AE483 Homework #2:
Quad-Rotor Dynamics and Control

(due at the beginning of class on Monday, October 3)

1. *Compute the applied forces and torques on a quad-rotor.* The rotor in Figure 1a generates a force $k_F\sigma^2 a$ and a torque $-k_M\sigma^2 a$ when spinning with angular velocity σa, where $k_F, k_M > 0$ are constant parameters and a is the axis of thrust (a unit vector). We call this rotor a "CCW-rotor," since it spins counter-clockwise about a. A "CW-rotor" (not shown) that spins clockwise about a would generate a force $k_F\sigma^2 a$ and a torque $k_M\sigma^2 a$. Note that a CCW-rotor typically *cannot* spin CW and that a CW-rotor *cannot* spin CCW—i.e., it is always the case that $\sigma \geq 0$. The Hummingbird in Figure 1b has two CCW-rotors and two CW-rotors, all generating thrust in the $-z_1$ direction. Assume that this vehicle has mass m and that k_F, k_M are the same for all rotors.

   (a) Compute the total force f^1_{rotors} and torque τ^1_{rotors} due *only* to the four rotors. Remember that a force f^1_i acting at a point p^1_i also produces a torque $\widehat{p^1_i} f^1_i$.

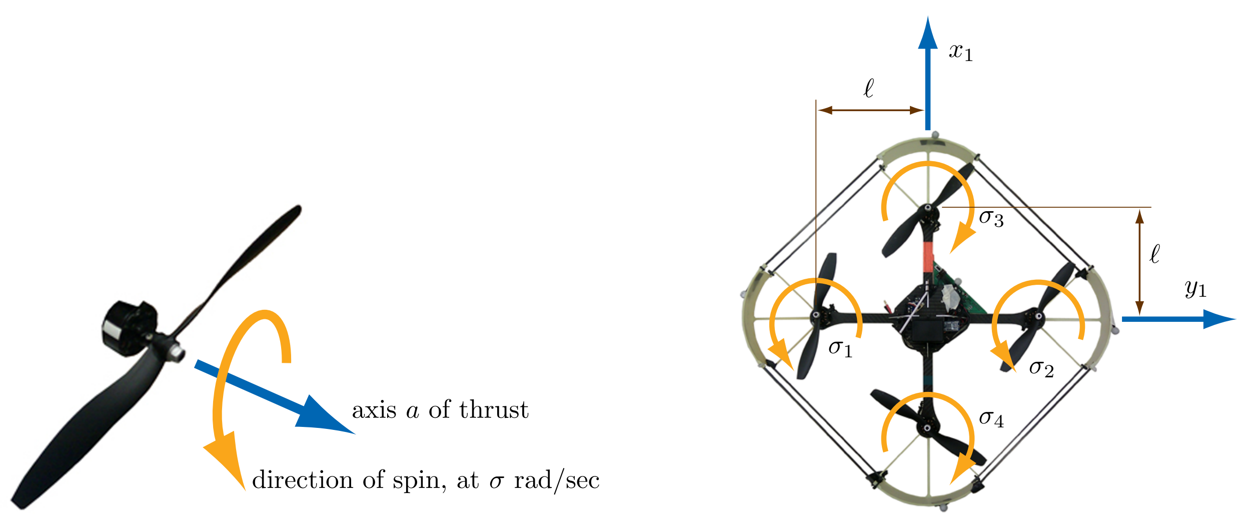
   (b) You should have found that

$$f^1_{\text{rotors}} = \begin{bmatrix} 0 \\ 0 \\ -u_4 \end{bmatrix} \qquad \text{and} \qquad \tau^1_{\text{rotors}} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \qquad \text{for some} \qquad u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}.$$

   Define

$$s = \begin{bmatrix} \sigma_1^2 \\ \sigma_2^2 \\ \sigma_3^2 \\ \sigma_4^2 \end{bmatrix}.$$

   Find the matrix $W \in \mathbb{R}^{4\times 4}$ that satisfies $u = Ws$.



(a) (b)

Figure 1: (a) A single CCW-rotor, and (b) a Hummingbird with two CCW- and two CW-rotors.

(c) When controlling a Hummingbird, we often think of u—rather than s—as the "input." Doing so only makes sense if W is invertible. Prove that it is.

(A square matrix is invertible if and only if its determinant is non-zero. So, you are being asked to show $\det W \neq 0$. You can find $\det W$ by hand or by using whatever software you wish, e.g., Mathematica or symbolic computation in MATLAB.)

(d) Even if W is invertible, it may not be possible to achieve a desired input $\bar{u}$. The reason is that elements of s are bounded below by zero and above by some maximum value $s_{\max} = \sigma_{\max}^2$. Here is one way to address this problem:

- compute the spin rates you want: define $\bar{s} = W^{-1}\bar{u}$
- compute the spin rates you can achieve: for each $i \in \{1, 2, 3, 4\}$, define

$$s_i = \begin{cases} 0 & \text{if } \bar{s}_i < 0, \\ s_{\max} & \text{if } \bar{s}_i > s_{\max}, \\ \bar{s}_i & \text{otherwise.} \end{cases}$$

- compute the corresponding input, which you know you can achieve: define $u = Ws$

Suppose that

$$\ell = 0.5 \qquad k_F = 10^{-5} \qquad k_M = 10^{-7}$$

and that each rotor can achieve a spin rate between 0 and 1000 radians per second. Use the algorithm described above to bound each of the following two desired inputs:

$$\bar{u} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 5 \end{bmatrix} \qquad \text{and} \qquad \bar{u} = \begin{bmatrix} 0.50 \\ -0.50 \\ 0.05 \\ 5 \end{bmatrix}.$$

You are encouraged to use MATLAB. (In fact, the last problem on this assignment will *require* you to implement the algorithm described above in MATLAB.)

(e) Find the total force f^0 and torque τ^1 on the Hummingbird (both due to the rotors and due to gravity), in terms of u. Note that—unlike in part (b)—the total force should be expressed in the coordinates of frame 0. Note also that some part of your answer should involve the rotation matrix R_1^0—please leave this matrix as "R_1^0" (don't express it in terms of Euler angles).

2. *Write the equations of motion for a quad-rotor.* The Newton-Euler equations that govern the motion of a rigid body are

$$\begin{aligned} f^0 &= m\dot{v}_{0,1}^0 \\ \tau^1 &= J^1\dot{w}_{0,1}^1 + \widehat{w_{0,1}^1}J^1 w_{0,1}^1, \end{aligned} \tag{1}$$

where

- 0 is the space frame and 1 is the body frame (with origin at the center of mass);
- $f^0 \in \mathbb{R}^3$ is the applied force, in the coordinates of frame 0;
- $\tau^1 \in \mathbb{R}^3$ is the applied torque, in the coordinates of frame 1;
- $m \in \mathbb{R}$ is the mass of the rigid body;
- $J^1 \in \mathbb{R}^{3\times3}$ is the moment of inertia of the rigid body, in the coordinates of frame 1;

- $v_{0,1}^0$ is the linear velocity of frame 1 with respect to frame 0, in the coordinates of frame 0;
- $w_{0,1}^1$ is the angular velocity of frame 1 with respect to frame 0, in the coordinates of frame 1.

Assuming the use of a body-fixed ZYX Euler angle sequence to represent orientation, the corresponding kinematic equations are

$$\begin{aligned}\dot{o}_1^0 &= v_{0,1}^0 \\ \begin{bmatrix}\dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3\end{bmatrix} &= \frac{1}{c_2}\begin{bmatrix}0 & s_3 & c_3 \\ 0 & c_2 c_3 & -c_2 s_3 \\ c_2 & s_2 s_3 & s_2 c_3\end{bmatrix} w_{0,1}^1.\end{aligned} \tag{2}$$

The rotation matrix produced by ZYX angles $(\theta_1, \theta_2, \theta_3)$ is

$$R_1^0 = \begin{bmatrix} c_1 c_2 & c_1 s_2 s_3 - s_1 c_3 & c_1 s_2 c_3 + s_1 s_3 \\ s_1 c_2 & s_1 s_2 s_3 + c_1 c_3 & s_1 s_2 c_3 - c_1 s_3 \\ -s_2 & c_2 s_3 & c_2 c_3 \end{bmatrix}. \tag{3}$$

The equations of motion (1)-(2) are often written more concisely as

$$\begin{aligned}\dot{o} &= v \\ \dot{\theta} &= Sw \\ f &= m\dot{v} \\ \tau &= J\dot{w} + \widehat{w}Jw,\end{aligned} \tag{4}$$

where it is understood that f, v, and o are in the coordinates of frame 0 and that τ, J, and w are in the coordinates of frame 1. For convenience, we define

$$o = \begin{bmatrix}o_1 \\ o_2 \\ o_3\end{bmatrix} \qquad v = \begin{bmatrix}v_1 \\ v_2 \\ v_3\end{bmatrix} \qquad \theta = \begin{bmatrix}\theta_1 \\ \theta_2 \\ \theta_3\end{bmatrix} \qquad w = \begin{bmatrix}w_1 \\ w_2 \\ w_3\end{bmatrix}.$$

Assuming that the moment of inertia matrix is diagonal, we also define

$$J = \begin{bmatrix}J_1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_3\end{bmatrix}.$$

Using your results from Problem 1, write the equations of motion (4) for a Hummingbird as a set of twelve ordinary differential equations, with variables

$$o_1, o_2, o_3, \theta_1, \theta_2, \theta_3, v_1, v_2, v_3, w_1, w_2, w_3$$

and inputs

$$u_1, u_2, u_3, u_4.$$

3. *Linearize the equations of motion about hover and discretize them.*

(a) Assume that $\theta = 0$. What inputs are required to maintain $v = \dot{v} = 0$ and $w = \dot{w} = 0$? This condition is called *hover with zero yaw*, for obvious reasons.

(b) Linearize the equations of motion of a hummingbird about the hover condition with zero yaw. In particular, find the matrices A_c and B_c that would express these linearized equations of motion in (continuous-time) state-space form:

$$\dot{x}_c(t) = A_c x_c(t) + B_c u_c(t), \tag{5}$$

where

$$x_c = \begin{bmatrix} \delta o \\ \delta\theta \\ \delta v \\ \delta w \end{bmatrix} \qquad \text{and} \qquad u_c = \begin{bmatrix} \delta u \end{bmatrix},$$

and where the symbol "δ" denotes the change in a variable from its hover condition.

It is helpful to remember that the derivative of a function $h(z)\colon \mathbb{R}^m \to \mathbb{R}^n$ at $z \in \mathbb{R}^m$ is

$$Dh(z) = \begin{bmatrix} \partial h_1/\partial z_1 & \cdots & \partial h_1/\partial z_m \\ \vdots & \ddots & \vdots \\ \partial h_n/\partial z_1 & \cdots & \partial h_n/\partial z_m \end{bmatrix}$$

and that a linear approximation to h at z is

$$h(z + \delta z) \approx h(z) + (Dh(z))\, \delta z.$$

(c) Suppose the input is constant on time intervals of length $\Delta t > 0$. In other words, for each non-negative integer $i \in \{0, 1, 2, \dots\}$, we have $u_c(t) = u_c(i\Delta t)$ for all $t \in [i\Delta t, (i+1)\Delta t]$. Then we can linearly approximate

$$x_c((i+1)\Delta t) \approx x_c(i\Delta t) + \dot{x}_c(i\Delta t)\Delta t.$$

Find the matrices A_d and B_d that would express this approximation in (discrete-time) state-space form:

$$x_d(i+1) = A_d x_d(i) + B_d u_d(i),$$

where $x_d(i) = x_c(i\Delta t)$ and $u_d(i) = u_c(i\Delta t)$. You are welcome to write your answer in terms of A_c and B_c, if that is helpful (i.e., there is no need to write out every element of A_d and B_d again). What you have done is *discretize* the system (5).

4. *Design and implement a control policy that maintains hover with zero yaw.* In this problem, you will continue building a quad-rotor simulator in MATLAB. Here, you will implement the dynamics and control part of this simulator. Your code will:

   - Create a movie of the quadrotor hovering, having started from the following initial state:

$$o_1^0 = \begin{bmatrix} 0.5 \\ -0.5 \\ -0.5 \end{bmatrix} \qquad (\theta_1, \theta_2, \theta_3) = \left(-\frac{\pi}{4}, \frac{\pi}{6}, \frac{\pi}{12}\right) \qquad v_{0,1}^0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad w_{0,1}^1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

   - Plot position as a function of time (i.e., components of o_1^0).
   - Plot orientation as a function of time (i.e., ZYX Euler Angles).
   - Plot linear velocity as a function of time (i.e., components of $v_{0,1}^0$).
   - Plot angular velocity as a function of time (i.e., components of $w_{0,1}^1$).
   - Plot input as a function of time (i.e., components of u).
   - Plot propellor spin rates as a function of time (i.e., $\sigma_1, \ldots, \sigma_4$).

   You will do so by modifying your code from HW1. Guidance will be posted to piazza (as well as submission details) and will be discussed at length in class. For now, we will say only that "control" means choosing the input u and that minimizing a cost is one way to make this choice. In particular, it is useful to know that the solution to

$$\begin{aligned} \underset{u_d}{\text{minimize}} \quad & \sum_{i=0}^{\infty} \left(x_d(i)^T Q_d x_d(i) + u_d(i)^T R_d u_d(i) \right) \\ \text{subject to} \quad & x_d(i+1) = A_d x_d(i) + B_d u_d(i), \qquad x_d(0) = x_0 \end{aligned} \tag{6}$$

   has the form

$$u_d(i) = -K_d x_d(i)$$

   where K_d can be found in MATLAB as follows:

```
Kd = dlqr(Ad,Bd,Qd,Rd)
```

   In any case, please start early on this problem.