

DESIGN AND CONTROL OF A TETRAHEDRON ROBOT USING REACTION WHEELS

by

Thongtong Eamsaard

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Mechatronics and Machine Intelligence

Examination Committee: Prof. Manukid Parnichkun (Chairperson)
Dr. Mongkol Ekpanyapong
Dr. Tanujjal Bora

Nationality: Thai
Previous Degree: Bachelor of Engineering in Aerospace Engineering
Kasetsart University, Thailand
Scholarship Donor: Royal Thai Government Fellowship

Asian Institute of Technology
School of Engineering and Technology
Thailand
May 2024

AUTHOR'S DECLARATION

I, Thongtong Eamsaard, declare that the research work carried out for this thesis was in accordance with the regulations of the Asian Institute of Technology. The work presented in it is my own and has been generated by me as the result of my own original research, and if external sources were used, such sources have been cited. It is original and has not been submitted to any other institution to obtain another degree or qualification. This is a true copy of the thesis, including final revisions.

Date: 20 April 5, 2024

Name: THONGTONG EAMSAARD

Signature:

ACKNOWLEDGMENTS

This thesis, “Design and Control of a Tetrahedron Robot Using Reaction Wheels”, would not have been possible without the invaluable support and guidance of many people.

I deeply appreciate my thesis advisor, Professor Manukid Parnichkun. His expertise in Mechatronics provided me with a strong foundation for this research. Professor Parnichkun's encouragement was instrumental in overcoming the challenges I faced throughout this project. I am also grateful for his support in securing the necessary equipment, supplies, and research methods to complete this thesis successfully.

I would like to thank my committee members, Dr. Tanujjal Bora and Dr. Mongkol Ekpanyapong, for their invaluable feedback and suggestions. Their critical review of the thesis helped me to significantly improve the accuracy and clarity of this work.

I am deeply grateful for my family's physical and emotional support, as well as their guidance and help during the thesis process. Their understanding was a constant force and source of strength and motivation to me.

I am also thankful to my ISE seniors for providing starter material for doing this thesis. They give advice and help with explaining language programming and general good design for the robot. These were crucial in setting the direction for this project.

Lastly, I would not be able to apply and study at the Asian Institute of Technology (AIT) without the Royal Thai Government providing me with a scholarship to attend. This has allowed me to study in an eco-friendly environment and interact with many amazing people at AIT.

ABSTRACT

In this thesis, the balancing control of a Tetrahedron robot using reaction wheels is investigated. Unlike previous approaches that similar configurations could only manage roll and pitch balancing simultaneously, this study proposes a Tetrahedron robot. This design enables the reaction wheels to contribute torque to all three principal axes of roll, pitch, and yaw concurrently. This approach offers advantages in terms of compactness and eliminates the requirement for perfect alignment with the principal axes.

The primary objective is to achieve stable balancing of the Tetrahedron robot in all three directions with this reaction wheels arrangement. To this end, a control algorithm to balance the robot is based on the Linear Quadratic Regulator. This controller can prioritize angle and angular velocity states of the robot, facilitating efficient control in the presence of external disturbances. Additionally, counters within the microcontroller are employed to minimize processing time and guarantee consistent cycle time for motor control adjustments.

The results demonstrate successful balancing of the tetrahedron robot in roll, pitch, and yaw directions, representing a significant advancement over previous configurations. The robot exhibits rapid settling times of 2.5 and 2.62 seconds for roll and pitch disturbances with a maximum error of 6.4 and 6 degrees, respectively. Yaw control achieves satisfactory heading maintenance with maximum disturbance of 10 degrees and a 7.8-second settling time. This work contributes a tetrahedron robot design with a compact reaction wheel arrangement and an LQR-based control algorithm. This design allows the robot to achieve effective balancing performance in all three principal axes, including yaw, which was previously unachievable with similar configurations.

CONTENTS

| | Page |
|---|-------------|
| AUTHOR'S DECLARATION | ii |
| ACKNOWLEDGMENTS | iii |
| ABSTRACT | iv |
| CONTENTS | v |
| LIST OF TABLES | vii |
| LIST OF FIGURES | viii |
| LIST OF ABBREVIATIONS | x |
| CHAPTER 1 INTRODUCTION | 1 |
| 1.1 Background of the Study | 1 |
| 1.2 Statement of the Problem | 2 |
| 1.3 Research Question | 3 |
| 1.4 Objectives of the Study | 3 |
| 1.5 Scope | 3 |
| 1.6 Contribution | 3 |
| CHAPTER 2 LITERATURE REVIEW | 5 |
| 2.1 Upright Pendulum Systems | 5 |
| 2.2 Reaction Wheels Inverted Pendulum | 8 |
| 2.3 3D Reaction Wheels Inverted Pendulum Testbeds | 11 |
| 2.4 Tetrahedron Geometry | 15 |
| 2.5 Chapter Summary | 15 |
| CHAPTER 3 METHODOLOGY | 16 |
| 3.1 Timeline | 16 |
| 3.2 Conceptual Design | 16 |
| 3.3 Mechanical Prototype Design | 17 |
| 3.4 Electrical Design | 20 |
| 3.5 Dynamic Model | 21 |
| 3.5.1 Coordinates Representation | 22 |
| 3.5.2 Generalized Force | 25 |
| 3.5.3 Kinetic Energy | 27 |
| 3.5.4 Potential Energy | 28 |

| | |
|---|-----------|
| 3.5.5 Lagrange Equations | 28 |
| 3.5.6 State-space Representation | 30 |
| 3.5.7 Linearized State-space | 31 |
| 3.6 Linear Quadratic Regulator | 33 |
| 3.7 Control Simulation | 33 |
| 3.8 Control Experiment | 34 |
| CHAPTER 4 RESULT AND DISCUSSION | 36 |
| 4.1 Control Gain Design | 36 |
| 4.1.1 Moment of Inertia Constant Estimation | 36 |
| 4.2 Simulation | 39 |
| 4.3 Experiment | 42 |
| 4.3.1 Using Simulation Gain | 42 |
| 4.3.2 Adjusted Control Gain in the Actual Robot | 43 |
| 4.4 Maximum Disturbances Angle | 45 |
| 4.4.1 Maximum Disturbance on Roll Direction | 45 |
| 4.4.2 Maximum Disturbance on Pitch Direction | 46 |
| 4.4.3 Maximum Disturbance on Yaw Direction | 48 |
| CHAPTER 5 CONCLUSIONS | 50 |
| 5.1 Conclusion | 50 |
| 5.1.1 New Tetrahedron Robot Using Three Reaction Wheels | 50 |
| 5.1.2 Control Algorithm Fitted with Tetrahedron Robot | 50 |
| 5.2 Recommendations | 51 |
| REFERENCES | 52 |

LIST OF TABLES

| Tables | Page |
|--|-------------|
| Table 3.1 Gantt Chart Task Schedule for the Master's Thesis | 16 |
| Table 3.2 States, Inputs, and Outputs of the System | 31 |
| Table 4.1 Inertia Constants | 38 |

LIST OF FIGURES

| Figures | Page |
|--|-------------|
| Figure 2.1 Inverted Pendulum Mechanism: Cart-pole and Reaction Wheel | 5 |
| Figure 2.2 Double Inverted Pendulum Cart-pole (Graichen et al., 2007b) | 6 |
| Figure 2.3 Inverted Pendulum in Rotary Motion | 7 |
| Figure 2.4 SMC Demonstration on Double Inverted Pendulum | 7 |
| Figure 2.5 Rotary Double Inverted Pendulum System | 8 |
| Figure 2.6 Schematics and Device of Reaction Wheel Inverted Pendulum System | 8 |
| Figure 2.7 Standard Orthogonal 3-wheel Configuration | 9 |
| Figure 2.8 Tetrahedron Configuration of Four Reaction Wheels | 9 |
| Figure 2.9 Different Geometry of Reaction Wheel | 10 |
| Figure 2.10 Model of a Cube in 2D | 11 |
| Figure 2.11 Model of the Three-Wheel Stick Robot | 11 |
| Figure 2.12 The Balancing Cube, Balancing Using Accelerometer | 12 |
| Figure 2.13 The Cubli Balances on Corner | 13 |
| Figure 2.14 Another Version of Cubli | 13 |
| Figure 2.15 The Inverted Cube with 3 Reaction Wheels Installed | 14 |
| Figure 2.16 The One-Wheel Cubli | 15 |
| Figure 2.17 Tetrahedron Properties | 15 |
| Figure 3.1 The Conceptual CAD of Tetrahedron Robot | 17 |
| Figure 3.2 Reaction Wheels Orientation Inside The Conceptual Robot | 17 |
| Figure 3.3 The Prototype Design in CAD | 18 |
| Figure 3.4 Reaction Wheels Orientation in Prototype Robot | 18 |
| Figure 3.5 Checking Components Fit Within the Structure | 19 |
| Figure 3.6 The Actual Robot Prototype Structure | 19 |
| Figure 3.7 Checking the Bending Degree of Structure of Top and Bottom Sheet | 20 |
| Figure 3.8 Simplified Circuit Diagram | 21 |
| Figure 3.9 The Tetrahedron Robot | 21 |
| Figure 3.10 Three Coordinates System | 22 |
| Figure 3.11 Motor to Frame Coordinates Schematic Diagram | 23 |
| Figure 3.12 Frame to Global Coordinates Schematic Diagram | 24 |

| | |
|--|----|
| Figure 3.13 DC Motor Schematic | 25 |
| Figure 3.14 Wheel Angular Acceleration Direction | 27 |
| Figure 3.15 Potential Energy Schematics | 28 |
| Figure 3.16 Simulation Schematics | 34 |
| Figure 3.17 Simulink Model of the Tetrahedron Robot | 34 |
| Figure 3.18 Manual Hand Orientation | 35 |
| Figure 4.1 Weighting the Actual Parts | 36 |
| Figure 4.2 Get the Robot Moment of Inertia at Reference Point | 37 |
| Figure 4.3 Get the Reaction Wheel Moment of Inertia | 37 |
| Figure 4.4 Simulation Results Using Initial Control Gain | 40 |
| Figure 4.5 Simulation Results Using Good Simulation Gain | 41 |
| Figure 4.6 Roll Displacement Angle of Gain from Simulation | 42 |
| Figure 4.7 Pitch Displacement Angle of Gain from Simulation | 42 |
| Figure 4.8 Yaw Displacement Angle of Gain from Simulation | 43 |
| Figure 4.9 Roll Displacement Angle of Good Actual Gain | 44 |
| Figure 4.10 Roll Displacement Angle of Good Actual Gain | 44 |
| Figure 4.11 Roll Displacement Angle of Good Actual Gain | 45 |
| Figure 4.12 Disturb 5-degree on Roll Direction | 45 |
| Figure 4.13 Disturb 10-degree Overshoot on Roll Direction | 46 |
| Figure 4.14 Max Disturbance of 6 Degree on Roll Direction | 46 |
| Figure 4.15 Disturb 5-degree on Pitch Direction | 47 |
| Figure 4.16 Disturb 10-degree Overshoot on Pitch Direction | 47 |
| Figure 4.17 Max Disturbance of 6 Degree on Pitch Direction | 48 |
| Figure 4.18 Disturb 5-degree on Yaw Direction | 48 |
| Figure 4.19 Max Disturbance of 10 Degree on Yaw Direction | 49 |
| Figure 4.20 Oscillation on Yaw Direction When Disturbed at 12 Degrees | 49 |

LIST OF ABBREVIATIONS

| | |
|-----|------------------------------------|
| IMU | = Inertial Measurement Unit |
| LQR | = Linear Quadratic Regulator |
| PID | = Proportional Integral Derivative |

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

Because inverted pendulum systems are unbalanced in an upward position, they are a well-known control theory challenge. Because of its instability, this system is frequently utilized as a basic reference point for novel control theories and notions. Since the system's introduction in 1908, numerous configurations such as the cart-pole pendulum, reaction wheel pendulum, and wheeled pendulum (Bobrow et al., 2021a) have been the subject of extensive research for years. In addition, the inverted pendulum dynamics represent many types of real-world problems, including satellite control, the launch of a rocket, a bicycle, the Segway, and so on.

Apart from the cart-pole inverted pendulums, that have linear motion to control cart, reaction wheel pendulum uses angular momentum from the rotating wheel to control the pendulum in upright position. This concept is used in satellites, where their attitude and stability are controlled by these reaction wheels. Each axis of the satellite is controlled by applying torque from the change in energy from the reaction wheel.

Many balancing robots use the concept of symmetry to initiate the idea on their 3D shape such as Sphere and Cube. These robots with balancing algorithms can do various tasks and act as playground to test new concepts of balancing. First to introduced 3D is Sphere because of its symmetry on every principal axis. Another 3D is introduced 2012, named Cubli, made by Swiss Federal Institute of Technology Zurich in Switzerland, showed its ability to balance on any edge and corner. Using three reaction wheels, this unique 3D inverted pendulum can achieve such tasks.

Tetrahedron, also known as a triangular pyramid, is the simplest polyhedron of all the others. This also is the only one that has fewer than 5 faces. Compared with the Cubli, the reaction wheels' rotational axis is in line with each of principal axis. However, the rotational axis of three reaction wheels on Tetrahedron Robot will be contributed to all three principal axes. Hence, combining the concepts from reaction wheels that Cubli used, The Focus of this work is to balancing the Tetrahedron Robot with three reaction

wheels attached as the new testbed, with all three of reaction wheels contributed to Roll, Pitch, and Yaw axis.

1.2 Statement of the Problem

Previously mentioned, balanced systems have broad applications in the domain of aerospace, new energy, machinery industries and so on. The inverted pendulum is the simplest system for balancing problems. Existing systems that use reaction wheels to balance, such as the Stick Robot, can only control roll and pitch direction. So, it is a challenge in the sense of controlling the balance on both roll, pitch, and yaw direction. In addition, these abilities to control all three axes can contribute to the development of more advanced systems.

A few different mechanisms have been proposed to control the balance on these axes. This demonstrates that this balancing problem can be balanced in a variety of mechanisms. However, previous research only based on the idea of using a single reaction wheel to control the balance on axis where the wheel is rotated on. Hence, a new mechanism which is better to balance the system using reaction wheels is presented to lower the degree of freedom of the system or the need for actuators in the future. The proposed mechanism will orient reaction wheels in a way that rotational axis is not directly on one of principal axes. Thus, Tetrahedron robot is introduced to this thesis as a new mechanism.

Furthermore, this is very complicated since each of reaction wheel's orientation had the coupling effect on the other two wheels. The dynamic model of tetrahedron robot will contain roll-pitch-yaw axes interaction all together from coupling effect along with their derivatives are associated with each other. Therefore, the control design will be far more challenging to control all three wheels all together.

A respectable level of physics and mathematics knowledge is required from the robot to operate the dynamic model connected to the three reaction wheels. Walking through deriving the model, it is possible to achieve the balance on unstable equilibrium of the robot at the desired target with the use of coupling effect of the system.

1.3 Research Question

How to Develop a New Mechanism that Balances the Robot in Which All Reaction Wheels are Coupled to Roll-pitch-yaw Axis and Each Wheel is not Perpendicular to the Others?

This new mechanism for tetrahedron robot was designed to be balanced using three reaction wheels and needed a control algorithm that fits with the newly proposed tetrahedron robot.

1. How to develop a new mechanism that balances the robot in which all reaction wheels are coupled to roll-pitch-yaw axis and each wheel is not perpendicular to the others?

1.4 Objectives of the Study

The intent of this thesis is to balance the tetrahedron robot. To achieve this main objective, the following sub-objectives are required.

1. To develop a new mechanism for tetrahedron robot using three reaction wheels balancing technique (three axes).
2. To design a control algorithm that fits with the newly proposed tetrahedron robot with reaction wheels.

1.5 Scope

The scope of this thesis is as follow:

- This new robot remains balancing with three axes within a tilted angle of 3 degree on roll, pitch, and yaw direction.
- The robot has a height less than or equal to 20 cm.
- The width of the widest part of the robot is 20 cm.
- An Initial inclined angle intents within range of 10 degrees.
- The weight of the robot is not over 5 kg.
- The robot has 3 DOFs (degrees of freedom), roll, pitch, and yaw.
- The robot is controlled by three reaction wheels.

1.6 Contribution

Previous research reviewed so far only pairs a reaction wheel to balance one of principal axis. The new mechanism design will arrange reaction wheels in a way that rotational axis is not directly on one of principal axes but contributes to all three-control direction

of the robot. A control algorithm that appropriates with the newly proposed mechanism will be designed to balance the robot. This robot can balance in three directions: roll pitch and yaw, when disturbed within limited range, the robot can return to original stable position in all three directions.

CHAPTER 2

LITERATURE REVIEW

Inverted Pendulum is always considered, among others, as the most fundamental benchmark in Control Theory and Robotics. One of ways to balance this unstable system is to use Reaction Wheels. As more concepts evolve, more reaction wheels are used to have higher controllability by increasing degrees of freedom. The inverted cube is an example as it uses three reaction wheels to balance on its corner. Using reaction wheels, balancing tetrahedron robot which its symmetry properties like cube is interested in this thesis. This section shows related works and reviews past studies that are necessary to this thesis.

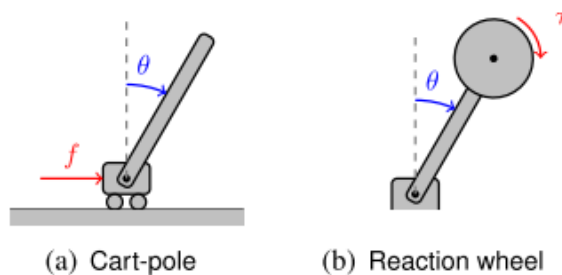
2.1 Upright Pendulum Systems

In contrast with a general pendulum in pendulum clock, the pivot point is under the center of the pendulum mass. Its concept has been a famous demonstration of using feedback control to stabilize the unstable systems. In the past, projects in the theme of robotics and mechatronics always selected the inverted pendulum as the test bench for validating their control algorithm.

Despite its simplicity look, an inverted pendulum system is a typical nonlinear dynamic system, in which it'll reach a stable equilibrium when pendulum is at pending position and an unstable equilibrium when pendulum is upright position (Bobrow et al., 2021a). Inverted pendulum generally has 2 types of balancing methods, one is cart-pole controlled by linear motion, and another is reaction wheel that controls the rotating wheel, exchanging angular momentum with the pendulum in Figure 2.1.

Figure 2.1

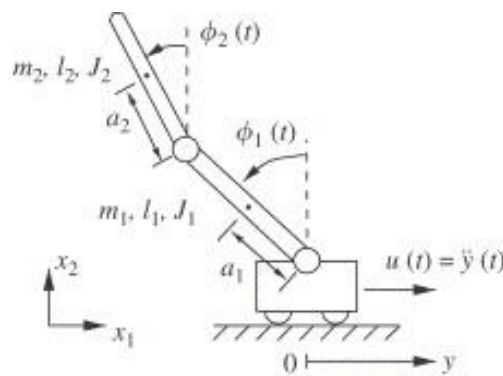
Inverted Pendulum Mechanism: Cart-pole and Reaction Wheel



There are more instances of inverted pendulum models which normally use these 2 types of mechanism. The double inverted pendulum on a cart uses a swing-up maneuver to make the pendulum up-right starting from resting state (Graichen et al., 2007a). While changing from resting state to up-right position, the problem is considered as a nonlinear feedforward control to make the pendulum at desired output position. To stabilize the swing-up movement, linear feedback control is used to accomplish the tasks in Figure 2.2. Experimental results were also performed. They were the proof of accuracy for the feedforward and feedback control algorithm.

Figure 2.2

Double Inverted Pendulum Cart-pole (Graichen et al., 2007b)



Control on rotary motion is also used to balance an inverted pendulum. Rotary Inverted Pendulum uses the rotation movement on the disk to balance an inverted pendulum attached at the rim of the disk (Akhtaruzzaman & Shafie, 2010). Linear Quadratic Regulator (LQR) shows significant results that it can balance on the unstable equilibrium point as shown in Figure 2.4. Another similar work uses an alternative control approach, Sliding Mode Controller (SMC), to balance double inverted pendulum (Chouhan, 2020). However, one of disadvantages using SMC is it creates chattering effect which can be dangerous for mechanical systems as shown in Figure 2.4. While the outcome of SMC is effective in simulation, it is not practical for complicated control schemes. Another variant of SMC, Higher Order SMC and Quasi-SMC, were also implemented and comparison between them shown that effects of chattering still persist bet in acceptable range for more advanced SMC.

Figure 2.3

Inverted Pendulum in Rotary Motion

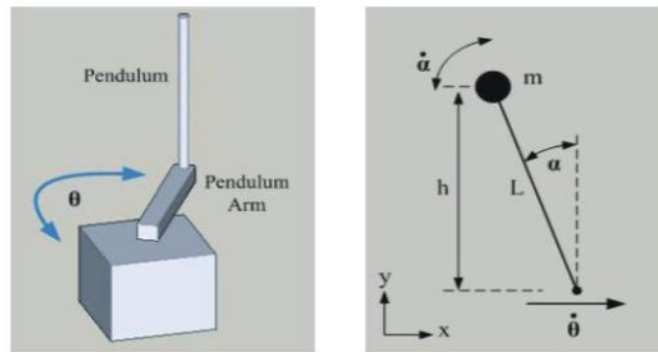


Figure 2.4

SMC Demonstration on Double Inverted Pendulum



The LQR algorithm is also used in the higher degree-of-freedom of its kind, Rotary Double Inverted Pendulum (RDIP). However, LQR alone cannot control the actual system, despite the simulation model (Sukontanakarn, 2011). As seen in Figure 2.5, adaptive LQR based on neural networks is offered to enhance the performance of the traditional LQR.

Figure 2.5

Rotary Double Inverted Pendulum System

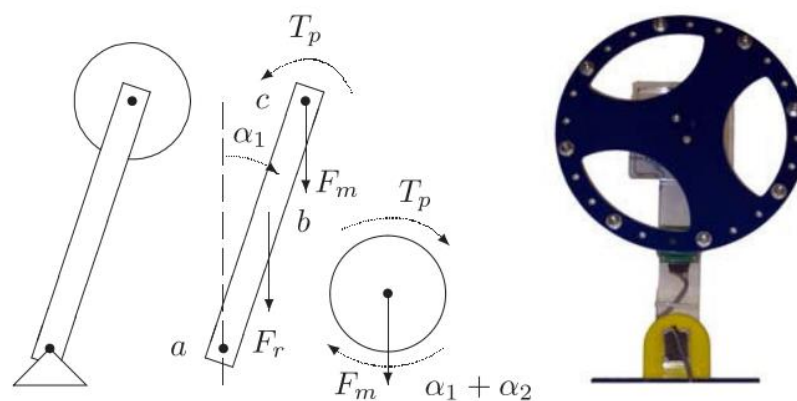


2.2 Reaction Wheels Inverted Pendulum

In most control algorithms to stabilize the classical inverted pendulum, horizontal forces on the support of pendulum are provided by cart mechanism. Using cart movement, it can be used to stabilize single or even multiple inverted pendulums. Unfortunately, the cart requires a decent amount of surface to operate a stabilized inverted pendulum. Furthermore, the actuators controlling the cart need to be placed in order to balance the inverted pendulum (Meyer et al., 2009b). Hence, another stabilization technique using moment exchange with a rotating inertia is used. This operates with a fixed pivot point and requires less operating area as shown in Figure 2.6.

Figure 2.6

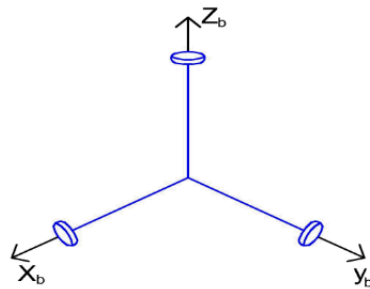
Schematics and Device of Reaction Wheel Inverted Pendulum System



In Aerospace Technology, reaction wheels are used to control the orientation of satellites, several configurations of reaction wheels are specified (Kök, 2012). Each reaction wheel torque will affect the axes of rotation of the robots. Thus, Standard Orthogonal 3-wheel configuration is the simplest one because each reaction wheel axis of rotation corresponds parallelly to the X, Y, and Z-axis of the satellites as shown in Figure 2.7.

Figure 2.7

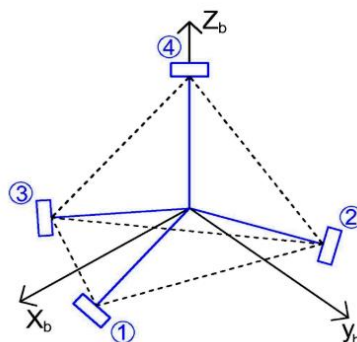
Standard Orthogonal 3-wheel Configuration



Another configuration of reaction wheel is the tetrahedron as shown in Figure 2.8. The advantage of this configuration is that torque produced from the reaction wheel is twice as much compared with single wheel. Normally, three reaction wheels is enough to control the satellites. But they use redundant fourth wheel as the common practice in satellite design. This comes with the penalty on large weight using this configuration (KARATAŞ, 2006). Since this is not the case in the thesis, only three reaction wheels are considered.

Figure 2.8

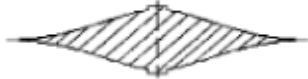



Tetrahedron Configuration of Four Reaction Wheels



The cross sectional geometry is also considered when creating reaction wheels (Kumar et al., 2015b). This paper uses STUDSAT-2 satellite for designing reaction wheel. The geometry introduced here has some of its advantages and disadvantages as shown in Figure 2.9. Reaction wheels used should be able to store the desired angular momentum, so the moment of inertia needed can be found using desired angular momentum and desired angular velocity. IMU sensor location also needs to be considered (Ramm & Sjöstedt, 2015). The optimal sensor placement needs to be far away from the reaction wheels, considering both electromagnetic induction from the motor, and the vibrations in the structure.

Figure 2.9

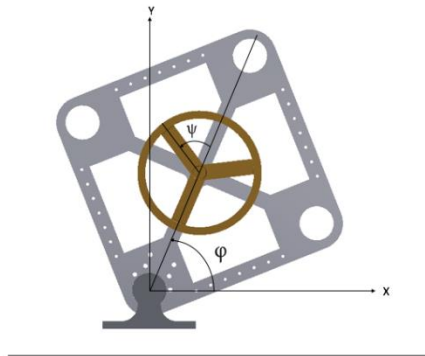
Different Geometry of Reaction Wheel

| Flywheel geometry | Cross section |
|-------------------------------|--|
| Disc |  |
| Modified constant stress disc |  |
| Conical disc |  |
| Flat unpierced disc |  |

In conventional inverted pendulum, the reaction wheel was attached to a rod that connected to pivot point, and that wheel uses the inverted pendulum to balance. The another 2D prototype of inverted pendulum was developed in the cube frame (Posada, 2017). The prototype and electrical design were presented, in addition to the multiple elements to validate the selected choices as shown in Figure 2.10. Then, a PID and LQR controller were implemented on both theoretical and the testbed to compare the behavior. The author also made a good note on each step that was executed.

Figure 2.10

Model of a Cube in 2D

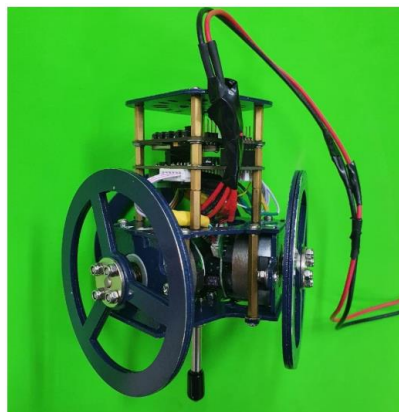


2.3 3D Reaction Wheels Inverted Pendulum Testbeds

In real applications, three-dimensional space is more prefer balancing mechanism benchmark than a simple inverted pendulum even though both came from the same concepts. The three wheels stick robot is an example of a 3D inverted pendulum. It is designed to balance itself on a single rod based on inverted pendulum concept and uses of reaction wheels (Vadrukchid, 2022). This robot balance itself by using the angular momentum generated by three reaction wheel and make it standing straight. A design in mechanism, electrical, and control are presented in this research as shown in Figure 2.11. The coupled effect between Roll and Pitch direction makes the LQR controller perfect for decoupled the system. The result shows that it can withstand an amount of disturbance and still balance itself upright.

Figure 2.11

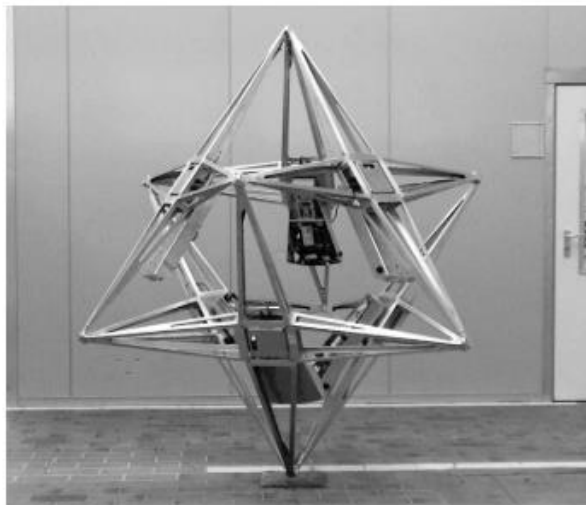
Model of the Three-Wheel Stick Robot



At Zurich, the Swiss Federal Institute of Technology, multiple accelerometers were also be used to determine a roll and pitch angle of a custom rigid body in this work (Trimpe & D'Andrea, 2010). This method is applicable to both static and dynamic scenarios of the body as shown in Figure 2.12. The estimated roll and pitch can be found using optimal linear estimator with least-squares techniques. In the two following years, the same institution published another fascinating balancing work. The Cubli is the 3D inverted pendulum testbed that can balance on its corner, its edge, jump up, and surprisingly walk on its own (Gajamohan et al., 2012).

Figure 2.12

The Balancing Cube, Balancing Using Accelerometer



The nonlinear dynamics of the Cubli were derived using Kane's equation that uses vector cross and dot products of vectors (Rambery et al., 2012). In the end, the reduced dynamics was used for control design. Two different control approaches were used, one was backstepping for parameter tuning, and other was feedback linearization for extended to control 3D case as shown in Figure 2.13 (Muehlebach & D'Andrea, 2016). The process of development of the Cubli, including its design choices, were presented in (Gajamohan et al., 2013). In 2016, another version of Cubli with more in-depth details was brought out as shown in Figure 2.14. The author focused on both theoretical and physical development of the cube. The hardware design is explained in detail and simple to follow though. PID and LQR were used to balance the cube, and were validated by both theoretical simulation and actual cube itself (Bjerke & Perhsson,

2016). On the same year, a review on mechanism of Cubli and control techniques that can be adapted to Cubli balancing problem. To summarize, modern predictive control (MPC) was suggested as to solve the braking mechanism, which is the biggest obstacle designing the robot (Singh et al., 2016).

Figure 2.13

The Cubli Balances on Corner

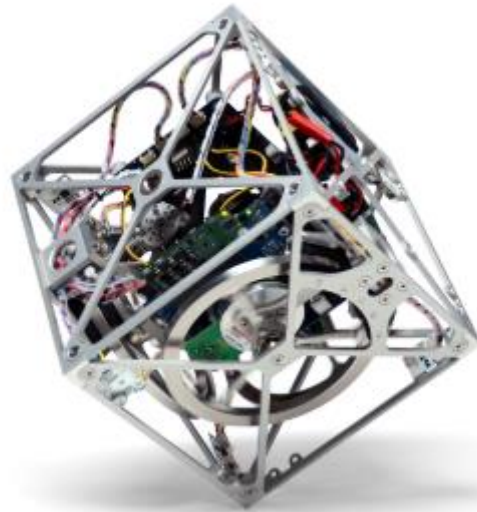
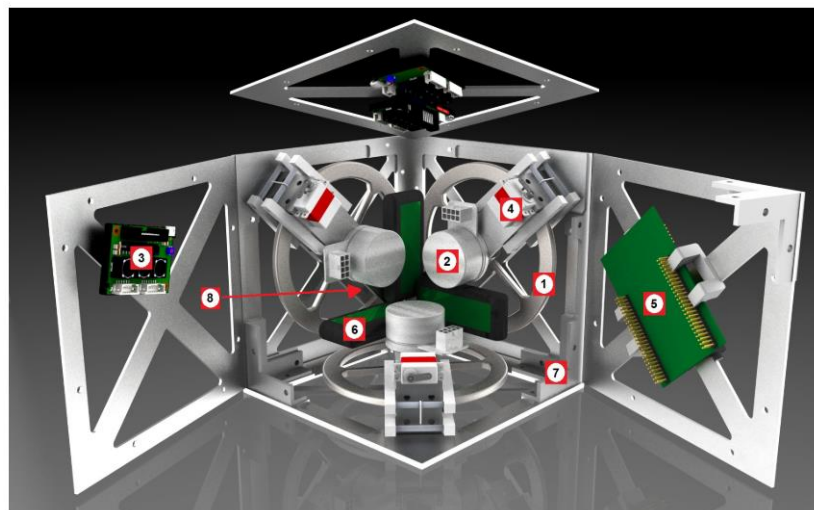


Figure 2.14

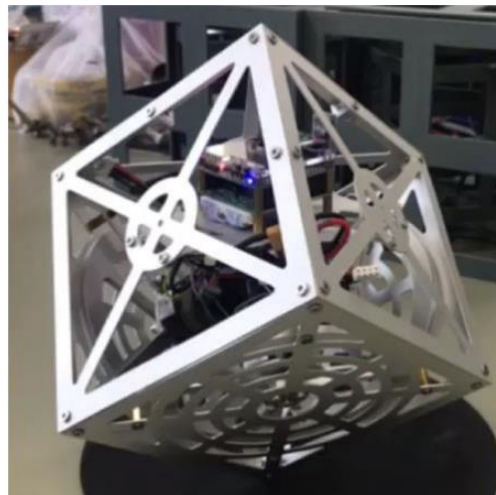
Another Version of Cubli



Other researchers took an alternative approach to balance this cubical robot. Lagrange's method is used to derive the dynamic model of the Cubli and verified in theory by numerical simulation (Chen et al., 2017). The attitude and heading reference system (AHRS) and parallel PID controller were proposed to balance the robot. AHRS, replacing IMUs, was used to derive the angle and analyzed the controllability of the cube robot (Liao et al., 2020). Another used reinforcement learning that proves already in many fields of study, it can solve complex problems, including mechatronics as shown in Figure 2.15. The results were compared with traditional LQR algorithms. The outcome demonstrates that in the field of balanced control research, reinforcement learning may be an optional technique. (Tangnararatchakit, 2021).

Figure 2.15

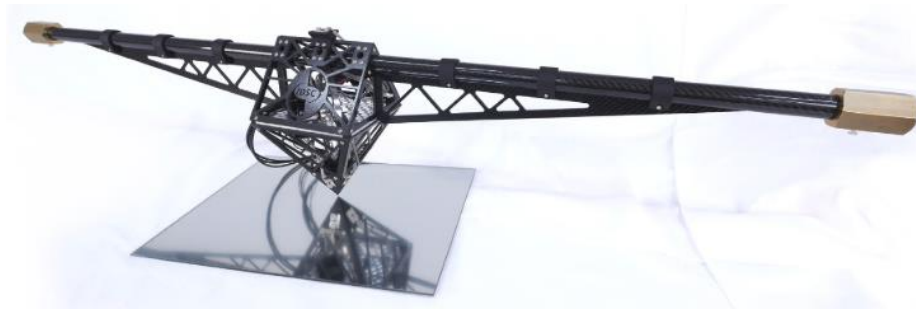
The Inverted Cube with 3 Reaction Wheels Installed



In 2023, a descendant of Cubli has been proposed as shown in Figure 2.16. A new modified name as the One-wheel Cubli has been presented (Hofer et al., 2023). It is a cube that has a unique beam attached to it. The distinctive point of this design is that only a single reaction wheel is required to balance the Cubli. By adding two masses at the end of a cantilever beam, rotational inertia in the pitch direction is significantly higher than the roll direction. This creates a separation in the underlying tilt dynamics and makes the system controllable. Controllability is maximized when the ratio of two principal inertia values equals to the square of the silver ratio. They also mention that the symmetric placements of IMUs on a regular polygon should be considered over non-symmetric placements.

Figure 2.16

The One-Wheel Cubli

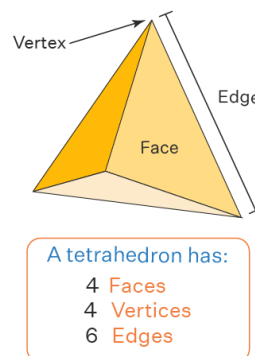


2.4 Tetrahedron Geometry

Tetrahedron is one of the three-dimensional shapes that has four triangular faces. Generally, one of the faces will be considered as the base, and the other three together form the pyramid. It has a triangular base and hence it is also called a triangular pyramid. It consists of 4 faces, 6 edges, in which all faces are identical triangles.

Figure 2.17

Tetrahedron Properties¹



2.5 Chapter Summary

Overall, this chapter walks through the principal idea of all previous research, the inverted pendulum. Then, more research goes through balance method with reaction wheels. Lastly, the 3D testbeds are reviewed, and these give the main contribution of their works.

¹ from <https://www.cuemath.com/geometry/tetrahedron/>

CHAPTER 3

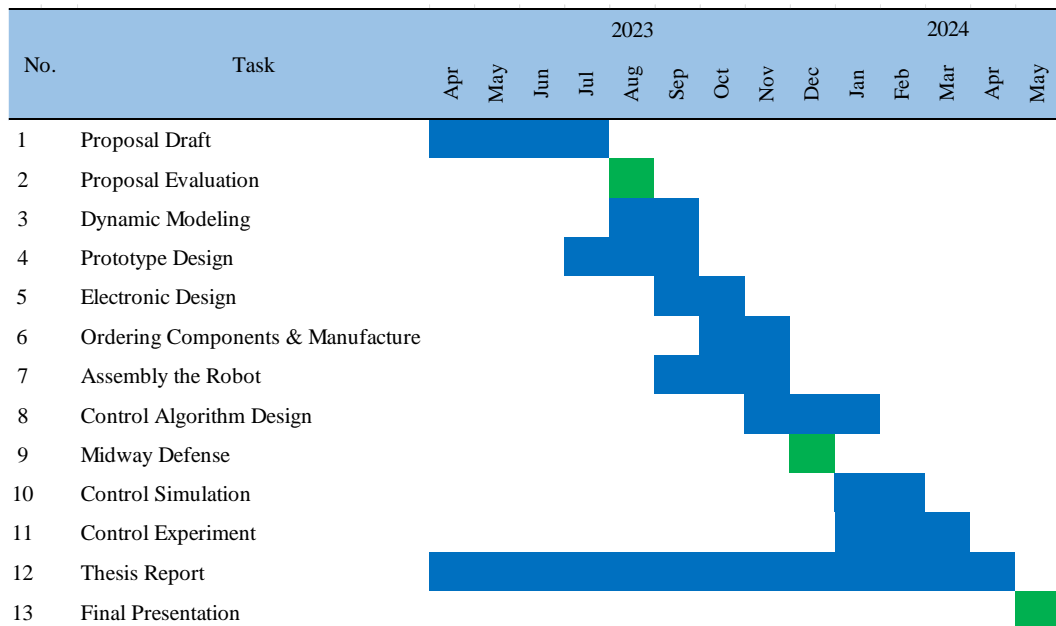
METHODOLOGY

This chapter gives an overview of the timeline of the thesis. To begin with, the methodology of robot's design is presented, with electrical components included, until finished the robot. Then, the dynamic model is determined from the actual robot using Lagrange method and rearranges into linearized state space of the system. Linear Quadratic Regulator is used for a control algorithm for this robot. Finally, control simulation and experiment are explained.

3.1 Timeline

Table 3.1

Gantt Chart Task Schedule for the Master's Thesis



3.2 Conceptual Design

Figure 3.1 and Figure 3.2 shows the conceptual design of the Tetrahedron Robot with three reaction wheels. The material for frame is initiated to be aluminum with thickness of 2 mm. This design gives more importance control on the roll and pitch direction of the robot, and to allow motor fitting, the reaction wheel rotational axis is set to be

around 60 degrees ($1 - \sqrt{3} - 2$ triangle) measured from horizontal plane of the robot with 120 degrees apart from top view position.

Figure 3.1

The Conceptual CAD of Tetrahedron Robot

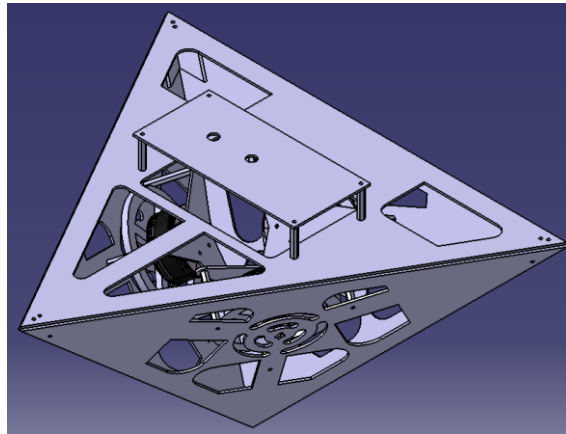
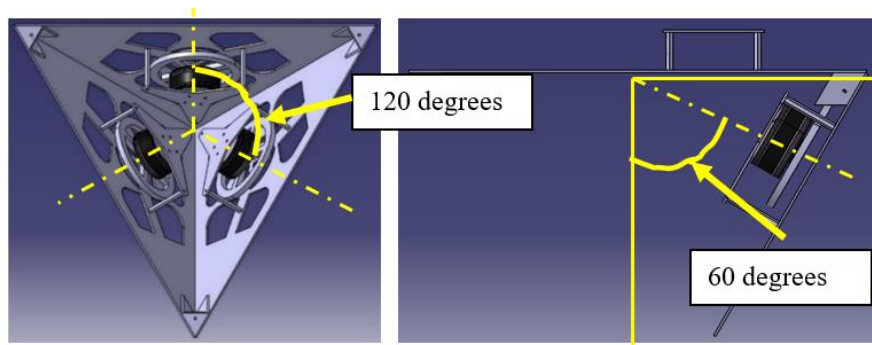


Figure 3.2

Reaction Wheels Orientation Inside The Conceptual Robot

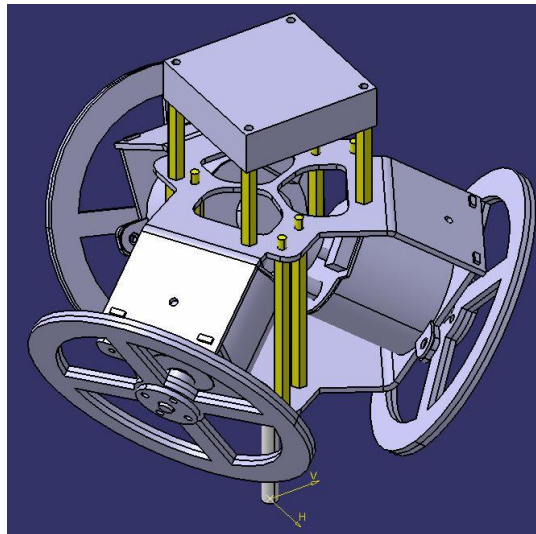


3.3 Mechanical Prototype Design

After realizing the need to add more components and make structure more rigid, the revised design is created as shown in Figure 3.3. The design considers reducing sharp edges to reduce stress concentration, creating a more appealing design, and preventing accidents during testing. Reaction wheels diameter is selected to compensate between diameter and weight. Circuit board is placed at the top of the robot with hex columns to support the board. The space between the board and the structure is designed to put small wires connected from board to the motor.

Figure 3.3

The Prototype Design in CAD



The motor orientation is checked as same as in conceptual design as shown in Figure 3.4. The motor alignment is shifted from the original center line in clockwise direction for 10 millimeters to make motors get as close as possible to the robot's center of mass. The structure is separated into two parts: the top and bottom sheet. Later when the design is finalized in CATIA V5 CAD software, The prototype CAD design is created with a Polylactic acid (PLA) plastic material from a 3D printer as shown in Figure 3.5. The prototype is made to check the dimensions and has no components collision.

Figure 3.4

Reaction Wheels Orientation in Prototype Robot

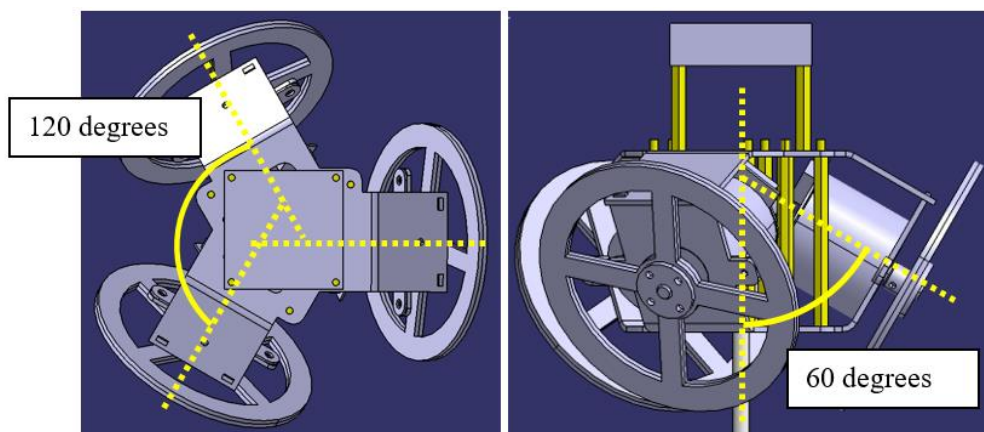
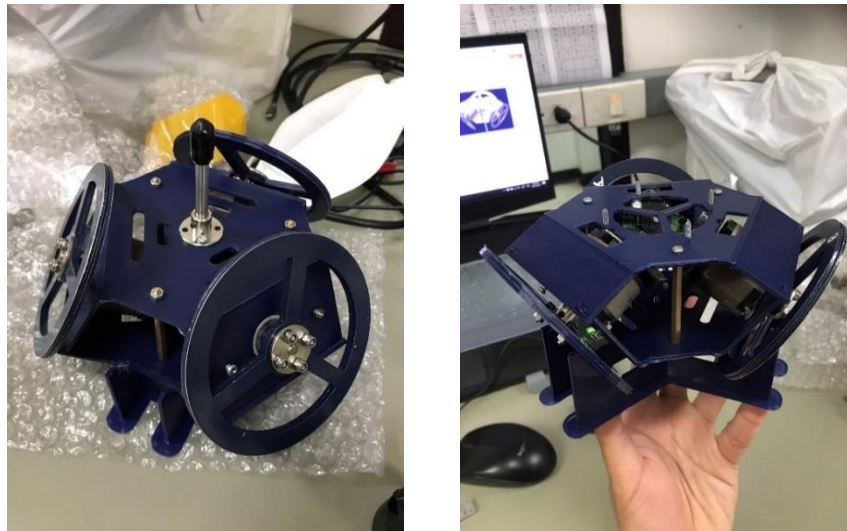


Figure 3.5

Checking Components Fit Within the Structure



After checking that all components can be fitted within the structure, the actual robot is made using Aluminum grade 1 sheet with 2 mm thickness. It is cut and folded by a laser cutting and sheet metal bending machine with the result pieces as shown in Figure 3.6. Aluminum is great for withstanding vibration, so it is chosen. The bending degree are checked to ensure correct bending process as shown in Figure 3.7. As Top sheet needs to be 30-degree bend angle, and Bottom sheet needs to be 60-degree bend angle.

Figure 3.6

The Actual Robot Prototype Structure

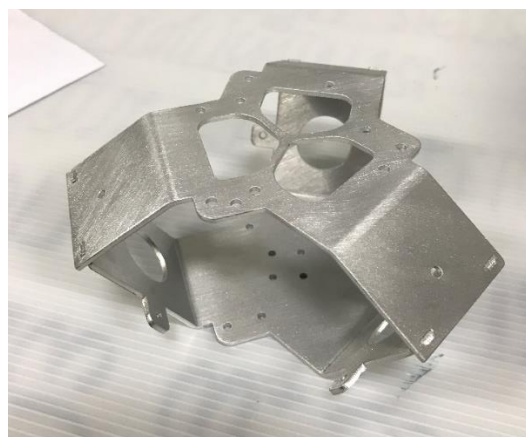


Figure 3.7

Checking the Bending Degree of Structure of Top and Bottom Sheet



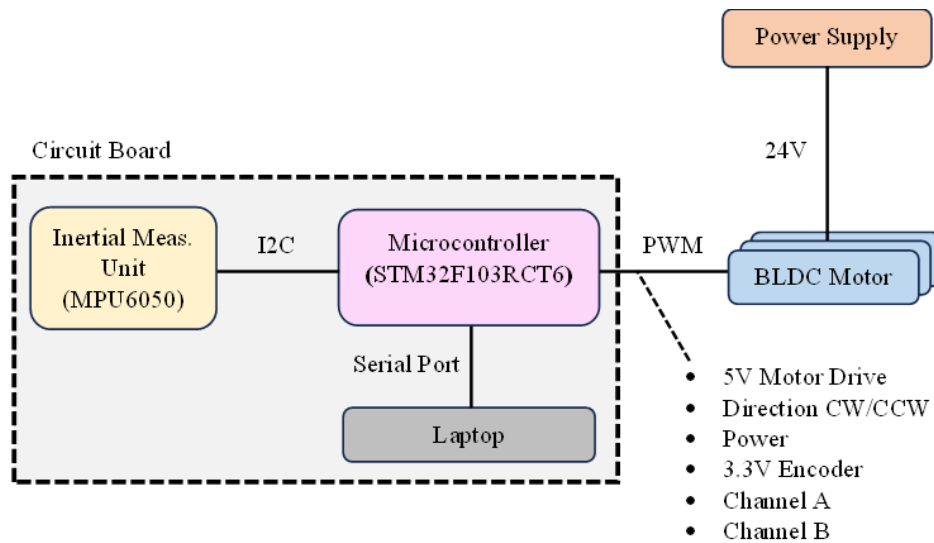
3.4 Electrical Design

The microcontroller (MCU) is STM32, a flexible microcontroller that can manage to communicate with varied hardware. An IMU used is MPU6050, for measuring the orientation and other motion-related features of the robot orientations' data. The sensor data is sent via I2C protocol. Each motor driving reaction wheel links to an encoder to keep track of motor shaft speed, and encoders are connected to the STM32.

Motors used to drive reaction wheels are brushless DC (BLDC) motors. This motor, with a 100-line dual channel A and B encoder and a driver built-in is used. Power supply of 24 volts is used to supply power to motors. Motor speed is controlled using Pulse Width Modulation (PWM) and there is one signal wire to control the rotation direction. All data records can be transferred to analyze further by connecting microcontroller via serial port to laptop and print out to read. The overall schematic for electric is presented in Figure 3.8.

Figure 3.8

Simplified Circuit Diagram

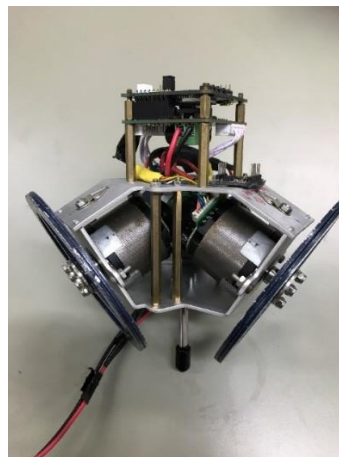


3.5 Dynamic Model

After finishing design and assembling the mechanicals and electrical components together, the complete robot is finished. Next step is to govern the dynamics model, describing the physical behaviors of this robot. The tetrahedron robot is made up of four rigid bodies: a frame structure and three reaction wheels. The other bodies, such as motors and microcontrollers, can be interpreted as part of one of these four bodies. The robot is assembled as shown in Figure 3.9 to finish the robot physical appearance.

Figure 3.9

The Tetrahedron Robot

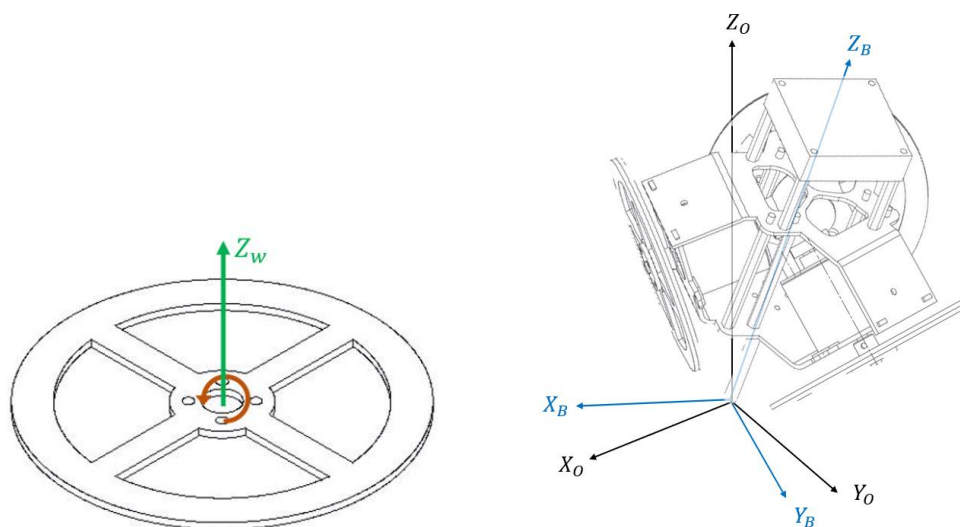


3.5.1 Coordinates Representation

Using the Lagrange methods to connect the action of forces on bodies with the motion of the robot is the primary technique employed in this thesis to build the dynamics model. The coordinate system of this robot is separated into three sub-systems: motor coordinates, frame coordinates and global coordinates. In the calculation process, motor coordinates are used to determine the rotational position and velocities from encoders. In which it will transform to frame coordinates, which have the gyro/accelerometer sensor. Finally, frame coordinates are moved to global coordinates for determining the dynamic model.

Figure 3.10

Three Coordinates System

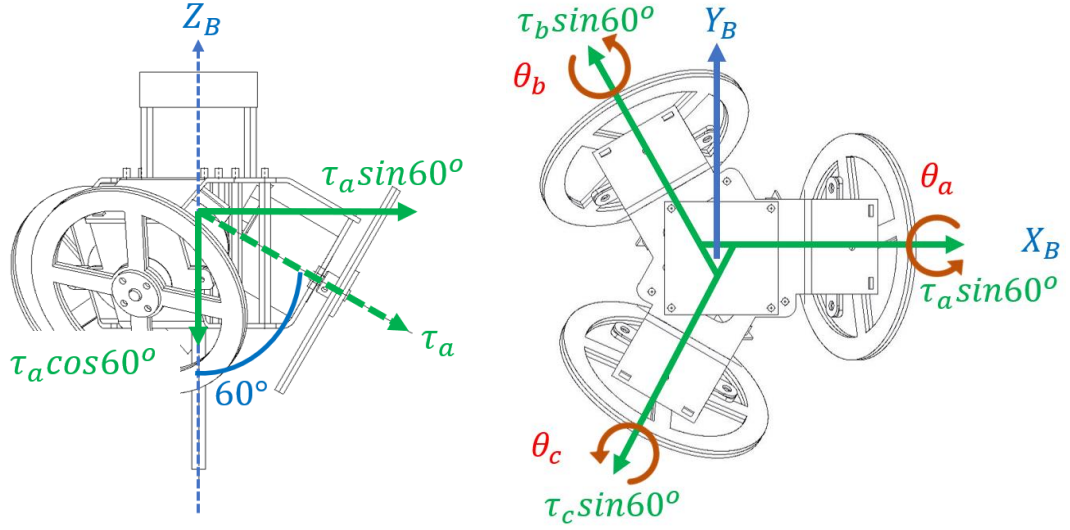


(a) motor coordinates (b) frame (blue) and global (black) coordinates

In motor coordinates, each motor is placed 120-degree apart and tilts downward 60 degrees measured from vertical axis as shown in Figure 3.11. In motor circuits, the manufacturer has an encoder built-in into the circuits. Thus, the position and velocity of the motor can be read out, which is also the same value as reaction wheels. For the frame coordinates, it has a gyro/accelerometer sensor attached to measure the Roll, Pitch and Yaw angle of the robot. Global coordinates will be used as reference with frame coordinates.

Figure 3.11

Motor to Frame Coordinates Schematic Diagram



θ_a , θ_b , and θ_c are denoted as the counterclockwise angle between the reaction wheels and the rod at the bottom. τ_a , τ_b , and τ_c are respectively torque of three reaction wheels which follow the same direction as counterclockwise angles. The green arrows indicate motor coordinates while the blue arrows are frame coordinates. X_B , Y_B and Z_B represent x-axis, y-axis and z-axis of the frame coordinates. The resultant force for each axis of frame coordinates related to torque from all three motors is shown in equation (3.1) through (3.3).

$$\sum \tau_{X_B} = \tau_a \sin(60) - \tau_b \sin(60) \sin(30) - \tau_c \sin(60) \sin(30) \quad (3.1)$$

$$\sum \tau_{Y_B} = \tau_b \sin(60) \cos(30) - \tau_c \sin(60) \cos(30) \quad (3.2)$$

$$\sum \tau_{Z_B} = -\tau_a \cos(60) - \tau_b \cos(60) - \tau_c \cos(60) \quad (3.3)$$

The relationship between torque of motor and frame coordinate can be represented with transformation matrix (T_{BM}) as shown in equation (3.4).

$$T_{BM} = \begin{bmatrix} \sin(60) & -\sin(60) \sin(30) & -\sin(60) \sin(30) \\ 0 & \sin(60) \cos(30) & -\sin(60) \cos(30) \\ -\cos(60) & -\cos(60) & -\cos(60) \end{bmatrix} \quad (3.4)$$

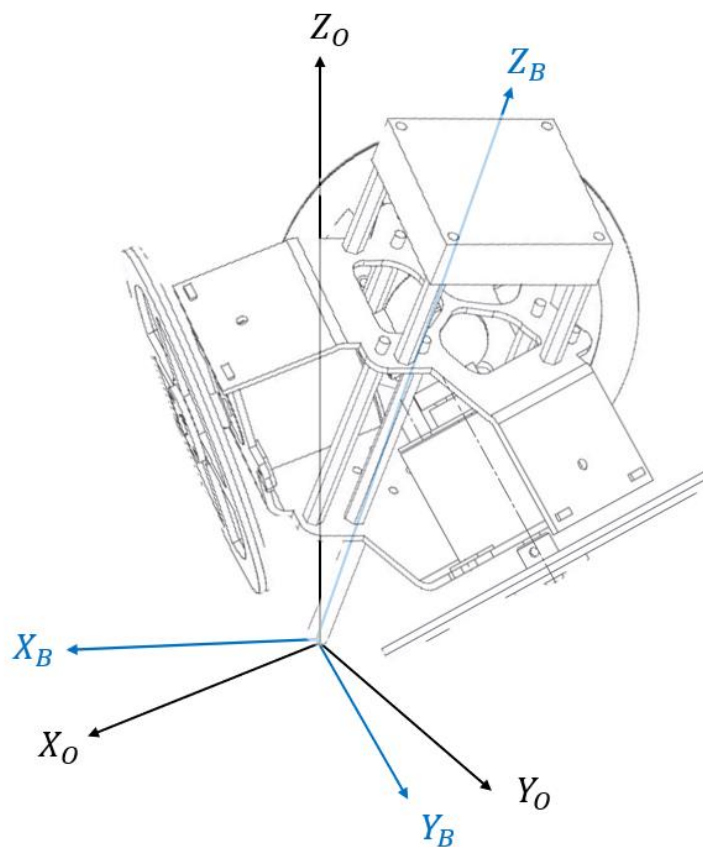
And relations can simplify into equation (3.5).

$$\begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_Z \end{bmatrix}_B = T_{BM} \begin{bmatrix} \tau_a \\ \tau_b \\ \tau_c \end{bmatrix}_M \quad (3.5)$$

After that, the transformation from frame to global coordinates can be shown using Figure 3.12. Frame coordinates $B-X_B Y_B Z_B$ and global coordinates $O-X_O Y_O Z_O$ are defined. The rotation that related between these two coordinates depends on the Roll angle (α) for X-axis, the Pitch angle (β) for Y-axis, and the Yaw angle (γ) for Z-axis. The rotation matrix, T_{OB} , relating the frame and global coordinate are shown as equation (3.6) through (3.9).

Figure 3.12

Frame to Global Coordinates Schematic Diagram



$$R_X(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (3.6)$$

$$R_Y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (3.7)$$

$$R_Z(\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

Where the total rotation matrix or transformation matrix

$$T_{OB} = R_Z(\gamma)R_Y(\beta)R_X(\alpha) \quad (3.9)$$

3.5.2 Generalized Force

The force acting for this system is a torque for each X, Y, and Z-axis. Torques are generated from reaction wheels rotating from motor reaction torque. Torque associates with X axis of roll is denoted as τ_α . For Y and Z axes are denoted as τ_β , and τ_γ respectively. To transform motor torques into global torques, it is shown in equation (3.10) below.

$$\begin{bmatrix} \tau_\alpha \\ \tau_\beta \\ \tau_\gamma \end{bmatrix}_O = -I_W T_{OB} T_{BM} \begin{bmatrix} \ddot{\theta}_A \\ \ddot{\theta}_B \\ \ddot{\theta}_C \end{bmatrix}_M \quad (3.10)$$

Whereas:

I_w denoted reaction wheel's moment of inertia

$\ddot{\theta}_A$ denoted the wheel A angular acceleration.

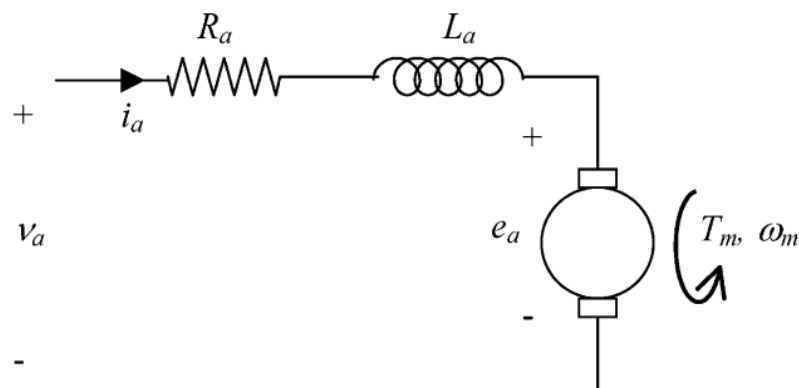
$\ddot{\theta}_B$ denoted the wheel B angular acceleration.

$\ddot{\theta}_C$ denoted the wheel C angular acceleration.

Since the wheels torques are directly resulted from each respective motor, they are varied by the voltage and angular velocity derived from DC motor equation as shown in Figure 3.13 and equation (3.11) below.

Figure 3.13

DC Motor Schematic



Where i_a is armature coil current (A)

R_a is armature coil resistance (Ω)
 L_a is armature coil inductance (H)
 V_a or V_{in} is motor armature / input voltage (V)
 e_a or V_b is back EMF voltage (V)
 K_T is motor torque constant
 K_b is back EMF constant
 T_m or τ is torque from motor ($N \cdot m$)
 ω_m or $\dot{\theta}_w$ is motor angular velocity (rad/s) same as wheels'

For V_b proportional with $\dot{\theta}$,

$$V_b = K_b \dot{\theta}_w \quad (3.11)$$

For T proportional with i_a ,

$$T = K_T i_a \quad (3.12)$$

From Kirchhoff Voltage Law,

$$V_a = i_a R_a + L \frac{d}{dt} i_a + V_b$$

Assume L is very small compared to R_a ,

$$V_a = i_a R_a + V_b \quad (3.13)$$

Combine equation (3.10) - (3.12),

$$\tau = \frac{K_T}{R_a} V_a - \frac{K_T K_b}{R_a} \dot{\theta}_w \quad (3.14)$$

To simplify, we combined constant coefficients into $K_1 = \frac{K_T}{R_a}$ and $K_2 = \frac{K_T K_b}{R_a}$

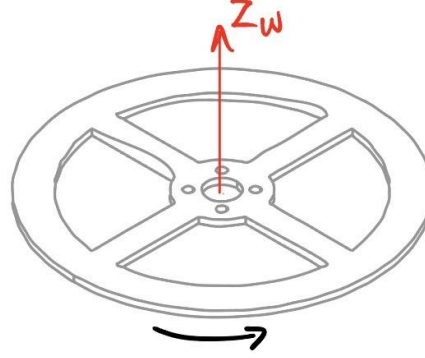
$$\tau = K_1 V_a - K_2 \dot{\theta}_w \quad (3.15)$$

Then, to find motor A, B and C torques, which denoted as τ_a, τ_b , and τ_c , can be found using same equation as equation (3.15). Combine $\tau = I\ddot{\theta}$, we now have ($V_a = V_{in}$)

$$\begin{aligned}
 I\ddot{\theta} &= K_1 V_{in} - K_2 \dot{\theta}_w \\
 \ddot{\theta} &= \frac{K_1 V_{in} - K_2 \dot{\theta}_w}{I_w}
 \end{aligned} \quad (3.16)$$

Figure 3.14

Wheel Angular Acceleration Direction



Wheels angular acceleration are shown in Figure 3.14 and in the equation (3.17) though (3.19) below. Assuming all wheels are in same shape, $I_a = I_b = I_c = I_w = I_{w_z}$ because wheel rotates around Z-axis respect to wheels.

$$\ddot{\theta}_a = \frac{K_1 V_a - K_2 \dot{\theta}_a}{I_w} \quad (3.17)$$

$$\ddot{\theta}_b = \frac{K_1 V_b - K_2 \dot{\theta}_b}{I_w} \quad (3.18)$$

$$\ddot{\theta}_c = \frac{K_1 V_c - K_2 \dot{\theta}_c}{I_w} \quad (3.19)$$

3.5.3 Kinetic Energy

Because this robot concept is like the simple inverted pendulum, it is rotating around the pivot point, which located at the end of rod in frame body. Hence, the velocities contributed to kinetic energy are sourced from angular velocities alone, resulting as angular kinetic energy. Each moving body can be added up to get the desired total kinetic energy.

$$T = T_B + \sum_{i=1}^3 T_{wi} \quad (3.20)$$

Where T_B is structure's kinetic energy and T_{wi} is the kinetic energy of the i -th reaction wheel, that is, A, B, and C. As shown in equation (3.21), for each reaction wheel inherits the total kinetic energy of reaction wheels and for the frame's kinetic energy as shown in equation (3.22).

$$\sum_{i=1}^3 T_{wi} = \sum_{i=1}^3 \frac{1}{2} I_w \dot{\theta}_i^2$$

$$= \frac{1}{2} I_w (\dot{\theta}_a^2 + \dot{\theta}_b^2 + \dot{\theta}_c^2)$$
(3.21)

$$T_B = \frac{1}{2} (I_x \dot{\alpha}^2 + I_y \dot{\beta}^2 + I_z \dot{\gamma}^2)$$
(3.22)

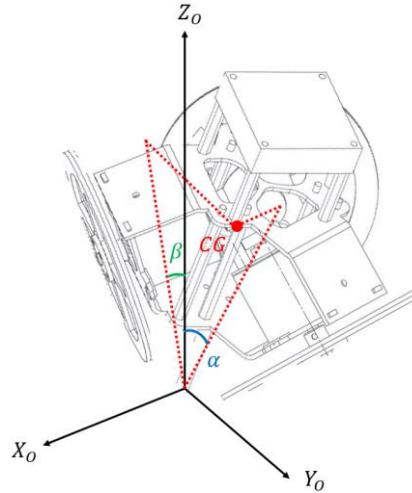
The angular velocity and inertia are contributed to the kinetic energy for each body. For the frame, the moment of inertia for each axis is represented by I_x , I_y , and I_z , denoted the assigned axis in subscript.

3.5.4 Potential Energy

The same pivot point, used in kinetic energy, can also be used for reference point in potential energy too. Since the body frame of robot rotates around that point, the potential energy from the system comes solely from frame's center of mass, including all three reaction wheels. Roll and pitch direction can contribute to potential energy as shown in Figure 3.15, it can be shown in the equation below.

Figure 3.15

Potential Energy Schematics



$$V = mgl \cos(\alpha) \cos(\beta)$$
(3.23)

3.5.5 Lagrange Equations

To find the dynamics model of system with Lagrange method, we need to find the system's potential and kinetic energy. Since we know both earlier, the Lagrangian (L) is the difference between these energies.

$$\begin{aligned}
L &= T - V \\
&= T_B + T_w - V
\end{aligned} \tag{3.24}$$

The dynamic equations, explaining the system behaviors, can be obtained by the following Lagrange equations as shown below.

$$\frac{\partial L}{\partial q_i} - \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) + \lambda \frac{\partial f}{\partial q_i} = 0 \tag{3.25}$$

Where q_i is the interested i -th generalized coordinate of the system. f is the force generalized in that i -th direction. The direction, distances, or angles, that are interested in, is assigned the systems' six generalized coordinates as follow:

$$q_{6 \times 1} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \theta_a \\ \theta_b \\ \theta_c \end{pmatrix} \tag{3.26}$$

And the force that constrained the motion for each generalized coordinate is torque respect to that coordinate. For each of coordinates will represent separated but may couple one dynamic equation.

$$\lambda \frac{\partial f}{\partial q_i} = \begin{pmatrix} \tau_\alpha \\ \tau_\beta \\ \tau_\gamma \\ \tau_a \\ \tau_b \\ \tau_c \end{pmatrix} \tag{3.27}$$

When combining equation (3.21) - (3.27), we get the dynamic model of the systems, each for generalized coordinates, which total of six dynamic equations.

$$\begin{aligned}
\ddot{\alpha} &= \frac{4mgl \cos(\beta) \sin(\alpha) - 2(\sqrt{3} \cos(\beta) \cos(\gamma) + \sin(\beta))K_{xa}V_a}{4I_x} \\
&+ \frac{(-2 \sin(\beta) + \cos(\beta) (\sqrt{3} \cos(\gamma) - 3 \sin(\gamma)))K_{xb}V_b}{4I_x} \\
&+ \frac{(\sqrt{3} \cos(\beta) \cos(\gamma) - 2 \sin(\beta))K_{xc}V_c}{4I_x} + \frac{(2\sqrt{3} \cos(\beta) \cos(\gamma) + 2 \sin(\beta))K_{ya}\dot{\theta}_a}{4I_x} \\
&+ \frac{(-\sqrt{3} \cos(\beta) \cos(\gamma) + 2 \sin(\beta) + 3 \cos(\beta) \sin(\gamma))K_{yb}\dot{\theta}_b}{4I_x} \\
&+ \frac{(-\sqrt{3} \cos(\beta) \cos(\gamma) + 2 \sin(\beta) - 3 \cos(\beta) \sin(\gamma))K_{yc}\dot{\theta}_c}{4I_x}
\end{aligned} \tag{3.28}$$

$$\begin{aligned}
& \ddot{\beta} \\
&= \frac{4mgl \cos(\alpha) \sin(\beta) + 2(\cos(\beta) \sin(\alpha) + \sqrt{3}(-\cos(\gamma) \sin(\alpha) \sin(\beta) + \cos(\alpha) \sin(\gamma))) (K_{xa}V_a - K_{ya}\dot{\theta}_a)}{4I_y} \\
&+ \frac{(\sin(\alpha) (2 \cos(\beta) + \sin(\beta) (\sqrt{3} \cos(\gamma) - 3 \sin(\gamma))) - \cos(\alpha) (3 \cos(\gamma) + \sqrt{3} \sin(\gamma))) (K_{xb}V_b - K_{yb}\dot{\theta}_b)}{4I_y} \\
&+ \frac{(\cos(\alpha) (3 \cos(\gamma) - \sqrt{3} \sin(\gamma)) + \sin(\alpha) (2 \cos(\beta) + \sin(\beta) (\sqrt{3} \cos(\gamma) + 3 \sin(\gamma)))) (K_{xc}V_c - K_{yc}\dot{\theta}_c)}{4I_y} \\
& \tag{3.29}
\end{aligned}$$

$$\begin{aligned}
& \ddot{\gamma} \\
&= \frac{2(\cos(\alpha) (\cos(\beta) - \sqrt{3} \cos(\gamma) \sin(\beta)) - \sqrt{3} \sin(\alpha) \sin(\gamma)) (K_{xa}V_a - K_{ya}\dot{\theta}_a)}{4I_z} \\
&+ \frac{(\cos(\alpha) (2 \cos(\beta) + \sin(\beta) (\sqrt{3} \cos(\gamma) - 3 \sin(\gamma))) + \sin(\alpha) (3 \cos(\gamma) + \sqrt{3} \sin(\gamma))) (K_{xb}V_b - K_{yb}\dot{\theta}_b)}{4I_z} \\
&+ \frac{(\sin(\alpha) (-3 \cos(\gamma) + \sqrt{3} \sin(\gamma)) + \cos(\alpha) (2 \cos(\beta) + \sin(\beta) (\sqrt{3} \cos(\gamma) + 3 \sin(\gamma)))) (K_{xc}V_c - K_{yc}\dot{\theta}_c)}{4I_z} \\
& \tag{3.30}
\end{aligned}$$

$$\ddot{\theta}_a = \frac{K_{xa}V_a - K_{ya}\dot{\theta}_a}{I_w} \tag{3.31}$$

$$\ddot{\theta}_b = \frac{K_{xb}V_b - K_{yb}\dot{\theta}_b}{I_w} \tag{3.32}$$

$$\ddot{\theta}_c = \frac{K_{xc}V_c - K_{yc}\dot{\theta}_c}{I_w} \tag{3.33}$$

3.5.6 State-space Representation

With state-space representation, instead of a set of equations that's complicated to deal with, it also can deal with system that's coupled. It is used for linear systems, that will be used later. For k order system, state variables: $x_1, x_2, x_3, \dots, x_k$, and control inputs:

u_1, u_2, \dots, u_l , the state vector $\mathbf{x}(t) = \begin{pmatrix} x_1(t) \\ \vdots \\ x_k(t) \end{pmatrix}$ and input vector $\mathbf{u}(t) = \begin{pmatrix} u_1(t) \\ \vdots \\ u_l(t) \end{pmatrix}$ will

have general form in $\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} = f(\mathbf{x}, \mathbf{u}, t)$. For the measured outputs: y_1, y_2, \dots, y_m has

output or observation vector $\mathbf{y}(t) = \begin{pmatrix} y_1(t) \\ \vdots \\ y_m(t) \end{pmatrix}$, and $\mathbf{y}(t) = f(\mathbf{x}(t), \mathbf{u}(t), t)$. With the

robot's state-space, input and output of the system will be as shown in Table 3.2.

Table 3.2*States, Inputs, and Outputs of the System*

| Variables | State | Input | Output |
|---|-------|-------|--------|
| Robot Orientation (α, β, γ) | ✓ | | ✓ |
| Robot Angular Velocity ($\dot{\alpha}, \dot{\beta}, \dot{\gamma}$) | ✓ | | |
| Reaction Wheels Angle ($\theta_a, \theta_b, \theta_c$) | ✓ | | ✓ |
| Reaction Wheels Angular Velocity ($\dot{\theta}_a, \dot{\theta}_b, \dot{\theta}_c$) | ✓ | | |
| Motor Voltage (V_a, V_b, V_c) | | ✓ | |

3.5.7 Linearized State-space

Finding a nonlinear function's gradient in each variable and translating it into a linear relation at that location or point is the process known as linearization. The previous dynamic equations model earlier are now reorganized as input vector \mathbf{u} and output state vector \mathbf{x} in equation (3.34).

$$\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} = f(\mathbf{x}, \mathbf{u}) \quad (3.34)$$

By linearizing about $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$, a steady-state point, this can alter the approximation. Consequently, changing $\mathbf{u} \rightarrow \bar{\mathbf{u}} + \Delta\mathbf{u}$ and $\mathbf{x} \rightarrow \bar{\mathbf{x}} + \Delta\mathbf{x}$.

$$\frac{d(\bar{\mathbf{x}} + \Delta\mathbf{x})}{dt} = f(\bar{\mathbf{x}} + \Delta\mathbf{x}, \bar{\mathbf{u}} + \Delta\mathbf{u}) \quad (3.35)$$

With $\Delta\mathbf{x} = \mathbf{x} - \bar{\mathbf{x}}, \Delta\mathbf{u} = \mathbf{u} - \bar{\mathbf{u}}$.

Using a Taylor's series expansion with only the first and the second term because $(x - \bar{x})^i \rightarrow 0$ when $i = 2, 3, 4, \dots$.

$$\frac{d(\bar{\mathbf{x}} + \Delta\mathbf{x})}{dt} = f(\bar{\mathbf{x}} + \Delta\mathbf{x}, \bar{\mathbf{u}} + \Delta\mathbf{u}) \approx f(\bar{\mathbf{x}}, \bar{\mathbf{u}}) + \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\bar{\mathbf{x}}, \bar{\mathbf{u}}} (\mathbf{x} - \bar{\mathbf{x}}) + \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{\bar{\mathbf{x}}, \bar{\mathbf{u}}} (\mathbf{u} - \bar{\mathbf{u}}) \quad (3.36)$$

Since both $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$ are equal to zero, $f(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = 0$ if the values of $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$ are selected under steady-state conditions. Lastly, the equation can be rewritten to the linearized form shown in the equation below.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (3.37)$$

Thus, the dynamic equation that linearized at steady-state point can represent in state-space matrix \mathbf{A} and input matrix \mathbf{B} for state-space matrix.

$$\mathbf{A} = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{mgl}{I_X} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{3}K_{2a}}{2I_X} & 0 & \frac{-\sqrt{3}K_{2b}}{4I_X} & 0 & \frac{-\sqrt{3}K_{2c}}{4I_X} \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{mgl}{I_Y} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{3K_{2b}}{4I_Y} & 0 & \frac{-3K_{2c}}{4I_Y} \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-K_{2a}}{2I_Z} & 0 & \frac{-K_{2b}}{2I_Z} & 0 & \frac{-K_{2c}}{2I_Z} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-K_{2a}}{I_w} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-K_{2b}}{I_w} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-K_{2c}}{I_w}
\end{bmatrix} \quad (3.38)$$

$$\mathbf{B} = \begin{bmatrix}
0 & 0 & 0 \\
\frac{-\sqrt{3}K_{1a}}{2I_X} & \frac{\sqrt{3}K_{1b}}{4I_X} & \frac{\sqrt{3}K_{1c}}{4I_X} \\
0 & 0 & 0 \\
0 & \frac{-3K_{1b}}{4I_Y} & \frac{3K_{1c}}{4I_Y} \\
0 & 0 & 0 \\
\frac{K_{1a}}{2I_Z} & \frac{K_{1a}}{2I_Z} & \frac{K_{1a}}{2I_Z} \\
0 & 0 & 0 \\
\frac{K_{1a}}{I_w} & 0 & 0 \\
0 & 0 & 0 \\
0 & \frac{K_{1b}}{I_w} & 0 \\
0 & 0 & 0 \\
0 & 0 & \frac{K_{1c}}{I_w}
\end{bmatrix} \quad (3.39)$$

The whole form of state-space is displayed below after the state and input matrix have been obtained as shown in equation (3.40).

$$\begin{bmatrix} \dot{\alpha} \\ \ddot{\alpha} \\ \dot{\beta} \\ \ddot{\beta} \\ \dot{\gamma} \\ \ddot{\gamma} \\ \dot{\theta}_A \\ \ddot{\theta}_A \\ \dot{\theta}_B \\ \ddot{\theta}_B \\ \dot{\theta}_C \\ \ddot{\theta}_C \end{bmatrix} = A \begin{bmatrix} \alpha \\ \dot{\alpha} \\ \beta \\ \dot{\beta} \\ \gamma \\ \dot{\gamma} \\ \theta_A \\ \dot{\theta}_A \\ \theta_B \\ \dot{\theta}_B \\ \theta_C \\ \dot{\theta}_C \end{bmatrix} + B \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \quad (3.40)$$

3.6 Linear Quadratic Regulator

Since the system is now linearized, it is reasonable to use a linear-type controller to control the balancing of the robot. Linear Quadratic Regulator (LQR) can give an optimal gain based on adjustable priority of given states order needed to be controlled and energy spent to control those states. It works by having a cost function to tell how the states can converge to the desired output.

For linear controller function,

$$u = -Kx \quad (3.41)$$

The optimal gain (K) that helps minimize cost function (J),

$$J = \int_0^{\infty} [x(t)^T Q(t)x(t) + u(t)^T R(t)u(t)] dt \quad (3.42)$$

Where Q is state-weighted matrix and R is the input-weighted matrix. In which both matrices are symmetrical. Then the optimal gain K for input control from (3.41) is found using the equation xxx

$$K = R^{-1}B'M \quad (3.43)$$

With the solution matrix, M , obtained by solving a Riccati's equation.

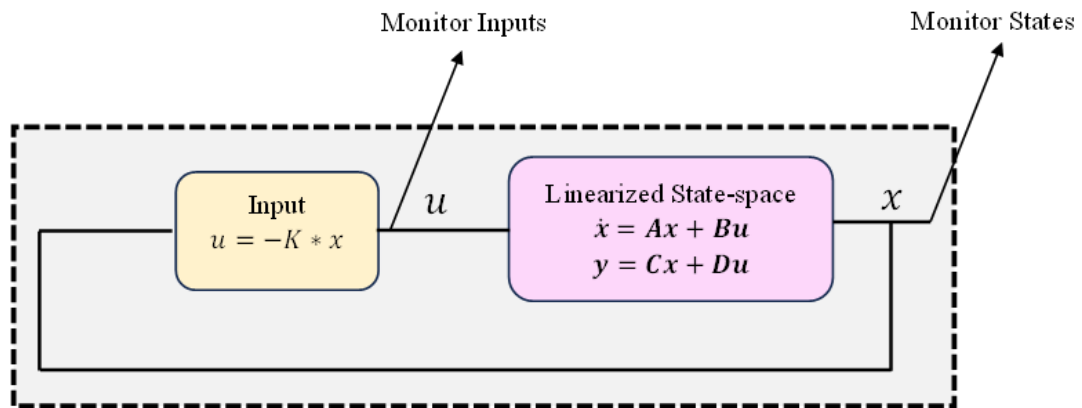
$$A'M + MA - MBR^{-1}B'M + Q = 0 \quad (3.44)$$

3.7 Control Simulation

The control gain (K) is then input to the predefined linearized state-space of robot's system in MATLAB Simulink. Inputs and Outputs of the system are monitored and plotted after the simulation ended. The schematic to monitor states and inputs is presented as shown in Figure 3.16.

Figure 3.16

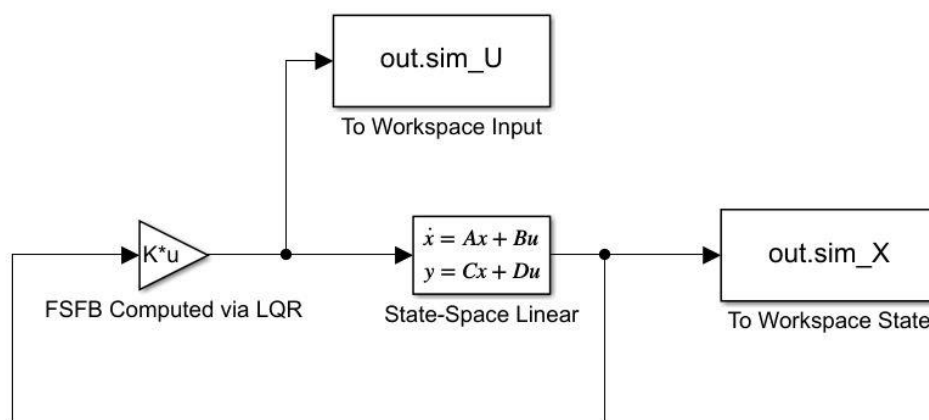
Simulation Schematics



To set the simulation in MATLAB Simulink, the Simulink model as shown in Figure 3.17 is created with closed-loop linear controller to state space. The monitored states and inputs are shared into the workspace of the program to be plotted in MATLAB plotter.

Figure 3.17

Simulink Model of the Tetrahedron Robot



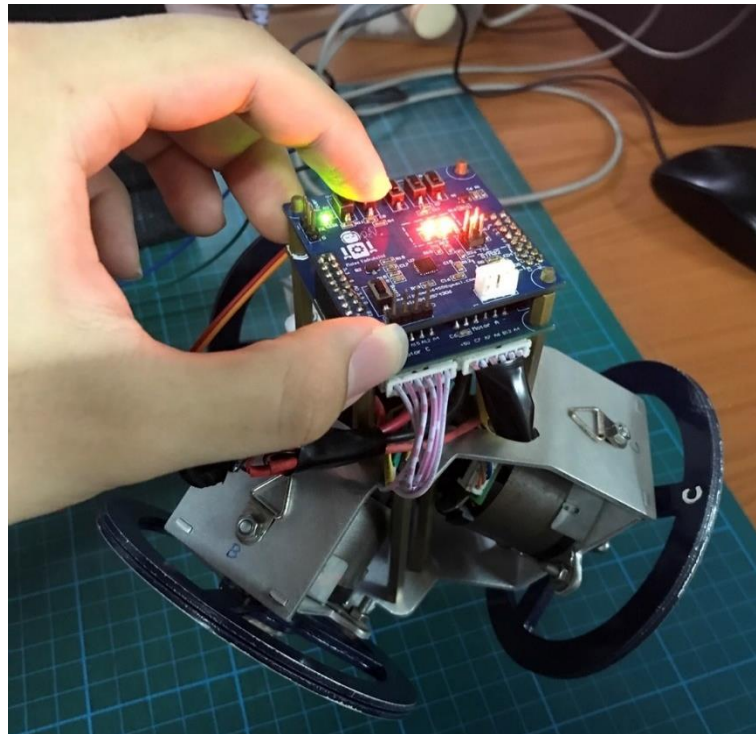
3.8 Control Experiment

The control experiment will use the actual robot to monitor the orientation of the system. After connecting the power supply, manual initial hand upright orientation as shown in Figure 3.18 is done to mark the reference point for the robot as zero reference. Another purpose of this manual orientation is to start IMU calibrate and balancing. Then the USB-to-TTL connected between the robot and USB COM port at computer is

used to read serial printout while the robot is balancing. After that, the log file is saved and then visualized with Microsoft EXCEL plotter.

Figure 3.18

Manual Hand Orientation



CHAPTER 4

RESULT AND DISCUSSION

The chapter presents the results on the following: LQR control gain design, simulating the behavior of system with MATLAB's Simulink, and experimenting with the real-world scenario with actual robot.

4.1 Control Gain Design

From the state space equations in (3.38) and (3.39), the moment of inertia of the robot and reaction wheels need to be found.

4.1.1 Moment of Inertia Constant Estimation

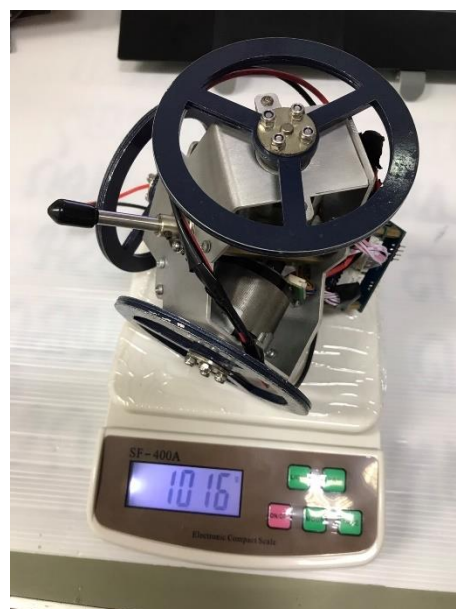
To estimate the moment of inertia of this robot, this thesis uses CAD Software to calculate the systems' moment of inertia for each principal axes. With X, Y, and Z axes in global coordinate in addition to reaction wheels. However, it must know the mass of every component of the assembly file. The mass is found using a digital scale, as shown in Figure 4.1 as an example measurement. The whole robot is also measured to check the correct sum up of all parts.

Figure 4.1

Weighting the Actual Parts



(a) Measure the motor of robot

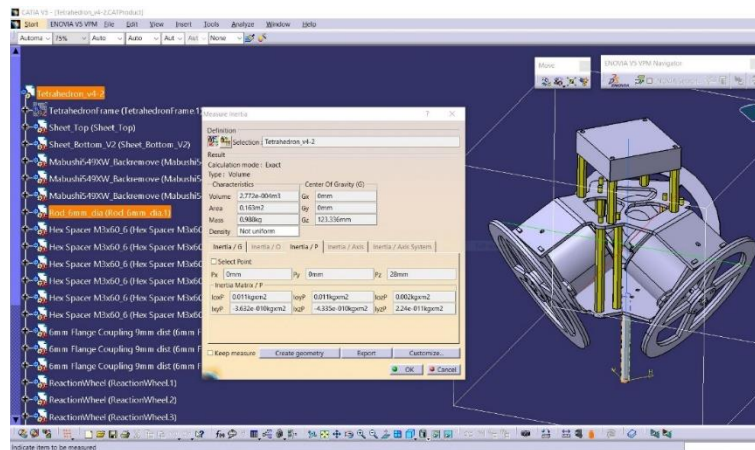


(b) Measure whole robot

The moment of inertia of the parts in CAD software are calculated by the weight and reference distance, in which the weight is not reflected in the actual robot parts. After adding the actual weight into each part, the software can now calculate the desired moment of inertia from any reference point, in this case it is at the bottom robot's shaft tip as shown in Figure 4.2.

Figure 4.2

Get the Robot Moment of Inertia at Reference Point



The reaction wheel's moment of inertia is achieved in the same way. The other inertia parameters for state-space can be obtained, including the robot's weight and the distance from its center of mass to the pivot point O . They are displayed in Table 4.1.

Figure 4.3

Get the Reaction Wheel Moment of Inertia

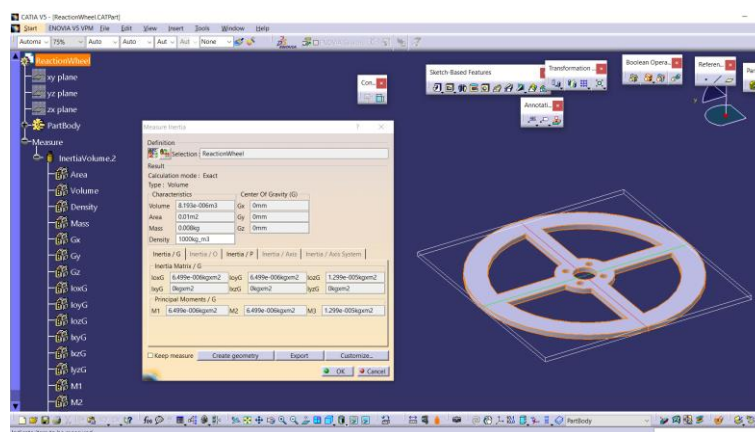


Table 4.1*Inertia Constants*

| Constant Parameter | Value |
|--------------------|---|
| I_{o_x} | $10,646,859.14 \times 10^{-9} \text{ kg/m}^2$ |
| I_{o_y} | $10,646,859.21 \times 10^{-9} \text{ kg/m}^2$ |
| I_{o_z} | $1,785,497.82 \times 10^{-9} \text{ kg/m}^2$ |
| I_w | $73,101.81 \times 10^{-9} \text{ kg/m}^2$ |
| m | $1015.10 \times 10^{-3} \text{ kg}$ |
| l | $123.34 \times 10^{-3} \text{ m}$ |
| g | 9.81 m/s^2 |

Next, all inertia constants are entered into the state-space of equations (3.38) and (3.39) using the parameters from Table 4.1. The following displays A and B matrices in numeric form.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 115.35 & 0 & 0 & 0 & 0 & 0 & 0 & 148.82 & 0 & -74.41 & 0 & -74.41 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 115.35 & 0 & 0 & 0 & 0 & 0 & 0 & 128.88 & 0 & -128.88 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -512.35 & 0 & -512.35 & 0 & -512.35 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -25028 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -25028 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -25028 \end{bmatrix} \quad (4.1)$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ -31.15 & 15.57 & 15.57 \\ 0 & 0 & 0 \\ 0 & -26.97 & 26.97 \\ 0 & 0 & 0 \\ 107.23 & 107.23 & 107.23 \\ 0 & 0 & 0 \\ 5238.03 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 5238.03 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 5238.03 \end{bmatrix} \quad (4.2)$$

At first, the state-weighted matrix Q and input-weighted matrix R from section 3.6 of the LQR controller are necessary. The initial state-weighted and input-weighted matrices used for simulation are shown in equations (4.3) and (4.4). Because every state is equally important to be stable, each state's weight value is equal to one. For LQR to prioritize minimizing all the control effort uniformly in the system, input-weighted matrix (R) are set to one.

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

The control gain K from LQR is then obtained from the MATLAB build-in function with given parameters: A, B, Q and R .

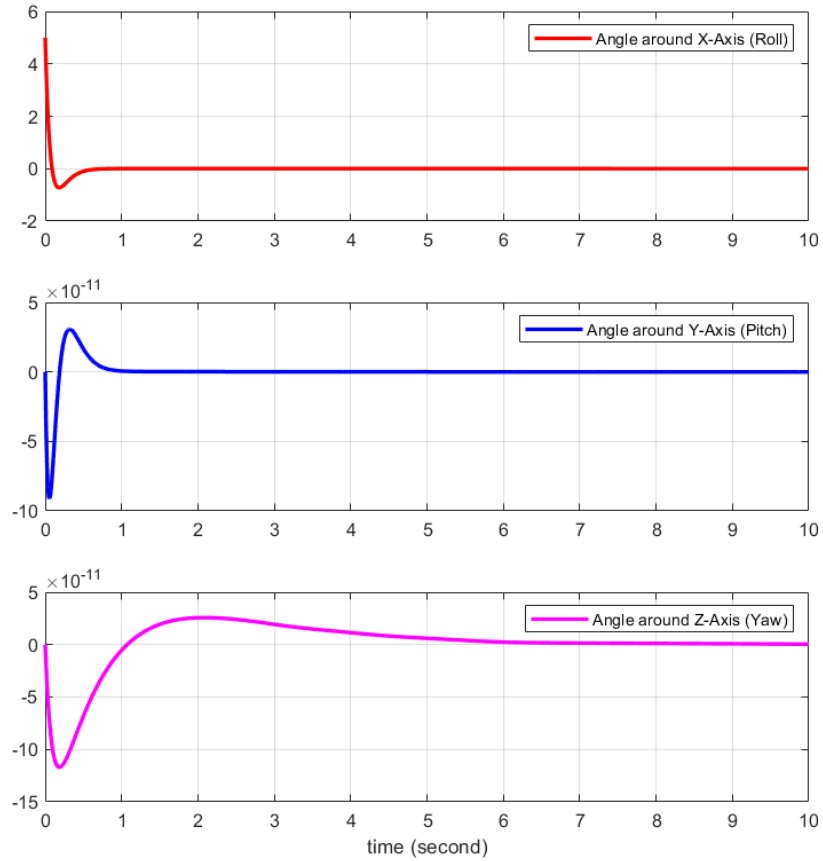
$$K = \begin{bmatrix} -11986 & 5992.8 & 5992.8 \\ -1116.0 & 557.99 & 557.99 \\ -1.274 \cdot 10^{-6} & -10380 & 10380 \\ -1.186 \cdot 10^{-7} & -966.47 & 966.47 \\ 1038.3 & 1038.3 & 1038.3 \\ 164.17 & 164.17 & 164.17 \\ -1.0 & -5.104 \cdot 10^{-10} & -3.162 \cdot 10^{-11} \\ -9.8894 & -0.0433 & -0.0433 \\ -3.858 \cdot 10^{-10} & -1.0 & -1.396 \cdot 10^{-10} \\ -0.0433 & -9.8894 & -0.0433 \\ -1.35 \cdot 10^{-10} & -3.101 \cdot 10^{-10} & -1.0 \\ -0.0433 & -0.0433 & -9.8894 \end{bmatrix} \quad (4.5)$$

4.2 Simulation

The control gain from LQR method from equation (4.5) was then used in the simulation to check the performance of control gain in MATLAB Simulink. To show the performance of the system to balance, an interrupt input of 5 degrees in Roll angle is initiated. The results show that for Roll and Pitch directions, they have a very fast settling time at 0.57 and 0.86 seconds. However, the result of this gain shows a very long settling time for Yaw direction at 7.42 seconds as shown in Figure 4.4.

Figure 4.4

Simulation Results Using Initial Control Gain



The reason for this is because all the states have equal importance. The robot cannot balance in yaw unless it can balance in roll and pitch first. Thus, Roll and pitch directions are more important to balance than yaw direction. The modification nevertheless maintains a higher balancing weight in those directions. Further, Minimizing the control effort for each unique control input is given equal weight by LQR. It's not possible to control all states with equal weight. Hence, Q and R are adjusted to be as in equation (4.6) and (4.7).

$$\mathbf{Q} = \begin{bmatrix} 65000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 40000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1200 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 20000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0007 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0009 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0007 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0009 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0007 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0009 \end{bmatrix} \quad (4.6)$$

$$R = \begin{bmatrix} 85 & 0 & 0 \\ 0 & 85 & 0 \\ 0 & 0 & 85 \end{bmatrix} \quad (4.7)$$

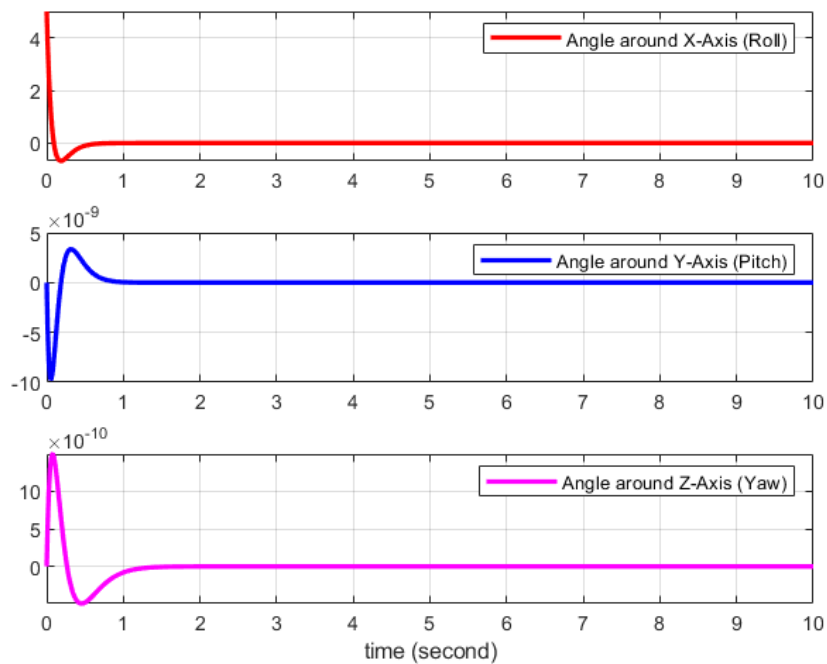
The control gain K as shown in (4.8) from LQR is then obtained from the MATLAB build-in function with new Q and R .

$$K = \begin{bmatrix} -11513 & 5756.6 & 5756.6 \\ -1072.1 & 536.03 & 536.03 \\ -2.849 \cdot 10^{-5} & -9970.3 & 9970.3 \\ -2.653 \cdot 10^{-6} & -928.34 & 928.34 \\ 984.59 & 984.59 & 984.59 \\ 155.69 & 155.69 & 155.69 \\ -0.003 & -2.279 \cdot 10^{-12} & -5.609 \cdot 10^{-12} \\ -9.5570 & -1.282 \cdot 10^{-4} & -1.282 \cdot 10^{-4} \\ -6.318 \cdot 10^{-12} & -0.003 & -1.43 \cdot 10^{-12} \\ -1.282 \cdot 10^{-4} & -9.5570 & -1.282 \cdot 10^{-4} \\ -3.677 \cdot 10^{-12} & -1.653 \cdot 10^{-11} & -0.003 \\ -1.282 \cdot 10^{-4} & -1.282 \cdot 10^{-4} & -9.5570 \end{bmatrix} \quad (4.8)$$

The result for yaw direction is now better. The settling time of roll and pitch remains nearly the same as previous simulation. Yaw's settling time is now approaching steady faster in 1.89 seconds as shown in Figure 4.5.

Figure 4.5

Simulation Results Using Good Simulation Gain



4.3 Experiment

When using the good gain from equation (4.8) directly to the actual robot, the yaw direction is oscillated. So, the gain value in yaw direction is adjusted to make the robot more stable and can return to its initial yaw angle position.

4.3.1 Using Simulation Gain

In the first experiment, the gain in equation (4.8) is used to test in actual robot. Roll displacement angle using good gain from simulation is shown in Figure 4.6. Pitch displacement angle using good gain from simulation is shown in Figure 4.7. In Figure 4.8, yaw displacement angle using good gain from simulation is shown.

Figure 4.6

Roll Displacement Angle of Gain from Simulation

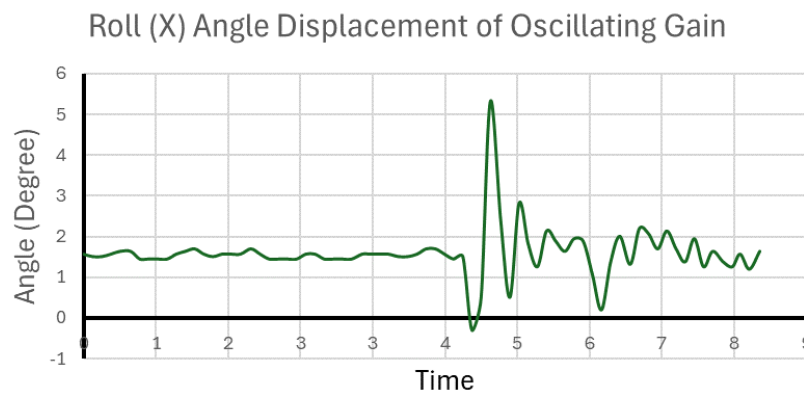


Figure 4.7

Pitch Displacement Angle of Gain from Simulation

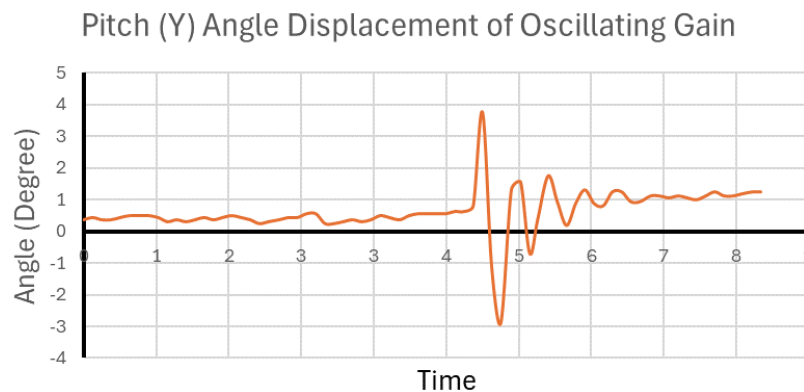
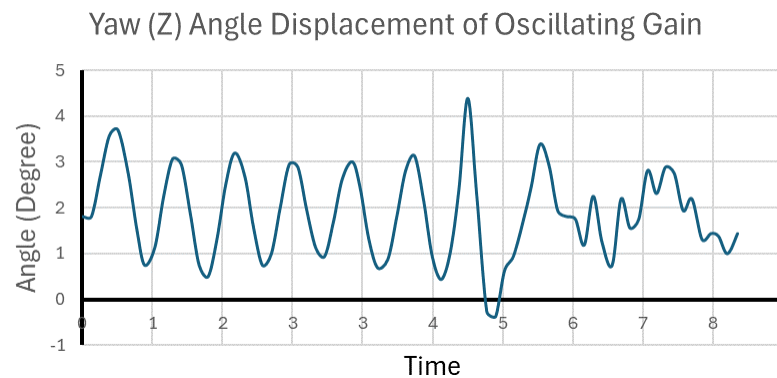


Figure 4.8

Yaw Displacement Angle of Gain from Simulation



Even though the robot can stay upright due to balancing in roll and pitch direction, yaw is not quite balanced. The result shows that the tetrahedron robot experiences an oscillation in yaw direction. The reason can be because of the actual robot system behavior cannot capture within the linearized state matrix A . The parameters of inertia of some parts of the robot, e.g. nuts, bolts, small wires, are less than the digital scale can measured.

Another thing is that, from input matrix in equation (3.39), all control inputs are coupled with all orientations' direction. Thus, the oscillation in yaw direction can cause the other two directions to oscillate too. To solve this unstable behavior, the feedback input gain is adjusted on the yaw direction to make the robot enter stability in yaw direction. Both yaw angle and yaw's angular velocity are increased to compensate for more stability in this direction.

4.3.2 Adjusted Control Gain in the Actual Robot

After adjusting the gain by observing from robot's behavior, the good gain as shown in equation (4.9) can maintain its original orientation while also balanced in upright position.

$$K = \begin{bmatrix} -510.0 & 305.0 & 305.0 \\ -5.0 & 2.5 & 2.5 \\ -1 \cdot 10^{-7} & -412.5 & 412.5 \\ -1 \cdot 10^{-7} & -3.21 & 3.21 \\ 0.81 & 0.81 & 0.81 \\ 0.1 & 0.1 & 0.1 \\ -1 \cdot 10^{-7} & -1 \cdot 10^{-7} & -1 \cdot 10^{-7} \\ -18.0 & 9.0 & 9.0 \\ -1 \cdot 10^{-7} & -1 \cdot 10^{-7} & -1 \cdot 10^{-7} \\ 9.0 & -18.0 & 9.0 \\ -1 \cdot 10^{-7} & -1 \cdot 10^{-7} & -1 \cdot 10^{-7} \\ 9.0 & 9.0 & -18.0 \end{bmatrix} \quad (4.9)$$

The experiment in roll when disturbed shows that the robot can orient back into its original direction with the settling time of 1.22 seconds as shown in Figure 4.9. Pitch displacement angle using adjusted gain is shown in Figure 4.10 with the settling time of 3.42 seconds. Yaw displacement angle using adjusted gain is shown in Figure 4.11 with the settling time of 5.1 seconds.

Figure 4.9

Roll Displacement Angle of Good Actual Gain

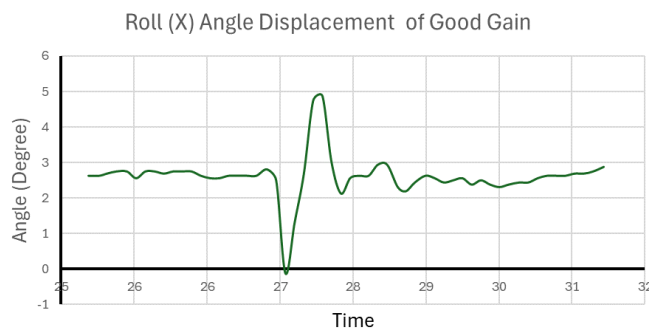


Figure 4.10

Roll Displacement Angle of Good Actual Gain

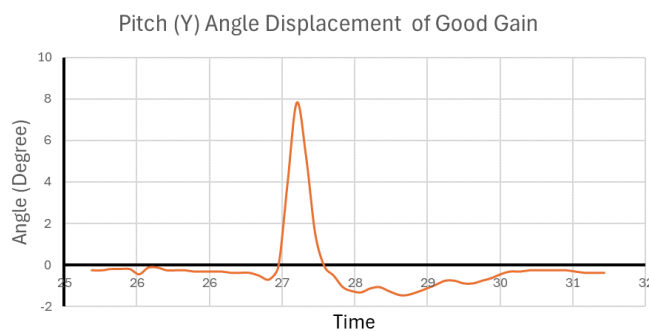
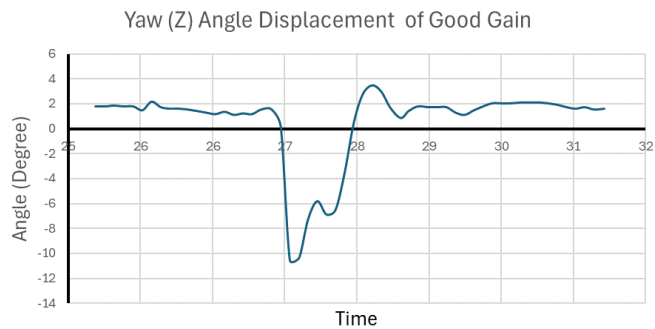


Figure 4.11

Roll Displacement Angle of Good Actual Gain



4.4 Maximum Disturbances Angle

To test out how much disturbance the robot with LQR controller can go back to balanced position, each direction of control is disturbed in 5-degree steps. The step is increased up until the robot loses its balance or the robot is not able to return to initial angle position before getting disturbed.

4.4.1 Maximum Disturbance on Roll Direction

At around 5 degrees disturbance of roll direction as shown in Figure 4.12, the robot can maintain its orientation and return with settling time of 2.1 seconds. At around 10 degrees disturbance of roll direction as shown in Figure 4.13, the robot tries to maintain its orientation, but it overshoots. Then, the robot is unbalanced and collapses.

Figure 4.12

Disturb 5-degree on Roll Direction

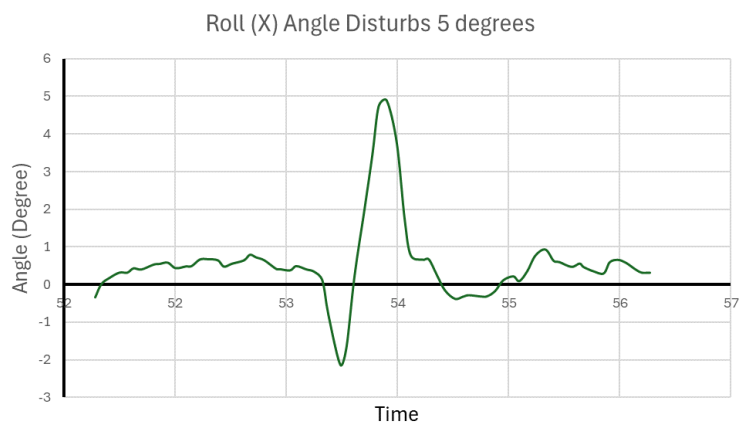
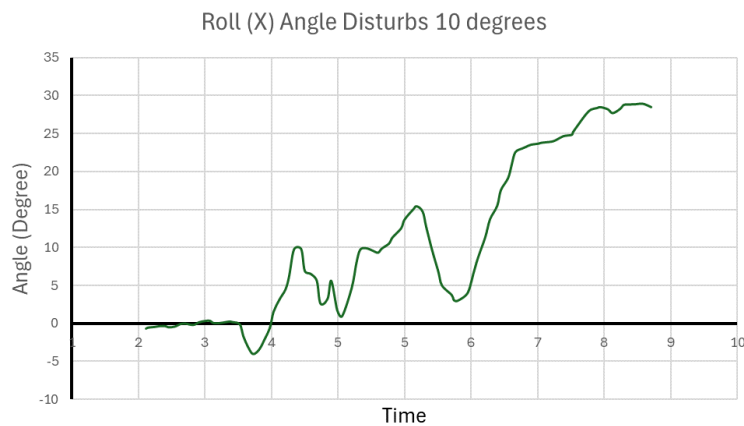


Figure 4.13

Disturb 10-degree Overshoot on Roll Direction



So, the angle between 5 and 10 degrees are tested with 1 degree apart and find out that the maximum degree disturbance in roll direction is around 6.40 degrees with the settling time of 2.5 seconds as shown in Figure 4.14.

Figure 4.14

Max Disturbance of 6 Degree on Roll Direction



4.4.2 Maximum Disturbance on Pitch Direction

At around 5 degrees disturbance of pitch direction as shown in Figure 4.15, the robot can maintain its orientation and return with settling time of 2.1 seconds. At around 8 degrees disturbance of pitch direction as shown in Figure 4.16, the robot tries to maintain its orientation, but it overshoots. Then, the robot is unbalanced and collapses at rest with pitch angle around 57 degrees.

Figure 4.15

Disturb 5-degree on Pitch Direction

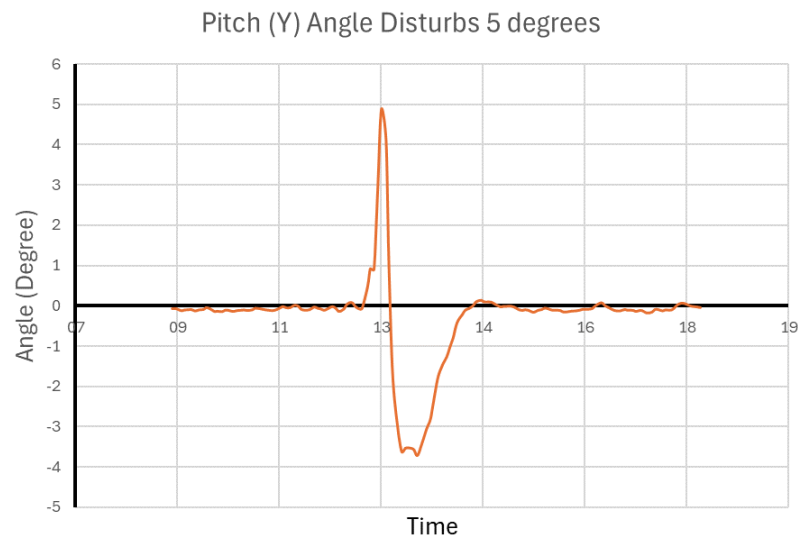
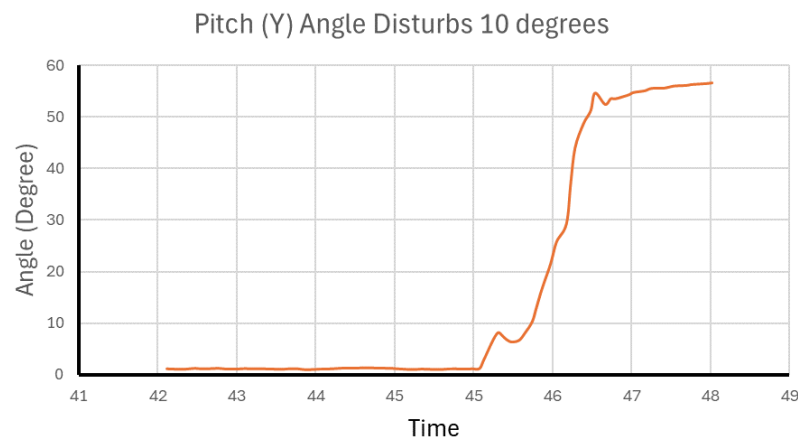


Figure 4.16

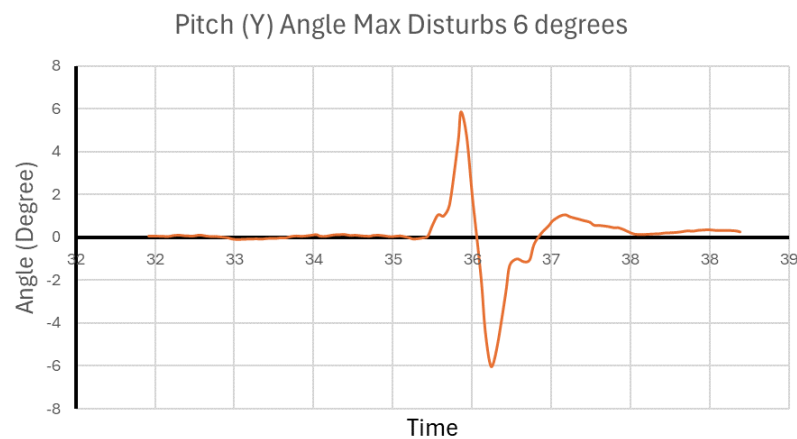
Disturb 10-degree Overshoot on Pitch Direction



So, the pitch angle between 5 and 8 degrees are tested with 1 degree apart and find out that the maximum degree disturbance in pitch direction is around 6 degrees with the settling time of 2.62 seconds as shown in Figure 4.17.

Figure 4.17

Max Disturbance of 6 Degree on Pitch Direction



4.4.3 Maximum Disturbance on Yaw Direction

At around 5 degrees disturbance of yaw direction as shown in Figure 4.18, the robot can maintain its orientation and return with settling time of 6.3 seconds. At around 10 degrees disturbance of yaw direction as shown in Figure 4.19, the tetrahedron robot can still secure its heading and manages to return with settling time of 6.8 seconds, longer than previous 5 degrees.

Figure 4.18

Disturb 5-degree on Yaw Direction

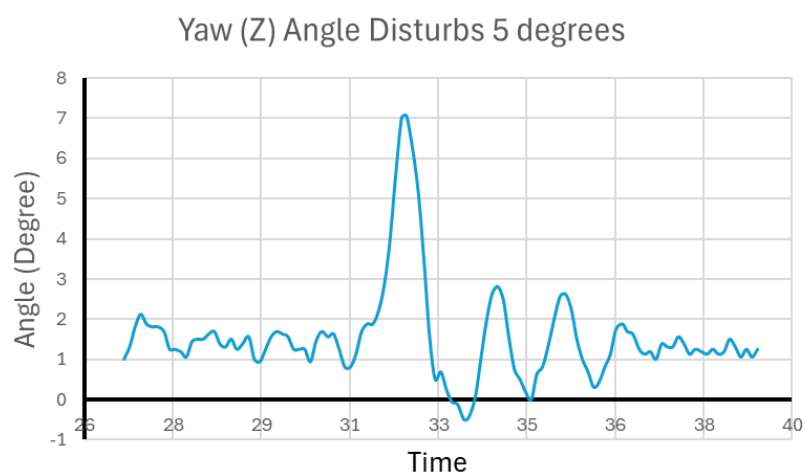
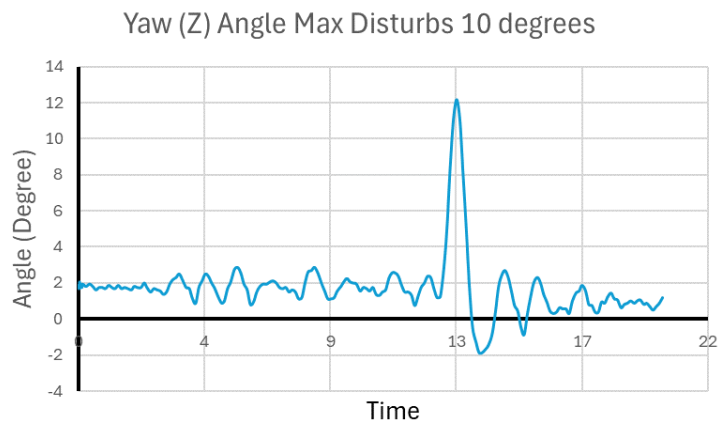


Figure 4.19

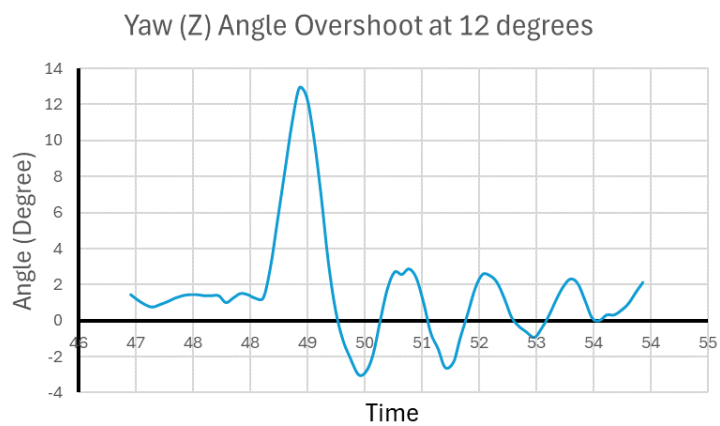
Max Disturbance of 10 Degree on Yaw Direction



When the yaw disturbance angle reaches 12 degrees as shown in Figure 4.20, the robot starts to rotate back and forth before collapses. This marks the limit of disturbance in yaw direction at 10 degrees.

Figure 4.20

Oscillation on Yaw Direction When Disturbed at 12 Degrees



CHAPTER 5

CONCLUSIONS

5.1 Conclusion

This thesis “Design and Control of a Tetrahedron Robot Using Reaction Wheels” has two study objectives as follow:

1. To develop a new mechanism for tetrahedron robot using three reaction wheels balancing technique.
2. To design a control algorithm that fits with the newly proposed tetrahedron robot with reaction wheels.

Which will be summarized with respect to each objective.

5.1.1 New Tetrahedron Robot Using Three Reaction Wheels

From designing the Tetrahedron robot by orienting the motor into three principal axes of the tetrahedron robot, this robot can maintain its roll, pitch, and yaw orientations. The positioning of other components is also placed close to robot center of mass to minimize the torque generated from reaction wheels to maintain its balance and heading.

The Dynamic model of tetrahedron robot is obtained by use of Lagrange method. It uses the system energy to governing dynamic model equations. Thus, it doesn't need to deal with complicated vector direction of force as in Newton's method. The obtained dynamic models can then be used to design control algorithms to balance and maintain its orientation.

The actual tetrahedron robot structure is designed to be light weight and makes its center of mass aligned in the middle of the body. This helps the robot to balance itself in the unstable equilibrium point.

5.1.2 Control Algorithm Fitted with Tetrahedron Robot

This tetrahedron robot could not perform desired behavior without its controller. The robot performs balancing using a linear quadratic regulator (LQR) as linear control method. LQR is considered because of its ability to rearrange states and inputs priority. Focusing on angle and angular velocity in roll, Pitch, and yaw direction, the robot has efficient control with the disturbances. To minimize the disruption and ensure fast

processing of actual robots, counters in microcontroller is used. The counter maintains constant cycle time for the robot to adjust the voltage of motor to compensate with its orientation.

With three reaction wheels balancing technique, it can maintain its upright position related to roll and pitch, and its heading related to yaw. Although initially appearing to be somewhat complex, the control rule is relatively compact and can balance itself into reference points of both roll, pitch, and yaw directions. In addition, the robot is also able to maintain its heading in yaw direction with an acceptable error range. In roll direction, the maximum disturbance angle is 6.4 degrees with the settling time of 2.5 seconds. For pitch direction, the robot can return with 2.62 seconds settling time from the maximum disturbance angle of 6 degrees. The maximum disturbance of yaw direction is 10 degrees with the settling time of 7.8 seconds.

The tetrahedron is designed in such a way that the reaction wheels are aligned to contribute torque to roll, pitch, and yaw. This makes the robot able to balance roll and pitch, while also able to maintain its heading in yaw direction.

5.2 Recommendations

1. The robot is suggested to have a built-in battery to reduce complications of the power wires connected to the robot.
2. The higher quality of materials used to create body frame may be considered to reduce the weight and increase strength to withstand the impact when robot is collapsed from experiment.

REFERENCES

- Akhtaruzzaman, M., & Shafie, A. A. (2010). Modeling and control of a rotary inverted pendulum using various methods, comparative assessment and result analysis. *2010 IEEE International Conference on Mechatronics and Automation, ICMA 2010, April 2014*, 1342–1347. <https://doi.org/10.1109/ICMA.2010.5589450>
- Bjerke, E., & Perhsson, B. (2016). *Development of a Nonlinear Mechatronic Cube* CHALMERS UNIVERSITY OF TECHNOLOGY. <https://publications.lib.chalmers.se/records/fulltext/233543/233543.pdf>
- Bobrow, F., Angelico, B. A., Martins, F. P. R., & Da Silva, P. S. P. (2021a). The Cubli: Modeling and Nonlinear Attitude Control Utilizing Quaternions. *IEEE Access*, 9, 122425–122442. <https://doi.org/10.1109/ACCESS.2021.3108426>
- Bobrow, F., Angelico, B. A., Martins, F. P. R., & Da Silva, P. S. P. (2021b). The Cubli: Modeling and Nonlinear Attitude Control Utilizing Quaternions. *IEEE Access*, 9, 122425–122442. <https://doi.org/10.1109/ACCESS.2021.3108426>
- Chen, Z., Ruan, X., & Li, Y. (2017). Dynamic modeling of a cubical robot balancing on its corner. *MATEC Web of Conferences*, 139, 1–5. <https://doi.org/10.1051/mateconf/201713900067>
- Chouhan, K. (2020). *Control of a Rotary Inverted Pendulum using first order and higher order Sliding Mode Controller*. July, 1–44.
- Gajamohan, M., Merz, M., Thommen, I., & D’Andrea, R. (2012). The Cubli: a cube that can jump up, balance, and “walk.” *IEEE/RSJ International Conference on Intelligent Robots and Systems*. https://ethz.ch/content/dam/ethz/special-interest/mavt/dynamic-systems-n-control/idsc-dam/Research_DAndrea/Cubli/Cubli_IROS2012.pdf
- Gajamohan, M., Muehlebach, M., Widmer, T., & D’Andrea, R. (2013). The Cubli: A reaction wheel based 3D inverted pendulum. *European Control Conference (ECC)*. <https://doi.org/10.23919/ecc.2013.6669562>
- Graichen, K., Treuer, M., & Zeitz, M. (2007a). Swing-up of the double pendulum on a cart by feedforward and feedback control with experimental validation. *Automatica*, 43(1), 63–71. <https://doi.org/10.1016/j.automatica.2006.07.023>
- Graichen, K., Treuer, M., & Zeitz, M. (2007b). Swing-up of the double pendulum on a cart by feedforward and feedback control with experimental validation.

- Automatica*, 43(1), 63–71. <https://doi.org/10.1016/j.automatica.2006.07.023>
- Hofer, M., Muehlebach, M., & D'Andrea, R. (2023). The One-Wheel Cubli: A 3D inverted pendulum that can balance with a single reaction wheel. *Elsevier*, 91(January). <https://doi.org/10.1016/j.mechatronics.2023.102965>
- KARATAŞ, S. (2006). *LEO Satellites: Dynamic Modelling, Simulations and some Nonlinear Attitude Control Techniques* (Issue April) MIDDLE EAST TECHNICAL UNIVERSITY. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=7f600b4b047bd77a8aac51f13944c1bf32d5278>
- Kök, I. (2012). *Comparison and Analysis of Attitude Control Systems of a Satellite Using Reaction Wheel Actuators* University of Würzburg; Lulea University of Technology. <https://www.diva-portal.org/smash/get/diva2:1020537/FULLTEXT02.pdf>
- Kumar, S., Sahay, D., Hegde, S. R., Sandya, S., Jha, A. K., & Mahalingesh, T. C. (2015a). Design and development of 3-axis reaction wheel for STUDSAT-2. *IEEE Aerospace Conference Proceedings, 2015-June*(March). <https://doi.org/10.1109/AERO.2015.7119181>
- Kumar, S., Sahay, D., Hegde, S. R., Sandya, S., Jha, A. K., & Mahalingesh, T. C. (2015b). Design and development of 3-axis reaction wheel for STUDSAT-2. *IEEE Aerospace Conference Proceedings, 2015-June*(March). <https://doi.org/10.1109/AERO.2015.7119181>
- Liao, T. L., Chen, S. J., Chiu, C. C., & Yan, J. J. (2020). Nonlinear dynamics and control of a cube robot. *Mathematics*, 8(10), 1–15. <https://doi.org/10.3390/math8101840>
- Meyer, J., Delson, N., & de Callafon, R. A. (2009a). Design, Modeling and Stabilization of a Moment Exchange Based Inverted Pendulum. In *IFAC Proceedings Volumes* (Vol. 42, Issue 10). IFAC. <https://doi.org/10.3182/20090706-3-fr-2004.00076>
- Meyer, J., Delson, N., & de Callafon, R. A. (2009b). Design, Modeling and Stabilization of a Moment Exchange Based Inverted Pendulum. *IFAC Proceedings Volumes*, 42(10), 462–467. <https://doi.org/10.3182/20090706-3-fr-2004.00076>
- Muehlebach, M., & D'Andrea, R. (2016). Nonlinear Analysis and Control of a Reaction Wheel-Based 3D Inverted Pendulum. *IEEE 52nd Annual Conference on Decision and Control (CDC)*, 25(1), 235–246. <https://doi.org/10.1109/TCST>

.2016.2549266

- Posada, D. (2017). *Design and control of an inertia wheel cube 2D prototype with a reaction wheel* [Universidad de los Andes, Colombia]. https://www.researchgate.net/publication/318433317_Design_and_control_of_an_inertia_wheel_cube_2D_prototype_with_a_reaction_wheel
- Ramberg, A. S., Halim, N. A., & Ahmad, R. R. (2012). A Numerical Comparison of Lagrange and Kane's Methods of an Arm Segment. *International Journal of Modern Physics: Conference Series*, 09, 68–75. <https://doi.org/10.1142/s2010194512005119>
- Ramm, A., & Sjöstedt, M. (2015). *Reaction wheel balanced robot: Design and Sensor Analysis of Inverted Pendulum Robot* [KTH ROYAL INSTITUTE OF TECHNOLOGY]. <http://www.diva-portal.se/smash/get/diva2:916271/FULLTEXT01.pdf>
- Singh, R., Tayal, V. K., & Singh, H. P. (2016). A review on Cubli and non linear control strategy. *1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems, ICPEICES 2016*, 1–5. <https://doi.org/10.1109/ICPEICES.2016.7853425>
- Sukontanakarn, V. (2011). *Control of Rotary Double Inverted Pendulum Using Neural Network Based Adaptive Linear Quadratic Regulator* [Asian Institute of Technology]. <http://ise.ait.ac.th/wp-content/uploads/sites/57/2020/12/CONTROL-OF-ROTARY-DOUBLE-INVERTED-PENDULUM.pdf>
- Tangnararatchakit, W. (2021). *Balancing of 3D Inverted Cube Using Reaction Wheels* (Issue May 2021). Asian Institute of Technology.
- Trimpe, S., & D'Andrea, R. (2010). Accelerometer-based Tilt Estimation of a Rigid Body with only Rotational Degrees of Freedom: BT - 2010 IEEE International Conference on Robotics and Automation (ICRA 2010). *Proceedings of the IEEE International Conference on Robotics and Automation*, 2630–2636. https://idsc.ethz.ch/content/dam/ethz/special-interest/mavt/dynamic-systems-n-control/idsc-dam/Research_DAndrea/Balancing_Cube/ICRA10_1597_web.pdf
- Vadrukchid, M. (2022). *Development and Control of the Stick Robot Using Three Reaction Wheels (final)* (Issue May 2022). Asian Institute of Technology.