Combating Aerodynamic Disturbances on a Satellite in Low Earth Orbit

ISHAAN KANDAMURI, Undergraduate, Aerospace Engineering
University of Illinois Urbana-Champaign, Illinois, USA

MICHAEL KARPOV, Undergraduate, Computer Science
University of Illinois Urbana-Champaign, Illinois, USA

KRIS ZINA, Undergraduate, Aerospace Engineering
University of Illinois Urbana-Champaign, Illinois, USA

***Abstract*—**

This project evaluates the performance limits of a four–reaction–wheel attitude controller operating on a low-Earth-orbit satellite subject to aerodynamic drag/disturbances. On top of the provided code, we implement atmospheric disturbance modeling based on the ISA. The satellite is simulated over a long time, and performance metrics include the time-to-settle and the capability to maintain pointing authority even under active disturbances. Additionally, the project aims to find regimes in which reaction wheel controls are insufficient, demonstrate the operational envelope for reaction-wheel–only attitude control in drag-dominated regimes and determine if supplementary control methods (e.g., magnetic torquers or thrusters) become necessary in lower-Earth orbits. Future work may explore more advanced aerodynamic disturbance models and additional exploration of control methods under aerodynamic disturbance.

I. INTRODUCTION

Almost all satellites on station experience aerodynamic torquing—some more than others, depending on their altitude. In low-Earth orbit (LEO), even the tenuous atmosphere imparts measurable drag and lift forces on a spacecraft's surface, generating disturbance torques that, while can be used as a source of passive control in their own right [1][2], can degrade pointing accuracy or even lead to loss of attitude control. This project builds upon the Python notebook developed in Homework 4 to investigate the altitude limits of a four–reaction–wheel attitude controller when confronted with aerodynamic torques acting on a rectangular-prism satellite.

We begin by modeling the satellite as a rigid body with principal moments of inertia aligned to its geometric axes. The aerodynamic torque is computed by simulating lift forces on each face of the prism using an altitude-dependent atmospheric density profile from the U.S. Standard Atmosphere 1976, with a similar method to Gonzales et al.[3] combined with an assumed coefficient of lift and the satellite's velocity through the residual atmosphere. These lift forces, resolved about the satellite's center of mass, produce a net disturbance torque that continually challenges the reaction wheels' ability to maintain a desired attitude.

As described in Section II, satellites are simulated through time-domain simulations spanning altitudes from 200 km, where we evaluate their settling time and ability to maintain pointing authority. The results of these simulations can not only inform satellite mission planners of the practical altitude constraints for reaction-wheel–based control but also highlight the necessity of auxiliary actuators (such as magnetic torquers or microthrusters) when operating in lower LEO. In the sections that follow, we detail the modeling approach, controller design, simulation setup, and outcome analysis that together answer the central question: "At what altitudes is a four–reaction–wheel controller still effective against aerodynamic torques?"

II. METHOD

A. Implementation Methodology

The simulation, along with all of the parameters modeled in the following subsections is implemented with Python, making use of the NumPy module for vector and matrix mathematics and the Matplotlib module for visualization. Other than hard-coded constants, no external datasets were utilized.

B. Spacecraft and Reaction Wheel

To model the dynamics of a spacecraft equipped with reaction wheels in a pyramid configuration, the total moments of inertia of the spacecraft were defined as:

$$J_B^s = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 300 \end{bmatrix} kg\ m^2 \tag{1}$$

And the moments of inertia for the reaction wheels were defined as:

$$J^{\parallel} = \begin{bmatrix} 0.01044 & 0 & 0 & 0 \\ 0 & 0.01044 & 0 & 0 \\ 0 & 0 & 0.01044 & 0 \\ 0 & 0 & 0 & 0.01044 \end{bmatrix} kg\ m^2 \tag{2}$$

In order to define the pyramid configuration of the reaction wheels, a set of unit vectors along the spin axes of the wheels in the spacecraft body frame is defined, as seen in Eq. 3. The orientation can be seen in Fig. 1.

$$W = \frac{1}{2} \begin{bmatrix} 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \\ \sqrt{2} & \sqrt{2} & \sqrt{2} & \sqrt{2} \end{bmatrix} \tag{3}$$

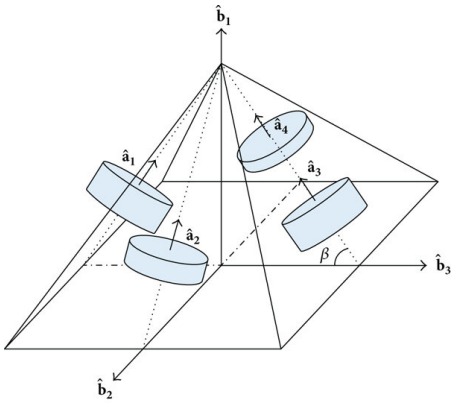

Fig. 1. Reaction wheel pyramid configuration

The effective inertia as seen in the body frame is calculated below:

$$J_B = J_B^s - WJ^{\parallel}W^T \tag{4}$$

C. Attitude Parameterization and Kinematics

The attitude of the spacecraft is represented by a quaternion, $q = [q_1,\, q_2,\, q_3,\, q_4]$, with vector part q_{1-3} and scalar part q_4. The corresponding attitude matrix, A, is:

$$A(q) = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_2q_1 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_3q_1 + q_2q_4) & 2(q_3q_2 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{5}$$

Quaternion kinematics, $\dot{q}$, follow as:

$$\dot{q} = \frac{1}{2}\begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}\omega \tag{6}$$

Where ω is the vector of the body-fixed angular velocities defined as:

$$\omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{7}$$

Where ω_1, ω_2, and ω_3 are the angular velocities along the principal axes.

D. Rotational Dynamics

The dynamics of the spacecraft were modeled with the following equations of motion:

$$\dot{\omega} = (J_B^s - WJ^{\parallel}W^T)^{-1}(L_B - WL_w - \omega \times (J_B^s\omega + WJ^{\parallel}\omega^w)) \tag{8}$$

$$\dot{\omega}^w = (J^{\parallel})^{-1}L_w - W^T\dot{\omega} \tag{9}$$

Where L_B is a vector of the external torques and L_w is a vector of the wheel torques. These are implemented in a function that calculates the dynamics of the system. This function returns $(\Delta t)(\dot{\omega})$ and $(\Delta t)(\dot{\omega}^w)$ for use in the Runge-Kutta time integrator, which is detailed in Subsection G.

E. Aerodynamic Torques

In order to model aerodynamic forces acting on the satellite body, the drag equation was utilized:

$$F_d = \frac{1}{2}\rho v^2 C_d(A\cos\alpha) \tag{10}$$

Where ρ is the density of air (as dependent upon altitude), v is the circular orbital speed, C_d is the drag coefficient, A is the reference area of the flat plate surface on which the drag is acting, and α is the angle of incidence. The circular orbital speed is obtained through Kepler's Third Law, as derived below:

$$v = \sqrt{\frac{GM}{r}} \tag{11}$$

Where GM is the standard gravitational parameter for a celestial body and r is the orbital radius. For Earth, $GM = 3.986 \times 10^{14}\, m^3/s^2$. The orbital radius was calculated by summing the radius of the Earth, $6.378 \times 10^6\, m$, with the orbital altitude being tested.

The drag coefficient was taken to be 2.2, which is typical for flat plates. The reference area was taken to be 1.0 m^2. In order to determine the angle of incidence, a unit vector for the free stream velocity in the inertial frame was defined as $u_I = [-1, 0, 0]$. This unit vector indicates that the flow is directed towards the negative x-direction, opposite the velocity. The normal vector of the body x-face, the surface upon which the drag force is assumed to act, was defined as $n_B = [1, 0, 0]$. Finally, in order to calculate the contribution of the drag force as based on the satellite's orientation, a dot product was used as follows:

$$\cos\alpha = u_b \cdot n_B \tag{12}$$

Within the simulation loop, the following logic was utilized to make sure that the drag force only affects the satellite when facing the flow:

Algorithm 1 Incidence Angle

if $\cos\alpha > 0$ **then** ▷ If plate is facing flow
  $A_{eff} = A * \cos\alpha$ ▷ Projected area
  $F_{mag} = \frac{1}{2}\rho v^2 C_d A_{eff}$
  $F = -F_{mag} * u_B$ ▷ Drag opposes stream
else
  $F = 0$

Finally, in order to calculate the torque applied on the satellite due to the aerodynamic force, the following product is utilized:

$$L_B = r_B \times F_d \tag{13}$$

Where $r_B = [L_{cube}/2, 0, 0]$ is the moment arm for the torque, where $L_{cube} = 0.10\, m$, a value consistent with the dimensions for a 1U CubeSat.

F. Controller Design

A Proportional-Derivative (PD) Controller generates reaction wheel torques and its purpose is to drive the current quaternion, q, to the command quaternion, $q_c = [0, 0, 0, 1]$ (which is constant), and drive the angular velocity ω to zero. To achieve regulation control, an error quaternion, δq, can be defined. q represents the rotation from the inertial frame to the current body frame, q_c represents the rotation from the inertial frame to the desired body frame, and δq represents the rotation from the body frame to the desired body frame. It is also important to note that δq is assumed to be the shortest quaternion. The error quaternion is calculated as follows:

$$\delta q = q_c \otimes q^* \tag{14}$$

Where q^* is the conjugate of the current quaternion, defined as:

$$q^* = \begin{bmatrix} -q_{1:3} \\ q_4 \end{bmatrix} \tag{15}$$

Eq. 14 is expanded as:

$$\delta q = \begin{bmatrix} q_{c4} q^*_{1:3} + q^*_4 q_{c1:3} - [q_{c1:3}\times]\, q^*_{1:3} \\ q_{c4} q^*_4 - q_{c1:3} q^*_{1:3} \end{bmatrix} \tag{16}$$

Where the skew-symmetric matrix $[q_{c1:3}\times]$ is defined as:

$$[q_{c1:3}\times] = \begin{bmatrix} 0 & -q_{c3} & q_{c2} \\ q_{c3} & 0 & -q_{c1} \\ -q_{c2} & q_{c1} & 0 \end{bmatrix} \tag{17}$$

The equation for the controller is as follows:

$$L_w = W_p[(-\omega_B^{BI} \times H) + k_p sign(\delta q_4) J_B q_{1:3} + k_d J_B \omega_B^{BI}] \tag{18}$$

Where W_p is the Moore-Penrose inverse, or the pseudoinverse, of W and H is the total angular momentum, expressed in the body frame. The pseudoinverse is calculated as follows:

$$W_p = W^T (W W^T)^{-1} \tag{19}$$

And the total angular momentum is calculated as follows:

$$H = J_B^s \omega_B^{BI} + W J^{\parallel} \omega^w \tag{20}$$

The controller also implements a saturation check to ensure that torque is not continuously applied when the wheel spin is near its maximum rate. This algorithm is shown below:

Algorithm 2 Saturation Limit

```
limit ← 1570                          ▷ Saturation limit, rad/s
tolerance ← 100                       ▷ 100 rad/s buffer
threshold = limit − tolerance
for each wheel do
    if abs(current speed) ≥ threshold then
        if sign(command torque) = sign(speed) then
            command torque ← 0
        end if
    end if
```

During the duration of the simulations, the aerodynamic torques are constantly affecting satellite body. The controller is activated halfway through the total time, at which point the torques are actively suppressed, and the controller actively attempts to drive the angular velocities to zero and the quaternion to the command quaternion.

G. Numerical Integration

In order to solve the nonlinear equations for the kinematics and dynamics, a fixed-step, fourth-order Runge-Kutta (RK4) numerical integrator is utilized with a time step of $\Delta t = 0.05s$. The ordinary differential equations being solved are those for $\dot{\omega}$, $\dot{\omega}^w$, and $\dot{q}$. Define $F(x) = [\dot{\omega}, \dot{\omega}^w, \dot{q}]$, where x, the state vector, is defined as $x = [\omega, \omega^w, q]$. Thus, the logic for the time integrator is as follows:

$$\begin{aligned} k_1 &= F(x_n) \\ k_2 &= F\left(x_n + \frac{\Delta t}{2} k_1\right) \\ k_3 &= F\left(x_n + \frac{\Delta t}{2} k_2\right) \\ k_4 &= F\left(x_n + \frac{\Delta t}{2} k_3\right) \\ x_{n+1} &= x_n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned} \tag{21}$$

Where x_n is the value at the current time step and x_{n+1} is the value at the next time step. After each update, the quaternion is normalized as:

$$q_{n+1} = \frac{q_{n+1}}{||q_{n+1}||} \tag{22}$$

III. PRELIMINARY RESULTS

A. Simulation Setup

The simulation was created as an extension of Homework 4. The key difference is that three different simulations were created for a torque-free scenario, an aerodynamic disturbances scenario, and finally a scenario where the PD controlled reaction wheels combated these aerodynamic disturbances. The second two scenarios had three sets of simulations for the three various altitudes of 250km, 500km, and 800km. The atmospheric densities are summarized in Table I.

Altitude (km)	Atmospheric Density (kg/m^3)
250	5.97×10^{-11}
500	4.76×10^{-13}
800	6.95×10^{-15}

TABLE I
Atmospheric Densities

These densities are used to calculate the magnitude of the drag force that causes an external torque on the body of the satellite.

B. Results and Analysis

The plots below depict the changes in the attitude of the CubeSat, its angular velocity, and its reaction wheels spin rate throughout the simulation. At each altitude, there is a plot where only disturbances are active, and one where there is active control induced from the reaction wheels to combat these disturbances. Additionally, at 250 km a plot depicting the change in the 3-2-1 Euler angles of the CubeSat is given in the disturbances scenario and the controlled scenario.

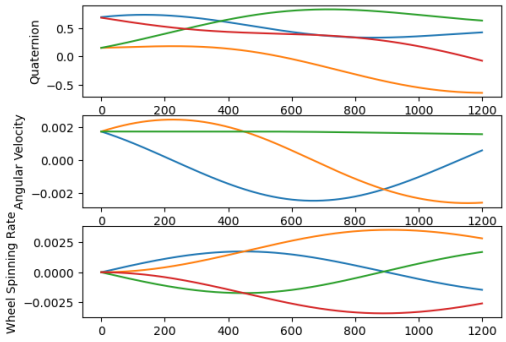

Fig. 2. Time vs Affected Satellite Parameters with Disturbances at

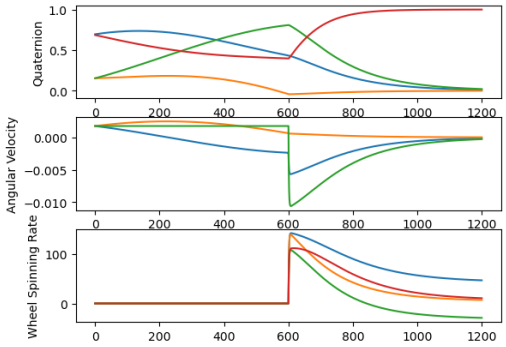

Fig. 3. Time vs Affected Satellite Parameters Controlled at 250 km

From the above plots, it is evident that the controller is effective at using the reaction wheels to stabilize and correctly orient the CubeSat. The CubeSat reaches the desired command quaternion of $[0, 0, 0, 1]^T$, and all angular velocity components reach zero. Additionally, once stability has been achieved, the spin rate of the reaction wheels also goes to 0.

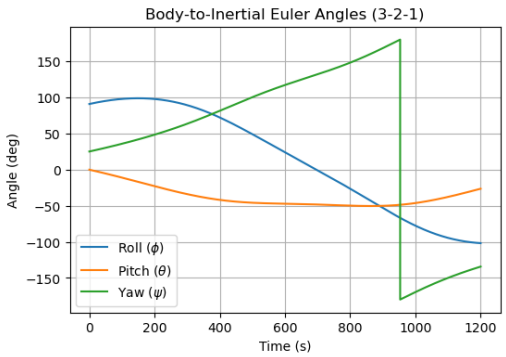

Fig. 4. Time vs 3-2-1 Euler Angles of Satellite with Disturbances at 250 km

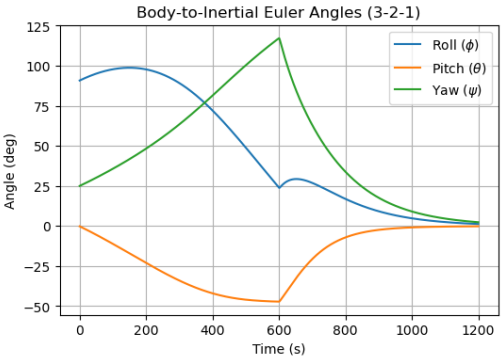

Fig. 5. Time vs 3-2-1 Euler Angles of Satellite Controlled at 250 km

From the plots above, it is evident that the orientation of the satellite is also corrected and stabilized by the end of the simulation as all Euler angles asymptotically reach 0 degrees, and the controller maintains this attitude once it has been achieved.

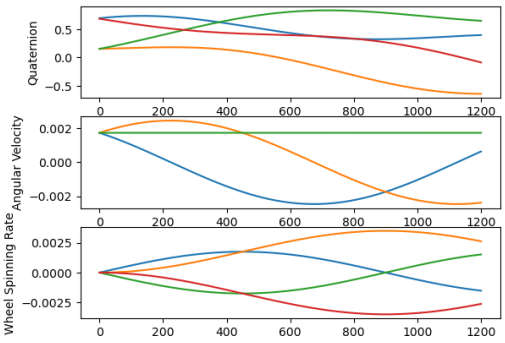

Fig. 6. Time vs Affected Satellite Parameters with Disturbances at

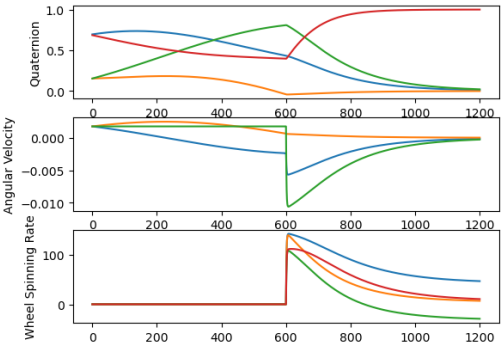

Fig. 7. Time vs Affected Satellite Parameters Controlled at 500 km

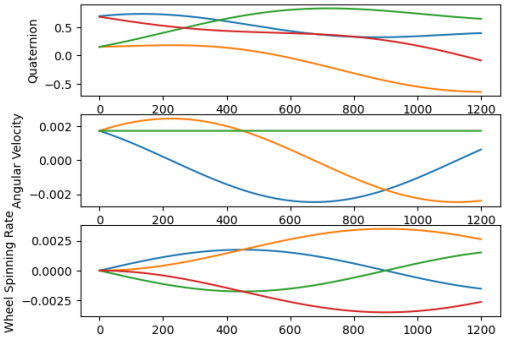

Fig. 8. Time vs Affected Satellite Parameters with Disturbances at 800 km

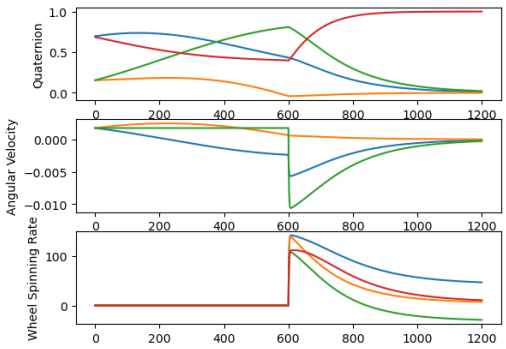

Fig. 9. Time vs Affected Satellite Parameters Controlled at 800 km

From the Figures 6, 7, 8, and 9, it is evident that at all three altitudes the satellite can be controlled to set orientation with the same reasoning as for the 250 km scenario. This proves that our controller is effective at all three chosen altitudes in which the CubeSat can potentially be in low-Earth orbit.

IV. CONCLUSION AND FUTURE WORKS

This project evaluated how well a simple four–reaction–wheel controller can keep a rectangular-prism satellite pointed in the presence of aerodynamic torque in low-Earth orbit.

These results show that reaction-wheel-only control is effective down to an altitude of at least 250 km. Furter experimenting could involve analysing conditions at altitudes lower than 250 km to get a more accurate range for the controller's effective region.

A. Future Work

To build on these findings, the following steps are recommended:

1) **Add more disturbance sources:** Include gravity-gradient, solar radiation pressure, and magnetic torques for a more realistic model.
2) **Try different controllers:** Compare LQR or adaptive controllers to see if they handle drag better.
3) **Momentum management:** Simulate using magnetic torquers or small thrusters to unload wheel momentum before saturation.
4) **Hardware testing:** Use a hardware-in-the-loop setup or digital twin simulation for better "real-world" simulation.

ACKNOWLEDGMENT

We would like to thank Hongrui Zhao, Brandon Crane, and the University of Illinois at Urbana-Champaign Aerospace Department for a fun and challenging final project!

REFERENCES

[1] Rawashdeh, Samir, and Lumpp, James Jr.
Aerodynamic Stability of Cubesats at ISS Orbit
JoSS, Vol. 2, No. 1, pp. 85-104 JoSS Database.

[2] Gonzales et. al
Modelling and Simulation of Very Low Earth Orbits
EUCASS, 8 DOI: 10.13009/EUCASS2019-176.

[3] Rawashdeh et al.
CubeSat Aerodynamic Stability at ISS Altitude and Inclination
AIAA Conference on Small Satellites, 2012 PDF.