Systems Space Program: Modelling Satellites with System Dynamics

MAE 3260 Final Group Work

By: Brianna Cheng, Owen Dankert, Joshua Delaney, Matthew Steiglitz

Topic of Interest: Satellites

Abstract:

This project focuses on studying a satellite by breaking it down into a system that has attitude control, reacts to disturbances, and uses actuators and sensors to control its attitude. Since the actual dynamics involved are very complicated, a simplified model will be developed based on topics discussed in class so far. The reasons for focusing on satellites are that they are fascinating, important, and are complicated enough to make for an interesting and educational project. To simplify the model, the satellite's rotation will be restricted to a single axis (similar to the rotational systems studied in class). Using what has been learned through the course of the class about PID controllers (including ODEs and TFs), actuator saturation, and steady state, a system-dynamics model of the satellite analyzing the behavior can be created. This model will also incorporate active disturbance rejection focusing on system response and performance requirements including rise time, settling time, and maximum overshoot.

Student	Task/Role + Takeaways
Joshua DeLaney	Worked on implementing the transfer function into MATLAB/Simulink in order to produce desired plots of the system
Brianna Cheng	Worked on researching and studying how different sensors and actuators contribute to the overall workings of a satellite
Owen Dankert	Worked on disturbance modeling and I learned a lot about how many different factors can cause disturbance within a satellite system.
Matthew Steiglitz	Worked on system model (plant, PI controller,

| | satellite), also helped with actuator section. I learned about how satellites use control theory for efficient and effective attitude control. |

List of concepts/skills used:

- ODEs, rigid body dynamics, OL system, CL system, PI controller, Matlab, Simulink, system response and plots (free response, step response), response parameters (ess, tr, ts), disturbance rejection, saturation & anti-windup

Modelling the Satellite Using System Dynamics

Matthew Stiglitz mbs324

OL System

Feedback Control

$R \rightarrow (+ -) \rightarrow [C] \xrightarrow{U} (+ +) \xrightarrow{T_{net}} [P] \xrightarrow{\ddot{\theta}} [1/s] \rightarrow (+) \rightarrow [1/s] \rightarrow (+) \xrightarrow{Y}$

PI control = no D control or filter

$R = \theta_r$
$Y = \theta$
$E = \theta_r - \theta$
$C = k_p \left(1 + \frac{1}{T_i s}\right) \leftarrow$ PI controller
$T_{ACS} = CU, \quad T_D = D$ (note: T is torque, not time constant, if it has a subscript)
$T_{net} = T_{ACS} + T_D \leftarrow$ Generated by ACS + disturbance torque
Rigid-body dynamics: $T = I\dot{\omega} = I\ddot{\theta}, \quad \ddot{\theta} = T/I$
$P = 1/I$
$\ddot{\theta} = P T_{net}$
$\dot{\theta} = \int_0^t \ddot{\theta} dt + \dot{\theta}_0 = \frac{\ddot{\theta}}{s} + \frac{\dot{\theta}_0}{s} \leftarrow$ ACS changes the $\omega = \dot{\theta}$ at the $= \dot{\theta}$
$\theta = \int_0^t \dot{\theta} dt + \theta_0 = \frac{\dot{\theta}}{s} + \frac{\theta_0}{s}$ rate $\ddot{\theta}$ using torque T

Simplified 1D satellite model:
Satellite rotates about cylinder axis
θ is output
T_{ACS} is input that controls $\omega = \dot{\theta}$
by varying $\ddot{\theta} = \dot{\omega}$

ω, T_{net} , θ

Note: ACS can saturate T, leading to suboptimal performance.
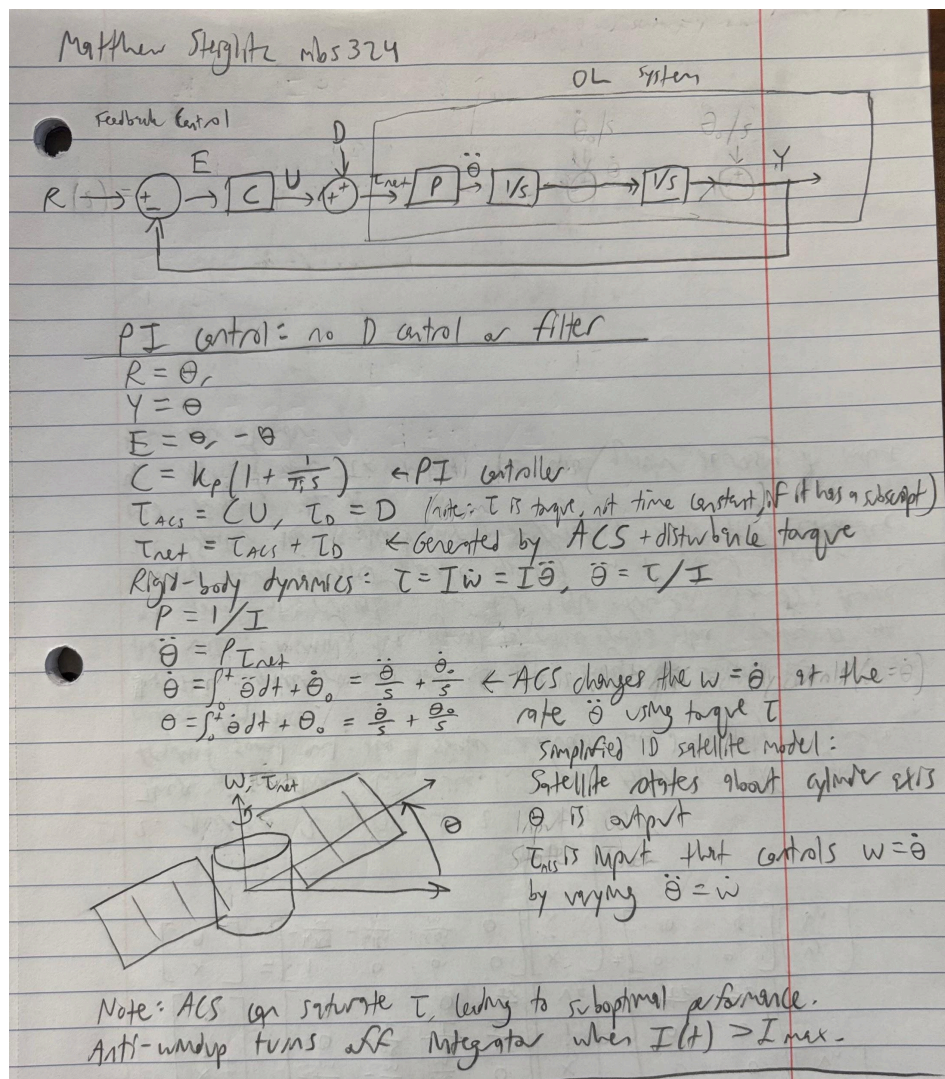Anti-windup turns off integrator when $I(t) > I_{max}$.

Figure 1: Model of Satellite

Torque is generated by the ACS (attitude control system = actuators) in order to change w; this relationship is governed by the satellite's rigid-body dynamics (plant). Output theta is measured by sensors and fed back into the feedback controller (electronics). We model this by integrating angular acceleration twice with initial conditions to extract angular position and feed it back into the controller.

We will use a PI controller in order to have zero steady state error. This is important for high-precision pointing applications often seen in satellites (telescopes, earth surveillance, communications, etc). However, we are keeping our model simple. We did not add derivative control because satellites mostly use PI controllers [2]. While the Lab 4 system included an electronic filter [7], we did not include this in our model. We did include disturbances and disturbance rejection because satellites often encounter disturbances from particles, solar wind, atmospheric drag, gravity, etc. We will be focusing on performance parameters of steady state error, rise time, and settling time. Note that there is no maximum overshoot in a PI system. We will try to avoid saturation by selecting system parameters carefully. We can also implement anti-windup to turn off the integrator if it exceeds a maximum value.

Our satellite can only rotate about one axis. In this model, it has angular velocity $w = \theta'$, angular position θ (measured by sensor/s), and moment of inertia I about the spin axis. The ACS, a system of actuators, exerts torque τ on the system. We assume that this axis is a principal axis (no products of inertia) and aligns with the intended spin axis, though these cannot be perfectly aligned in real life (resulting in nutation and coning). While an actual 3D satellite would want to change the position of its angular momentum vector, our satellite can only rotate about one axis (in order to avoid tedious and complicated math beyond the scope of this course) and change the magnitude of its angular momentum vector (and therefore the magnitude of w).

We will simulate the system's response to open loop (with different ICs) and step input. There is negligible damping in space. With atmospheric drag, if the net drag force's line of action does not pass through the center of mass, a disturbance torque will be generated. There is also technically a damping effect on the satellite's rotation, but this is negligible. We assume b = 0 in order to simplify the system.

We can leave results in terms of variables, but if we want to plot any results, we must either normalize the plots or choose plausible values. We opted for the latter. There are many different types of satellites of different sizes and masses, from tiny nanosats to large space

stations. Let us consider the Earth observation satellite Landsat 8. It weighs approximately 2000 kg with a 2.4 m diameter and 3 m length [8]. It also has one 9 x 0.4 m deployable solar array [9]. For the satellite in our model, let's say that the solar panels are light enough to have a negligible effect on I. Triple Junction GaAs Solar Cells like those used in the Landsat 8 have a density of around 0.086 g/cm^2, so these panels would weigh around 3 kg each and total solar panel I would be less than 100 kgm^2 and can therefore be neglected, since these figures are approximate anyways.

Solid cylindrical satellite has $I = \frac{1}{2}mr^2 = \frac{1}{2} * 2000 * (1.2)^2 = 1440 \ kg \cdot m^2$ about its spin axis (vertical axis).

Typical CMG saturation values vary widely depending on the size, geometry, hardware, etc. Looking at the CMGs offered on satsearch [10], a typical CMG for a medium to large satellite would saturate on the order of 10-100 Nm. In our case, let's say that our CMG or other ACS saturates at around 50 Nm.

Given this information, we can make transfer functions. However, because the system is not a second order system with damping, we cannot analytically find ess, tr, and ts to select parameters. Instead, we will find these values numerically from the system response plots. Let's say that precision is important and we want ess = 0 (possible with integrator). We also want to reasonably quickly improve performance and earnings. If we are moving at max torque the whole time (assuming instantaneous acceleration and deceleration), it will take the satellite around 20-30 seconds to complete a 2π rad rotation. This is a slower time scale than the systems we have been working with so far due to the large moment of inertia of the heavy satellite. This gives us a reference point for plotting the system response.

We used Simulink to test the model, specifically utilizing the code from the week 11W group work. As seen in Figure 2 below, the overall model structure was not altered, but a test disturbance of 0.2 was added. In addition, the plant was changed to match the OL system seen in Figure 1 and the controller (seen in Figure 3) and the proportional control (P) in the PI Controller was set equal to $\frac{1}{I}$.
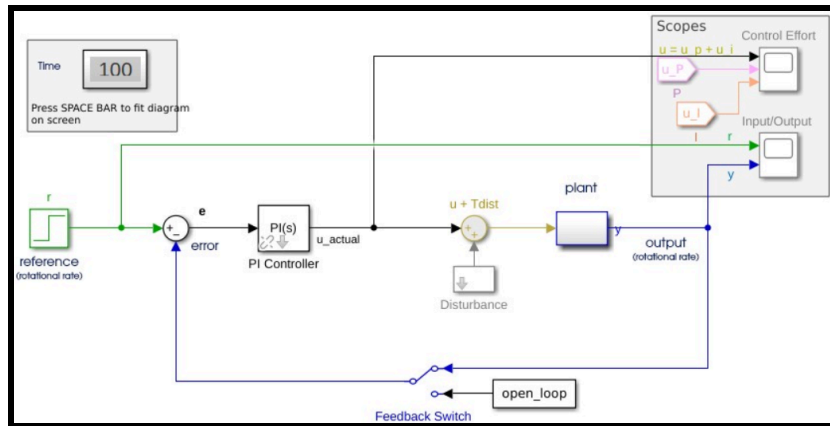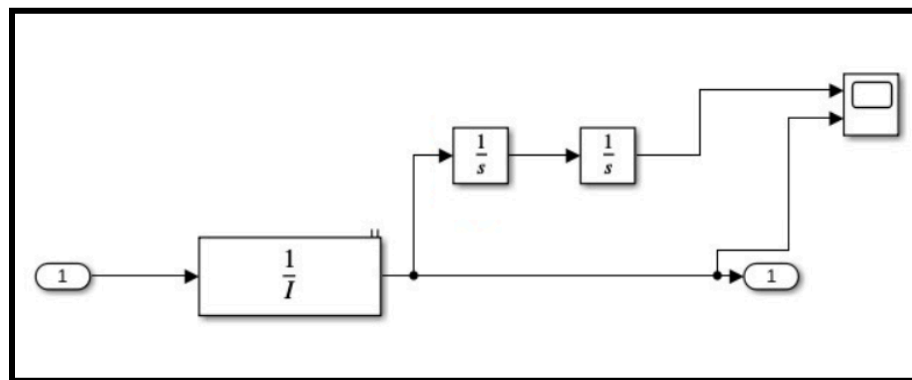
Figure 2: Overall Model of structure

Figure 3: Controller

Using $I = 1440 \ kg \cdot m^2$, $r_0 = 0$, and $r_{final} = 50$ we got the following plots:
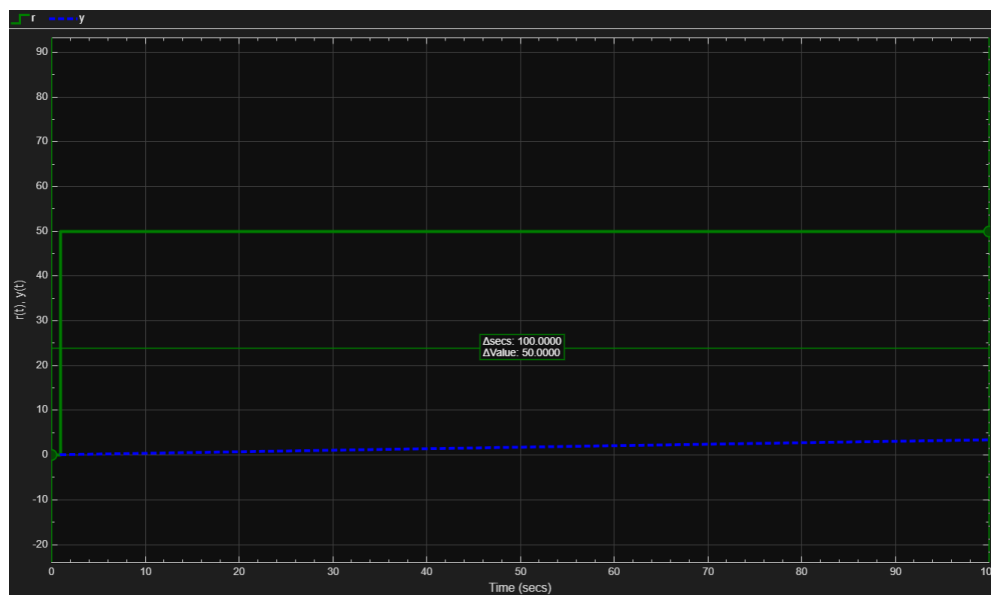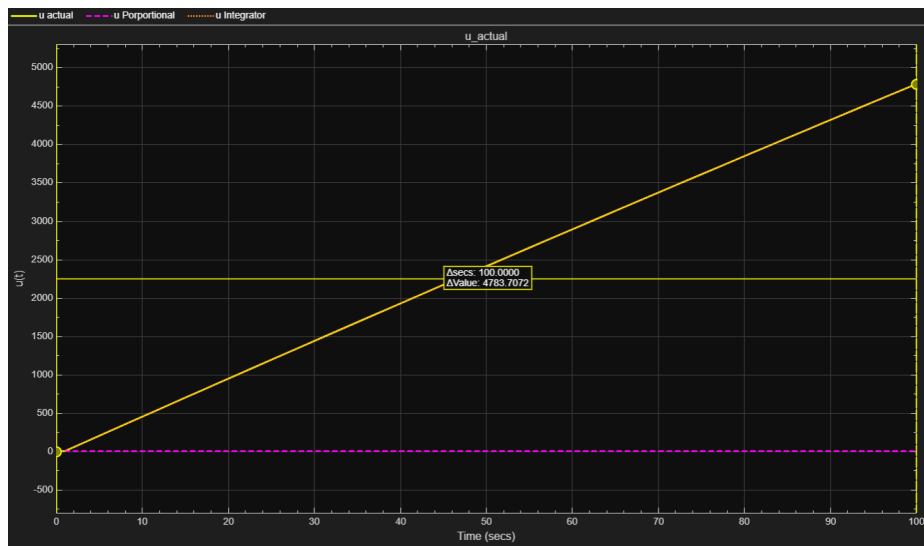

Figure 4:

Figure 5:

We were unable to produce a plot that showed a desired result of a steady state error of zero. For a typical PI controller, the plots should resemble those found when performing the week 11 group work (see Figures 6 and 7). Alterations were made to the initial and final reference values as well as changing the amount of time the simulation would run in order to fix the issue. Unfortunately, we were unable to fix the issue and the cause of this error may have been a result of incorrect implementation of the simulink (especially when considering our lack of experience with Simulink).
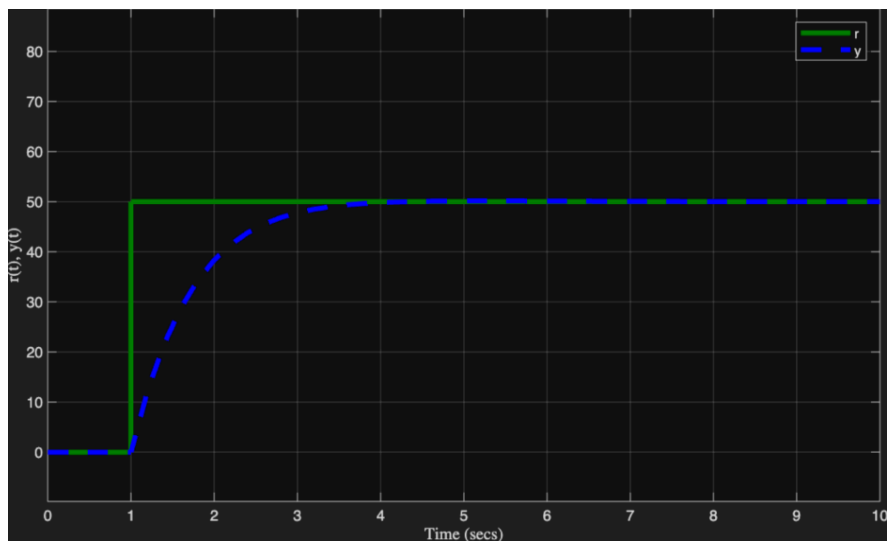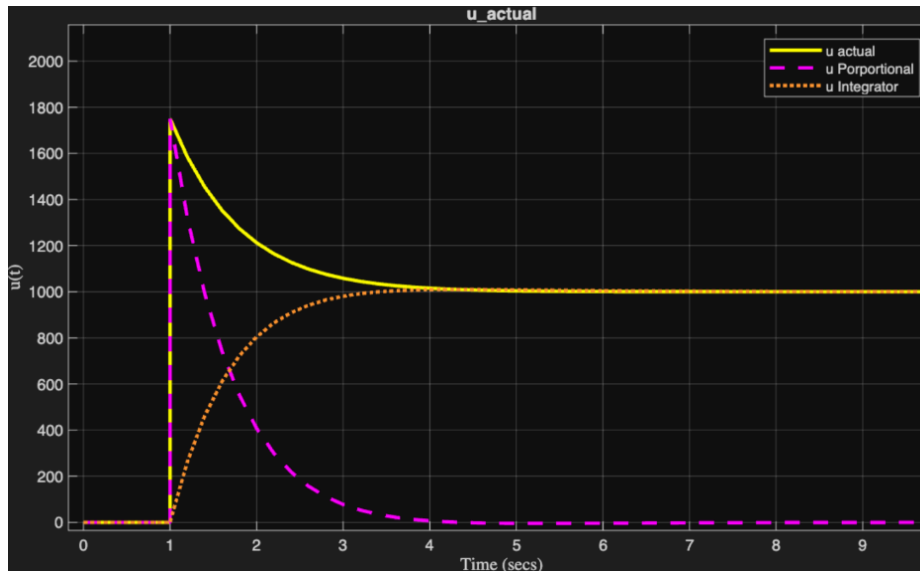

Figure 6:

Figure 7:

Disturbance Modeling

There are a couple of things that can cause disturbances in space, the two main ones being debris collisions and atmospheric drag. When it comes to atmospheric drag, this is a disturbance torque, and the calculations shown in Figure 8 show an estimate for what the torque from atmospheric drag would be.

$$F_D = \rho U^2 A \, C_D$$
$$= (4 \times 10^{-12})(7500)^2 (24.3)(0.8)$$
$$= 0.6006 \, N$$
$$R = 1.2$$
$$\boxed{D = R \, F_D = 0.00078 \, Nm}$$

Figure 8:

Our rough estimate for the disturbance torque was around 0.00078 Nm; however, this would likely depend on the rotation speed of the satellite when the attitude is being adjusted. As shown in the model in Figure 1, this torque would be applied during the D part of the block

diagram, which is where the disturbance is accounted for. In the case of our satellite, this disturbance torque would be relatively small due to the thin atmosphere in LEO; however, it is still important to consider.

Disturbance from collisions in this case would happen when some sort of debris hits the satellite, causing its attitude to change. We would likely model this as an impulse, as it would be a near-instantaneous collision that applies an impulse to the satellite. Figure 9 shows a calculation of the change in angular velocity using the mass of the colliding particle, the particle's velocity, and the distance from the satellite's center of mass at which the collision occurs.

$$M_d = \text{debris mass}$$
$$V_d = \text{debris velocity}$$
$$r = \text{collision distance from COM}$$

$$\Delta L = I \Delta \omega$$
$$\Delta \omega = \frac{M_d V_d r}{I}$$

$$J_d = M_d V_d$$

$$I = 1440 \text{ kg m}^2$$

$$L_d = J_d r = M_d V_d r$$

$$\boxed{\Delta \omega = \frac{M_d V_d r}{1440 \text{kg} m^2}}$$

Figure 9:

As seen from this formula, the change in angular velocity is proportional to the particle mass and velocity, along with the collision distance from the center of mass. To see roughly the effect that this would have on the satellite, a sample calculation is provided in Figure 10 showing how a sample particle with a mass of 0.01 kg, a velocity of 10,000 m/s, and a collision distance of 1.2 m away from the center of mass would affect the angular velocity.

sample calculation:
Particle with 0.01 kg & Velocity of 10000 m/s
assume r = 1.2 m

$$\Delta \omega = \frac{(0.01 \, kg)(10000 \, m/s)(1.2 \, m)}{1440 \, kg \, m^2}$$

$$\boxed{\Delta \omega = 0.083 \, rad/s}$$

Figure 10:

Assume that our satellite was initially at rest and at the correct attitude. This collision would cause an instantaneous increase in the angular velocity of the satellite to around 0.083 radians per second, which would cause the satellite to move out of the correct position. This is the main way these collisions would occur, allowing the system designed earlier to kick in to move itself into the correct position again, hence why it is so important to consider collisions such as these when modeling a satellite.

Different Types of Sensors:

Star Trackers:

Star trackers are attitude sensors used to help keep track of spacecraft and their special whereabouts. By capturing images of stars, mapping landmarks, and matching those landmarks to a known catalog of stars, they easily determine the orientation of a spacecraft. The device starts off by identifying bright points in an image and then computing the coordinates of the star with respect to the spacecraft. Then the device matches and cross references the gathered geometries with information in a star catalog to identify the star. After this, the tracker computes the attitude measurement and as a result can provide precision-pointing data.

Magnetometer:

Similar to a star tracker magnetometers are another type of attitude sensors used to estimate the spatial orientation of a spacecraft. Instead of using images of stars, a magnetometer

measures the local magnetic field and compares it to Earth's magnetic fields. These sensors become more reliable when optical sensors such as star trackers become unusable.

Sun sensor:

Sun sensors are another kind of attitude sensor that measures the direction of the sun with respect to a spacecraft to help with precision pointing. By analyzing sun vectors or angles of the sun, the sun sensor is able to compute the direction of the sun in the frame of reference of the spacecraft providing one axis of orientation. Coupled with other devices the sun sensor can provide all three axes. Sun sensors are very important components in a satellite, but their only limitation is that they cannot work without the sun. So, when the sun is blocked for any reason, the sun sensors may not work.

Referenced data from [11] and [12] for summaries of all three sensors

Different Types of Actuators

Reaction Wheels:

RWs are simple (especially math-wise), cheap, and lightweight but use much more energy to change angular momentum than CMGs. RWs work by changing the magnitude of angular momentum h and therefore angular velocity w for constant I. [3]

Control moment gyroscopes (CMG):

CMGs change the direction (but not magnitude) of angular momentum by rotating a fast-spinning disc about the disc's transverse axis. The energy required to rotate the disk about its transverse axis is quite low (theoretically 0 if friction is ignored) since work $W = \int \tau \cdot d\theta$, torque vector which rotates the disc about its transverse axis is perpendicular to the disc's angular displacement vector (which is parallel to its angular momentum vector). If work is 0, then power is 0 after the disk has spun up and there is no friction. Very efficient! This is why CMGs are good. If you want to change the magnitude of the angular momentum like a RW but more efficiently, you can arrange a pair of CMGs such that the horizontal component of the angular momentum vectors cancels out and net angular momentum is in the desired direction.

The magnitude of this angular momentum can be efficiently changed by rotating the CMGs equally away from or towards the shared axis. [1 & 2]

For these reasons, it is best to use a paired CMG array in our model. However, our model works with any attitude control system (ACS) that controls torque by varying w at rate commanded by the feedback controller.

Thrusters:

Thrusters are simplest of all, but are quite inefficient compared to the other actuators. Thrusters exert an external moment on the spacecraft (nonzero net moment), while RWs and CMGs only exert an internal moment, and the net moment is zero (Newton's Third Law). The thrusters are expelling gas, which releases energy from pressure and/or combustion. The gas pushes on the thruster, exerting an external torque on the spacecraft. (We are considering only torques because forces are irrelevant to our model.)

References:

[1] MAE 4060 Lecture 18: "Spacecraft Rigid-Body Dynamics," Mason Peck

[2] MAE 4060 Lecture 20: "Attitude Kinematics: Attitude Sensors Determination," Mason Peck

[3] MAE 4060 Lecture 21: "Momentum-Actuator Technologies," Mason Peck

[4] CharmLab Stanford Engineering, Attitude Control by Daniel Washington and Nabil Wessel

https://charm.stanford.edu/ENGR1052016/DanielWashingtonNabilWessel

[5] MAE 3260 Lecture 11W, "Integral PI Control and Windup," Mark Campbell

[6] MAE 3260 Lecture 11F, "PID Control," Mark Campbell

[7] MAE 3260 Lab 4 Manual, Mark Campbell

[8] Landsat 8 by Landsat Missions

https://www.usgs.gov/landsat-missions/landsat-8

[9] Nasa Landsat 8

https://science.nasa.gov/mission/landsat-8/

[10] Control Moment Gyroscopes by Satsearch

https://blog.satsearch.co/2024-01-10-control-moment-gyroscopes

[11] Space System Design, MAE 342, Princeton University, Robot Stengel

https://stengel.mycpanel.princeton.edu/MAE342Lecture13.pdf

[12] Attitude Determination and Control Systems by Scott R. Starin, NASA Goddard Space Flight Center John Eterno, Southwest Research Institute

https://ntrs.nasa.gov/api/citations/20110007876/downloads/20110007876.pdf