|  |
| --- |
| Esraa Mohamed – Gehad Essam – Islam Sami Mohamed Hussein – Youhana Mikhaiel Under the supervision of Eng. Ahmed Farrag  11-8-2016 |

|  |
| --- |
| National Authority For Remote Sensing & Space Science (NARSS) |
| Reaction momentum wheel |
| Educational Satellite Program (ESP) |

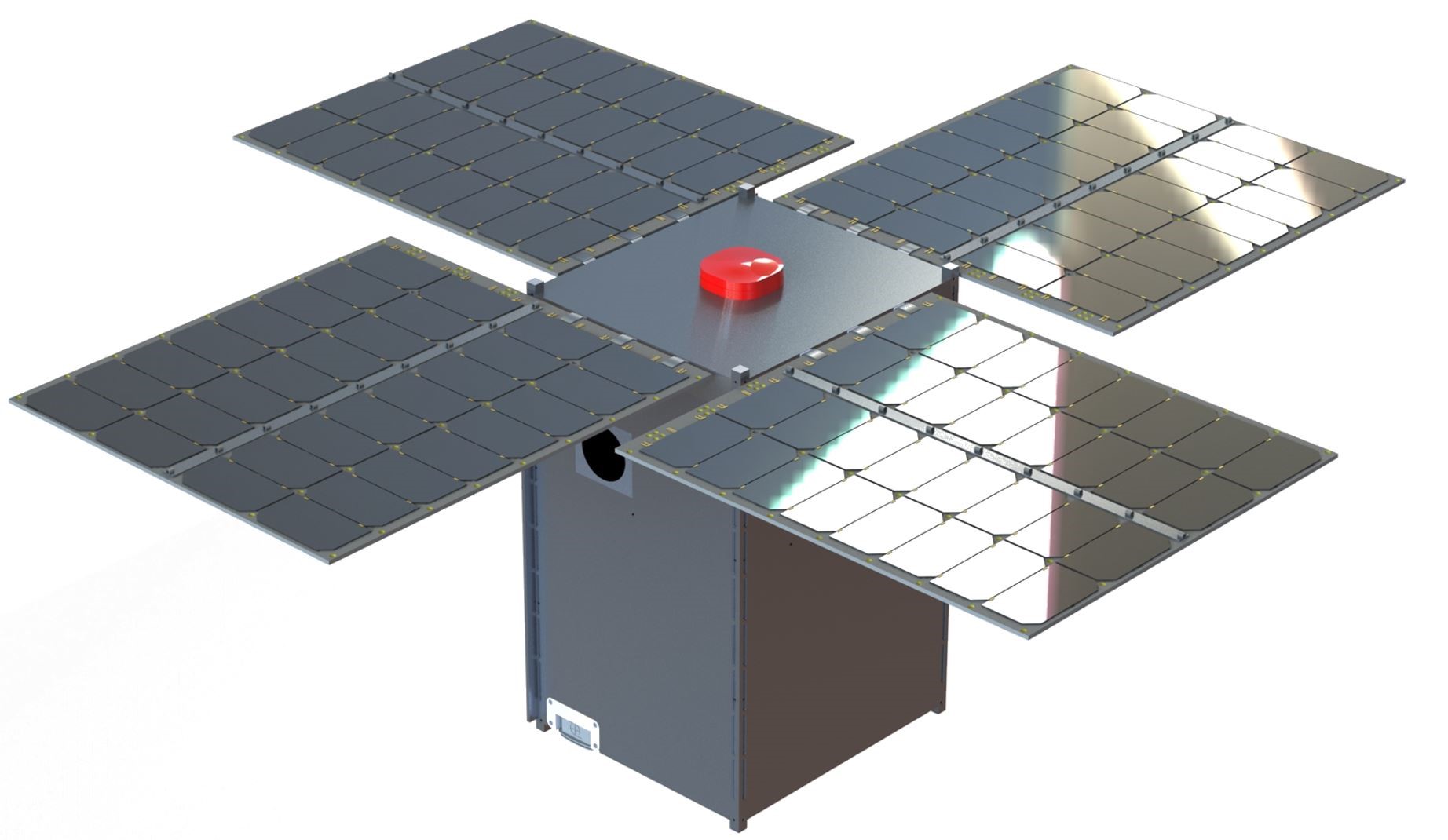


Table of Contents

**Table of figures2**

**Abstract3**

**Introduction4**

**1. Mechanical Design5**

1.1 Satellite body design5

1.2 Reaction wheel design7

**2. Satellite subsystems13**

2.1 On-board computer subsystem13

2.2 Communication subsystem14

2.3 Payload subsystem16

2.4 Attitude determination 16

2.5 Power subsystem17

**3. Control18**

3.1 PID control18

3.2 Signal Constraint21

3.3 Parameter estimation22

**4. Results24**

**5. Acknowledgments25**

**6. References26**

**7. Appendix27**

# Table of Figures

Figure 1-1: Final design for the satellite …………………………………………………………………………………………………5

Figure 1-2: Satellite after integrating its parts with bearing …………………………………………………………..………6

Figure 1-3: Placing components inside the cuboid in SOLIDWORKS …………………………………………………….…7

Figure 1-4: Comparison between hollow and solid cylinders …………………………………………………………………8

Figure 1-5: Shape of reaction wheel ………………………………………………………………………………………………………9

Figure 1-6: Reaction wheel with upward and downward necks ……………………………………………………..………9

Figure 1-7: Steel wheel’s inertia versus inner radius with different outer radii ………………………………………9

Figure 1-8: Aluminum wheel’s inertia versus inner radius with different outer radii ……..………………….…10

Figure 1-9: Cover and stand for the motor …………………………………………………………….…………………………….10

Figure 1-10: The encoder fixed ………………..………………………………………………………………………………………..…11

Figure 1-11: Final appearance for the satellite while testing …………………………………………………………..……12

Figure 2-1: Arduino Mega board ……………………………………………………………………………………………………………13

Figure 2-2: Arduino IDE interface ……………………………………………….…………………………………………………………14

Figure 2-3: Bluetooth module HC-05 …………………………………………………….………………………………………………15

Figure 2-4: Stimulated ground station using labview ……………….……………………………………………………………15

Figure 2-5: TTL serial camera ……..…………………………………………………………………………………………………………16

Figure 2-6: SD card module …………………..………………………………………………………………………………………………16

Figure 2-7: MPU-9150 ……………………..……………………………………………………………………………………………………16

Figure 2-8: Motor Shield ……………………………………………………………………………….………………………………………17

Figure 3-1: PID simulated on Simulink …………………………………………………………………………………..………………19

Figure 3-2: Euler angles simulated on Simulink ……………………………………………………………………..………………19

Figure 3-3: Model of DC motor ……………………………………………………………………………………………..………………20

Figure 3-4: Signal constraint toolbox …………………………………………………………………..………………..………………21

Figure 3-5: Signal constraint toolbox output ………………………………………………………….……………..………………21

# Abstract

Cubesats are small satellites used in researching in space with approximate dimensions around 100x100x120 mm cubic units and a weight ranging from 1-1.5 kgs and commercial-off the shelves electrical and structural components are often used in it. They are often sent to space to be placed on the International Space Station and they are also used in educational purposes by training university students and even high school students how to deal with the satellite subsystems when designing these cube satellites. When these cube satellites are sent to space, their orientation needs to be adjusted from time to time using several techniques.

# Introduction

Cubesats and satellites in general have several subsystems integrated together to have a ready satellite to be ready for launching and to do its own mission required from it. One of the most important subsystems used in these cubesats is the Attitude and Determination Control Subsystem abbreviated as ADCS which is needed to overcome the natural forces found in space such as solar pressure, magnetic torques, gravity gradient and aerodynamic forces. ADCS is used to mainly to adjust the orientation of the satellite in space in three dimensions and this could be used in many applications such as adjusting the orientation of the satellite to capture an image for certain region on earth or to protect some parts which are sensitive to direct sunlight or to set the position of the satellite to have maximum gain of energy from solar cells mounted on the walls of it. There are several ways to control the orientation of the satellite such as magnetic torque rods, control moment gyros, thrusters and momentum reaction wheels. Momentum reaction wheels are the technique used in most of the cube satellites but it is not yet developed nor implemented in picosatellites and this technique is the one that was applied in this project.

Conservation of angular momentum is used in this project to control the attitude of the satellite which means that the magnitude of angular momentum of the reaction wheel is equal to the magnitude of angular momentum of the satellite’s body but in opposite direction given by the following equation.

The main mission of this project is to send a command to the satellite via Bluetooth to go to a certain angle so the wheel accelerates and a torque is generated to rotate the satellite and then when reaching the desired angle, the angular velocity gets constant which means that the acceleration is equal to zero and the position of the satellite is fixed at this desired angle after that an automatic command is sent to the camera to capture an image, saving it in the SD card and then sending it from the SD card to the ground station developed using LABVIEW.

# Mechanical design

# 1.1 Satellite body design

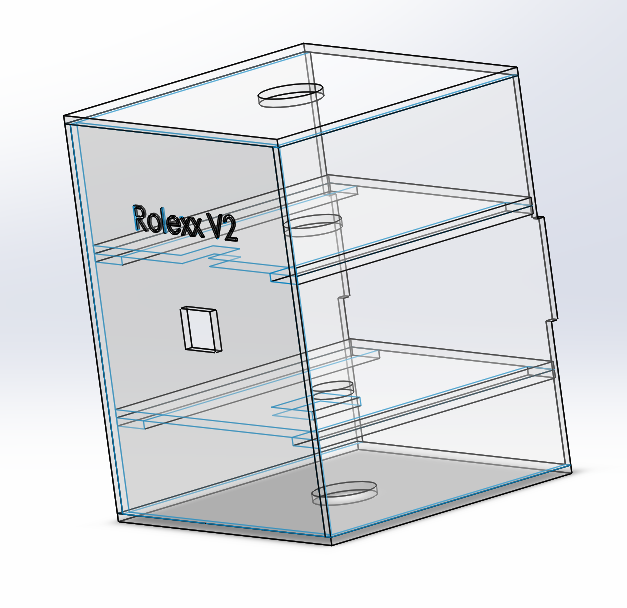
The structure design was planned to be mainly dependent on a small size and a weight not exceeding 1.5 kg with a transparent material to be able to see what’s inside the satellite while the mission is tested. The structure was chosen to be made of acrylic with density 1200 and a 1000x500x4mm sheet was used. Going to the design procedure, the satellite size was decided to be 140x140x180mm with a cuboid shape positioned vertically to maximize the inertia that could be reached from this volume and the stability of the system around the vertical axis which we wish to rotate our satellite about. Several body structures were designed on SOLIDWORKS 2015 to compare between them and choose the best for our mission with the suitable constraints we have and then the best design was picked ***(Fig. 1-1)*** and it was manufactured using a laser cutter as to guarantee high accuracy for the measurements.

Figure -1: Final design for the satellite

The manufacturing process was not that hard but it was challenging as to have a regular cuboid with horizontal inner stages for the satellite components as to maintain the best output from the system and guaranteeing good accuracy for the MPU-9150 readings and the torque which is generated from accelerating the reaction wheel. After cutting the parts, they were integrated together using super glue and small pieces of cork to insure that they are attached together tightly to have the stability required. Now, the satellite body is ready ***(Fig. 1-2)****.*

The satellite body needed to be hanged as to simulate what’s really happening in the space where there is no gravity, so a wooden frame was manufactured to hold our cuboid. Then, back to the satellite a ball bearing was placed on the top face of



Figure 1-: Satellite after integrating its parts with bearing

the satellite. The cuboid satellite was hanged in the wooden frame using a 12mm hex bolt screw which is 100mm long and by using two nuts with the same inner diameter of the screw attached tightly to the rotating part of the bearing so we can have a rotating cuboid but we must consider the friction generated from the bearing to be able to set the specifications of the reaction wheel.

The cuboid was designed so that the two horizontal stages could be placed out of it so that connections could easily be modified without any problem or waste of time or effort. So that it has three stages, one for the motor mounted by the wheel, another for the battery and one for the rest of the components. Holes were made in each stage to pass the wire connections smoothly among different stages and also a hole on the side face for the camera.

In this project, conservation of momentum is used to rotate the satellite in the opposite direction in which the wheel is rotating so the inertia of the cube is needed to be calculated so that the components where placed initially on SOLIDWORKS 2015 ***(Fig. 1-3)*** to estimate the approximate inertia of the satellite thus estimating the required inertia for the wheel after choosing the motor knowing its angular velocity in rpm.

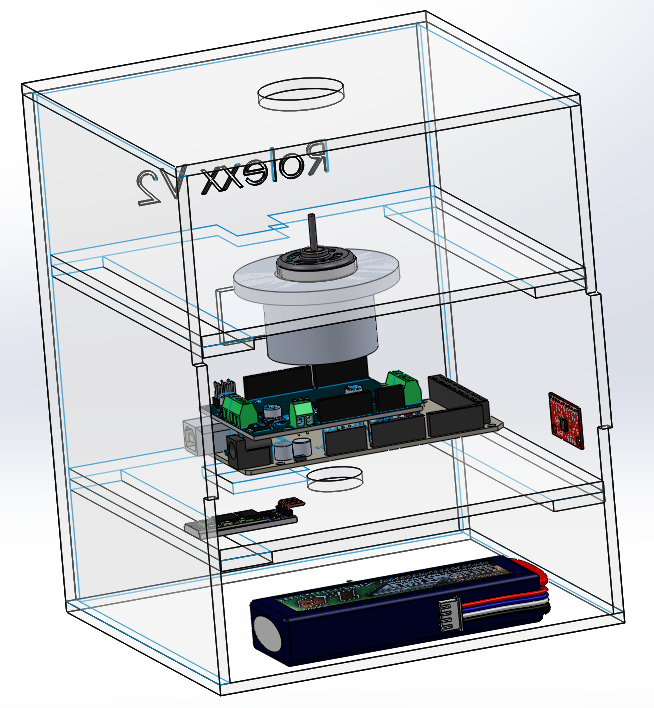


Figure 1-: Placing components inside the cuboid in SOLIDWORKS

# 1.2 Reaction wheel design

Moving forward in this project, it’s time to begin designing the reaction wheel with specific inertia to do the task required from it in a good way. Now, the approximate value of the inertia of the satellite is known and the rpm of the motor is known without the gear box while we are applying no load on the motor’s shaft and the mission is to be able to rotate the satellite with an angle of in nearly 10 seconds so this specification is translated into rpm to be:

so knowing the minimum inertia needed for the reaction wheel could be known easily now by applying conservation of angular momentum where

To start the design phase of the wheel, there are some parameters which cannot be changed or have a certain limit as to simplify analysis. The diameter of the motor’s shaft is about 1.5mm, the height of the system of the shaft and the wheel cannot exceed 45mm and finally the diameter of the wheel cannot exceed 100mm. The inertia of a hollow cylinder is given by

where is the wheels weight, is the inner diameter of the hollow cylinder and is the outer diameter.

The inertia of a solid cylinder is given by

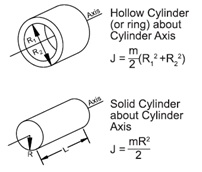
It is obvious that a hollow cylinder with the same outer diameter of a solid cylinder has a larger inertia ***(Fig. 1-4)*** so as to optimize between the value of the inertia and the volume of the wheel and the weight of the system, a disk with 5mm in thickness was made and placing on it a hollow cylinder with the same outer diameter and 5mm in thickness also ***(Fig. 1-5)***.

Figure 1-4: Comparison between hollow and solid cylinders

The next step in designing the wheel was to create a place where the encoder should be positioned as to measure the rpm generated from the motor so a 10mm neck was made in the top face of the wheel so that we code place the plastic wheel which interrupts the encoder. A 10mm neck was made on the bottom face heading downwards as the wheel to be fixed in the shaft of the motor ***(Fig. 6)***.

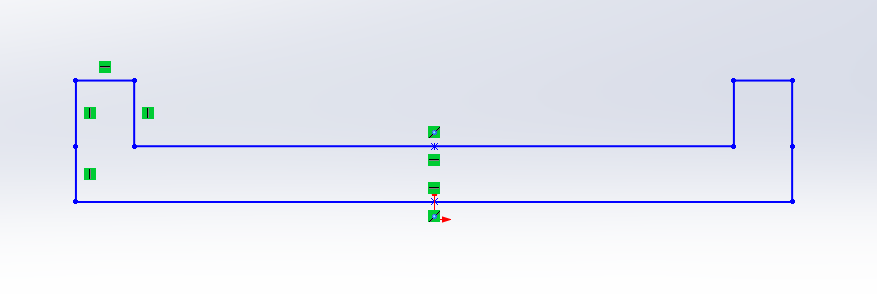
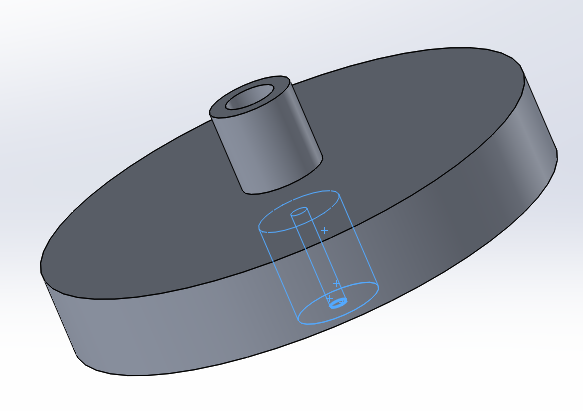


Figure 1-5: Shape of reaction wheel

Figure 1-6: reaction wheel with upward and downward necks

A MATLAB code was developed to draw graphs for different inertias on the y-axis and the inner radius of the hollow cylinder on the x-axis with different outer radii varying from 30mm to 45mm with a step of 5mm to be manufactured easily. Two graphs were made one when manufacturing the wheel from steel ***(Fig. 1-7)*** and the other one when manufacturing it from aluminum ***(Fig. 1-8)***.

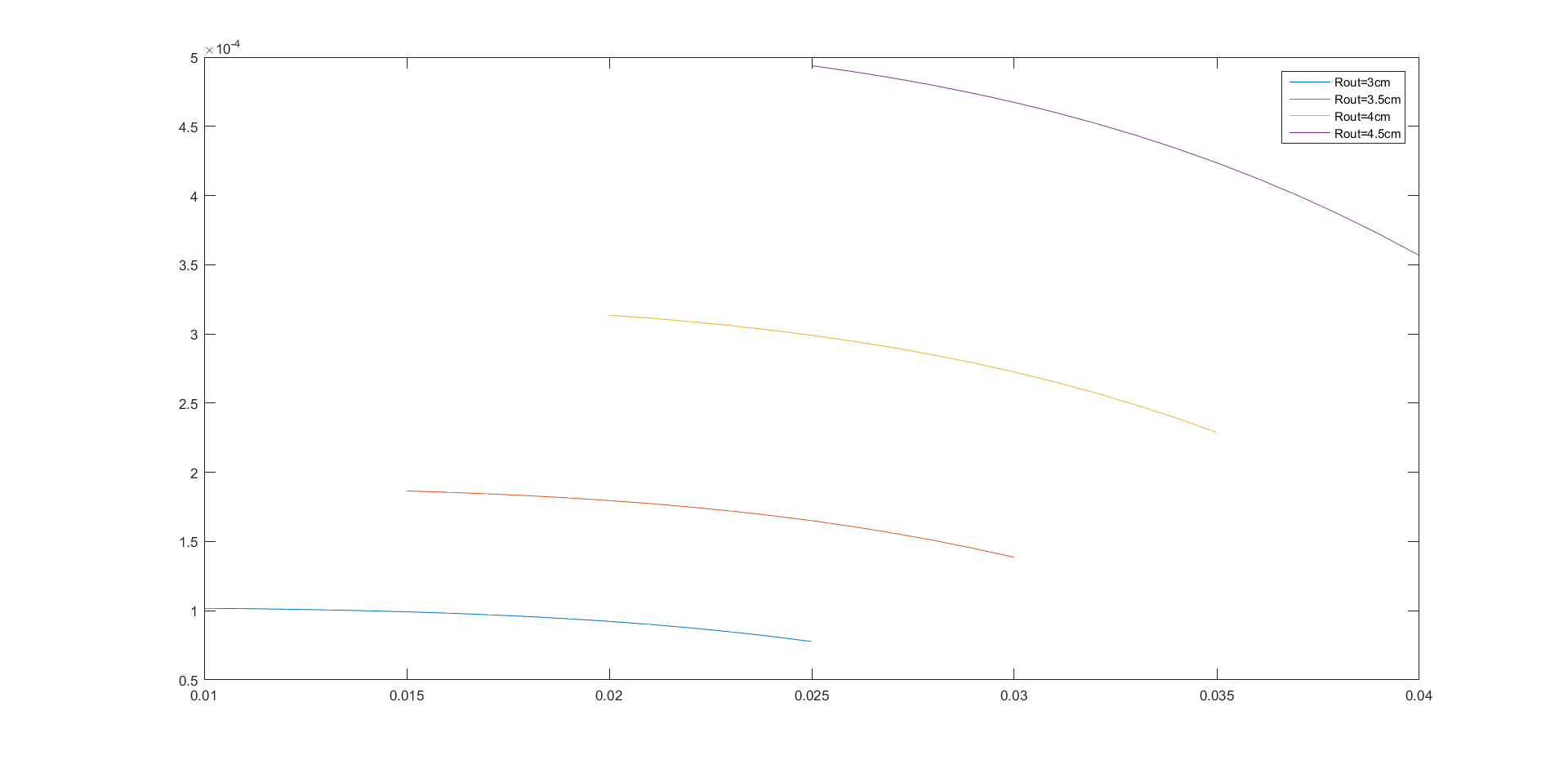


Figure 1-7: Steel wheel's inertia versus inner radius with different outer radii

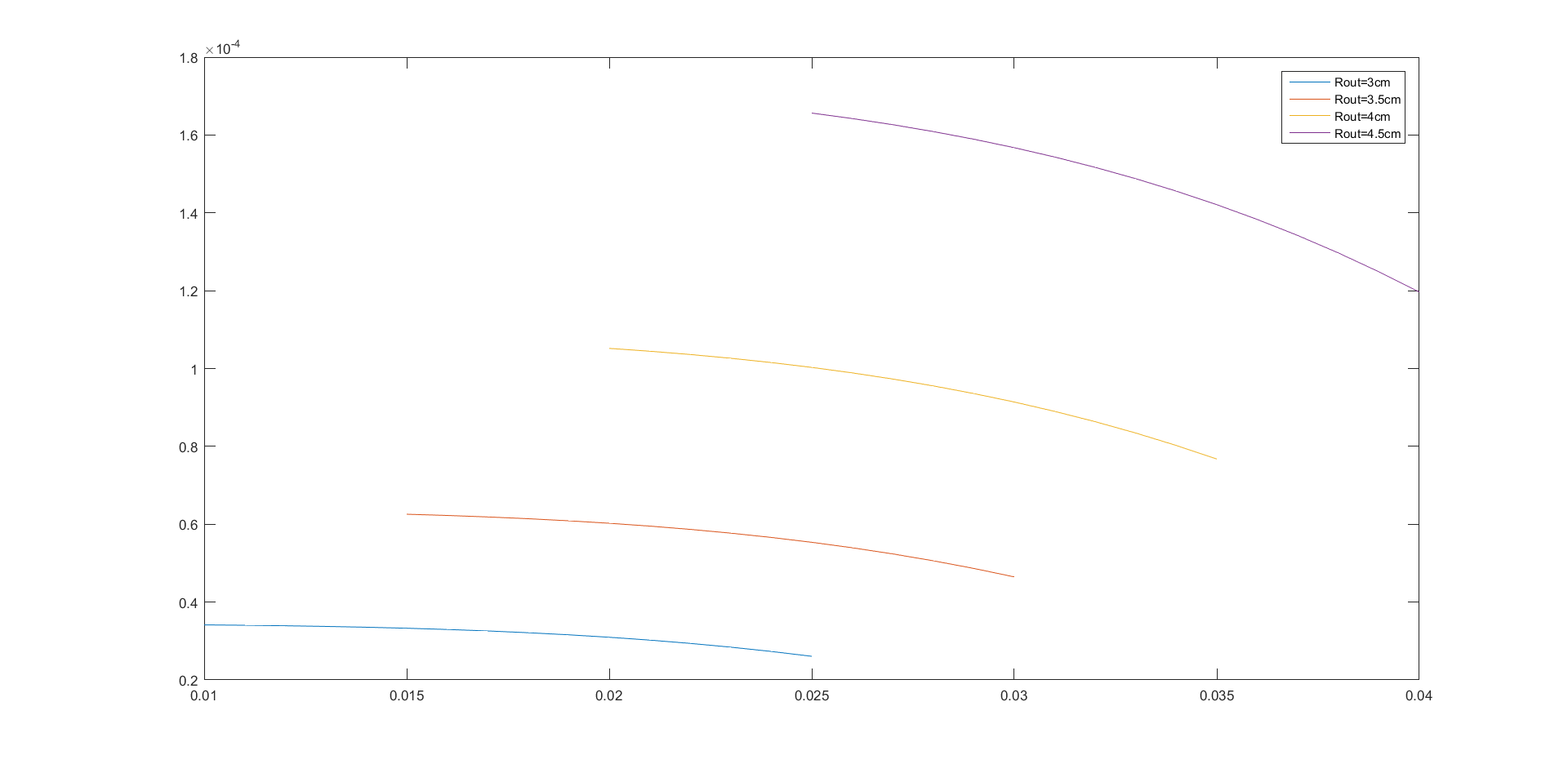


Figure 1-8: Aluminum wheel's inertia versus inner radius with different outer radii

So picking the inertia needed was not that hard then specifying the dimensions corresponding to the desired inertia. The inertia includes the disk as a solid cylinder, the upward neck, the downward neck and the frame of the disk as hollow cylinders. Finally, the wheel is designed on SOLIDWORKS 2015 and was given the same material to make sure that the moment of inertia got from the MATLAB program was the same and that’s what happened.

The next question was how to fix the motor to withstand the vibrations generated from the high rpm. An aluminum body was designed to be as a cover for the motor with inner radius equal to the radius of the motor and then it is fixed in the first stage with nails and nuts ***(Fig. 1-9)***.

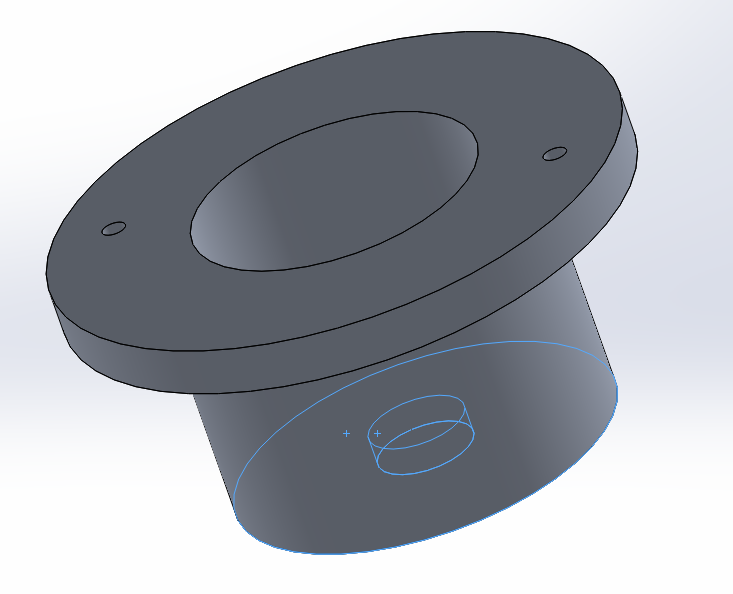


Figure 1-9: Cover and stand for the motor

Everything was ready to go to El-Sabteyya to be manufactured. The blacksmith was shown the design of both the wheel and the motor cover which was not that easy to make but anyways they were made successfully with no problems and the wheel was fixed in the shaft of the motor with two nails of the sides of the wheels downward neck. The upward neck of the wheel was drilled in it to fix the encoder wheel above it ***(Fig. 1-10)***. The hardware could be integrated now to make sure that all these parts are working together harmonically.

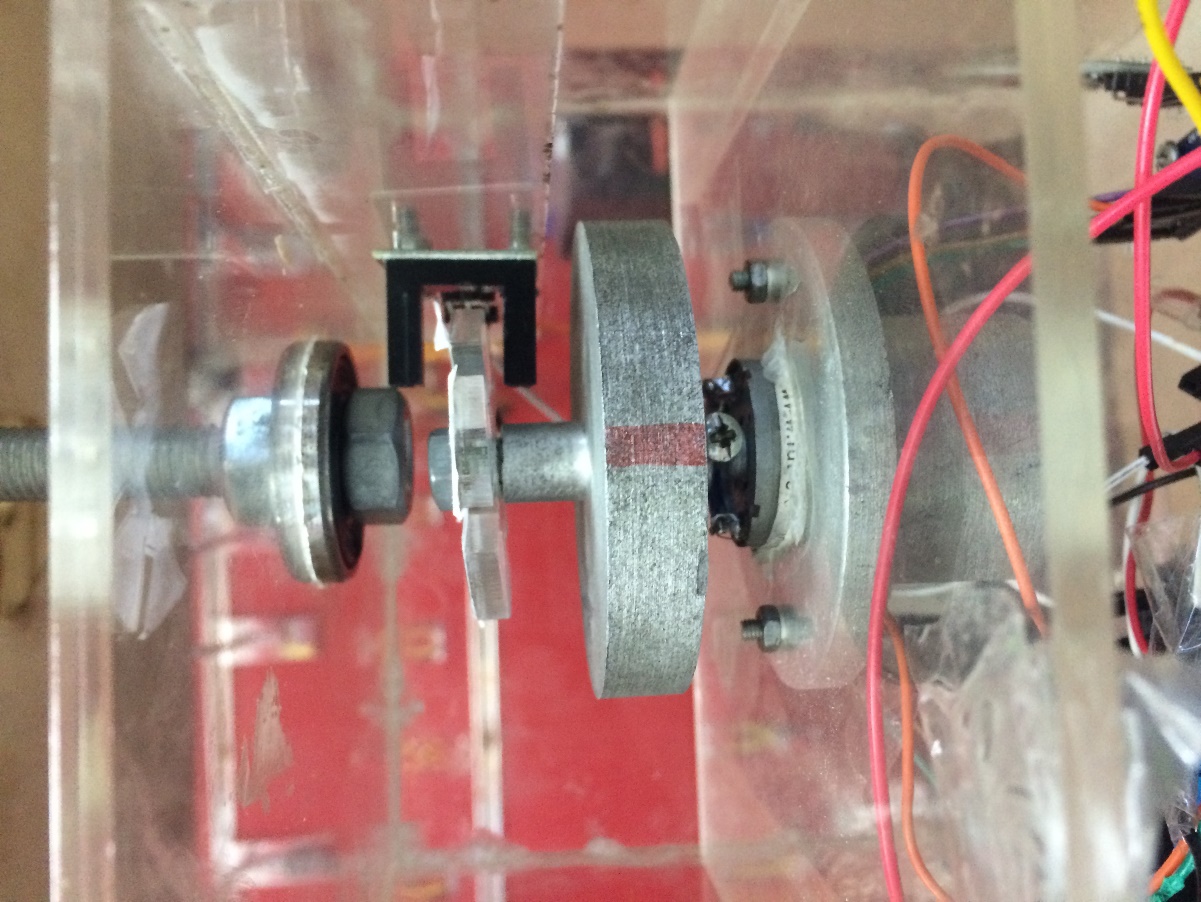
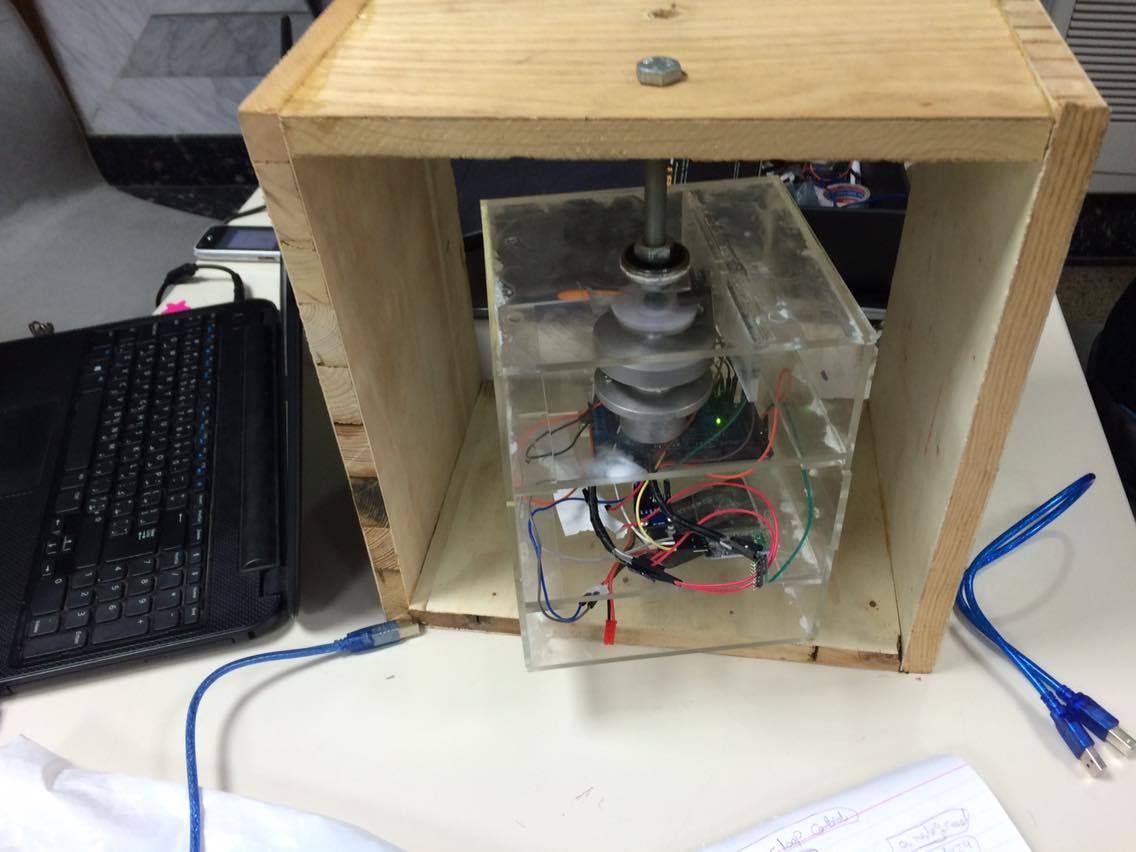


Figure 1-10: The encoder fixed

The next step was fixing the electrical components which are used in this project which are motor shield mounted on Arduino Mega, MPU-9150, SD card module, Bluetooth module, Camera, optical encoder and 12V battery. Arduino Mega with the shield was fixed by two screws the same as the MPU-9150 which was fixed horizontally as much as possible. SD card and Bluetooth modules were both fixed by using double face tape.

The final appearance for the satellite ***(Fig. 1-11)*** was that we have three stages; two intermediate stages and the bottom face. On the first intermediate stage, the motor mounted by the reaction wheel and the encoder wheel were placed. On the second intermediate stage, Arduino Mega mounted by the motor shield, Bluetooth module, SD card module and MPU-9150 were placed. On the bottom face, the 12V battery was placed. Finally, the camera was positioned on the side face of the cuboid between stages one and two.

Figure 1-11: Final appearance for the satellite while testing



# Satellite subsystems

# 2.1 On-board computer subsystem

The most important subsystem is the OBC which integrates the function of all the satellite components to fulfill the desired mission from it so Arduino Mega ***(Fig. 2-1)*** was used. Arduino Mega is a microcontroller board based on the ATmega (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Mega is compatible with most shields.

Figure 2-1: Arduino Mega board

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers if using a motor shield. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

This board has 265 KB of flash memory for storing code so it was hard to save the image taken by the camera on it as it is very small. Arduino IDE ***(Fig. 2-2)*** was used to compile the code integrating the codes running the whole system together based on a programming language called Processing which also supports the languages C and C++.



Figure 2-2: Arduino IDE interface

# 2.2 Communication subsystem

As trying to stimulate the wireless communication between sat communication subsystem and ground station. Bluetooth module HC-05 ***(Fig. 2-3)*** master/slave was used to send commands that include the required orientation. In addition, it was used to observe the cube-sat actual orientation by the gyroscope readings as well as showing the camera images as stated before. Software used for building the interface of the stimulated ground station ***(Fig. 2-4)*** is LABVIEW software.

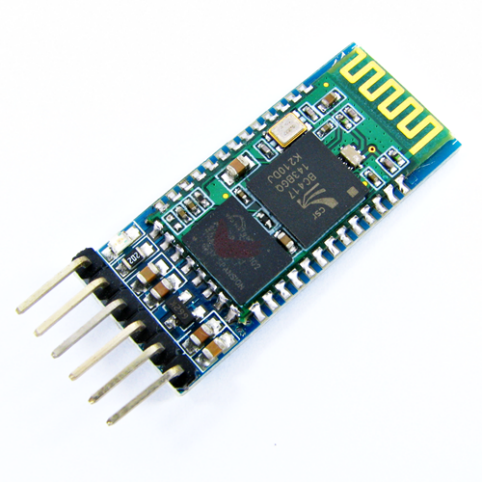


Figure 2-3: Bluetooth module HC-05

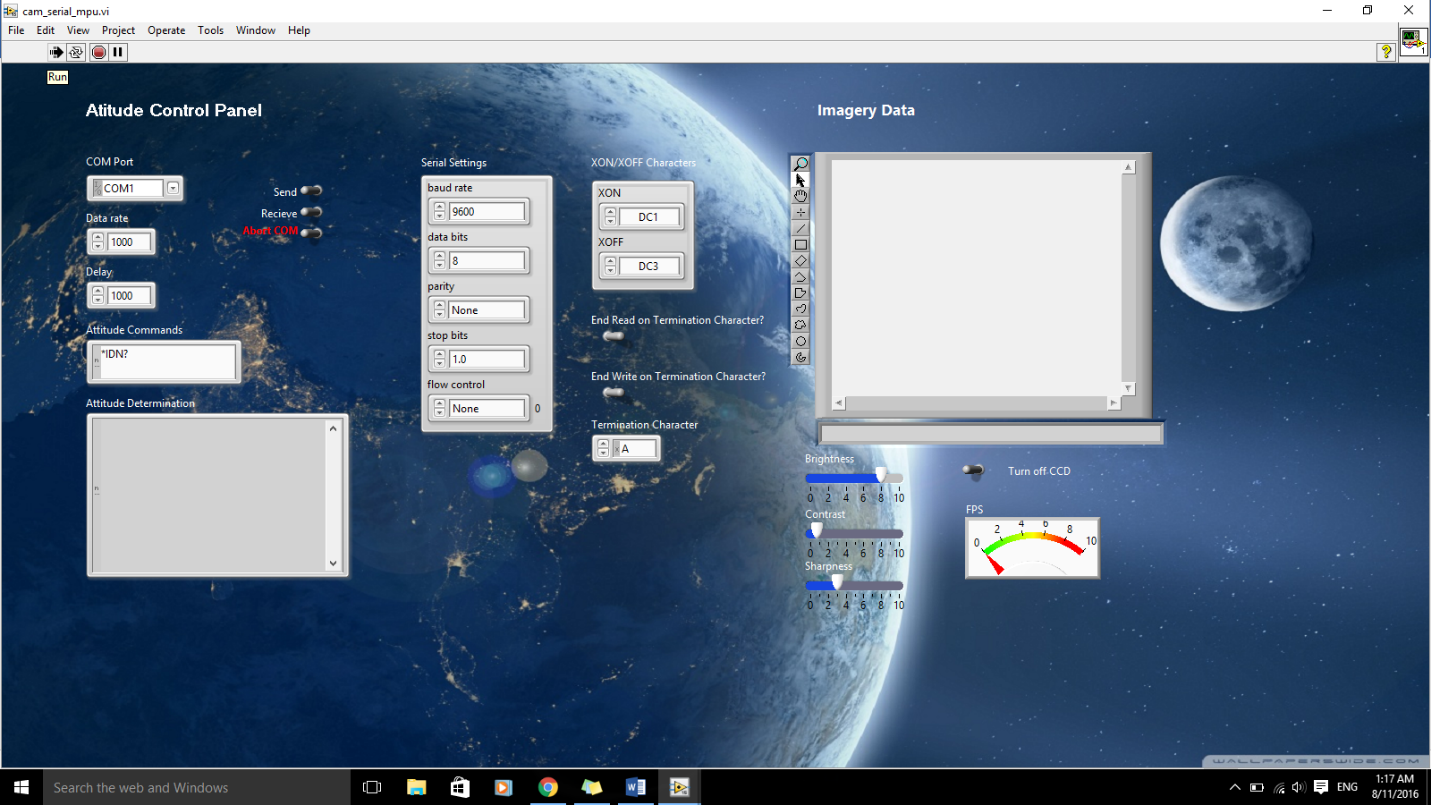


Figure 2-4: Stimulated ground station using labview

# 2.3 Payload subsystem

One of the major purposes for using the reaction wheel is adjusting the orientation for accurate imaging system. To apply that on a smaller scale in the used mock up, TTL serial camera ***(Fig. 2-5)*** was used to take images at the specified orientation. Image then has to be transferred wirelessly stimulating the real situation. Unfortunately, due to the low memory of Arduino board, SD card ***(Fig. 2-6)*** was used to store the image before transmittance process.

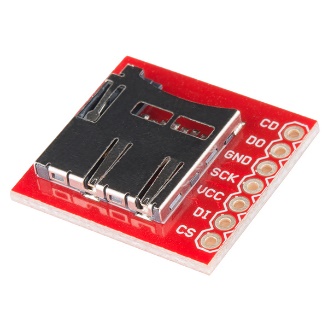


Figure 2-6: SD card module

Figure 2-5: TTL serial camera

# 2.4 Attitude determination

To be able to appropriately control the orientation, an accurate observation of the yaw angle is required. MPU-9150 module ***(Fig. 2-7)*** was used. Some filters were used to get accurate readings such as Kalman filter. However, Kalman filter was proved by experiment to be inaccurate in measuring yaw angle in some initial positions. Several iterations were implemented until the desired accuracy was achieved.



Figure 2-7: MPU-9150

# 2.5 Power subsystem

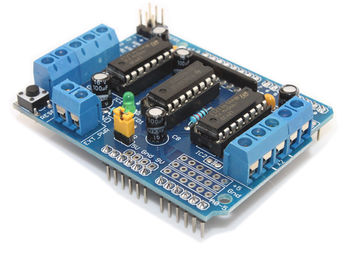
A 12V DC battery is connected to a Motor Shield to supply required power for the motor and Arduino Mega and thus for the rest of the components used. The Motor Shield ***(Fig. 2-8)*** permits us to control the speed of the motor and its direction by the use of Arduino Mega. It allows us to address Arduino pins easily as to deal easily with the motor. It supplies an output voltage up to 12V. There are three ways to power the Arduino and the Motor Shield; first we get Arduino power from USB cable and the Motor Shield power from the power external pins in this way we remove the yellow jumper found on the shield. The second method is by powering the Arduino from the DC jack and placing the jumper in its place. The third method is by connecting the power external pins of the Motor Shield to the rechargeable battery and placing the jumper. The third method was used as to minimize the cables and wires as much as possible and trying not to increase extra weight in mock-up satellite.

Figure 2-8: Motor Shield

# Control

# PID manual tuning

PID control is a very common technique for control loop feedback mechanism as the error between the desired set point and a measured process variable. A signal comes from the PID goes back to the controller to move the actuator depending on three gains; one multiplied by the integral of the error, the second one multiplied by the derivative of the error and the last one multiplied by the error itself.

https://lh4.googleusercontent.com/WExNB_vpHnfQTx7XfsoMnfCDFFqbiazmukenKMRcBlk5NAXlsb8TTgSd-1M3t8WpZ5-xAYop9sS7TGPhJc6OE9CvbHvaQwVmfke9-lNtkHA9_QyRVUL5XXdcWSoJfyT_X35LTP71

* accounts for present values of the error. For example, if the error is large and positive, the control output will also be large and positive.
* accounts for past values of the error. For example, if the current output is not sufficiently strong, error will accumulate over time, and the controller will respond by applying a stronger action.
* accounts for possible future values of the error, based on its current rate of change.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parameter | Rise time | overshoot | Settling time | Steady state error | stability |
| KP | decrease | Increase | Small change | Decrease | degrade |
| KI | decrease | increase | increase | eliminate | degrade |
| KD | Minor change | decrease | decrease | No effect | Improve if Kd is small |

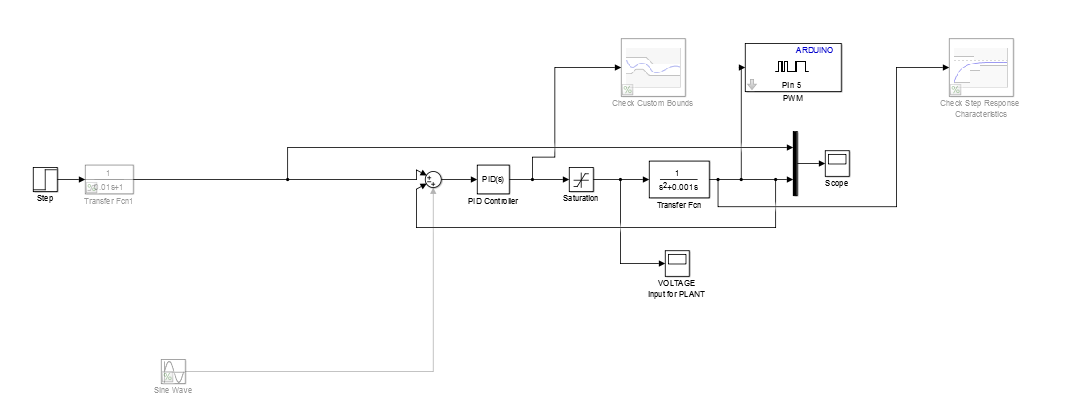
Two parallel routes were taken; the first was by trial and error it was noticed that adjusting the proportional and derivative gains at certain filter coefficient is sufficiently enough for adjusting the steady state error which gave us =7, =2 and N=100 where . It is important to declare that filtering coefficient under 100 results in a bad transient response and above 100 goes out stability margins at high time responses. This was simulated on Simulink ***(Fig. 3-1)***.

Figure 3-1: PID simulated in Simulink

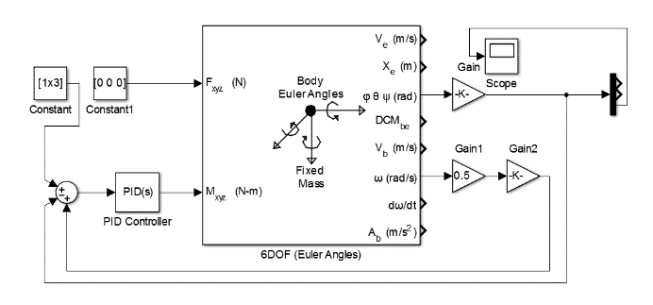
The second route was using Euler angles ***(Fig. 3-2)***. It works as the following: it has 6 inputs 3 of them are forces ,the forces in the 3 independent axis, which were all put to zero .The other 3 are torques; 2 of them are zeros and the last one is the required torque and is obtained from the PID block after getting the required theta. There are multi outputs from the Euler angles block such as linear velocity, angular velocity, linear displacement, angular displacement, linear acceleration, angular acceleration and others. Only two parameters from the output are needed which are theta and the angular velocity. The use of theta is clear a DEMUX is made which takes the yaw angle and pass it in a negative feedback loop to the PID block and the response is obtained from it while the usage of the angular velocity is multiplying it by a constant and make a positive feedback loop to the PID block to compensate the torque lost in friction.

Figure 3-2: Euler angles simulated on Simulink

The mentioned PID control above was made to the outer loop to control the angle of the satellite but another inner control loop was made to control the motor’s angular velocity using the encoder fixed above the reaction wheel. For the MPU, it senses the obtained angle and compare it with the needed angle, if the obtained angle was smaller than that of the needed angle, it sends a feedback to the system so that the system can calculate the difference and increase the angle and in the case that the obtained angle was bigger than the required one, the MPU sends a feedback to the system in order that it decreases the angle.

For the encoder, it senses the rpm (angular velocity) of the motor by observing the encoder wheel, if the rpm was higher than that in the sent command, the encoder send a feedback to the system in order that the system send another command to decrease the rpm of the motor, and in case that rpm was lower than that in the sent command, the encoder send a feedback to the system in order that the system send another command to increase the rpm of the motor which needs the motor to be modelled. The inner loop technique wasn’t used due to time restrictions.

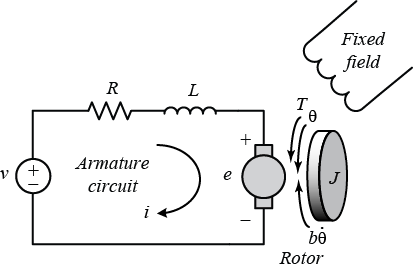


Figure 3-3: Model of DC motor

The motor transfer function is:

So the constants of the motor found in the transfer function must be estimated to be able to enter the transfer function of the motor to the control system. This could be done using parameter estimation or system identification.

# 3.2 Signal Constraint

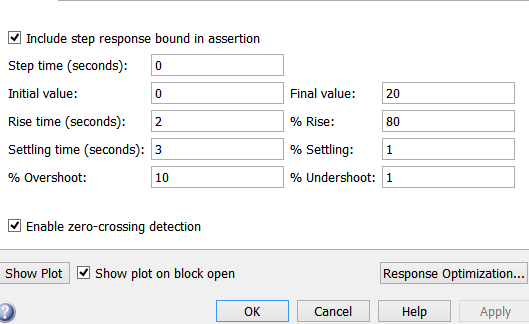
****Using Simulink system tool box. Signal constraint tool ***(Fig. 3-4)*** was used to get PID values in a methodological approach. This tool allows user to optimize the response characteristics to obtain the desired output. Examples of these characteristics are settling time, rise time & overshoot percentage. By iterating numerically until the error approaches zero. Solution converges and values of proportional, differential and integral gains are obtained.

Figure 3-4: Signal Constraint toolbox

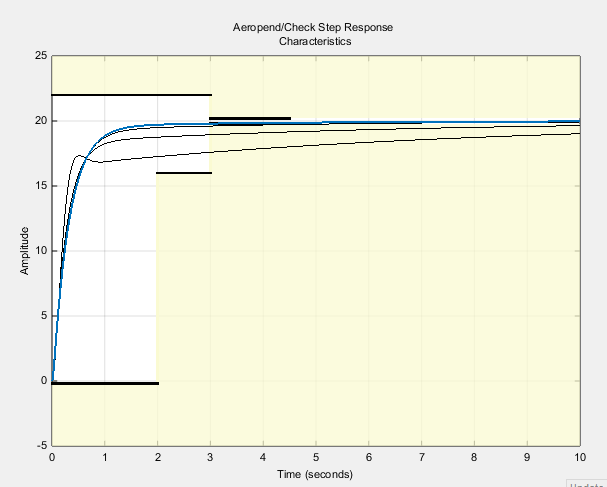
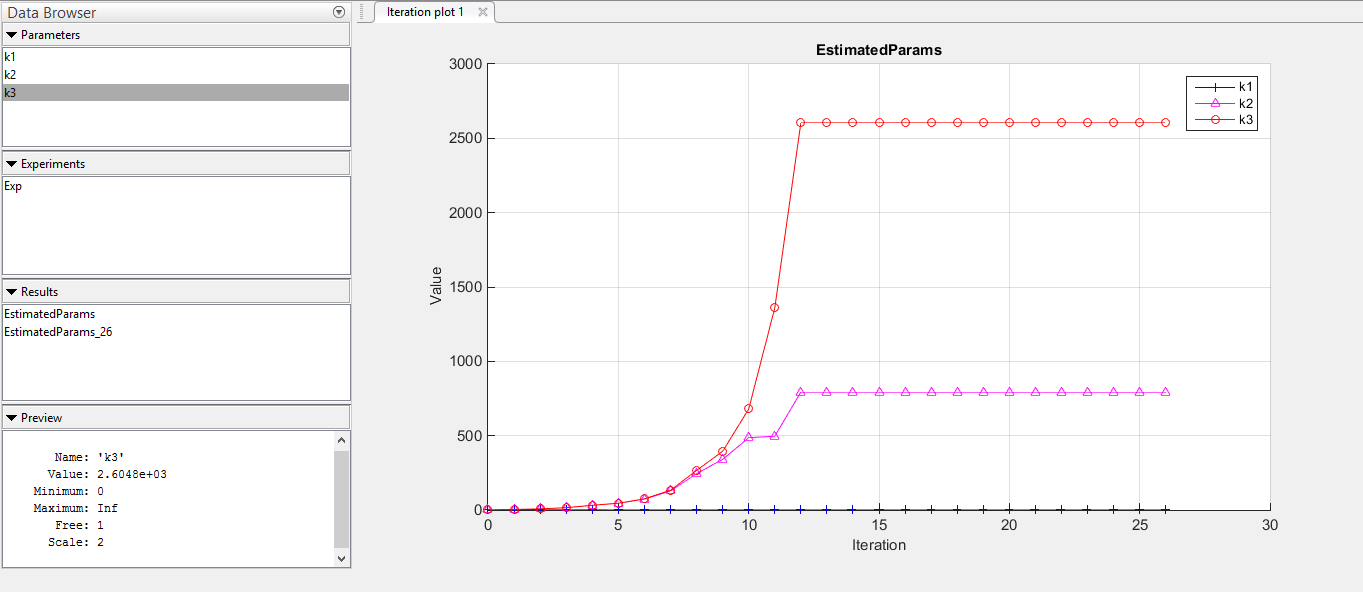
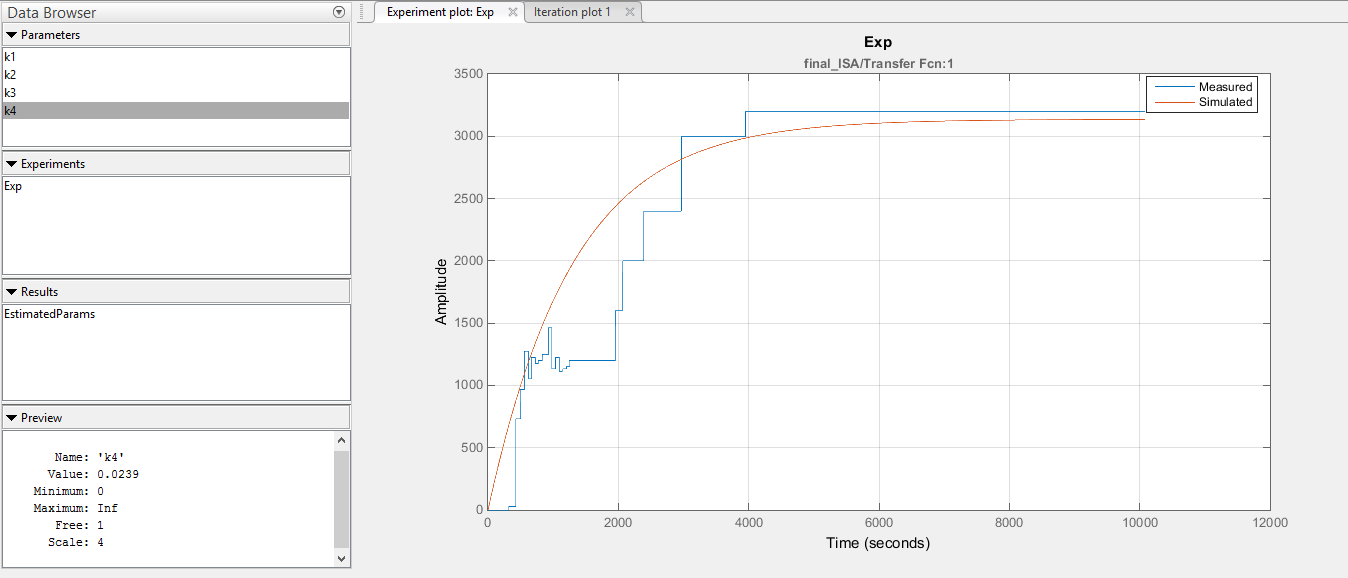


Figure 3-5: Signal constraint toolbox output

# 3.3 Parameter estimation

For parameter estimation, data acquisition is done in order to obtain the RPM(angular velocity) of the motor thanks to the encoder, then the measured data is imported and the estimated data is selected, the parameters are identified after that as well as their ranges. The following step is performing parameter estimation followed by validation using Simulink design optimization.





After iterating to solve for the unknown motor parameters, the estimated values were obtained from each line of the graph. Although this is an accurate numerical method, reasonable values of the parameters were not acquired, and further research is required to debug the source of error. So instead, PID manual tuning using adaptive PID code was the reliable method for us to get conservative and aggressive gains and reach an almost zero steady state error in a stable behavior.