Session 5 - Problem Statement

Anirudh Agarwal

March 20, 2024

1 Objectives

The objectives of this session are:

1. Complete the code and search for session 4 using the solutions provided. There are several leaps taken in determining the data structure of the outputs and a lot more things are outputted. Best to look through the solution to get a sense of what needs to be done first.
2. Get VSCode on your computer for python since VSCode is compatible with github and has more extensions. Install Code GPT VScode extension and use it to enhance your coding experience and speed.
3. Find basic explainers on what an ADCS subsystem in a satellite is and does.
4. Find papers and industry standard references showing the capabilities of ADCS for cubesats of various sizes.
5. Find explainers on what magnetorquers are and how are they used.
6. Make a table with ADCS capabilities. This could include: torque, angular momentum, size and mass of ADCS system, size and mass of craft, MOI of craft, height of orbit and any other relevant information.
7. Write a program for reaction wheel calculations including the following functions:
   (a) Given a box size, get the maximum size of reaction wheels
   (b) Given a reaction wheel and box size, get the max angular momentum stored and return factor of safety for angular momenta of the crafts
8. Write a program for magnetorquer calculations including the following functions:
   (a) Given a box size, get the number layers and turns used and wire length to fill completely.
   (b) Given layers, turns, and current, get the magnetic moment
9. Using these 2 programs' outputs, write a program to compare the capacity of each ADCS box to the angular momenta of the crafts and the given magnetorquer requirements to show which boxes can be used up to what range.

2 Search pointers

2.1 ADCS - what is it and why?

ADCS is Attitude Determination and Control System. It is a subsystem in the spacecraft that allows the spacecraft to point in one direction. This website gives more details about what the ADCS is and how satellites use it.

As a communication satellite, we need to point to a ground station. This is affected by the orbit, speed, and the MOI. MOI is mostly dependent on the solar panels since they have the largest dimension and the MOI is proportional to the 4th power of the length for a given density. Thus, we had to do the solar panels mechanism first to get the MOIs, then get the angular velocity and accelerations, to finally determine now what is needed for the ADCS. **Everything is connected!**

2.2 Papers for ADCS specs

You can search ”Attitude Determination and Control System” key words on Mendeley and see what shows up for ADCS. Select the ”open access” option on Mendeley to make sure you can download the pdf of the paper and share it in the Google drive. At first, you will find generic ADCS papers like this paper [3] on the reliability of ADCS or this paper [1] for the ADCS of a 6U CubeSat.

Go through the references of the papers and searching specifically for ”review papers on adcs capabilities for cubesat” on Google scholar, you should find papers like this [4] that cover the capabilities of a vareity of ADCS systems used in space. From the papers, gather the maximum angular momentum, torque, reaction wheel size, total mass, total size and/or any other features listed in various tables and make a comparison table to showcase what is out there in the market and what has been previously used in space.

2.3 Magnetorquers

Magnetorquers are a part of the overall ADCS system. They are basically just electromagnets that generate a magnetic field that we can control, which interacts with the earth’s magnetic field and gets rid of the excess angular momentum of the craft. This magnetorquer system available online shows what it looks like and what it can do. There are many others like this that can be found online. Check out this IMTQ magnetorquer and this CubeSatshop Magnetorquer rod that showcase the capabilities. Search for options like this that are found online and make a table for the length, mass, number of turns, voltage, current, power, and/or any other features available.

However, it will be difficult to get much information from suppliers. Find what you can. But focus on the papers, just like the ADCS. Search ”magnetorquer design” on google scholar and on Mendeley and find papers like [2] this one which is about design for attitude control using just magnetorquers. From all the papers you find, make a table comparing the capabilities and specs as mentioned before.

3 Fundamentals of the code

Use Code GPT extension and try out the paid version of GPT-4 and GitHub-copilot to get the syntax for various code. If you find existing libraries for certain functions, see if you can use those instead of some of the functions I give.

3.1 Inputs to the code from text files

3.1.1 Previous code outputs

The first input file would be the minmaxTP from the solution of session 4. I have re-done the design and found that the MOI values are over double of what was used previously, so I will use the newer values for the code.

We get a range of values from minmaxTP. Create an array of 5 values linearly distributed throughout that range as Pvals = np.linspace(minp,maxp,5). Here, minp and maxp are the minimum and maximum angular momenta given in the minmaxTP output from session 4. The variable will be Pvals[i] where i goes from 1-5.

3.1.2 Given inputs

We are restricted by the maximum size of the ADCS box. We take that as an input and make the ADCS capacity accordingly. The ADCS box sizes are $side = \{11, 12, 13, 14, 15\}$ in cm. The magnetorquer requirements don’t have a straightforward calculation we could find. So, the requirements go as $moment\_needed = \{0.2, 0.3, 0.4, 0.5, 0.6\}$ in Am^2 . This is what the moment must be compared against for the range of angular momenta being considered.

3.1.3 Motor and structure inputs

These inputs are about the motor speed, size, and some structure constraints:

1. RPM: Motor RPM = 6000 (speed in rotation per minute)

2. hwheel: wheel thickness (Length of motor shaft) (cm) = 0.97 cm
3. rho: density of the reaction wheel material (Aluminium 2.7 g/cc)
4. clwheel: clearance between the wheels (cm) = 0.53 cm
5. clwall: clearance between wheel and wall (cm) = 0.15 cm
6. Rmot: Radius of the motor (cm) = 1.5 cm (including mounting cover)
7. Lmot: Length of the motor (cm) = 2 cm (from wheel base to end)
8. gmot: minimum gap needed between motors (cm) = 1 cm

3.1.4 Magnetorquer specs

These are the requirements for the magnetorquer only:

1. Rcore: radius of the core (cm) = 0.475 cm
2. Dmagcover: Diameter of magnet cover setup (cm) = 2 cm
3. tstruc: thickness of structure around the magnet (cm) = 0.2 cm
4. Lcap: Length of the cap for ends (cm) = 0.3 cm
5. Voltage: voltage used for power input (V) = 3.3V
6. Uperm: Magnetic permeability of the core material (dimensionless) (700 for Stainless Steel 410)
7. rhocore: density of core material (Stainless Steel 410: density 7.85 g/cc)
8. twire: coil wire thickness/ diameter (cm) = 0.02cm. Use this wire current reference link for wire properties.
9. Imax: maximum current allowable for the wire (A) = 0.088 A
10. wireResist: resistance of the wire per unit length (ohm/m) = 0.556 ohm/m

We max out the magnetic moment for a given wire constraint and compare the magnetic moments for different thicknesses.

3.2 Intermediate values to be calculated and stored

3.2.1 Reaction wheel values

These are the values related to the reaction wheel part of the program:

1. Rwheel: wheel radius (cm)
2. Mwheel: mass of the wheel (g)
3. MOIwheel: moi of wheel (gcm^2)
4. maxPwheel: maximum angular momentum of the wheel (gcm^2rad/s). Match units with craft angular momentum to calculate FOS.
5. clstruc: clearance from the structure when wheel is attached. (cm) This should be more than clwall.

3.2.2 Magnetorquer values

These are the values related to themagnetorquer part of the program:

1. Lcore: length of the core (cm). The portion of the core wrapped with wire.
2. turnsPL: number of turns per layer. $turnsPL = Lcore/twire$
3. resNeeded: resistance needed (ohm): $resNeeded = Voltage/Imax$
4. minWlen: minimum length of the wire based on resistance (m): $minWlen = resNeeded/wireResist$
5. numLayers: number of layers of loops
6. wlen: length of wire to cover Lcore length a whole number times (m)
7. Iactual: actual current at the given wlen.
8. Nturns: number of turns total: $Nturns = numLayers * turnsPL$
9. Nd: demagnetization factor (dimensionless)

Moment is an output value, but it can be calculated together with all these values as the class object created for the magnetorquer related values should also have the moment.

I suggest making one class for objects that contain information about the reaction wheel aspects, and one class for objects containing magnetorquer related information. Having such classes allows us to have dedicated functions to calculate all the values for each class object as an internal object function. Thus, we don't have to make different arrays for every single variable. We can just make one object array and each object contains all the relevant variables to do the necessary calculations.

Build the classes in a way that the common variables are declared as separate global variables in the main code while the variables specific to only reaction wheel or only magnetorquer calculations are in dedicated classes. This helps to streamline the code.

3.3 Outputs to be written in text files

Text outputs:

1. maxPwheel: maximum angular momentum that can be stored in a wheel of max possible size spinning at max given RPM (gcm^2rad/s)
2. moment: magnetic moment of the magnetorquer (Am^2)
3. FOSrw[]: factor of safety for angular momentum for each box size (reaction wheel FOS): $FOSrw[i] = maxPwheel[i]/Pvals[i]$
4. FOSmag[]: factor of safety for magnetic moment for each box size: $FOSmag[i] = moment[i]/moment\_needed[i]$
5. MinRPM: the minimum RPM the wheel can run at to still store pvals[i] moment of inertia. (RPM): $MinRPM[i] = RPM/FOSrw[i]$

Note that Rwheel and MOIwheel are also important for the design but those are already recorded in the intermediate values.

Plots:

1. Reaction wheel angular momentum stored and reference values vs box size.
2. Magnetic moment and reference values vs box size

There can be 4 of these to differentiate from the wheels for 4 wire sizes used.

4 Diagrams and Equations

4.1 Figures and Variables Labelled

The wheels are attached to a skeleton with 3 axes pointing out. The motor goes into the skeleton and the wheel is attached on top of that.

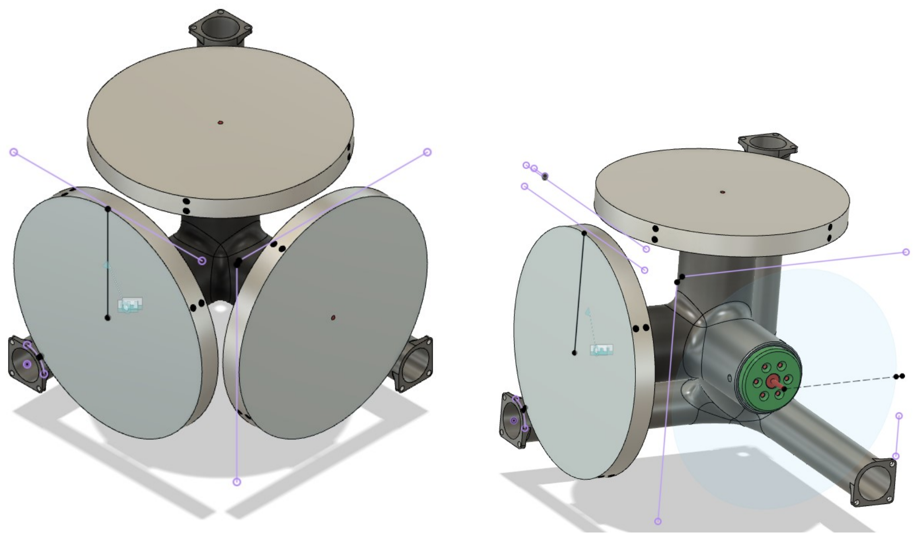

Figure 1: Visualizing the attachment of wheels

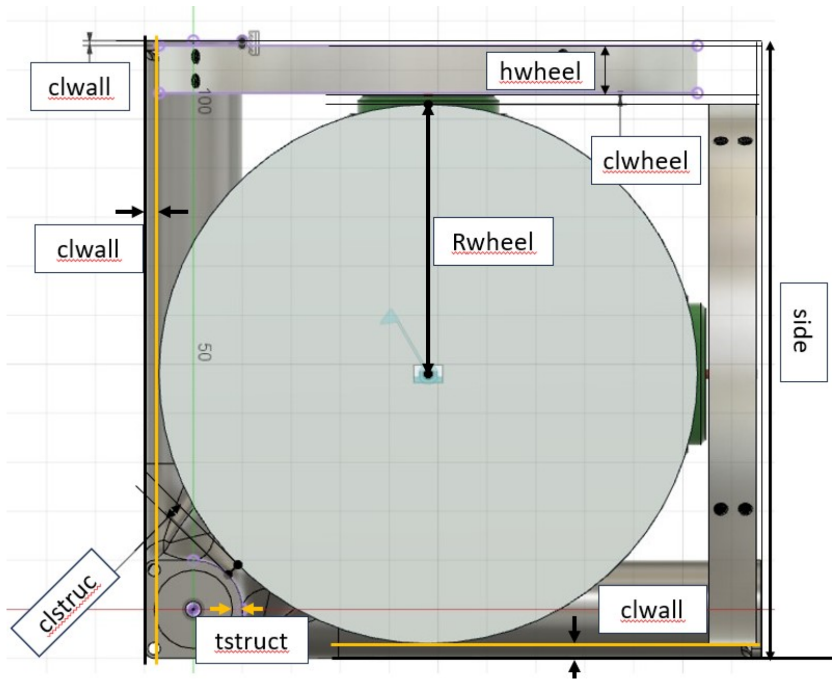

Figure 2: Dimensions and variables for reaction wheel calculations

Figure 2 shows how the wheel is spaced away from the walls and how the side and the clearances are defined.
From this we can see that $2 \cdot (clwall + Rwheel) + hwheel + clwheel = side$.
This gives $Rwheel = (\frac{side - hwheel - clwheel}{2}) - clwall$

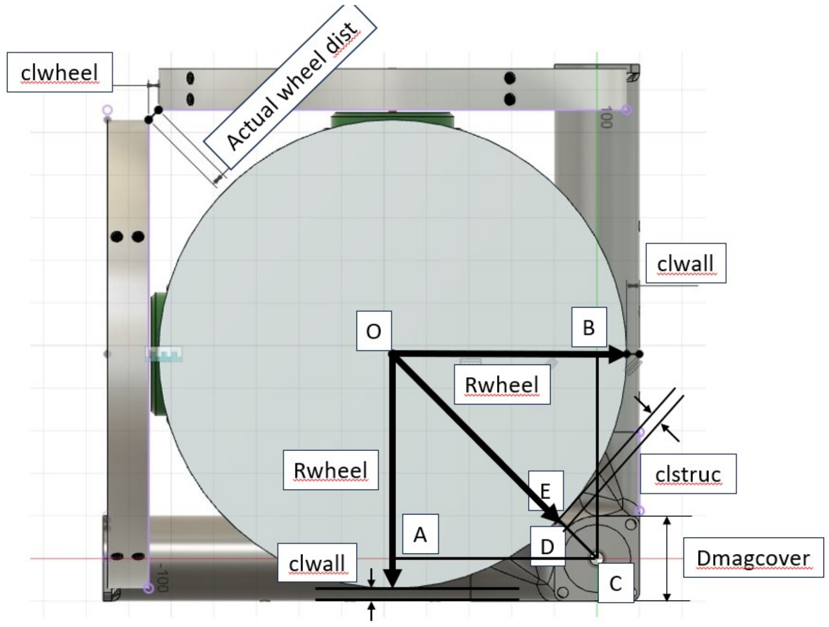

Figure 3: Dimensions and variables for reaction wheel calculations

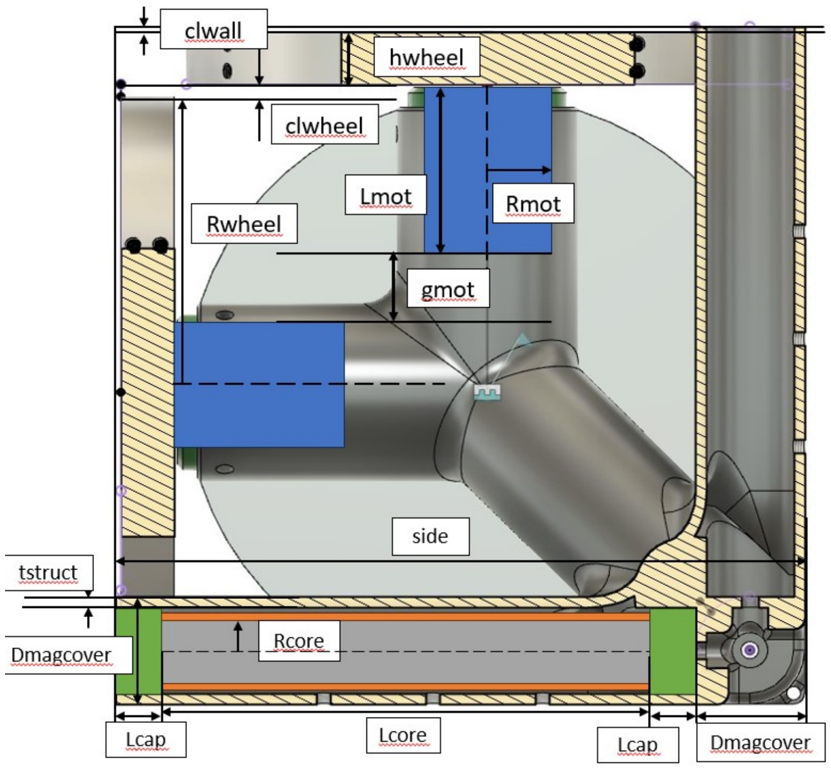

Figure 4: Dimensions and variables for magnetorquer calculations and min Rwheel

Figure 3 shows the calculation for the clearance of the wheel from the structure, clstruc. We have $OA = OB = BC = AC = (Rwheel + clwall - Dmagcover/2)$
Since OC is the hypotenuse, we get: $OC = (Rwheel + clwall - Dmagcover/2)\sqrt{2}$ and $OE = Rwheel$ and $CD = Dmagcover/2$ and $clstruc = ED$ so,
$clstruc = (Rwheel + clwall - \frac{Dmagcover}{2})\sqrt{2} - (Rwheel + \frac{Dmagcover}{2})$

Figure 4 shows the dimensions for the magnetorquer calculations and the ones for the minimum Rwheel size.
For the magnetorquer, we see that: $side = 2 * Lcap + Lcore + Dmagcover$ giving $Lcore = side - (Dmagcover + 2 * Lcap)$
The variable tstruct is input but not used in this code since Dmagcover already includes it.
For the minimum Rwheel calculation, we see that at minimum: $Rwheelmin + clwheel > Lmot + gmot + Rmot$ so, $Rwheelmin >= Lmot + gmot + Rmot - clwheel$

We check for this in the code, and get the minimum possible size. Given Rwheelmin is 3.97cm, we can ignore this since all wheel radii were greater than 3.97cm.

4.2 Equations for intermediate values

4.2.1 Reaction wheel values

Equation for Rwheel:

$$Rwheel = \left(\frac{side - hwheel - clwheel}{2}\right) - clwall \tag{1}$$

Equation for Mwheel:

$$Mwheel = rho \cdot \pi \cdot Rwheel^2 \cdot hwheel \tag{2}$$

This is because the mass is density times volume and the volume of a cylinder is $V = \pi R^2 H$.
Equation for MOIwheel:

$$MOIwheel = \frac{Mwheel \cdot Rwheel^2}{2} \tag{3}$$

This is the reference to the MOI of the cylinder (disk) that is the wheel.
Equation for maxPwheel:

$$maxPwheel = MOIwheel \times \frac{RPM \cdot \pi}{30} \tag{4}$$

This is because: $RPM/60 = RPS | RPS * 2pi = rad/s | RPM * pi/30 = rad/s$
Equation for clstruc:

$$clstruc = (Rwheel + clwall - \frac{Dmagcover}{2})\sqrt{2} - (Rwheel + \frac{Dmagcover}{2}) \tag{5}$$

If $clstruc < clwall$ then clearances need to be re-evaluated. Just output a warning if this turns out to be the case. Maybe suggest using a smaller Rwheel reduced by 2mm or 0.2cm.

4.2.2 Magnetorquer values

Equation for Lcore:

$$Lcore = side - Dmagcover - 2 \times Lcap \tag{6}$$

Equation for Nd:

$$N_d = \frac{4(\ln(\frac{Lcore}{Rcore}) - 1)}{(\frac{Lcore}{Rcore})^2 - 4 \cdot \ln(\frac{Lcore}{Rcore})} \tag{7}$$

Since the difference between r and Rcore is not significant and all the magnetic field is mostly concentrated in the core anyways, we simplify and use only Rcore for Nd.

Code for numLayers:
We start with resNeeded = Voltage/Imax and get $minWlen = 100 \times resNeeded/wireResist$ (cm). Then we see how many layers of loops it takes to surpass this length. In each layer, we have the same number of loops as turnsPL remains constant.
In each layer, the radius of the loop increases a tiny amount. This changes the length of a single loop in every layer. Thus, for each layer j (1-n), we update wlen as:
$wlen = 2\pi r \times turnsPL$
$numLayers = numLayers + 1$
$r = r + twire/2$

numLayers starts at 1 and r starts at Rcore+twire/2. In each loop we check if $wlen >= minWlen$ and as soon as that is true, we get the number of layers and the total length wlen.
The total number of loops is: $numLoops = numLayers \times turnsPL$

Equation for Iactual:

$$Iactual = \frac{Voltage \times 100}{wlen \times wireResist} \tag{8}$$

The 100 is there since wlen is in cm and wireResist is in ohm per meter.

4.3 Equations for output values

4.3.1 Magnetorquers

Equation for moment:

$$moment = numLayers \times \pi Rcore^2 \times turnsPL \times Iactual \left(1 + \frac{uperm - 1}{1 + (uperm - 1) \cdot Nd}\right) \tag{9}$$

This is the total magnetic moment generated by the magnetorquer which we can compare to the reference values for the moment.

5 Tips for making the code

- Think carefully about the data structure and define classes and arrays according to the math involved and how things are calculated. We can make functions to work for a given box size and use those repeatedly for each size. However, for the magnetorquer, another set of functions are needed for each layer.
- 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 just mm as needed.
- Consider how to record various values especially the values to be used for the plots and decide the data structure and classes accordingly.
- 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?

References

[1] V.-D. J. J. J. E. O. R. A. A. M. J. F. C.-L. P. V. H. . C. A. Cortiella, A. 3cat-2: Attitude determination and control system for a gnss-r earth observation 6u cubesat mission. *European Journal of Remote Sensing, 49. https://doi.org/10.5721/EuJRS20164940*, 2016.

[2] R.-B. N. P. M. M. R. . P. J. Jovanovic, N. Design of magnetorquer-based attitude control subsystem for foresail-1 satellite. *IEEE Journal on Miniaturization for Air and Space Systems, 2(4). https://doi.org/10.1109/jmass.2021.3093695*, 2021.

[3] A.-T. A. Mansouri, A. A mathematical model to optimize the reliability of the satellite attitude determination and control system. *Scientia Iranica, 29(6 E), https://doi.org/10.24200/sci.2020.56030.4518*, 2022.

[4] V.-T. O. P. L. A. G. R. . P. E. Rassõlkin, A. Adcs development for student cubesat satellites - taltech case study. *Proceedings of the Estonian Academy of Sciences, 70(3), 268–285. https://doi.org/10.3176/PROC.2021.3.06*, 2021.