ASEN 3728 Aircraft Dynamics
Programming Homework 3

Due date listed on Gradescope.

In this assignment, you will create a nonlinear model of a conventional aircraft, simulate its longitudinal motion, and compare the results with a linear approximation. The aircraft is the TTwistor, a twin engine research aircraft designed here at CU Boulder and pictured below. The aircraft parameters are given in the Matlab struct returned by the `ttwistor` function.

All files are available by cloning the git repository at `https://github.com/zsunberg/Aircraft-Dynamics-Materials` and navigating to the `assignments/P3` directory. A zip file is also available at `https://github.com/zsunberg/Aircraft-Dynamics-Materials/raw/main/zips/assignments/P3.zip`. It is possible that there will be bugfixes to the assignment after it is released. These will be announced on Piazza.

Task 1. Implement the `lonAeroForcesAndMoments` function and run the tests as indicated in `TEMPLATE_report.m` to verify that it is implemented correctly. See the documentation of the function for details about inputs and outputs. You may look at `latAeroForcesAndMoments.m` to see a similar implementation for lateral dynamics.

Assume that the lift and moment coefficients are linear functions of α, $\hat{q}$, and δ_e, and that $C_D = C_{D_{\min}} + K(C_L - C_{L_{\min}})^2$.Additionally, assume that all derivatives related to $\dot{\alpha}$ are zero. Use the thrust model already implemented in `TEMPLATE_lonAeroForcesAndMoments.m` and assume that the thrust is aligned with the x axis of the aircraft. Note that the field `CL0` is the lift coefficient at zero angle of attack - not the lift coefficient at the minimum drag angle of attack or the lift coefficient at trim.

Task 2. Implement the `aircraftDynamics` function and run the tests as indicated in `TEMPLATE_report.m` to verify that it is implemented correctly. See the documentation of the function for details about inputs and outputs. Much of the code for this can be copied from the `monospinnerDynamics` function from assignment P1.

Task 3. Run the `evaluate` function to produce a `submission.json` file that certifies that the tests pass for your implementations of `lonAeroForcesAndMoments` and `aircraftDynamics`.

Task 4. Use the `estimateAlon` function to estimate A_{lon} with the trim state and control settings in `TEMPLATE_report.m`. Calculate the eigenvectors of the resulting A_{lon} matrix and indicate which of the vectors correspond to the short period and phugoid modes.

Task 5. Determine an initial state for the four-dimensional linear model with an initial pitch deviation of $\Delta\theta = 10°$ that only excites the *phugoid mode*. Use this initial deviation state and the trim state to determine a twelve-dimensional initial state for the nonlinear model that starts with the same pitch angle and minimizes short period oscillations. Simulate both the linear and nonlinear models for 50 seconds starting at these initial conditions and plot the pitch angle. Provide axis labels with units for the plots and indicate with a legend which are the linear and nonlinear results.

Task 6. Determine an initial state for the four-dimensional linear model with an initial pitch deviation of $\Delta\theta = 10°$ that only excites the *short period mode*. Use this initial deviation state and the trim state to determine a twelve-dimensional initial state for the nonlinear model that starts with the same pitch angle and minimizes phugoid oscillations. Simulate both the linear and nonlinear models for 10 seconds starting at these initial conditions and plot the pitch angle. Provide axis labels with units for the plots and indicate with a legend which are the linear and nonlinear results.

Task 7. Write three to five sentences answering the following question: For which of the two initial conditions in the previous tasks was there a greater difference between the behavior of the linear and nonlinear models? What caused this divergence?

Deliverables

In order to use the template files, rename them by removing `TEMPLATE_`. To produce the report with plots, using the Matlab command `publish('report.m', 'pdf')` is highly recommended. Submit the following files to Gradescope:

- `submission.json` (make sure that the Gradescope autograder runs successfully when you submit!)
- `report.pdf` containing required output from the tasks.
- `lonAeroForcesAndMoments.m`
- `aircraftDynamics.m`
- Any additional supporting functions you may have written.