1 2D Quad rotor dynamics

In this section I will calculate the dynamic equations for the 2D simplified version of a quad rotor.

1.1 State Variables

We will have our ξ = (q[t], u[t])

$q = [x, z, \theta]$: with respect to world frame

$u = [u1, u2] = [F1, F2]$ with respect to the body frame

1.2 Calculating the Lagrange

$$T = \frac{1}{2} m (\dot{x}^2 + \dot{z}^2) + \frac{1}{2} I_{zz} \dot{\theta}^2 \quad (K.E)$$

$$V = mgz \quad (P.E.)$$

$$\therefore L = T - V \quad (\text{LAGRANGE})$$

$$= \frac{1}{2} m (\dot{x}^2 + z^2) + \frac{1}{2} I_{zz} \dot{\theta}^2 - mgz$$

1.3 CALCULATING THE EULER-LAGRANGE EQUATIONS

EQ. OF MOTION

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \left(\frac{\partial L}{\partial q}\right) = \begin{bmatrix} F_W \\ M_W \end{bmatrix}$$

Forces and Moments on the world frame

As shown in Figure 1, F1 and F2 are in body coordinates. So we will need to transform it to world coordinates. We can do that by using trigonometry:

$$F_W = \begin{bmatrix} -F_1 \sin\theta - F_2 \sin\theta \\ -F_1 \cos\theta - F_2 \cos\theta \end{bmatrix}$$

$$M_W = F_2 \cdot \frac{L}{2} - F_1 \cdot \frac{L}{2}$$

Ignoring the air resistance and any drag, we get our equations of motion:

$\Rightarrow$ EQ. OF MOTION =

$$= \begin{bmatrix} m\ddot{x} \\ m\ddot{z} - mg \\ \ddot{\theta} I_{zz} \end{bmatrix} = \begin{bmatrix} -F_1 \sin\theta - F_2 \sin\theta \\ -F_1 \cos\theta - F_2 \cos\theta \\ L/2\,(F_2 - F_1) \end{bmatrix}$$

REARENGE IF FOR:

$$M(q)\ddot{q} + \underbrace{C(q,\dot{q})}_{\text{NO C TERM}} + G(q) + \underbrace{D(\dot{q})}_{\text{NO DRAG}} = F$$

$$\underbrace{\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}}_{\substack{\text{INERTIA} \\ \text{MATRIX}}} \begin{bmatrix} \ddot{x} \\ \ddot{z} \\ \ddot{\theta} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ -mg \\ 0 \end{bmatrix}}_{\substack{\text{GRAVITACIONAL} \\ \text{MATRIX}}} = \underbrace{\begin{bmatrix} -F_1 \sin\theta - F_2 \sin\theta \\ -F_1 \cos\theta - F_2 \cos\theta \\ L/2\,(F_2 - F_1) \end{bmatrix}}_{F}$$

$$\Rightarrow M(q)\ddot{q} + G(q) = F$$

$$\Rightarrow \ddot{q} = M^{-1}(q)\left(F - G(q)\right) **$$

2 GETTING A PROJECTION

To get a projection for my system I will follow the model provided in the class notes:

$$\mathscr{P}(\bar{\xi}) = \begin{cases} \dot{x} = f(x,u) \qquad x(0) = x_0 \\ u = \bar{u} + K(\bar{x} - x). \end{cases}$$

First we need to find a ***dx = f(x, u)***. So let's expand our state and apply the solution obtained of our dynamics.

$$x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} x \\ z \\ \theta \\ \dot{x} \\ \dot{z} \\ \dot{\theta} \end{bmatrix} \qquad \dot{x} = \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{z} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{z} \\ \ddot{\theta} \end{bmatrix}$$

$$** \Rightarrow \quad \dot{x} = f(x,u) = \begin{bmatrix} \dot{q} \\ M^{-1}(q)(F - G(q)) \end{bmatrix}$$

The projection can be finalized by substituting **u** and solving the set of equations (see Mathematica code).

3 THE COST EQUATION

The cost equation will be the basic quadratic equation used through the quarter of the form:

$$L = \frac{1}{2}\left((x - x_d)^T Q (x - x_d) + \frac{1}{2}(u - u_d)^T R (u - u_d)\right)$$

$$J = \int_0^T L \, dt + \frac{1}{2}(x - x_d)^T P_1 (x - x_d)$$

3.1 PICKING THE CONSTANTS

For this project it is important to mention the choices that were made for each constant with the purpose of prioritizing position rather than anything else.

Q:

For this constant we give bigger weights to enforce the positions (x, z) and we reduce the weights of the other terms. It is important to reduce weight of **ϴ**, due to the quad rotor dynamics it can only move in **x** by tilting, for this reason by reducing this value we make the quad more likely to move in **x**.

R:

We are going to give small weights to let the rotors apply whatever force necessary to enforce the position.

P1:

Likewise Q, we are going to enforce the position (x, z) better than the other terms.

4 THE GOAL

As mentioned in section 1 the algorithm has as objective to optimize the trajectory of a quad rotor flying in a circular movement. Therefore the following functions will be used as goal:

$$x_d(t) = \cos(3t)$$

$$\dot{x}_d(t) = \frac{d}{dt}\cos(3t)$$

$$z_d(t) = \sin(3t)$$

$$\dot{z}_d(t) = \frac{d}{dt}\sin(3t)$$

$$\theta_d(t) = 0$$

$$\dot{\theta}_d(t) = 0$$

$$F1d = 1.5$$

$$F2d = 1.5$$

F1d and **F2d** gets a small value to try to avoid being negative, what is impossible for a quad rotor since its rotors are usually not capable of generating negative thrusts.

5 ALGORITHM EXECUTION CONSIDERATIONS

The algorithm will take around five minutes to run and will provide an animation of a bar like 2D Quad rotor as well as the plots for {x, z, Θ}. In order to try to make the execution faster I have increased the maximum error allowed to ϵ=10, however the cost converges and given enough time (if you are patient enough) it will provide even better results than 10, probably converging to something very close to 0 eventually.