Abstract

As a Cubesat is a major target of minimizing size and making satellites more practical, It was important to think of a better actuator with better size and weight. This thesis is targeting a lighter, better controlled actuator which is an Omnidirectional Magnet which will substitute the reaction wheel being used and control the Cubesat using only electromagnetic field.

Contents

1 Introduction

Permanent magnets have been always used in magnetic manipulation of medical devices [2].As a target of achieving less weight on the Cubesat, an Omnimagnet is used in the Cubesat instead of the reaction wheel to control the tilting about the three axes.
Implementation of such a project would require a sacrifice of so many influential considerations on a real Cubesat as this a low cost model of a targeted design as the target of the project as a whole is to test the efficiency of controlling a Cubesat using Omnidirectional magnets from the aspect of weight, power and control, linearizing through this the friction, and other external forces affecting a Cubesat and testing the design on a bearing allowing free movement in the 3 axes. Furthermore, the design of the Omnimagnet is based on a paper written by Petruska and Abbot (An Omnidirectional Electromagnet for Remote Manipulation) where three nested solenoids cube shaped with a spherical core inside are used.
The Thesis is divided into three main parts, First, the sensors, microcontroller and modeling software and hardware used. Next, comes the sensor data fusion which will be mpu6050. Third, the linearized model of the system with the plant designed on SIMULINK .Finally, the Conclusion of the thesis and efficiency tests measured from the data.

1.1 Objectives

At first,Develop a relation between the angles of tilting of the Cubesat and the Current supplied to the Omnimagnet through the driver circuit. Where the magnetic dipole moment is calculated as a summation of the solenoids together with the inner spherical core and then related to the current to finally model the Omnimagnet power relative to the attitude needed
Secondly, Choosing the correct power source for the Cubesat to get the power needed to use the Omnimagnet, where it is being discussed whether to use a battery which will we have a problem using it as result of its relatively high weight or design a super capacitor which will do the job. Calculate the external influences on the model done using matlab as external torques and the mass of the Omnimagnet and the Cubesat itself. And finally choosing the bearing or how will we suspend the mass of the Cubesat so that we can test the Omnimagnet without so much influence from friction and making a near model of space. By finishing this point, it means that the design constants of the Omnimagnet have been totally determined and only the plant is left.
After finishing the mathematical relations and the getting the reading from the sensor, a current and voltage will be modeled on Simulink and getting the PID controllers constants. Testing the modeled plant and the constants from matlab on the Omnimagnet and acquire a conclusion for such a first time combination of an electromagnet controlled Cubesat will be done. The rest of the time left is for testing the driver circuit on the Omnimagnet, measuring the current and field modeled before and rewriting the results in a thesis style.

1.2 Tools

Recalling that this project is testing only the Cubesat as a basic model with certain weight and specifications, and that the main target is testing the Omnimagnet in controlling such a shape and weight, It would only be a one kilogram model of a shape(not yet discussed) containing the sensors, Omnimagnet and the processor used. Until now the processor used is arduino Uno and whether or not (Myrio)will be used is being discussed based on the price, time for shipment and the availability of the processor in the labs of the university. The simulation of the design part of the Omnimagnet is done using Comsol, the control law that will be developed will be done using matlab will be a based on equations from a research done before in [1]. a model for the Earths magnet field will be a permanent magnet with known magnetic field for the calculations and the modeling of the system . The sensor being used for determining the tilting angle is the MPU6050, an IMU(inertial measuring unit) which includes a gyro and an accelerometer and getting the readings using the DMP(digital processor inside the MPU6050).
Reading from the IMU used to get the tilting angle of the Cubesat and the usage of these angles to relate it to the current supplied to the Omnidirectional magnet as a target of generating the correct magnetic field to control the Cubesat and designing a filter (a complementary filter as a first attempt and kalman filter if the readings were not accurate enough or the drift of the gyroscope has overtaken or had a huge contribution)that will combine the readings of the gyro and the accelerometer . (19/2/2017) this date might be shifted a week if the interface of the DMP is to be understood and used for accurate readings.

1.3 Previous and Current Challenges

1.3.1 Size Problem

One of the main challenges faced in design of the Cubesat was the size itself which is the target of designing the Cubesat in the first place. The size of the Cubesat has prevented the solar panels or whatever source of power used to supply enough power to the instruments used by NASA, Up until now it is not estimated that this problem will be faced on our low cost model as the power from the Omnimagnet hasnt been measured yet for driving the any external tool. The only problem for now is the weight of the power source and Omnimagnet and whether they will be in the range based on the design for a suitable power to control a 1 kg shape. Doug Rowland, a solar scientist at NASA, faced this dilemma when gathering data from his Firefly Cubesat. He built it to investigate the correlation between lightning and gamma radiation, but his Cubesat can only download 20 milliseconds slots of data to Earth each day. The Firefly just doesnt have enough electrical power to simultaneously run its GPS receiver, its communications antenna and our experiment at the same time, Rowland said. On a big spacecraft, youd have a thousand times as much data, at least, and youd have other ways to transmit the data down to Earth. Max Gleber (The future of Cubesats).

1.3.2 Sensor Readings

The second main problem that might face us, is the readings from the sensors and whether the magnetometer will be affected by the Earths magnetic field or the used permanent magnet and whether we will need to use a very low pass filter or a shield to prevent the magnetic field from affecting the readings. This problem might be solved by developing a filter between the gyro and the accelerometer and predicting the third angle where the axis is aligned with the gravity which the accelerometer cant predict based on the idea that the accelerometer can measure the forces relative to the gravitational force. The DMP inside the sensor will be interfaced as a second solution for such a problem but then the approach of how the third angle is measured is hindered.(Update on previous week) If we decided to use the magnetometer to measure the position of the Cubesat, then we will counter the problem of magnetic field from Omnimagnet coils induced that might get the sensor into saturation, one solution was mentioned to alternate the time of generating the field from the coils and measuring the angle, as when the sensor is functioning the coil is off but this may need another extension to the code and another complication that we might not need in the first place.

2 Data Fusion

2.1 Inertial Measurment Unit

Analyzing the readings from the IMU (inertial measurement unit)gives raw values at the very first so The registers adjusted in the interface of MPU6050 are predefined in a library written by Invensense (the designer of the sensor) as writing the whole code for the sensor from scratch will take a huge amount of time which is not our main target to interface the sensor from the beginning.

2.2 Offsets

the offsets of the sensor has been calculated using simple math equations by taking samples from the readings of the sensor every 10 ms and adding around 1000 samples in a variable it was divided by the number of samples to get an average for the offset . Next was determining the sensitivity of the sensor which was normally adjusted at 2g (accelerometer) and 250 degree/s for the gyro. I readjusted these sensitivities to decrease the sensitivity of the sensor as there are so many factors affecting the sensor from moving cars or various vibrations in the background. It was adjusted to be 4g for the accelerometer and 1000 degrees/s for the gyro (Using 0x1B and 0X1C registers of the sensor as mentioned in the datasheet).

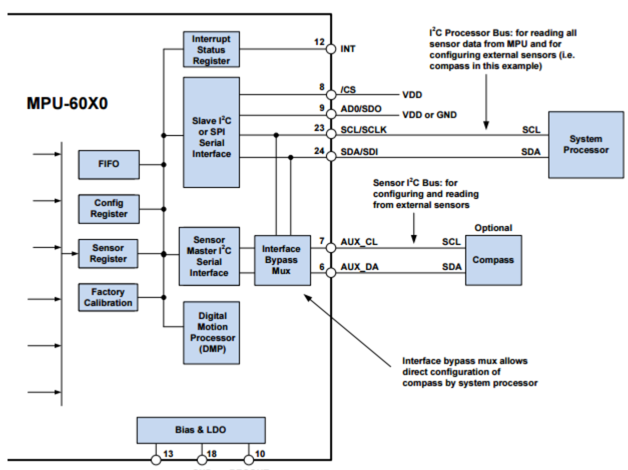


Figure 1: MPU signals system

The offsets calculated for X, Y and Z direction was 140, 24 and -90 respectively. After subtracting the offsets from the sensors raw reading, the raw values of the gyro and the accelerometer was then divided by their sensitivity mentioned in the datasheet of the sensor which are 8100 for the accelerometer and 32.8 for the gyro based on a fit made at 25 degree Celsius. The accelerometer conversion from raw values to degrees is done using simple math by relating the forces to the gravity of earth by : ¡atan2 (accy, accz)*(57296.0/1000.0)¿ , where atan refers to arctangent and the ratio at the end is the conversion of radian to degree. A complementary filter is used in the first place to check the accuracy and the offsets calculated by the samples where it uses the stability of the gyroscope together with a low pass accelerometer reading where a ratio of 0.99 to 0.01 was given to the gyro and accelerometer respectively : 0.99 * (comp Angle + gyro * dt) + 0.01 * roll, where dt is the change in time and roll is the angle measured from the accelerometer after conversion. Further data fusion will be done to calculate the third angle as the accelerometer together with the gyro can only calculate the pitch and the roll only as mentioned before but for now will we consider the data from the DMP sufficient enough for testing the whole system and then turning back to the concept behind the DMP and the sensor interface as the microcontroller might be changed for a better a speed and accuracy of the system. Shown here, is the readings after measuring the offsets and showing consistency of the readings with a complementary filter with only the pitch and the roll at zero degree angle.

2.3 Digital Processor

The Sensor as mentioned before, uses the digital processor inside which automatically generate readings for the Euler angels, quaternions and acceleration with and without the gravitational components which will help us in the dynamic model and let us handle the idea of rotation of matrices that will be mentioned in the next chapter.

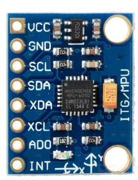


Figure 2: Image courtesy of [arduino playground]

3 Quaternions and Euler Angles

3.1 Gimbal Lock

As we are dealing with rotations and orbital mechanics, one can define these rotations and movement of the rigid body in space using Euler angles (, ,), Euler angles has always been used to represent such mechanics, but practically when we represent a rotation from frame another using Euler angles, it will have a singularity at a certain position where we divide by cos and the angle equals 90 degree where the function is undefined at this position and the matrix representing this frame or rotation will span the R2 where it defines only 2 dimensions of the three which is called in mechanical representation a Gimbal lock, without getting into too much details in that part, the better solution for representing the orientation in space has always been the quaternions which is a vector representation together with a scalar part where it defines the R3 using four parameters and have no singularity, q = (W, I, J, K) where represents the scalar part the and the three other parameters are the vectors part. Further information on quaternions can be found in [1]. Below can be found a representation of Euler rates in terms of angular velocity which has a rotation matrix base on Euler angles[1]:

$$\frac{d}{dt}\begin{bmatrix}\theta\\ \phi\\ \emptyset\end{bmatrix} = \begin{bmatrix}1 & S\Theta T\phi & C\theta T\phi\\ 0 & C\phi & -S\phi\\ 0 & S\phi/C\theta & C\phi/C\theta]\end{bmatrix}\begin{bmatrix}P\\ Q\\ R\end{bmatrix} \tag{1}$$

Where θ , ϕ , $\emptyset$ represents the 3 euler angles and their rates. It is obvious in the last raw that it is divided by Cos theta which will be undefined at some orientation making such a singularity that should be taken into consideration.

3.2 Quaternion definition

According to [16]Quaternions have 4 dimensions as mention before (each quaternion consists of 4 scalar numbers),one real dimension and 3 imaginary dimensions. Each of these imaginary dimensions has a unit value of the square root of -1, but they are different square roots of -1 all mutually perpendicular to each other. So a quaternion can be represented as follows:

$$a + ib + jc + kd \tag{2}$$

While the complex numbers are obtained by adding the element i to the real numbers which satisfies $i^2 = 1$, the quaternions are obtained by adding the elements i, j and k to the real numbers which satisfy the following relations.

$$i^2 = j^2 = k^2 = -1 \tag{3}$$

$$i \times j = -(j \times i) = k \tag{4}$$

$$j \times k = -(k \times j) = i \tag{5}$$

$$k \times i = -(i \times k) = j \tag{6}$$

3.3 Frames

The orientation of a rigid body in space has been mention that is representing the quaternion, but such an orientation is based on a frame which defines the point and refer it to a reference such as the inertial frame of the Earth, from that point we can mention the frames that we will be dealing with through controlling the position of the Cubesat in space and also the angular velocity and acceleration.

3.3.1 Inertial Frame

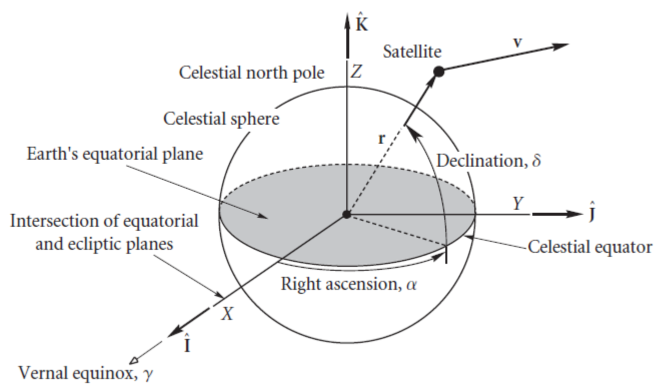


Figure 3: Inertial body Frame

Its Origin at the center of Earth, z-axis pointing out of the north-pole, x and y-axes fixed in space which is not an accelerated frame.

3.3.2 Body Frame

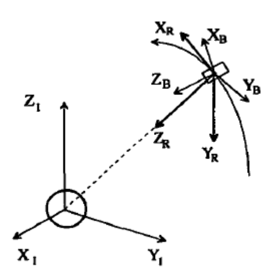


Figure 4: Body Frame

Origin at the center of mass in the satellite, where x, y and z moves and rotates with the satellite. y is the axis of maximum inertia. Z is the axis of minimum inertia. x is defined as direction of travel. This frame will be our reference frame based on the readings from the sensor instead of rotating each reading but some other parameters will need to be rotated from the inertial frame after though.

3.3.3 Orbital Frame

This frame is located at a distance(R_{earth} + altitude) from the center of Sensors at the center of the satellite. Its z-axis always points towards the center of Cubesat and the x-axis points in the velocity direction according to [3].

3.4 Attitude Operator

Imagining a Basis vector (orthonormal) X and Y where we want to rotate from one frame to another using rotation matrices, let $A = XY*$ where A is the attitude operator and * denotes the adjoint of linear transformations. A here is orthogonal based on the orthonormality of X and Y where $AA*$ is the identity operator, In brief this expression attitude operator will have the eigenvalues$(1, e^j\theta, e^-j\theta)$ where the first eigenvalue will represent a rotation about it with an angle θ and of course the multiplication here of the eigenvalues will be equal based on the right hand convention of the basis vector. As explained by [3].

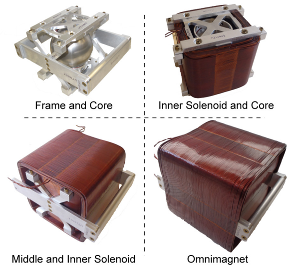


Figure 5: Image courtesy of [4]

4 Modeling

4.1 Actuator Modeling

Moving on to get a relation between the current that will be induced in the coils to generate a certain amount of field to get also a specific dipole moment that will control the attitude of the Cubesat which has been the second main objective in this thesis. One can define the dipole moment derived by [4] on a model for the Omnimagnet being used in this thesis with the ferromagnetic core and three cuboidal coils. Based on [3] :

$$T = M \times B \tag{7}$$

Where B is the magnetic field, T is the torque and M is the magnetic dipole moment induced. Having M_x, M_y and M_z being the components of the dipole moments:

$$\begin{bmatrix} J_x(8LR_c^3 \int_{B1}^{B2} \frac{1}{\sqrt{(1+2s^2)}} Ds + \frac{L^4}{6}(B2^3 - B1^3) \\ J_y(8LR_c^3 \int_{B1}^{B2} \frac{1}{\sqrt{(1+2s^2)}} Ds + \frac{L^4}{6}(B2^3 - B1^3) \\ J_z(8LR_c^3 \int_{B1}^{B2} \frac{1}{\sqrt{(1+2s^2)}} Ds + \frac{L^4}{6}(B2^3 - B1^3) \end{bmatrix} \tag{8}$$

then from (3) we can separate the current from the equation and have the torque after that in terms of current.

let $J_x(8LR_c^3 \int_{B1}^{B2} \frac{1}{\sqrt{(1+2s^2)}} Ds + \frac{L^4}{6}(B2^3 - B1^3)/area$ equals to K . then we have :

$$\begin{bmatrix} K & 0 & 0 \\ 0 & K & 0 \\ 0 & 0 & K \end{bmatrix} \begin{bmatrix} I_x \\ I_y \\ I_z \end{bmatrix} = T \times B \tag{9}$$

then we can have the current to be :

$$I = K^{-1}TS(B) \tag{10}$$

where $S(B)$ is the skew symmetric matrix of the magnetic field which keeps us from performing cross products.

4.2 Rigid Body Modeling

referring to [7], we can say that the dynamic model for any rotating body can be represented by

$$\dot{h}_{total} = N_t - w \times h_{total} \tag{11}$$

where h is the angular moment of the body,N_t is the total torques acting on the system(external) and the last term is the decoupling term where w is the angular velocity of the body.
illustrating this equation and setting the inertia of the body to be constant we get

$$I\frac{d}{dt}w = N_c + N_d - ws(w)I \tag{12}$$

where I is the inertia of the body represented in a vector form and multiplied by the identity matrix to satisfy the equation size and N_d, N_c are the disturbance and control torques respectively. and then to get the equation in terms of the $\dot{w}$:

$$\dot{w} = I^{-1}(N_c + N_d - ws(w)I) \tag{13}$$

and of course the term $s(w)$ is

$$s(w) = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 - w_! & \\ -w_2 & w_1 & 0 \end{bmatrix} \tag{14}$$

as described before by [7]. the kinematics equation relative can be derived and related to the angular velocity of the system and it is described by

$$\dot{q} = \frac{1}{2}\lambda q \tag{15}$$

where λ is the skew symmetric matrix of the angular velocity and the $\dot{q}$ is the rate of change of the quaternion position of the system, as the angular velocity here is assumed to be constant to satisfy the conditions of calculating the position change.

this compact notation together with dynamic relation is referred to in the block diagrams in Simulink where the kinematic notation can be rewritten and separated as a vector and scalar part which gives

$$\dot{g} = -\frac{1}{2} w \times g + \frac{1}{2} q_1 w \tag{16}$$

and the rate of change of q_1 which is the scalar part is

$$\dot{q_1} = -\frac{1}{2} w^T g \tag{17}$$

where g is the vector part of the quaternion and these notations are the form used in simulating the kinematic block of Simulink.
we can then come to a combined dynamic and kinematic form for the system which will be further linearized and tested with a non linear controller in the next section.

$$\frac{d}{dt}\begin{bmatrix} w \\ g \\ q_1 \\ h \end{bmatrix} = \begin{bmatrix} I^{-1}(N_c + N_d - ws(w)I) \\ -\frac{1}{2} w \times g + \frac{1}{2} q_1 w \\ -\frac{1}{2} w^T g \\ N_c \end{bmatrix} \tag{18}$$

4.3 Quaternion Error

To design a controller that would fit the system and acquire a stable form, an error for the attitude of the cubesat should be derived and as we are dealing with quaternions and a body oriented in 3 dimensions we cannot calculate the error position using subtraction normally so the following equation based on [4] represents the error calculation relative to rotations in space, same as euler angles but we can say

$$R(\overline{q}) = R(q_d)R(q) \tag{19}$$

where q_d is the desired quaternion. this equation is then calculated using the complex conjugate of the desired quaternion

$$\overline{q} = q_d^* \otimes q \tag{20}$$

where it translates to

$$\overline{q} = \begin{bmatrix} q_{1d} & -g_d^T \\ g_d & q_{1d} I_{3x3} - S(g_d) \end{bmatrix} q \tag{21}$$

Taking the error signal to be the last 3 terms of the quaternion product then the error signal will be

$$\epsilon = q_{1d} g - q_1 g_d - g_d \times g \tag{22}$$

where g as mention before is the gibs vector which is the vector terms of the quaternion and the q_1 is the scalar part of the quaternion.

5 Non-Linear Controller

Before dealing with the controller introduced the error derivative will be calculated to be a another reference for the controller

$$\dot{\epsilon} = \dot{q_{1d}} g + q_{1d}\dot{g} - \dot{q_1} g_d - q_1 \dot{g_d} - \dot{g_d} \times g - g_d \times \dot{g} \tag{23}$$

5.1 PD (Non-Linear

The first controller that will be tested is a simple non-linear pd controller based on the error and derivative of the error signals but without a reference to the angular velocity of the body which might cause problems that will be discussed later on but firstly let's define the controller

$$\tau = k_p \epsilon + k_d \dot{\epsilon} \tag{24}$$

where this controller just prevents the oscillations in the system through the derivative part with together with simple magnifying of the error using the proportional part, it can be mention that this controller with such low magnetic field might not be capable of tracking the target but the stability test of the system will be done using lyapunov equation which will be discussed further on in the thesis

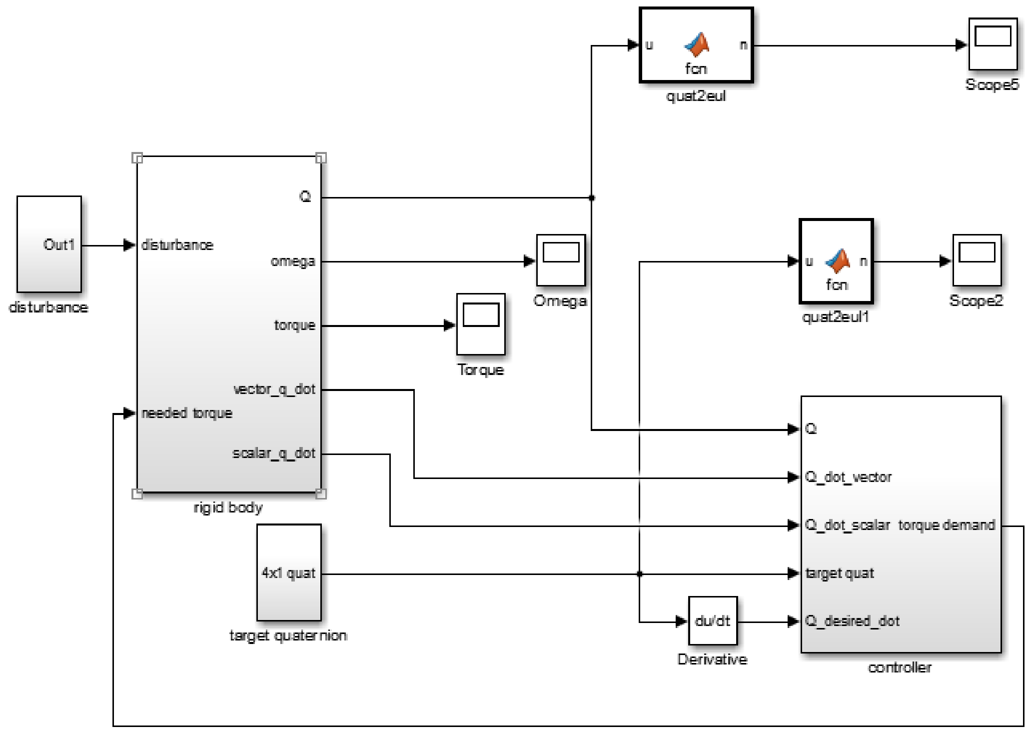
5.1.1 Lyapunov Stability Test

6 Simulation

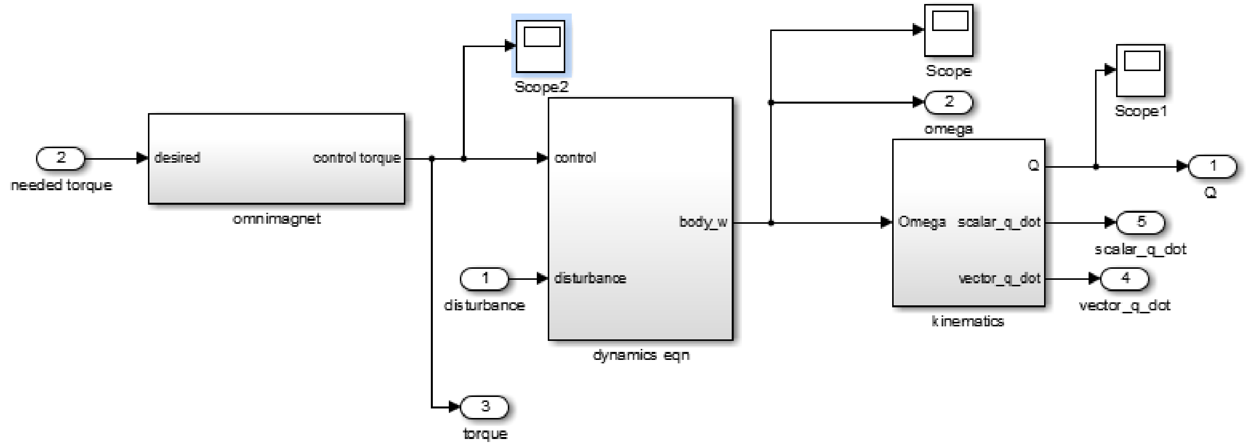
7 References

1 petruska,an omnidirectional electromagnet for remote manipulation,2013.
2 zdenko tudor, design and implementation of attitude control for 3-axes magnetic coil stabilization of a spacecraft,2011
3 john ting-yung wen and kenneth kreutz-delgado, the attitude control problem,transactions on automatic control, IEEE, 1991.
4 fossen,2011

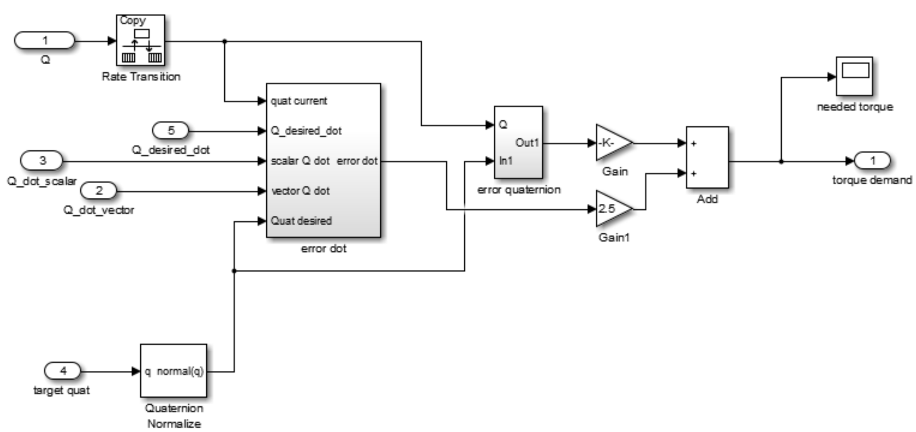
A Simulink Model



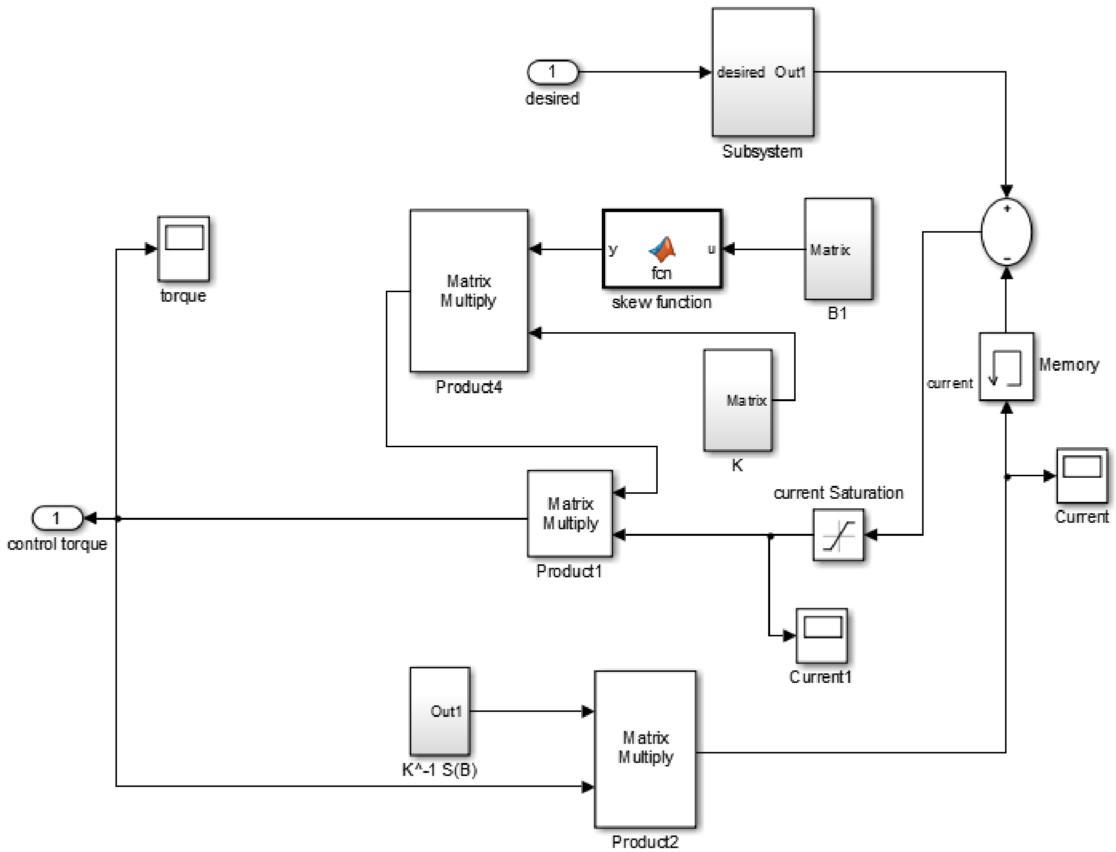
Rigid Body Model with controller



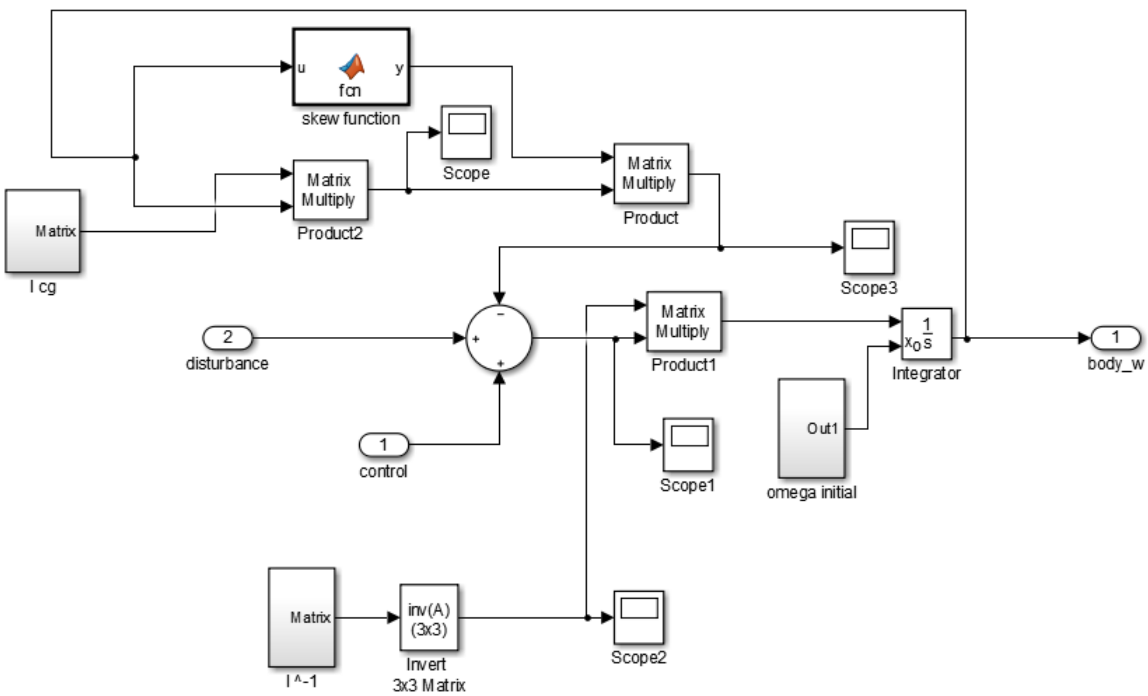
Dynamic and Kinematic model



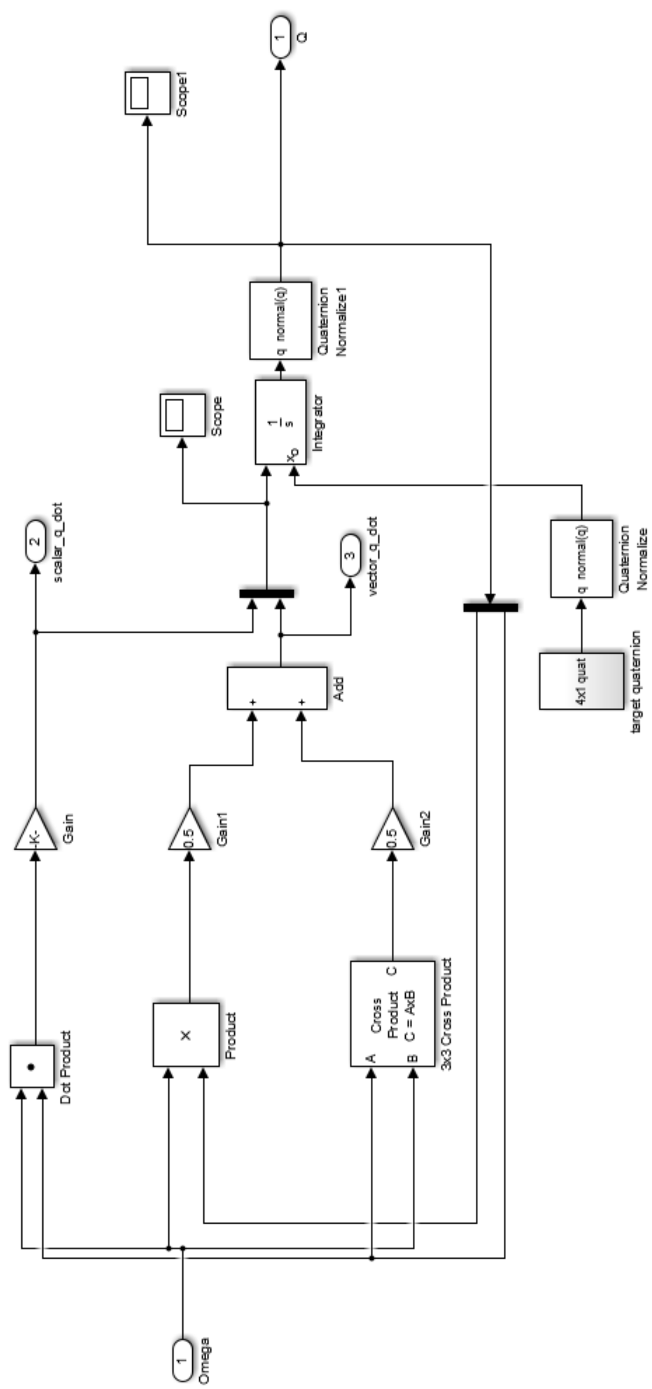
Controller model



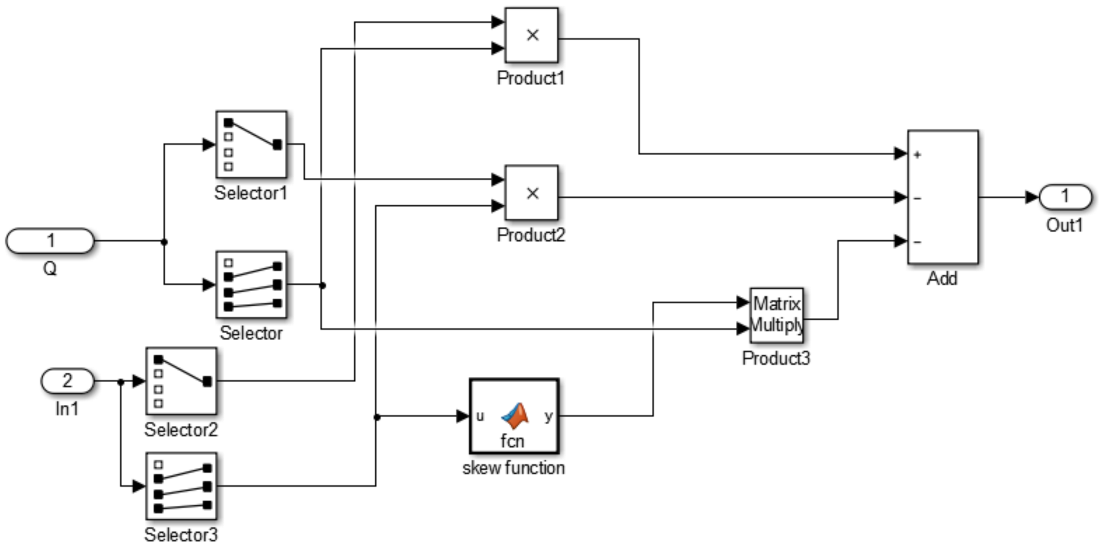
Omnidirectional Magnet Model



Dynamics Model



Kinematics Model



Calculating Error

B Sensor Readings

```
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.50    0.82
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.50    0.82
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.51    0.81
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.51    0.81
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.51    0.81
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.51    0.81
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.51    0.81
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.52    0.80
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.51    0.80
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.52    0.79
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.52    0.79
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.52    0.79
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.52    0.79
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.53    0.78
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.53    0.78
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.53    0.78
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.53    0.78
quat    0.75    0.00    -0.01   -0.66
ypr     82.00   0.53    0.78
quat    0.75    0.00    -0.01   -0.66
```