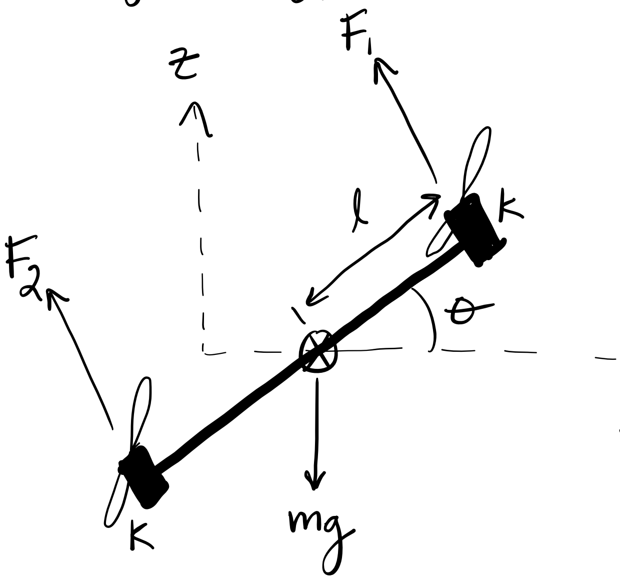
Planar drone example.

1. Continuous nonlinear dynamics
2. Controller design
3. Measurements

1. Continuous nonlinear dynamics

Consider a simplified planar model of a drone:
(no drag, no gyroscopic forces, no nonlinear aero effects)



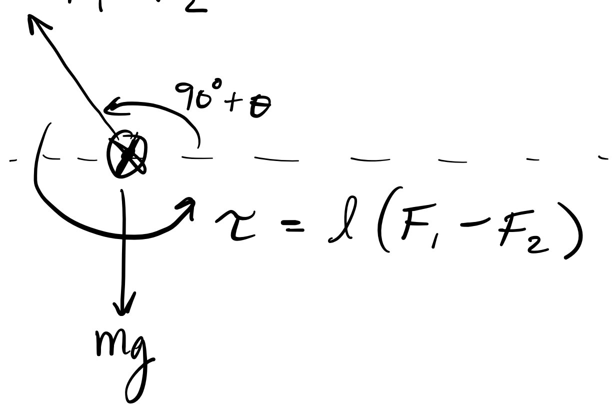
States: θ, $\dot{\theta}$, x, $\dot{x}$, z, $\dot{z}$

controls: u_1, u_2 : motor speed

parameters: m : mass (known)
k : $F_1 = k u_1$ (unknown)
l : length of arm (known)
I_{yy} : moment of inertia (known)

Rewrite:

$F = F_1 + F_2$



Dynamics:

$$I_{yy}\ddot{\theta} = \tau = lk(u_1 - u_2)$$

$$m\ddot{\vec{x}} = \vec{F} + \vec{mg}$$

$\to m\ddot{x} = -F\sin\theta$

$\to m\ddot{z} = F\cos\theta - mg$

$\cos(90+\theta) = -\sin(\theta)$

$\to m\ddot{x} = -k\sin\theta(u_1 + u_2)$

$\to m\ddot{z} = k(u_1 + u_2)\cos\theta - mg$

Rewrite contros: (to make simplify)

$j_1 = u_1 - u_2$

$j_2 = u_1 + u_2$

Continuous State Space

Write in form: $\dot{\vec{x}} = f(\vec{x}, \vec{u})$

$$\dot{\vec{x}} = \frac{d}{dt}\begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \\ z \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ \tau / I_{yy} \\ \dot{x} \\ -F\sin\theta/m \\ \dot{z} \\ (F\cos\theta - mg)/m \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ lk/I_{yy}\, j_1 \\ \dot{x} \\ -k\sin\theta/m\, j_2 \\ \dot{z} \\ (k\, j_2 \cos\theta - mg)/m \end{bmatrix}$$

Write in control affine form:

$$\dot{\vec{x}} = \frac{d}{dt}\begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \\ z \\ \dot{z} \end{bmatrix} = \underbrace{\begin{bmatrix} \dot{\theta} \\ 0 \\ \dot{x} \\ 0 \\ \dot{z} \\ -g \end{bmatrix}}_{f_0} + \underbrace{\begin{bmatrix} 0 \\ lk/I_{yy} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{f_1} j_1 + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ -k\sin\theta/m \\ 0 \\ k\cos\theta/m \end{bmatrix}}_{f_2} j_2$$

add the unknown parameter k as a static state:

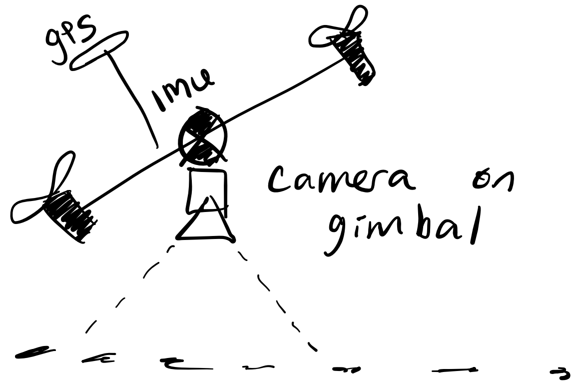
$$\dot{\vec{x}} = \frac{d}{dt}\begin{bmatrix} \theta \\ \dot\theta \\ x \\ \dot x \\ z \\ \dot z \\ k \end{bmatrix} = \overset{f_0}{\begin{bmatrix} \dot\theta \\ 0 \\ \dot x \\ 0 \\ \dot z \\ -g \\ 0 \end{bmatrix}} + \overset{f_1}{\begin{bmatrix} 0 \\ lk/I_{yy} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}} \dot{j}_1 + \overset{f_2}{\begin{bmatrix} 0 \\ 0 \\ 0 \\ -k\sin\theta/m \\ 0 \\ k\cos\theta/m \\ 0 \end{bmatrix}} \dot{j}_2$$

Send picture to AI (e.g. Claude) to get latex code, and python function.

Allows us to write dyamics as:

$$\dot{\vec{x}} = f(\vec{x}, \vec{u}) = f_0(\vec{x}) + f_1(\vec{x})j_1 + f_2(\vec{x})j_2$$

Measurements:



Consider 3 options:

$$\vec{y}_a = \begin{bmatrix} x \\ z \\ \theta \\ k \end{bmatrix} \begin{matrix} \text{— gps} \\ \text{— gps} \\ \text{— imu tilt sensor} \\ \text{— suppose we have a good model} \end{matrix}$$

$$\vec{y}_b = \begin{bmatrix} \dot{x}/z \\ \theta \end{bmatrix} \begin{matrix} \text{— optic flow from downward cam.} \\ \text{— imu tilt sensor} \end{matrix}$$

$$\vec{y}_c = \begin{bmatrix} \dot{x}/z \\ \theta \\ \ddot{x} \\ \ddot{z} \end{bmatrix} \begin{matrix} \text{— optic flow from downward cam.} \\ \text{— imu tilt sensor} \\ \text{— acceleration from imu} \\ \text{— acceleration from imu} \end{matrix}$$

$\ddot{x}$ & $\ddot{z}$ are not states, need to rewrite as states, based on dyamics eqns.

seperate controls: $\dot{\vec{x}} = f(\vec{x}) + f(\vec{x})u_1 + \cdots + f(\vec{x})u_n$

$$\dot{\vec{x}} = \frac{d}{dt}\begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \\ z \\ \dot{z} \end{bmatrix} = \underbrace{\begin{bmatrix} \dot{\theta} \\ 0 \\ \dot{x} \\ 0 \\ \dot{z} \\ -g \end{bmatrix}}_{f_0} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ -\sin\theta/m \\ 0 \\ \cos\theta/m \end{bmatrix}}_{f_1} F + \underbrace{\begin{bmatrix} 0 \\ I_{yy}^{-1} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{f_2} \tau_y$$

$$\dot{\vec{x}} = f_0(\vec{x}) + f_1(\vec{x})F + f_2(\vec{x})\tau_y$$