UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FACULTY OF ELECTRICAL ENGINEERING

COOPERATIVE BETWEEN TWO HUMANOID ROBOTS IN COMPLETING TASKS

Tan Ying Ying

Bachelor Degree in Mechatronic Engineering

2017

I hereby declare that I have read through this report entitled "Cooperative between two humanoid robots in completing task" and has found that it has comply the partial fulfilment for awarding the degree of Bachelor of Mechatronics Engineering

Signature : ………………………………………………………

Name : ………………………………………………………

Date : ………………………………………………………

COOPERATIVE BETWEEN TWO HUMANOID ROBOTS IN COMPLETING TASKS

TAN YING YING

A report submitted in partial fulfilment of the requirement for

the degree of Bachelor of Mechatronics Engineering

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2017

I declare that this report entitled "Cooperative between two humanoid robots in completing task" is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ....

Name : ....

Date : ....

To my beloved father and mother

ACKNOWLEDGEMENT

First and foremost, I would like to thank Universiti Teknikal Malaysia Melaka (UTeM) for providing me such a good study environment and facilities throughout my four-year degree study. I consider myself as a very lucky individual for able to grab the opportunity to study in such wonderful university. I feel grateful for having chance to meet many wonderful people and professionals who provide a good experience throughout this bachelor program.

Bearing in mind previous I am using this opportunity to express my deepest gratitude and special thanks to Faculty of Electrical Engineering's lecturers who provide me with information and preparation during the process of doing FYP.

I am using this opportunity to express my deepest gratitude and special thanks to my final year project's supervisor, Mr Mohd Bazli bin Bahar. Despite of being extraordinary busy with his duties, he took time out to guide and keep me on the correct path while dealing with my FYP.

Special thanks to my family members, coursemates, peers and lecturers for their moral support and technical support. I choose this moment to acknowledge their contribution gratefully.

I perceive as this opportunity as a big milestone in my skill development. I will strive to use gained skills and knowledge obtained in the best possible way.

ABSTRACT

The implementation of robotic technologies in current society has significantly increased over the years especially in industrial and manufacturing field. It has been a great help in ease human daily life. There are plenty of researchers focus on robot-robot collaboration and human operator-robot collaboration. However, lack of research was done for collaboration between two humanoid robots. This is due to the stability and the complexity of controller in control humanoid robots. Although the control of humanoid robot is difficult, there are still some researchers focus on developing humanoid robots which possesses some human characteristic and behaviour of human being. One of the characteristic that possesses by human being is the gregarious, humans are able to work in team to solve problems together. In this project, the main focus is on the movement of humanoid robot's arm. The proposed control system implement IF-THEN rules for the velocity regulator. The rules are set based on the angle disparity in gyroscope and the velocity of robot arm's movement. PID controller are used to regulate the system so the lifting process can be optimized. To verify the utility of the controller proposed, simulations were conducted using Nao robot in V-Rep environment. There were several simulations conducted to test the reliability of the controller and to prove the performance of the humanoid robot increases as the number of robot increases. The result of simulation shows that two Nao robots perform better than single robot when dealing with heavy object according to the object orientation range which is 66.24% less than single robot. Besides, the stability of robot for single robot is less stable compared to two Nao robots during object lifting as shown by foot force sensitive resistor response.

ABSTRAK

Penggunaan robotik telah meningkat setiap tahun terutamanya dalam sektor industri dan pembuatan. Penggunaan robot dalam kehidupan banyak membantu dalam memudahkan tugasan dan kehidupan manusia. Terdapat banyak penyelidikan yang membincangkan kerjasama antara robot dengan robot dan robot dangan manusia. Oleh disebabkan oleh masalah kestabilan dan kerumitan pengawal dalam mengawal robot humanoid, tidak banyak penyelidikan yang memberi tumpuan dalam kerjasama antara robot humanoid. Walaubagaimanpun, terdapat juga penyelidik yang memberi tumpuan dalam pembangunan robot humanoid yang mempunyai sifat-sifat manusia. Kajian tersebut dapat membantu manusia dalam menyelesaikan masalah yang melebihi kemampuan manusia. Sebagai contoh, bekerja dalam suasana yang bahaya, bekerja untuk masa yang panjang dalam sektor pembinaan dan lain-lain. Salah satu sifat yang dimiliki oleh manusia ialah hidup dalam kumpulan, manusia dapat bekerjasama sama sendiri dalam menyelesaikan masalah. Dalam projek ini, tumpuan akan diberikan kepada pergerakkan tangan robot humanoid. Sistem pengawal yang dicadangkan mengaplikasikan peraturan *IF-THEN* untuk pengawalan halaju. Peraturan tersebut ditetapkan berdasarkan perbezaan sudut dalam giroskop dan halaju pergerakan lengan robot. PID pengawal digunakan untuk mengawal selia sistem supaya proses mengangkat dapat dioptimumkan. Untuk mengesahkan utiliti pengawal yang dicadangkan, simulasi dijalankan menggunakan Nao robot dalam persekitaran V-Rep. Terdapat beberapa ujian yang dilakukan untuk menguji kebolehpercayaan pengawal dan untuk membuktikan prestasi robot humanoid bertambahbaik jikalau bilangan robot humanoid bertambah. Berdasar kepada lingkungan orientasi objek, dua robot Nao menunjukkan prestasi yang lebih baik daripada satu robot Nao semasa menangani objek berat dengan 66.24%. Selain itu, robot untuk kategori satu robot Nao tidak seimbang seperti robot untuk kategori dua robot Nao semasa mengangkat objek menurut kepada graf FSR.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

This chapter describe the background, motivation and problem statements to give an idea of the contribution of this research study. The objective, scope and significance of the study are also described here. Lastly this chapter ends with an outline of the thesis.

1.2 Research Background

The purpose of this thesis is to develop a method to perform collaboration of humanoid robot in completing task. Multiple robots have more advantages compared to single robot in the case that the object needed to be carried is large, heavy or the object that has special shape which could not be carried by single robot [1]. Previously, the main focus of cooperative manipulations were performed by wheeled robots, but due to high level of manoeuvrability, humanoid robot get more attention now[2]. Yet, in completing this cooperative feature, there are many aspects involved in the process or action. For instance, the stability of each humanoid robot, the trajectory of humanoid robot in object manipulation, the communication among humanoid robot are the challenges in this field.

As mentioned by Hanzhong Zheng and J.Jumadinova, the potential applications of multi-robot system are highly diverse in various sectors included landmine detection, search-and-rescue operations and others[3]. These sectors involved in huge amount of action, detection, reaction, communication, cooperation and control. Many model and theory are suggested to perfectionate and improve multi-robot system in terms of cooperative feature including Artificial Immune Network, Cooperative Learning and Master-slave system[4]–[6], [7], [8], [9].

After revise past research, most researchers focus in human-robot collaboration and non-humanoid robot collaboration. Only several researches are focusing in humanoid robots' collaboration[1], [5], [6].

1.2.1 Motivation

This project was carried out based on various motivation which focus on humanoid robots that could manipulate object like human. Similar to human, humanoid robot has high manoeuvrability which sometimes can substitute human in tasks. For example, humanoid robot ARMAR-III developed by Karlsruhe Institute of Technology (KIT) serves as kitchen helper as shown in Figure 1.1[10]. Humanoid robot are expected to serve as human's companions or assistance in the future[11].

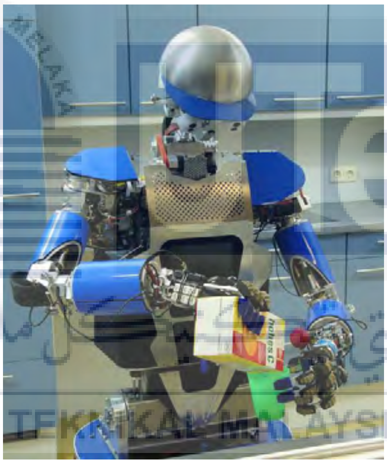


Figure 1.1: The picture of ARMAR-III[10].

Humanoid robots with the ability to lift object able to substitute human from many aspects like serve as labour in goods transportation, serve as waiter in restaurant and serves as bellhop in hotel.

1.2.2 Problem in control humanoid robots to work cooperatively

The main challenge in humanoid robot cooperation is about how humanoid robots can perform simultaneously in completing tasks. In solving the problem, the main component of the humanoid robot is the control scheme. This control scheme must be considered and well-planned for the sake of good performance in object manipulation. Either side of robot which could not react to another robot can cause failure when object manipulation.

Control scheme can be divided into two parts, (1) phase planning and (2) interaction control. As for motion control, the focus of the control scheme is to follow the motion that has been planned to perform object manipulation. Phase planning is a proper plan to separate the whole movement into several phases. The phases are usually separated based on the task requirement at the particular time. However, the motion of humanoid robots in manipulating object are irregular[12]. In case of object transportation with two humanoid robots, mutual position shifts may occur due to the body swinging of robots. Hence, it is necessary to correct the position in a real-time manner[13]. During object manipulation like pushing object required a correct posture to maximize utilization and reduces the load endured at each joint.

The interaction control can be diverse into a few types, included master-slave, Q-learning and reinforcement learning method as well as Artificial Immune Network. Master-slave control is one of a good approach to exchange data among master robot and slave robot. Yet, there is a huge setback for this system at which the whole system will collapse if master robot malfunction. Furthermore, there are time delay exists in the transmission of data between the master and slave side [14]. Q-learning and reinforcement learning has advantage when cooperate with human but not for multi-robot system. Artificial Immune System can perform 2-sided interaction.

1.3 Problem Statements

Cooperative humanoid robots can be applied in hazardous situation such as work under radioactive environment, transport explosive disposal and industrial applications. In present research, a high performance and robust controller was developed for the safe load handling, transportation and trajectory checking[1]. In most of the situation, especially handling complex tasks, a single robot might face problem to accomplish the task on its own.

For example, having problem in lifting or moving object. Hence, cooperation between robots is the best way to complete the tasks. Similar to bio organism, in order to receive help from allies, there must be a sort of contact or communication to inform them or regulate. For example, sound, smoke, light and others. In other words, these signal or communication can help in deliver signal. However, physical signals as mentioned are not suitable to be implemented in a robot network. This is when wireless connection or electrical signal come into solving these issues in coordinate the velocity and amount of torque of the robots in lifting object to balance the object lifted.

Thus, the research question is *how a communication network can be established so other robot can receive the signal?*

Both humanoid robot needs a control scheme in order to regulate their torque or velocity in lifting object to ensure the object lifted always balance. At least two joints from the arm of the Nao robot are needed to lift object, the torque and velocity must be carefully controlled to avoid from causing imbalance of the object lifted. Since both Nao robot might be differed from each other in terms of response time and condition of actuator, the deviation of time, torque and velocity between two humanoid robots may lead to failure of the operation due to imbalance from one side of the robot.

The second research question is *how to control the robot's motion so both humanoid robots can regulate themselves according to the stability of the object?*

1.4 Objective of the Research

From the research question stated in problem statement, the research objectives are:

1. To implement proper trajectory planning for lifting an object using humanoid robot.
2. To design and develop a control system that allows robots to cooperate in lifting object.
3. To develop and analyse the performance of two humanoid robots compared to single humanoid robot.

1.5 Scope of Research

The scope of the research are as follows:

- Simulation work is done by using V-rep.
- Simulations are done using Nao humanoid robot.
- Two Nao humanoid robots are used in the simulations.
- The control method focuses on controlling the motion of the arm which are shoulder, elbow, wrist and finger.
- Nao robots only perform object lifting.

1.6 List of Contribution

This research will contribute to the development of humanoid robot by equipped it with the ability to cooperate with other humanoid robot which similar to human nature. With the capability to cooperate with other humanoid robots, the robot can manipulate objects with its partner which can replace human labour in dangerous environment. For example, manufacture sector and construction sector.

Besides, the number of humanoid robots involved are the key element which brought the significance. The advantages of using two humanoid robots are overwhelming single humanoid robots. In general, the time consume and the energy consumption is short and less as compare to single robot.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter begins with an overview of robotic system. Then, the past works of robotic system in terms of robot's dynamic stability, object manipulation and followed by collaboration between robots. This chapter also discuss about the constraints in the research field and the significance of it. At the end of this chapter, a summary for the whole chapter was discussed.

2.2 Robotic System Overview

According to K.H. Low, robotic is the art, knowledge base as well as the skill in design, apply and use robots in human endeavors[15]. Beside robot, robotic system consists of other devices and systems which collaborate with robot to perform necessary tasks. Robotics is an inter-disciplinary subject that involve mechanical engineering, electrical engineering and electronic engineering, computer science, biology and many other disciplines. The designation of robots depends on the ideas and theory developed in the disciplines involved.

Robotics is broadly defined as the intelligent and interactive connection of perception to work through cognition and planning. The following technologies are including under this general definition:

- Kinematics, dynamics, control and simulation of robots.
- Sensing and perception: vision and other non-contact sensors; tactile and other contact sensing systems.

- Systems control theory and applications as related to the modelling of robotic systems.
- Robot mobility and navigation.
- Robotics-related computer hardware and software components, architectures and systems.
- Advanced command and programming languages for robots.
- Linkages to computer aided design, engineering and manufacturing information systems.
- Electronic and manufacturing science and technology as related to robotics.
- Man-machine interfaces are related to robotics.

As the year goes, the demands for robot that can perform human's tasks increases. According to Appin Knowledge Solutions, many organizations are involved in various fields of robotics[16]. These field of robotics can be broadly categorized as:

- Robotic Manipulator
  Robotic manipulators have become useful and economical tools in manufacturing, medicine and other industries.
- Wheeled Mobile Robots
  Wheeled mobile robots perform many tasks in industry as well as military.
- Legged Robots
  Locomotion on the ground can be realized with slider, liver and wheel. The benefits of legged robots are better in mobility, stability on the platform, energy efficiency as well as has smaller impact on the ground.
- Underwater Robots
  Camera-equipped underwater robots serve many purposes including tracking of fish and searching for sunken ships.
- Flying Robots
  Flying robots have been used effectively in military maneuvers and often mimic the movements of insects.
- Robot Vision
  Provide machines with sensors that mimic that capabilities of the human vision system. This process is the creation of the sensing devices that capture the same raw information light that the human vision system uses.
- Artificial Intelligence

Artificial Intelligent (AI) is a branch of computer science and engineering that deals with intelligent behaviour, learning and adaption in machines.

- Industrial Automation

  Assists human operators with the physical requirements of work, reduces the need for human sensory and mental requirements.

Besides, humanoid robot is getting attention as their actions and motions are built based on human's nature. For example, sitting, lifting and interact with nature. These features or function of humanoid robot can perform human tasks or dangerous job [6], [17]. Study and imitate from human nature, interaction of the robot with environment also become one of the study among researchers[13]. Until now, humanoid robots are able to interact with human operator or another robot in completing tasks like Nao robot does. The development of this study can bring huge advantages to human society especially in manufacturing sector.

2.3 Collaboration between Humanoid Robots

In human society, the cooperation between human are one of the uniqueness in human. Human are able to complete complex tasks due to this uniqueness, a simple example would be cooperation to lift a heavy object. The same concept applies on humanoid robot, as the technology getting advance, humanoid robot can assist human in various tasks [18], [19] and for certain extend they can replace human in some simple job or dangerous places [12]. Among all of the robots, humanoid robots have the potential in handling multiple tasks and walk on any terrain, like human beings. The limitation of actuator output cause humanoid robot unable to work efficiently as human does, hence multiple humanoid robots working cooperatively will be a solution to exploit the capability of robots [6], [13].

Multiple robots can increase the efficiency and the robustness of the system unlike using single robot. A greater number of the robots can produce a self-organization system that is consequently robust to environment changes [4]. The characteristics of collective autonomous mobile robots can be categorized into three: (1) distribution of autonomy, (2) cooperation and (3) diversity [7]. As the complexity of tasks increase in real situation, single robot design has great limitations [8] and multiple robots are expected to perform the complex task cooperatively [7].

Every motion of the robot started with standing position at which the Centre of Mass (CoM) does not deviate from original position that allows both two humanoid robots to be stable. However, when both robot start to manipulate the object given, the CoM started to change and instant calibration is needed to ensure robots does not lost their stability. Besides, in order for robots to manipulate object, for example lifting object simultaneously, both robots have to be synchronized to lift the object together. The ability to manipulate object must be designed with a proper control mechanism and strong musculoskeletal to support the motion.

2.3.1 Challenges in Collaboration between Humanoid Robots

There are many challenges encountered in this situation. However, the main challenge in collaboration between humanoid robots are control problem and stability problem.

From point of view in term "control", using cooperative robots arises from the necessity of more complex mechanical analysis and control strategy design [20]. The difficult part is the mechanical analysis for dual robots that mainly comes from closed chain system, which consists of more kinetically and dynamically complex than typical serial manipulator configuration [1]. In addition, complex tasks is divided to several sub-tasks, hence the order of solving the sub-task must be resolved before controlling humanoid robot [21]. In terms of stability problems, the main issue is that humanoid robot has unstable balance and difficult to operate [13]. As the tasks are getting more complex , and it is required for humanoid robot to have high stability [22].

From the problem encountered in other research, it shows that robots' cooperation problem highlights the essential of appropriate method to solve the deviation of Centre of Mass (CoM) and the control of robot in completing tasks. For every robot's movement especially lifting object, stability is a very important element to avoid humanoid robot from falling, hence relocation of CoM needed to be solved. During object manipulation, a control method is needed to stabilize the robot and allow robot to correct their manipulation process via feedback.

2.3.2 Method to Maintain Stability

The main components of the collaboration of humanoid robots are the (1) stability and (2) motion control [1]. Balance is the ability to maintain body centre of mass (CoM) over its base of support whilst performing task with minimal postural way. A good balance in both static and dynamics activities can reduce the usage of energy and minimize fatigue. According to [23], for robot to work normally and efficiently, it is inevitable to equip robots with a good balance capabilities during the manipulation tasks.

To maintain the stability of humanoid robot, existing research has been focused on compensating zero moment point (ZMP) error for biped balance [22]. The ZMP is defined as the point on the ground about which sum of all the moments of the active forces is equal to zero. If the ZMP is within the convex hull of all contact points between the feet and ground, the biped robot is possible to walk. Hereafter, this convex hull of all contact points is called the stable region [24].

There are several researchers have been focus to improve stability via different method. For instance, Fuzzy System [24], Tactile Sensing System [23], Mass Augmentation Method [25] and other methods. Y.S. Cha and etc. propose to improve the humanoid robot's performance and enhance its capability to deal with tasks in dynamic and time varying environments [25]. It aims at improving humanoid robot's stability of the ZMP and robustness of walking through uncertain terrains. Kitti Suwanratchatamanee and etc. proposed a method to control balance of the robot using tactile sensing system at which the experiment only focus on the prototype tactile sensing hand and foot [27]. In another paper, a method called mass argumentation method is proposed by Y.S. Cha and etc. covered the balance control for the whole body of humanoid robot[25]. Mass argumentation method is the combination of MECoM (Motion Embeded CoM) Jacobian implemented in Y.Choi and etc. and the CoM-ZMP controller[28].

Different methods have been proposed to solve stability problem, but currently there are no method to ensure robot's stability can perform as perfect as human being. Researcher can only improve the stability by reduce the time in stability regulation when robot moves. Due to the research done by different researchers, there are difficulties in comparing the method. The experiment conducted by Meng Joo Er and Yi Zhou are based on walking motion differ with K. Suwanratchatamanee and Y.S. Cha which conduct the stability test which based on object lifting[24]–[26].

2.3.3 Object Manipulation Control

Object manipulation like push object and lift item are norm in human daily life[29]. In robotic system, there are two aspects to be concern, stability and inertia force whilst object manipulation[29], [30]. To solve both of these problem, H. Arisumi and etc. proposed two method which used centre of percussion (CoPn) of the system to solve stability issue and a design method of preliminary motion to solve the issue of insufficient momentum to lift up object[31]. In another research paper, S. Chen proposed a method known as squatting down and lifting object[32]. This method is a basic motion that ensure the humanoid robot can keep balance while lifting object.

There are also researchers that combine various method that allows humanoid robot to manipulate object. Hung-Yi Lee and etc. suggested a method named whole body motion planning[29]. Tzuu-Hseng S.Li and etc. used accelerometers and force meters with a fuzzy controller to design a dynamic balanced controller[33]. Via the compensation of the controller, the stability and the robustness of adapting to lift the weight can be strengthen. Hui Liu and etc. have proposed a method at which combine the usage of ultrasonic sensors and artificial neural networks [34]. Another method to perform object manipulation is predicting lifting task [35]. For this method, sensors like force sensors, distance sensors and sonars are installed at different part of the robot and use as data collector to predict the object position.

Masaki Murooka and etc. proposed a method called as whole-body holding manipulation by humanoid robot based on transition graph of object motion and contact [36]. This method is the combination of two method, transition graph generation and robot posture planning.

After read from the past researches, some researchers focus on the posture configuration at the beginning and apply on the robot. The posture created is identical to human's posture, for example the squat down motion planned by S. Chen and etc. [32]. However, the humanoid unable to regulate and relocate the CoM during lifting process like the humanoid robot who used fuzzy controller and ANN.

2.3.3.1 Collaboration between Robots in Object Manipulation

Recently, researchers have been focus on the study of collaborative robot between human-robot and robot-robot [13]. S.A.A. Moosavian and etc. state multiple robots have advantage over single robot since it can transport large objects, heavy objects and even objects of a special shape. Hence, S.A.A. Moosavian and etc. have published a paper about the comparison of Modified Transpose Jacobian (MTJ) controller and Transpose Jacobian (TJ) controller in object manipulation of two humanoid robots. Both algorithm is applied to control two cooperative humanoid robots for tracking the stable planned trajectory. H.G. Kim and etc. proposed the usage of PID controller and fuzzy controller to perform collaboration between robots [12]. PID controller used to correct the errors of direction and the speed from the object's horizontal and vertical components. M.H. Wu and etc. have proposed to construct a central controller that controls all the robot simultaneously without distinguishes leader and follower robot [6]. The synchronous movement of the robot can achieve high responsibility and high robustness in carrying object.

Y.Inoue and etc. proposed two Cooperative Learning method which are Classifier system and Q-learning [13]. Classifier system enable robot to adapt to dynamic environment in simulation while Q-learning enable robot can guarantees the state transition in the environment of a Markov decision process converges into the optimal direction in simulation.

There are various controllers or methods which enable researcher to achieve the cooperative behaviour in object manipulation. The controllers proposed included Modified Transpose Jacobian (MTJ) controller, Transpose Jacobian (TJ) controller, PID controller and fuzzy controller to plan, regulate or control the robot in object manipulation. Different controllers and algorithm affect the types of robot. In general, there are two types of robots categorised in this section, leader-following type and symmetry type. For leader-following type, the dependence on the planning robot is high, any disturbance or situation that affect it can cause the whole movement planning to be fail. Plus, the existence of time-lag in this type of robot cause unexpected falling. For symmetry type, in order to control two humanoid robots using only a central controller require more complex mechanical analysis and control strategy design in order to perform complex tasks.

2.3.3.1.1 Cooperative Behaviour among Robots

There are algorithms designed based on human immunity system (HIS) in controlling the movement of robots. N.R.Ramli proposed an algorithm that allows robots to improve the cooperative behaviour by imitate the biological immune system and immune response[4]. They proposed a theory named Somatic Hypermutation theory to integrate with immune network algorithm in order to improve cooperative behaviour among robots by learning mechanism. Yunyun Gao and Zhizeng Luo proposed an Artificial Immune Network (AIN) model's algorithm for the multi-robot system based on biological immune system which are known as immune-based static allocation algorithm and dynamic algorithm[9]. Static allocation algorithm imitates the properties of immune system and utilize the interaction among the antibodies. Meanwhile, dynamic algorithm integrates autonomous cooperation into self-learning of the robot. Dong-Wook Lee and Kwee-Bo Sim have proposed a Swarm-Immune Algorithm based on immunity system in distributed autonomous robotic system (DARS) [7]. The control scheme for this particular research is based on clonal selection and idiotopic network hypothesis which is claimed to be decision making in optimal swarm strategy.

Based on the past research, the basic of Artificial Immune Network (AIN) are made up of two algorithms, one for working cooperatively and the other is developed for self-learning behaviour. Different researches have emphasized on different part of the AIN model. Conclude from the three paper above, each of the paper has proposed different theory or algorithm in building the communication or control network for multiple robot.

2.3.3.2 Object Manipulation between Human Operator-Robot

Human-robot collaboration (HRC) is a sub-discipline which vastly developed in Robotic [37]. For human operator-robot in object manipulation, human operator can easily receive or accept object from robot, yet there are challengers for robot to perform this manner [38]. W.Sheng and etc. proposed the method, imitation learning and reinforcement learning with a proactive controller[19].

As proposed by P.Evrard and the research team, the problem is tackled using Programming by Demonstration (PbD) [39]. Gaussian mixture model (GMM) is used to encode the collerations between the different variables of the task while Gaussian mixture regression (GMR) are used to reproduce the demonstrated task. For the research done by W.Sheng and etc. , Gaussian mixture model (GMM) and Gaussian mixture regression (GMR) are used for the first phase which enable robot to reach out and grab the selected object and using proactive controller for collaboration phase[19].

Both methods used GMM and GMR but W.Sheng and etc. used extra controller to compensate[19]. Proactive controller allows robot take proactive actions based on human motion prediction, hence humanoid robot can perform a better prediction and reaction toward human-operators action.

2.4 Methodology to Maintain Stability

Meng Joo Er and Yi Zhou proposed a method called Fuzzy Q-Learning (FQL). FQL is applied in order to improve the Zero Moment Point (ZMP) performance by intelligent control of the trunk of a humanoid robot. FQL is an extension of the original Q-learning that tunes FIS conclusions. They conducted the software simulation using three condition, FQL with Gaussian MF, FQL with triangle MF and FQL with scale reward. The graph is then compared with ideal ZMP. External random noises are included in the simulation to demonstrate the uncertain time-varying situation for biped walking[24].

Kitti Suwanratchatamanee and etc. proposed a method to control balance of the robot using tactile sensing system at which the experiment only focus on the prototype tactile sensing hand and foot. The experiment is conducted by using humanoid robot to lift up objects which in half and quarter of robot's body weight with single left arm. The balance control is done by using tactile sensing [27]. In another paper, a method called mass argumentation method is proposed by Young-Soo Cha and etc. cover the balance control for the whole body of humanoid robot [25]. Mass argumentation method is the combination of MECoM (Motion Embeded CoM) Jacobian implemented in [30] and the CoM-ZMP controller. The experiment is conducted with humanoid robot, Mahru which stand still with a basket and the object will be loaded and unloaded. Two object with different weight is used, which is 0.5kg and 2 kg respectively. Then, experiment is conducted with and without the activation of mass augmentation.

There are two different situations here to test the stability of humanoid robot using ZMP. In Meng Joo Er and Yi Zhou's research, the simulation shown in the result for FQL with Gaussian MF is just slightly different compared to ideal ZMP when humanoid robot walks. In comparing another two study, the humanoid robot with tactile sensing is better than Motion-Embeded CoM (MECoM) Jacobian resolution method. The humanoid robot with tactile sensing system can lift 1.025 which is approximate 40% of humanoid robot's weight for 21.5 seconds. For Motion-Embeded CoM (MECoM) Jacobian resolution method, the maximum weight can lift by a 67kg humanoid robot is 3.5kg which is 0.0522% of robot weight. From the aspect of robustness, the study method conducted by Meng Joo Er and Yi Zhou in the stability of walking needed more stability than static robot with object lifting. Hence, the study proposed by Meng Joo Er and Yi Zhou is more suitable to maintain stability of mobile robots[24].

2.5 Methodology for Object Manipulation

H.Arisumi and etc. proposed two method which used centre of percussion (CoPn) of the system to solve stability issue and a design method of preliminary motion to solve the issue of insufficient momentum to lift up object. The first method is the robot is asked to stand up when a singular configuration is reached. By using inertia of the object, provides an acceleration to reduce the power needed to support the load of the object. The other method is that the robot is asked to crouch quickly under the object and to reach a position where robot can lift the heavy object[31].

In another research paper, S.Chen and etc. proposed a method known as squatting down and lifting object. This method is a basic motion that ensure the humanoid robot can keep balance while lifting object. This paper states that the planning of squatting down and lifting object belongs to the static planning, hence inertia force can be ignored for this case. There are four phases of movements, which are lift arms, fold arms and open legs, squat and lift up. The simulation is made using experimental data that satisfy the equation proposed[32].

There are also researchers that combine various method that allows humanoid robot to manipulate object. H.Y.Lee and etc. suggested a method named whole body motion planning[40]. The relationship between the velocity of the end-effectors and COG by the angular velocity of each joint are obtained using fixed-leg Jacobian and COG Jacobian to calculate Inverse Kinematics. Then, generate a suitable trajectory and conduct a series of simulation and experiments on different situation like are pushing a heavy object, robot use two arms to lift an object and raising an object with an arm. T.H.S.Li and etc. used accelerometers and force meters with a fuzzy controller to design a dynamic balanced controller[33]. The humanoid robot is equipped with centre process unit, image process unit, sensors, the integrated power circuit board and SOPC chip, Altera EP1C12F324C8. With sensors attached on the humanoid robot, humanoid robot can adapt to environment by using the compensation of the controller. Hence, the stability and the robustness of adapting to lift the weight can be strengthen. Hui Lui and etc. have proposed a method at which combine the usage of ultrasonic sensors and artificial neural networks [34]. There are 4 stages mentioned in this method, starting with measurement of distance using ultrasonic sensor, establish a nonlinear relationship between ultrasonic distance and joint controlling values and forecast the next-step joint controlling values by using Artificial Neural Network (ANN). Humanoid robot is expected to grasp a selected item with a gap in between them with 10 set of data prior inserted to it. The data will read by the ANN and use to do the arm real-time controlling.

Another method to perform object manipulation is predicting lifting task [35]. For this method, sensors like force sensors, distance sensors and sonars are installed at different part of the robot and use as data collector to predict the object position. These data are then used in two different process either online or offline. Finally, predict whether the lifting motion is success or fail. The whole process for this method is conducted in simulation.

M.Murooka and etc. proposed a method called as whole-body holding manipulation by humanoid robot based on transition graph of object motion and contact[36]. This method is the combination of two method, transition graph generation and robot posture planning. For this method, humanoid robot can change the way to lift or grasp according on the friction of the object. This method can solve the stability and inertia problem encountered when a robot manipulate object. If the friction of the object is low, the robot will slightly lift the object from a side and lift it with a hand at the bottom and another hand as supportive to hold another side.

The condition for each study are different in terms of weight of load, distance walked and time before the humanoid robot fall. The weight lift in is not clearly mentioned by T.H.S.Li and etc. , only the distance is mentioned which is 30cm[33]. For H.Arisumi and etc., a 58kg humanoid robot lift 4.5 kg but does not walk after lifting object[31]. In H.Y.Lee's research, the robot successfully lift a 10kg object and perform side walking for 10 steps and walking straight for 6 steps[40]. For [36], the humanoid robot can regulate its movement to lift a 10kg load. From the overall result, the method proposed by M.Murooka and etc. is better as the way of object manipulation depends on friction of the object[36]. However, this method is not practical when the object inside the box is fragile.

2.5.1 Object Manipulation between Robot-Robot

S.Ali A. Moosavian and etc. have published a paper about the comparison of Modified Transpose Jacobian (MTJ) controller and Transpose Jacobian (TJ) controller in object manipulation of two humanoid robot. Both algorithm are applied to control two cooperative humanoid robot for tracking the stable planned trajectory [41].

Yukata Inoue and etc. proposed two Cooperative Learning method which are Classifier system and Q-learning. The study is using a humanoid robot, HOAP-1 which is 5.9kg. The lifting movement of the humanoid robot is done for forward, backward, leftward and rightward for 10 times each. All the simulation is done by using simulation environment [5].

There are researchers Han-Guen and etc. proposed the usage of PID controller and fuzzy controller to perform collaboration between robots. PID controller used to correct the errors of direction and the speed from the object's horizontal and vertical components. Meanwhile, fuzzy controller used to solve the irregular motion of humanoid robot during carrying object [12].

Classifier method allows humanoid robot to perform the learning of correct its position and better adaption to the dynamic environment, but might result in poor performance due to overfitting. For Q-learning can guarantees the state transition in the environment of a Markov decision process converges into the optimal direction in simulation but it is not adaptive to surrounding. MTJ algorithm can helps in regulate the stability during lifting object only. For fuzzy controller, fuzzy rule allows humanoid robot in correcting the speed of master and slave robot whist avoid robot from falling.

2.5.1.1 Cooperative Behaviour among Robots

N.R. Ramli and her team designed an algorithm which has similar to the human immunity system's behaviour. In N.R. Ramli and her team experiment, two robots are assigned in pushing four boxes with different weight and size, which are 1kg small box, 1 kg big box, 2kg big box and 3kg big box respectively. If a robot fail to push the selected box, robot will mutate its action to cope its ability with the environment. The robot will seek help from other robots if the robot still fail to push the box after mutation[4].

Yunyun Gao and Zhizeng Luo proposed an Artificial Immune Network (AIN) model for the multi-robot system based on biological immune system at which the properties of immune system are imitated and applied among robots to perform tasks and cooperation. The research is performed in simulation environment. Two different experiment are carried out in this research, simulation with immune-based algorithm and simulation with greedy algorithm. The robot team is assigned to fix the problem indicated by alarm that exist in the simulation environment. The simulation environment is a 30*30 m^2 field. 30 alarms and 5 heterogeneous robots are distributed randomly. The speed of the robots are different from each other which are 0.5, 0.6, 0.8, 0.8, 1*m/s* respectively. Performance is determined by using the total time to eliminate all the alarm[9].

Dong-Wook Lee and Kwee-Bo Sim have performed their experiment in simulation environment. Two different experiment are carried out in this research, simulation with swarm algorithm applied and simulation without swarm algorithm applied. A total of 50 robots are distributed in a 10m*10m area and they are required to perform a number of tasks which are 10, 100, 200 and 500[7].

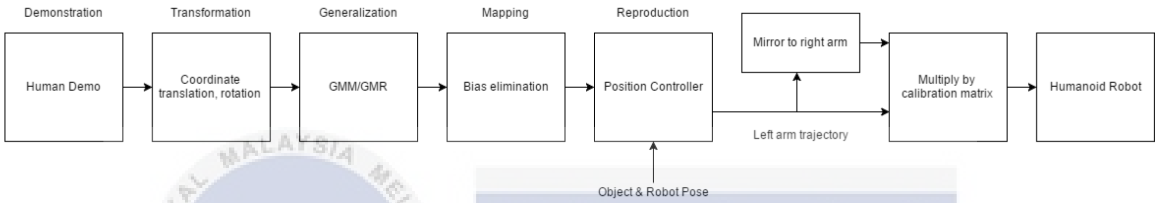
Among the three papers, N.R. Ramli and her team proposed the concept and testing methodology, but do not provide any figure or result for the method proposed[4]. The performance index of immune-based algorithm and greedy algorithm are compared. Immune-based algorithm used the total time of 60.14s while greedy algorithm use 65.40s to complete the experiment [9]. Based on the graph provided by Dong-Wook Lee and Kwee-Bo Sim, simulation without swarm algorithm, robots make decision based on local information and select strategy itself[7]. After apply the algorithm, it is clearly shown that each robot decides strategy by mutual relationship.

2.5.2 Object Manipulation between Human Operator-Robot

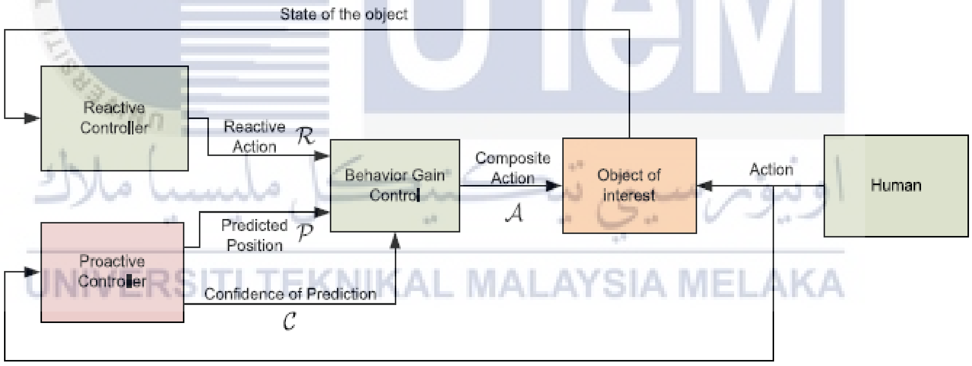
Human-robot collaboration (HRC) is a research field with a wide range of applications and high economic impact [39]. For human operator-robot in object manipulation, human operator can easily receive or accept object from robot, yet there are challengers for robot to perform this manner [38].

P.Evrard and etc. prepared two sets of scenarios to be demonstrated by the robot. For first scenario, human operator closes eyes and the user initiate and end the task. The other scenario at which the role inverted and act as leader. Then the robot is asked to demonstrate task the same as the teaching phase [39].

The experiment conducted by W.Sheng and etc. is using Nao robot. The experiment started with motion capture by the robot to provide the position and motion information of table. Then, by using proactive controller and reactive controller to calibrate the movement of the robot. Figure 2.1(a) shows the block diagram of imitation learning phase. Figure 2.1(b) shows the frame work for second stages [19].



(a) The frame work for phase 1 of table lifting.



(b) The frame work for phase 2 of table lifting.

Figure 2.1: The frame work of table lifting. [42]

P.Evrard and etc. perform lifting task 71 trials and humanoid robot successful replicate for 38 times[39]. For W.Sheng and etc., robot is tested 100 times, the result is obtained in root mean square error (RMSE) [19]. It is clearly shown the trial with prediction has lower error compare to the trial without prediction. Comparing two method mentioned, The method proposed by W.Sheng and etc. have better controller compare to another method.

2.6 Significance

All the definition on stability and control method proposed were mainly for better understand and improve the motion of robots in completing the movement. The result obtained from these researches can benefit human in different field mainly manufacturing sector.

An example of the application is in building and construction environment. Humanoid robot can help in transporting building material in construction sites. Besides, with the development of the humanoid robot, there are possibility for robot to replace human in dangerous zones [6]. This can further reduce the injury rate and mortality rate of construction workers.

The successful of this research can contribute in medical field as well, the lifting of casualties to emergency stretcher. Hence, robot can reduce heavy load handled by caregivers at nursing home and hospital [35].

Besides, for human robots which own the quality that able to cooperate with other robot, they are able to work with military in inspection when necessary. They also can help in remove obstacle when they encountered for example fallen tree trunk.

2.7 Summary

From the literature review, the two main problems which are communication and controller can be tackle from different method. Sensors are playing important role in regulating the velocity of the humanoid robots. Hence, both of the robots will be influenced in terms of velocity whether to be accelerate, decelerate or remain the velocity to ensure the object is always stable. In terms of controller, there are various situation have been analysed, robot manipulate object, robot-robot manipulate object together as well as human-robot manipulate object together. In order to coordinate the cooperation of the robots, PID controller is the most suitable system for this case.

The target of this study has been verified to be the communication network and perform cooperative behaviour in humanoid robots while completing tasks. Between the issues that being proposed which are communication and humanoid robot control method, this study will focus on how the humanoid robots are cooperated and completed the task given. The task is assigned to be lifting object together between two robots.

Stability of the robots and object are the key variable which are going to be analyse in this project. The stability of the robot is the main concern for humanoid robot while performing any motion and movement. To determine whether the lifting is success, sensors like gyroscope and Force Sensitive Sensor (FSR) are implemented to obtain the data and graph to show the stability of the robot.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

In this chapter, system configuration, the proposed method and methodology to verify the proposed method are discussed. The region boundaries are explained in system configuration. The proposed method begins with the general overview of the method before explaining the communication or signal transfer, followed by robots' feedback and motion. The communication is depending on the signal transfer by sensor to both robots. The feedback and motion phase is the phase where IF-THEN rules are applied in order to regulate their velocity of the joint to maintain the stability of the object manipulated. In the research methodology, discussion on the general system of the Nao robot and its embedded sensory unit were made. The general experimental setup using the Nao robot was stated towards the end of the chapter.

Chapter 1 and chapter 2 indicates the research background, objectives, problem statement, scope, lists of contribution and literature review. The algorithm and the list of simulations for the analysis will further discussed in subtopics in chapter 3.

3.2 System Design Phase

For the system design phase, the first stage would be construct an algorithm with the help of flow chart. Then, build up coding and finally simulate using V-rep environment.

3.2.1 The Trajectory of Robot's Arm

Trajectory refers to a time history of position, velocity as well as acceleration of a joint[43]. Equation (3.1) shows the general formula for the cubic polynomial. The joint velocity and acceleration for the joint are as stated in equation (3.2) and equation (3.3) respectively.

$$\theta(t) = A + Bt + Ct^2 + Dt^3 \tag{3.1}$$

$$\dot{\theta}(t) = B + 2Ct + 3Dt^2 \tag{3.2}$$

$$\ddot{\theta}\,(t) = 2C + 6Dt \tag{3.3}$$

There are two assumptions made to solve the cubic polynomial. The assumptions are stated as follows:

1. Initially, t = 0, the θ (0) and the velocity, $\dot{\theta}$ (0) = 0.
2. At the end of the trajectory, t = t_f the θ (t_f) and the velocity, $\dot{\theta}$ (t_f) = 0.

After considering the assumption, the four coefficients in the polynomial are described as in equation(3.4), equation (3.5), equation (3.6) and equation (3.7).

$$A = \theta_0 \tag{3.4}$$

$$B = 0 \tag{3.5}$$

$$C = \frac{3}{t_f^2}\,(\theta_f - \theta_0) \tag{3.6}$$

$$D = \frac{2}{t_f^3}\,(\theta_f - \theta_0) \tag{3.7}$$

Figure 3.1 shows the position, velocity and acceleration graph for cubic polynomial trajectory.

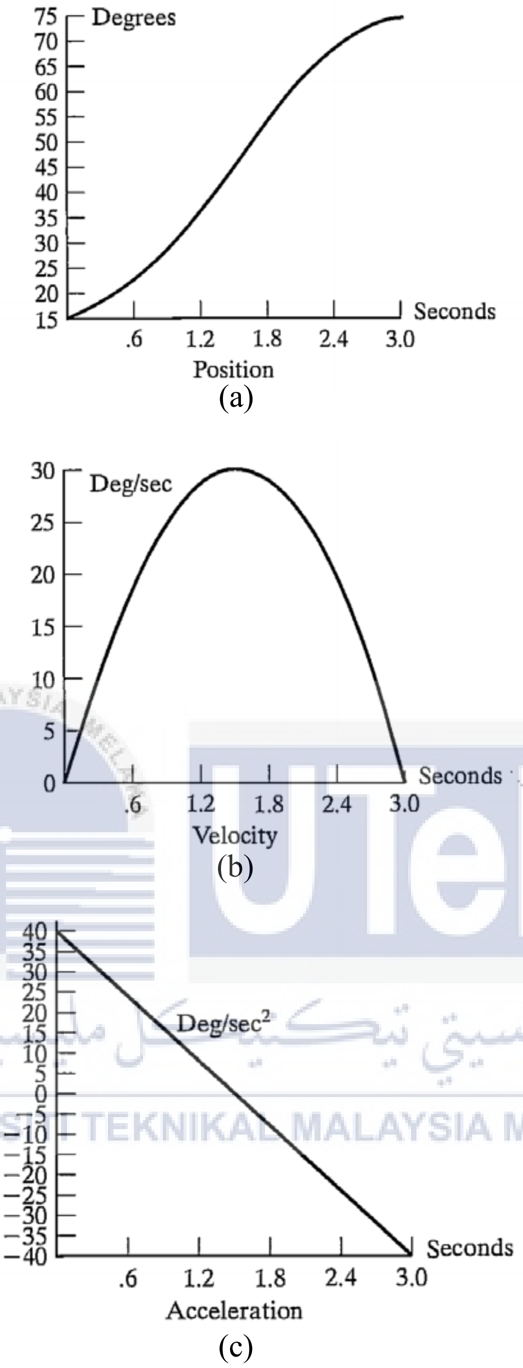


Figure 3.1: The position (a), velocity (b) and acceleration (c) profile for cubic polynomial[43].

3.2.2 The Proposed Communication and Feedback Method

Figure 3.2 shows the overview of the proposed method to perform communication and feedback motion. This method is designed with two main stages which are (1) Communication and (2) Feedback System/Self-Regulating System.

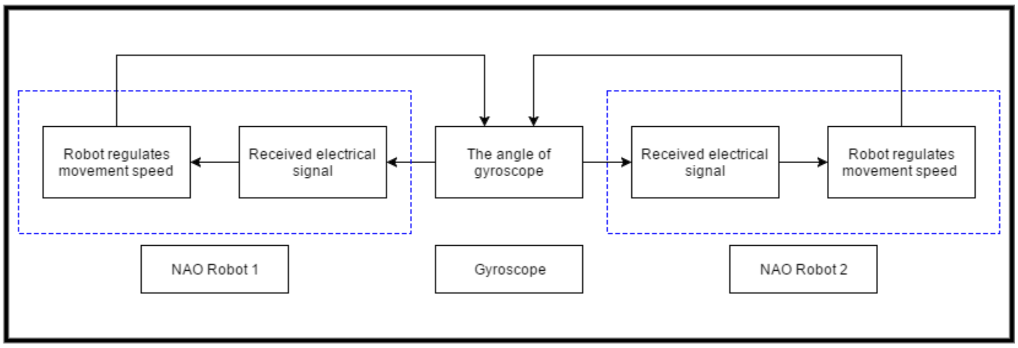


Figure 3.2: The function diagram for overall system

The communication agent is gyroscope at which the angle values are sent to both Nao robot and regulation will be made by referring to partners' arm velocity. After receiving the electrical signal sent by gyroscope, IF-THEN rules used as the action selection controller to change the velocity of robot's arm. With associated with PID controller and take reference arm velocity into account, the velocity of the arm will be regulated and restore the balance of the lifted object. The flow is as shown in Figure 3.3.

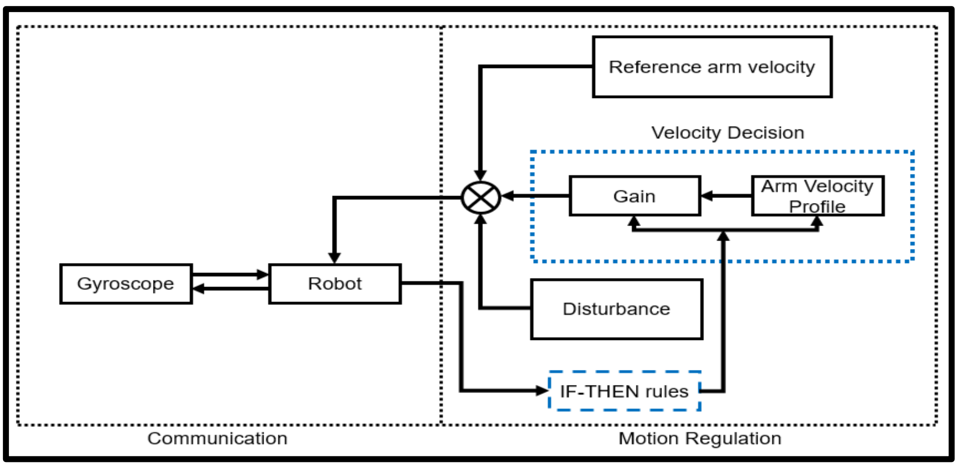


Figure 3.3: The function diagram of the relationship in between gyroscope and Nao robot.

Figure 3.4 shows the flow chart for the regulation of velocity during cooperation. When gyroscope sense either one side of the box lifted higher, the gyroscope will transmit the deviate value to both Nao robot, the Nao robot which lifted higher will remain constant velocity while the other Nao robot will increase its speed. As shown in Figure 3.4, Nao robot arms will regulate according to the deviation of the gyroscope by using Pythagoras Theorem and velocity formula ($v = \frac{d}{t}$). The further regulation will be conducted by using suitable PID controller. Until the robot arm moved to certain degree, the whole operation stops.

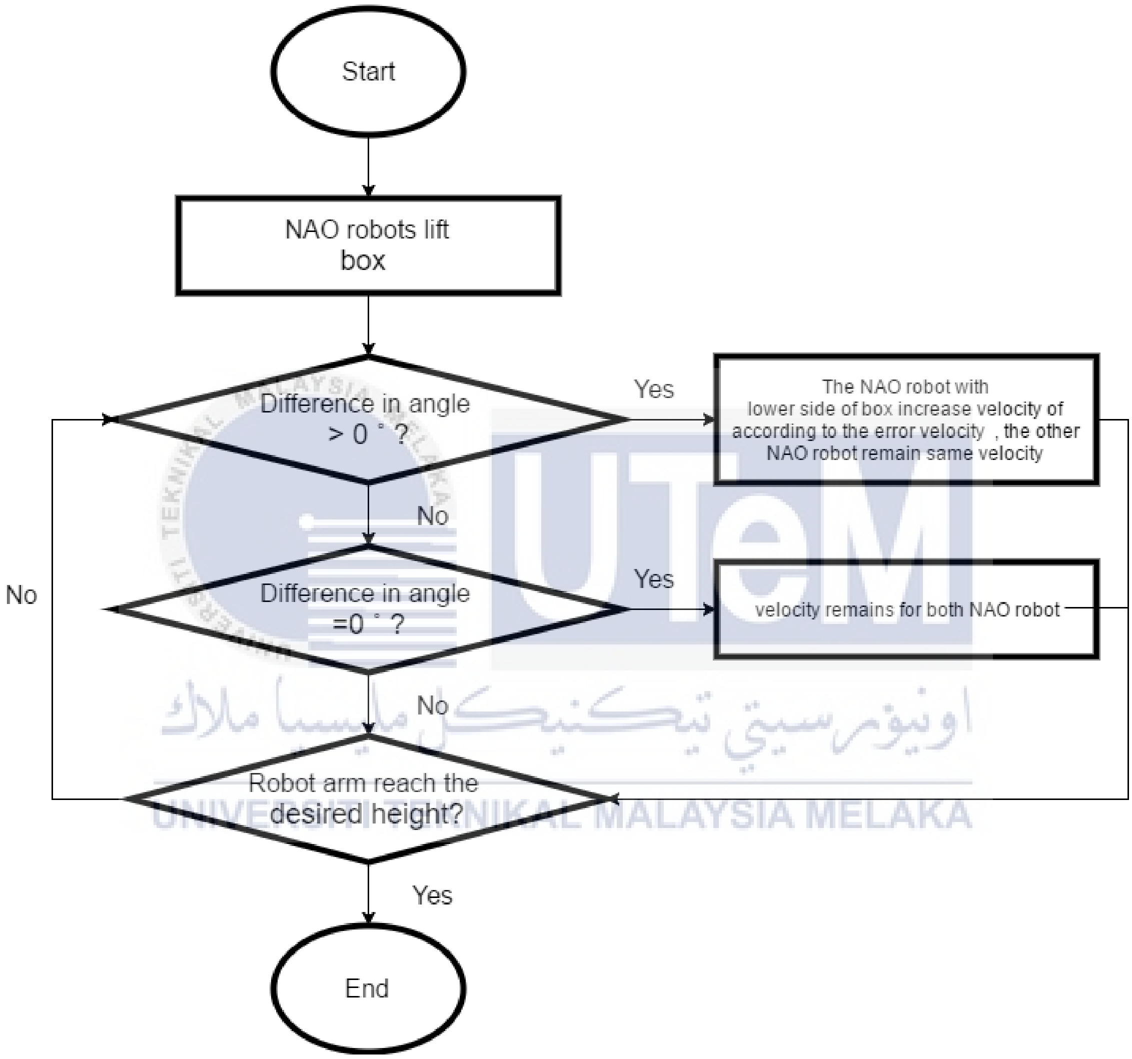


Figure 3.4: The flow chart for the system.

3.2.2.1 IF-THEN Rule

Generally, the rules are set based on gyroscope's reading. The dependant variable, velocity will change according to the gyroscope's reading. In principle, the dependant variable are set to ensure that the velocity of robots' arms are always regulate according to the stability of the object lifted.

The IF-THEN rules used as the action selection controller, it was set as follows:

IF: G(left) > G(right)
THEN: 1. Left Nao velocity remain 2. Right Nao velocity increase
IF: G(left) < G(right)
THEN: 1. Left Nao velocity increase 2. Right Nao velocity remain
IF: G(left) = G(right)
THEN: 1. Velocity remain for both Nao robot

3.2.2.2 PID Controller

Proportional-Integral-Derivative (PID) controller is the most common used controller in trajectory control of robot. The controller consists of three coefficients which are proportional, derivative and integral. The three components help in reduces the differences between actual and desired trajectory. Equation (3.8) shows the general equation for PID controller:

$$u(t) = K_p\, e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t) \tag{3.8}$$

Where: K_p = the gain for proportional.
K_d = the gain for derivative.
K_i = the gain for integral.
$e(\tau)$ = the differences between the actual and desired data

PID controller can perform function for PD, PI and PID controller. Proportional component helps in increase the speed of the control system response, yet oscillation will become larger when K_p is too large and lead to unstable system. Integral component used to eliminate steady state error of the system. However, integral component will create overshoot when error equal to zero. The derivative component enables controller additional control action when error changes consistently. There are multiple methods used to tune PID controller, for example Ziegler-Nicolas method, Pole Placement Method, Pole Zero Cancellation Method and others[44], [45].

For this system, average RMSE value is used to defined the proportional gain, K_p. Then, Ziegler Nichols tuning method is applied to obtain the PID. Equation (3.9) shows the formula for Ziegler Nichols damped oscillation method[46].

$$u(t) = K_p \left(e(t) + \frac{1}{Ti}\int_0^t e(\tau)\, d\tau + T_d \frac{d}{dt} e(t)\right) \quad (3.9)$$

Where: K_p = the gain for proportional.

T_i = the time for derivative.

T_d = the time for integral.

3.3 System Implementation Phase

For system implementation phase, there were various programming platforms can be connected with V-rep environment including Python Spyder, Matlab and Microsoft Visual Studio. The characteristic of each platform is shown in Table 3.1.

Table 3.1: The characteristic of different programming language platform.

Characteristic	Python Spyder	Matlab	Microsoft Visual Studio
Programming Language	Python	C, C++, C#, Java, Fortran & Python	C, C++, C#
Product Description	An integrated development environment (IDE) for scientific programming in the Python language.	A platform which optimized in solving engineering and scientific problems which can interfaces to various programming language.	An integrated development environment (IDE) from Microsoft which enables tool sharing and eases the creation of mixed-language solutions.

From the three softwares compared in Table 3.1, the three software have their own advantage. Both Matlab and Python Spyder can develop coding and perform analysis whereas Microsoft Visual Studio focuses on completing full system. In this project, Python Spyder is chosen to develop the coding for the humanoid robot.

3.4 System Configuration

Based on the proposed method, there are two types of variable needed to be determined and fixed which are constant variable and measured variable.

The constant variable included the number of robots, the weight of object, the height of the platform and the region boundary. For measured variable, PID gain, the stability of humanoid robot based on 0the value of Force Sensitive Sensor(FSR), the stability of object based on the reading of gyroscope and the velocity of robot's arm.

3.4.1 Number of robots

The number of robot is set to two humanoid robots. Both single and two humanoid robots will undergo lifting task, but different sensor will evaluate the stability of the robots. The reason the stability simulation is done using different number of robots is to prove two humanoid robots can reduce the burden on robot as compared to single robot.

3.4.2 Weight of Object

Weight of the object is set to be 0.5 kg[47]. The simulation will start with single humanoid robot. Besides measure the stability of the robot, the success rate for Nao robot to lift the weight is also take into consideration. The weight of the object will be one of the key to verify the third objective which is to prove two humanoid robots can perform better compared to single humanoid robot.

3.4.3 Region Boundaries

The simulation is conducted using Nao robot that regulate its motion every time they receive signal from sensor until the object successfully reached a certain height. The area of interest is only focus on the arms of the Nao robot. This is due to the motion of the Nao robot only involved shoulder and elbow when lifting object. The workspace of the humanoid robot's shoulder joint set to be maximum 170°in pitch rotation and no yaw rotation involved as shown in Figure 3.5. For elbow joint, the joint will involve only in roll rotation for maximum 105° as shown in Figure 3.6.

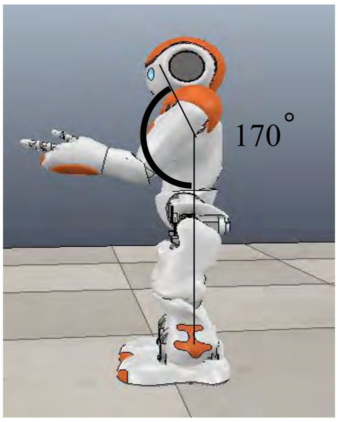


Figure 3.5: The maximum angle the robot arm can lifted.

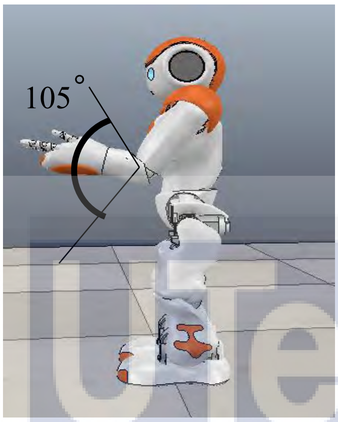


Figure 3.6: The maximum angle the robot elbow can lifted.

3.4.4 Measured Variable

The measured variable in this project are mainly focus on the analysis of stability of robot and object. Stability is analyse based on the graph generated by Force Sensitive Sensor and gyroscope. Each Nao robot's feet are equipped with 4 Force Sensitive Resistor. The gyroscope is attached to the object when the simulation involved 2 humanoid robots in lifting. The analysis is made based on the fluctuation of the graph.

3.5 Research Methodology: Simulation Analysis

Several simulations are conducted to validate the method proposed. Two Nao robots are used in conduct the simulation. Nao robot is a suitable tool in this simulation due to its human-like structure and they are able to connect to various sensors. Nao robot as shown in Figure 3.7 is a robot with 25 DOF.

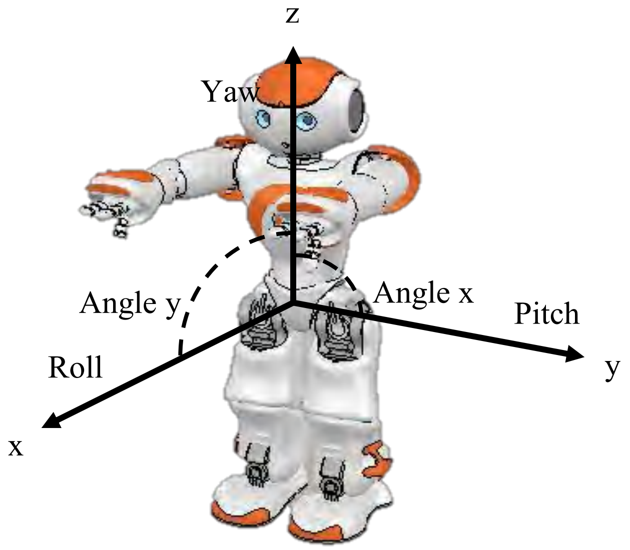


Figure 3.7: The picture of Nao robot.

Nao robot's arm are made up of six joints which are shoulder pitch joint, shoulder roll joint, elbow yaw joint, elbow row joint, wrist yaw joint and hand. Too much of degree of freedoms increased the complexity during lifting motion. In order to reduce the complexity, shoulder pitch joint and hand are the only joints which chosen to be moved during the object manipulation simulations.

3.5.1 Gyroscope

3-axis gyroscope is used throughout all the simulation planned. A gyroscope is a type of sensors or devices which measure the angular velocity of an object with respect to the inertial frame [48], [49]. The gyroscope is planned to attach on the box which will be lifted by Nao robot as shown in Figure 3.8. The angle values are computed from the sensor for Nao robot to regulate its movement and maintain the stability of the box.

Gyroscope is used as the communication medium instead of master-slave method is due to the master-slave system will fail if the master in the system experiences failure. Hence, the whole master must be replaced and repair. However, gyroscope experiences failure, the system can recover by just changing the gyroscope attached on the object.

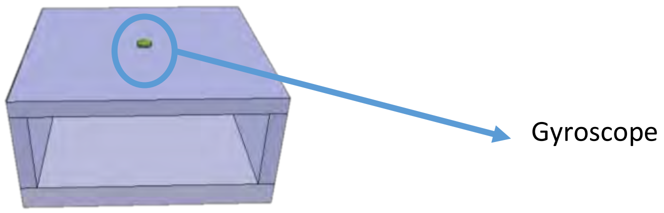


Figure 3.8: The position of gyroscope attached on the object.

For the simulations planned, the box is set to lift 0.055m from the platform as shown in Figure 3.9. The lifting only affected Z-axis throughout the simulations while stability will focus on x-axis.

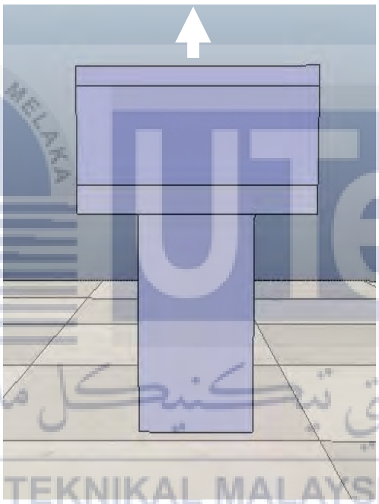


Figure 3.9: The direction of box lifted by humanoid robot.

3.5.2 Force Sensitive Resistors

Force Sensitive Resistors located at the foot of the Nao robot. Each foot equipped with four FSR, which are FL ,FR ,RL and RR. as shown in Figure 3.10. The value return from the FSR will be used as the supporting evidence for robot stability.

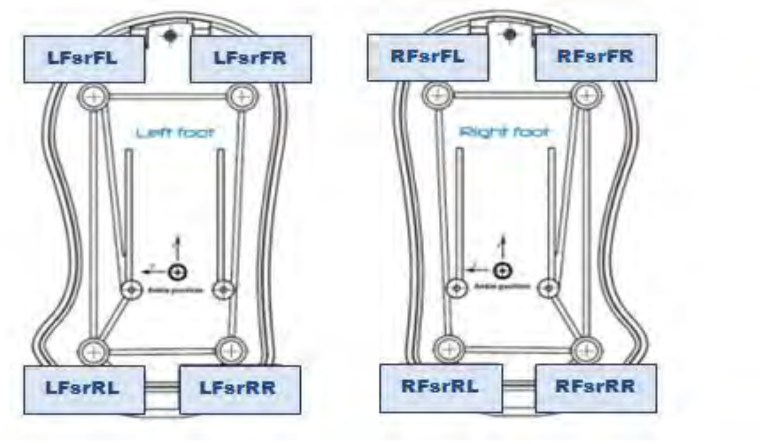


Figure 3.10: The distribution of the FSR at the feet of Nao robot.

The equation used to calculate the balance of the robot is as shown as equation (3.10) whereas Figure 3.11 shows the front and back region of a Nao robot foot[50].

$$R = \frac{1}{M_1 + M_2}(M_1 X_2 + M_2 X_1) \tag{3.10}$$

Where: M_1 = The average FSR value for front region

M_2 = The average FSR value for back region

X_1 = The length between the tip of the foot and the middle region,0.1m

X_2 = The length between the middle region to the heel,0.05m

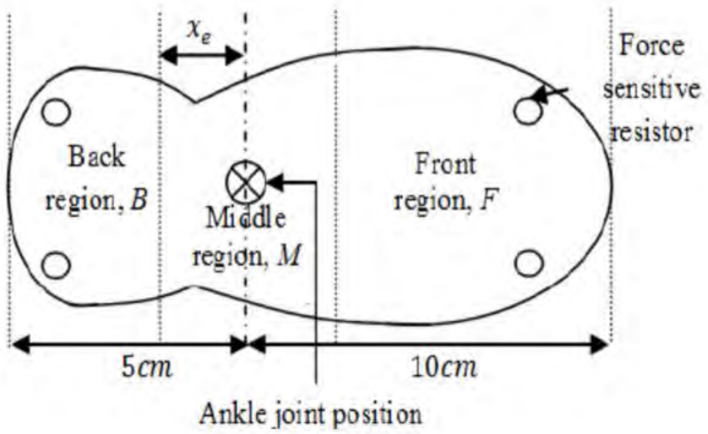


Figure 3.11: The defines of the region for Nao foot[50]

3.5.3 General Experimental Setup

There are in total of 3 simulations planned in order to achieve the objectives. The first simulation is mainly focus on trajectory planning which in line with the first objective.

Second simulation is conducted to design a suitable controller to be implemented in the system. The comparison will be performed in between system with implementation of controller and without controller based on object orientation and force sensitive resistor(FSR)'s reading. The comparison of result for both simulation is in line with the second objective of the project at which to design a better control system to allow robots to perform cooperative behaviour.

The third simulation is conducted to compare the performance between single humanoid robot and two humanoid robots with controller implementation. During the comparison, the Nao robot is required to lift the box with different weight. Third simulation is in line with the third objective.

3.5.3.1 Trajectory Planning using Velocity and Cubic Polynomial

As a preparation for other simulations, a proper trajectory of robot arms is important to ensure robot's movement are smooth whist perform lifting. Hence, both velocity and cubic polynomial equation are implemented for trajectory planning. This simulation is in line with the first objective stated in Chapter 1 which is planning a proper trajectory for Nao robot. For this simulation, Nao robot will lift its arm without any object hold in its hand as shown in Figure 3.12. The analysis on the trajectory will be based on the angular velocity versus time graph. The ideal graph curve is as shown in Figure 3.1(b).

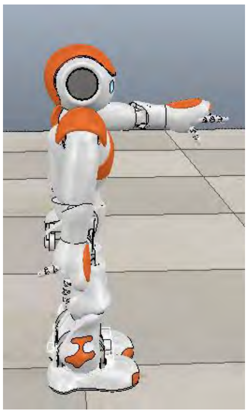


Figure 3.12: The Nao robot lifts its arm with proper trajectory.

3.5.3.2 Single Humanoid Robot Simulation

This simulation will be conducted to compare with performance of two Nao robots in object manipulation. The simulations run in the V-rep environment, the first step is to set up the simulation environment as shown in Figure 3.13. The box is placed 0.265 m above the ground. The box with dimension of 0.30m × 0.28m × 0.15m is set to 0.5kg and 2.5kg respectively.

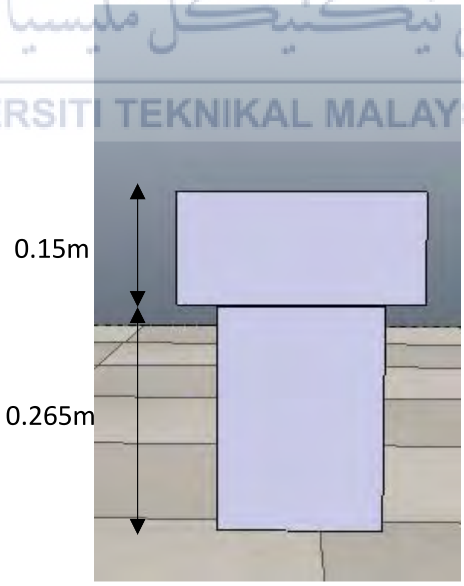


Figure 3.13: The environment setup for experiment 2 and 3.

3.5.3.3 Collaboration of Humanoid Robot with and without the Regulation from Controller Simulation

For second simulation, the objective is to validate the performance of the robot with PID controller is better than the robot without controller.

Similar to first and second simulation, the setup is the same but the amount of Nao robot is increased to two Nao robots. The first step is to set up the simulation environment as shown in Figure 3.14. The box is placed 0.265 m above the ground. The boxes with dimension of 0.30m × 0.28m × 0.15m is set to the weight of 0.5kg. The gyroscope is attached to the box as shown in Figure 3.8.

The analysis of the result will be based on the stability of the lifted box. Besides provides feedback to Nao robots, the gyroscope also generate graph that can be used to analyse and calculate performance of the system by analyse the angle deviation from horizontal component.

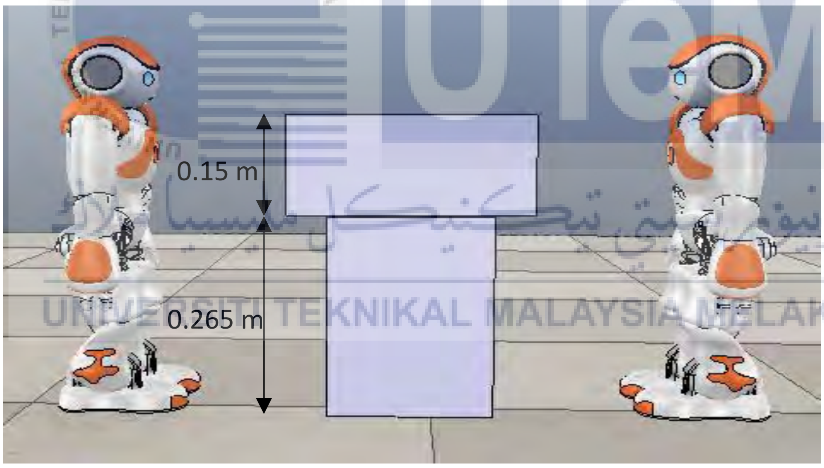


Figure 3.14: The environment setup for simulation 2 and 3.

The analysis of the result will be based on box orientation and Force Sensitive Resistor (FSR) which able to determine the stability of the box and Nao robot. The error velocity is obtained from equation (3.11).

$$\text{Velocity} = \frac{d}{t} \qquad (3.11)$$

Where: d = distance

t = time

The distance is obtained from the error angle, and by substituting the error angle to Pythagoras Theorem to get the vertical distance as shown in equation (3.12).

$$\text{Distance} = \sin\theta * h \tag{3.12}$$

Where: θ = Error angle

h = Hypotenuse, the length of the box

The deviation angles obtained from box orientation is used to calculate average root mean square error (RMSE) value. The smaller the average RMSE value is, the better the performance of the controller. Equation (3.13) shows the equation used to calculate average RMSE value.

$$\text{Average RMSE Value Formula}=\sqrt{\frac{1}{\mathrm{m}}\sum_{\mathrm{i}=1}^{\mathrm{m}}\left(\frac{\sum_{\mathrm{k}=1}^{\mathrm{N}}\left(e^{(i)}(\mathrm{k})\right)^{2}}{\mathrm{N}}\right)} \tag{3.13}$$

Where: m = The number of simulations

N = The amount of sampled data

$e^{(i)}(\mathrm{k})$ = The error functions of i^{th} simulation

3.6 Summary

The method proposed has two phases where Nao robot will regulate shoulders' velocity according to the signal emitted by gyroscope. According to the changes in gyroscope, the stability can be determined and the signal can be transmitted to Nao robot's shoulder. The signal received by Nao robot will regulate and compensate similar to a closed loop system. All the simulation will repeat for 5 times to validate the result obtained. The analysis will be done on the graph generated by FSR and gyroscope.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

There will be 3 simulations conducted for this project and the result will be discussed in this chapter. To better understand the study, the discussion. The experimental data are acquired from gyroscope and FSR using Python Spyder and V-Rep environment.

4.2 Implementation of Proper Trajectory

The objective of this simulation is to generate a proper trajectory which can be applied on Nao robot. There are two different method are used to test the motion, which are based on velocity and based on cubic polynomial trajectory equation.

4.2.1 Result

Based on the result, both method can perform the motion completely. Figure 4.1 shows the changes of velocity using velocity method while Figure 4.2 shows the changes of velocity using cubic polynomial equation which applied on Nao robot.

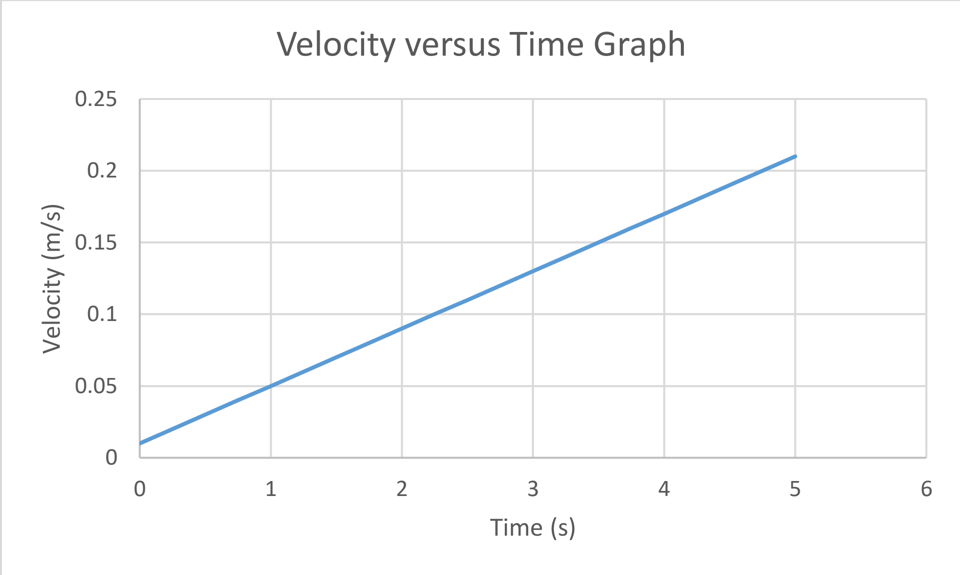


Figure 4.1: The velocity-time graph using velocity.

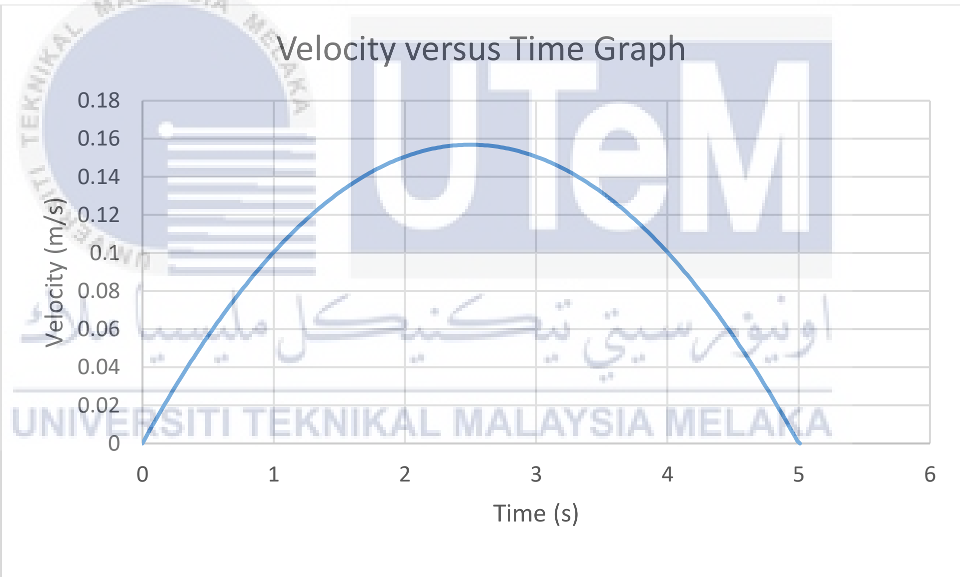


Figure 4.2: The velocity-time graph using cubic polynomial equation.

4.2.2 Discussion

According to graph in Figure 4.1 and Figure 4.2, the simulation shows both Nao robots' shoulder can move smoothly until reached target angle which is 30˚ from initial position. For Figure 4.1, the initial velocity can be set according to the user. In this case, the initial velocity is set to be zero and for every loop the velocity is set to increase the velocity by 0.01m/s. Hence, the velocity is directly proportional to the time, the motion of the shoulder will be getting faster and hard to coordinate during cooperation with another Nao robot.

For Figure 4.2, the velocity of the shoulder is controlled using cubic polynomial equation. Compared to velocity-based motion control, cubic polynomial equation easy to coordinate when there are disturbances that influence the velocity of motion. To perform better in object manipulation, cubic polynomial equation should be implemented for robot motion.

4.3 Implementation of PID Controller

The objective of this simulation is to develop a suitable controller suitable for the system. The simulation starts with determine a suitable proportional gain, then compare with the system without controller. Then, identify the PID controller from data obtained from Proportional controller using Ziegler-Nichols tuning method.

For this project, proportional controller is determined by the minimum average RMSE value. Then the K_p is then used as the ultimate gain, K_u in Ziegler Nichols damped oscillation tuning method to obtained the PID value.

4.3.1 Proportional Gain Coefficients Identification

Proportional gain is tuned and obtained from lowest RMSE. The objective of this simulation is to observe the effect of gain tuning on the performance of the Nao robot. The coefficient will influence the gain sent to the new velocity generated by the controller.

Initially, the proportional gain is $K_p = 1$, at which there are no proportional controller implemented to the Nao robot. Then, to identify the suitable proportional gain, the range of K_p is set between 0.5 and 1.5. The gain is identified by selecting the gain which can produce lowest RMSE.

The value of RMSE for each gain is retrieve started from 1.47 second to 5.0 second due to the gain only affected the motion within that particular period. 10 trials are conducted for each of the test with different gain coefficient. In this simulation, the disturbance range is set to be [0,0.1] with constant object weight of 0.5kg .

4.3.1.1 Result

The simulation is repeated for 10 times and the average RMSE values are calculated based on formula. Table 4.1 and Figure 4.3 shows the comparison of average RMSE value for different gain coefficient within the range, $0.5 < K_p < 1.5$.

Table 4.1: The average RMSE value for each K_p

K_p	RMSE
0.5	1.2226
0.6	1.1430
0.7	1.0794
0.8	1.0489
0.9	1.0648
1.0	1.1035
1.1	1.1086
1.2	1.1148
1.3	1.1179
1.4	1.1361
1.5	1.1539

Average RMSE value reached minimum when gain coefficient is 0.8. The RMSE decrease starting from 0.5 to 0.8 and increase afterward. The average RMSE value for K_p=0.8 is 1.0489 while for K_p = 1.0, the average RMSE value is 1.1035.

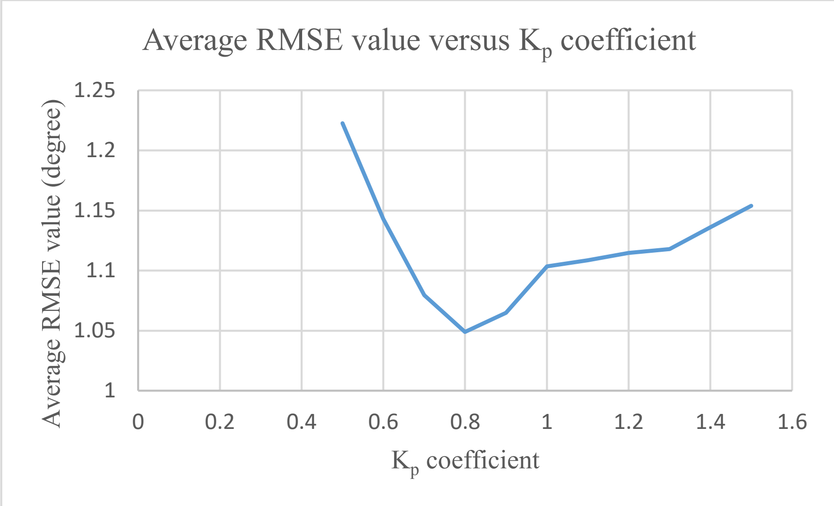
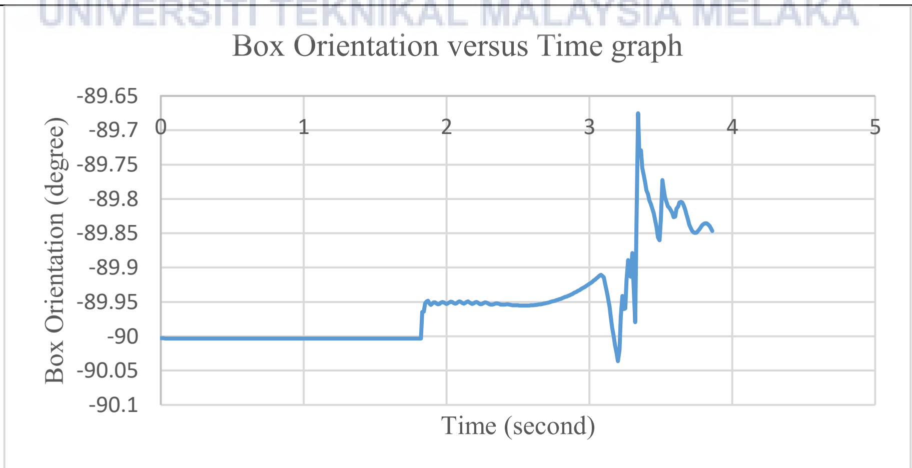


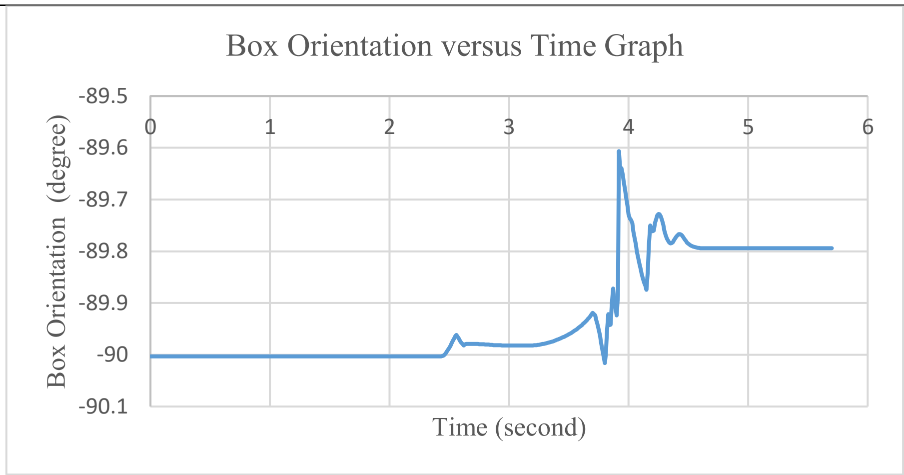
Figure 4.3: The average RMSE versus gain coefficient within the range of 0.5 to 1.5.

To validate the performance of the system with controller is better than system without controller, the disturbance is set to the same value which is 0.1. The performance of object orientation is shown in Table 4.2.

Table 4.2 shows the box orientation scenario for both K_p=0.8 and K_p=1.0 at which the disturbance is set to the same value. The box orientation shows more negative value if the object slanting from Nao robot 2 (right) to Nao robot 1 (left) and vice versa.

Table 4.2: Box orientation for K_p = 0.8 and K_p = 1.0

K_p	Box
0.8	

1.0	

Based on Table 4.2, The fluctuation range for K_p=0.8 is from -89.68° to -90.04°at which the difference is 0.36°. The fluctuation range for K_p=1.0 is from -89.61° to -90.02°at which the differences is 0.41°. The box orientation range for K_p=0.8 is 12.2 % less than the box orientation range for K_p=1.0.

Table 4.3 shows the joint position and velocity for both K_p=0.8 and K_p=1.0 at which the disturbance for both Nao robots are set to be 0.1. Both of the graph has identical trend, at which the position of the joint stopped when the joints moved 35°from initial position. The velocity of Nao robot 1 and Nao robot 2 for both scenario shows rough parabolic shape of the graphs are similar to the trajectory planned in simulation 1.

Table 4.3: Position vs Time Graph and Velocity vs Time Graph for $K_p = 0.8$ and $K_p = 1.0$

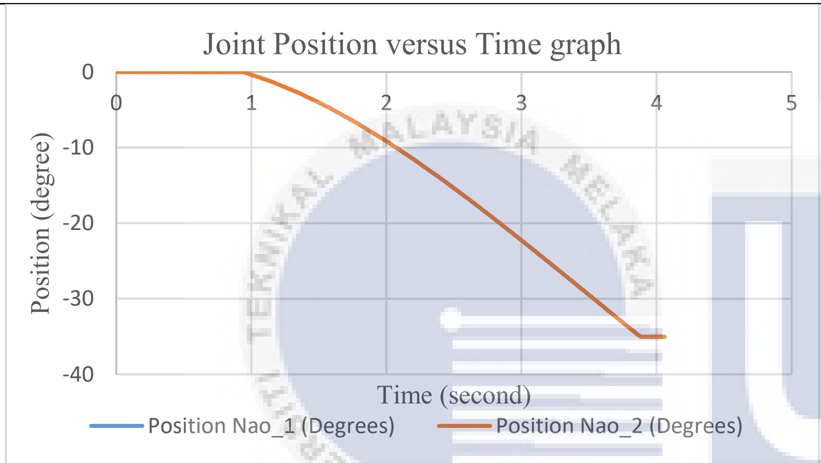
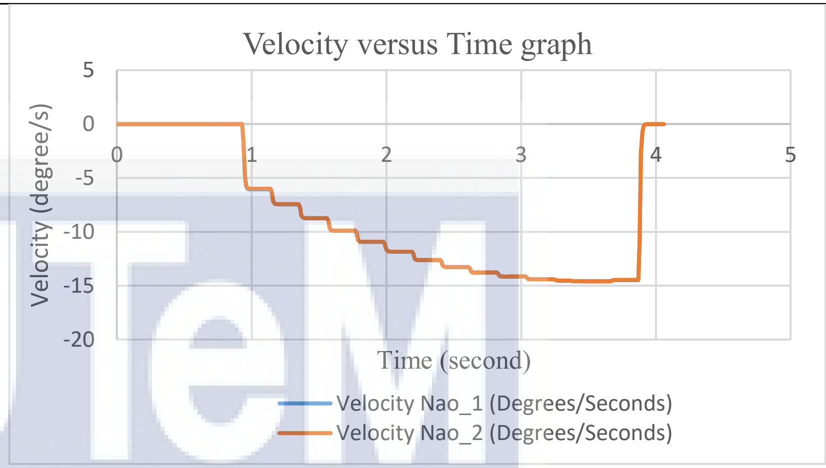
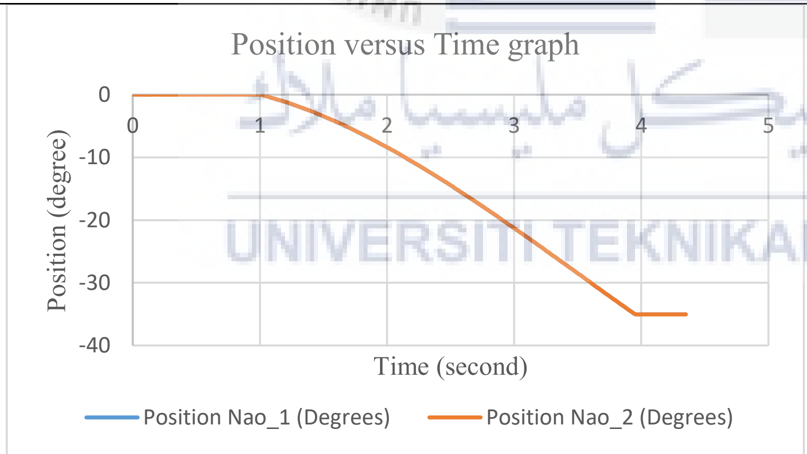
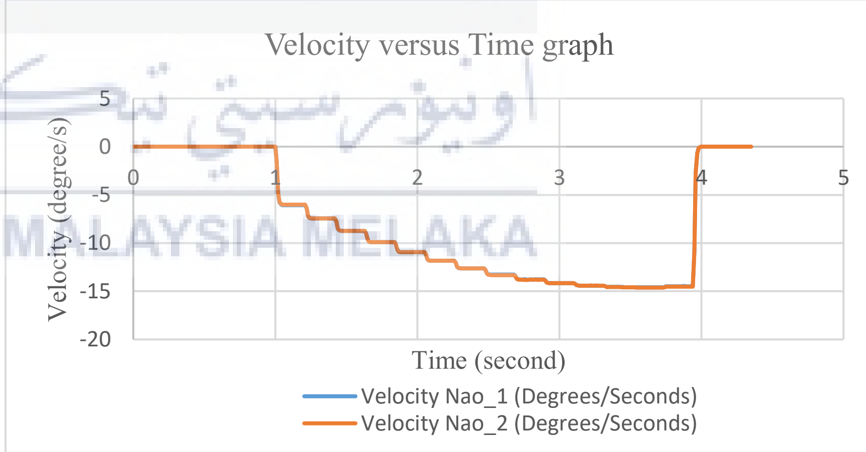
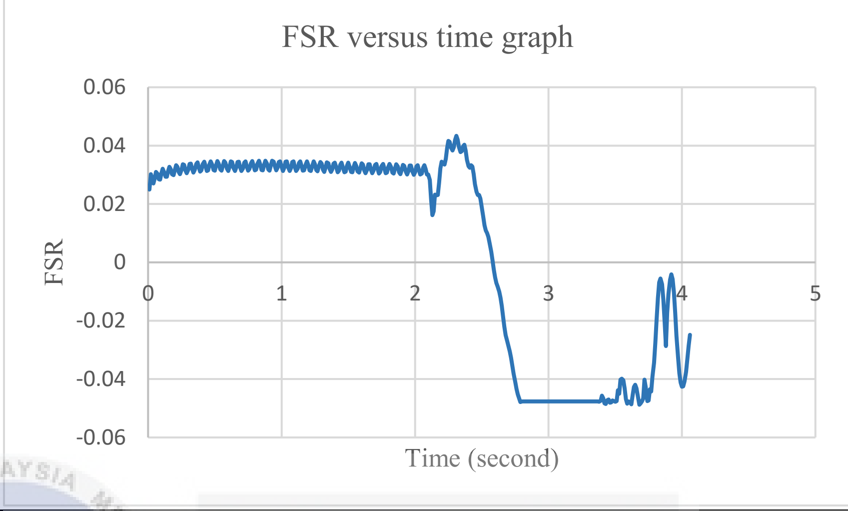
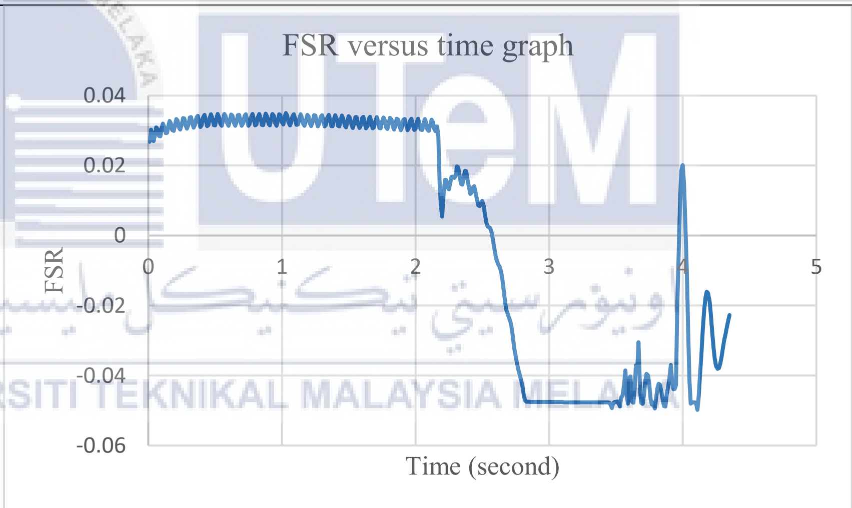
K_p	Position	Velocity
0.8		
1.0		

Table 4.4 shows the graph of FSR for both scenario, the Nao robot 1 for K_p=0.8 and K_p=1.0 started to change starting from 2.07 second and 2.17 second respectively.

Table 4.4: FSR graph for K_p = 0.8 and K_p = 1.0

<table>
<tr><th>Kp</th><th>FSR</th></tr>
<tr><td>0.8</td><td>
</td></tr>
<tr><td>1.0</td><td>
</td></tr>
</table>

For K_p=0.8, the Nao robot 1's FSR started to rise at 2.16 second after a drop for 0.09 second until 2.31 second with value 0.0434. Then, the FSR dropped to -0.048 at 2.79 second and stayed constant until 3.38 second. The graph started to fluctuate until the lifting motion stopped at 4.06 second. In the simulation, the Nao robots do not show any sign of falling.

For K_p=1.0, the FSR started to drop at 2.17 second with 0.0332 until 2.2 second with a value of 0.0054 then, the FSR rose to a small peak with 0.0198 at 2.31 second. FSR value is then dropped to -0.48 at 2.83 second and stayed constant until a fluctuation started at 3.51 seconds. The graph fluctuated until the lifting motion stopped at 4.35 second. In the simulation, the Nao robots in the scenario do not show any sign of falling.

4.3.1.2 Discussion

Both graph in Table 4.2 shows up and down fluctuation due to the regulation of the box orientation. Despite the system regulated the velocity each time when there are errors, the internal and external disturbances existed and affected velocity of joints. However, these disturbances can be minimized but cannot be eliminated completely. Hence, proportional gain tuning is used to minimized the average RMSE value.

Joint orientation is deviated 5 ° from desired angle which is 30 ° because of the loop of the coding inserted. The angle of last loop is fulfilling the condition which is slightly less than 30° from initial orientation. Hence, the loop continued to execute the code and cause the orientation to be increase and exceeded the desired angle.

The velocity for both system without controller and with proportional controller have identical graph trend as the trajectory planned in simulation 1. It is proved that Nao robots are performing cubic trajectory as planned in objective 1. The reason the graph is not perfectly follow cubic trajectory velocity graph is disturbance added in the system. The disturbance regulation of shoulder joint has increase the velocity to reach target orientation before the time planned.

The adjustment of gain coefficient improved the performance of the system by reduced the average RMSE. This statement is supported by the orientation of the box. Based on Table 4.2, the fluctuation of the box for K_p = 1.0 is greater than the fluctuation of the box for K_p = 0.8. This can be proved by the range of fluctuation, system with K_p = 0.8 has the changes of 0.36 degree while system with K_p = 1.0 has the changes of 0.41 degree. The smaller the fluctuation, the better the stability of the box. By setting the disturbance to constant for both Nao robot, the improvement induced by the proportional gain can be observed clearly.

For the stability of the robot, both FSR values show stable according to Table 4.4. To compare the stability for K_p = 0.8 and K_p = 1.0, both system with and without controller does not show any sign of falling and the value is smaller than -0.05 or exceeded 0.1, hence, Nao robots is deemed to be stable. However, the fluctuation occurred at the end of the lifting shows that system with K_p = 0.8 has better stability than system with K_p = 1.0 during lifting process.

From the evidence from the fluctuation range of the system plus the robots is stable, proportional controller with K_p = 0.8 is shown better than system without controller.

4.3.2 PID Tuning and Comparison with Other System

Derivative gain and Integral gain is tuned using Ziegler Nichols tuning method. From the proportional gain coefficients obtained from section 4.3.1, K_p=0.8 is then used as K_u in order to identify the derivative gain and integral gain. For the coefficient of P_u, it can be obtained from the graph for K_p=0.8 illustrated in Table 4.2. After obtaining the PID, system without controller, system with proportional controller and system with PID controller will be compare in terms of performance to obtain the suitable controller for the system. The comparison will be conducted with same random number to validate the performance of the three controllers. Then the controller will be compared by conducted simulation with random number with the range of [0,0.1], [0,0.2] and [0,0.3] for 5 times each. The performance will be analysis based on box orientation and FSR value.

4.3.2.1 Result

To obtain derivative coefficient and integral coefficient, Ziegler-Nichols tuning method is implemented using the formula show in Table 4.5.

Table 4.5: Table shows the formula to implement Ziegler-Nichols into the system.

Ziegler-Nichols method			
Control Type	K_p	T_i	T_d
PID	$0.6K_u$	P_u / 2	P_u/8

Since K_p=0.8 produced the lowest average RMSE value, hence we assumed K_p=K_u. For the value of period for one oscillation, P_u, since the graph for K_p=0.8 shown in Table 4.2 is a damping graph, the P_u is obtained using damped oscillation method.

Obtained from the graph for K_p=0.8 shown in Table 4.2,

$$P_u = 0.4s.$$

After substituting K_u and P_u into the formula mentioned in Table 4.5, the calculation is shown in Table 4.6.

Table 4.6: Table shows the value for K_p, T_i and T_d

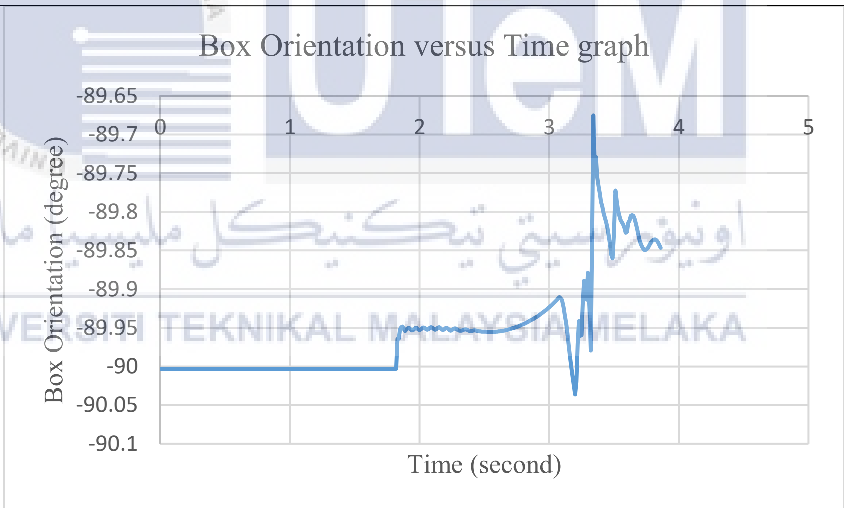
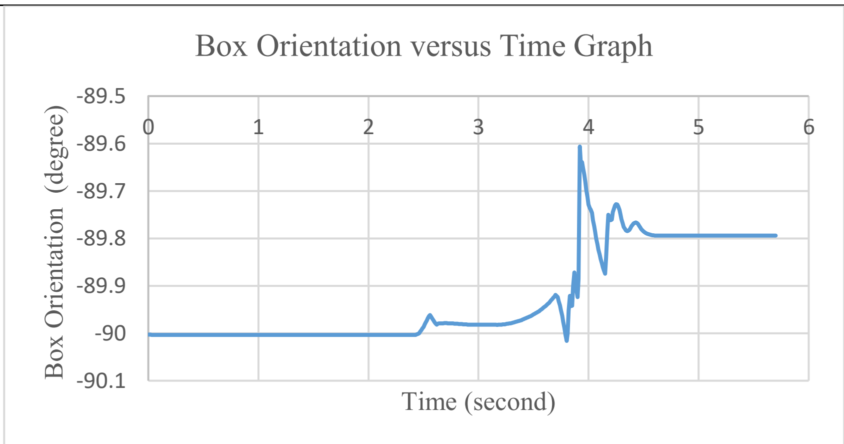
Control Type	K_p	T_i	T_d
PID	0.6×0.8 = 0.48	0.4/2 = 0.2	0.4/8 = 0.05

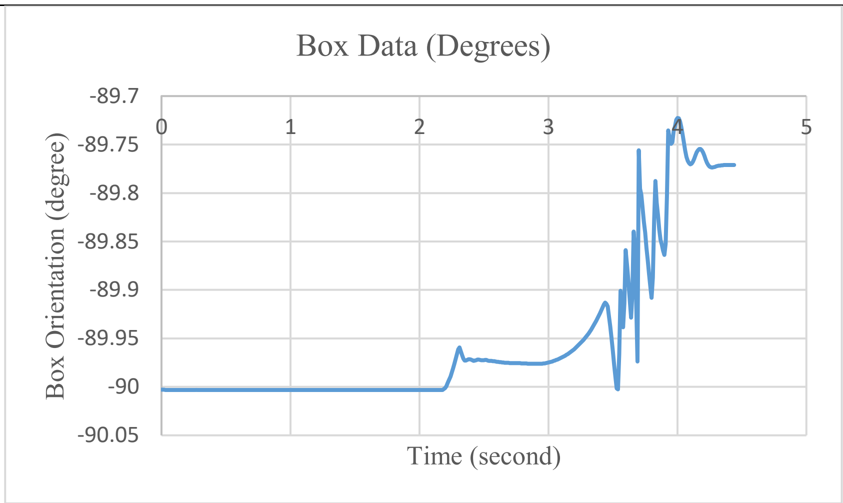
By substituting K_p, T_i and T_d into equation (3.9),

$$u(t) = 0.48\left(e(t) + \frac{1}{0.2}\int_0^5 e(\tau)\,d\tau + 0.05\frac{d}{dt}e(t)\right)$$

To validate the performance of the system, the disturbance is set to the same value which is 0.1. Table 4.7 shows the box orientation scenario for system without controller, proportional controller and PID controller, at which the disturbance is set to the same value.

Table 4.7: Box Orientation for system without controller, system with proportional controller and system with PID controller when random number is set to 0.1.

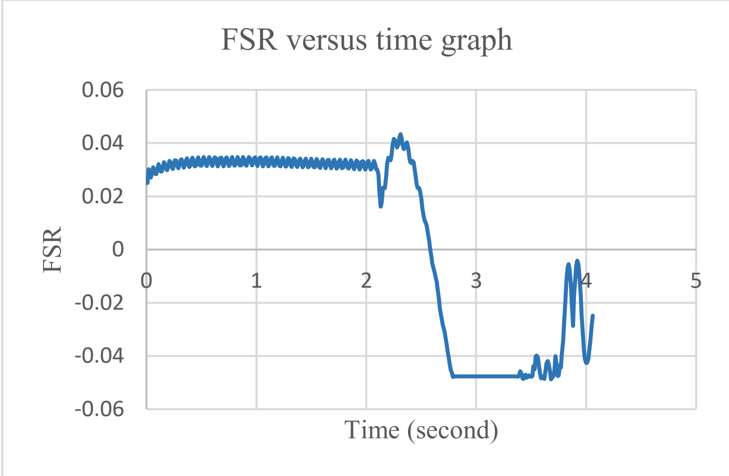
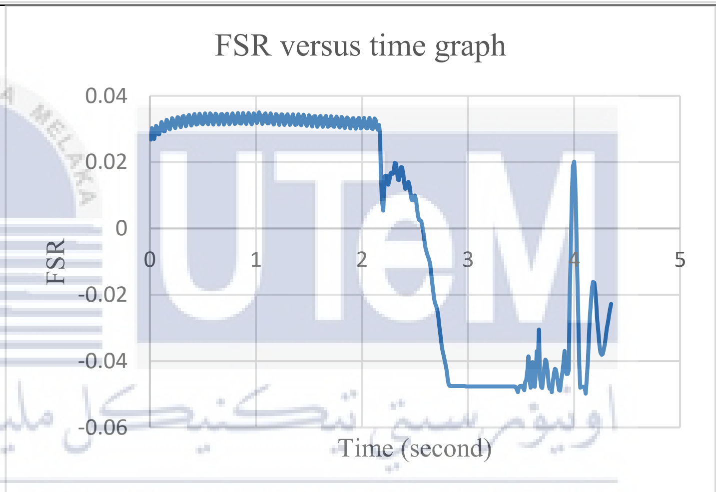
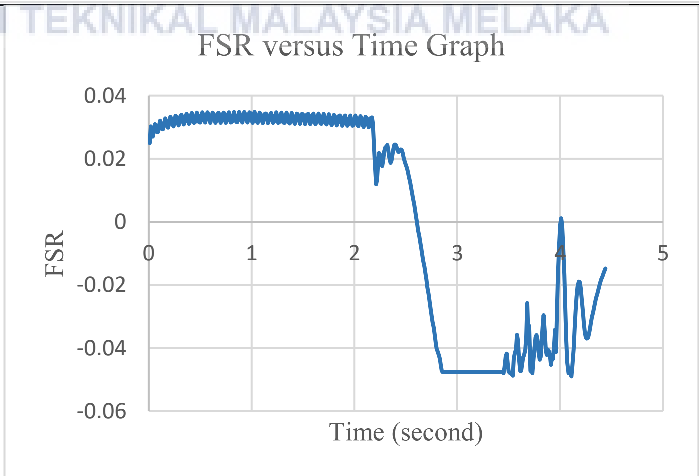
<table>
<tr><th>Controller</th><th>Box Orientation</th></tr>
<tr><td>No Controller</td><td></td></tr>
<tr><td>Proportional</td><td></td></tr>
</table>

PID	

The fluctuation range for system without controller is from -89.61˚ to -90.02˚at which the difference is 0.41˚. The fluctuation range for proportional controller is from -89.68˚ to -90.04˚at which the differences is 0.36˚. The fluctuation range for PID controller lies between -89.72˚and -90.01˚ which give the differences of 0.28˚.

Table 4.8 shows the graph of FSR for three different controllers, the Nao robot 1 for proportional controller started to change gradually starting from 2.07 second.

Table 4.8: FSR value for system without controller, system with P controller and system with PID system.

<table>
<tr><th>System</th><th>FSR Nao 1</th></tr>
<tr><td>No Controller</td><td>
</td></tr>
<tr><td>Proportional</td><td>
</td></tr>
<tr><td>PID</td><td>
</td></tr>
</table>

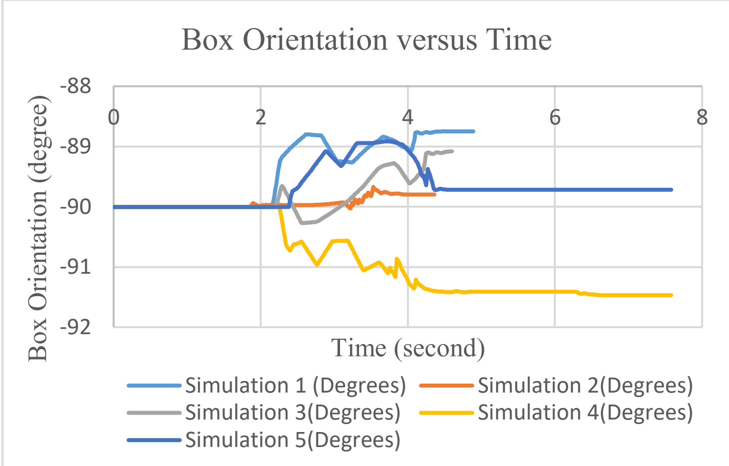
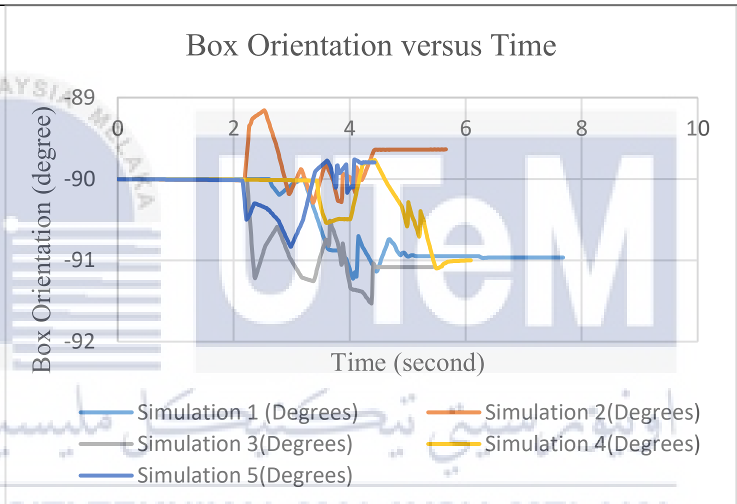
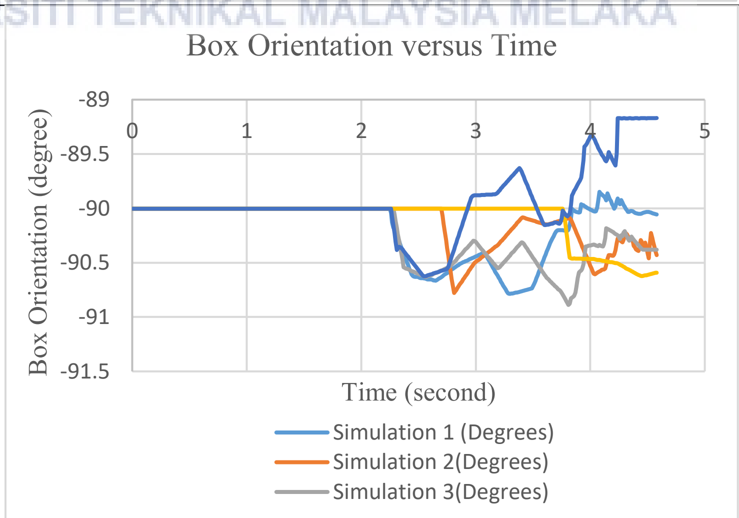
For system with proportional controller, the Nao robot 1's FSR started to rise at 2.16 second after a drop for 0.09 second until 2.31 second with value 0.0434. Then, the FSR dropped to -0.048 at 2.79 second and stayed constant until 3.38 second. The graph started to fluctuate until the lifting motion stopped at 4.06 second. In the simulation, the Nao robots do not show any sign of falling.

Nao robot 1 for system without controller started to change gradually starting from 2.17 second. The FSR started to drop at 2.17 second with 0.0332 until 2.2 second with a value of 0.0054 then, the FSR rose to a small peak with 0.0225 at 2.42 second. FSR value is then dropped to -0.48 at 2.83 second and stayed constant until a fluctuation started at 3.51 seconds. The graph fluctuated until the lifting motion stopped at 4.35 second. In the simulation, the Nao robots in the scenario do not show any sign of falling.

Similar to system with PID controller, both Nao robot do not show any sign of falling. Nao robot 1 started to change gradually starting from 2.18 second. The FSR started to drop at 2.18 second with 0.0312 until 2.21 second with a value of 0.0119 then, the FSR rose to a small peak with 0.0198 at 2.31 second. FSR value is then dropped to -0.48 at 2.9 second and stayed constant until a fluctuation started at 3.43 seconds. The graph fluctuated until the lifting motion stopped at 4.42 second.

Table 4.9 shows the comparison of box orientation for system without controller, with proportional controller and PID controller for random number range, [0,0.1].

Table 4.9: Box orientation for system without controller, system with P controller and system with PID system in the range of 0 to 0.1.

<table>
<tr><th>Controller</th><th>Box</th></tr>
<tr><td>No controller</td><td>
</td></tr>
<tr><td>Proportional</td><td>
</td></tr>
<tr><td>PID</td><td>
</td></tr>
</table>

After 5 trials for each system, the range of box orientation for system without controller has maximum range of 1.47 degree which is generated in simulation 4. Proportional controller has maximum range of 1.525 degree which is generated by simulation 2. For PID controller, the maximum range is 1.44 degree which is the lowest among the three systems.

The performance is further determined by using average RMSE value to calculate. The deviation angle from -90˚will be considered as error. By applying equation (3.13), the average RMSE value are calculated as shown in Table 4.10.

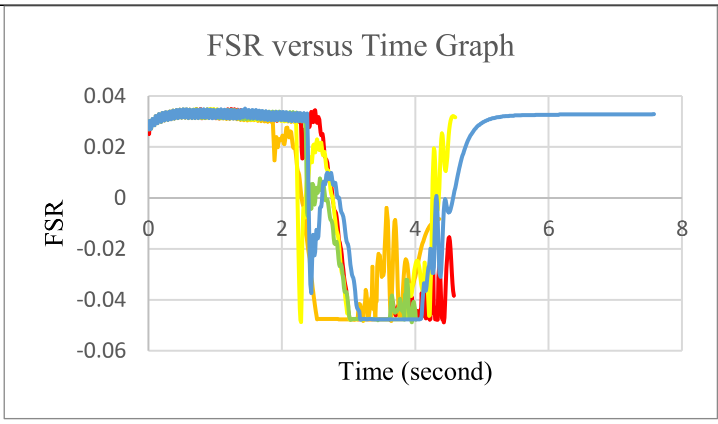
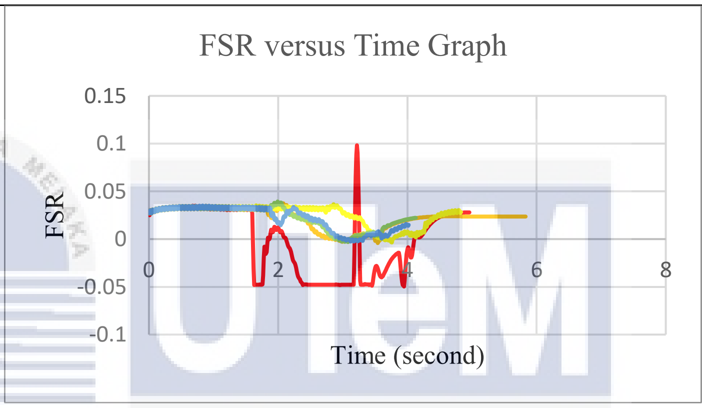
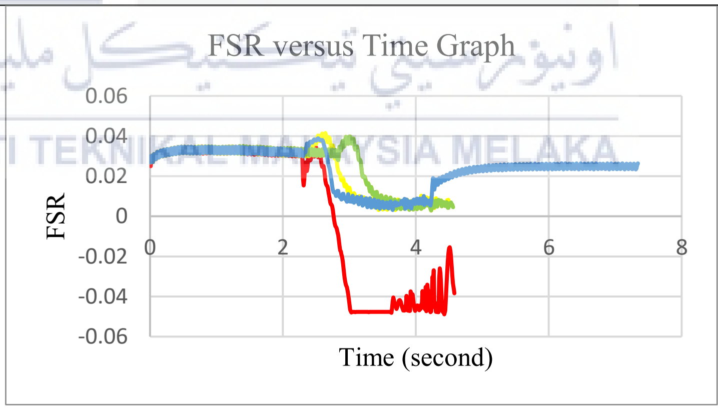
Table 4.10: The comparison of average RMSE value for three systems.

System	Average RMSE Value	Differences (Percentage)
Without Controller	1.391493	0
With Proportional Controller	1.337079	-3.91 %
With PID Controller	0.999232	-28.19 %

System with PID controller show the lowest average RMSE value with value 0.9992 and the highest average RMSE value is system without controller with value of 1.3915. By taking system without controller as the reference, the percentage in terms of difference in average RMSE value is tabulated in Table 4.10.

Nao robot will fell if FSR value exceeded the stability range. For this simulation, the stability range is set in between -0.05 and 0.1. Table 4.11 shows the FSR graph for three different system with 0.5kg weight and random number range of 0 to 0.1m/s. Each graph is illustrating 5 different simulation conducted which are indicated by red, orange, yellow, green and blue colour.

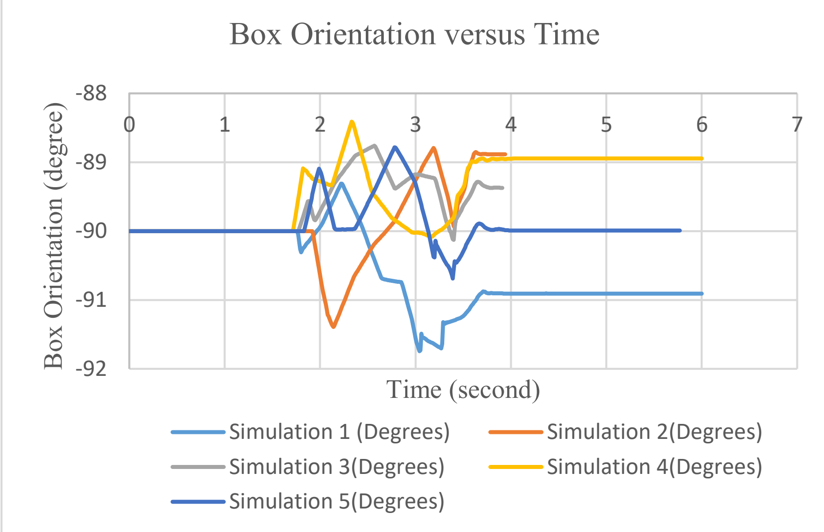
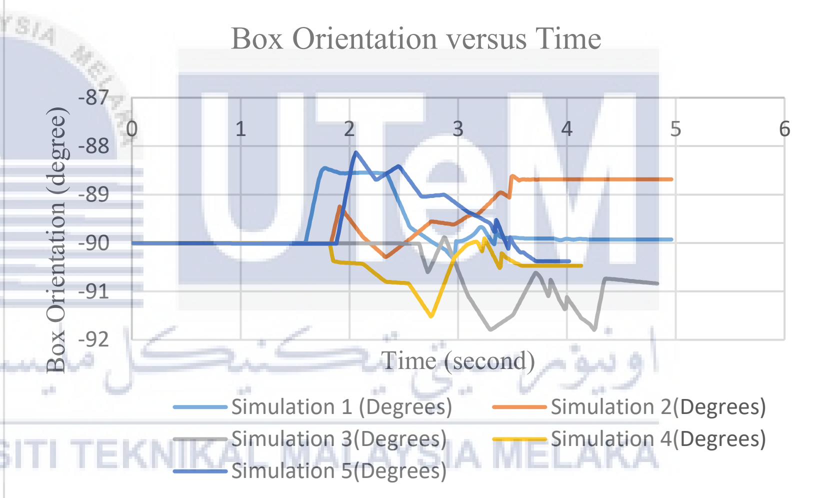
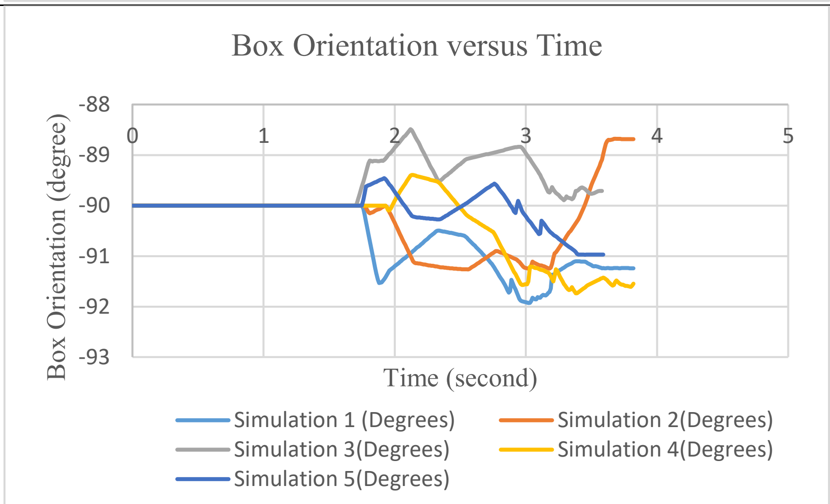
Table 4.11: FSR value for system without controller, system with P controller and system with PID system for range of RN, [0,0.1].

System	FSR Nao 1
No Controller	
Proportional	
PID	

Based on Table 4.11, the three systems do not show any sign of robot falling. However, system without controller shows great fluctuation during lifting operation and all the graphs are almost approach to the stability range's boundary. For the other two systems with controller implementation, all the simulations lie within stability region and only one simulation shows instability for each system respectively.

Table 4.12 shows the comparison of box orientation for system without controller, with proportional controller and PID controller for random number range, [0,0.2].

Table 4.12: Box orientation for system without controller, system with P controller and system with PID system in the range of 0 to 0.2.

Controller	Box
No controller	
Proportional	
PID	

After 5 trials for each system, the range of box orientation for system without controller has maximum range of 2.53 degree which is generated in simulation 2.

Proportional controller has maximum range of 2.19 degree which is generated by simulation 2 has the lowest range among the three systems. For PID controller, the maximum range is 2.55 degree which is the highest among the three systems.

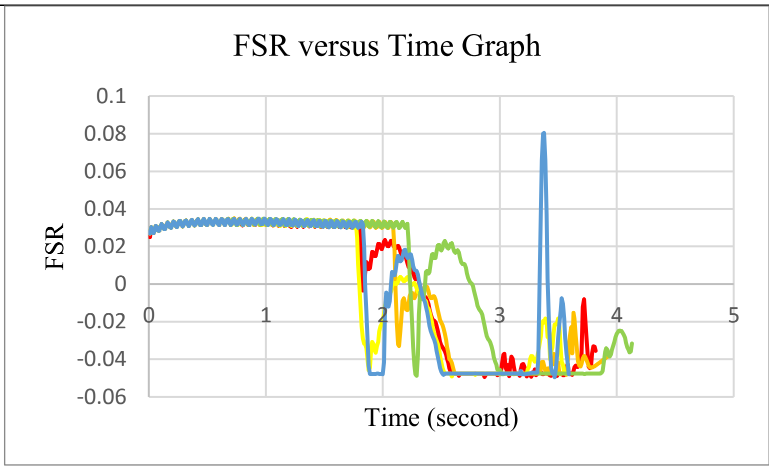
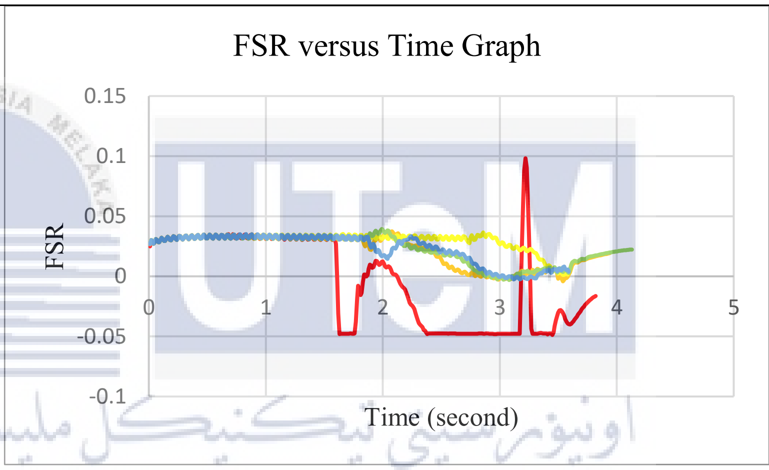
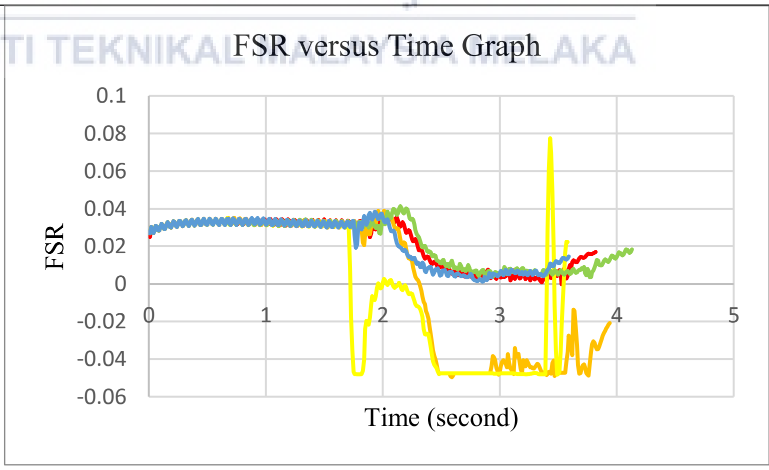
By applying equation (3.13), the average RMSE value are calculated as shown in Table 4.10. By taking system without controller as the reference, the percentage in terms of difference in average RMSE value is tabulated in Table 4.13.

Table 4.13: The comparison of average RMSE value for three systems.

System	Average RMSE Value	Differences (Percentage)
Without Controller	1.334326979	0
With Proportional Controller	1.302426563	-2.39 %
With PID Controller	1.227094744	-8.04 %

System with PID controller show the lowest average RMSE value with value of 1.227 and the highest average RMSE value is system without controller with value of 1.334. The other concern for robot performance is the stability of robot. Table 4.14 shows the FSR graph for three different system with 0.5kg weight and random number range of 0 to 0.2m/s.

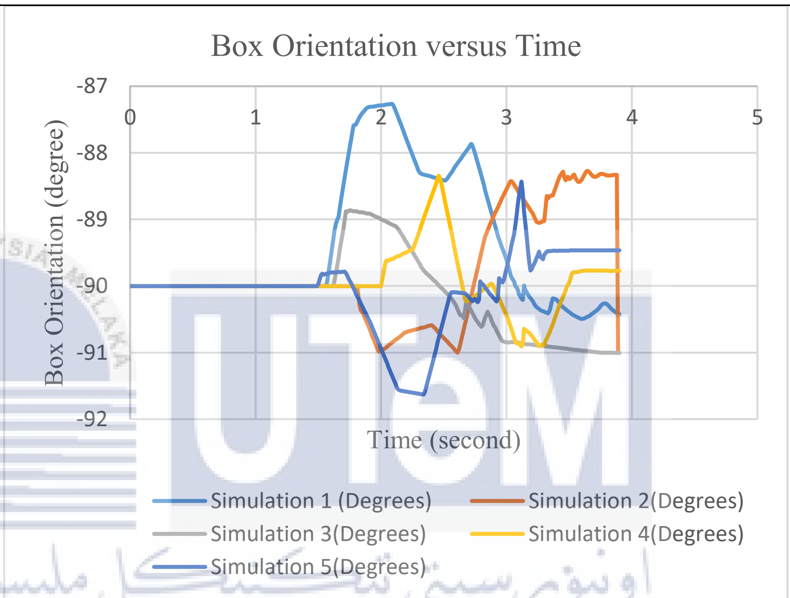
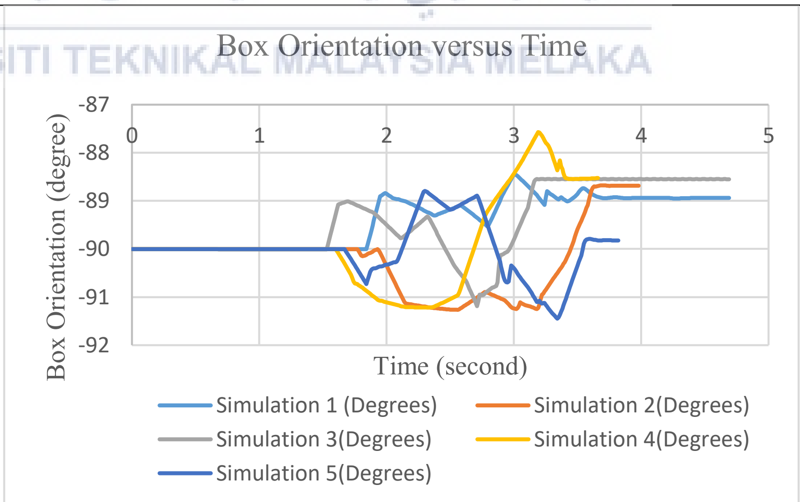
Table 4.14: FSR value for system without controller, system with P controller and system with PID system for range of RN, [0,0.2].

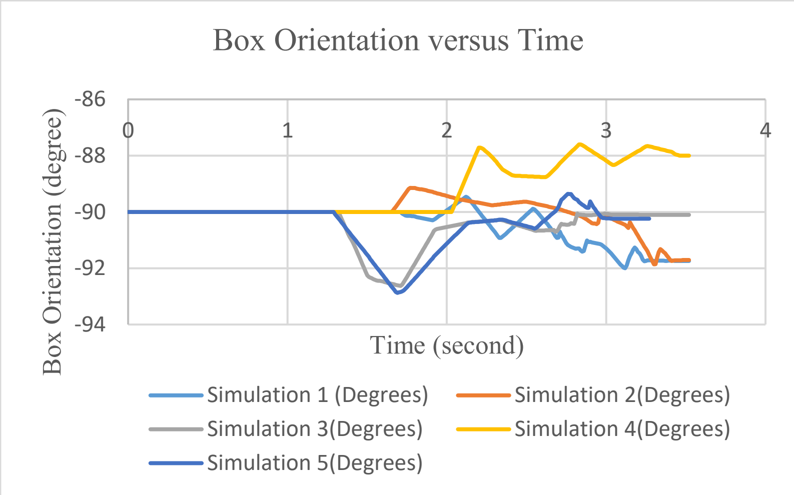
<table>
<tr><th>System</th><th>FSR Nao 1</th></tr>
<tr><td>No Controller</td><td>
</td></tr>
<tr><td>Proportional</td><td>
</td></tr>
<tr><td>PID</td><td>
</td></tr>
</table>

Based on Table 4.14, the three systems do not show any sign of robot falling. However, system without controller shows great fluctuation during lifting operation and all the graphs are almost approach to the stability range's boundary. For the system with proportional controller implementation, all the simulations lie within stability region and only one simulation shows instability. For PID system, there are two simulations with approached the stability boundary in the middle of lifting operation, simulation 2 which indicated by orange line and simulation 3 indicated by yellow line.

Table 4.15 shows the comparison of box orientation for system without controller, with proportional controller and PID controller for random number range, [0,0.3].

Table 4.15: Box orientation for system without controller, system with P controller and system with PID system in the range of 0 to 0.3.

<table>
<tr><th>Controller</th><th>Box</th></tr>
<tr><td>No controller</td><td>
</td></tr>
<tr><td>Proportional</td><td>
</td></tr>
</table>

PID	

After 5 trials for each system, the range of box orientation for system without controller has maximum range of 3.71 degree which is generated in simulation 1. Proportional controller has maximum range of 3.55 degree which is generated by simulation 4. For PID controller, the maximum range is 3.48 degree which is the lowest among the three systems.

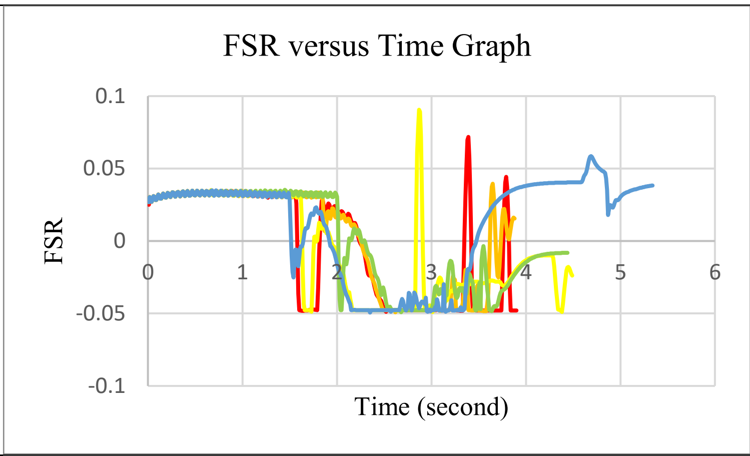
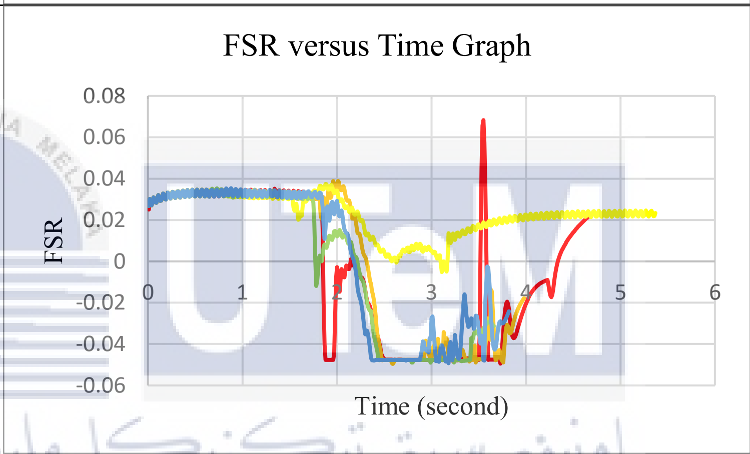
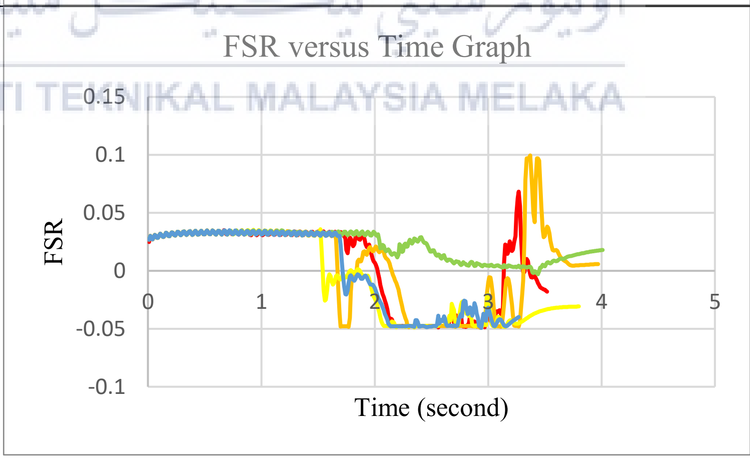
By taking system without controller as the reference, the percentage in terms of difference in average RMSE value is tabulated in Table 4.16.

Table 4.16: The comparison of average RMSE value for three systems.

System	Average RMSE Value	Differences (Percentage)
Without Controller	1.360981938	0
With Proportional Controller	1.352317387	-0.64 %
With PID Controller	1.291346189	-5.12 %

System with PID controller show the lowest average RMSE value with value of 1.2913 and the highest average RMSE value is system without controller with value of 1.361. By taking system without controller as the reference, the percentage in terms of difference in average RMSE value is tabulated in Table 4.16. Table 4.17 shows the FSR graph for three different system with 0.5kg weight and random number range of 0 to 0.3m/s.

Table 4.17: FSR value for system without controller, system with P controller and system with PID system for range of RN, [0,0.3].

<table>
<tr><th>System</th><th>FSR Nao 1</th></tr>
<tr><td>No Controller</td><td>
</td></tr>
<tr><td>Proportional</td><td>
</td></tr>
<tr><td>PID</td><td>
</td></tr>
</table>

Based on Table 4.17, the three systems do not show any sign of robot falling. Since most of the simulation for three systems approached stability boundary, to compare the performance of the systems, the fluctuation of the graph is made as the subject to compare. System without controller implementation shows the most fluctuate graph as compared to the other two systems.

4.3.2.2 Discussion

Based on the box orientation graph shown in Table 4.7, it is clearly show that there is a communication medium which sends signal to Nao robots and connects two Nao robots together as the object orientation shows a sign of regulation after box orientation deviated from -90 °. Hence, for the research question, *how a communication network can be established so other robot can receive the signal?* The communication network can be established through a communication medium like sensor. For this case, the communication medium between the Nao robots is gyroscope, gyroscope provide the box orientation to the robot, so the robot can perform joint velocity regulation based on the signal received. The box orientation result shown in Table 4.7 illustrated the regulation occurred in each system. The better the regulation of the system, the smaller the deviation of degree from -90°, a position at which the box does not need regulation.

Table 4.18: The fluctuation range and differences of range for RN = 0.1

System	Fluctuation Range	Differences
No Controller	-89.61° to -90.02°	0.41°
Proportional Controller	-89.68° to -90.04°	0.36°
PID Controller	-89.72°and -90.01°	0.28°

From differences shown in Table 4.18, it is clearly show that system with implementation of controller has small range compared to system without controller. Hence, implementation of controller is the answer for second research question: *how to control the robot's motion so both humanoid robots can regulate themselves according to the stability of the object?* Controller helps in compensate the error and regulate joint velocity, so the box orientation does not deviate to far from the reference angle which is -90 °.

The controllers are compared by using different range of random number to imitate disturbance that existed in real world. By comparing the average RMSE value for the systems in three different range, it is clearly shown the average RMSE value increased as the range of the random number getting bigger. Taking PID controller as reference, the average RMSE value increase 0.2923 from 0.999 to 1.2913 which is 29.26 % after increase the random number from [0,0.1] to [0,0.3].

Coupled with the FSR graph which shown the number of simulation which approach stability region's boundary getting more as the range of random number increase. This can conclude that the controller designed has almost reached their limit to compensate the disturbance when the range of disturbance is 0 to 0.3m/s. However, during the comparison, PID controller shows a good performance compared to other system by obtaining the lowest average RMSE value for the three random number ranges.

4.4 Comparison of Single Nao Robot and Two Nao Robots in Object Manipulation

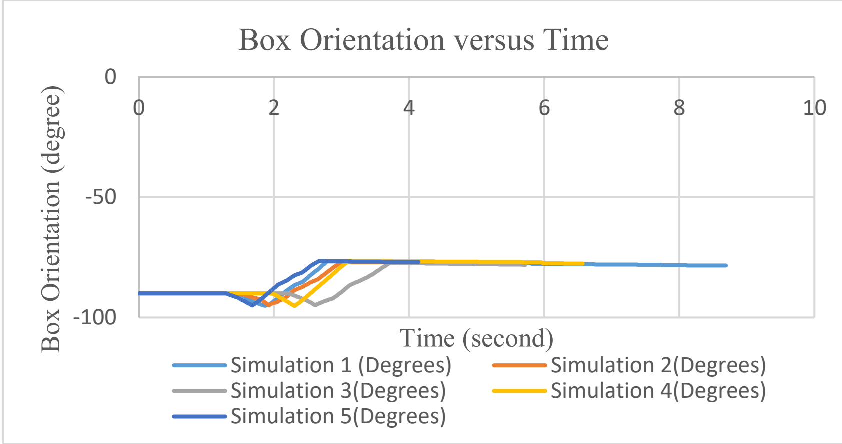
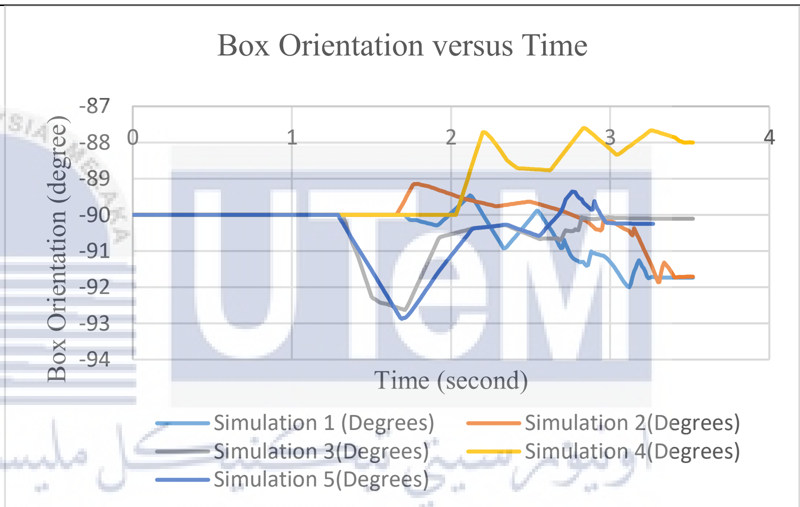
The objective of this comparison between single Nao robot and two Nao robots in terms of performance in object manipulation. Similar to comparison of system with and without controller, the analysis will be done according to box orientation and Nao robot's stability. Both simulation will be conducted in similar environment to make comparison. The comparison is made between single Nao robot and two Nao robots with implementation of controller by using different weight, 0.5kg and 2.5kg. Each simulation is done for 5 times.

4.4.1 Result

The comparison will be based on two elements which are box orientation and FSR readings. Each of the comparison with repeat for 5 times and graph will be displayed in a form of table before doing comparison. The random number range is set to be 0 to 0.3m/s and the weight is set to 0.5 kg and 2.5 kg respectively.

Table 4.19 shows the result of box orientation between single Nao robot and two Nao robots for object weight of 0.5kg and constant random number range of [0,0.3].

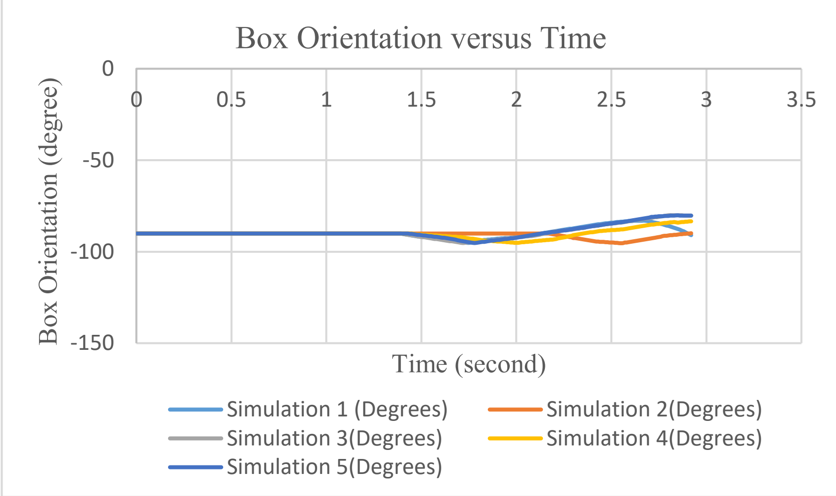
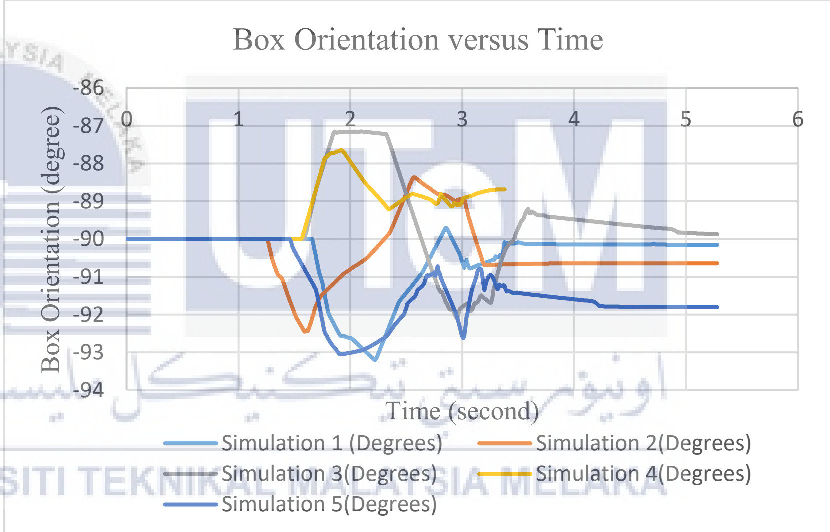
Table 4.19: The result for 5 trials between single Nao robot and Two Nao robot in terms of Object Orientation with a weight of 0.5kg for RN = [0,0.3].

Scenario	Graph
Single Nao Robot	
Two Nao Robot	

After 5 trials for each system, the range of box orientation for single robot system, the maximum deviation of the simulation is 17.88 degree which is generated in simulation 5. The range of box orientation for two Nao robots' system, the maximum deviation of the simulation is 3.054 degree which is generated in simulation 5.

Table 4.20 shows the result of box orientation between single Nao robot and two Nao robots for object weight of 2.5kg and constant random number range of [0,0.3].

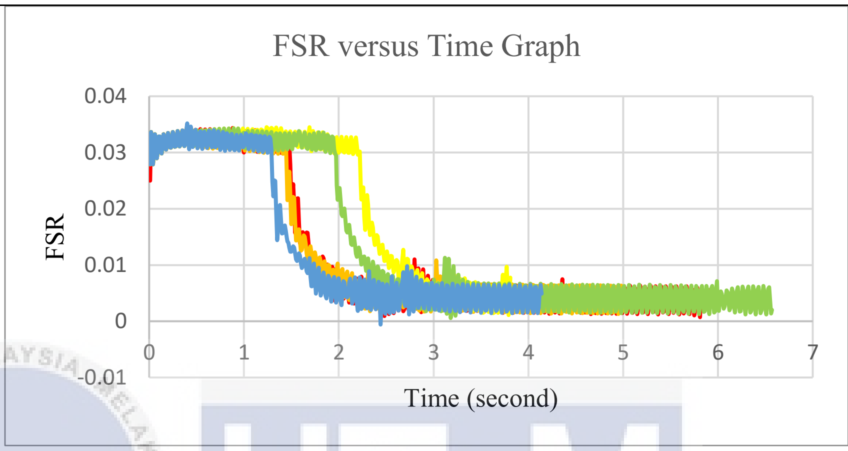
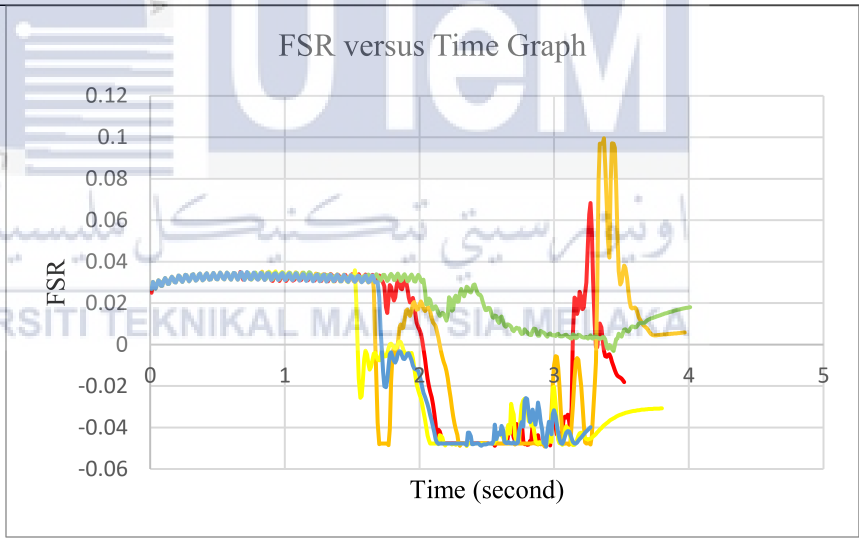
Table 4.20: The result for 5 trials between single Nao robot and Two Nao robot in terms of Object Orientation with a weight of 2.5kg for RN = [0,0.3].

<table>
<tr><th>Scenario</th><th>Graph</th></tr>
<tr><td>Single Nao Robot</td><td>
</td></tr>
<tr><td>Two Nao Robot</td><td>
</td></tr>
</table>

After 5 trials for each system, the range of box orientation for single robot system, the maximum deviation of the simulation is 14.99 degree which is generated in simulation 5. The range of box orientation for two Nao robots' system, the maximum deviation of the simulation is 5.06 degree which is generated in simulation 3. The deviation of graph for single robot system shows decrement, yet the deviation is still very high.

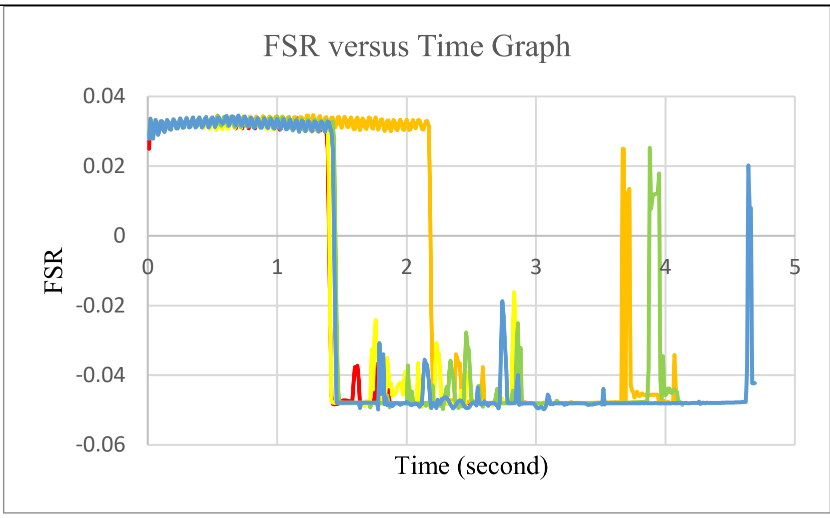
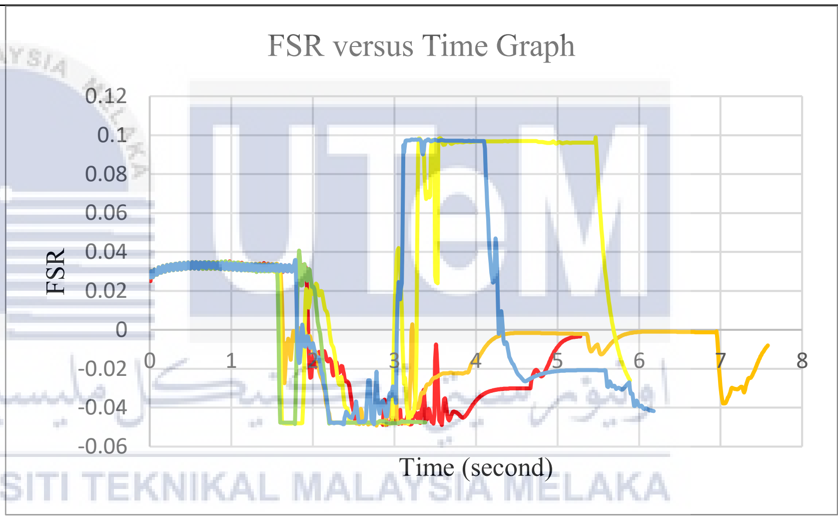
The FSR graph shown in Table 4.21 and Table 4.22 shows the stability of the Nao robot. Each graph is illustrating 5 different simulation conducted which are indicated by red, orange, yellow, green and blue colour.

Table 4.21: The barycentre reading vs time graph for both single Nao robot and two Nao robots for weight = 0.5kg.

Scenario	
Single Nao Robot	
Two Nao Robot	

Based on Table 4.21, the three systems do not show any sign of robot falling. However, FSR graph for both single and two Nao robot shows serious fluctuation especially single robot system. For two Nao robots with PID controller implementation, four out of five graphs are almost approach to the stability range's boundary.

Table 4.22: The barycentre reading vs time graph for both single Nao robot and two Nao robots for weight = 2.5kg.

<table>
<tr><th>Scenario</th><th></th></tr>
<tr><td>Single Nao Robot</td><td>
</td></tr>
<tr><td>Two Nao Robot</td><td>
</td></tr>
</table>

Based on Table 4.22, the three systems do not show any sign of robot falling, yet single robot in V-rep environment fell after the joint position reached to target position. For two Nao robot systems, the FSR value also show a high fluctuation range and all the graph are approaching the stability boundary.

4.4.2 Discussion

In terms of object manipulation, it is clearly shown that object lifting with two Nao robots has better performance than single robot since the box is lifted from 2 directions. Starting from single Nao robot, the box orientation change dramatically when the Nao robot is lifting the object. The graphs for single robot in Table 4.19 shows identical trend, which is decrement then increment. The changes have shown that first the box is lifted from Nao robot side, slanting from Nao robot side to the other side. Yet when the shoulder lifted to certain level of height, the object slips toward robot arm and thus the orientation of box show decrement. This situation is not a preferable scene to be occurred if the object lifted is fragile or too heavy.

On the other hand, slippage problem doesn't occur for object manipulation of two Nao robots since the box is lifted from both side. Whenever, there is angle deviated from 90˚, the system will respond and regulate until the shoulder reached the desired height.

From the simulation, the Nao robots for both single object manipulation and two Nao robots object manipulation seems to have no failure during the process. All the robots are managed to carry out the duty without fail. However, single Nao robot fell after the 2.5kg weight object lifting process ended. Besides, the stability of single robot can be considered as not very stable when lifting 0.5kg object due to the changes of FSR is very frequent, even with the whole graph is located inside the range between -0.05 to -0.1.

Based on Table 4.20, the stability of the box is better when it is lifted by two Nao robots. This can be proved by the range of the graph's fluctuation, the maximum fluctuation range for single robot is 14.99 degree which is 66.24 % larger than two Nao robots with value of 5.06 degree. By summing up evidence, two robot has better performance than single robot especially dealing with heavy object.

4.5 Conclusion

The success in cubic polynomial trajectory implementation allows the shoulder joint to react according to the setting time and easy to control the process. During the controller design simulation, proportional controller is the first to be design by equating the derivative gain and integral gain to zero. To validate the result, first the disturbance for the systems are set to constant which is 0.1, then the result is then compared and analysed. The three systems are conducted with different random number, 0 to 0.1, 0 to 0.2 and 0 to 0.3 with constant weight of 0.5 kg. Random number is function as the disturbance to the system as there are impossible to have ideal system in real world.

From the evidence collected from average RMSE and FSR value, it is clearly shown that the PID controller is the most suitable controller for this system. However, the designed PID controller is about to reached its limit as the random number range of [0,0.3] has started to cause system with PID controller to become unstable.

For the comparison between single robot and two robots in object manipulation, it is clearly show that two robots are better than one with the supporting evidence from the range of fluctuation of the box orientation and FSR.

CHAPTER 5

CONCLUSION

5.1 Conclusion

To perform a full motion of cooperative object manipulation, a suitable controller and trajectory planning is important. A suitable controller can increase the performance of the system.

In the trajectory planning, two methods were compared and analysed. Cubic polynomial trajectory is more suitable and convenient for control. By using cubic polynomial trajectory equation, the velocity can be controlled by the time, unlike the other method which increase every time the loop is repeated which lead to difficulties in controlling the velocity of the joint.

The implementation of proportional controller enable system to perform better. P controller with the lowest average RMSE value, $K_p = 0.8$, has 12.2 percent less fluctuate than system without controller in the situation random number is lies between 0 and 0.1. The PID controller is then developed using Ziegler-Nichols method. After comparison with other system using different random number range, the implementation of PID controller has lowest RMSE among the system and it is proved to improve the performance of the system better than proportional controller.

Finally, the comparison results between single Nao robot and two Nao robots can clearly present that two is better than one. Taking into account the box orientation and the reading of FSR, two Nao robots with implementation of controller perform better.

The cooperative behaviour of robot can be further developed using other communication medium. The object manipulation activity can be further developed, for example squat and lift object or walking while lifting object which similar to human motion. Besides, a system to control multiple humanoid robots in object manipulation can be developed in carrying heavy object or object with large surface area.

REFERENCE

[1] S. A. A. Moosavian, A. Janati, and M. H. Ghazikhani, ‘Object manipulation by two humanoid robots using MTJ control’, in *2011 IEEE International Conference on Mechatronics and Automation*, 2011, pp. 1286–1290.

[2] S. A. A. Moosavian, M. H. Ghazikhani, and A. Janati, ‘Stable gait planning and motion control of two cooperative humanoid robots’, in *Control, Automation and Systems (ICCAS), 2011 11th International Conference on*, 2011, pp. 1123–1128.

[3] Hanzhong Zheng and J. Jumadinova, ‘Using Boolean networks for consensus in multi-robot environmental monitoring tasks’, in *2016 IEEE International Conference on Electro Information Technology (EIT)*, 2016, pp. 0565–0570.

[4] N. R. Ramli, S. Razali, and M. Osman, ‘A conceptual model for multi-robot cooperation inspired by immune network theory and somatic hypermutation’, in *2015 IEEE Student Conference on Research and Development (SCOReD)*, 2015, pp. 495–499.

[5] Y. Inoue, T. Tohge, and H. Iba, ‘Object transportation by two humanoid robots using cooperative learning’, in *Evolutionary Computation, 2004. CEC2004. Congress on*, 2004, vol. 1, p. 1201–1208 Vol.1.

[6] M. H. Wu, A. Konno, S. Ogawa, and S. Komizunai, ‘Symmetry cooperative object transportation by multiple humanoid robots’, in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 3446–3451.

[7] Dong-Wook Lee and Kwee-Bo Sim, ‘Artificial immune network-based cooperative control in collective autonomous mobile robots’, in *Robot and Human Communication, 1997. RO-MAN ’97. Proceedings., 6th IEEE International Workshop on*, 1997, pp. 58–63.

[8] Ching-man Chan, ‘Artificial Immunity Inspired Cooperative Failure Recovery Framework for Mobile Multi-Robot System’. 2014.

[9] Yunyuan Gao and Zhizeng Luo, ‘Dynamic task allocation method based on immune system for cooperative robots’, in *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, 2008, pp. 1015–1020.

[10] L. Laursen, ‘A Robot’s Body of Knowledge’, 02-Nov-2010. [Online]. Available: http://spectrum.ieee.org/robotics/artificial-intelligence/a-robots-body-of-knowledge

[11] ‘Humanoid Robotics’, *Humanoid Robotics*, 26-Dec-2016. [Online]. Available: http://www.ieee-ras.org/humanoid-robotics.

[12] H. G. Kim, J. U. An, J. I. Moon, and D. H. Kim, ‘Simulation based for intelligent control system of multi — Humanoid robots for dynamic load carrying’, in *Control Automation and Systems (ICCAS), 2010 International Conference on*, 2010, pp. 101–104.

[13] Y. Inoue, T. Tohge, and H. Iba, ‘Object transportation by two humanoid robots using cooperative learning’, in *Evolutionary Computation, 2004. CEC2004. Congress on*, 2004, vol. 1, p. 1201–1208 Vol.1.

[14] A. Forouzantabar, H.A. Talebi, Senior Member, and A.K. Sedigh, ‘Bilateral Control of Master-Slave Manipulators with Constant Time Delay’, Jul. 2011.

[15] K.H. Low, *Robotics: Principles and Systems Modeling*, Fourth edition. Pearson Education South Asia Pte.Ltd., 2007.

[16] Appin Knowledge Solutions, *Robotic/Appin Knowledge Solutions.* New Delhi: Infinity Science Press LLC, 2007.

[17] E. Schierer, ‘Humanoid Robots for Human Life Support A2 - Kopacek, P.’, in *Improving Stability in Developing Nations through Automation 2006*, Oxford: Elsevier, 2006, pp. 101–105.

[18] K. Suwanratchatamanee, M. Matsumoto, and S. Hashimoto, ‘Balance control of humanoid robot in object lifting task with tactile sensing system’, in *Human System Interactions (HSI), 2011 4th International Conference on*, 2011, pp. 431–436.

[19] W. Sheng, A. Thobbi, and Y. Gu, ‘An Integrated Framework for Human–Robot Collaborative Manipulation’, *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2030–2041, Oct. 2015.

[20] C. R. Carignan and D. L. Akin, ‘Cooperative control of two arms in the transport of an inertial load in zero gravity’, *IEEE J. Robot. Autom.*, vol. 4, no. 4, pp. 414–419, Aug. 1988.

[21] S. Kamio and H. Iba, ‘Search Algorithm of the Order of Object Transportation by Multiple Robots’, in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 4097–4102.

[22] Jong Hwan Kim, Y. Weimin, Jun Jo, S. Peter, and Hyun Myung, ‘Falling Prevention System from External Disturbances for Humanoid Robots’, in *Robot Intelligence Technology and Applications 3*, Springer International Publishing, 2015, p. pp 97-105.

[23] K. Suwanratchatamanee, M. Matsumoto, and S. Hashimoto, ‘Balance control of humanoid robot in object lifting task with tactile sensing system’, in *Human System Interactions (HSI), 2011 4th International Conference on*, 2011, pp. 431–436.

[24] Meng Joo Er and Yi Zhou, ‘Intelligent Fuzzy Q-Learning Control of Human Robots’, in *Advances in Neural Networks- ISNN 2005*, 1st ed., Jun Wang, Xiaofeng Liao, and Zhang Yi, Eds. Springer-Verlag Berlin Heidelberg, 2005, p. 947.

[25] Y. S. Cha, S. M. Hong, D. Kim, B. J. You, and S. R. Oh, ‘Whole-body humanoid balance control with dynamically loading/unloading objects’, in *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, 2009, pp. 1687–1691.

[26] V. Derhami, V. J. Majd, and M. Nili Ahmadabadi, ‘Exploration and exploitation balance management in fuzzy reinforcement learning’, *Theme Forecast. Classif. Learn.*, vol. 161, no. 4, pp. 578–595, Feb. 2010.

[27] K. Suwanratchatamanee, M. Matsumoto, and S. Hashimoto, ‘Balance control of humanoid robot in object lifting task with tactile sensing system’, in *Human System Interactions (HSI), 2011 4th International Conference on*, 2011, pp. 431–436.

[28] Y. Choi, D. Kim, Y. Oh, and B. J. You, ‘Posture/Walking Control for Humanoid Robot Based on Kinematic Resolution of CoM Jacobian With Embedded Motion’, *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1285–1293, Dec. 2007.

[29] H. Y. Lee, H. P. Huang, and H. K. Hsu, ‘Lifting motion planning for humanoid robots’, in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, 2014, pp. 1174–1179.

[30] H. Arisumi, J. R. Chardonnet, A. Kheddar, and K. Yokoi, ‘Dynamic Lifting Motion of Humanoid Robots’, in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 2661–2667.

[31] H. Arisumi, J. R. Chardonnet, A. Kheddar, and K. Yokoi, ‘Dynamic Lifting Motion of Humanoid Robots’, in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 2661–2667.

[32] S. Chen, S. Sun, H. Shen, and B. Chen, ‘Kinematic analysis and simulation of squatting down and lifting object for humanoid robot’, in *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2013, pp. 2732–2736.

[33] T. H. S. Li, Chia-Ling Hsu, Chun-Yang Hu, Yu-Te Su, Ming-Feng Lu, and Shao-Hsien Liu, 'SOPC based weight lifting control design for small-sized humanoid robot', in *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*, 2008, pp. 789–793.

[34] H. Liu, N. Stoll, S. Junginger, and K. Thurow, 'A new method for mobile robot arm blind grasping using ultrasonic sensors and Artificial Neural Networks', in *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2013, pp. 1360–1364.

[35] S. Uematsu, Y. Kobayashi, A. Shimizu, and T. Kaneko, 'Prediction of object manipulation using tactile sensor information by a humanoid robot', in *Robotic and Sensors Environments (ROSE), 2012 IEEE International Symposium on*, 2012, pp. 37–42.

[36] M. Murooka *et al.*, 'Whole-body holding manipulation by humanoid robot based on transition graph of object motion and contact', in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, 2015, pp. 3950–3955.

[37] B. Chandrasekaran and J. M. Conrad, 'Human-robot collaboration: A survey', in *SoutheastCon 2015*, 2015, pp. 1–8.

[38] K. Yamane, M. Revfi, and T. Asfour, 'Synthesizing object receiving motions of humanoid robots with human motion database', in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013, pp. 1629–1636.

[39] P. Evrard, E. Gribovskaya, S. Calinon, A. Billard, and A. Kheddar, 'Teaching physical collaborative tasks: object-lifting case study with a humanoid', in *2009 9th IEEE-RAS International Conference on Humanoid Robots*, 2009, pp. 399–404.

[40] H. Y. Lee, H. P. Huang, and H. K. Hsu, 'Lifting motion planning for humanoid robots', in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, 2014, pp. 1174–1179.

[41] S. A. A. Moosavian, A. Janati, and M. H. Ghazikhani, 'Object manipulation by two humanoid robots using MTJ control', in *2011 IEEE International Conference on Mechatronics and Automation*, 2011, pp. 1286–1290.

[42] W. Sheng, A. Thobbi, and Y. Gu, 'An Integrated Framework for Human–Robot Collaborative Manipulation', *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2030–2041, Oct. 2015.

[43] John J. Craig, *Introduction to Robotics*, Third. Pearson, 2004.

[44] D. M. Sajnekar, S. B. Deshpande, and R. M. Moharil, 'Efficient PID controller tuning method selection to be used in excitation system of brushless synchronous generator',

in *2016 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC)*, 2016, pp. 413–418.

[45] M.Sugisaka and D.Hazry, 'Development of a proportional control method for a mobile robot', p. 9, 2006.

[46] M. Shahrokhi and Alireza Zomorrodi, 'Comparison of PID Controller Tuning Methods', *Sharif Univ. Technol.*, 2013.

[47] H. Mellmann, M. Scheunemann, and O. Stadie, 'Adaptive Grasping for a Small Humanoid Robot Utilizing Force- and Electric Current Sensors', p. 11.

[48] A. Palacios-Laloy, J. Rutkowski, Y. Troadec, and J. M. Leger, 'On the critical impact of HF power drifts for miniature helium-based NMR gyroscopes', *IEEE Sens. J.*, vol. PP, no. 99, pp. 1–1, 2016.

[49] R. Mazgut, P. Spanik, J. Koscelnik, and P. Sindler, 'The measurement of balance by the accelerometer and gyroscope', in *2014 ELEKTRO*, 2014, pp. 192–196.

[50] M. B. Bahar, M. F. Miskon, F. Ali, A. Z. Shukor, and M. H. Jamaludin, 'Investigating the relation between standing period and stability of humanoid robot when performing STS motion', in *2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, 2015, pp. 99–104.

APPENDICES

APPENDIX A (CODING)

```
import vrep
import sys
import random
import math

vrep.simxFinish(-1)
clientID=vrep.simxStart('127.0.0.1',19999,True,True,5000,5)

if clientID!=-1:
    print 'connected to remote API server'
else:
    print 'connection not succesful'
    sys.exit('error')
#SHOULDER DECLARE
errorcode,RShoulder=vrep.simxGetObjectHandle(clientID,'RShoulderPitch3#1',vrep.simx_opmode_oneshot_wait)
errorcode,LShoulder=vrep.simxGetObjectHandle(clientID,'LShoulderPitch3#1',vrep.simx_opmode_oneshot_wait)
errorcode,RShoulder2=vrep.simxGetObjectHandle(clientID,'RShoulderPitch3#2',vrep.simx_opmode_oneshot_wait)
errorcode,LShoulder2=vrep.simxGetObjectHandle(clientID,'LShoulderPitch3#2',vrep.simx_opmode_oneshot_wait)
#ELBOW DECLARE
errorcode,Relbow_roll3=vrep.simxGetObjectHandle(clientID,'RElbowRoll3#1',vrep.simx_opmode_oneshot_wait)
```

```
errorcode,Lelbow_roll3=vrep.simxGetObjectHandle(clientID,'LElbowRoll3#1',vrep.simx_opmode_oneshot_wait)
errorcode,Relbow2_roll3=vrep.simxGetObjectHandle(clientID,'RElbowRoll3#2',vrep.simx_opmode_oneshot_wait)
errorcode,Lelbow2_roll3=vrep.simxGetObjectHandle(clientID,'LElbowRoll3#2',vrep.simx_opmode_oneshot_wait)
#FSR FOR RIGHT FOOT Nao 1
errorcode,RFsrFL=vrep.simxGetObjectHandle(clientID,'Nao_RFsrFL#1',vrep.simx_opmode_oneshot_wait)
errorcode,RFsrFR=vrep.simxGetObjectHandle(clientID,'Nao_RFsrFR#1',vrep.simx_opmode_oneshot_wait)
errorcode,RFsrBL=vrep.simxGetObjectHandle(clientID,'Nao_RFsrRL#1',vrep.simx_opmode_oneshot_wait)
errorcode,RFsrBR=vrep.simxGetObjectHandle(clientID,'Nao_RFsrRR#1',vrep.simx_opmode_oneshot_wait)
#FSR FOR LEFT FOOT Nao 1
errorcode,LFsrFL=vrep.simxGetObjectHandle(clientID,'Nao_LFsrFL#1',vrep.simx_opmode_oneshot_wait)
errorcode,LFsrFR=vrep.simxGetObjectHandle(clientID,'Nao_LFsrFR#1',vrep.simx_opmode_oneshot_wait)
errorcode,LFsrBL=vrep.simxGetObjectHandle(clientID,'Nao_LFsrRL#1',vrep.simx_opmode_oneshot_wait)
errorcode,LFsrBR=vrep.simxGetObjectHandle(clientID,'Nao_LFsrRR#1',vrep.simx_opmode_oneshot_wait)
#FSR FOR RIGHT FOOT Nao 2
errorcode,RFsrFL2=vrep.simxGetObjectHandle(clientID,'Nao_RFsrFL#2',vrep.simx_opmode_oneshot_wait)
errorcode,RFsrFR2=vrep.simxGetObjectHandle(clientID,'Nao_RFsrFR#2',vrep.simx_opmode_oneshot_wait)
errorcode,RFsrBL2=vrep.simxGetObjectHandle(clientID,'Nao_RFsrRL#2',vrep.simx_opmode_oneshot_wait)
errorcode,RFsrBR2=vrep.simxGetObjectHandle(clientID,'Nao_RFsrRR#2',vrep.simx_opmode_oneshot_wait)
#FSR FOR LEFT FOOT Nao 2
```

```
errorcode,LFsrFL2=vrep.simxGetObjectHandle(clientID,'Nao_LFsrFL#2',vrep.simx_opmode_one
shot_wait)
errorcode,LFsrFR2=vrep.simxGetObjectHandle(clientID,'Nao_LFsrFR#2',vrep.simx_opmode_one
shot_wait)
errorcode,LFsrBL2=vrep.simxGetObjectHandle(clientID,'Nao_LFsrRL#2',vrep.simx_opmode_one
shot_wait)
errorcode,LFsrBR2=vrep.simxGetObjectHandle(clientID,'Nao_LFsrRR#2',vrep.simx_opmode_on
eshot_wait)

#START TO LIFT OBJECT FOR SHOULDER Nao 1
float,init = vrep.simxGetJointPosition(clientID,RShoulder,vrep.simx_opmode_oneshot_wait)
float,initL = vrep.simxGetJointPosition(clientID,LShoulder,vrep.simx_opmode_oneshot_wait)
float,limitL = vrep.simxGetJointPosition(clientID,LShoulder,vrep.simx_opmode_oneshot_wait)
float,limit = vrep.simxGetJointPosition(clientID,RShoulder,vrep.simx_opmode_oneshot_wait)
change = (init-limit)*180/3.141592654

#START TO LIFT OBJECT FOR ELBOW Nao 1
float,init_E =
vrep.simxGetJointPosition(clientID,Relbow_roll3,vrep.simx_opmode_oneshot_wait)
float,initL_E =
vrep.simxGetJointPosition(clientID,Lelbow_roll3,vrep.simx_opmode_oneshot_wait)
float,limitL_E =
vrep.simxGetJointPosition(clientID,Lelbow_roll3,vrep.simx_opmode_oneshot_wait)
float,limit_E =
vrep.simxGetJointPosition(clientID,Relbow_roll3,vrep.simx_opmode_oneshot_wait)
#change_E = (init_E-limit_E)*180/3.141592654

#START TO LIFT OBJECT FOR SHOULDER Nao 2
float,init2 = vrep.simxGetJointPosition(clientID,RShoulder2,vrep.simx_opmode_oneshot_wait)
float,init2L = vrep.simxGetJointPosition(clientID,LShoulder2,vrep.simx_opmode_oneshot_wait)
float,limit2L =
vrep.simxGetJointPosition(clientID,LShoulder2,vrep.simx_opmode_oneshot_wait)
float,limit2 = vrep.simxGetJointPosition(clientID,RShoulder2,vrep.simx_opmode_oneshot_wait)
```

```
#START TO LIFT OBJECT FOR ELBOW Nao 2
float,init2_E =
vrep.simxGetJointPosition(clientID,Relbow2_roll3,vrep.simx_opmode_oneshot_wait)
float,initL2_E =
vrep.simxGetJointPosition(clientID,Lelbow2_roll3,vrep.simx_opmode_oneshot_wait)
float,limitL2_E =
vrep.simxGetJointPosition(clientID,Lelbow2_roll3,vrep.simx_opmode_oneshot_wait)
float,limit2_E =
vrep.simxGetJointPosition(clientID,Relbow2_roll3,vrep.simx_opmode_oneshot_wait)

#OBJECT
errorcode,box=vrep.simxGetObjectHandle(clientID,'Cuboid22',vrep.simx_opmode_oneshot_wa
it)

objposition = []
Fsr1fl = []
Fsr1fr = []
Fsr1bl = []
Fsr1br = []
Fsl1fl = []
Fsl1fr = []
Fsl1bl = []
Fsl1br = []
Fsr2fl = []
Fsr2fr = []
Fsr2bl = []
Fsr2br = []
Fsl2fl = []
Fsl2fr = []
Fsl2bl = []
Fsl2br = []
time = []
angleerror = []
sdegree = []
```

```
ti = vrep.simxGetLastCmdTime(clientID)/1000.0000
obj2=0
obj3=0
ta=0

while change < 30:
    ta = (vrep.simxGetLastCmdTime(clientID))/1000.00000
    t = (vrep.simxGetLastCmdTime(clientID))/1000.00000 - ti
    IP=-30*3.141592654/180
    IP_E=-30*3.141592654/180
    IP2=-30*3.141592654/180
    IP2_E=-30*3.141592654/180
    float,initL = vrep.simxGetJointPosition(clientID,LShoulder,vrep.simx_opmode_oneshot_wait)
    float,initL_E =
vrep.simxGetJointPosition(clientID,Lelbow_roll3,vrep.simx_opmode_oneshot_wait)
    float,init2L =
vrep.simxGetJointPosition(clientID,LShoulder2,vrep.simx_opmode_oneshot_wait)
    float,init2L_E =
vrep.simxGetJointPosition(clientID,Lelbow2_roll3,vrep.simx_opmode_oneshot_wait)
    FP2=init2L*3.141592654/180
    FP2_E=init2L_E*3.141592654/180
    FP=initL*3.141592654/180
    FP_E=init2L_E*3.141592654/180
    tf0=5
    ti0=0
    a=IP
    a_E=IP_E
    a2=IP
    a2_E=IP_E
    b=0
    b_E=0
    b2=0
    b2_E=0
```

```
c=((3.000/pow(tf0,2))*(FP-IP))
c_E=((3.000/pow(tf0,2))*(FP_E-IP_E))
c2=((3.000/pow(tf0,2))*(FP-IP))
c2_E=((3.000/pow(tf0,2))*(FP_E-IP_E))
d=-((2.000/pow(tf0,3))*(FP-IP))
d_E=-((2.000/pow(tf0,3))*(FP_E-IP_E))
d2=-((2.000/pow(tf0,3))*(FP-IP))
d2_E=-((2.000/pow(tf0,3))*(FP_E-IP_E))
t=(vrep.simxGetLastCmdTime(clientID))/1000.0000-ti
s=a+b*t+(c)*(pow(t,2)+(d)*pow(t,3))
s_E=a_E+b_E*t+(c_E)*(pow(t,2)+(d_E)*pow(t,3))
s2=a2+b2*t+(c2)*(pow(t,2)+(d2)*pow(t,3))
s2_E=a2_E+b2_E*t+(c2_E)*(pow(t,2)+(d2_E)*pow(t,3))
v=b +2.000*(c)*t+3.000*(d)*(pow(t,2))
v_E=b_E +2.000*(c_E)*t+3.000*(d_E)*(pow(t,2))
v2=b2 +2.000*(c2)*t+3.000*(d2)*(pow(t,2))
v2_E=b2_E +2.000*(c2_E)*t+3.000*(d2_E)*(pow(t,2))
float,obj = vrep.simxGetObjectOrientation(clientID,box,-1,vrep.simx_opmode_oneshot_wait)
float,degree = vrep.simxGetObjectOrientation(clientID,RShoulder,-
1,vrep.simx_opmode_oneshot_wait)
obj1 = obj[2]
objposition.append(obj1)
degree1 = degree[2]
float,fsr1fl = vrep.simxReadForceSensor(clientID,RFsrFL,vrep.simx_opmode_oneshot_wait)
float,fsr1fr = vrep.simxReadForceSensor(clientID,RFsrFR,vrep.simx_opmode_oneshot_wait)
float,fsr1bl = vrep.simxReadForceSensor(clientID,RFsrBL,vrep.simx_opmode_oneshot_wait)
float,fsr1br = vrep.simxReadForceSensor(clientID,RFsrBR,vrep.simx_opmode_oneshot_wait)
float,fsl1fl = vrep.simxReadForceSensor(clientID,LFsrFL,vrep.simx_opmode_oneshot_wait)
float,fsl1fr = vrep.simxReadForceSensor(clientID,LFsrFR,vrep.simx_opmode_oneshot_wait)
float,fsl1bl = vrep.simxReadForceSensor(clientID,LFsrBL,vrep.simx_opmode_oneshot_wait)
float,fsl1br = vrep.simxReadForceSensor(clientID,LFsrBR,vrep.simx_opmode_oneshot_wait)
float,fsr2fl = vrep.simxReadForceSensor(clientID,RFsrFL2,vrep.simx_opmode_oneshot_wait)
float,fsr2fr = vrep.simxReadForceSensor(clientID,RFsrFR2,vrep.simx_opmode_oneshot_wait)
float,fsr2bl = vrep.simxReadForceSensor(clientID,RFsrBL2,vrep.simx_opmode_oneshot_wait)
```

```
float,fsr2br = vrep.simxReadForceSensor(clientID,RFsrBR2,vrep.simx_opmode_oneshot_wait)
float,fsl2fl = vrep.simxReadForceSensor(clientID,LFsrFL2,vrep.simx_opmode_oneshot_wait)
float,fsl2fr = vrep.simxReadForceSensor(clientID,LFsrFR2,vrep.simx_opmode_oneshot_wait)
float,fsl2bl = vrep.simxReadForceSensor(clientID,LFsrBL2,vrep.simx_opmode_oneshot_wait)
float,fsl2br = vrep.simxReadForceSensor(clientID,LFsrBR2,vrep.simx_opmode_oneshot_wait)
R1S1 = fsr1fl[2]
if R1S1 < 0:
   R1S1 = 0.001
Fsr1fl.append(R1S1)

R1S2 = fsr1fr[2]
if R1S2 < 0:
   R1S2 = 0.001
Fsr1fr.append(R1S2)

R1S3 = fsr1bl[2]
if R1S3 < 0:
   R1S3 = 0.001
Fsr1bl.append(R1S3)

R1S4 = fsr1br[2]
if R1S4 < 0:
   R1S4 = 0.001
Fsr1br.append(R1S4)

L1S1 = fsl1fl[2]
if L1S1 < 0:
   L1S1 = 0.001
Fsl1fl.append(L1S1)

L1S2 = fsl1fr[2]
if L1S2 < 0:
   L1S2 = 0.001
Fsl1fr.append(L1S2)
```

```
L1S3 = fsl1bl[2]
if L1S3 < 0:
   L1S3 = 0.001
Fsl1bl.append(L1S3)

L1S4 = fsl1br[2]
if L1S4 < 0:
   L1S4 = 0.001
Fsl1br.append(L1S4)

R2S1 = fsr2fl[2]
if R2S1 < 0:
   R2S1 = 0.001
Fsr2fl.append(R2S1)

R2S2 = fsr2fr[2]
if R2S2 < 0:
   R2S2 = 0.001
Fsr2fr.append(R2S2)

R2S3 = fsr2bl[2]
if R2S3 < 0:
   R2S3 = 0.001
Fsr2bl.append(R2S3)

R2S4 = fsr2br[2]
if R2S4 < 0:
   R2S4 = 0.001
Fsr2br.append(R2S4)

L2S1 = fsl2fl[2]
if L2S1 < 0:
   L2S1 = 0.001
```

```
  Fsl2fl.append(L2S1)


  L2S2 = fsl2fr[2]
  if L2S2 < 0:
    L2S2 = 0.001
  Fsl2fr.append(L2S2)


  L2S3 = fsl2bl[2]
  if L2S3 < 0:
    L2S3 = 0.001
  Fsl2bl.append(L2S3)


  L2S4 = fsl2br[2]
  if L2S4 < 0:
    L2S4 = 0.001
  Fsl2br.append(L2S4)


  if initL>=IP:
    x= random.uniform(0,0.2)
    y= random.uniform(0,0.2)
    v=v+x
    v2=v2+y
    print v,v2
    if v>0:
      vrep.simxSetJointTargetVelocity(clientID,Relbow_roll3,-
v_E,vrep.simx_opmode_streaming)

vrep.simxSetJointTargetVelocity(clientID,Lelbow_roll3,v_E,vrep.simx_opmode_streaming)
      vrep.simxSetJointTargetVelocity(clientID,Relbow2_roll3,-
v_E,vrep.simx_opmode_streaming)

vrep.simxSetJointTargetVelocity(clientID,Lelbow2_roll3,v_E,vrep.simx_opmode_streaming)
      time.append(ta)
      if obj1>-1.5708:
```

```
            obj2=obj1+1.5708
            d=abs(obj2*180/3.142)
            errorangle=(0.9 * math.sin(d))/t
            v2=v2+abs(0.5*errorangle)
            print errorangle, v2 ,obj1,'a'
            vrep.simxSetJointTargetVelocity(clientID,RShoulder,-v,vrep.simx_opmode_streaming)
            vrep.simxSetJointTargetVelocity(clientID,LShoulder,-v,vrep.simx_opmode_streaming)
            vrep.simxSetJointTargetVelocity(clientID,RShoulder2,-
v2,vrep.simx_opmode_streaming)
            vrep.simxSetJointTargetVelocity(clientID,LShoulder2,-
v2,vrep.simx_opmode_streaming)
            angleerror.append(obj2)

         elif obj1<-1.5708:
            obj2=obj1+1.5708
            d=abs(obj2*180/3.142)
            errorangle=(0.9 * math.sin(d))/t
            v=v+abs(0.8*errorangle)
            print errorangle, v ,obj1,'b'
            vrep.simxSetJointTargetVelocity(clientID,RShoulder,-v,vrep.simx_opmode_streaming)
            vrep.simxSetJointTargetVelocity(clientID,LShoulder,-v,vrep.simx_opmode_streaming)
            vrep.simxSetJointTargetVelocity(clientID,RShoulder2,-
v2,vrep.simx_opmode_streaming)
            vrep.simxSetJointTargetVelocity(clientID,LShoulder2,-
v2,vrep.simx_opmode_streaming)
            angleerror.append(obj2)


         else:
            vrep.simxSetJointTargetVelocity(clientID,RShoulder,-v,vrep.simx_opmode_streaming)
            vrep.simxSetJointTargetVelocity(clientID,LShoulder,-v,vrep.simx_opmode_streaming)
            vrep.simxSetJointTargetVelocity(clientID,RShoulder2,-
v2,vrep.simx_opmode_streaming)
            vrep.simxSetJointTargetVelocity(clientID,LShoulder2,-
v2,vrep.simx_opmode_streaming)
```

```
        break
    else:
      v = 0
      vrep.simxSetJointTargetVelocity(clientID,RShoulder,0,vrep.simx_opmode_streaming)
      vrep.simxSetJointTargetVelocity(clientID,LShoulder,0,vrep.simx_opmode_streaming)
      vrep.simxSetJointTargetVelocity(clientID,Relbow_roll3,0,vrep.simx_opmode_streaming)
      vrep.simxSetJointTargetVelocity(clientID,Lelbow_roll3,0,vrep.simx_opmode_streaming)
      vrep.simxSetJointTargetVelocity(clientID,RShoulder2,0,vrep.simx_opmode_streaming)
      vrep.simxSetJointTargetVelocity(clientID,LShoulder2,0,vrep.simx_opmode_streaming)

vrep.simxSetJointTargetVelocity(clientID,Relbow2_roll3,0,vrep.simx_opmode_streaming)
      vrep.simxSetJointTargetVelocity(clientID,Lelbow2_roll3,0,vrep.simx_opmode_streaming)

      break
  else:
    v = 0
    vrep.simxSetJointTargetVelocity(clientID,RShoulder,0,vrep.simx_opmode_streaming)
    vrep.simxSetJointTargetVelocity(clientID,LShoulder,0,vrep.simx_opmode_streaming)
    vrep.simxSetJointTargetVelocity(clientID,Relbow_roll3,0,vrep.simx_opmode_streaming)
    vrep.simxSetJointTargetVelocity(clientID,Lelbow_roll3,0,vrep.simx_opmode_streaming)
    vrep.simxSetJointTargetVelocity(clientID,RShoulder2,0,vrep.simx_opmode_streaming)
    vrep.simxSetJointTargetVelocity(clientID,LShoulder2,0,vrep.simx_opmode_streaming)
    vrep.simxSetJointTargetVelocity(clientID,Relbow2_roll3,0,vrep.simx_opmode_streaming)
    vrep.simxSetJointTargetVelocity(clientID,Lelbow2_roll3,0,vrep.simx_opmode_streaming)

    break
```

APPENDIX B (GANTT CHART FOR PSM 1)

No	Task	Week													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Title Discussion														
2	Journal and Conference Paper Searching														
3	Analyse Journal and Conference Paper Collected														
4	Chapter 2 Writing														
5	Chapter 1 Writing														
6	Experiment Planning														
7	Chapter 3 Writing														
8	Coding														
9	Draft Submission														
10	FYP Presentation														

APPENDIX C (GANTT CHART FOR PSM 2)

No	Task	Week														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Chapter 3 Writing															
2	Experiment 1															
3	Experiment 2															
4	Experiment 3															
5	Data Collection and Analysis															
6	Chapter 4 Writing															
7	Chapter 5 Writing															
8	Draft Submission															
9	FYP Presentation															