ASEN 3801 Aerospace Vehicle Dynamics and Control Lab
Fall 2025

Lab 4: Quadrotor Simulation and Control

Assigned Tuesday, October 14, 2025
Due Monday, November 10, 2025, at 11:59pm

OBJECTIVES

1. Simulate the equations of motion (EOM) for a quadrotor aircraft
2. Investigate the concepts of trim and static stability
3. Simulate and compare the nonlinear and linear quadrotor EOM
4. Implement feedback control on quadrotor simulation
5. Design and implement feedback control for nonlinear and linearized quadrotor models

BACKGROUND

In this lab assignment students will create a simulation of the nonlinear equations of motion of a quadrotor aircraft, compare them to a linearized model, design feedback control laws for the position control of the aircraft, and analyze the resulting control system response. This lab assignment uses models and equations presented in ASEN 3728. Although key aspects of those models will be given here, students should refer to their notes and materials from that course.

Parrot Mambo Minidrone

In this lab assignment, students will create a simulation of the full nonlinear equations of motion of the Parrot Mambo Minidrone (Figure 1). Smead Aerospace operates these minidrones in the Autonomous System Programming, Evaluation, and Networking (ASPEN) Lab. Note that these quadrotors are different from the ones used in the Lab 2 assignment, but are flown in the same motion capture system in the ASPEN Lab. A short video describing the drone can be found here:

https://www.youtube.com/playlist?list=PLfIpWHYDHoWxHJ0fVvhWiATo1UsvypJmI

Figure 1. The Parrot Mambo minidrone

Quadrotor parameters

These are the parameters for the Parrot Mambo Minidrone:

Quadrotor mass, m:	0.068 kg
Radial distance from CG to propeller, d:	0.060 m
Control moment coefficient, k_m:	0.0024 N*m/(N)
Body x-axis Moment of Inertia, I_x:	5.8E-5 kg*m^2
Body y-axis Moment of Inertia, I_y:	7.2E-5 kg*m^2
Body z-axis Moment of Inertia, I_z:	1.0E-4 kg*m^2
Aerodynamic force coefficient v:	1E-3 N/(m/s)^2
Aerodynamic moment coefficient μ:	2E-6 N*m/(rad/s)^2

Quadrotor equations of motion and linearized model

The quadrotor nonlinear equations of motion are derived from the general aircraft equations of motion by applying specific equations for the moment of inertia matrix, the aerodynamic forces and moments, and the control (motor) forces. A short review of these equations was provided in the additional lab assignment resources and students can refer to those slides for the full set of equations. Likewise, the linearized equations of motion were also presented and are provided in the slides.

For this assignment, students will use modal response theory to design and analyze closed loop control laws for the Parrot Mambo minidrone. Modal analysis was covered in detail in ASEN 3728. In summary, students will derive state space models that result from their control laws, and analyze the model response by looking at the eigenvalues and eigenvectors of the resulting matrices.

Calculating motor forces from control force and moments

In this lab assignment, students will implement feedback control laws from aircraft state variables to the control moments L_c, M_c, and N_c. While it is straightforward to implement such a control law in simulation, in practice the motor thrust forces f_1, f_2, f_3, and f_4 are what must be passed to the quadrotor actuators. Therefore, we need to be able to calculate the motor control thrusts from the desired control forces and moments. Recall the following equations (from ASEN 3728 notes) for the control force and moments:

$$Z_c = -f_1 - f_2 - f_3 - f_4$$

$$L_c = \frac{d}{\sqrt{2}}(-f_1 - f_2 + f_3 + f_4)$$

$$M_c = \frac{d}{\sqrt{2}}(f_1 - f_2 - f_3 + f_4)$$

$$N_c = k_m(f_1 - f_2 + f_3 - f_4).$$

These equations are all linear in the motor thrust forces and can be rewritten in matrix form as:

$$\begin{bmatrix} Z_c \\ L_c \\ M_c \\ N_c \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 & -1 \\ -\frac{d}{\sqrt{2}} & -\frac{d}{\sqrt{2}} & \frac{d}{\sqrt{2}} & \frac{d}{\sqrt{2}} \\ \frac{d}{\sqrt{2}} & -\frac{d}{\sqrt{2}} & -\frac{d}{\sqrt{2}} & \frac{d}{\sqrt{2}} \\ k_m & -k_m & k_m & -k_m \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}$$

The 4x4 matrix is constant and invertible, allowing us to solve for the motor thrust forces when given the control forces and moments:

$$\begin{bmatrix} -1 & -1 & -1 & -1 \\ -\frac{d}{\sqrt{2}} & -\frac{d}{\sqrt{2}} & \frac{d}{\sqrt{2}} & \frac{d}{\sqrt{2}} \\ \frac{d}{\sqrt{2}} & -\frac{d}{\sqrt{2}} & -\frac{d}{\sqrt{2}} & \frac{d}{\sqrt{2}} \\ k_m & -k_m & k_m & -k_m \end{bmatrix}^{-1} \begin{bmatrix} Z_c \\ L_c \\ M_c \\ N_c \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}$$

Dominant pole design method

In this assignment, students are asked to use the "dominant pole" design method. In this approach, the slowest pole is specified, eg. by its time constant or natural frequency. The slowest mode will "dominate" the overall behavior if the other poles are significantly faster. In this case, a good design rule is to make the other poles 5-10x faster than the slow, dominant pole.

Feedforward speed command

This assignment asks students to determine the "open loop" or "feedforward" speed command to change the position of the quadrotor. This is different from the feedback control process described in lecture that uses the gain K_4. Here, the reference speed command is explicitly given as a function of time based on the desired behavior of the quadrotor.

Consider the reference forward speed command $\Delta u_r^E(t)$ specified from time t_1 to time t_2. If we assume that the inner loop control works well and $\Delta u^E(t) = \Delta u_r^E(t)$, then if we know the position at t_1, the position at t_2 is:

$$\Delta x_E(t_2) = \Delta x_E(t_1) + \int_{t_1}^{t_2} \Delta u_r^E(t)\, dt$$

The challenge is then to determine the speed command as a function of time to achieve a desired final result. In this lab, you are asked to consider a reference command that is constant over a given short time interval and zero afterward.

LAB TASKS

This lab assignment is divided into three main sections. Each section builds on the previous one, so students should work on them in order. As with previous lab assignments, when problems ask students to create specific functions, the function call must be created exactly as specified.

Lab Task 1: Nonlinear Equations of Motion

1. Write a function to plot the full aircraft states and control inputs that result from a simulation run:

   ```
   function PlotAircraftSim(time, aircraft_state_array, control_input_array,
   fig, col)
   ```

   This function is used to plot the results of a full simulation once it is complete. It takes as input the length n vector which holds the time corresponding to the n^{th} set of variables, the 12 x n array of aircraft states, the 4 by n array of control inputs $[Z_c, L_c, M_c, N_c]^T$, the 6 x 1 vector of figure numbers to plot over, and the string `col` which indicates the plotting option used for every plot, eg. `col = 'b-'`.

   The function should plot 6 figures. There should be four figures each with three subplots for the inertial position, Euler angles, inertial velocity in body frame, and angular velocity, respectively. There should be one figure with four subplots for each control input variable. Finally, there should be one figure that shows the three-dimensional path of the aircraft, with positive height upward in the figure. This figure should indicate the start (green) and finish (red) of the path with different colored markers.

   The function must be able to be called multiple times for different simulation runs with different `col` indicators. Thus, the function should call each figure before plotting and include the `hold on` command. However, it should not clear the figures before plotting.

   For example:

   ```
   figure(fig(1));
   subplot(311);
   plot(time, aircraft_state_array(1,:), col); hold on;
   subplot(312);
   plot(time, aircraft_state_array(2,:), col); hold on;
   …
   ```

2. Create a simulation of a quadrotor in MATLAB using `ode45`. Create the function:

   ```
   function var_dot = QuadrotorEOM(t, var, g, m, I, d, km, nu, mu, motor_forces)
   ```

   for use by `ode45` to simulate the full nonlinear equations of motion where: `t` is time; `var` is the 12 x 1 aircraft state vector; `g` is the acceleration due to gravity; `m` is mass; `I` is the inertia matrix; `d`, `km`, `nu`, and `mu` are the remaining quadrotor parameters; `motor_forces = [f1; f2; f3; f4]` is the 4 x 1 vector of motor forces, and `var_dot` is the 12 x 1 derivative of the state vector. Include attitude dynamics and kinematics using the Euler angle attitude representation. Use the body frame representation of translational and angular velocities that is presented in ASEN 3728 and

summarized in the accompanying notes. Add control forces and moments but do not include aerodynamic forces and moments until Problem 2 from the rotors.

- Determine trim thrusts for the rotors for steady hovering flight.
- Simulate this trim state for 10 secs and verify that it produces equilibrium motion.

3. Add aerodynamic forces and moments due to drag to the simulation. Again run the simulations for 10 secs.

    - Verify that this does not alter the trim state for steady hover. Explain your verification process and provide relevant plots as evidence to support it.

4. Use the EOM and simulation from Problem 1.3 to perform additional simulations:

    - Analytically derive the trim state and rotor thrust trim values for a constant velocity translation at 5 m/s East, while maintaining a yaw of 0 deg. Simulate this trim state for 10 secs and verify it produces the expected trim motion.
    - Derive the trim state if a yaw of 90 deg is to be maintained instead while translating 5 m/s East. Verify this trim state in your simulation as well.

5. Is steady hovering flight stable for the quadrotor? Determine this through simulation, and through the behavior of the hardware demonstration system physically, and via plots of translation and rotation over time. A video of the Parrot Mambo flying with no control is available at the link below. The data file "RSdata_nocontrol.mat" describes the behavior of the quadrotor in the video and is provided with this lab assignment in Canvas (read the document "ASEN_3128_Quadcopter_Data_File_Instructions.pdf" for instructions on how to work with the data contained in the .mat file).

    https://youtu.be/rcItTz1nSFs

Lab Task 2: Linear Equations and Simple Control

1. Simulate the response of the **nonlinear quadrotor equations of motion** to initial condition deviations from the steady hover trim state as follows **(use a 10 sec simulation window)**:

    a) Deviation by +5 deg in roll
    b) Deviation by +5 deg in pitch
    c) Deviation by +5 deg in yaw
    d) Deviation by +0.1 rad/sec in roll rate
    e) Deviation by +0.1 rad/sec in pitch rate
    f) Deviation by +0.1 rad/sec in yaw rate

    Plot and discuss the resulting behavior. Does it make physical sense? Do the results agree with your conclusion in Problem 1.5? For this problem, use `PlotAircraftSim` to create a set of 6 plots per deviation condition.

2. Develop a **linearized model of the full quadrotor dynamics** about a steady hover trim state. Repeat Problem 2.1 using the linearized dynamics model, and compare linearized and non-linearized behaviors on the same plots that you obtained for Problem 2.1 (ie. turn in one final set of plots for the combined results of Problem 2.1 and 2.2, using the same 10 sec simulation window).

    Do not plot responses from multiple deviations on the same plot. For example, don't plot responses to deviations of +5 deg roll and +5 deg pitch on the same plots, but do plot linearized and nonlinear responses for +5 deg roll on the same plots).

    Your linearized equations of motion function should be of the form:

    ```
    function var_dot = QuadrotorEOM_Linearized(t, var, g, m, I, deltaFc, deltaGc)
    ```

    Where `var` is the 12x1 column vector of state deviations from the steady hover trim condition, ordered as presented in the supplementary document. `deltaFc` and `deltaGc` are deviations from the steady hover trim condition.

3. Create a function to calculate the control vectors `Fc` and `Gc`. The function takes as input the 12x1 aircraft state `var`, aircraft mass `m`, and gravitational acceleration `g`. The control force in the body z-direction should still equal the weight of the quadrotor. Set the control moments about each body axis proportional to the rotational rates about their respective axes, but in the opposite sign of the angular velocity with a gain of 0.004 Nm/(rad/sec):

    ```
    function [Fc, Gc] = RotationDerivativeFeedback(var, m, g)
    ```

4. Write a function to calculate the motor thrust forces given the control force and moments:

    ```
    function motor_forces = ComputeMotorForces(Fc, Gc, d, km)
    ```

    where `motor_forces` is the 4 x 1column vector $[f_1, f_2, f_3, f_4]^T$. The control thrust and moment vectors are both 3x1 column vectors, while `d` is the distance from aircraft CG to each rotor and `km` is the control moment coefficient.

5. Add the feedback controller to your nonlinear quadrotor equations of motion. Create a new equation of motion function with the feedback controller from Problem 2.4 in it:

```
function var_dot = QuadrotorEOMwithRateFeedback(t, var, g, m, I, nu, mu)
```

Repeat simulations for the setups in Problems 2.1.d through 2.1.f. What is the effect of this control law? Support your argument with new plots for each setup, overlaying the controlled and uncontrolled nonlinear systems. Also calculate and plot the motor thrust forces for the controlled and uncontrolled systems.

Lab Task 3: Feedback Control Design and Implementation

1. Design a feedback control system for the linearized lateral and longitudinal dynamics of the quadrotor to stabilize roll and pitch attitude. Use feedback gains on roll rate and pitch rate, along with feedback gains on roll and pitch angles. **Design these feedback gains to produce modal behavior for the "top two" states (eg. roll and roll rate) in the lateral and longitudinal sets such that the modes have real eigenvalues, and one eigenvalue dominates the time constants in each set with a value of 0.5 sec.** Keep the angular rate feedback control from Problem 2.3 for the spin portion of the system.

2. Create a function to calculate the control vectors `Fc` and `Gc`. The function takes as input the 12x1 aircraft state `var`. The control force in the body z-direction should still equal the weight of the quadrotor (hard code the values in the function along with the control gains). Set the control moment using the control laws from Problem 3.1.

   ```
   function [Fc, Gc] = InnerLoopFeedback(var)
   ```

3. Using the function from Problem 3.2, create a new EOM function to simulate the response of the **closed loop <u>linearized</u>** system. Use this new function to simulate initial condition deviations from the steady hover trim state as follows **(use a 10 sec simulation window)**:
   a. Deviation by +5 deg in roll
   b. Deviation by +5 deg in pitch
   c. Deviation by +0.1 rad/sec in roll rate
   d. Deviation by +0.1 rad/sec in pitch rate

   Submit plots of the aircraft state variables and control force and moments versus time for each case. Discuss the resulting behavior. Does it correspond to the expected behavior from the linearized modal response theory? Is steady hover now a stable flight condition?

4. Repeat Problem 3.3 (ie. create a new EOM function that adds the control to the nonlinear quadrotor dynamics) using the **<u>nonlinear</u>** dynamics model together with the feedback control design for the linearized system, and compare the closed loop linearized and nonlinear behaviors.

5. Using the feedback control gains K_1 and K_2 designed in Problem 3.1 for the linearized lateral and longitudinal rotation dynamics of the quadrotor as an "inner loop" control law, arranged so that the roll and pitch angles track their corresponding reference angles, design an "outer loop" tracking control law with gain K_3 to cause the translational inertial velocity components to track corresponding reference commands, as shown in the block diagram (Figure 2) for the linearized lateral set. Design K_3 so that the closed loop three-state systems (lateral and longitudinal) have real eigenvalues, with corresponding modal time constants no larger than 1.25 sec. Use MATLAB's `eig` function to find the eigenvalues of the system for a range of gains K_3 and plot the locus of these eigenvalues in the complex plane. Use this locus plot to determine gain values that satisfy the design objectives. Pay careful attention to the sign of K_3 in the longitudinal set.

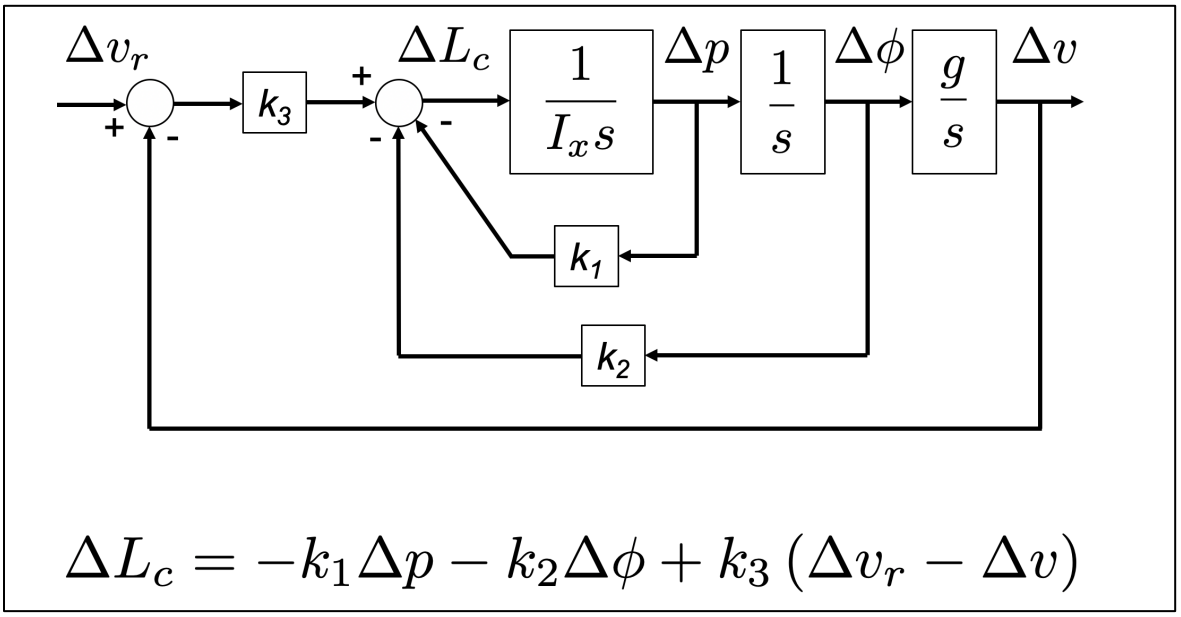

Figure 2. The block diagram for the lateral speed guidance of a quadrotor aircraft.

6. Design "open loop" or "feedforward" commands Δv_r^E and Δu_r^E to cause the corresponding inertial displacements Δy_E and Δx_E (assuming $\psi_0 = 0$) to change by 1.0 m in 2.0 sec and then come to rest. Design these reference commands assuming the reference speed is achieved instantaneously, eg. that $\Delta u^E(t) = \Delta u_r^E(t)$. Create a function to calculate the control vectors `Fc` and `Gc` for a controller that uses the gains from Problem 3.1, Problem 3.5, and the reference commands designed here. The function takes as input the current time `t` and the 12x1 aircraft state `var`. The control force in the body z-direction should still equal the weight of the quadrotor. The control moment is determined by the gains from Problem 3.1, Problem 3.5, and the reference commands designed. All gain values should be hard coded into this function. Likewise, the time t should be used to determine the reference commands.

```
function [Fc, Gc] = VelocityReferenceFeedback(t, var)
```

7. Using the controller from Problem 3.6, create another nonlinear EOM function and simulate this system in MATLAB, for both longitudinal and lateral translation, one at a time, and verify that the simulation behaves as desired. Provide reasoning for the differences between expected and actual behavior.

SUBMISSION REQUIREMENTS & FORMAT

Each group must submit a concise lab writeup through Gradescope. ***Submit a single PDF*** file for your lab writeup document as a group submission through Gradescope, including all plots and code generated for the corresponding problems. Where the lab assignment asks a question or requires plots, include the result in the main narrative of your submission. Where the assignment asks you to write a function, include the full text of the function as an appendix.

Be sure to add titles to all figures that include both the problem number and a description of the plot, eg. Prob#3 Output x versus Time. The assignment will be evaluated based on i) correct answers (including concise explanation of solution and concise discussion/analysis of results – these must be provided separately from code, eg. not put into code comments or code output); ii) proper commenting and documenting of code; and iii) the quality of the figures submitted (e.g. labeling, axis, etc).

Be sure to select the correct pages that are associated with each problem in Gradescope and double-check that your submission is readable.

All lab assignments should include the Team Participation table and should be completed and acknowledged by all team members. Description of the Team Participation table is provided in a separate document.