Attitude Determination and Control for CubeSats

Ahmed Mohammed El-sayed Yousef , Mohammed Khaled Shahat
MTC, Egypt, ahmedyousef44262@gmail.com, m.khaled@gmail.com
Supervisors: Dr. Mohamed S. Korany, Dr. Mohammed Abozied, Dr. Hossam Hendy, Prof. Dr. Yehia Elhalwagy
MTC, Egypt, engmsk45@mtc.edu.eg, mohammed.abozied@mtc.edu.eg, h.hendy@mtc.edu.eg, y.elhalwagy@mtc.edu.eg

Abstract*—Recently, CubeSats have been becoming economically attractive in many applications. As they provide cheap facilities to validate new ideas using the advanced technologies and payloads. Reaction wheels (RWs) are an effective solution and rely on the simple principle of conservation of angular momentum. The objective of this project is to develop an attitude control system (ACS) to achieve a predesiged mission for a Cubesat graduation project using a low-cost and reliable reaction wheel, the mission is called MTCsat. A two axis balanced sticks (inverted pendulum) actuator is desired to implement the attitude determination and control subsystems (ADCS). The mathematical model of the inverted pendulum reaction wheel is obtained using the sensor measurements. The attitude determination is calculated using quaternion algorithm and implemented using PID controller by linearizing the system model. In this project, the proposed control approach is validated by a hardware implementation using a reaction wheel actuator with a suitable encoder.

***Index Terms*—Cubesat Attitude Control, Inverted Pendulum Reaction wheel**

I. Introduction

In recent years, CubeSats have grown significantly in popularity by researchers and scientists in many civilian applications, e.g., surveillance, weather forecast, meteorological monitoring, utilizing the advanced technologies and payloads [1]. CubeSats provide cheap and ease of launch facilities to validate new ideas in the space industries considering their short development cycle comparing to bigger satellites [2]. Moreover, commercial electronics components can be used in CubeSats design, as we can ignore are radiation in Low Earth Orbits (LEO). To achieve the specified missions, the typical Cubesat consists of necessary systems, e.g., electrical power, telemetry, communication, and attitude determination and control system (ADCS).

A. Attitude Determination and Control System

The main function of ADCS is attitude stabilization and attitude maneuver which achieve the required maneuver/re-orientation in space. Practically, ADCS processes the sensors data to determine the spacecraft attitude to activate the specified actuators using a properly control technique.

Generally, ADCS consists attitude determination subsystem (ADS) and attitude control subsystem (ACS). ADS is the calculation process of the spacecraft orientation vector relative either in the inertial reference or body frames, typically in the form of the Euler angles or Quaternion. ADS consists sensors, algorithms, and mathematical models The main function of ACS is to orientate the spacecraft with sufficient accuracy in the desired attitude, as in [3]. In the space, we need to control the spacecraft attitude control to point specified payloads, e.g., communications antennae, navigational instruments, and/or solar panels [2].

The block diagram of the attitude control system is represented in Fig 1.

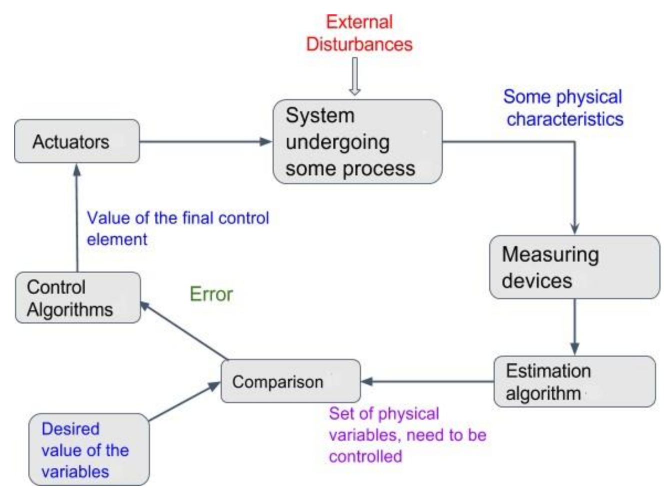


Fig. 1. ADC block diagram which illustrate the interference between the system components with specified algorithms to resist any disturbances andreach the desired setpoint.

In the literature, many researchers on Cubsat control design pays particular attention to use proportional-integral- derivative (PID) controllers [4], [5] and linear quadratic regu- lator (LQR) [5], [6]. LQR is based on solving a mathematical optimization problem considering the system dynamic and objective function. Generally, PID controller is the simplest and the most widely used controller because it is easy to design and it is more resistive to the system uncertainties.

Typically, ACS consists actuators and attitude controller.

B. Reaction Wheel Actuators

The attitude actuators, which may be passive or active, generate the required torque to adjust the attitude. Practically, active actuators are faster and more accurate than passive ones but with more power consumption.

In the literature, there are many active actuators, e.g., thrusters, reaction wheels, and electromagnets. Practically, the thruster accuracy depends on the specified minimum impulse. While moment gyros are used for larger more agile satellites. However, electromagnets coils use the earth's magnetic field to generate very small torque [2], [7]. An attitude control system [3] uses an inverted pendulum to generate a sufficient torque to overcome any external dis- turbance. Otherwise, the reaction wheel has high inertia which generates a suitable torque on the spacecraft to change the angular velocity in desired axis, e.g., in [7]. The reaction wheel has very higher accuracy compared with other actuators, however they have larger mass and so more power consumption.

C. Project Contributions and Layout

The main contributions of this project are:

1) mathematical model of the system dynamics
2) designing a suitable controller
3) hardware implementation

The paper is organized as follows. Section II outlines the problem formulation including the system nonlinear mod- elling and linearization. Section III presents the underlying PID controller. Then, the analysis of the simulation results is discussed in Section IV. Finally, the conclusion is presented.

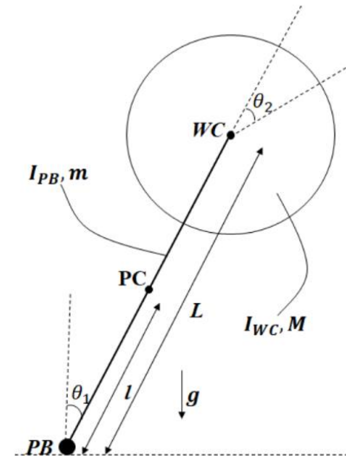


Fig. 2. Free Body Diagram of the Reaction Wheel Inverted Pendulum [8] which clarify the needed parameter for the Mathematical Model.

II. MATHEMATICAL MODELING OF ONE AXIS INVERTED PENDULUM

To design the suitable controller, we need to develop the mathematical model of the system dynamics. The governing equations for the satellite's attitude are expressed using angular kinetic and angular kinematics equations. Angular kinetic equations express the rate of change in angular velocities due to external torques and disturbances and The angular kinematics equations specify the relationship between absolute angular velocity of the satellite and its orientation in the space. We use the Lagrangian approach [9] to derive the nonlinear model which can be linearized. Because Lagrangian method is simpler than vector forces and accelerations and deals with scalar energy functions.

The reaction wheel pendulum has two degrees of freedom as shown in Fig 2 where θ_1 is an angle between the pendulum and a fixed point, and θ_2 is an angle between pendulum line and arbitrary axis on reaction wheel. So $(\dot{\theta}_1, \dot{\theta}_2)$ are angular velocities of the pendulum and reaction wheel. θ_1 must be controlled to be equal zero.

At first, the potential and kinetic energy equations are calculated for the pendulum and reaction wheel as follows:

The potential energy for both are

$$V_p = \text{mlg} \cos(\theta_1) \quad (1)$$

$$V_w = MLg \cos(\theta_1) \quad (2)$$

The translational and rotational kinetic energy of the pendulum are

$$T_{PT} = \frac{1}{2}(ml^2)\dot{\theta}_1^2 \quad (3)$$

$$T_{PR} = \frac{1}{2} I_{PC} \dot{\theta}_1^2 \quad (4)$$

The translational and rotational kinetic energy of the reaction wheel are

$$T_{WT} = \frac{1}{2}(ML^2)\dot{\theta}_1^2 \quad (5)$$

$$T_{WR} = \frac{1}{2} I_{WC}\left(\dot{\theta}_1 + \dot{\theta}_2\right)^2 \qquad (6)$$

So the total potential and kinetic energy for the pendulum and reaction wheel are

$$V = mlg\cos(\theta_1) + V_w + MLg\cos(\theta_1) \qquad (7)$$

$$T = \frac{1}{2} I_{WC}\left(\dot{\theta}_1 + \dot{\theta}_2\right)^2 + \frac{1}{2}(ML^2)\dot{\theta}_1^2 + \frac{1}{2} I_{PB}\dot{\theta}_2^2 \qquad (8)$$

where $I_{PB} = (I_{PC} + ml^2)$. The Lagrangian function is

$$L = T - V \qquad (9)$$

The Lagrangian method is expressed as

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_k}\right) - \frac{\partial L}{\partial_k} = 0, \qquad k = 1, \ldots, n \qquad (10)$$

Using Eq.(9,10), the system can be finally modeled as

$$(ML^2 + I_{PB})\ddot{\theta}_1 + I_{WC}\ddot{\theta}_1 + I_{WC}\ddot{\theta}_2 - (ml + ML)g\sin(\theta_1) = 0 \qquad (11)$$

$$I_{WC}\ddot{\theta}_1 + I_{WC}\ddot{\theta}_2 - 0 = \tau \qquad (12)$$

where τ is the applied torque. Because the maximum value of θ_1 equal $\pm$ 5 degrees the system model can be linearized as

$$\ddot{\theta}_1 = \frac{(ml+ML)g\theta_1 - \tau}{(ML^2+I_{PB})} \qquad (13)$$

$$\ddot{\theta}_2 = \frac{(ml+ML)g\theta_1}{(ML^2+I_{PB})} + \frac{(ML^2+I_{PB}+I_{WC})\tau}{(ML^2+I_{WC})} \qquad (14)$$

As the applied torque τ is generated by a motor, so the electric motor must be modeled as shown in Fig 4. The torque is expressed as

$$\tau = K_t i \qquad (15)$$

and the back electro motive force (emf) is defined as

$$EMF = K_e\dot{\theta}_2 \qquad (16)$$

Where K_t, K_e are the torque and back emf constants, respectively. Assume $K_t = K_e = K$ and using the Newton second law and Kirchoff voltage law, the motor model is expressed as:

$$\ddot{\theta}_2 + b\dot{\theta}_2 = Ki \qquad (17)$$

$$L\frac{di}{dt} + Ri = V - K\dot{\theta}_2 \qquad (18)$$

I. IMPLEMENTATION AND SIMULATION

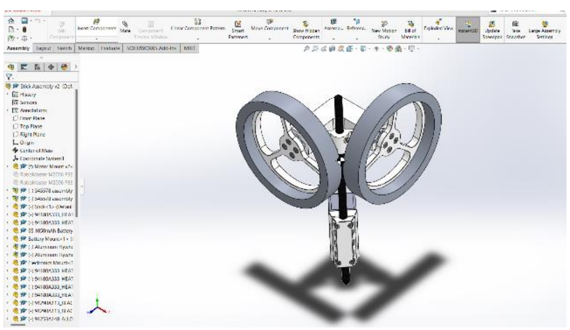


Fig. 3. the designed inverted pendulum by using SOLIDWARK softwarefor simulation before manufacturing the mechanical components.

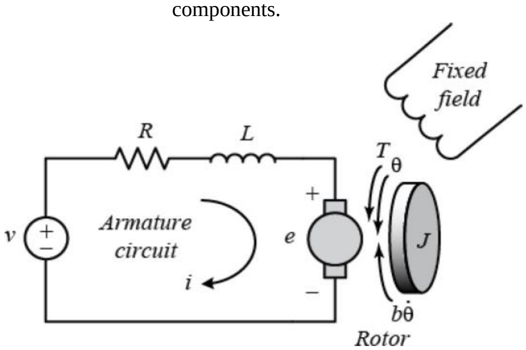
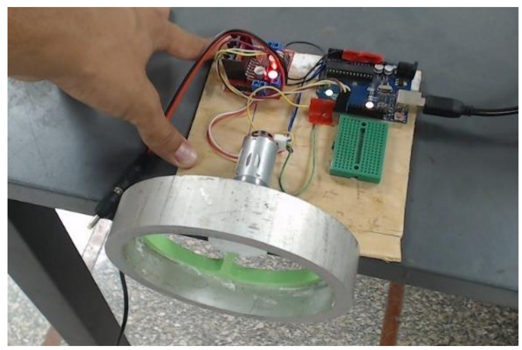


Fig. 4. The free body diagram of the electric motor which clarify theneeded parameter for the Mathematical Model.

A. Building the System Structure in SOLIDWARK Software

We used SOLIDWARK Software (Fig 3) to build the system structure to obtain the system



parameters as in Table I.

TABLE I
DESIGN CONSTANTS

L	**172.3 mm**	**M**	**0.378 kg**
l	**222.3 mm**	**m**	**1.014 kg**
I_{WC}	**0.0112 $kg.m^2$**	I_{PC}	**0.024 $kg.m^2$**

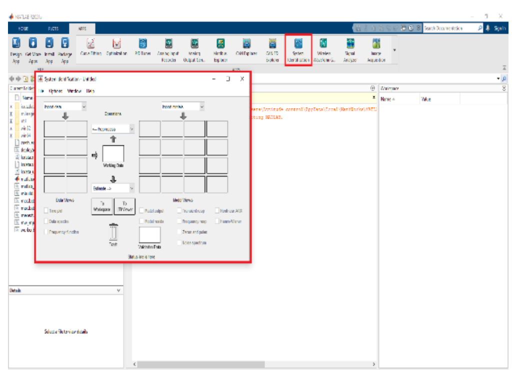


Fig. 5. system identification tool in MATLAB software to estimate the motor transfer function based on the collected input and output data for a different time intervals.

Fig. 6. motor connection with Arduino to collect input and output data for a different time intervals by using Arduino code

B. Motor System Identification

For more accurate modeling, the system identification tool in MATLAB is used to determine the transfer function of the motor, see Fig 5. We wrote a Arduino code to collect the input and output data from an magnetic shaft encoder connected to the motor as shown in Fig 6 over different time intervals to obtain more accurate transfer function by system identification tool in MATLAB software to be applied on the overall system model.

Finally, the system model is implemented in SIMULINK as shown in Fig 7 to design a suitable controller.

C. Control System Design

In this work, we design PID controller to stabilize the overall system using MATLAB/SIMULINK.

The general form of PID controller is given as follows

$$u(t) = K_P e(t) + K_I \int_0^t e(t)dt + K_D \frac{de(t)}{dt} \quad (19a)$$

$$e(t) = y_r(t) - y_m(t) \quad (19b)$$

The error signal $e(t)$ is the difference between the reference y_r and the measured output y_m. This error is used to generate the proportional, integral, and derivative actions to determine the control signal $u(t)$ applied to the system. The aim is to minimize the error, i.e., the difference between the reference and the system output, that achieved by tuning the PID gains.

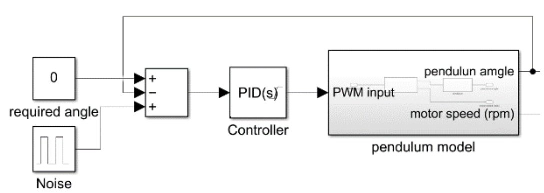


Fig. 7. overall system block diagram on MATLAB/SIMULINK to simulate and design the suitable controller.

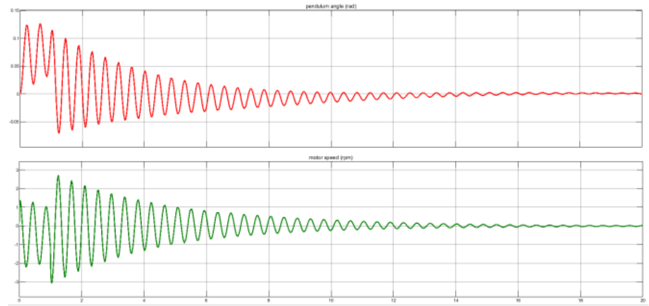


Fig. 8. simulated system output, the upper curve clarify the pendulum angle response over time to reach the zero angle ,and the lower curve clarify the changing in motor speed to stabilize the pendulum and reach the zero angle.

. The proportional term K_P has the main effect in changing the output value. The integral term K_I sums up all the past error value and controls the speed by which the system steady state error reduced. The derivative term K_D estimates the controller future response depending on the error rate over time, this leads to enhance the stability andobtain a smooth response. Thus, all three coefficients needto be tuned to force the system output to track the desiredstate, see Table II.

TABLE II
THE SELECTED PID CONTROLLER GAIN

Gain	K_p	K_i	K_d
value	9.51826	0.60438	15.77828

D. Simulation Results

Fig 8 illustrates the system output which are the pendulum angle and motor speed.

II. HARDWARE IMPLEMENTATION

In this work, we implement the proposed control approachto verify the simulation results. The implementation process consits attitude determination and attitude control.

A. Attitude Determination

In this work, we use an integrated 6-axis motionsensor module ”MPU6050” controlled by Arduino controller (Fig 10), which consists a 3-axis Gyroscope, 3- axis Accelerometer, and digital motion processor. Because “MPU6050” sensor is a small size and applicable for any microcontroller.

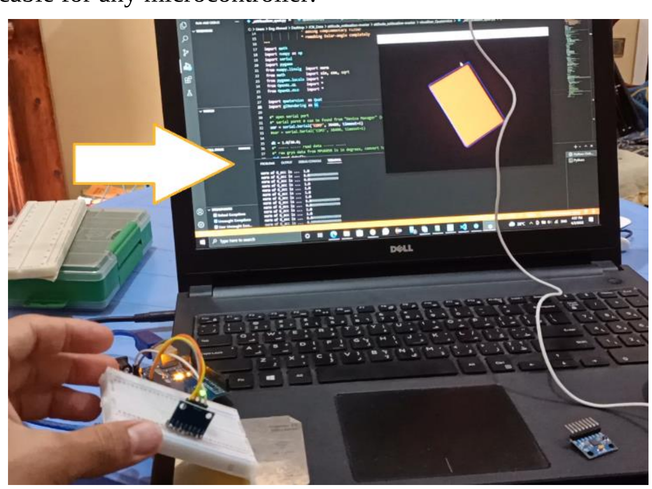


Fig. 9. monitor the attitude from ”MPU6050” and arduino on PC by using Python code.

In this work, we fuse the output data from the gyroscope and the accelerometer by using the “quaternion calculation” [10]. Assume $q = [q_0, q_1, q_2, q_3]^\top$ and $p = [p_0, p_1, p_2, p_3]^\top$, so, quaternion calculation is represented as

$$q * p = \begin{bmatrix} q_o * p_o - \left(q_1 * p_1 + q_2 * p_2 + q_3 * p_3\right) \\ q_o * p_1 + q_1 * p_o + \left(q_2 * p_3 - q_3 * p_2\right) \\ q_o * p_2 + q_2 * p_o + \left(q_3 * p_1 - q_1 * p_3\right) \\ q_o * p_3 + q_3 * p_o + \left(q_1 * p_2 - q_2 * p_1\right) \end{bmatrix} \quad (20)$$

As quaternions have no obvious physical interpretation so they can’t be visualized. Then conversion from quaternion to Euler angles is done to visualize the needed rotation angles by:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \frac{2arccor\left(q_o\right)}{\sqrt{1-q_o^2}} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

These conversion equation is implemented in *MPU6050* using the *Arduino* code as shown in Fig 10.

B. Attitude Control System Implementation

In this project, we see a reaction wheel, see Fig 11, which is an electric motor-driven rotor with the following advantages:

- Relatively light with high inertia.
- Low require power.
- Higher torque.
- More controllable.

It generates a suitable torque to control the angular velocity in desired axis to creates an equal and opposite amount of momentum that generated from disturbances.

To rotate this reaction wheel, we use ”JGA25-370-21.3k” motor with a suitable encoder, see Fig 11 because of its suitable torque.

As mention above, we implement an PID code in *Arduino* to control the inverted pendulum using a feedback signal ”motor speed” from the *encoder* and the *MPU6050*, Fig (12) shows the overall implemented system including mechanical structure and electric components.

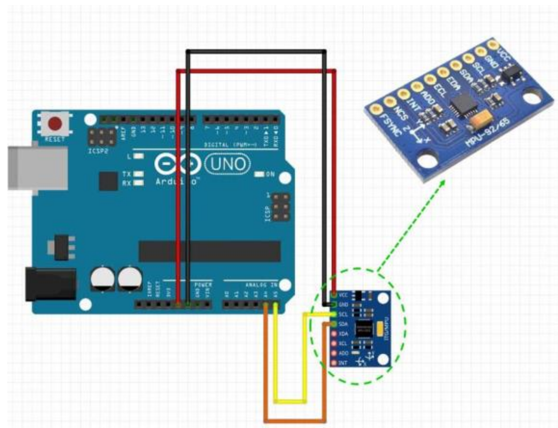


Fig. 10. Arduino connection with MPU6050 IMU sensor to provide attitude determination by Arduino code based on Quaternion algorithm.

Fig. 11. The manufactured reaction wheel based on the simulation outputs.

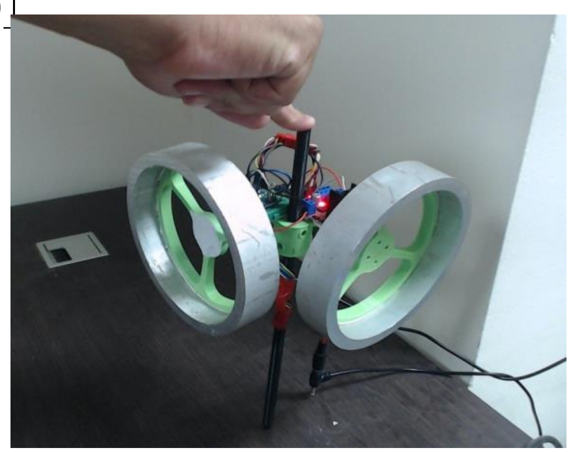


Fig. 12. final shape for the project

III. CONCLUSION

The objective of this project is to develop an attitudecontrol system for a Cubesat using a low-cost and reliable reaction wheel. A two axis balanced sticks (inverted pendulum) actuator is designed to implement the attitude determination and control subsystems. The mathematical model of the inverted pendulum is obtained. The attitude determination is calculated using Quaternion. The attitude control system is implemented using PID controller and by linearizing the system model. In this project, the proposed control approach is validated by a hardware implementation using a reaction wheel actuator with a suitable encoder.

The electronics design of the reaction wheels is able to achieve the required RPM values by employing a PID controller, thus using a minimum time to achieve the desiredspeed. The structural design of the reaction wheel system is done considering the physical constraints as well as the mass budget by using aluminum.

IV. FUTURE WORK

Possible extensions for attitude determination can be obtained from gyro and corrected using accelerometer data by integrating MPU6050 & GPS module using kalman filter to achieve more accurate data and getting cube sat full navigation information.

Further extension of attitude control can achieve more smooth and accurate control using kalman Filter. Moreover, the attitude control code can be written in STM board instead of Arduino board for a higher performance processing speed. Finally, we can design a mechanical structure for a real *3U cube sat* and applying the attitude determination and control section based on the structure model.

REFERENCES

[1] "Cubesat missions and applications," in *Cubesat Handbook* (C. Cap- pelletti, S. Battistini, and B. K. Malphrus, eds.), pp. 53–65, Academic Press, 2021.

[2] F. W. Long, *Design and Testing of a Nanosatellite Simulator Reaction Wheel Attitude Control System*. PhD thesis, Master's thesis, Utah State University, 2014.

[3] P. Fortescue, G. Swinerd, and J. Stark, *Spacecraft systems engineering*.John Wiley & Sons, 2011.

[4] M. A. Samuels, *The design and testing of a three-degree-of-freedom small satellite simulator using a linear controller with feedback linearization and trajectory generation*. PhD thesis, Master's thesis, Utah State University, 2014.

[5] C. W. Crowell, *Development and Analysis of a Small Satellite Attitude Determination and Control System Testbed*. PhD thesis, Master's thesis, Massachusetts Institute of Technology, 2011.

[6] W. Kang and H.-H. Yeh, "Co-ordinated attitude control of multi- satellite systems," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 12, no. 2-3, pp. 185–205, 2002.

[7] S. Janson, B. Hardy, A. Chin, D. Rumsey, D. Ehrlich, and D. Hinkley, "Attitude control on the pico satellite solar cell testbed-2," 2012.

[8] P. Brevik, "Two-axis reaction wheel inverted pendulum," Master's thesis, NTNU, 2017.

[9] Y. Yuan, "Global solutions to special lagrangian equations," *Proceed-ings of the American Mathematical Society*, pp. 1355–1358, 2006.

[10] Y. Yang, "Spacecraft attitude determination and control: Quaternion based method," *Annual Reviews in Control*, vol. 36, no. 2, pp. 198– 219, 2012.