Session 4 - Problem Statement

Anirudh Agarwal

November 22, 2023

1 Objectives

The objectives of this session are:

1. Complete the code and search for session 3 based on the solutions provided.
2. Create a github account and upload the codes made so far including the pdf files of your solutions on github.
3. Find orbital speed at various orbit heights and ground speed for various latitudes.
4. Find contact information of the suppliers and service providers for the given encapsulants.
5. Find reaction wheel balancing services in Singapore (and India).
6. Write a function to give orbit speed with height.
7. Write a function to get angular speed and angular acceleration at each point in the horizon.
8. Get max torque and max angular momentum for 10 MOIs at 10 heights and make a 3D plot.

2 Search pointers

2.1 GitHub

Check out my github profile and see how I have uploaded the Session 2 and Session 3 problem statements and solutions. Use the same problem statement pdfs but put your own pdf solutions. Create individual github profiles and each person in a team can upload the pdfs as well as the codes. Going forward, explore github more and check out github co-pilot which is freely available now. See if you can use it to help you code based on the problem statement given.

Absolutely try it with session 4 and session 5 codes and see if you can save any amount of time or effort by using this, given the extremely detailed problem statements I provide. Do share how co-pilot or another AI or any other tool helped you improve your code efficiency and made you more productive. That will be extremely valuable going forward in your career as you will be able to outperform your peers very significantly by leveraging the rapidly changing landscape of the best tools available.

We see that github is a platform primarily for code. So this platform may not be the best for the searched information even though pdfs can be uploaded.

So, check out other portfolio websites where you can start to create your profile. Here are some lists:

- List of sites by hubspot: https://blog.hubspot.com/marketing/free-portfolio-websites
- List of sites by Wix: https://www.wix.com/blog/best-portfolio-websites

You can play around with various options to see which site you are most comfortable with and make your profile. With companies like wix and squarespace, you can even make your own website which has more detailed information about you and a variety of documents and uploads. This is above and beyond what you can do with LinkedIn and your resume which everyone already has. This will help you stand apart.

2.2 Orbit speeds and ground speeds

Find graphs that show relationships between speed and height and gives equations as available. This source gives a good idea of speed vs height and provides the equation for it. Write more details about what kind of satellites (GPS, GEO etc) are typically put at what height.

Similarly, for the ground speed, find graphs and images that include equations. This source for instance, shows how speed changes with latitude with a diagram of the earth rotating. However, the units are in km/h and we prefer SI units m/s. Find other sources, or just convert yourself.

2.3 Encapsulants

There are generally 2 types of encapsulating material usually used, PVA and Polyolefin. Polyolefin itself have 2 types, elastomer and thermoset. You can look through the following links and companies and contact them if you want, I am tied down by my EPS.

You need to know that the encapsulant will experience degradation, what kind of degradation and effects you need to know before you can think of a way to counter it. One of the factors is UV degradation, there are others that you need to check. Main result of degradation is the yellowing of the encapsulant.

- https://multimedia.3m.com/mws/media/1216185O/3m-solar-tech-data-sheet-po8110.pdf
- https://us.mitsuichemicals.com/service/product/solarasce.htm
- https://www.dow.com/en-us/brand/engage-pv.html
- https://www.borealisgroup.com/industries/energy/solar-encapsulant
- http://www.ever-thriving.com/productshow.asp?id=30
- https://www.hbfuller.com/en/north-america/products-and-technologies/markets-and-applications/new-energy/encapsulants

These encapsulats are an alternative to the adhesives we are searching for, to cover the solar panels. Thus, the same issues of transparency, being space grade, and degradation apply. Do check if the material is listed on the Jaxa or the other materials site shared previously.

If the material has been used for encapsulating solar cells previously for satellites, that would be perfect.

2.4 Reaction wheel balancing

We need reaction wheels for the ADCS. The wheel geometry and the box size will be determined by the requirements of the wheel balancing service. I have found the following services:

1. Dynamic Laser Solutions: https://dynamiclasersolutions.com/
2. Dynamic balancing and alignment: https://www.uni-drive.com/dynamic-balancing-and-alignment/

It is not clear where they are and whether they provide the balancing service or just sell the balancing machines. We are looking for wheel balancing service providers that can help balance our reaction wheels with the following properties:

1. Mass: 50 - 500g range. Closer to 150g.
2. RPM: 10,000 - 20,000 range. Close to 15,000 RPM

When searching use the terms "reaction wheel balancing" or "dynamic wheel balancing" to avoid car wheel balancing services showing up in the search results. Car wheels are much heavier and much less precise in the alignment. We need a service designed for very light wheels like we would have in our ADCS system.

Note that the reaction wheels are rigid, so the test can be done at much lower RPMs (<1000 RPM) as the wheel does not significantly deform with speed. If the wheel balancing services do not have the 15k RPM option, 1K RPM also works.

3 Fundamentals of the code

This time the code is much more straightforward than ever before. Less input data involved and less overall data handling. The purpose of doing these calculations is to size the ADCS motors and reaction wheels to determine what to put in our ADCS system. The ADCS helps point the spacecraft and it should be able to maintain a lock with a fixed point on the ground so the spacecraft can do its communication. To maintain this lock as the spacecraft is moving in its orbit and the ground is moving underneath, and the craft needs to continuously rotate to keep facing the same ground station. This means it needs to be able to hit a max rotation speed and a max rotation acceleration to maintain the lock. That is what we calculate here. In the final session, we use these values to size the wheels and choose the motors.

3.1 Inputs to the code from text files

The first input file would be for the earth parameters and constants used throughout. The inputs will be:

1. Rearth: Earth radius (m)
2. Mearth: Earth mass (kg)
3. Gconst: Gravitational constant (SI units)
4. Tday: Earth rotation time period/ length of day (seconds)

The only other inputs needed by the functions are:

1. Horbit: Orbit height (km)
2. moi: MOI of the craft (SI units)

Everything else will be directly calculated through the code. We fix the number of steps to a 100 for the arrays.

3.2 Intermediate values to be calculated and stored

The intermediate values that come up in the process are:

1. Vorbit: orbit velocity for given height
2. Hangle: angle for horizon in degrees
3. Thorizon: time in the horizon of the ground station
4. theta[]: angle from the x axis that locates the craft
5. elevationAng[]: elevation angle as seen from the ground station as a function of theta[]
6. deltaT: time difference between 2 time steps
7. Wcraft[]: angular velocity of the craft (array for values from horizon to apex)
8. maxW: maximum angular momentum hit given by: max(Wcraft)
9. alpha[]: angular acceleration of the craft (array for values from horizon to apex)
10. maxAlpha: maximum angular acceleration hit given by: max(alpha)

3.3 Outputs to be written in text files

The output values are just 2:

1. torque: the torque the spacecraft needs to apply to rotate (torque = moi*maxAlpha)
2. Pcraft: angular momentum of the craft (Pcraft = moi*maxW)

However, these need to be plotted for 10 different mois and 10 different heights to cover a range of values and make a relatively smooth 3D plot for torque as well as for Pcraft.

Sure these are the main values, however, the main objective here is to get the minimum and maximum torque and angular momentum values to get the range of capabilities for our ADCS system which is to be designed in the final session. Thus, additional outputs to save would be::

1. min and max torque
2. min and max angular momentum

4 Diagrams and Equations

4.1 Figures and Variables Labelled

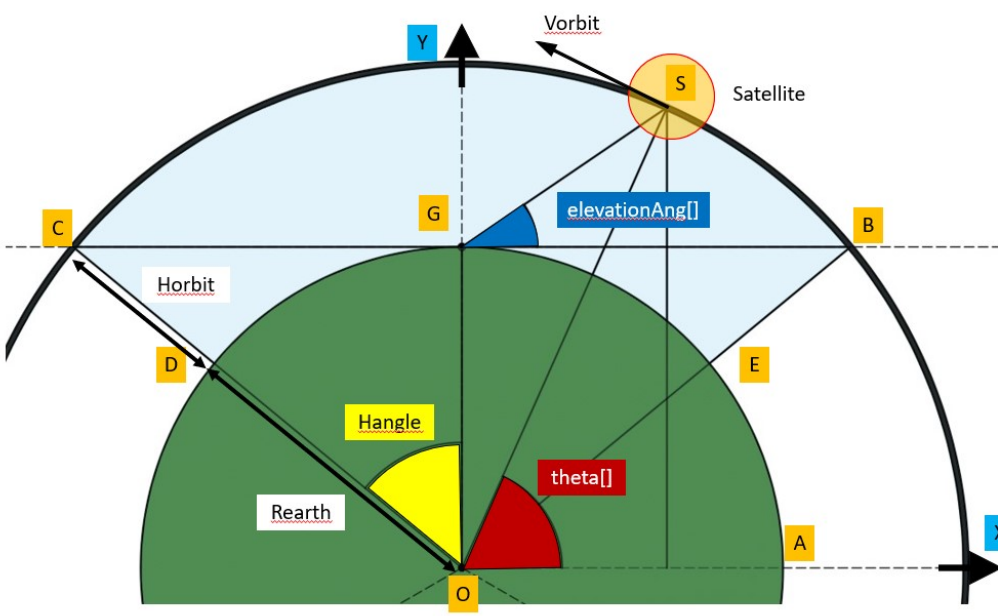

Figure 1: Figure showing the craft going over the horizon

Here, we see the black line, arc BSC as the orbit of the satellite. The satellite itself is at point S moving with velocity Vorbit. The height of the orbit above the earth determines this orbit's velocity. G is the ground station with which the satellite needs to maintain contact. The satellite is constantly pointing in the direction SG to maintain the lock with the ground station.

The line BC is the line of the horizon. The satellite is only visible to the ground station if it is above BC. Thus, the satellite must be pointing in the direction BG before coming up to the horizon at point B and continue to point along SG (toward G) until it goes out of the horizon at point C.

When this pointing starts, theta[0] is angle AOB which is $\pi/2 - Hangle$. Then it goes to $pi/2$ when S crosses the Y axis. The motion on the other side of the Y-axis is identical to the motion before so we only need to calculate until it hits the vertical. During this time, the elevationAng[] goes from 0 to $\pi/2$ radians as the craft goes from being just above the horizon to being directly overhead.

In the worst-case scenario, the ground would be moving in the exact opposite direction to the craft giving us the minimum possible time in the horizon and consequently demanding the largest possible acceleration. That means that the craft is rotating in the arc BSC and the ground is rotation in the arc DGE. The sum of these angular velocities about the center point O and the Z axis in figure 1 gives us the net angular velocity for the craft so we know how quickly it passes from horizon to horizon. In the frame of the earth, we add the earth's angular velocity to the satellite and the diagram still holds. Only the Vorbit is different, which is accounted for in the new angular velocity.

4.2 Equations for intermediate values

Equation for Vorbit:

$$Vorbit = sqrt\left(\frac{Gconst \cdot Mearth}{Rearth + Horbit}\right) \tag{1}$$

Equation for Hangle:

$$Hangle = \arccos\left(\frac{Rearth}{Rearth + Horbit}\right) \tag{2}$$

This will give Hangle in radians. To write to the text file, use degrees. For calculations, keep all angles in Radians so that trigonometric functions and inverse functions can be used.

Equation for Thorizon:

$$Thorizon = \frac{2 \cdot Hangle}{\frac{Vorbit}{Rearth+Horbit} + \frac{2\pi}{Tday}} \tag{3}$$

The worst-case scenario is when the earth is rotating in the opposite direction to the direction of the orbit. It is unlikely but possible. This also accounts for situations when the satellite orbit may be at an angle to the rotation of the earth or even perpendicular. Thus, for system design, we go for the worst-case scenario. The time in the horizon is given by the angle to be traversed divided by the net angular velocity. The angular velocity of the craft about the center of the earth is v/r. The angular velocity of the earth is $2\pi/timeperiod$.

Calculation for theta array:

We want to get acceleration at 100 time steps from horizon to apex. For that, we need to have 101 speed values and in turn, 102 angle values. Thus, the angle array would have 102 values.

$$theta[i] = (\pi/2 - Hangle) + \frac{Hangle \times i}{101} \tag{4}$$

Where i goes from 0-101 to get 102 values. This way, when i is zero, theta is exactly at the horizon and when i is 101, theta is 90 degrees from the horizontal, aka, vertical. Hangle is measured from the vertical, thus the 90-Hangle to exactly match the horizon.

Equation for deltaT:

$$deltaT = Thorizon/202 \tag{5}$$

There are 101 chunks of time in between 102 steps of theta. These are symmetrically spread on both sides of the horizon. Thus, there are 202 chunks of time from horizon to horizon.

The time stamp for theta is $time[i] = theta[i] * deltaT$ since the orbital speed is constant and it takes the same amount of time to travel through the same angle.

Equations for elevation angle array:

$$elevationAng[0] = 0 \tag{6}$$

$$elevationAng[101] = \pi/2 \tag{7}$$

$$elevationAng[i] = arctan\left(\frac{(Rearth + Horbit) \cdot \sin(theta[i]) - Rearth}{(Rearth + Horbit) \cdot \cos(theta[i])}\right) \tag{8}$$

Where i goes from 1 to 100. Python may not play well with exactly 0 and 90 degrees so we set the elevation angle for those values separately.

Equation for Wcraft:

$$Wcraft[i] = \frac{elevationAng[i+1] - elevationAng[i]}{deltaT} \tag{9}$$

Where i goes from 0 to 100 giving 101 values of Wcraft. They start from the position theta[0.5] and go to theta[100.5] or from time deltaT*0.5 to deltaT*100.5 seconds.

When writing the code, I realise that the max angular momentum is reached only when the craft is directly overhead. Thus the equation for Wmax would be:

$$Wmax = \frac{2 \cdot Hangle \times (Rearth + Horbit)}{Thorizon \times Horbit} \tag{10}$$

This is because the velocity of the craft relative to the ground station G is given by the angular velocity about the center of the earth ($2 * Hangle/Thorizon$) times the radius of the orbit ($Rearth + Horbit$). The angular velocity about the point G (ground station) is given by this relative velocity divided by the distance from G to the orbit when the craft is directly overhead i.e. Horbit.

Similar to Wcraft[i], the equation for alpha is:

$$alpha[i] = \frac{Wcraft[i+1] - Wcraft[i]}{deltaT} \tag{11}$$

Where i goes from 0 to 99 giving 100 values for alpha. They start from theta[1] and go to theta[100] or time deltaT to 100 deltaT. This way we can tell at what position and time do we hit the maximum speed and acceleration respectively, if needed.

Once we have the arrays, the maximum values can be captured using the max() function and stored. Plots can be made for elevation angle,

Note: the equations for the output values are given at the definition. The main aspect of this problem statement is the 3D plot and the 100 values being calculated.

5 Tips for making the code

- Think carefully about the data structure and define classes and arrays according to the goal of making the 3D plot. Look at the inputs of the 3D surface plot function and decide accordingly.
- Sometimes the values become too small or too large in SI units. Do all the calculations in SI units but when writing in the output files, try CGS units or simply mNm for torque instead of Nm.
- Note that all these equations cannot be used as is. In order to do a 100 instances with different moi and Horbit values, all these equations need to be implemented as functions which can be called in a nested loop for various MOI and height values.
- Keeping that in mind, think about what class objects to declare to make arrays if any. How can the code be modularized to make it more efficient? Does AI/github co-pilot have any suggestions?