ASEN 3728 Aircraft Dynamics
Programming Homework 1

Due date listed on Gradescope.

In this assignment, you will write a Matlab program to simulate the dynamics of a monospinner vehicle[1]. The monospinner is a flying machine with a single moving part. At one end, at a length l from the center of mass along the x axis, it has a propeller that produces an aerodynamic thrust, f_P, in the $-z$ direction and moment, τ_P about the $-z$ direction. The control input is the thrust $\mathbf{u} = [f_P]$, and the moment is proportional to this thrust: $\tau_P = k_m f_P$. Numerical values for all of the constants are contained in the code templates. The rest of the electronics are placed carefully to produce dynamics that allow the vehicle to fly with a spinning motion that can be seen in the video here: `https://www.youtube.com/watch?v=P3fM6VwXXFM`.

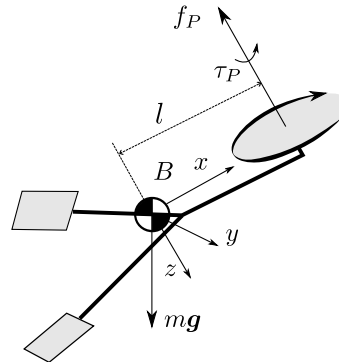

Figure 1: The monospinner vehicle.

Figure 2: Coordinates, forces, and moments.

A zip file containing all code for the assignment is also available at `https://github.com/zsunberg/Aircraft-Dynamics-Materials/raw/main/zips/assignments/P1.zip`. Alternatively, you may clone the git repository at `https://github.com/zsunberg/Aircraft-Dynamics-Materials` and navigate to the `assignments/P1` directory. It is possible that there will be bugfixes to the assignment after it is released. These will be announced on Canvas. Optionally, you can animate your simulation using the `monospinnerAnimation` function (example in `animate.m`).

Task 1. Create the `rotation321` function that returns the rotation matrix R_E^B given a vector of the Euler angles ϕ, θ, and ψ. You can test this function by running the command `run(testRotation321)` in Matlab. The file `TEMPLATE_rotation321.m` contains a template for this function.

Task 2. Create the Matlab function `attitudeInfluence321` that returns the attitude influence matrix, T, given a vector of the Euler angles ϕ, θ, and ψ. You can test this function by running the command `run(testAttitudeInfluence321)` in Matlab. The file `TEMPLATE_attitudeInfluence321.m` contains a template for this function.

Task 3. Create the Matlab function `monospinnerDynamics` that returns the state derivative $\dot{\mathbf{x}}$ given the time

t, state $\mathbf{x}$, and control input $\mathbf{u}$ [1]. You can test this function by running the command `run(testMonospinnerDynamics)` in Matlab. The file `TEMPLATE_monospinnerDynamics.m` contains a template for this function. You do not need to include aerodynamic forces or moments other than those created by the idealized rotor model described above, however, be sure to account for the fact that the force generated by the rotor is not applied directly through the center of mass.

Task 4. In Matlab, run the command `evaluate('your.gradescope.email@colorado.edu')` (replacing the email address with the one you use for Gradescope). This will run the tests on your code and produce a file called `submission.json` that certifies that your code passes the tests. You will upload `submission.json` to Gradescope to receive credit for this assignment.

Task 5. Using `ode45`, simulate the monospinner dynamics for 5 seconds with a constant thrust of 2.3 N, $\mathbf{u} = [2.3]$, starting from a zero initial condition. Plot the position, attitude, velocity, and angular rate of the monospinner over time by modifying and calling the function in the `TEMPLATE_plotStateHistory.m` file. Which angular rate initially grows the fastest? Why?

Task 6. Using `ode45`, simulate the monospinner dynamics for 5 seconds with no thrust, $\mathbf{u} = [0]$ starting from the initial condition $\mathbf{x}(0) = [0; 0; 0; \ 0; 0; 0; \ 1; 1; -20; \ 0; -5; 0]$. Plot the position, attitude, velocity, and angular rate of the monospinner over time by modifying and calling the function in the `TEMPLATE_plotStateHistory.m` file. Since the only force acting on the monospinner is gravity, it should follow a ballistic trajectory meaning that the downward velocity should increase linearly with time. Why does the velocity term w oscillate? Since there are no torques acting on the monospinner, the angular momentum should be constant. Why do the angular rate components p, q, and r oscillate?

Task 7. In the model you coded for this assignment, it would likely be impossible to stabilize the monospinner to steady state flight. What real-life forces and/or moments were not modeled that make it possible for the real monospinner to reach steady state flight? Why are they needed?

Deliverables

In order to use the template files, rename them by removing `TEMPLATE_`. To produce the report with plots and question answers, 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 plots and answers to the questions (a couple sentences each) for Tasks 5 to 7.
- `rotation321.m`
- `attitudeInfluence321.m`
- `monospinnerDynamics.m`
- `plotStateHistory.m`

References

[1] Weixuan Zhang, M. W. Mueller, and R. D'Andrea, "A controllable flying vehicle with a single moving part," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 3275–3281. [Online]. Available: http://ieeexplore.ieee.org/document/7487499/

---

[1] A better name for this function would be `monospinnerEOM`, but we did not have time to change it before releasing the assignment