variables:**
  - $x_1 = x$: Horizontal position.
  - $x_2 = \dot{x}$: Horizontal velocity.
  - $x_3 = y$: Vertical position.
  - $x_4 = \dot{y}$: Vertical velocity.
  - $x_5 = \theta$: Pitch angle (orientation).
  - $x_6 = \dot{\theta}$: Angular velocity.

- **Inputs:**
  - u_1: Thrust from the left motor.
  - u_2: Thrust from the right motor.
- **Parameters:**
  - m: Mass of the quadrotor.
  - g: Gravitational acceleration.
  - l: Distance from the center of mass to each motor.
  - I: Moment of inertia about the pitch axis.

COM: 1.Equilibrium Point (Hover)

- **Position and velocity:**
  $x = 0,\ \dot{x} = 0,\ y = 0,\ \dot{y} = 0.$
- **Angle and angular velocity:** $\theta = 0,\ \dot{\theta} = 0.$
- **Thrusts:** The net force must balance gravity, and the torque must be zero.
- $\theta = 0$, so $\cos(\theta) = 1$, $\sin(\theta) = 0$.
- $\ddot{y} = 0$, so:

$$\frac{1}{m}(u_1 + u_2)\cos(0) - g = 0 \quad \Rightarrow \quad u_1 + u_2 = mg$$

- $\ddot{\theta} = 0$,

$$u_1 - u_2 = 0 \quad \Rightarrow \quad u_1 = u_2 u_1 + u_2 = mg, \quad u_1 = u_2 \quad \Rightarrow \quad u_1 = u_2 = \frac{mg}{2}$$

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{1}{m}(u_1 + u_2)\sin(x_5) \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \frac{1}{m}(u_1 + u_2)\cos(x_5) - g \\
\dot{x}_5 &= x_6 \\
\dot{x}_6 &= \frac{l}{I}(u_1 - u_2)
\end{aligned}$$

COM: 2.Computing Jacobian

$$A = \left.\frac{\partial f}{\partial x}\right|_{(x_e,u_e)}, \quad B = \left.\frac{\partial f}{\partial u}\right|_{(x_e,u_e)}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{m}(u_1+u_2)\cos(x_5) & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{m}(u_1+u_2)\sin(x_5) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ \frac{\sin(x_5)}{m} & \frac{\sin(x_5)}{m} \\ 0 & 0 \\ \frac{\cos(x_5)}{m} & \frac{\cos(x_5)}{m} \\ 0 & 0 \\ \frac{l}{I} & -\frac{l}{I} \end{bmatrix}$$

- **In Equilibrium point**

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{g}{m} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m} & \frac{1}{m} \\ 0 & 0 \\ \frac{l}{I} & -\frac{l}{I} \end{bmatrix}$$

- **Prediction Horizon:** $N = 20$, with sampling time $T_s = 0.1$ s.
- Total prediction time: $N \cdot T_s = 20 \cdot 0.1 = 2$ seconds.

Objective Function

$$J = \sum_{k=1}^{N} \left((r_k - Cx_k)^T Q (r_k - Cx_k) + u_k^T R_k u_k + \Delta u_k^T S \Delta u_k + x_k^T Q_{\text{state}} x_k \right)$$

- **Tracking:** $(r_k - Cx_k)^T Q (r_k - Cx_k), \quad Q = 5 \cdot I_{3\times 3}$
- **Control Effort:** $R_k = I + w \cdot x_{0.5}^2 \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad w = 10$
- **Smoothness:** $\Delta u_k^T S \Delta u_k, \quad S = 5 \cdot I_{2\times 2}$
- **State Penalty:** $Q_{\text{state}} = \text{diag}(0, 2, 0, 2, 0, 5)$ (penalizes $\dot{x}, \dot{y}, \dot{\theta}$)

- **Constraints:**
  - **Inputs:** $(0 \leq u_k(1), u_k(2) \leq mg, \quad u_k(1) + u_k(2) \leq mg \cdot (1 - \alpha |x_{0,5}|), \quad \alpha = 5$
  - **States:** $(-10 \leq x, y \leq 10, \quad 5 \leq \dot{x}, \dot{y} \leq 5, \quad 0.1 \leq \theta \leq 0.1, \quad -2 \leq \dot{\theta} \leq 2$
  - **Torque:** $(\left| \frac{l}{I}(u_k(1) - u_k(2)) \right| \leq 2$

Cost Function Terms(2/1)

- **Tracking Error:** $(r_k - Cx_k)^T Q (r_k - Cx_k), \quad Q = 5 \cdot I_{3\times 3}.$
  - Ensures x, y, θ track references r_x, r_y, r_θ
  - High Q prioritizes accurate position and angle

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

- **Control Effort:** $u_k^T R_k u_k, \quad R_k = I + w \cdot x_{0.5}^2 \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad w = 10$
  - Penalizes large thrusts u_1, u_2
  - Dynamic R_k: Reduces thrust when $|\theta|$ is large, ensuring cautious speed

Cost Function Terms (2/2)

- **Smoothness:** $\Delta u_k^T S \Delta u_k, \quad S = 5 \cdot I_{2\times 2}, \quad \Delta u_k = u_k - u_{k-1}$
  - Minimizes rapid thrust changes, reducing energy and actuator wear
- **State Penalty:** $x_k^T Q_{\text{state}} x_k, \quad Q_{\text{state}} = \text{diag}(0, 2, 0, 2, 0, 5)$
  - Penalizes high velocities $(\dot{x}, \dot{y})$ and angular velocity $(\dot{\theta})$
  - Reduces energy (via drag) and stabilizes motion

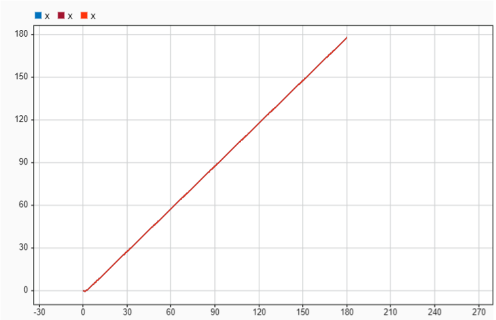
Constraints

- **Inputs:**
  - $0 \leq u_k(1), u_k(2) \leq mg, \quad m \cdot g = 49.05\,\text{N}$
    - Ensures thrusts are positive and within motor limits
  - $u_k(1) + u_k(2) \leq m \cdot g \cdot (1 - \alpha \cdot |x_{0,5}|), \quad \alpha = 5$
    - Reduces total thrust when $|\theta|$ is large, ensuring cautious speed
    - Example: At $|\theta| = 0.1$, limit is 24.525 N
  - $\left|\frac{l}{I}(u_k(1) - u_k(2))\right| \leq 2$
    - Limits torque, reducing energy-intensive angular acceleration
- **States:**
  - $-10 \leq x, y \leq 10$ m: Defines workspace
  - $-5 \leq \dot{x}, \dot{y} \leq 5$ m/s: Limits speed, reducing drag
  - $-0.1 \leq \theta \leq 0.1$ rad: Keeps linearization valid (e.g., $\sin(\theta) \approx \theta$)
  - $-2 \leq \dot{\theta} \leq 2$ rad/s: Limits angular velocity, stabilizing orientation

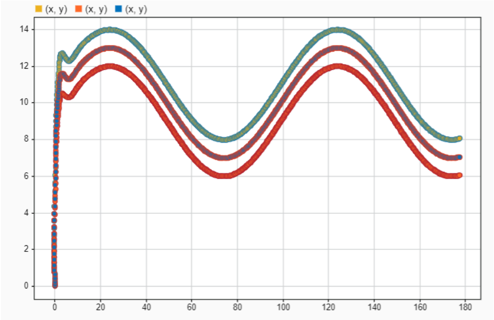
DMPC vs CMPC

Aspect	Decentralized MPC	Centralized MPC
Structure	Each quadcopter has its own MPC, optimizing locally	Single MPC optimizes for all quadcopters globally
Computation	Distributed: Smaller problems per quadcopter.	Centralized: Large problem with all states/inputs
Communication	Requires sharing states among quadcopters.	All data sent to a central controller
Scalability	Scales well: Add quadcopters with minimal impact.	Poor scalability: Problem size grows with number of quadcopters
Fault Tolerance	Robust: Failure of one quadcopter doesn't affect others.	Vulnerable: Central failure impacts all quadcopters
Performance	Suboptimal: Local optimization may miss global goals	Optimal: Global optimization for all quadcopters

Results

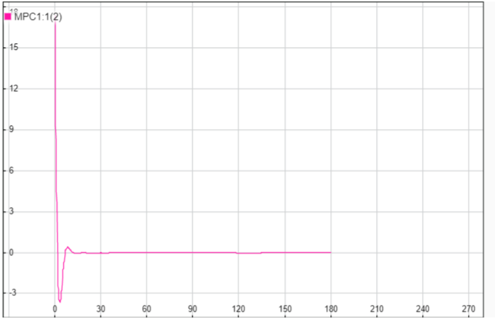


Lateral Displacement

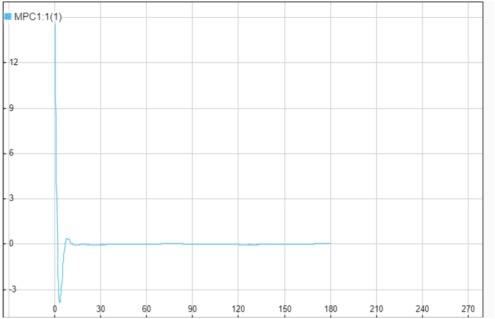


UAVs Positions

Results



U1 Control Action
very small after 30s



U2 Control Action
very small after 30s

Thanks For your Attention!