Homework #2

(due: March 15)

David Felipe Alvear Goyes

(P1) Consider the following reference trajectory for a KUKA robot's joint angles.

$$\theta_{\mathrm{ref}}(t) = (\frac{\pi}{2}\sin t, -\frac{\pi}{2}\sin t, 0, 0, 0, 0, 0)$$

In this problem, using the robot's dynamic model, we control its joint velocities to track the trajectory. Suppose that the initial joint angles $\theta(0)$ of the robot is $\theta(0) = \theta_{\mathrm{ref}}(0)$. Calculate the torque control input $\tau(t)$ to regulate the robot's joint velocities to track the trajectory. Using the provided Pybullet exercise code, validate your answer (implement your answer in `torque_compute.py`).

Solution

- we have the manipulator dynamics as follow:

$$\tau = M(\theta)\ddot{\theta} + V(\theta,\dot{\theta}) + G(\theta) \quad (1)$$

- Given that this expression is non-linear we can use feedback lineorization to make the system linear:

defining $\tau = M(\theta)\tilde{\tau} + V(\theta,\dot{\theta}) + G(\dot{\theta}) \quad (2)$

and using eq (1) and (2):

$$\boxed{\ddot{\theta} = \bar{\tau}}$$

Then $\rightarrow$ $\tau = M(\theta)\ddot{\theta} + V(\theta,\dot{\theta}) + G(\theta)$

we can use this expression for the open loop control. and we can track the trajectory.

- We need to follow the trajectory

$$\theta_{ref} = (\frac{\pi}{2}\sin t, \frac{\pi}{2}\sin t, 0, \cdots 0)$$

$$\dot{\theta}_{ref} = (\frac{\pi}{2}\cos t, -\frac{\pi}{2}\cos t, 0 \cdots 0)$$

we can approximate $\boxed{\ddot{\theta} = \dfrac{\dot{\theta}_{ref}(t+h) - \dot{\theta}(t)}{h}}$

(P2) We want to control a quadrotor to maintain its current altitude (by making the z-axis velocity to zero in the global frame) and to attain a given desired orientation (pitch and roll). The control input to the quadrotor is the (vertical) force and 3-axis torque. Design the following two controllers to compute the force control input and torque control input separately.

(a) Using the dynamic model, design a controller that allows the quadrotor to maintain its current altitude for any given orientation.

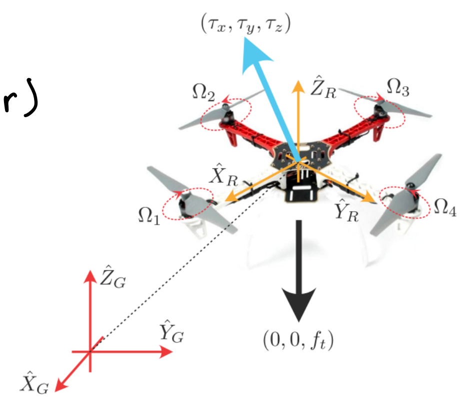
(b) Design a LQR controller to regulate the quadrotor's orientation.

Using the provided Pybullet exercise code, validate if the two controllers you design can be used to solve the main problem (implement your answer in `control_compute.py`).

Solution

Design the following two controllers to compute the force control input and torque control input.

a) Design a controller to maintain the altitude for any given orientation.



b : the truster factor

State $x \in \mathbb{R}^{12}$, $x = (x, y, z, u, v, w, \phi, \theta, \psi, p, q, r)$

$u \in \mathbb{R}^4$ $\quad u = (f_t, \tau_x, \tau_y, \tau_z)$

Dynamics of the quadrator

$$m \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix} + \underline{R_z(\psi) R_y(\theta) R_x(\phi)} \begin{pmatrix} 0 \\ 0 \\ f_t \end{pmatrix}$$

$$R_y(\theta) R_x(\phi) = \begin{pmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & C\phi & -S\phi \\ 0 & S\phi & C\phi \end{pmatrix} = \begin{pmatrix} C\theta & S_\theta S_\phi & S_\theta C_\phi \\ 0 & C\phi & -S\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{pmatrix}$$

$$R_z(\psi) R_y(\theta) R_x(\phi) = \begin{pmatrix} C\psi & -S\psi & 0 \\ S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} C\theta & S_\theta S_\phi & S_\theta C_\phi \\ 0 & C\phi & -S\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{pmatrix}$$

$$R_z(\psi) R_y(\theta) R_x(\phi) = \begin{pmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi - C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{pmatrix}$$

$$R_z(\psi)R_y(\theta)R_x(\phi)\begin{pmatrix}0\\0\\f_t\end{pmatrix} = \begin{pmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi - C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{pmatrix}\begin{pmatrix}0\\0\\f_t\end{pmatrix}$$

$$R_z(\psi)R_y(\theta)R_x(\phi)\begin{pmatrix}0\\0\\f_t\end{pmatrix} = \begin{pmatrix} C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi S_\theta C_\phi - C_\psi S_\phi \\ C_\theta C_\phi \end{pmatrix} f_t$$

Then $\rightarrow$

$$m\begin{pmatrix}\ddot{x}\\\ddot{y}\\\ddot{z}\end{pmatrix} = \begin{pmatrix}0\\0\\-mg\end{pmatrix} + \begin{pmatrix} C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi S_\theta C_\phi - C_\psi S_\phi \\ C_\theta C_\phi \end{pmatrix} f_t$$

Our goal is to control the altitude of the quadrator then:

$$m\ddot{z} + mg = C_\theta C_\phi f_t$$

$$\ddot{z} = \frac{C_\theta C_\phi f_t - mg}{m}$$

For an object governed by Newton's law of motion

$$F(t) = m\ddot{z} = \cancel{m}\frac{C_\theta C_\phi f_t - mg}{\cancel{m}}$$

$$\rightarrow \ddot{z} = \frac{C_\theta C_\phi f_t - mg}{m}$$

To stabilize the quadrotor we need z-axis velocity to zero in the global frame. $\dot{z}_{ref} = 0$

$$\ddot{z} = \frac{\cancel{\dot{z}_{ref}}^{\,0} - \dot{z}}{h} \quad \text{then:}$$

$$\frac{C_\theta C_\phi f_t - mg}{m} = \frac{-\dot{z}}{h} \rightarrow \boxed{f_t = \frac{-m\dot{z}}{C_\theta C_\phi h} + \frac{mg}{C_\theta C_\phi}}$$

B) Design LQR controller to regulate quadrator's orientation

State $x \in \mathbb{R}^{12}$, $x = (x, y, z, u, v, \omega, \phi, \theta, \psi, p, q, r)$

$u \in \mathbb{R}^4$ $u = (f_t, \tau_x, \tau_y, \tau_z)$

From the dynamics of the model we have:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = R_{zyx}(\psi, \theta, \phi) \begin{pmatrix} u \\ v \\ v \end{pmatrix}$$

Our goal is to control the orientation of the quadrotor. (Pitch and roll)

then, we can define the torque control input as:

$u = (\tau_x, \tau_y, \tau_z)$. and using the linearization of the Dynamics of the system we can redefine for the torque control input.

The state of the new system can be:

$x = (\phi, \theta, \psi, p, q, r)$ and $u = (\tau_x, \tau_y, \tau_z)$

then $\rightarrow$ The state equation for the linearized system around the equilibrium for orientation control is:

$$\frac{d}{dt}\begin{pmatrix} \phi \\ \theta \\ \psi \\ p \\ q \\ r \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{A} \begin{pmatrix} \phi \\ \theta \\ \psi \\ p \\ q \\ r \end{pmatrix} + \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ I_x^{-1} & 0 & 0 \\ 0 & I_y^{-1} & 0 \\ 0 & 0 & I_z^{-1} \end{pmatrix}}_{B} \begin{pmatrix} \tau_x \\ \tau_y \\ \tau_z \end{pmatrix}$$

$\rightarrow$ Now, we have the linearized system with the form:

$$\boxed{\dot{x} = Ax + Bu} \qquad \dot{\hat{x}} = A\hat{x} + B\hat{u} \quad \hat{u} = u - \bar{u}$$

$\rightarrow$ then given a reference signal $\dot{x}_{ref} = A x_{ref} + B\bar{u}$

we can d fine $\tilde{x} = x - x_{ref}$ and $\tilde{u} = u - \bar{u}$

thus: $\boxed{\dot{\tilde{x}} = A\tilde{x} + B\tilde{u}}$

$$u = \bar{u} + K(x - x_{ref}) \rightarrow u = K(x - x_{ref})$$

For this system $\bar{u} = 0$

we can use LQR to design K such that $\tilde{u} = KX(t)$ and minimize:

$$\int_{t=0}^{\infty} \left(X^T(t) Q X(t) + U^T(t) R U(t) \right) dt$$

We can select $Q = I$ and $R = I$ to apply Linear Quadratic Regulator.