Homework 9

EECS/BioE/MechE C106A/206A
Introduction to Robotics

Due: November 8, 2022

Problem 1. Feedforward Trajectory Tracking with Jacobians

Say we are working with a Baxter robot and we would like the end effector of our robot arm to perform some trajectory in its workspace. This trajectory is given to us as a smooth trajectory of rigid transforms, $g(t) \in SE(3)$. Assume we have access to both $g(t)$ and $\dot{g}(t)$ for $t \in [0, T]$, and that the trajectory starts from the current position of the robot, $g(0)$, and ends up at some desired position $g(T)$ in T seconds. Recall that we can only give the robot jointspace commands, so we want to convert this workspace trajectory into a jointspace trajectory $\theta(t)$ such that $g_{ST}(\theta(t)) = g(t)$ for $t \in [0, T]$ where $g_{ST}(\theta)$ is the forward kinematics map.

(a) How would you solve this problem using an inverse kinematics solver? Why might this be undesirable?

(b) Write down an expression for the desired spatial workspace velocity $\hat{V}$ that we want the end effector to perform at time t, in order to perform the trajectory g.

(c) Assume we also have efficient access to the spatial jacobian $J(\theta)$. Let $\theta(t)$ be some jointspace trajectory that satisfies the required constraints. Write down an expression for $\dot{\theta}(t)$ in terms of the velocity you computed in the previous part (recall the Moore-Penrose pseudoinverse that we spoke about in lecture).

(d) Explain how we may compute $\theta(t)$ from your answers to the previous two parts.

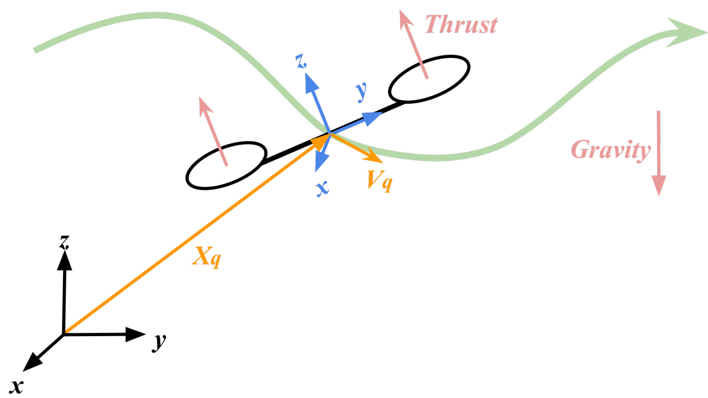
(e) Consider the following strategy to make the robot arm track the desired trajectory. At each timestep, we use (b) to compute the desired spatial velocity, and then we use (c) to compute the desired joint velocity. Then, we send this joint velocity as a command to the robot. Do you think this strategy will allow us to track the desired trajectory well?

Hint: Remember to consider when the solution to (c) *becomes ill-conditioned, in addition to other factors that may lead to poor tracking.*

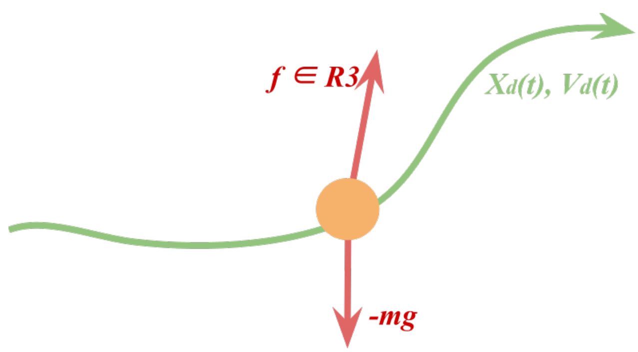
Problem 2. Heavier than Air

Background

Let's briefly discuss some high level design requirements for quadrotor controllers.

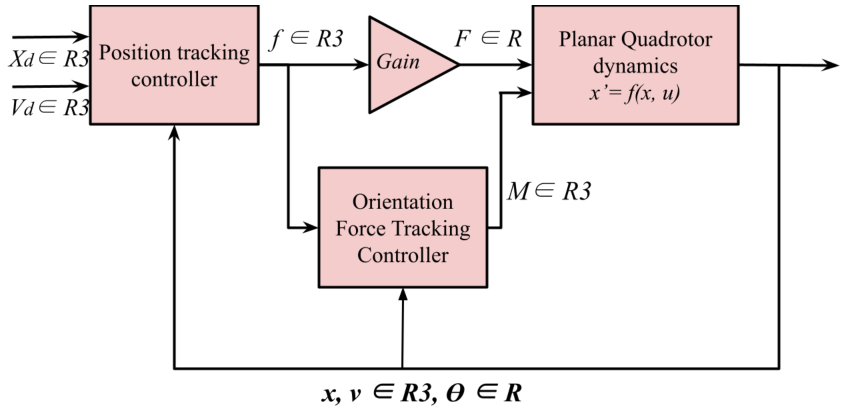


Imagine that we want to design a feedback controller for a simple *planar* quadrotor, which is constrained to travel and rotate in the XY plane. Our controller should *track* a trajectory $x_d \in \mathbb{R}^3$ and provide stable flight. To accomplish these objectives simultaneously, we split our controller into stages.
The first stage of our controller finds a force vector $f \in \mathbb{R}^3$ that allows the quadrotor to track a position $x_d \in \mathbb{R}^3$. It *abstracts away* the orientation of the quadrotor, and treats the system as a point mass with a force input f.



Above: By treating the quadrotor as a single point mass, we can easily develop a position controller.

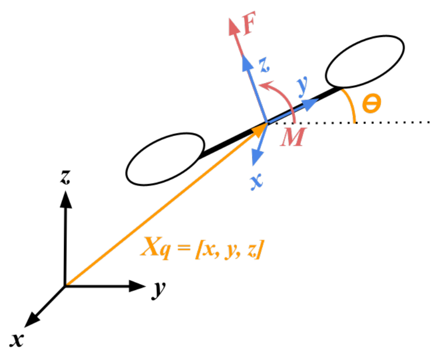
Due to the orientation of its propellers, the quadrotor can only output a force vector along its Z-axis! Thus, to track the force f, we develop an orientation controller to align the Z-axis of the quadrotor with the force vector. This provides the quadrotor with a stable orientation and the ability to output the force vector needed for position control.



Notice the "cascaded" structure of the two feedback controllers, where the output of one controller is passed into the next. The control inputs to this system are a total thrust force $F \in \mathbb{R}$ and moment $M \in \mathbb{R}$ produced by the propellers.

Questions

(a) Consider the following planar quadrotor, which is constrained to move only in the YZ plane and rotate only about the X-axis.



Using Lagrange's equations, show that the following three equations model the dynamics of the quadrotor:

$$m\ddot{y} = -F\sin\theta \tag{1}$$
$$m\ddot{z} = F\cos\theta - mg \tag{2}$$
$$I_{xx}\ddot{\theta} = M \tag{3}$$

Where y, z are the coordinates of the mass center, θ is the orientation angle, m is mass, I_{xx} is the inertia about the X-axis through the mass center, and F, M are the thrust force and moment applied by the propellers.

(b) If $e \in \mathbb{R}^n$ is the system error and $c \in \mathbb{R}$ is a constant, prove that if the ODE below describes the system error in time:

$$0 = K_p e + K_d \dot{e} + c\ddot{e} \tag{4}$$

We can choose $K_p, K_d \in \mathbb{R}^{n \times n}$ such that $\lim_{t \to \infty} e(t) = 0$. *Hint: Choose convenient K_p and K_d matrices and examine an arbitrary row of the ODE.*

(c) If $x_q \in \mathbb{R}^3$ is the point mass position, $f \in \mathbb{R}^3$ is the force vector input, and $\hat{z} = [0, 0, 1]^T$ the dynamics of a point mass under gravity are:

$$m\ddot{x}_q = f - mg\hat{z} \tag{5}$$

Using a PD controller with gains $K_p, K_d \in \mathbb{R}^{3 \times 3}$, find an expression for $f \in \mathbb{R}^3$ such that point mass position error $e = x_d - x_q$ is described by:

$$0 = K_p e + K_d \dot{e} + m\ddot{e} \tag{6}$$

Hint: plug your formula for f into the point mass dynamics.

(d) To output a thrust vector f, the Z-axis of the quadrotor must point in the same direction as f. Write an expression for the desired angle θ_d of the quadrotor in the world frame in terms of $f = [f_x, f_y, f_z]^T \in \mathbb{R}^3$.

(e) Using a PD controller with gains $K_\theta, K_\omega \in \mathbb{R}$, find an expression for the moment $M \in \mathbb{R}$ such that orientation error $e_\theta = \theta_d - \theta$ is described by:

$$0 = K_\theta e_\theta + K_\omega \dot{e}_\theta + I_{xx}\ddot{e}_\theta \tag{7}$$

Hint: plug your formula for M into the quadrotor rotational dynamics.

(f) Implement the functions in the file **controllers.py** and test your controller by running **run_simulation.py**. Tune your gains to produce a response with minimal oscillations that reaches the goal states, plotted in red.
A correctly tuned controller should produce a response similar to that below. Attach your plots to your solution. *Hints: Use diagonal gain matrices, raise K_d to reduce oscillation, angular gains should be very small.*