

# Development of detumbling prototype for a 1U CubeSat

# 1. Introduction

CubeSat is a miniaturized version of a satellite that is composed of cube units (U) of 10 cm by 10 cm by 10 cm. Government Engineering College Barton Hill is developing this 1U CubeSat. Its mission is to serve as a platform for university students to learn to design and develop a 1U CubeSat bus capable of being reused in subsequent missions. A key system in this bus is the Attitude Determination and Control System (ADCS). The ADCS is responsible for determining and controlling a satellite's orientation in orbit. One of its first requirements is to reduce the rotation imparted by the CubeSat's deployer into a more stable motion, a pointing operation often referred to as detumble. This detumbling process can be managed either passively or actively. Passive control mechanisms are simple, often requiring no moving parts or power. For example, a passive control system could use permanent magnets or hysteresis rods to magnetically control orientation. This has several major drawbacks such as a limited attitude pointing accuracy of about  $\pm 10$  degrees about Earth's magnetic field, constraining all other pointing requirements for power, communications and other sensors [1]. Another approach uses active control mechanisms like reaction wheels or magnetorquers to control the spacecraft's attitude and orientation. While somewhat more complicated, these systems allow the satellite to be more precisely controlled. This cubesat's planned ADCS will utilize an active control system using magnetorquers, also called torque rods, to manage detumble and some pointing requirements. This paper will lay out the design, assembly and testing of these torque rods. Add intro here

## 2. Cubesat simulation

Before developing flight ready CubeSat, it is necessary to simulate the CubeSat in a virtual environment. Initially we had to develop an open source simulation script in Matlab for determining the global position, angular dynamics, detumbling effects etc which is catered to meet our requirements and specifications. The script was created based on “Space flight Mechanics” written by Dr. Carlos Jos ´e Montalvo, University of South Alabama as well as his Matlab scripts from his official repository. The entire code is available on github at <https://github.com/NEONGASHMEN/arduinodemo1U>

### 2.0.1 Assumptions and reference values for the CubeSat

Orbital inclination =  $98^\circ$

Mass of CubSat =  $2kg$

$$\text{Mass moment of inertia, } M_{3 \times 3} = \begin{bmatrix} 0.006 & 0 & 0 \\ 0 & 0.006 & 0 \\ 0 & 0 & 0.006 \end{bmatrix}$$

Maximum magnetic moment from Magnetorquer =  $0.2Am^2$

Resolution of the Magnetorquer = 256 (8bit)

The magnetic field model chosen was IGRF. General gravitational model is formulated where the gravitational perturbations from celestial bodies other than earth is neglected. Atmospheric drag as well as solar pressure is neglected.

### 2.0.2 The control algorithm for the simulation

The first step in our workflow was to define an arbitrary earth centered, non - rotating reference frame. CubeSat parameters shall be defined with respect to this reference frame. On this reference frame the Z-axis points towards the north pole and the X-axis points towards the equator through the Prime Meridian. The conversion between the aforementioned reference frame to the real world latitude and longitudes the equations given in 5.1 and 5.2 can be used.

The next step is to derive initial conditions for the CubeSat, referenced as initial state. Initial state is an array constituting of 13 elements.

$$S_i = [ X_i \ Y_i \ Z_i \ V_{xi} \ V_{yi} \ V_{zi} \ q_{0i} \ q_{1i} \ q_{2i} \ q_{3i} \ w_{xi} \ w_{yi} \ w_{zi} ]$$

$X_i, Y_i, Z_i$  are position coordinates given by (500,0,0) for 500km altitude at prime meridian

$V_{xi}, V_{yi}, V_{zi}$  are velocity components with respect to body frame given by 5.3  
 $q_{0i}, q_{1i}, q_{2i}, q_{3i}$  are initial quaternions of CubeSat with respect to its body frame, initially taken as zero.

$w_{xi}, w_{yi}, w_{zi}$  are initial angular velocities of CubeSat, initially taken as 10deg /s.

Runge Kutta 4 method of iteration 5.5 is used to find out consecutive states during the satellite's orbit. A partial differential equation which yields the state parameters are fed to the RK-4 algorithm and iterated for consecutive timesteps to obtain the state parameters at consecutive timesteps. The PDE is of the form;

$$\frac{d[S]}{dt} = f([S], t) \quad \& \quad [S]_{t=0} = [S]_0$$

and  $f()$  is given by,

$$\left( \frac{dX}{dt}, \frac{dY}{dt}, \frac{dZ}{dt} \right)_{t_i} = V_{t_{i-1}}$$

$$\left( \frac{d^2X}{dt^2}, \frac{d^2Y}{dt^2}, \frac{d^2Z}{dt^2} \right)_{t_i} = - \left( \frac{GM}{r^2} \right) \hat{r}$$

$$\left( \frac{dQ_0}{dt}, \frac{dQ_1}{dt}, \frac{dQ_2}{dt}, \frac{dQ_3}{dt} \right)_{t_i} = \omega_{t_{i-1}}$$

(Obtained from Euler angles to quaternions conversion 5.6)

$$\left( \frac{d^2\omega}{dt^2} \right) = I^{-1}(T_m)$$

(From rotational inertial equation 5.4)

Here the torque from magnetorquers  $T_m$  is computed from B-dot control algorithm. The algorithm takes in the current angular velocity of the CubeSat and the magnetic field around the satellite to determine the required magnetic moment that is to be produced by the Magnetorquer, in order to achieve efficient detumbling. The B-dot algorithm states that the magnetic moment that is to be produced should be mutually perpendicular to the angular velocity vector ( $\vec{\omega}$ ) and the Earth's magnetic field ( $\vec{B}$ ). To emulate real world noisy sensors, a random noise generation is also hard coded in the script. This gives magnetic field (magnetometer sensor) with an error of few microteslas as well as angular velocity (gyro-error) with an error of few milli radians. This gave us the opportunity to implement a filtering algorithm which minimises the error. A modified Kalman filter is employed 5.7 - combining the current sensor data and previous sensor value, thereby smoothening any steep variations in the measured value. This corrected data is fed to the B-dot algorithm.

$$\vec{\mu} = k. \left( \vec{B} \times \vec{\omega} \right)$$

Here the control variable  $k$  can be adjusted as per the mission's requirements to fine tune the rate of detumbling. In our case  $k$  is adjusted to generate magnetic moments between  $-0.2$  to  $0.2Am^2$ . In the script, the IGRF model is used to obtain the magnetic field at current location. The torque produced, in the magnetometer is given by,

$$\vec{T}_m = \vec{\mu} \times \vec{B}$$

Each consecutive states are stored in each iteration of timestep. For post processing, the numerical values are exported as excel and for visualisation, plotted in a graph.

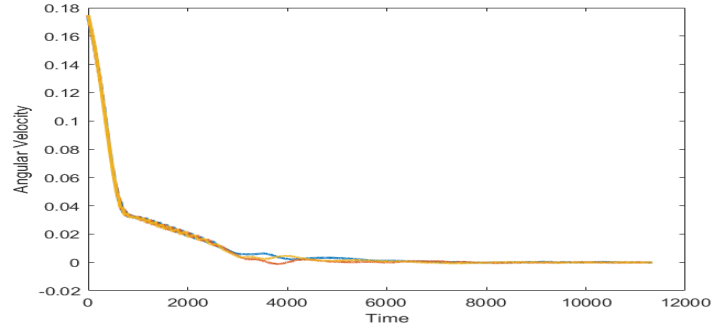
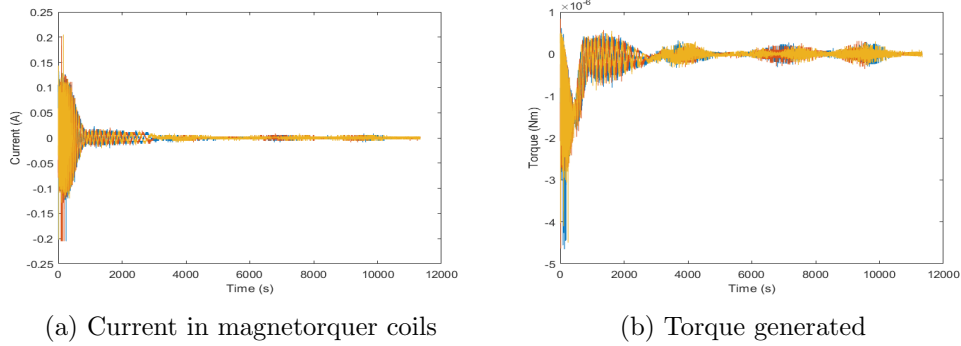


Figure 2.1: Decrease in Cubesat's angular velocity



From these results, the duration for detumbling from  $\vec{\omega} = 10\hat{i} + 10\hat{j} + 10\hat{k}$  (in degrees) to  $\vec{\omega} = 0.01\hat{i} + 0.013\hat{j} + 0.01\hat{k}$  was found to be 106.7667 mins (1.132 orbits) 4.2. Plot between Angular velocity of the CubeSat, Current in each magnetorquers and Torque produced from each torquers V/S time in seconds is given above 2.2a2.2b.

## 3. Designing and testing the magnetorquer using iron core

### 3.0.1 Aim

To pr

In this design we used iron as the core with diameter 5.5mm and number of turns 1088 and gauge 28SWG.

We tested it theoretically using the code written by ourself and practically using the magnetometer. Both the tests did not yield the desired result. So we decided to use Stainless steel FR430 instead of iron core and change the number of turns to 350.

### 3.0.2 Procedure

### 3.0.3 Inference

### 3.0.4 Result

# 4. Designing and testing of Helmholtz Coil

We constructed a Helmholtz coil in order to test the detumbling system developed for our cubesat. Using Helmholtz coil a constant magnetic field is generated. Here we produced magnetic field greater than earth magnetic field to achieve detumbling at faster rate.

## 4.0.1 Objective

The magnetotorquer in the actual cubesat produces the opposite torque depending on the earth's magnetic field. Since earth's magnetic field is in the range of microtesla, it takes hours to stabilize the cubesat. But in the prototype of the cubesat we should demonstrate the detumbling process in a much less time. In order to achieve this task we provide a bigger magnetic moment value in milliteslas so the detumbling process achieves in a greater speed. The optimum value of magnetic moment was fixed as 3 milliteslas keeping residual angular velocity low and achieving detumbling within few minutes. We have to design a Helmholtz coil to test detumbling system in these conditions.

## 4.0.2 Theoretical Calculations

The formula used to calculate the theoretical value of magnetic field at the centre of two coils is given below

$$B = \frac{8}{5\sqrt{5}} \frac{\mu_0 N I}{R}$$

(Derived equation of magnetic field at centre of Helmholtz 5.8)

Where n=number of turns I= current(measured using multi meter) R=radius of the Helmholtz coil.

The magnetic field produced by the Helmholtz coil depends upon number of turns, current through the coil and the resistance per unit length of the coil. The resistance per unit length in turn depends upon the gauge of the wire used. So we have to fix the gauge of wire, value of current and number of turns in a perfect combination so as to achieve desired power draw. If we decrease the gauge of wire, thickness increases and we get less power draw but when thickness increases winding coil becomes difficult. We also limited the number of turns so that we are able to hand wind the coil. For this we wrote a program in Matlab to fix the required gauge satisfying the conditions by

plotting number of turns versus current for each gauge of wire and found the desired value of number of turns as 150, current as 4.6, and selected 14 SWG copper wire . Thus the desired 3mT is acquired through theoretical calculations.

Matlabcode used for plots: github link- [https://github.com/NEONGASHMEN/arduino10/blob/main/helmholtz/hhz\\_vals.m](https://github.com/NEONGASHMEN/arduino10/blob/main/helmholtz/hhz_vals.m)

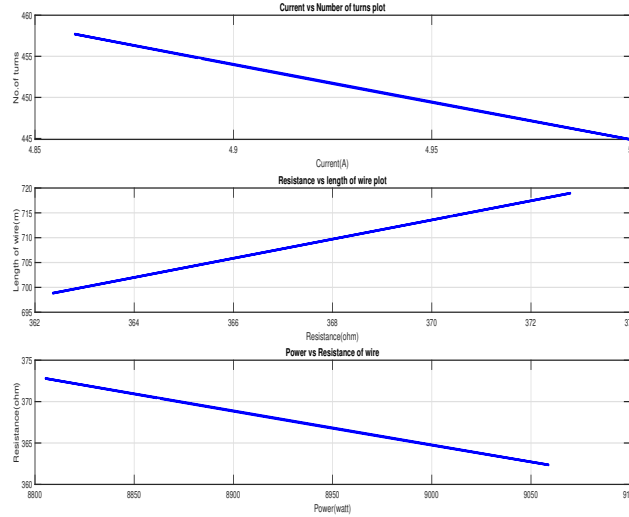


Figure 4.1: Graphs used for fixing wire gauge

#### 4.0.3 Miniature model of Helmholtz coil

To check our theoretical assumptions before the fabrication of actual helmholtz coil we made a miniature model by 3D printing it with ABS(Acrylonitrile Butadiene Styrene) with diameter 10cm and wound with 28 AWG copper wire.

We tested it and proved that the theoretical value of the magnetic field at the centre of the two coils is equal to the practical value of magnetic field measured using EMF detector.

#### 4.0.4 Fabrication of Helmholtz coil

We wound the wire around the plywood frame ,150 turns per coil, which was calculated from the formula.The Helmholtz coil is designed to produce 3mT magnetic field and current. Plywood was selected to build the framework of helmholtz coil. Plywood was choosed because of its relative permeability, which has the value close to that of air and also because it is easily available and is cheap. A hacksaw was used to to cut the plywood in the desired dimensions of the frame. In helmholtz coil distance between the centre of two coils should be equal to the radius of coil .A distance of 20cm was



fixed taking into consideration of the desired magnetic field produced ,since magnetic field is inversely proportional to distance and also considering the fact that the cubesat is 10 cm in width. Then we wound both the coils manually and a SMPS was used to control the voltage applied to the coil. Here we used a 12V SMPS according to our current requirement. We used a multimeter to measure the current flowing through the coil and measured the magnetic field using a mobile application. The magnetic field was found as 2.1mT. Thus we encountered an error of 8.6 percentage but it is negligible. The power draw was found out to be less than 60W.

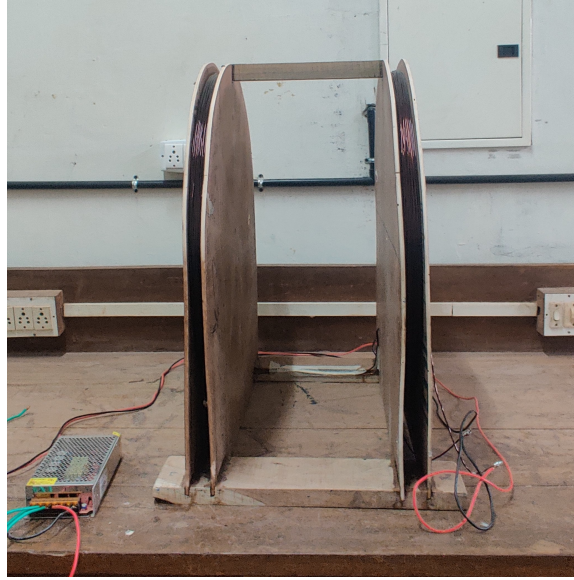


Figure 4.2: Helmholtz coil

#### 4.0.5 Result

Desired uniform magnetic field is produced using the Helmholtz coil.

## 5. Equations

$$\left. \begin{aligned} \rho &= \sqrt{x^2 + y^2 + z^2} \\ \theta_E &= \cos^{-1} \left( \frac{z}{\rho} \right) \\ \psi_E &= \tan^{-1} \left( \frac{y}{x} \right) \end{aligned} \right\} \text{Cartesian to spherical coordinate conversion} \quad (5.1)$$

$$\left. \begin{aligned} \lambda_{LAT} &= 90 - \theta_E \frac{180}{\pi} \\ \lambda_{LON} &= \psi_E \frac{180}{\pi} \\ h &= \rho - R_E \end{aligned} \right\} \text{Latitude, Longitude and height} \quad (5.2)$$

$$v = \sqrt{\mu \left( \frac{2}{r} - \frac{1}{a} \right)} \quad \text{Vis-viva equation, NB: for the simulation } a \approx r \quad (5.3)$$

$$\dot{\vec{\omega}} = I^{-1} \left( T_{Propulsion} + T_{Magnetorquer} + T_{Reactionwheel} - \dot{\vec{\omega}} \times \vec{H} - \dot{I} \vec{\omega} \right) \quad (5.4)$$

$$\left. \begin{aligned} \text{If, } \frac{dy}{dx} &= f(t_n, y_n) \text{ and } y(t_n) = y_n \\ k_1 &= f(t_n, y_n) \\ k_2 &= f\left(t_n + \frac{h}{2}, y_n + h \frac{k_1}{2}\right) \\ k_3 &= f\left(t_n + \frac{h}{2}, y_n + h \frac{k_2}{2}\right) \\ k_4 &= f(t_n + h, y_n + h k_3) \\ k &= \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) \\ y(t_n + h) &= y_n + h k \end{aligned} \right\} \text{RK4} \quad (5.5)$$

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \left[ \begin{aligned} &\cos\left(\frac{\phi}{2}\right) \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right) \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) \\ &\sin\left(\frac{\phi}{2}\right) \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) - \cos\left(\frac{\phi}{2}\right) \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) \\ &\cos\left(\frac{\phi}{2}\right) \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right) \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) \\ &\cos\left(\frac{\phi}{2}\right) \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) - \sin\left(\frac{\phi}{2}\right) \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) \end{aligned} \right] \quad \left. \vphantom{\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}} \right\} \text{Euler to Quarternion conversion} \quad (5.6)$$

$$X_{filtered} = S \cdot X_{current} + (1 - S) X_{previous} \quad , 0 < S < 1, S \text{ being the sensor reliabilty} \quad (5.7)$$

$$\left. \begin{aligned}
\left(\frac{R}{2}\right) &= 2B_1 \left(\frac{R}{2}\right) \\
&= \frac{2\mu_0 n I R^2}{2\left(R^2 + \left(\frac{R}{2}\right)^2\right)^{\frac{3}{2}}} \\
&= \frac{\mu_0 n I R^2}{\left(R^2 + \left(\frac{R}{2}\right)^2\right)^{\frac{3}{2}}} \\
&= \frac{\mu_0 n I R^2}{\left(R^2 + \frac{1}{4}R^2\right)^{\frac{3}{2}}} = \frac{\mu_0 n I R^2}{\left(\frac{5}{4}R^2\right)^{\frac{3}{2}}} \\
&= \left(\frac{4}{5}\right)^{\frac{3}{2}} \frac{\mu_0 N I}{R} \\
&= \frac{8}{5\sqrt{5}} \frac{\mu_0 N I}{R}
\end{aligned} \right\} \text{magnetic field at centre of Helmholtz} \quad (5.8)$$