

EARLY STAGE DEVELOPMENT OF DYNAMIC SIMULATION PLATFORM FOR REACTION WHEELS CONTROLLED CUBESAT MODEL

Ryan Fadhilah Hadi
Dr. RiantoAdhy Sasongko
Dr. Ridanto Eko Poetro

Abstract

In the last decade, the trend in utilizing Cubesat as research and commercial purposed spacecraft has significantly increased. Until 2014, there are around 300 Cubesats that have successfully been deployed or already achieved its mission objectives. Yet, it is still considered as costly technology, as well as the way to launch it into orbit – \$10,000 to build and around \$100,000 to launch. By looking at these facts, the design process of Cubesat has to be considered much to eliminate the risk of failure.

One of the crucial subsystems that have to be designed well regarding Cubesat's mission which are commonly for earth observation and technology demonstration is the Attitude Determination and Control Subsystem (ADCS). The ability of Cubesat to control itself to a specific desired attitude is the key in achieving the mission objective. The limitations of space and power inside the Cubesat also have to be taken into account. Therefore, the performance of ADCS has to be arranged at its best. This condition can be achieved by having a simulation process.

In supporting the simulation process, a simulation platform is developed. The simulation platform which consists of cubesat model with its subsystems, inertia platform to support it, and system to support Hardware in Loop Simulation (HILS) is intended to simulate the attitude control performance of the Cubesat model. The simulation platform is also can be used to show the dynamic phenomenon (e.g: gyroscopic phenomenon) for academic purposes. Even though, the simulation platform still has several limitations such as friction and effect of gravity gradient.

In this early stage, the ADCS and On-Board Computer (OBC) is simplified by replacing it with microcontroller, the actuators that are used are modified brushless DC motors to mimic the actual reaction wheels, and the HILS system is achieved by using combination of MatLab & SimuLink through serial communication. The process ends with having several case studies to be evaluated.

This paper discusses the early stage of development process of the simulation platform with objectives to design, manufacture, and evaluate the simulation platform, and to be able to run an open loop simulation of the designed model. In the future, this platform can be used to simulate particular mission as case study, and to assess the performance of designed ADCS.

Key Words: Cubesat, Simulation Platform, ADCS, Attitude Control, HILS, Open Loop Simulation.

1. Introduction

Currently, the importance of spacecraft cannot be isolated from our daily activities. It has fundamental roles in current information era. A quite number of Cubesat are being designed, manufactured, and operated by students to study and develop satellite technology. Some space agency like NASA and ESA also give students chance to launch their Cubesat to orbit earth in low orbit for free, piggybacking their rocket which having mission to outer space.

In the process of designing and developing Cubesat, there are a lot of aspects that have to be specified. Every possible thing that may happen has to be calculated and planned to avoid failure. Therefore, a simulation of every component of satellite, including mission and launch simulation, is usually performed to check the reliability and performance of satellite with respect to disturbances in its mission environment.

This research is intended to design and develop a simulation platform for Cubesat ADCS development. It is also a continued development of previous research by Hagarly¹, who made an inertia platform to simulate the dynamic performance of planar platform affected by reaction wheels. The main difference is that author's research is developing the ADCS and platform for a specific size of Cubesat instead of planar satellite platform.

2. Methodology

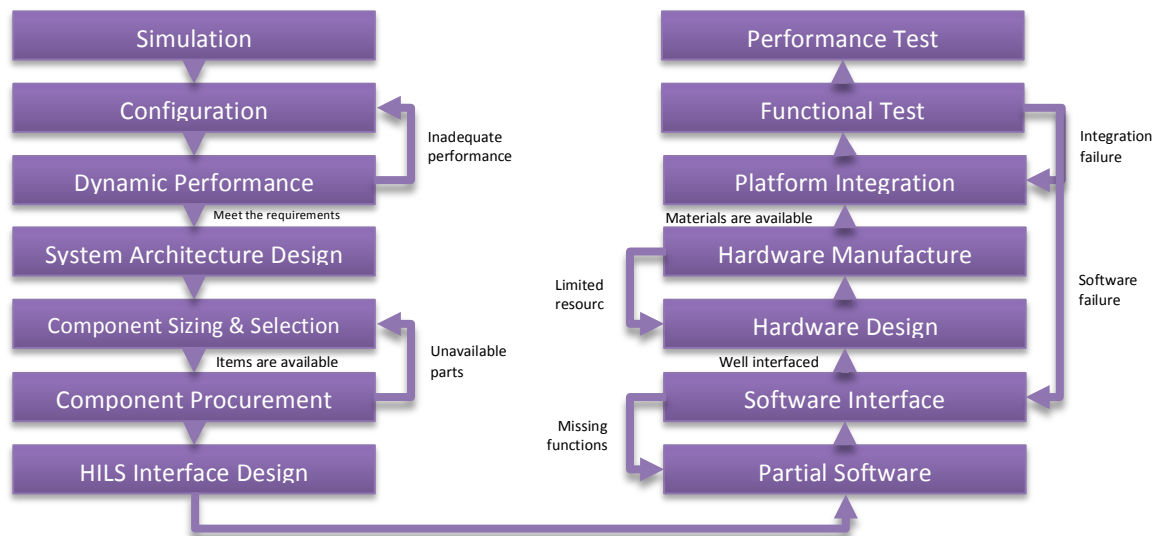


Figure 1 – Research Methodology

At first, the numeric simulation developed in Matlab and Simulink environment is conducted by using the design tool² that has been developed before. The simulation includes configuration and dynamic performance evaluation. After that, the system architecture is designed. By looking at the result of the simulation, and the system architecture, the component sizing and selection is conducted.

The next process is procuring components needed, both mechanical and electrical items. And then, the design of Hardware in Loop Simulation interface is performed. After that, based on the design, the software interface is developed and the hardware platform is manufactured. At the end, the simulation platform is tested after it is integrated. The testing includes functional test and performance test.

3. Simulation Platform

3.1 Hardware In Loop Simulation (HILS)

In running the simulation platform, the concept of Hardware in Simulation Loop is used. This concept involves interfacing between several systems that shown below:

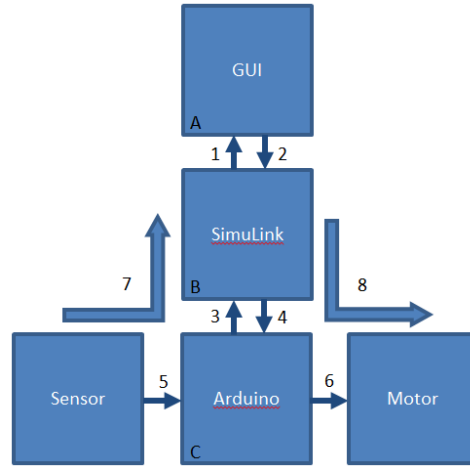


Figure 2 – HILS Architecture

From the figure above, the systems and interfaces can be described in the table below:

A	GUI	GUI is used to visualize the actual data received, and to send command to the simulation platform. Developed in MatLab using GUIDE Tool.
B	SimuLink	Simulink is used to bridge the GUI and Arduino. Also used for the control system being used.
C	Arduino	Arduino is used as data and command handler. Data is sent using serial communication. The value received from sensors converted to byte data type before it is written via serial function. While the data received from Simulink converted to float before it is sent to ESC. The arduino is also used to calculate angle and angular velocity using complementary filtering method.
1	GUI to SimuLink	The communication achieved by assigning blocks in Simulink as real-time object in GUI. The GUI then process the real-time object and visualize it.
2	SimuLink to GUI	The communication achieved by assigning parameter in the GUI to Matlab base workspace. And then, the Simulation Command of "Update" is used to update variable inside the Simulink blocks.
3	Arduino to SimuLink	The communication is done using Serial Receive block in the SimuLink. The serial data type sent by Arduino is single. Data Type Conversion is used to convert the data type from single to double, to be able to be processed further.
4	SimuLink to Arduino	The communication is done using Serial Transmit block in the SimuLink. The serial data type sent to Arduino must be single, while the data type being processed inside the Simulink is double. Data Type Conversion is used to convert the data type from double to single, to be able to be read by Arduino.

5	Sensor to Arduino	The communication is achieved by using I2C concept. "Wire.h", a library used for I2C is used in the arduino to access the data needed. The data that is received contains gravity force magnitude on each axis (Acc_x, Acc_y, Acc_z), raw angular velocity on each axis (Gyr_x, Gyr_y, Gyr_z), and filtered angle on each axis (Ang_x, Ang_y, Ang_z).
6	Arduino to Motor	The communication is achieved by using "Servo.h" library, where the ESC is assumed as servo with range of PWM input from 0 to 180. The PWM input then calibrated to get the null, min, and max RPM of the motor.
7	Sensor to GUI	The sensor measures a value. The value then read by Arduino using I2C concept. This value then converted to byte data type, sent it to SimuLink using serial. The SimuLink then receives the value, converts it to double data type, and sends it to a scope. The scope then assigned as real-time object by a function in the GUI, so the GUI can get the value and visualize it.
8	GUI to Motor	Users input command in the GUI. A function in the GUI then assign the input value to Matlab base workspace and use "Update" command in the Simulink. The input value then processed in the Simulink, and the results sent to Arduino by using serial transmit. The Arduino then read the serial transmission, separate it for every 4 bytes received, and compile it as a data with float data type. The compiled data then sent to the ESC to drive the motor.

3.2 Hardware

The platform is using mechanism of common gyroscope where the spinning wheel inside is replaced with the satellite. By using this configuration, the satellite then can be placed as there is no gravity force acting on it, so it can change its attitude freely³. This mechanism is simple and easy to build. Its disadvantage is only come from its bearings that used to eliminate the friction in the rotating axes. The friction in the bearing is still can affect the dynamic performance.

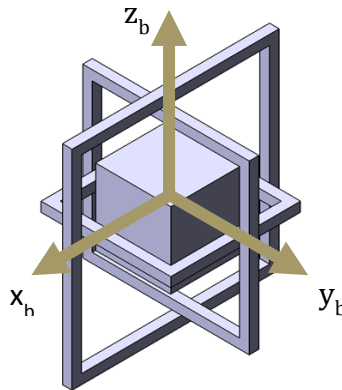


Figure 3 – Inertia Platform Design

The attitude determination module used is a 6 Degree of Freedom Micro Electro-Mechanical Sensor (MEMS), includes an accelerometer and a gyro. The module is responsible to determine current state of attitude (angle and angular velocity) of the cubesat. These sensor modules then

send the data obtained to On-Board Computer (OBC). The OBC itself is substituted by a Microcontroller Unit (MCU) to reduce cost and improve the flexibility of the required software application.

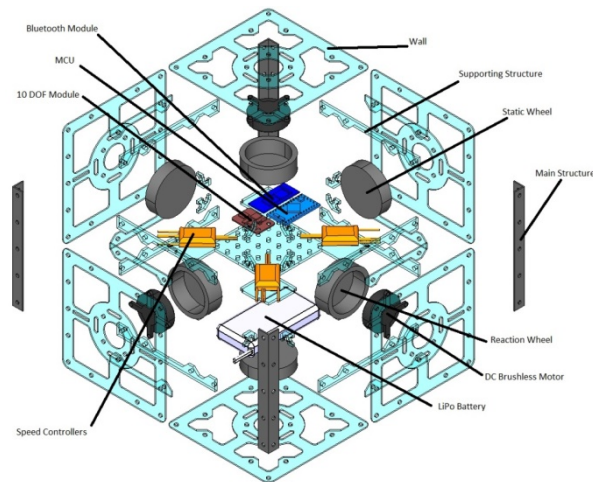


Figure 4 – Exploded View of Cubesat Model Design

To control the attitude of cubesat model, a set of reaction wheels are used. The reaction wheels are driven by high torque DC brushless motor. To simplify the brushless control design, Electronic Speed Controllers (ESCs) are used to control the rotation direction and angular velocity of the brushless motors. The motor is not equipped with encoder, and can only operate one way by receiving command from ESC. The power provided by a 7.4 V Lithium Polymer Battery. In the making of structure model, aluminum and acrylic glass are used. Machined aluminum billets are also used as reaction wheels attached to the brushless motors, and static wheels to balance the mass³.

3.3 Software

First of all, there is a Graphical User Interface (GUI) that developed to be used by user. The GUI can be used to see the actual state of the simulation platform (e.g: angle, angular velocity, and RPM of the reaction wheels on each axis). Users also can use it to give command to the system, such as running the reaction wheels on specific RPM, or giving desired angle for closed loop simulation.

The data visualization function in the GUI comes from obtained data processed by MCU. The MCU processed the data from 6 DOF sensor, with results of 6 DOF raw data (gravity force magnitude from accelerometer and angular velocity from gyro on each axis) and calculated angle and angular velocity using complementary filtering method.

The command function in the GUI is processed in the Simulink that run in the background. The processed command then sent to MCU to be applied by ESC to drive the reaction wheels. There is also a control system scheme in the Simulink to run the closed loop function. So, the MCU only acts as data and command handling, while the control system is ran by Simulink.

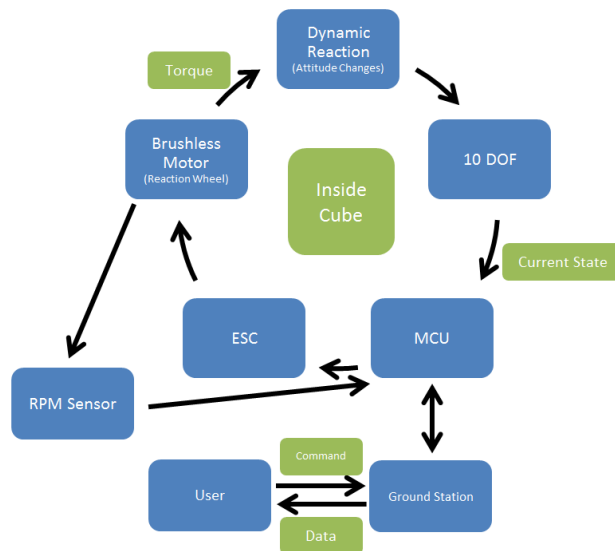


Figure 5 – Simulation System Workflow

4. Test and Evaluation

4.1 Command & Data Handling

There are two set of handling evaluations that are conducted in this part:

1. Data Handling, consists of Sensor Reading which ensures that the sensors that are used are able to read the actual condition that is happening at the time; Simulink Serial Receive which ensures that the Simulink can receive the data from microcontroller and send it to the GUI; and GUI Real-time Object (RTO) Receive which ensures that the GUI is able to show the data.
2. Command Handling, consists of Command Assigning which ensures the ability of the microcontroller to assign command to a specific component in the right way; Simulink Serial Send which ensures that the Simulink can send the command from GUI to microcontroller; and GUI Simulation Update which ensures that the GUI is able to send the desired command to the simulation platform.

4.2 Functional Test

The functional test is divided into two parts, static and dynamic test. The static test ensures the ability of the IMU to read the situation. At first, the platform is rotated for 45 degree, clockwise and counter-clockwise, in every axis consecutively.

From the figure below, it can be seen that the IMU sensor measured the correct angle of the platform. It also can be examined that when the model is rotated in an axis, the angle in other axes are also following. This is because of the coupling between all the axis that is occurred because of the measurement are all sourced from same data which are the data from accelerometer and gyro. But, the DMP has a function to compensate this phenomenon, and this compensation can be seen obviously because the angles in other axes are going back to its original state right after they changed.

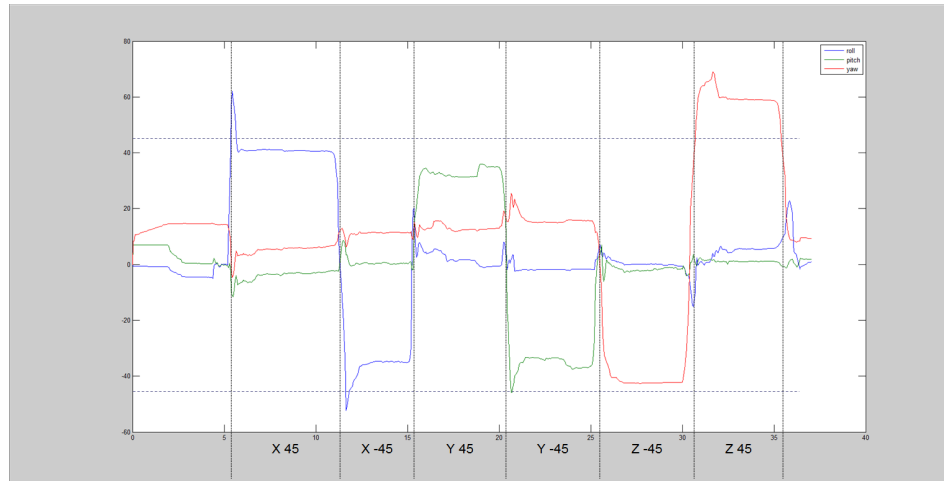


Figure 6 – Static Test: Angle Measurement

The dynamic test ensures the ability of the simulation platform to perform a dynamic state. First examination is done to check the ability of the reaction wheels in rotating the attitude. The given input is a step input, 15% of the power, clockwise and counter-clockwise, and applied for 5 seconds each. The result can be viewed below:

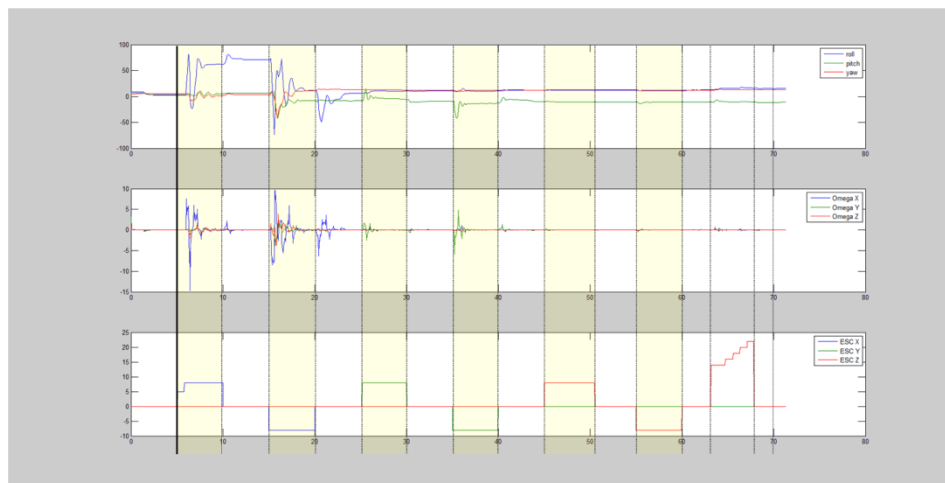


Figure 7 – Dynamic Test: Reaction Wheels

It can be observed that the reaction wheels are functioning, yet it still has a quite number of flaws. At first, when the reaction wheel in the X axis is driven, the dynamic reaction of the model is quite good. It even rotated for more than one rotation, and it also applied for the counter-clockwise state.

The condition is different when the reaction wheels in the Y axis and Z axis are driven. The driving motor in the Y axis only affect the model a bit, this is happened because of the inadequate torque that is produced by the actuator. Yet, it still showed a quite good performance, at least better than the performance when the reaction wheel in the Z axis is driven. This task only showed us a small reaction. This is because of some issues that will be described in the next sub chapter.

To check the performance of the platform, the gyroscopic phenomenon has to be proven. This can be achieved by giving a specific angular velocity to the reaction wheels in the Z axis, and give an external torque in Y or X axis by using torque produced by the reaction wheel in that particular axis. The correct phenomenon will create a reaction in the other axis.

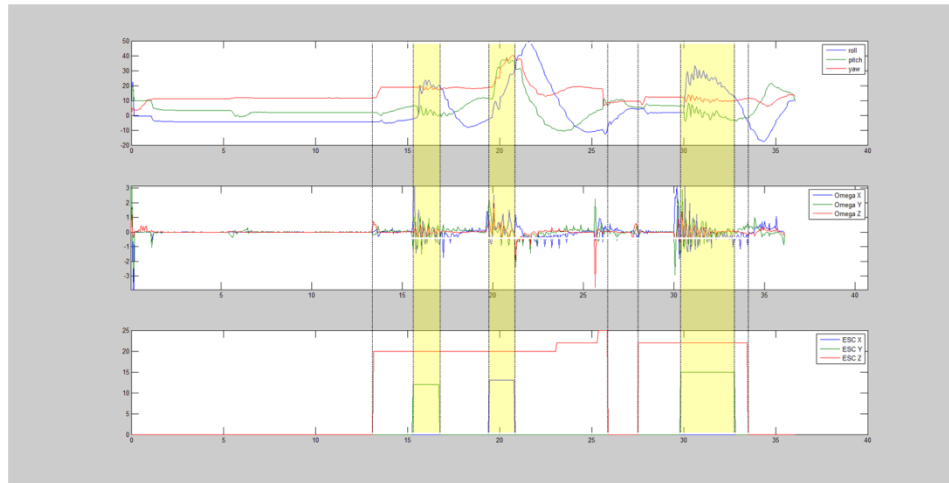


Figure 8 – Dynamic Test: Gyroscopic Phenomenon with Reaction Wheels

It can be observed that when the model is disturbed in the Y axis, the angle in the X axis changed, and this change is oscillating with the angle change of Y axis. It also occurred when the axis that is disturbed is the X axis. Therefore, it can be concluded that the simulation platform is able to show the gyroscopic phenomenon.

4.3 Platform Evaluation

The gyroscopic mechanism, manual manufactured platform, and the selected components number of disadvantages. From the platform, the problem that are encountered are:

- Gyroscopic mechanism has a possibility of gimbal lock phenomenon, where the mechanism loses an axis because there are two axes placed in the same rotating axis. This can make an error in the calculation. Usually this can happen when the first and the second axis of the order are coincided.
- Gyroscopic mechanism is also sensitive to the effect that is caused by gravity gradient, especially gyroscopic mechanism that is cube shaped. Even when the object is balanced perfectly, the difference in the distance to the ground between a side and the other one will make the mechanism to fall and cannot pretend its attitude.
- The bearings are used to reduce the friction that happens on every axis. In this platform, there are contacts between inertia platform frames and bearing slots that increase the friction on it.
- The main structures are drilled manually, and they have quite numbers of inaccuracy. This condition leads to the imperfect attachment of the walls and affects the position of the reaction wheels that will change the dynamic performance of the simulation platform.
- The component configuration – its position referred to body axes – affects the weight distribution of the cubesat model. This condition has to be managed by balancing the cubesat model perfectly. Nevertheless, this task is quite difficult because of the cube shape. The most ideal shape to achieve the perfect weight distribution to obtain the condition where the center of gravity is aligned perfectly to the rotation axis is by using a ball shape. This unbalanced cubesat model affects its dynamic performance significantly; making it is unable to perform a closed loop simulation.
- The wheels are made by using lathe machine manually, and quite inaccurate. This leads to vibrating wheels when it is driven by the motor, and this vibration affects the

performance of the sensor in gathering data, especially the accelerometer because it is sensitive to external forces that are applied to the object.

- The Arduino Pro Mini that acts as the OBC/OBDH is actually has a limited function and quite poor performances in processing the data. It also has limited port for other items. But, it is small and quite cheap, and in the case of simulation, it is able to do its tasks quite well.
- The motor is actually has good performance as it can spin with the angular velocity of up to 14,000 rpm, and it also is able to produce enough torque to rotate the heavy wheel.
- The ESC is one of the issues that is occurred in making this platform. In this case, the ESC that is used is Hobbyking 10A with reversible. It is chosen because of its reversible function, but it is actually underpowered to drive the motor. And, the default firmware that is used is quite old-fashioned, and has problem in supplying power to the motor in right order in high RPM. The motor will stutter when it is given a high PWM input, so it cannot be used to its maximum potential.

5. Conclusion and Future Works

To conclude, the simulation platform that is intended to simulate the dynamic performance of a cubesat and to observe the capability of ADCS that is being used has successfully been developed at early stage. This simulation platform consists of cubesat model with simplified OBDH/C, ADCS, and actuators; Inertia platform, and system that are provided to achieve the HILS system.

From this first stage, there are quite many developments that can be made. First, the improvement of mechanic parts such as weight balance can be resolved; friction at the bearing and vibration from brushless motor can be eliminated to achieve better performance. Besides, the electrical components that are used can be replaced by more proper and better components such as ESC with better firmware to support two directions rotation, better brushless motor, and more complete sensor.

In addition, the control system for the cubesat platform can be developed to simulate attitude control of the model. The reaction wheel configuration can also be altered to be tetrahedral arrangement, or other adjustment. The actuator can be replaced with actual actuator such as actuator from GOLIAT nano satellite, as well as changed it with other kind of actuator such as magnetorquer and CMG. Lastly, a system to simulate perturbation in the outer space can also be added.

References

- ¹ Hutasuhut, H. M., "Early Development of Spacecraft Dynamic Simulator", Undergraduate Theses, Aeronautics and Astronautics Dept, Institut Teknologi Bandung, Indonesia, 2011.
- ² Hadi, Ryan. F., and Sasongko, R. A., "Cubesat Dynamic Design and Visualization Tools in MatLab Environment", International Seminar on Aerospace Science and Technology 2014, Lembaga Penerbangan dan Antariksa, Tangerang (submitted for publication).
- ³ Meissner, David M. A Three Degrees of Freedom Test-Bed for Nanosatellite and CubeSat Attitude Dynamics, Determination, and Control. Thesis. Naval Postgraduate School, 2009. N.p.: n.p., n.d. Print.
- ⁴ Hadi, Ryan. F., and Sasongko, R. A., "Dynamic Simulation Platform Design for Cubesat with Reaction Wheel Control System", AIAA Reg VII-AU Student Conference 2014, AIAA (submitted for competition).