Development of a single axis 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 +/- 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.

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°
Mass of CubSat = $2kg$
Mass moment of inertia, $M_{3\times3} = \begin{bmatrix} 0.006 & 0 & 0 \\ 0 & 0.006 & 0 \\ 0 & 0 & 0.006 \end{bmatrix}$

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 arbitary 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 6.1 and 6.2 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.

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

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 6.3

$q_{0i}, q_{1i}, q_{2i}, q_{3i}$ are initial quarternions 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 $10\deg/s$.

Runge Kutta 4 method of iteration 6.5 is used to find out consecutive states during the satellte's orbit. A partial differential eqution 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 quarternions conversion 6.6)

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

(From rotational inertial equation 6.4)

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, inorder to achieve efficient detumbling. The B-dot algorithm states that the magnetic moment that is to be produced should be mututally perpendcular to the angular velocity vector ($\vec{\omega}$) and the Earth's magnetic field ($\vec{B}$). To emulate real world noissy 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 6.7 - 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.

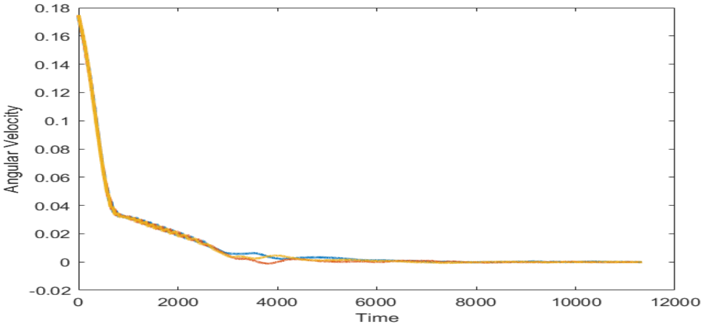
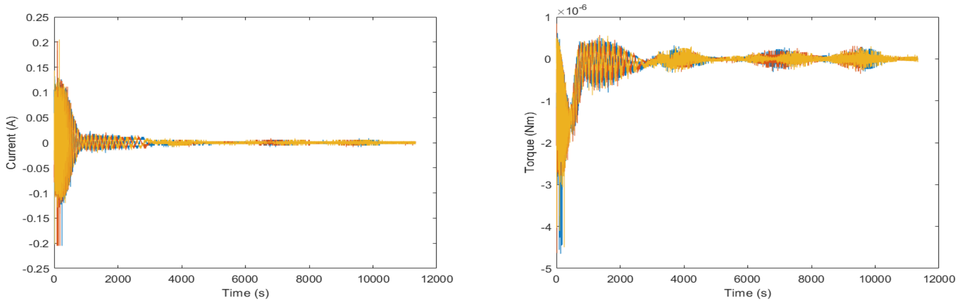


Figure 2.1: Decrease in Cubesat's angular velocity



(a) Current in magnetorquer coils

(b) Torque generated

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) 4.3. 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

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.

3.0.1 Procedure

3.0.2 Inference

3.0.3 Result

4. Designing and testing of Helmholtz Coil

We constructed a Helmholtz coil inorder to test the detumbling system developed for our cubesat. Using Helmholtz coil a constant and uniform 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 microteslas and hence it takes hours to stabilize the cubesat. But in the prototype of the cubesat we should demonstrate the detumbling process in a much lesser time. Inorder to achieve this task we provide a bigger magnetic field value in the order of milliteslas so the detumbling process occurs at a greater speed. The optimum value of magnetic moment was fixed as 1 millitesla keeping residual angular velocity low and achieving detumbling within few minutes. This is obtained by modifying the aforementioned CubeSat simulation script. We have to design a Helmholtz coil which generates the required magnetic field.

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 **??**)

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

For accomodating the CubeSat along with the airbearing inside the Helmholtz cage, we've alotted a gap of 26cm between the Coils. Therefore the radius of the Helmoltz coils should also be made to be equal to 26cm for maintaining uniform magnetic field at the midpoint of the two helmholtz cols. The cage needs to be fabricated as per the dimensions given below 4.1.

The magnetic field produced by the Helmholtz coil also depends upon number of turns and the current through the coil. The resistance per unit length depends upon

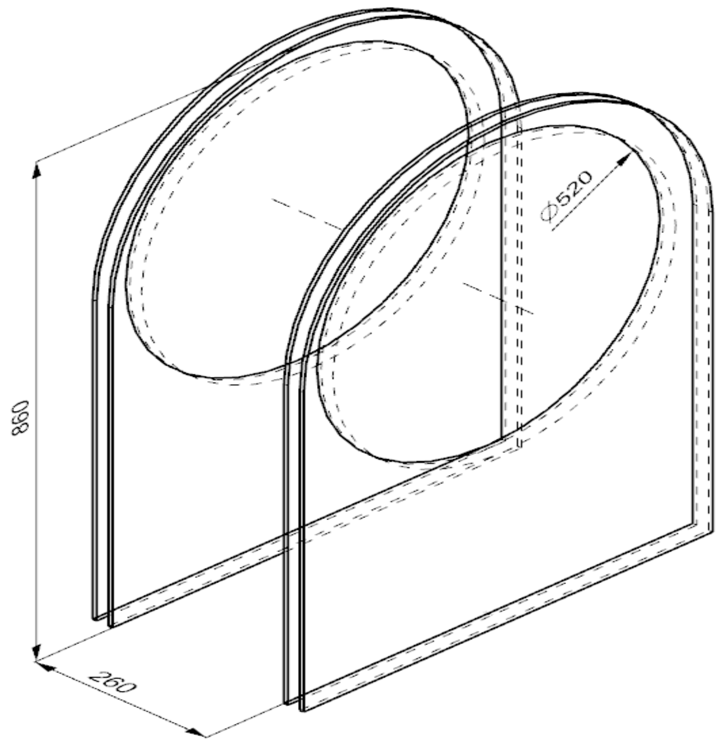


Figure 4.1: Model of the Helmholtz cage

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.

To acheive this we wrote a script in Matlab that iterates between different guages of wire and different number of turns to find the current, power draw etc for each combination.

For 10SWG,
N = 50, I = 5.8A and P = 5.76W (obtained from 6.8)
N = 51, I = 5.7A and P = 5.66W
......
N = 200, I = 1.45A and P = 1.43W

For 11SWG,
N = 50, I = 5.8A and P = 7W
......
N = 200, I = 1.45A and P = 1.75W

For 12SWG,
N = 50, I = 5.8A and P = 8.7W
......
N = 200, I = 1.45A and P = 2.17W

and so on...

We found an optimum combination of 14SWG Copper wire with 150 turns and 1.966A 4.2 that produces the required magnetic field.

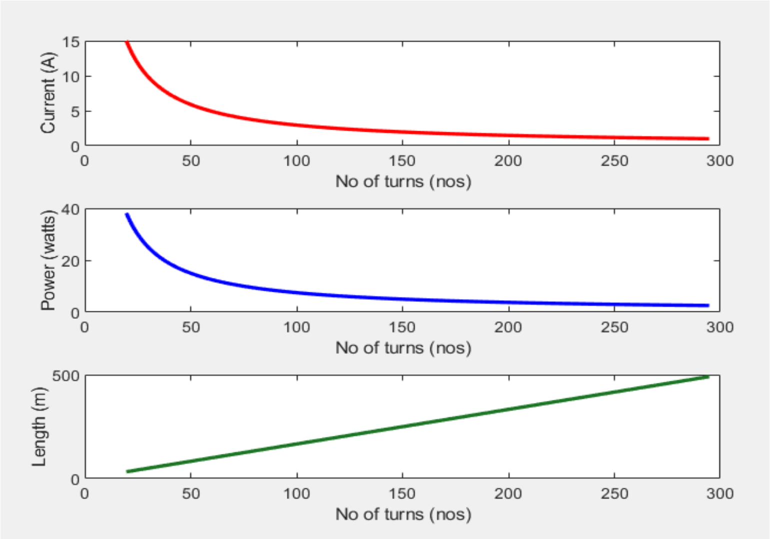


Figure 4.2: Current and Power draw for 14SWG wire

Matlabcode used for plots: github link- `https://github.com/NEONGASHMEN/arduinodemo1U/blob/main/helmholtz/hhz_vals.m`

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 winded 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

A Plywood frame of aforementioned dimensions was fabricated using a woodcutter and adhesives. 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. We wound the wire about 150 turns per coil around the plywood frameby hand. The Helmholtz coil is designed to produce 1mT magnetic field. A SMPS was used to control the voltage applied to the coil. Here we used a 12V SMPS according to our current requirement and for safety reasons.

We used a multimeter to measure the current flowing through the coil and measured the magnetic field using a magnetometer. The current was limited to 1.96A using a variable rheostat and the magnetic field was found to be xmT. The margin of error is x. The power draw was found out to be less than xW.

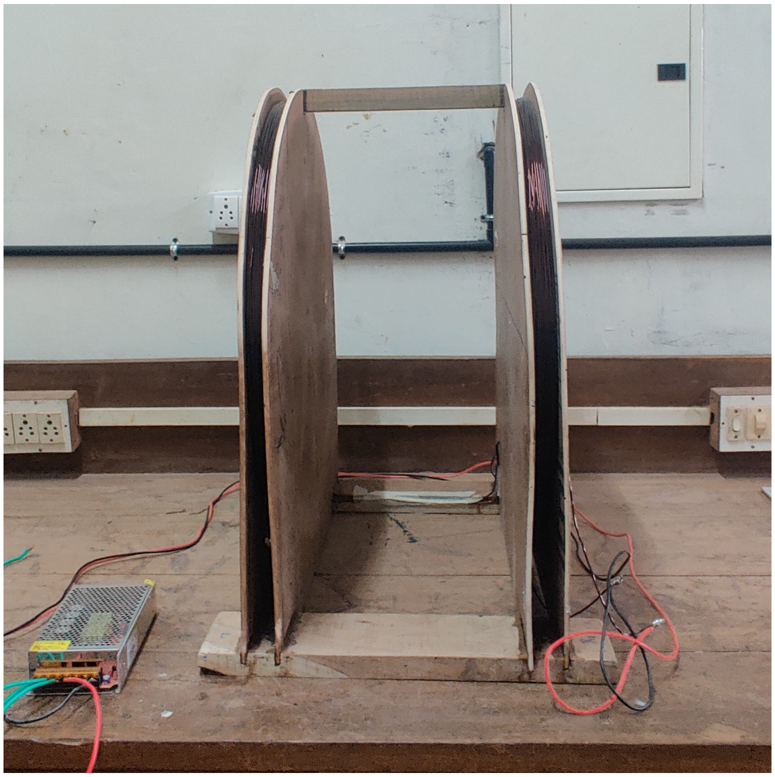


Figure 4.3: Helmholtz coil

5. Design and Development of Airbearing

An airbearing is just a bearing in which moving surfaces are kept apart by a layer of air. Use of air as a medium creates a near friction less coupling between the mating surfaces. As the detumbling takes place at the orbit, we needed to have a frictionless environment to replicate the actual scenario. The objective was to create a frictionless free to rotate platform for the CubeSat to rest upon.

5.0.1 Design and Analysis

The main part of an airbearing is a porous material which is capable of producing uniform layer of air over which the corresponding mating part can be rested. The porous material selected for our purposes is Graphite (1.7g/cc density).

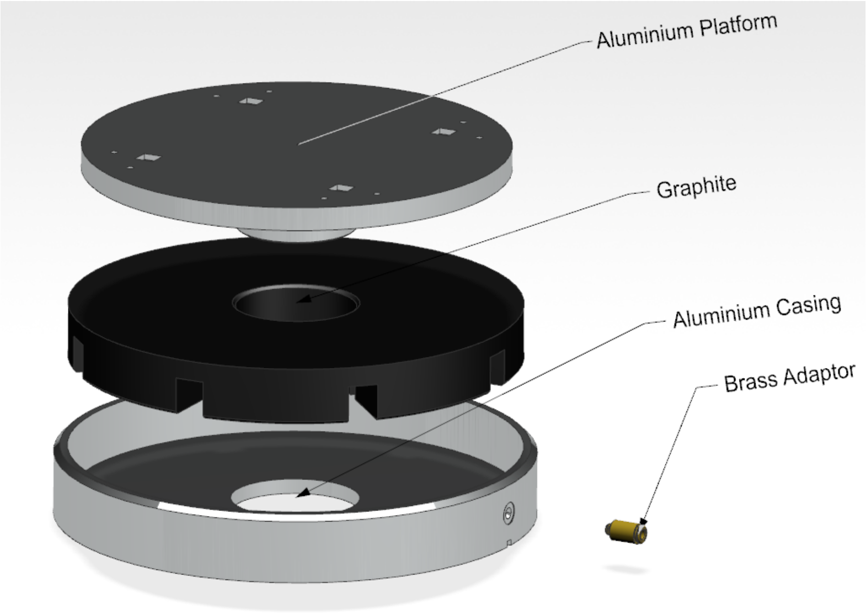


Figure 5.1: Air Bearing

The platform was made keeping in mind the base area of the CubeSat. A generous amount of 180mm diameter platform was designed for this case. The entire bearing was required to be low profile due to space constraints in the testing rig. A thickness of 25mm of graphite was first chosen to minimise the pressure drop through the graphite while maintaining sufficient stiffness. Air flow channels to allow the pumped air to

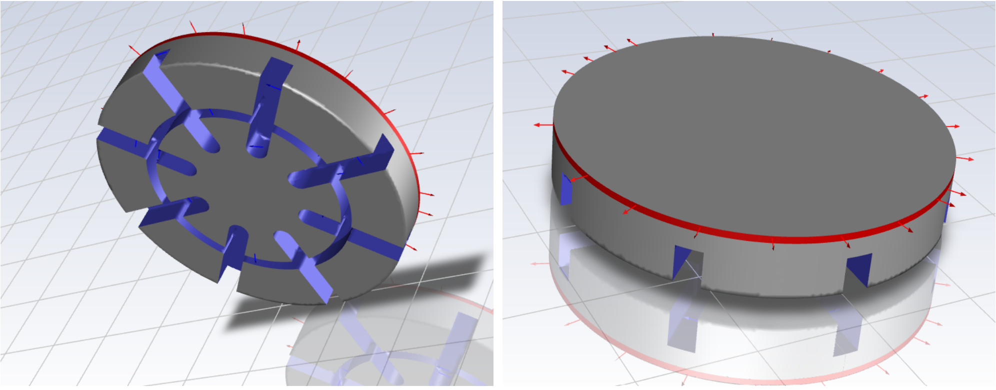
spread uniformly spread accross the entire graphite, is also designed. The performance of the designed graphite section was analysed on Ansys Fluent.
Fluid inlet is assigned as the air passages on the graphite. The air compressor that would be used alongside the air bearing was rated 24L capacity. Hence the flow rate from the compressor at 60psi was calculated as follows,

Cubic feet per metre, $CFM = \frac{V\left(P_f - P_i\right)}{14.7 t_{fill}}$

where, t_{fill} is the fill time to reach 60psi pressure and V is the tank capacity in cubic feet.

Hence the volume flow rate at compressor outlet was calculated to be $0.001088 m^3/s$ From the flow rate the velocity of flow was found and is set as the velocity inlet boundary condition (shown in blue color in 5.2a). The outlet is set to be ambient pressure condition (shown in red color in 5.2b). The porous zone of the graphite is set with the following parameters,

Flow permeability, $k = 1.85 \times 10^{-15}$ [6.9]
Viscous resistance, $R = 5.405 \times 10^{14}$
Mean diameter of graphite particle, $D_p = 3.16069 \times 10^{-6}$
Porosity of graphite, $\epsilon = 0.25$
Inertial loss coefficient, $C = 5.314 \times 10^{7}$ [6.10]



(a) Velocity inlet

(b) Pressure outlet

A density based solver is employed and satidfactory convergence was obtained after about 1000 iterations. The static pressure contour plot of the airflow is shown below.

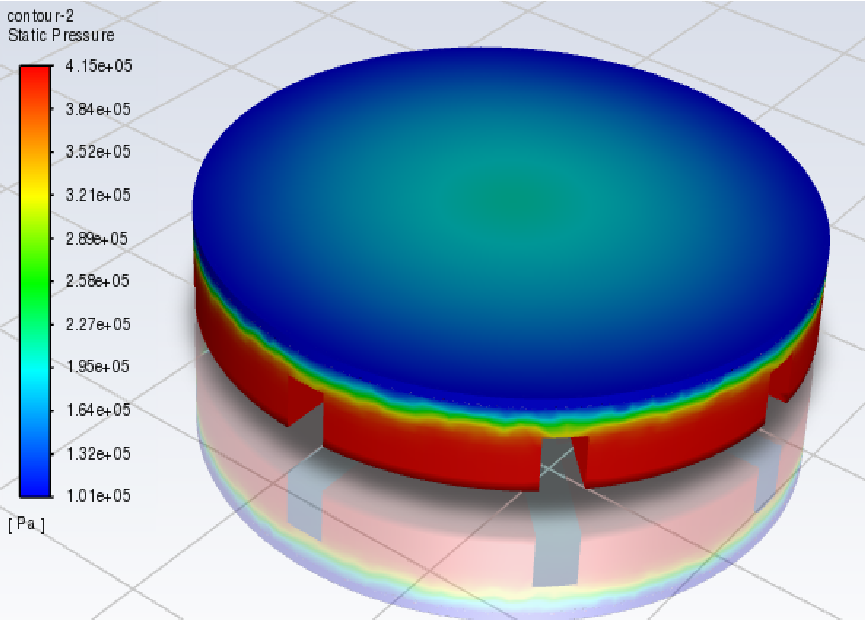


Figure 5.3: Pressure contour over the graphite layer

The results shows about 137684 Pa of static pressure on the bearing platform, which translates to 3924 N force. That is about 400 kg lift capacity. This is more than enough for our required application.

5.0.2 Fabrication

Casing and Platform

These were fabricated mainly by turning in lathe at 300RPM. A disk shaped Aluminium material of 210x50mm dimensions were turned using orthogonal cutting to yield the platform. A combination of orthogonal cutting, drilling and boring was done on a second piece of the same dimension to obtain the casing. The air inlet hole was machined using a vertical milling machine with a drill size of 8mm. The hole was later tapped with a thread of 1/8BSP, to fasten the tubing adaptor. The air vent groove was machined on the uderside of the casing using a 3mm end mill cutter.

Graphite

The air flow path on the graphite was machined using a CNC and later sanded manually to obtain better finishing.

The graphite was joined together with the casing using an airtight epoxy sealant. The platform is placed over the graphite to yield the finished bearing

6. Equations

$$\left.\begin{array}{l}\rho = \sqrt{x^2+y^2+z^2} \\ \theta_E = \cos^{-1}\left(\frac{z}{\rho}\right) \\ \psi_E = \tan^{-1}\left(\frac{y}{x}\right)\end{array}\right\}\text{Cartesian to spherical coordinate conversion} \tag{6.1}$$

$$\left.\begin{array}{l}\lambda_{LAT} = 90 - \theta_E\frac{180}{\pi} \\ \lambda_{LON} = \psi_E\frac{180}{\pi} \\ h = \rho - R_E\end{array}\right\}\text{Latitude, Longitude and height} \tag{6.2}$$

$$v = \sqrt{\mu\left(\frac{2}{r}-\frac{1}{a}\right)}, \qquad \text{Vis-viva equation, NB: for the simulation } a \approx r \tag{6.3}$$

$$\dot{\vec{\omega}} = I^{-1}\left(T_{Propulsion} + T_{Magnetorquer} + T_{Reactionwheel} - \dot{\vec{\omega}} \times \vec{H} - \dot{I}\vec{\omega}\right) \tag{6.4}$$

$$\left.\begin{array}{l}\text{If, } \frac{dy}{dx} = f\left(t_n, y_n\right) \text{ and } y\left(t_n\right) = y_n \\ k_1 = f\left(t_n, y_n\right) \\ k_2 = f\left(t_n + \frac{h}{2}, y_n + h\frac{k_1}{2}\right) \\ k_3 = f\left(t_n + \frac{h}{2}, y_n + h\frac{k_2}{2}\right) \\ k_4 = f\left(t_n + h, y_n + hk_3\right) \\ k = \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \\ y\left(t_n + h\right) = y_n + hk\end{array}\right\} \quad \text{RK4} \tag{6.5}$$

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \left.\begin{bmatrix} \cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) - \cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) - \sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) \end{bmatrix}\right\}\text{Euler to Quarternion conversion} \tag{6.6}$$

$$X_{filtered} = S.X_{current} + (1-S)\,X_{previous} \quad ,0 < S < 1\text{,S being the sensor reliabilty} \tag{6.7}$$

$$\left.\begin{array}{l} \left(\frac{R}{2}\right) = 2B_1\left(\frac{R}{2}\right) \\ = \frac{2\mu_0 n I R^2}{2\left(R^2+\left(\frac{R}{2}\right)^2\right)^{\frac{3}{2}}} \\ = \frac{\mu_0 n I R^2}{\left(R^2+\left(\frac{R}{2}\right)^2\right)^{\frac{3}{2}}} \\ = \frac{\mu_0 n I R^2}{\left(R^2+\frac{1}{4}R^2\right)^{\frac{3}{2}}} = \frac{\mu_0 n I R^2}{\left(\frac{5}{4}R^2\right)^{\frac{3}{2}}} \\ = \left(\frac{4}{5}\right)^{\frac{3}{2}} \frac{\mu_0 N I}{R} \\ = \frac{8}{5\sqrt{5}} \frac{\mu_0 N I}{R} \end{array}\right\} \text{magnetic field at centre of Helmholtz} \qquad (6.8)$$

$$k = \left(\frac{D_p^2}{150}\right)\left(\frac{\epsilon^3}{(1-\epsilon)^2}\right), \qquad \text{Permeability - Porosity relation} \qquad (6.9)$$

$$C = \left(\frac{3.5}{D_p}\right)\frac{1-\epsilon}{\epsilon^3}, \qquad \text{Medium porosity - Inertial loss coefficient relation} \qquad (6.10)$$