BALANCING OF 3D INVERTED CUBE USING REACTION WHEELS

by

Warat Tangnararatchakit

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

Examination Committee:	Prof. Manukid Parnichkun (Chairperson)
	Dr. Mongkol Ekpanyapong
	Dr. Pisut Koomsap

Nationality:	Thai
Previous Degree:	Bachelor of Engineering in Automotive Engineering Chulalongkorn University Thailand
Scholarship Donor:	His Majesty the King's Scholarships (Thailand)

Asian Institute of Technology
School of Engineering and Technology
Thailand
May 2021

AUTHOR'S DECLARATION

I, Warat Tangnararatchakit, 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 are 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:

Name: Warat Tangnararatchakit

Signature:

ACKNOWLEDGMENTS

I want to express my deep and sincere gratitude to my research advisor, Prof. Manukid Parnichkun, professor of Industrial System Engineering, Asian Institute of Technology, Thailand, for allowing me to do research and provide additional guidance throughout this research. His knowledge, vision, and insight have deeply motivated me.

I am incredibly grateful to the royal Thai government for providing me this excellent opportunity to study at Asian Institute of Technology with the scholarship support of His Majesty the King's Scholarships.

ABSTRACT

Reinforcement learning has become a powerful tool for tackling various problems in the past few years. It is well-establish that this method can solve complex problems in many fields of study. This thesis presents an aim to explore an alternate approach to balance the inverted cube structure. To test the hypothesis that reinforcement learning can balance a cube structure on its corner using three reaction wheels as actuators. Simultaneously, the control algorithm developed traditionally using the LQR method and alternatively using reinforcement learning. Additionally, the experiment result was compared to evaluate the reinforcement base control algorithm's performance and the LQR algorithm. The result showed that reinforcement learning could find good controller gain value to balance the inverted cube on its destinated corner. These results suggested that reinforcement learning can be an alternative method to solve the balance control problem.

CONTENTS

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

g	= Earth gravitational acceleration
I	= Moment of inertia of the inverted cube with corner as pivot
l	= Distance from the pivot to center of mass of inverted cube
m	= Mass of inverted cube
ω	= Angular velocity of inverted cube
θ	= Angular position of inverted cube
α	= Angular acceleration of inverted cube
π	= Probability generated by actor network
a_t	= Action taken by network at time step t
s_t	= State Matrix at time step t
G	= Returned reward from simulation to actor and critic network
V	= Predicted reward value from critic network

CHAPTER 1
INTRODUCTION

1.1 Background of the Study

The inverted pendulum has been a fundamental problem in control theory because of its unbalanced nature in an upward position. This system has been used as a basic benchmark of many control algorithms. Other configurations of this system had been extensively researched for many decades, such as reaction wheel pendulum, bi-axial pendulum, and cart inverted pendulum. One interesting configuration of this system was introduced in 2013. The 3D cube was a combination of 3 axes inverted pendulum, balanced on its corner using a traditional control method.

AI has been used as a tool to find a solution to the problem for a long time. One of the topics that many researchers had been exploring is AI as a solution for a game like a game of chess and a maze. Markov Decision Process is one of many AI algorithms. This thinks the process focuses on the quality of each possible move for each state of the system. However, this process requires the known quality table of that system which can be very hard to acquire. Reinforcement learning improves that decision process because it allows the AI system to learn a quality table for any system by controlling the quality function. Combined with the improvement in the current-day computer's computing power, reinforcement learning becomes a potent tool to tackle a complex control system.

Many research pieces have explored reinforcement learning in many applications such as board games, navigation, robotic control. By nature of reinforcement learning, it has the potential to discover another strategy that might be overlooked by a traditional problem-solving method. This result-based problem solving allows a researcher to skip some of the complexity in modeling the system. Additionally, reinforcement learning can adapt to the user's requirement, enabling it to self-correct the system while discovering its operating condition.

Reinforcement learning is an area of machine learning, based on the Markov Decision Process, which explores and develops an optimal solution to maximize the accumulated reward from each state of the problem or system. By nature of how reinforcement learning develops the answer, it is not required to know the problem's exact model. This allows reinforcement learning to be a powerful candidate in controlling a complex system. Few research types explored the potential in finding more optimal control of the inverted pendulum system than the conventional method.

In balancing the object on its corner using three perpendicular reaction wheels as a control actuator, the combination of torque or momentum can counteract the cube object's gravitational force. The model of the system can be a very complex combination of multiple states of the system. With reinforcement learning, it is possible to bypass all modeling the system and generate a control algorithm to balance the corner.

1.2 Statement of the Problem

The complex non-linear system has been the main problem for the control engineer to solve for a very long time. While different systems have difficulty reaching optimal control, one of the most challenging tasks to analyze any system is creating a precise system model. After acquiring a system model, many control algorithms can be selected to suit the system's need to get exact and robust control. Simultaneously, it is possible to use traditional control laws like PID, LQR, or adaptive control to get a pleasing result. However, a system's model's requirement to create a control algorithm can be very difficult to solve, especially when the system involves many degrees of freedom. Reinforcement learning offers another solution to control the problem by allowing modeless control possible.

Reinforcement learning can be used in control problems by allowing engineers with another method to solve control problems without the requirement of knowing the exact model of the system. By its ability to optimize its parameter from the reward function, reinforcement learning can be trained by presenting the model's states to the system and adjusting itself to maximize the reward. These properties can be utilized to achieve optimum control of the system and become additional options to solve other control problems. A further advantage of using the reinforcement learning method is that it is possible to create a control algorithm that adapts its control parameter according to the change of model properties. While reinforcement learning shows many properties in solving a control problem, there is not much research that used reinforcement learning to solve the balance control problem. This research aims to explore its potential as an additional candidate to balance multiple degrees of freedom system.

The inverted cube structure becomes an attractive candidate to explore the reinforcement learning approach to create a control algorithm because of its multiple degrees of freedom and non-linear nature when standing on its corner and complex configuration of reaction wheels. This system will be a reasonable benchmark in explore reinforcement learning potential in solving balance control problems.

1.3 Can Reinforcement Learning-Based Control be Alternative Method in Balance Control Problem?

Reinforcement learning-based control method was tested as an alternative way to calculate controller gain besides the LQR method

1. Can reinforcement learning-based method become an option when tackling balance control problem?
2. What is the performance difference between reinforcement learning-based and LQR methods?

1.4 Objectives of the Study

The research's fundamental intent is to develop control solutions from reinforcement learning methods that can keep cube balance on its corner. The following objectives are the main objectives of this research.

1. To achieve on corner balance of inverted cube structure with reaction wheels.
2. To compare control performance of the LQR algorithm and reinforcement learning-based algorithm.

1.5 Scope of the Study and Limitation.

The response from the reinforcement learning base algorithm will be compared with an LQR control algorithm. While balance at a designed position, the cube can reject external disturbances and return to a created state position.

CHAPTER 2
LITERATURE REVIEW

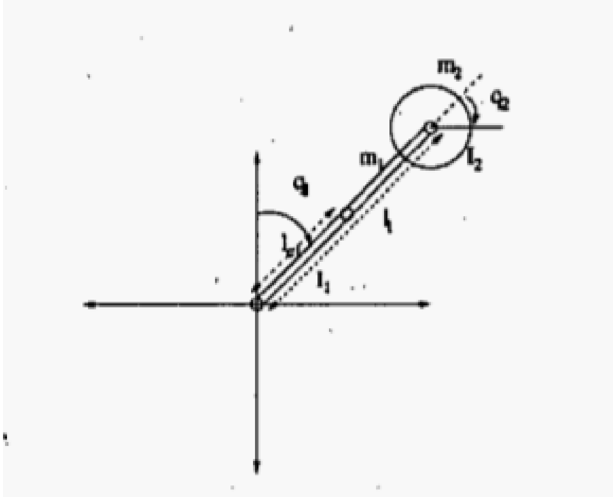
The study pursuit a methodology to use reinforcement learning to generate an optimal control strategy for balancing the cube on its corner using three perpendicular reaction wheels. Thereupon the literature review is focused on the primary system of an inverted pendulum, the control method that had been used, and the implementation of reinforcement learning in balancing control.

2.1 Inverted Pendulum

In control theory, the inverted pendulum system has been the central part of the control system design problem for a very long time. While the primary physical nature remains the same, researchers use different means to control the system. The inverted rotary pendulum is controlled by the rotation of the pendulum base and cart pendulum balanced by the moving cart's sliding motion. **B. Bapiraju, K. N. Srinivas, P. P. Kumar, and L. Behera (2004)** had studied another configuration of the inverted pendulum that was controlled by the reaction wheel at the endpoint of the pendulum. Their work concluded the algorithm that can balance the inverted pendulum at the proper top position using linearization of the pendulum model and fuzzy control algorithm.

Figure 2.1

Free Body Diagram of Reaction Wheel Inverted Pendulum



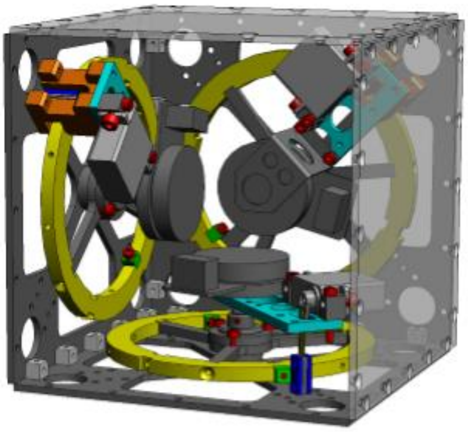
A more complex system of the inverted pendulum that utilizes the reaction wheel as controlled output to the system was researched**. L. H. Chang and A. C. Lee (2007)** created an inverted pendulum in the form of 2 DOF cart-based robots, which used 3 phase control strategy to bring the inverted pendulum the maximum upright position.

2.2 On corner balance cube robot (Cubli)

M. Gajamohan, M. Merz, I. Thommen, and R. D'Andrea (2013) introduced another complex configuration of this system in the form of cube shape robot (Cubli) that can balance and jump from resting position using three axes of the reaction wheel. Their system controlled the robot using 2 phase control which was divided into the jumping mode and balanced mode, in balance control their used linearization of cube model to transform the non-linear system to a more straightforward linear system. In 2017 M. Gajamohan, M. Merz, I. Thommen, and R. D'Andrea published another paper regarding jump up a cube robot's function. The jump-up process was achieved by impulse generated by breaking of high-speed rotation of reaction wheel. To accomplish the desire jumping trajectory, these researchers use the combination of the system model and learning algorithm to make robots learn the best speed that the reaction wheel needed to follow the target trajectory from resting position to the cube's on-corner position.

Figure 2.2

3D Model of Cubli

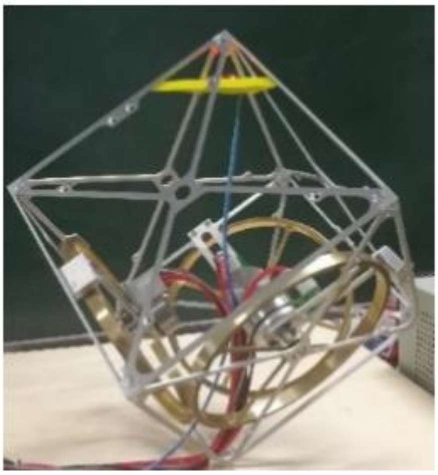


After the initial introduction of the Cube robot from previous research, **Rupam Singh, Vijay Kumar Tayal, and Hemandar Pal Singh (2016)** conducted research reviewing the additional control algorithm and mechanism of Cubli's structure. Finally, they proposed modern adaptive control as the most suitable algorithm and other considerations of Cubli's design from its requirement in dealing with an impulse from reaction wheels.

Zhigang Chen, Xiaogang Ruan, and Yuan Li (2017) proposed another method of analyzing this inverted cube robot using the Lagrangian approach to solve energy functions. This research's simulation result showed a cube's response in zero forced scenario and torque from reaction wheels. This dynamic model provided a different approach to develop a control algorithm to balance this cube robot on its corner.

Figure 2.3

Zhigang Chen, Xiaogang Ruan, and Yuan Li's Prototype of Cubli



2.3 Control Method

The main component of the inverted pendulum system to achieve an upright position is its control law. The selection of control law has direct consequences on the performance of an inverted pendulum system.

2.3.1 PID

PID controllers are one of the most common controlling algorithms used in controlling objectives. PID generates a signal from an error from the difference between the setpoint and measured value of the controlled system. While it is more suitable for a linear system, **WANG Jia-Jun (2015)** used a double PID controller to control the inverted pendulum's position by generating a signal from measure input of inverted pendulum angle and place of the cart. This allows control of the cart position and the balance of the inverted pendulum at the same time.

Further research had been conducted to improve the performance of the PID controller for an inverted pendulum system. **Sankalp Paliwal (2017)** researched the fractional order PID controller, a general form of PID in the Laplace domain. The result showed that fractional order PID has a lower overshot compare to the traditional PID controller.

2.3.2 Fuzzy Controller

Fuzzy control theory is a controlled algorithm base on the decision-making process using a fuzzy set of input states and output action. This control simulates the human brain's decision-making to achieve control of the system**. Luo Hong-yu and Fang Jian (2014)** developed a fuzzy control rule to control inverted pendulum, and simulation results showed better performance than regular PID controller and more robust to change of properties of the system.

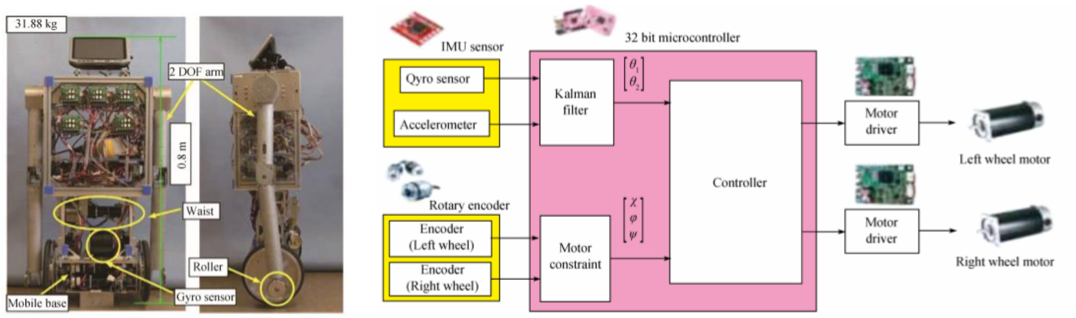
2.3.3 Linear-Quadratic Regulator (LQR)

Linear-Quadratic Regulator is another controller design method that tries to minimize the cost function that depends on both the state weighting matrix and the control weighting matrix. Research of **Ramashis Banerjee, Naiwrita Dey, Ujjwal Mondal, and Bonhihotri Hazra (2018)** showed the double inverted pendulum model's analysis well as the linear feedback gain matrix that make double inverted pendulum stable in simulation using the LQR method.

Magdi S. Mahmoud and Mohammad T. Nasir (2017) researched the LQR algorithm to control wheeled inverted pendulum robot to remain at the upright position while performing other tasks with two different robot arms. This research shows the LQR algorithm's ability to control an inverted pendulum system while dealing with the uncertainty of robot arms motion, which acts as an additional disturbance.

Figure 2.4

Wheeled Inverted Pendulum Robot and Control Diagram.



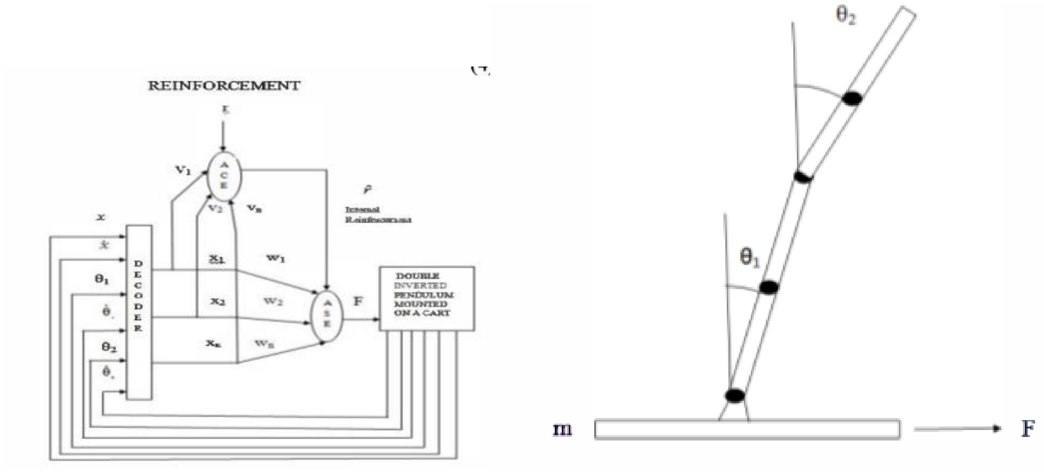
2.3.4 Reinforcement Learning

Richard S. Sutton, Andrew G. Barto, and Ronald J. Williams (1992) introduced reinforcement learning in the control perspective as a direct adaptive optimal control by comparing reinforcement learning as animal learning. The idea of reaching a satisfying conclusion by adjusting action selection to estimate the long-term consequences of action comparable to optimal control using the system's current state was introduced. Additionally, the authors also proposed reinforcement learning as an alternative solution for the non-linear system, reducing the cost of developing control algorithms compared to indirect system estimators.

Within the computer's increase in computing power, reinforcement learning has become a powerful solution to tackle complex non-linear systems like an inverted pendulum. Instead of using multiple phase control algorithms to bring the pendulum from a downward position to an upright position, reinforcement learning allows the computer to explore the best solution for each of the states that inverted pendulum. **Sudhir Raj (2016)** utilized this powerful tool to achieve steady control of double inverted pendulum and better results than the LQR control algorithm.

Figure 2.5

Sudhir Raj's Reinforcement-based Control Block Diagram and Double Inverted Pendulum.



Yue Chao, Liu Yongxin did another research, and Wang Linglin (2018) used reinforcement learning to achieve steady-state balance and spin up of reaction wheel inverted pendulum. This research utilized online and offline learning methods to reduce a neural network's programming and computing time. The result showed that after 300 trials, the network could control the cart inverted pendulum to the upright position.

Figure 2.6

Simulink Diagram of Real-time Swing-up of Reinforcement Learning

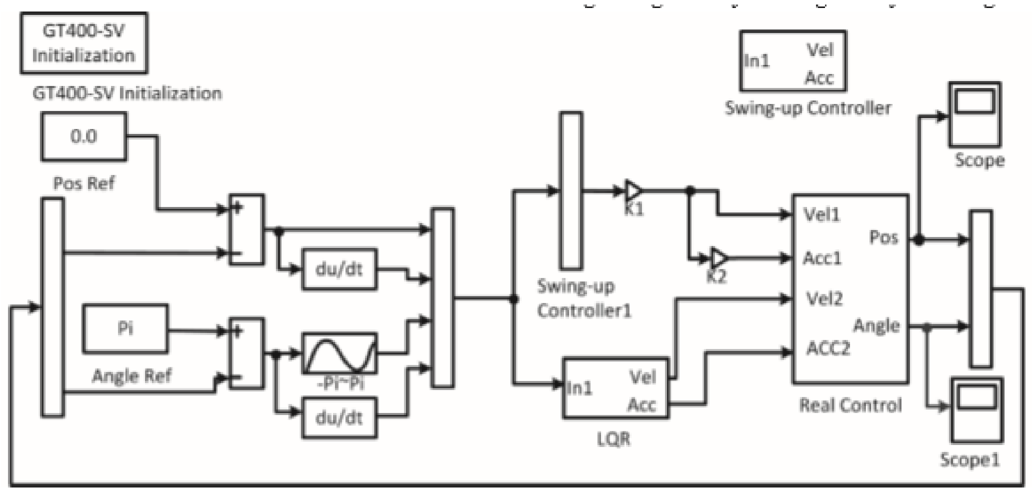
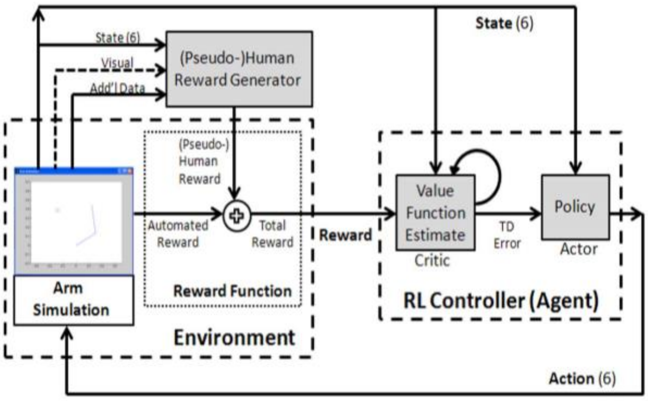


Fig. 6.Simulink module diagrams of real-time swinging up of reinforcement learning

Aside from the inverted pendulum, **Kathleen M. Jagodnik, Philip S. Thomas, Antonie J. van den Bogert, Michael S. Branicky, and Robert F. Kirsch (2017)** used reinforcement learning to develop adaptive control of computer-generated arm using the human reward. Their research trained arm movement simulation to replicate picking up and moving an object to designed locations. By comparing both humans generate bonus and pseudo-human developed reward, the researchers evaluated the system's training time. However, both methods result in functional electrical stimulation that can recreate human arm motion in terms of performance. This showed that reinforcement learning could adapt according to trainers' satisfaction, in this case, humans.

Figure 2.7

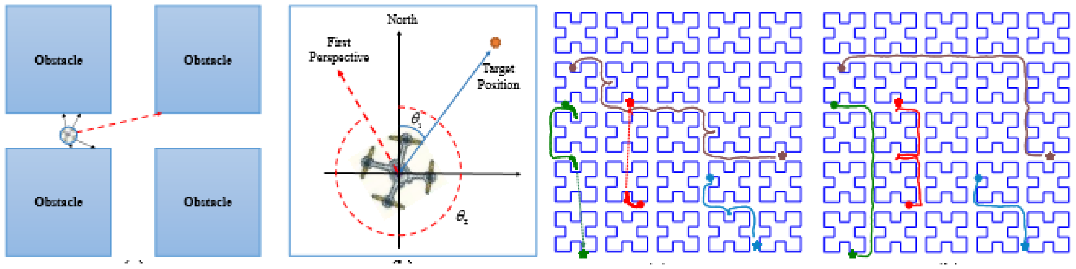
*Block Diagram of the Actor-critic Reinforcement Learning***.**



Chao Wang, Jian Wang, Xudong Zhang, and Xiao Zhang (2017) researched reinforcement learning ability to navigate uncrewed aerial vehicles. Using the actor-critic approach, deep reinforcement learning shows the ability to navigate complex environments and avoid obstacles without predetermining the territories' map. While simulation shows promising results, the researcher stated additional remake in real-world tasks regarding sensor requirement and unobservable noise in a natural environment.

Figure 2.8

Aerial Vehicles Obstacle Detection and Navigation Using Reinforcement learning



From previous research, reinforcement learning has many applications it can be applied to and can learn optimal solutions in that problem or satisfy the trainer's need.

2.4 Chapter Summary

This chapter contains a literature review of previous research concerning balancing the inverted pendulum system with multiple configurations and different ways to balance it. Previous research also explored reinforcement learning algorithms in many fields of study. Previous research had indicated that reinforcement learning is an alternative tool for tackling a complex problem

CHAPTER 3
METHODOLOGY

This research focused on implementing a reinforcement learning-based algorithm and LQR algorithm to control three perpendicular reaction wheels to achieve stability at the cube structure's corner point. An inverted cube robot with three reaction wheels was built as a natural system for controller implementation.

3.1 Mechanical Design

The system consists of two parts, an inner cube of 3 perpendicular and outer structures. The inner cube contains all the control mechanics and electronics such as motors, motor drivers, and microcontroller modules. The challenge of design will be balancing the structure weight and the angular momentum generated by each reaction wheel. The center of the cube's mass should be as close to the structure's center as possible to achieve a balanced structure. The three-dimensional models' part of the inverted cube robot was drew using the SOLIDWORK program and then assembled as a base design for the construction of the inverted cube robot

Figure 3.1

3D Drawing of Inverted Cube's Parts

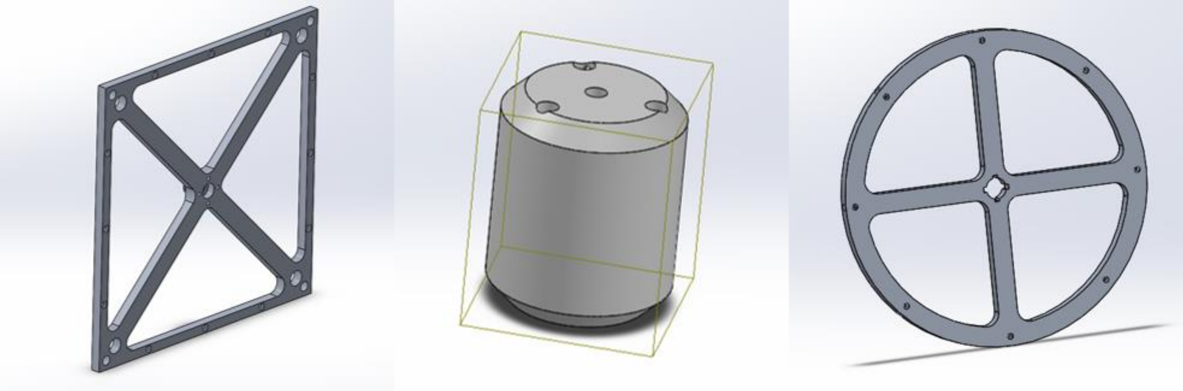


Figure 3.2

Solidwork Assembly of The Inverted Cube

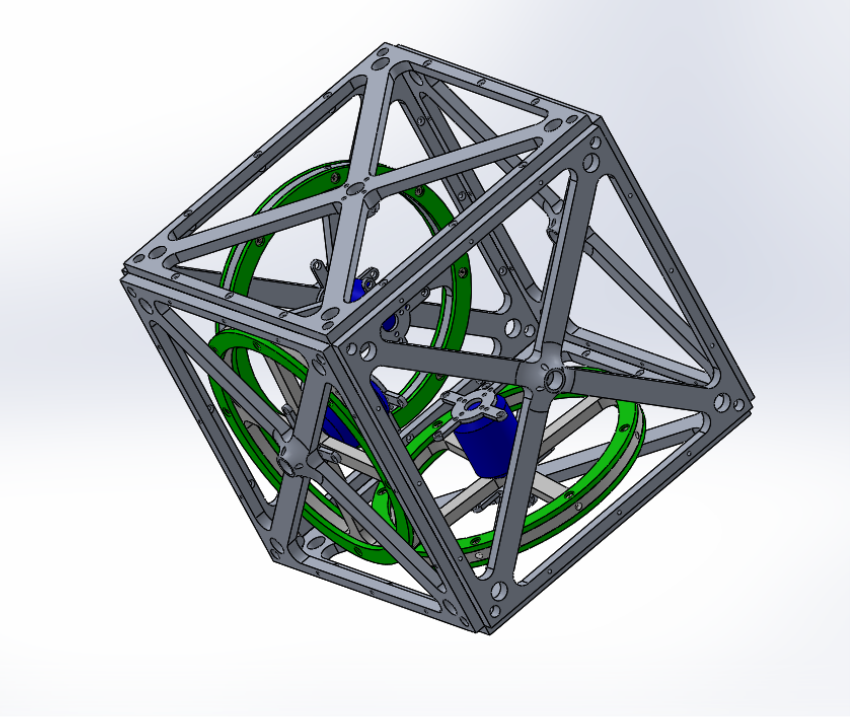
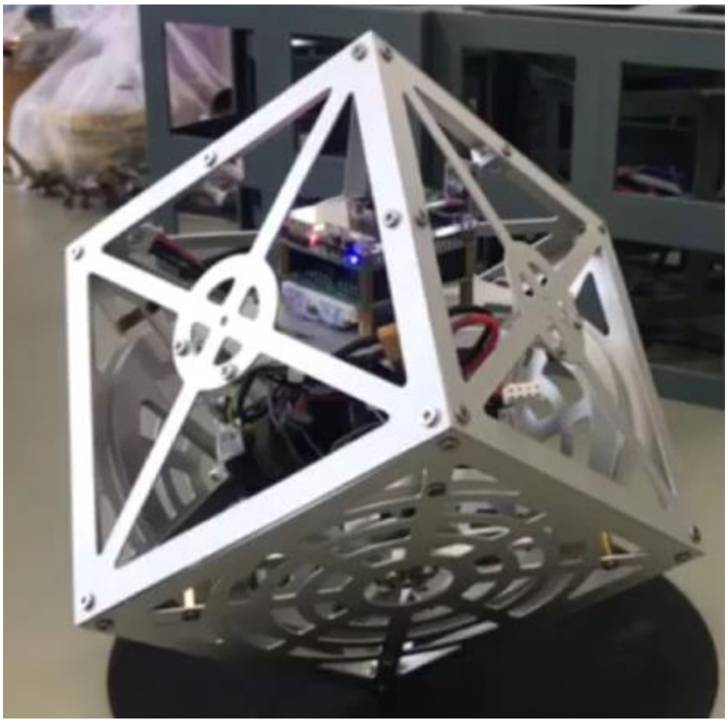


Figure 3.3

Inverted Cube With 3 Reaction Wheels Installed



In term of the electrical component of Inverted Cube, all part is mount inside the cube structure. The STM32f103rct6 (ARM-based 32-bit MPU) was used as the central controller unit by mounting and a motor drive unit, as shown in Figure. 3.5. 12V brushless motors with a build-in encoder unit used as an actuator to drive reaction wheels, picture of the motor mount along with the reaction wheel were shown in Figure. 3.6. The accelerometer and gyroscope unit MPU6050 were mounted inside the cube structure with a reading configuration equal to zero for roll and pitch angle at on corner balance position to collect the cube's orientation data.

Figure 3.4

Circuit Connection

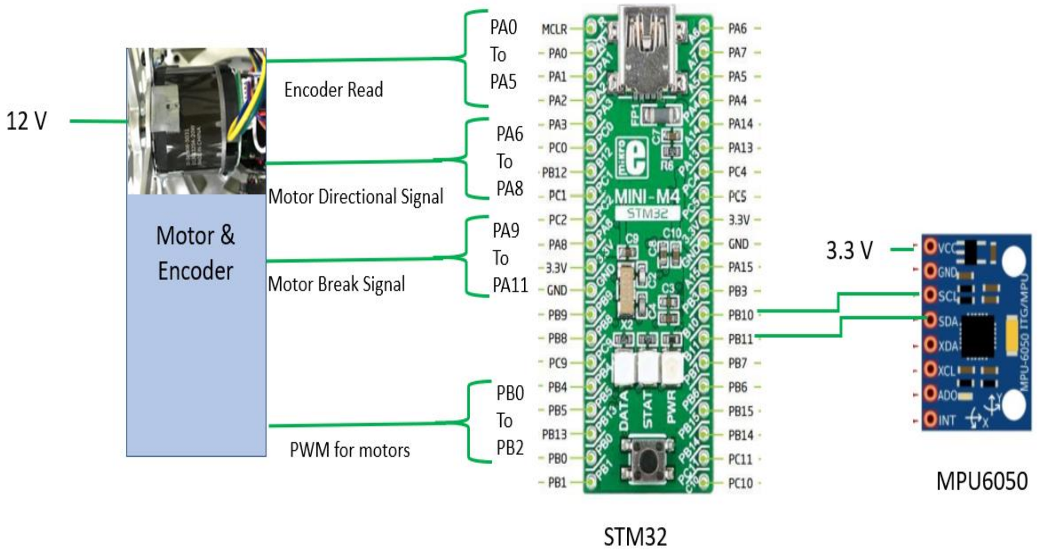


Figure 3.5

STM32f103rct6 and Motor Driver Unit

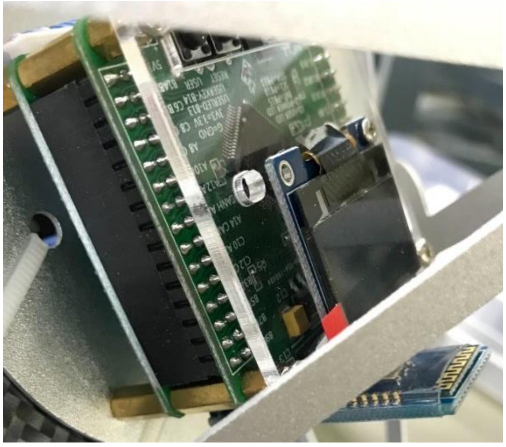
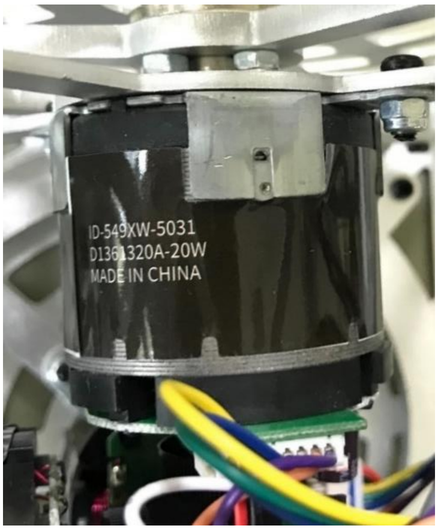


Figure 3.6

Blushless Motor and Reaction Wheel



3.2 Control Algorithm

This section explains the process of controlling the inverted cube robot's components and balancing the inverted cube robot on its corner.

3.2.1 System Identification

Inverted Cube system was model as two axes inverted pendulum with actuator providing torque at the end of a pendulum. Consider torque equation (1), Let α denotes angular acceleration of inverted cube, ω denotes the inverted cube's angular velocity, θ denotes the current angle of the inverted cube. In contrast, consider at balance as angle equal to zero, *g* indicates earth gravitational acceleration, m represents the mass of inverted cube, *l* denotes the distance from the pivot to center of mass of inverted cube, *I* denotes cube moment of inertia, and T represents torque that controller provides to the system.

$$\alpha I = (mg \sin\theta)l - T \tag{1}$$

From equation (1), consider the inverted cube's angle (X1) and angular velocity (X2) as states one and two of the system result in the state-space equation of the system as shown in equation (2) (3).

$$\dot{X}1 = X2 \tag{2}$$

$$\dot{X}2 = -\frac{T}{I} + (m * g * \sin X1) * \frac{l}{I} \tag{3}$$

Because the motor's limitation of torque can provide with the reaction wheel's acceleration, the inverted cube was operated with a slight error angle. Linearization of inverted cube state-space equation results in equation (4), (5).

$$\dot{X}1 = X2 \tag{4}$$

$$\dot{X}2 = -\frac{T}{I} + (m * g * X1) * \frac{l}{I} \tag{5}$$

The value of the state-space equation's parameter was measured, and the result is shown in Table 3.1

Table 3.1

Inverted Cube's Parameter

Parameter	Value (unit)
Mass (m)	1.5 (kg)
Gravity (g)	9.81 (m/s^2)
Distance from Pivot (l)	0.11 (m)
Moment of Inertia (I)	0.1815 (kg m^2)

3.2.2 Reaction Wheels Speed Control

To generate the correct amount of torque for the inverted cube to balance itself on the corner, precise control of brushless motor speed is needed. To achieve that STM32 controller read the speed data from each motor's encode and provide a close loop control signal using PD controller. The motor control and motor control response diagram was shown in Figure 3.7 and Figure 3.8, respectively.

Figure 3.7

Control Diagram of Brushless Motor

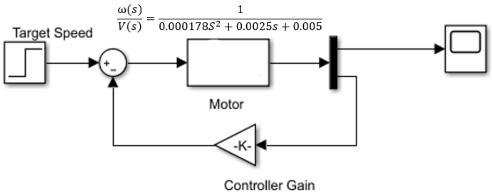
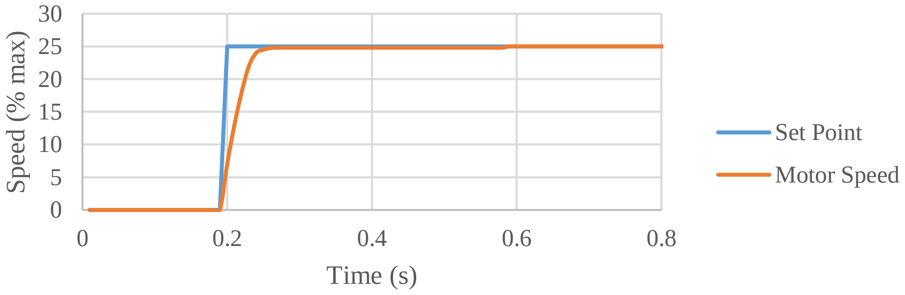

$\frac{\omega(s)}{V(s)} = \frac{1}{0.000178S^2 + 0.0025s + 0.005}$
Target Speed
+
–
Motor
-K-
Controller Gain

Figure 3.8

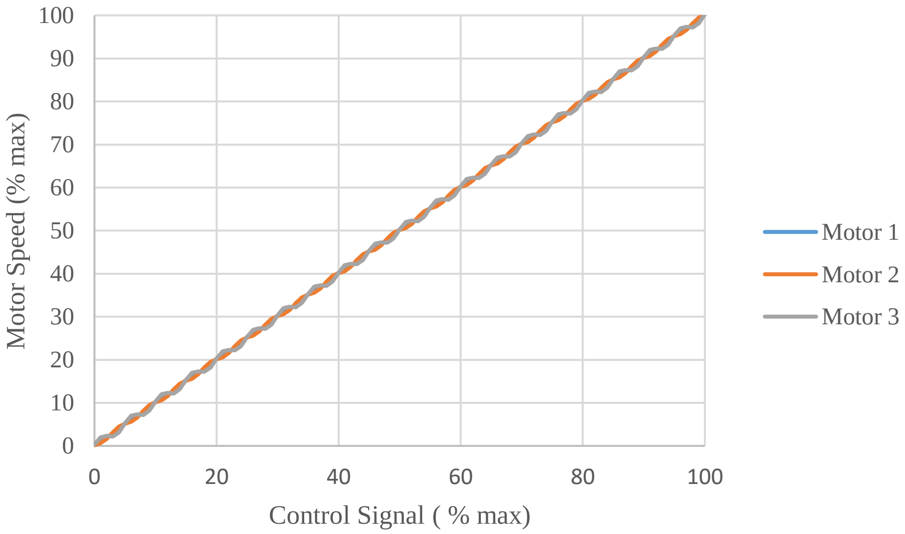
Motor Speed in Respond to Step Function.



Motor speed to control all motors' signal was measured to confirm that all motors have the same output compared to each other. The measurement is shown in Figure. 3.9 shows that all motors have close to identical speed output from the same control signal.

Figure 3.9

Graph of Motors' Speed (% max) – Control signal (% max)



3.2.3 LQR Controller

The state-space equation of the inverted cube from the section above uses the LQR algorithm with the Q and R matrix (6).

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \ R = [2] \qquad (6)$$

All states of the inverted cube can be observed with accelerometer and gyroscope data. The feedback control was implemented in the Simulink model in the Matlab program. Then the control gain was implemented to feedback control of the inverted cube. Each axis's control signal was divided into each reaction wheel using a transformation matrix shown in (7), and the state data was collected.

$$\begin{matrix} T_1 \\ T_2 \\ T_3 \end{matrix} = \begin{bmatrix} 1 & 0 \\ -0.5 & 0.866 \\ -0.5 & -0.866 \end{bmatrix} * \begin{bmatrix} T_{roll} \\ T_{pitch} \end{bmatrix} \tag{7}$$

3.2.4 Reinforcement Learning Base Control

The state-space equation from the section above was used to create a model for actor and critic-based neural networks to train. The neural network was designed to select the action of increasing or reducing controller gain value according to the reward received for each simulation episode to allow the neural network to adjust controller gain based on reinforcement learning. This approach of training reduced the space of training environment to acquire controller gain value and separate requirement of fast control cycle time from calculation time of neural network during balancing on corner of the inverted cube.

The actor model took state data which consisted of controller gain value and initial position and initial velocity of the inverted cube. Then the actor provided probability of each action in action space which consisted of increasing and decreasing of each controller gain value by difference amount. The detail structure of actor model was shown in Table 3.2. Action of controller gain value adjustment was shown in Table 3.3.

Table 3.2

Detail Structure of The Actor Neural Network

Layer	Output Size	Activation Function
Input_1	4	Adam
Dense_1	128	Relu
Dense_2	12	Softmax

Table 3.3

Detail of Instruction of Each Action

Action No.	Action	Value
1, 2, and 3	Increase Controller Gain K1	0.1, 0.01, and 0.001 respectively
4, 5, and 6	Decrease Controller Gain K1	0.1, 0.01, and 0.001 respectively
7, 8, and 9	Increase Controller Gain K2	0.1, 0.01, and 0.001 respectively
10, 11, and 12	Decrease Controller Gain K2	0.1, 0.01, and 0.001 respectively

The critic model shared the same two initial layer with actor model which took in state data and an estimation of total reward in future episode. The neural network aim to minimize the different between on state value function estimated by the critic model and expected returned reward value during simulation of respond of neural network's action. The detail structure of critic model was shown in Table 3.4.

Table 3.4

Detail Structure of The Critic Neural Network

Layer	Output Size	Activation Function
Input_1	4	Adam
Dense_1	128	Relu
Dense_3	1	None

The input structure of state data provided to artificial neural network during training consisted of initial angular position and initial angular velocity of the inverted cube combined with controller gain value of the current episode. The state matrix was shown in (8).

$$State = \begin{bmatrix} \theta_0 \\ \omega_0 \\ K1 \\ K2 \end{bmatrix} \tag{8}$$

Other Hyperparameter parameter related to the training of the neural network was shown in Table 3.5.

Table 3.5

Hyperparameter of The Neural Network

Hyperparameter	Value
Optimizer	Adam
Learning Rate	0.01
Reward Discount Factor	0.99
Clear Running Reward	750
Number of Time Step per Episode	200

The neural network's reward function used the inverse of the total error value of the inverted cube's angle from the balance position with a cost function discount from energy spend. This error value is angular position difference of the inverted cube in each time step of the simulation. The equation of the reward function was shown in (9).

$$Reward = \sum_{i=1}^{200}(\frac{0.01}{error_i}) - abs(Controller\ Gain) \tag{9}$$

The training process of neural network started with provide initial state to neural network model then the prediction of action probability acquired from actor network determine the sampling action provide to the inverted cube simulation. The network store probability of each action, critic’s value, and reward value returned from simulation. After each episode, neural network calculated expected reward from combination of reward value from simulation while giving less weight to early episode.

To update the actor network, loss function is based on policy gradient with critic as a state dependent based line. This was to maximize the network probability output for action that yield the highest reward value. For the critic network, the calculation of loss function used different between expected reward and predicted reward from critic network with Huber loss. This was to train critic network to better predict expected reward of the inverted cube simulation. The entire model was updated with backpropagation using combination of actor loss and critic loss to computed gradient from the combination of loss. Then network apply gradient parameter to update changeable parameter inside actor and critic network. Equation of actor loss and critic loss that network aim to minimize was shown in equation (10) and (11) respectively.

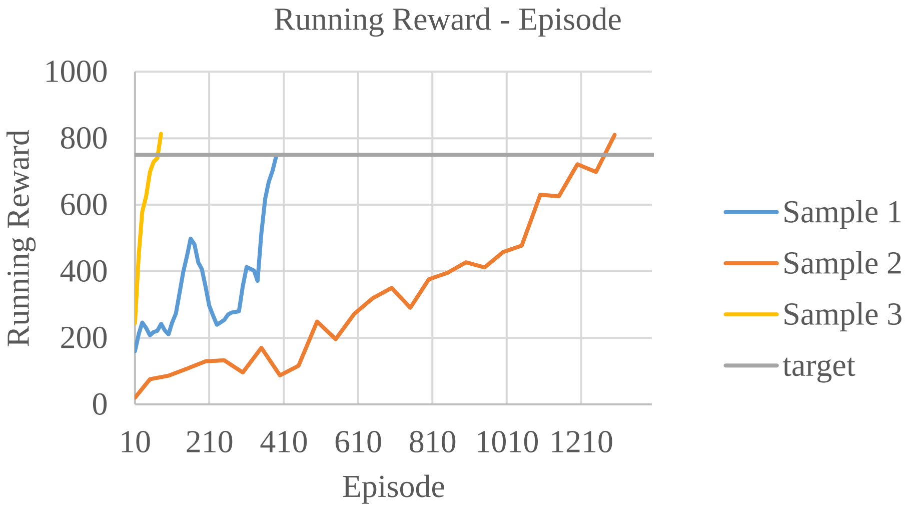
$$Actor\ Loss = \sum_{t=0}^{T}(-\log \pi(a_t|s_t) * [G(a_t, s_t) - V^{\pi}(s_t)]) \tag{10}$$

$$Critic\ Loss = Huber\ loss(G, V^{\pi}) \tag{11}$$

To end the training process, network was set to reach acceptable running reward value which indicated that the neural network could provide controller gain value that result in acceptable respond for consecutive episode. The sample of running reward of the training of the neural network model was shown in Figure 3.10.

Figure 3.10

Graph of Running Reward – Episode During Training of The Model



3.3 Chapter Summary

This chapter contains the methodology of this thesis which consisted of the inverted cube construction process and method to acquire controller gain value to balance the inverted cube using torque from reaction wheels. The control section explains the approach of system identification and controller gain calculation using the LQR method and reinforcement learning-based method.

CHAPTER 4

RESULT

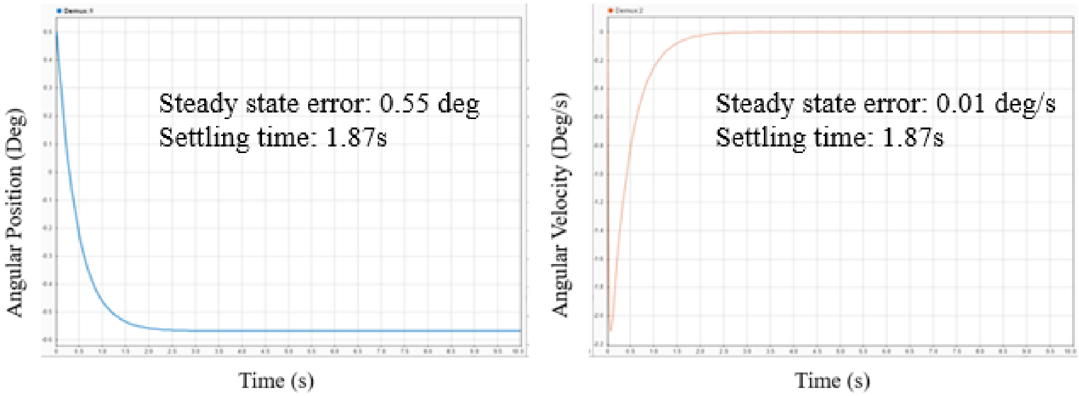
This section shows the result of the implementation of controller gain value from the LQR method and reinforcement learning-based method in MATLAB Simulink and the inverted cube robot

4.1 Implementation Result of LQR algorithm

Implementations result of controller gain from LQR algorithm in Simulink model in Matlab program show that the controller acquires from this method can bring inverted cube from initial that of small error back to steady-state as shown in Figure 4.1

Figure 4.1

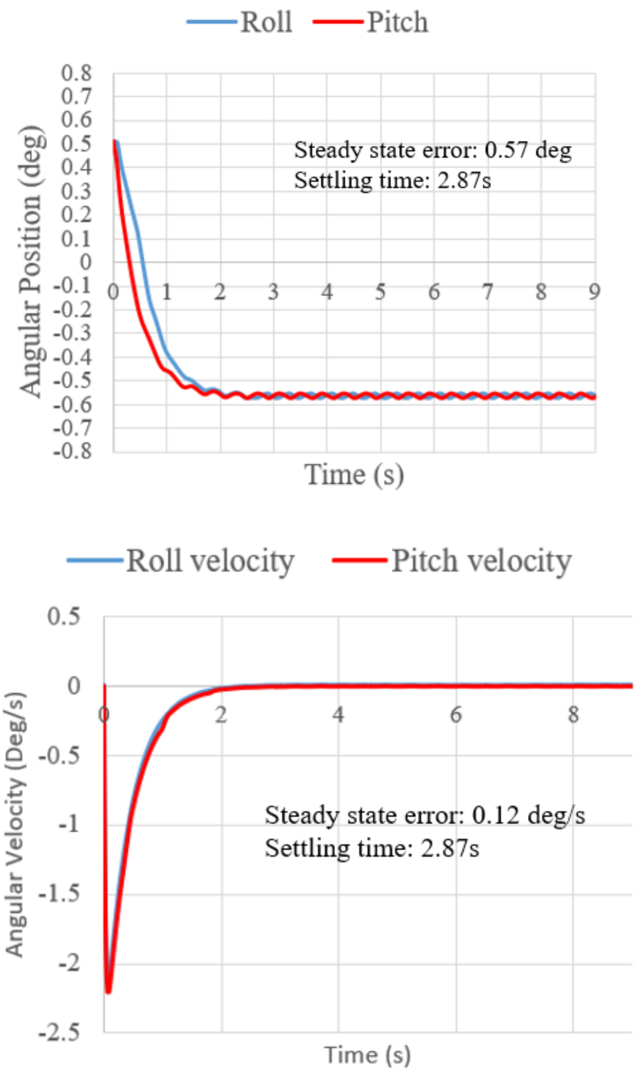
Result of Matlab Simulink Implementation of LQR Gain



Implementations resulting from the same control gain in an actual, inverted cube also show that the cube can balance itself on the corner when starting the system with a small error value. The state data (cube angle and angular velocity) was collected and shown in Figure. 4.2

Figure 4.2

Inverted Cube Angle Responds After implement LQR Controller Gain



Steady state error: 0.57 deg
Settling time: 2.87s

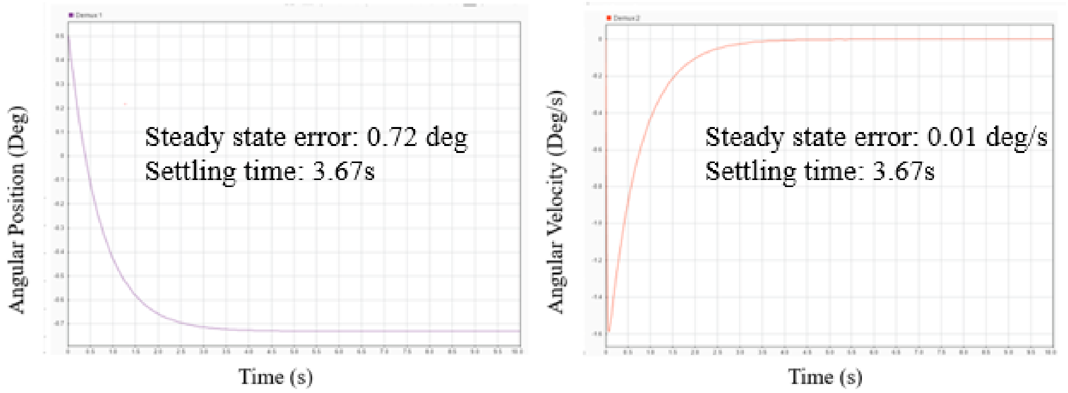
Steady state error: 0.12 deg/s
Settling time: 2.87s

4.2 Implementation Result of Reinforcement Learning-Based Control

After the training, the result controller gain value acquired from neural network show in Simulink implementation that the gain value can balance inverted cube from small error at the initial state, as shown in Figure 4.3

Figure 4.3

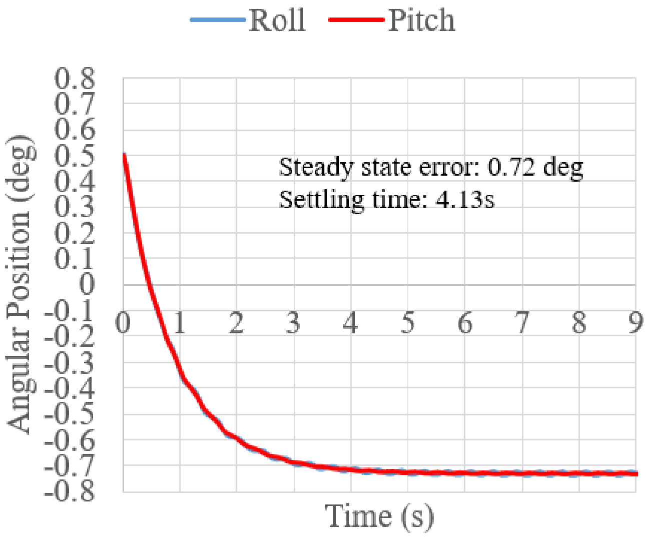
Result of Matlab Simulink Implementation of Reinforcement Learning Gain

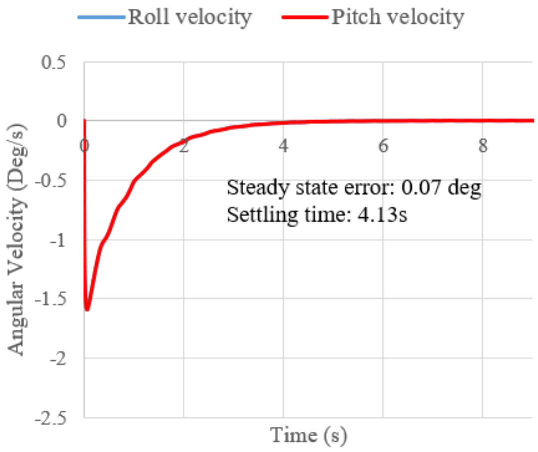


Implementations resulting from the same control gain in an actual, inverted cube also show that the cube can balance itself on the corner when starting the system with a small error value. The state data (cube angle and angular velocity) was collected and shown in Figure. 4.4

Figure 4.4

Inverted Cube Angle Responds After Implement Reinforcement Learning Gain





4.3 Robustness Comparison Between Two Method

Change to parameter of the inverted cube was made by adding more weight to the inverted cube as an experiment to see how both control method will reaction to that change. The result of LQR method and reinforcement learning-based method was shown in Figure 4.5 and Figure 4.6 respectively.

Figure 4.5

Result of LQR Controller After the Change in Parameter

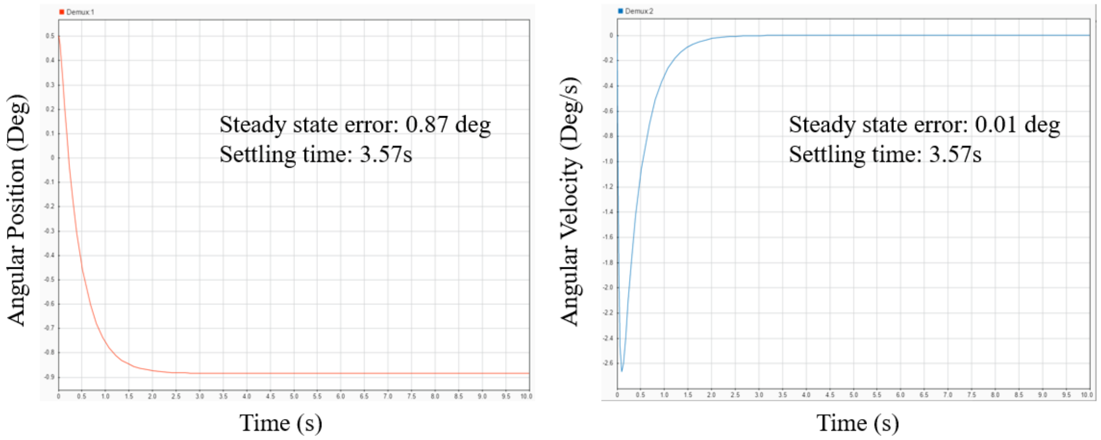
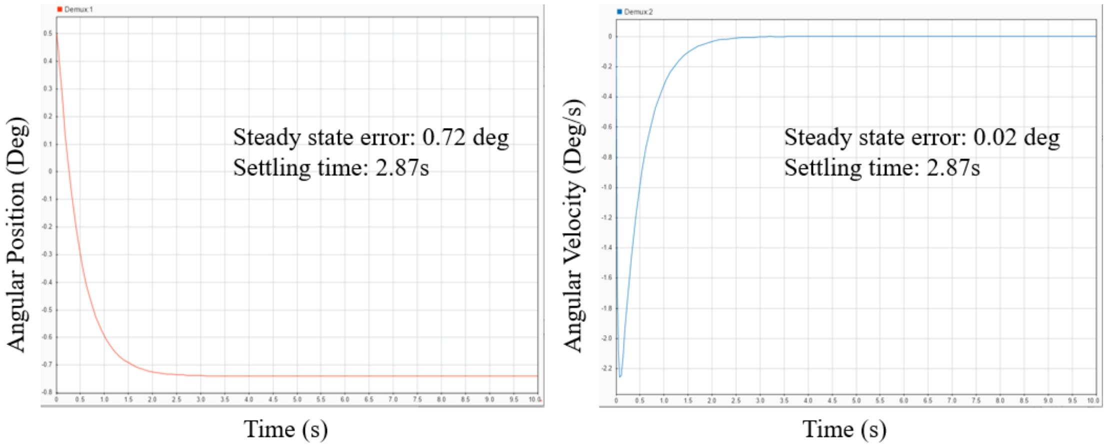


Figure 4.6

Result of Reinforcement Learning-based Control After the Change in Parameter



The result shown that respond of LQR became slower and had more steady state error after the change of the inverted cube parameter. This phenomenon occurred because LQR method acquired controller gain value from system transfer function which changed with additional weight. On the contrary reinforcement learning-based control method’s respond maintain similar respond to the configuration of the inverted cube before change was implemented.

4.4 Chapter Summary

This chapter contains both controller gain value from the LQR method and reinforcement learning-based simulation method and the inverted cube structure to balance the cube structure on the destinated corner.

CHAPTER 5
CONCLUSION

This thesis presents two possible methods to achieve balance control of inverted cube robots with reaction wheel as actuators. The inverted cube maintains its balance position with the initial position close to the balancing position using torque provided by three reaction wheels install inside its structure. Control successfully provides a proper control signal to keep the inverted cube in a balance position using gain acquire from both the LQR method and reinforcement learning-based method.

The LQR approach to balance the inverted cube robot on its corner shows that the balance's angular position had some steady-state error. The above event occurred because the controller only considers two-state from state measurement of the cube's angular velocity and angular position. Therefore, in the actual implementation, the inverted cube had some vibration at the balance position due to correction of position from a small error value. Accordingly, the inverted cube can neglect the external force's disturbance when the correction's torque requirement does not exceed the brushless motor's saturation value. The state above also directly impacted the inverted cube robot's initial position can bring itself up to a balanced position.

While the reinforcement learning-based method provides a very similar response in balancing the inverted cube robot's action, learning the optimal controller gain value depends heavily on the random sampling of action in turning controller gain value. The above-caused reinforcement learning-based method to have varied about episode requirement to reach the worthy goal of reward value to terminate the learning process. The final controller gains value from this method can correctly balance the inverted cube on its corner and neglect external disturbance like the LQR method. Because of the method's randomness nature, the controller gain value that clears the termination condition can also vary from each training session. Nevertheless, this method's result-based nature makes each controller gain results from the training process to balance the inverted cube on its corner.

While the controller gain value from both methods successfully balances the inverted cube on its corner, the significant difference from both methods is the lack of reinforcement learning-based method to prioritize each state, unlike the LQR method. Another downside of the reinforcement learning-based method is that sometimes because of its explorative nature, the controller's adjustment action moves far away from the optimal gain value and moves into territory that reward function cannot differentiate the difference in response inverted cube anymore. Because of the above reason, initial values of controller gain randomly generate had a significate impact on the training process of the neural network to reason acceptable controller value to balance the inverted cube robot.

From above, it can be concluded that both the LQR method and the reinforcement learning-based controller method can be used as a solution to control inverted cube with reaction wheels to balance itself on the corner. This result shows that the reinforcement learning method can be an alternative approach to achieving the dynamic system's balance control. Additional studies could use reinforcement learning as an alternative method to find an optimal control gain for other systems to further explore reinforcement learning ability in control-related problems.

CHAPTER 6
DISCUSSION

The result indicates that both the LQR method and reinforcement learning-based method can balance inverted cube robots. After implementing controller gain from both ways, the inverted cube balance itself on a single conner when inverted cube robot's initial position start close to balance on the corner position.

In line with the hypothesis, the inverted cube's response to the LQR method and the reinforcement learning-based method is similar to each other in the scope that both ways can balance the inverted cube on its corner, neglecting some amount of disturbance from the external force. The reinforcement learning-based control method's result proves that it can be used as an alternative method for balance control problems. Nevertheless, the torque saturation provided from the motor and reaction wheel is the critical factor that can disrupt the system when external force was applied long enough to the inverted cube. While the controller gain value from the reinforcement-based learning method has some random element added to it during the learning period, the controller can use this method to determine gain value without the system's transfer function requirement. Nevertheless, because of limitation due to the number of episodes required for the learning method to achieve controller gain that can balance the inverted cube, the learning process uses computer simulation to create scenarios instead of an actual experiment. Further research is required to explore the reinforcement learning-based control method's ability to optimize controller gain for a different type of balance system.

REFERENCES

B. Bapiraju, K. N. Srinivas, P. P. Kumar, and L. Behera, "On balancing control strategies for a reaction wheel pendulum," in Proc. IEEE 1st India Annu. Conf., Dec. 2004, pp. 199–204.

L. H. Chang, A. C. Lee, Design of non-linear controller for bi-axial inverted pendulum system, IET Control Theory Appl., vol. 1, no. 4, pp. 979-986, 2007

S. Kim and S. J. Kwon, "Non-linear optimal control design for underactuated two-wheeled inverted pendulum mobile platform," IEEE/ASME Trans. Mechatronics, vol. 22, no. 6, pp. 2803-2808, Dec. 2017.

Richard S. Sutton, Andrew G. Barto, and Ronald J. Williams "Reinforcement Learning is Direct Adaptive Optimal Control," IEEE Control Systems Magazine, 1992

Magdi S. Mahmoud and Mohammad T. Nasir "Robust Control Design of Wheeled Inverted Pendulum Assistant Robot," IEEE/CAA Journal of Automatica Sinica, 2017

Kathleen M. Jagodnik, Philip S. Thomas, Antonie J. van den Bogert, Michael S. Branicky, and Robert F. Kirsch, "Training an Actor-Critic Reinforcement Learning Controller for Arm Movement Using Human-Generated Rewards" IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol. 25, 2017

Chao Wang, Jian Wang, Xudong Zhang, and Xiao Zhang "Autonomous Navigation of UAV in Large-scale Unknown Complex Environment with Deep Reinforcement Learning" IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2017

M. Gajamohan, M. Merz, I. Thommen, and R. D'Andrea, "The Cubli: A cube that can jump up and balance," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013, pp. 3722– 3727.

Rupam Singh, Vijay Kumar Tayal, and Hemandar Pal Singh "A Review on Cubli and Non Linear Control Strategy," 1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems, 2016

Zhigang Chen, Xiaogang Ruan, and Yuan Li "Dynamic Modeling of a Cubical Robot Balancing on Its Corner," MATEC Web of Conferences 139, 2017

Sudhir Raj, "Reinforcement Learning based Controller for Stabilization of Double Inverted Pendulum," 1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES-2016) doi: 10.1109/ICPEICES.2016.7853147

Yue Chao, Liu Yongxin, and Wang Linglin. "Design of Reinforcement Learning $ lgorathm for Single Inverted Pendulum Swing Control," Chinese Automation Congress (CAC), 2018, doi: 10.1109/CAC.2018.8623253

WANG Jia-Jun, "Position and speed tracking control of inverted pendulum based on double PID controllers," Proceedings of the 34th Chinese Control Conference July 28-30, 2015.

Sankalp Paliwal, "Stabilization of Mobile Inverted Pendulum Using Fractional Order PID Controllers," International Conference on Innovations in Control, Communication and Information Systems (ICICCI), 2017

Luo Hong-yu and Fang Jian, "An inverted pendulum fuzzy controller design and simulation," International Symposium on Computer, Consumer and Control, 2014

Ramashis Banerjee, Naiwrita Dey, Ujjwal Mondal, and Bonhihotri Hazra "Stabilization of Double Link Inverted Pendulum Using LQR," International Conference on Current Trends towards Converging Technologies (ICCTCT), 2018

VITA

The writer was rewarded first-class honor Bachelor of Engineering from Chulalongkorn University in the automotive engineer department. The writer was also rewarded His Majesty the King's Scholarships to study at Asian Institute of Technology in the mechatronics engineering field.