ASEN 5519 Small UAS Guidance and Control

Homework 2 Assignment

Assigned: Monday, January 30, 2023

Due: 11:59 PM, Thursday, February, 2023

Background

A state vector formulation will be used throughout this course to describe the full nonlinear aircraft dynamics. Aircraft dynamics are described by a set of 12 different equations we derived in lecture. For these equations we have the state vector

$$\mathbf{x} = \begin{bmatrix} x_E & y_E & z_E & \phi & \theta & \psi & u^E & v^E & w^E & p & q & r \end{bmatrix}^T \tag{1}$$

where $\begin{bmatrix} x_E & y_E & z_E \end{bmatrix}^T$ describe the aircraft's inertial position, $\begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ are the Euler angles that describe the aircraft orientation, $\begin{bmatrix} u^E & v^E & w^E \end{bmatrix}^T$ describe the aircraft velocity, and, $\begin{bmatrix} p & q & r \end{bmatrix}^T$ describe the aircraft angular velocity. The control surfaces are combined into the input vector

$$\mathbf{u} = \begin{bmatrix} \delta_e & \delta_a & \delta_r & \delta_t \end{bmatrix}^T. \tag{2}$$

I good mnemonic to remember the order is 'EARTh'.

Nondimensional Aerodynamic Forces and Moments

In order to simulate the aircraft dynamics we need to be able to relate the aerodynamic forces $\begin{bmatrix} X & Y & Z \end{bmatrix}^T$ and moments $\begin{bmatrix} L & M & N \end{bmatrix}^T$ to the states and control inputs. However, aerodynamic data is typically given in terms of lift and drag coefficient derivatives. Therefore, we first determine the lift, drag, and thrust coefficients and then convert the results into the body coordinate coefficients. The expressions given below will be derived and discussed in lecture. The non-dimensional aerodynamic coefficient are:

$$\begin{aligned}
C_L &= C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\hat{q} + C_{L_{\delta_e}}\delta_e \\
C_D &= C_{D_{min}} + K(C_L - C_{Lmin})^2 \\
C_T &= 2\frac{S_{prop}}{S}C_{prop}\frac{\delta_t}{V_a^2}\left[V_a + \delta_t(k_m - V_a)\right]\left[k_m - V_a\right] \\
C_Y &= C_{Y_\beta}\beta + C_{Y_p}\hat{p} + C_{Y_r}\hat{r} + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r \\
C_l &= C_{l_\beta}\beta + C_{l_p}\hat{p} + C_{l_r}\hat{r} + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r \\
C_m &= C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\hat{q} + C_{m_{\delta_e}}\delta_e \\
C_n &= C_{n_\beta}\beta + C_{n_p}\hat{p} + C_{n_r}\hat{r} + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r
\end{aligned} \tag{3}$$

where $\hat{p}$, $\hat{q}$ and $\hat{r}$ are nondimensional versions of the angular velocity components, $K = 1/(\pi e AR)$ is determined from the aspect ratio AR and Oswald's efficincy factor e, S_{prop} is the cross-section area the engine's propellor sweeps out, C_{prop} is the engine coefficient, and k_m is a motor constant. In the case of the data provided for this lab, the body coordinate of the aircraft is aligned with the thrust line of the engine. As a result, the lift coefficient may not be zero when the angle of attack is zero. The terms C_{L_0} and C_{m_0} are the coefficient values for zero angle of attack, not the trim values. All non-dimensional coefficients are linear in the aircraft state except the drag coefficient C_D and the thrust coefficient C_T. The thrust coefficient is a function of the airspeed V_a. Values for the coefficients are provided in the aircraft parameter file `ttwistor.m`.

Equation (3) provides the aerodynamic coefficients for lift and drag, which must be converted into the body X- and Z- coefficients (the remaining apply directly to the body coordinates. These coefficients are

$$\begin{aligned}
C_X &= -C_D\cos\alpha + C_L\sin\alpha \\
C_Z &= -C_D\sin\alpha - C_L\cos\alpha.
\end{aligned} \tag{4}$$

Dimensional Aerodynamic Forces and Moments

Expressions for related the nondimensional rates and coefficients to their dimensional counterparts are summarized here:

$$\begin{aligned}
\hat{p} &= (pb)/(2V) \\
\hat{q} &= (q\bar{c})/(2V) \\
\hat{r} &= (rb)/(2V)
\end{aligned} \tag{5}$$

$$
\begin{aligned}
X &= SQC_X \\
Y &= SQC_Y \\
Z &= SQC_Z \\
L &= bSQC_l \\
M &= \bar{c}SQC_m \\
N &= bSQC_n
\end{aligned}
\tag{6}
$$

where $Q = \frac{1}{2}\rho V^2$ is the dynamic pressure, and L is the roll moment (not lift).

Matlab Structures

One of the challenges creating a simulation capability that can be generalized to different aircraft models is passing the aircraft parameters to the functions that need them. In this class we will use a Matlab (or langiuage of choice) structure to create a single data type that includes all aircraft parameters. In particular, you will be given one M-file for each aircraft that defines a structure that includes all parameters for that aircraft. Unlike variables or arrays that index entries by a number within parenthesis, e.g. "x(1)", a structure defines each component by a unique label separated from the main variable name by a period, e.g. "my_struct.x".

In this lab assignment we will study the aerodynamics of the RECUV TTwistor unmanned aircraft. The file `ttwistor.m` is provided based on results provided by the Athena Vortex Lattice (AVL) software and additional computational fluid dynamics simulations. In the M-file the structure "aircraft_parameters" is defined to include all mass properties, aircraft geometry, and stability derivatives for the aircraft. For example:

```
% ttwistor.m
%
aircraft_parameters.m = 5.74;
...
aircraft_parameters.CLalpha = 1234;
aircraft_parameters.Cmalpha = 5678;
```

. . .

Density and Altitude

The air density ρ plays an important role in the creation of aerodynamic forces and moments. The density varies as a function of altitude (height). There are a variety of models that describe this relationship. We will use the `stdatmo.m` function provided by Matlab in all our simulations, e.g. `density = stdatmo(height)`. NOTE; the input is height which is the negative inertial-z position component.

Problems

Submit all code through the course web site as well as a pdf or txt document answering the questions below. Only submit code for running one of the parts of Problem 2.

Problem 1

Create the following functions:

a. `[wind_angles] = AirRelativeVelocityVectorToWindAngles(velocity_body)`

Given the air relative velocity vector in body coordinates $\mathbf{v}_B$, this function returns the wind angles in the column vector $[V_a,\ \beta,\ \alpha]^T$.

b. `[aero_force, aero_moment] =`
`AeroForcesAndMoments_BodyState_WindCoeffs(aircraft_state, aircraft_surfaces,`
`wind_inertial, density, aircraft_parameters)`

Create the above function that takes as input the aircraft state, the control input vector, the inertial wind velocity in inertial coordinates, the air density, and the aircraft_parameters structure and returns the **aerodynamic** force and moment acting on the aircraft expressed in body coordinates. For this function the propulsive force is considered part of the aerodynamic force, and it DOES NOT include weight. The moment includes aerodynamic and propulsive moments. The output of the function should be two vectors, one for the force and one for the moment.

c. `[aircraft_forces, aircraft_moments] =`
`AircraftForcesAndMoments(aircraft_state, aircraft_surfaces,`
`wind_inertial, density, aircraft_parameters)`

Create the above function that takes as input the aircraft state, the control input vector, the inertial wind velocity in inertial coordinates, the air density, and the aircraft_parameters structure and returns the total force and moment acting on the aircraft expressed in body coordinates. The total force includes the aerodynamic force, propulsive force, and weight. The total moment includes aerodynamic and propulsive moments. The output of the function should be two vectors, one for the force and one for the moment.

d. `[xdot] = AircraftEOM(time,aircraft_state,aircraft_surfaces,wind_inertial, aircraft_parameters)` Implement the full equations of motion by returning the derivative of the state vector, $\dot{\mathbf{x}}$. Note, the input `time` is needed for the Matlab simulation tools. Although it is not used, it needs to be included.

e. `PlotSimulation(time, aircraft_state_array, control_input_array, col)`

This function is used to plot the results of a complete simulation once it is completed. It takes as input the length n vector which holds the time corresponding to the set of variables, the 12 by n array of aircraft state, the 4 by n array of control inputs, and the string *col* which indicates the plotting option used for every plot, e.g. $col = $ '$b-$'.

The function should create a total of 6 figures. There should be four figures each with three subplots for the position, Euler angles, inertial velocity, 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(1);
subplot(311);
plot(time, aircraft_state_array(1,:), col); hold on;
subplot(312);
plot(time, aircraft_state_array(2,:), col); hold on;
```

Problem 2

Use the progamming environment of your choice to simulate the aircraft in the following conditions. Plot and describe the results. When describing the results use quantitative terms as much as possible, e.g. "the pitch angle oscillated with a period of 5 seconds and decayed with a settling time of 1.2 sec". In all cases be sure to let the simulations run long enough to see the behavior reach steady state. NOTE: Even though specifications may be given in degrees or deg/sec, angles are always specified as radians unless specifically noted (e.g. in the figure labels).

1. All aircraft states, control inputs, wind angles, and background wind are zero except height $h = 1655$ m (Boulder's elevation), $\alpha = 0°$, and airspeed $V_a = 18$ m/s.

2. Initial aircraft state and control input the same as Part 1, but background wind $\mathbf{w}^E = [10,\ 10,\ 0]^T$ m/s. Why is the behavior so different?

3. No background wind and initial aircraft state and control surface inputs

$$\mathbf{x}_0 = [0\text{m}, 0\text{m}, -1800\text{m}, 15°, -12°, 270°, 19\text{m/s}, 3\text{m/s}, -2\text{m/s}, .08°/\text{s}, -0.2°/\text{s}, 0°/\text{s}]$$

$$\mathbf{u}_0 = [5°, 2°, -13°, .3].$$

Problem 3

Consider the aircraft from Problem 2.1. When the aircraft is initialized, is it in equilibrium? Justify your answer based only on the information given regarding the initial condition, not based on the simulation result.