國立臺灣師範大學科技與工程學院電機工程學系

碩士論文

Department of Electrical Engineering

College of Technology and Engineering

National Taiwan Normal University

Master's Thesis

# 人型機器人於電動機車之平衡與轉向控制之應用

# Balance and Steering Control of a Humanoid Robot on an Electric Scooter Application

林謙

LIN, Chien

指導教授 ：包傑奇 教授

Advisor: Prof. Jacky Baltes

中華民國 110 年 9 月

September 2021

# Acknowledgment

很高興能有機會加入包傑奇(Jacky Baltes)教授的實驗室(Educational Robotics Centre)，在這個實驗室中擁有許多學習與犯錯的機會，從起初對機器人與人工智慧的嚮往,到學會並應用，參加比賽以及研討會。

非常感謝在學習的過程中有實驗是的夥伴與我共度 2 年的時光，Hanjaya , Christmann , Danny , Green , Song , Shane , Eko , Mike , Jaesik and Jeehyun ，無論是曾經一起努力或現在仍在實驗適中的成員，都在我低落或是失去方向的時候能夠給予我幫助並點亮我熱情的益友。而最感謝的是我們的教授 Jacky，給予我無限自由的發揮空間並不段鼓勵我要不停學習與堅持。記憶中最深刻的是參加 FITI 比賽時進入了前 40 強之後被所有的評委否定我們的努力，當時真的對自己的選擇產生了很大的動搖，懷疑選擇人型機器人可能是錯誤的選擇，但 Jacky 給我信心並帶領我們繼續堅持，被否定並不是失敗而是有更多的進步空間。無論選擇那一個領域都會遇到困難與磨練，會有無數的聲音試圖告訴我放棄，但實驗室的這些夥伴與我的指導教授總是在我的背後支撐我。

我從未想過在碩士的階段中能夠接觸到這麼多不同的想法和來自各地的研究生，韓國、巴西、伊朗、德國等等，也從未想過我會有榮幸參與機器人騎電動車的大計畫，跟最特別的一群人做最特別的研究。我會帶著 ERC 實驗是的精神去到新的地方，也期許在未來能有機會繼續共事。

最後獻上十二萬分的感謝給 Jacky 教授，感謝能成為您的學生！

# Balance and Steering Control of a Humanoid Robot on an Electric Scooter Application

Student: LIN, Chien Advisor : Prof. Jacky Baltes

Department of Electrical Engineering
College of Technology and Engineering
National Taiwan Normal University

## ABSTRACT

In recently, humanoid robots and autonomous vehicles are two famous and challenging fields. Many people focus on automation and intelligence, and desire to apply their expectations for the future to reality. From the perspective of general-purpose robots, the humanoid robots are capable of naturally operating in any real environment. That includes the humanoid robot operates the vehicles, which is an interesting challenge for the state-of-the-art in both fields.

The contribution of this thesis is operating a two-wheeled electric scooter in reality by using large sized humanoid robot. The inverses kinematic for operating the two-wheeled electric scooter in large sized humanoid robot is applied in four different inverse kinematic method and find the optimal solution to operating the steering motion. The different inverse kinematic includes Jacobian pseudo inverse kinematic, PSO inverse kinematic, hybrid inverse kinematic using Jacobian pseudo and PSO, and the Jacobian pseudo inverse kinematic with momentum. By using hybrid inverse kinematic method, a high efficiency motion result can be obtained, and combines the low computational time of iteration method, Jacobian pseudo inverse kinematic with momentum, we successfully have a fast inverse kinematic in an efficient motion planning.

To adapt the robot-scoot system, a 3D model for throttle and emergency brake system was developed, and implement a PID controller from physics-based simulation environments in previous work [34] to reality. Therefore, the robot-scoot system is applied fast inverse kinematic and PID controller for the balancing test. The result of steering motion is under the restrictions from our robot servo.

**Keywords:** Humanoid Robot, Two-Wheeled Vehicles, Inverse Kinematic, PID, PSO

# Table of Contents

# List of Figures

# List of Tables

# List of Symbols

| | |
|---|---|
| ML | Machine Learning |
| DL | Deep Learning |
| RL | Reinforcement Learning |
| PID | Proportional-Integral-Derivative |
| TWV | Two-wheeled Vehicles |
| FWV | Four-wheeled Vehicles |
| D-H | Denavit-Hartenberg |
| FK | Forward Kinematics |
| IK | Inverse Kinematics |
| EE | End Effector |
| PSO | Particle Swarm Optimization |

# Chapter 1. Introduction

## 1.1 Background and Motivation

### 1.1.1 Background

Humanoid robots have become an emerging and challenging research field during the past years and played a central role in robotics research and many applications. Since 1927, Herbert Televox, was the first ever humanoid robot, built by Ron Wensley. It made people start imaging the humanoid robots to be general-purpose robots. In [1], three main essential features of humanoid robots are declared as follows:

(i) Humanoid robots can work in the built-for-human environment.

(ii) Humanoid robots can use humans' tools.

(iii) Humanoid robots have a shape similar physical structures to actual humans, such as a torso with a head, two arms, and two legs.

As mention, humanoid robots are designed to be human like. Moreover, people expect humanoid robot can be general-purpose robot, not only human like but also act like human. From the view above, humanoid robots are more suitable for co-existing with people in an environment than other types of robots [3,4].

In nowadays, the development of deep learning [2], which make complex tasks are possible to be solved with heuristic data or limited knowledge. It enables people to tackle long-standing challenges in wide variety of fields, from chemistry to medicine and engineering. People also start making a toward challenges that the humanoid robot drive the two wheeled vehicles. The function of two wheeled vehicles, such as bicycles and motorcycles, are very different from four-wheeled vehicles. Two-wheeled vehicles are not self-stable when it's not moving or in low speed. Humanoid robot controlling the two wheeled vehicles will be an interesting issue and may prove a valuable benchmark regarding the state-of-the-art in control.

The study topic uses the large-sized humanoid robotic THORMANG3 and a commercially available electric scooter which manufactured by the Gogoro company[1]. In Figure 1.1 shows the THORMANG3 and Gogoro scooter.

[1] https://www.gogoro.com/tw/

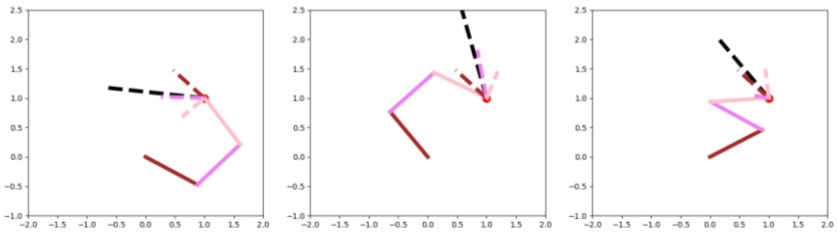
Figure 1.1: THORMANG3 robot on the Gogoro scooter.

### 1.1.2 Motivation

To manipulate humanoid robot operating a two-wheeled vehicle and balance when vehicle is moving in a floating speed range, a considerable degree of precision is required to apply the correct steering and an efficient movement of steering action is critical to a real time robot-scooter system which need to react as fast and efficiency as possible.

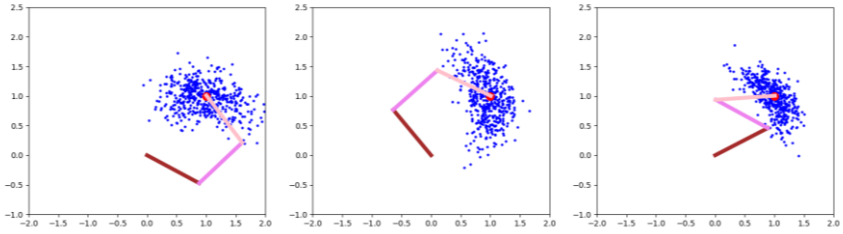
In this thesis, the purpose is using different poses of robot arm and optimize the solution for robot arm operating the two-wheeled electric scooter. For instance, a robot arm that more than two-link joint has more than one answer can achieve to the desired position. However, the different poses may have their efficiency direction to move. Figure 1.2 shows the efficiency value in different poses when all joint rotate counterclockwise. The dashed line direction is corresponding to its link color and it is computed by link movement in 0.1 radians (~5 degrees) effects to EE, and the length of dashed line is the magnitude of movement. The black dashed line combines all the component of the link movement, and the length of line means the efficiency value to this direction.

Moreover, the Figure 1.3 shows three different poses that add the white noise with μ is 0 and σ is 0.2,$N(0,0.2)$, into the joint value and plot the EE with the blue point to see how sensitivity to noise it takes from the pose. In hence, we can observe different poses resist to the noise from the blue point range from the result in Figure 1.3, and the Figure 1.3 (c) is more resistance to the noise due to the blue points are more concentrated.



(a) (b) (c)

Figure 1.2 : Three different poses and their efficiency direction with all joint do counterclockwise rotation.



(a) (b) (c)

Figure 1.3 : Three different poses and their sensitivity to noise.

However, each link could have two direction to move, counterclockwise and clockwise. By computing all the possible condition, the maximum value of efficiency and each link of movement are show in Figure 1.4. Thus, the black dashed line and its opposite side are the most efficiency direction to move in their pose. And the examples are not consider the IK constrain about collision but only for visualising the efficiency direction and sensitivity to noise.

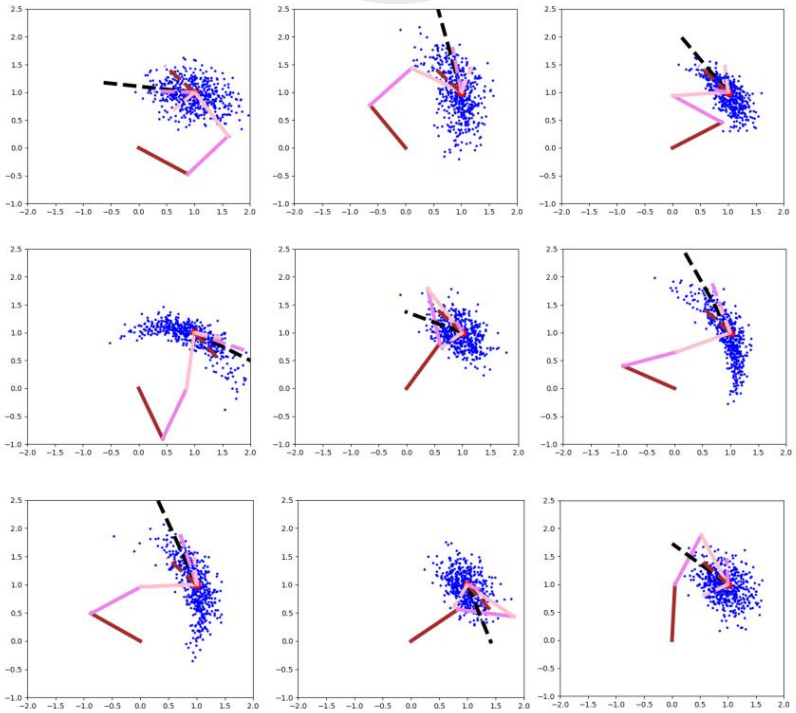


Figure 1.4 : Nine different poses with their most efficiency direction and sensitivity to noise.

## 1.2 Research Aim and Objectives

### 1.2.1 Research Aim

In order to fulfill the desire for general-purpose robotics it is necessary to be able adapt and perform in a wide range of human environments and tasks. An interesting task that has not been widely discussed in past research is humanoid robotic operation of two-wheeled vehicles. The main work is large-sized humanoid robotic operating an electric scooter.

### 1.2.2 Objective

- Develop a 3D object to assist humanoid robot operating the throttle of electric scooter.
- Develop a throttle limited to guarantee the range of acceleration.
- Develop an emergency brake system to interrupt the operation and stop the electric scooter.
- Develop a steering-based PID controller to balance the robot-scooter system.
- Develop different inverse kinematics for humanoid robot steering motion.
- Evaluate, compare and discuss the method and results on different inverse kinematics.
- Apply developed IK in real robot and compare revolving restriction to actuator.

## 1.3 Structure of the Thesis

The rest of the thesis is organized as follows. Chapter 2 presented an overview of the literature review on two-wheeled vehicle dynamics, inverse kinematics and PSO algorithm. Chapter 3 describes the methodology of a humanoid robotic operates an electric scooter, which illustrates the specification of large-sized humanoid robotic THORMANG3 and electric scooter Gogoro scooter. Moreover, Chapter 3 describes the robot-scooter system fix 3D objects for robot-scooter to do emergency break and throttle in low speed range, the target position for humanoid robot operates scooter steering motion, PID controller, forward kinematics and difference method of inverse kinematics, including Jacobian pseudo IK, PSO IK, hybrid IK and Jacobian pseudo IK with momentum method to control humanoid robot operating the electric scooter. Chapter 4 provides the experiment results of Jacobian pseudo IK, PSO IK, hybrid IK and Jacobian pseudo IK with momentum method, comparing results from Jacobian pseudo IK and Jacobian pseudo IK with momentum, evaluating Jacobian pseudo IK with momentum to optimal solution from hybrid IK results with efficiency steering motion. Then, comparing the result of PID in simulation with reality result, and the robot servo restriction. Chapter 5 shows the conclusion and future work.

# Chapter 2. Literature Review

## 2.1 Two-wheeled Vehicle Dynamics

Compare with four-wheeled vehicles, two-wheeled vehicles are more difficult to keep balance according to its designed structures. Two-wheeled vehicles required continuous input to maintain stability [5]. In addition, due to the dynamics of the system [6], the two-wheeled vehicles can get into self-stable when the speed is higher than 5 m/s. However, reaching the high speed with the self-stable range means riskier if the controller applies with unexpected small error for steering commands, the momentum will effort to the system and flip over the two-wheeled vehicles immediately. It is necessary to have an accuracy balancing controller for control the two-wheeled vehicles.

In the literature, balancing controllers can be classified into three main categories [7].

(i) Controlling balance by controlling the steering angle [8-11].

(ii) Controlling balance by changing the center of mass [12-14].

(iii) Controlling balance by flywheel [15,16].

Steering based controller in (i), which aims to achieve balance by controlling the angle of the steering bar handle. Manipulate the center of mass in (ii), which usually consists in installing a balancer subsystem that can manipulating the rod of mass. And the flywheel system in (iii), which certainly important and relevant to the research community but not align with our goal of having a humanoid robot operates two-wheeled vehicle. In (ii), if consider the robot to be a center of mass controller as well, the robot's upper body is analogous to rod of mass in such system. However, the robot has to be fix on the scooter as shown in figure 3.6 which incapacitated the robot by the safety factor and required effective observation data from robot sensors be reliable to estimate the pose of robot-scooter system. As stated above, steering-based (i) controller applied to a robot-scooter system is important.

As indicated that the robot fixed on the scooter, the robot-scooter system can model as an inverted pendulum model [17]. The model can be seen in Figure 2.1, where the mass of the bicycle is concentrated at a specific point, located at a height h above the ground level and at a distance d from the rear-wheel. R is the turning radius, $\alpha$ is the front-wheel angle, $\theta$ is the roll angle, w is the wheelbase, $\psi$ is the rear-wheel yaw angle and the pair (x,y) are the Cartesian coordinates of the bicycle displacement. In this type of model, it's possible to design a steering controller to stabilize the system such as in [18].

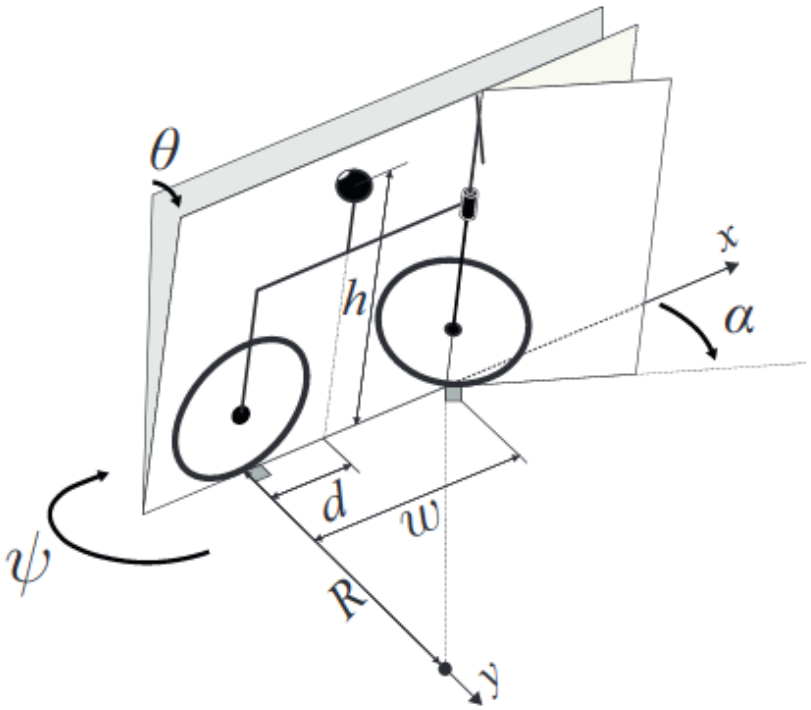



Figure 2.1: Bicycle modeled as an inverted pendulum model [17].

In [19], a dynamics of bicycle system is greatly analyzed including higher-order and non-linear models. From Figure 2.2, we can infer that the angular velocity around the vertical z axis is $\omega = \frac{V\delta}{b}$. And the bicycle momentum can be computed as below:

$$J\frac{d^2\varphi}{dt^2} - mgh\varphi = \frac{DV}{b}\frac{d\delta}{dt} + \frac{mV^2h}{b}\delta. \quad (2.1)$$

Where J is the moment of inertia with respect to the x-axis, and$D = -J_{xz}$ is the inertia product with respect to the xz axes, m is the total mass of the system, a is the distance from center of mass to center of rear wheel, b is the distance between center of front wheel and rear wheel, V is the velocity along the x axis, h is the distance from the center of mass to the ground, δ is the steering angle, and φ is the roll angle.

Apply the control law (2.2) for this model, and k gain from [19] analysis is required $k > \frac{bg}{V^2}$to stable the system.

$$\delta = -k\varphi \quad (2.2)$$

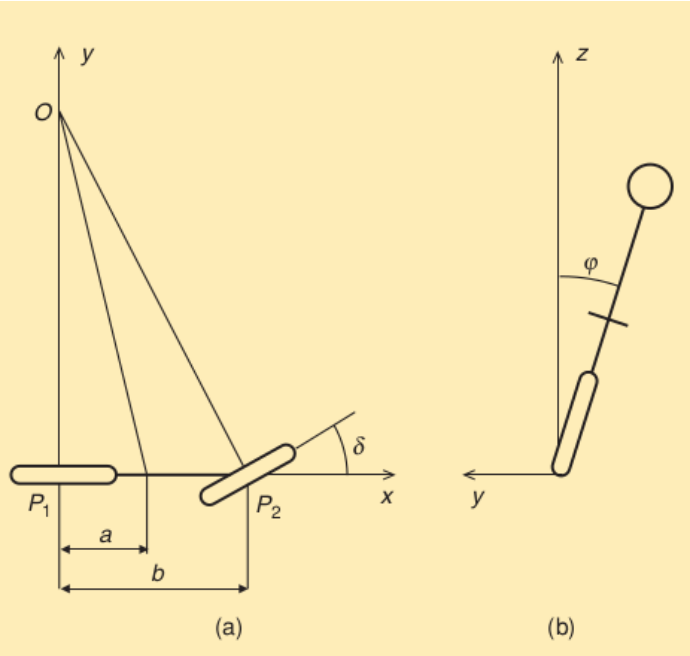



Figure 2.2: (a) Top view and (b) front view of bicycle model.

## 2.2 Inverse Kinematics

For humanoid robot operating a scooter, high accuracy and low time cost motion are necessary to make an efficiency effect to the robot-scooter system. To reach high accuracy and low time cost, inverse kinematics is the solution of robot motion planning [20]. In addition, the author mentions some relative work for knowing the limitation before starting inverse kinematics. In [21], the target is in reachable or non-reachable and its work space should be clarified as shown in Figure 2.3. Once the robot model has been defined, it can locate the end effector position by forward kinematic, and the inverse kinematic method can decide the appropriate joint value that can move the end effector to desired target position.

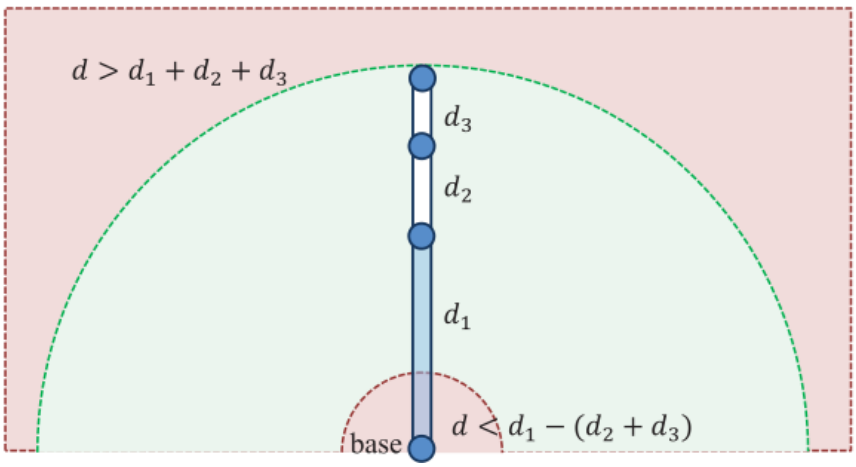



Figure 2.3: The reachable and unreachable cases of IK problem. The reachable target position should between $d_1 + d_2 + d_3$and$d_1 - (d_2 + d_3)$[21].

To build the kinematic model for our robot arm, the most population approach is used Denavit-Hartenberg [22] to represent the kinematic structure and this approach also calculate the end effector with knowing joints angle values. The concept of inverse kinematic method is when a target was set and calculate backward from the target. Inverse kinematic methods are classified to four methods:

- **Analytical Methods**: Analytical methods are meant to find all possible solutions as a function of structure is given, show as Figure 2.4.
- **Numerical Methods**: Numerical methods are using iterations to achieve a satisfactory solution by a cost function, the difference between end effector and target. The numerical methods can be divided into three categories: Jacobian, Newton and Heuristic method. In general, Jacobian and Heuristic method are most often used.[22,23]
- **Learned Methods**: Learned methods can divide into two groups: the error-based and the example-based method. Error-based methods improve the inverse kinematic estimate that is used to reach the desired target position. Example-based methods use example configuration data for learning the inverse kinematic estimate. In recently, the deep learning methods are amount used to inverses kinematic research.[24,25]
- **Hybrid methods**: Hybrid methods combines two or more methods to solve inverse kinematic problem that can reduce the computation times and eliminate error as possible.[26,27,28,29]

An analytical method of a two-link kinematic have two lengths $l_1$ and $l_2$ and the target position with $(x, y)$. The joint angles can be determined as:

$$\theta_1 = cos^{-1}\left(\frac{l_1^2+x^2+y^2-l_2^2}{2l_1\sqrt{x^2+y^2}}\right) \tag{2.2}$$

and

$$\theta_2 = cos^{-1}\left(\frac{l_1^2+l_2^2-(x^2+y^2)}{2l_1l_2}\right) \tag{2.3}$$

In Figure 2.4, it's obvious there are two possible solutions can reach the target position $(x, y)$.

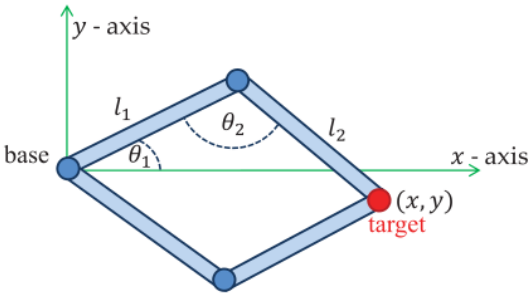



Figure 2.4: An analytical method of a two-link kinematic.

The Jacobian methods offer a linear approximation to the inverse kinematics problem. The Jacobian matrix J can be described as:

$$J(\theta)_{ij} = \left(\frac{\partial s_i}{\partial \theta_j}\right) \tag{2.4}$$

Where $i = 1, \ldots, k$ and j1, … , $n$ , the k is the number of end effector and the n is the number of joints. Thus, the Jacobian matrix J should be a $kxn$ matrix which the vectors of end effector ($s_i$) effected by a small changed with joints value ( $\theta_j$ ). Therefore, the Jacobian matrix J, which is a linear approximation formula, can be consider as a coordinates matrix and move the end effector to the target position. In Figure 2.5, the linear approximation of the actual motion has the error, then using iterations to obtain a satisfactory solution to correct the error.

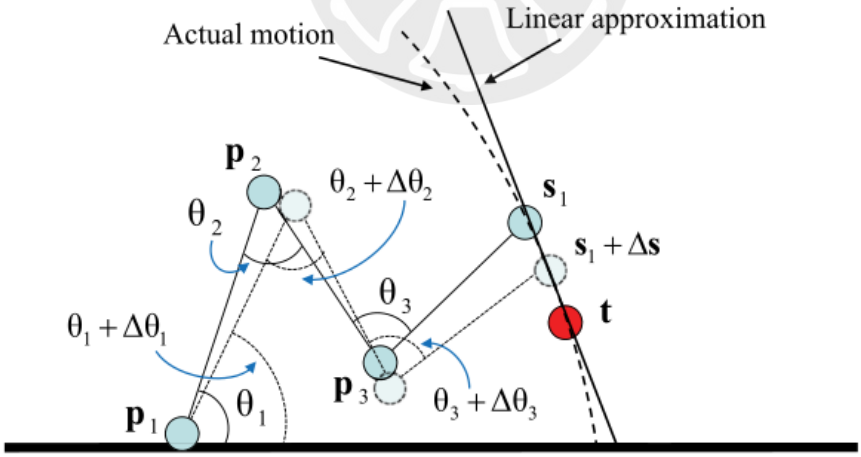



Figure 2.5: The linear approximation of the actual motion.

## 2.3 Particle Swarm Optimization (PSO)

PSO algorithm [33], which is a computational method that optimizes a problem. By having the group of particles with random initial speed vector and evaluate personal best and global best, the particles update speed vector in each generation until reach the target. The Figure 2.7 shows the flow chat of PSO algorithm.

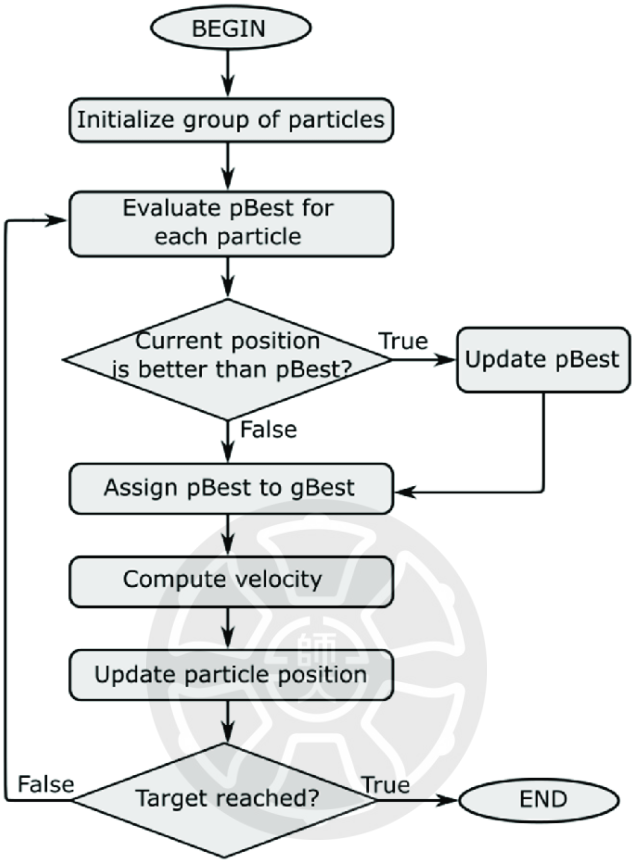



Figure 2.7: The flow chart of PSO algorithm, the particles update in each generation until reach the target.

The particle behavior in a PSO can be modeled as:

$$V_i(t+1) = w \times V_i(t) + c_1 \times rand() \times \left(pbest_i - x_i(t)\right) + c_2 \times rnad() \times \left(Gbest - x_i(t)\right) \quad (2.7)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2.8)$$

c1 and c2 are acceleration constants, rand() is random number between 0 and 1, w is inertia weight factor, Gbest is the best previous position among all the particles and pbest is the best previous position of particle i.

# Chapter 3. Methodology

## 3.1 The Robot - THORMANG3

The robot used in this paper was a humanoid robot THORMANG3 (Tactical Hazardous Operations Robot3), shows in Figure 3.1. THORMANG3 is a full-size commercially available humanoid robot developed by the Korean company ROBOTIS[2]. It has two Intel NUC computers, DLink DIR-806A wireless router, 29 degrees of freedom (DoF), a MicoroStrain 3DM-GX4-25 IMU and several other sensors. Table 3.1 shows the complete specifications of THORMANG3, and Figure 3.2 Shows a complete diagram of THORMANG3 system and interfaces. THORMANG3 is a full-size humanoid robot has a bit over 137 centimeters of height, it is a suitable size to sit on the scooter and manipulate the steering handle as human-like behavior.

Figure 3.1: THORMANG3: Tactical Hazardous Operations Robot 3.

| | |
|---|---|
| Degrees of Freedom | 29 |
| Actuator | 200W x 10 / 100W x 11 / 20W x 8 |
| Computer | Intel NUC (i5 Processor) – 8GB RAM DDR4 x 2 |
| Wireless Router | DLink DIR-806A |
| Camera | Logitech C920 HD Camera |
| LiDAR | Hokuyo UTM-30LX-EW |
| F/T Sensor | ATi Mini58-SI-2800-120 x 2 |
| IMU | MicroStrain 3DM-GX4-25 |
| Battery | 22V, 22000mA – 18.5V, 11000 mA |
| Height | 137.5 cm |
| Weight | 42 kg |

Table 3.1: Hardware specifications of THORMANG3.

[2] https://www.robotis.com/

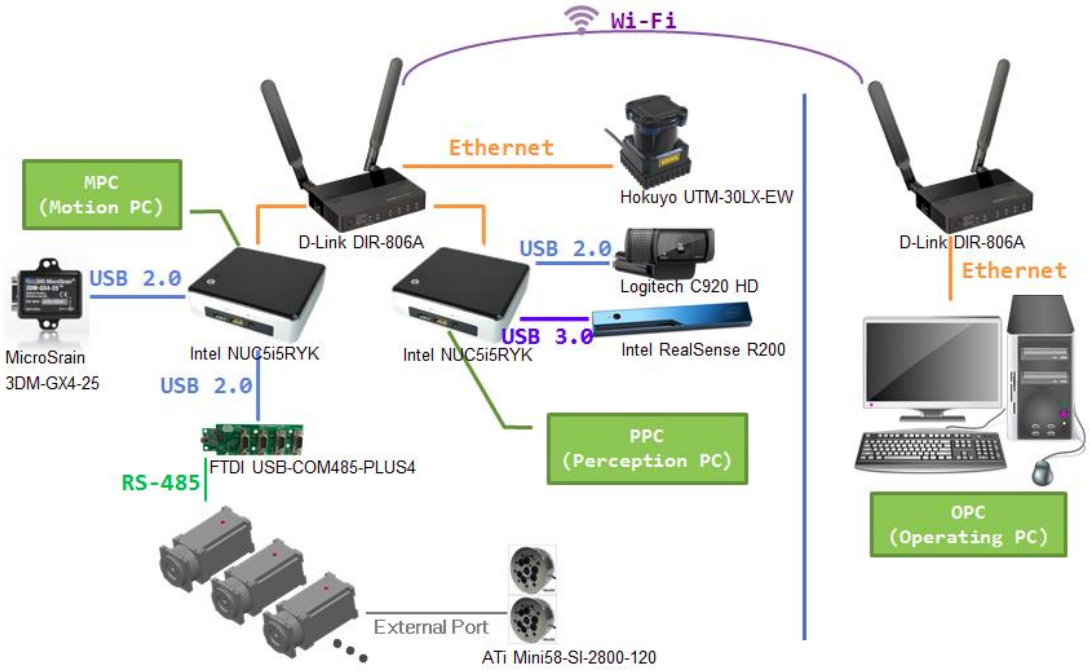



Figure 3.2: THORMANG3 system and interfaces.

## 3.2 The Two-wheeled Vehicle - Gogoro scooter

The two-wheeled vehicle we utilized was a commercially available Gogoro VIVA Plus scooter, shows in Figure 3.3. Gogoro VIVA Plus is a electric scooter, it is a good choice for our purposed goal because the acceleration is more stable than scooter used gasoline. Figure 3.4 and Table 3.2 show more details and specifications about Gogoro VIVA Plus. And the inertia matrix for whole body is (3.1), and the unit is in $kg \cdot m^2$.

$$\begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} = \begin{bmatrix} 4.40515 & 0.117584 & -1.66168 \\ -0.117584 & 18.0137 & 0.098875 \\ 1.66168 & -0.098875 & 14.8791 \end{bmatrix} \tag{3.1}$$

Figure 3.3: Gogoro VIVA Plus scooter.

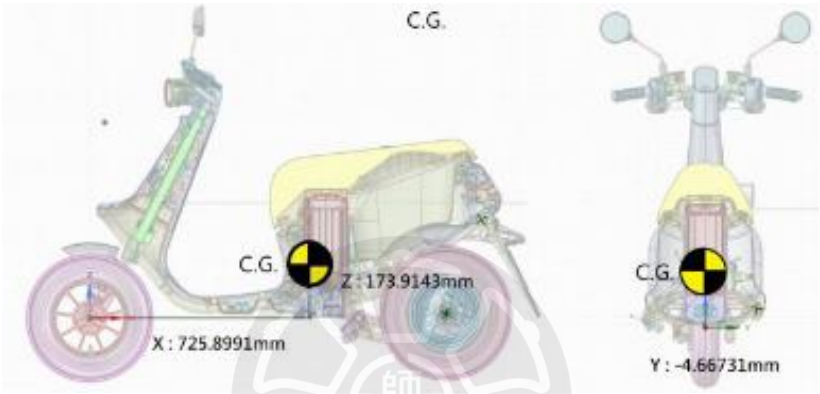



Figure 3.4: Center of Gravity of the Gogoro VIVA Plus scooter.

| | |
|---|---|
| Wheel base | 1164 mm |
| Seat height | 740mm |
| Weight (with batteries) | 79.2 kg |
| Max power output | 3kW @ 500 rpm |
| Max horsepower | 4.02 hp @ 500 rpm |
| Max torque (Motor / Wheel) | 115 Nm @ 200 rpm |
| Hill climb ability | 20% (11°): 25km/h<br>10% (6°): 40 km/h |
| Max lean angle | Left: 36°<br>Right: 41° |
| Max riding range p/ battery | Approx. 85 km |

Table 3.2: Specifications of the Gogoro VIVA Plus scooter.

## 3.3 Robot-Scooter System

Before testing the system in reality, the system must test in the Gazebo simulation environment with ODE physics engine to avoid unexpected damage occurred by rough testing in real world. The 3D robot-scoot system model in Gazebo simulator is presented in Figure 3.5.

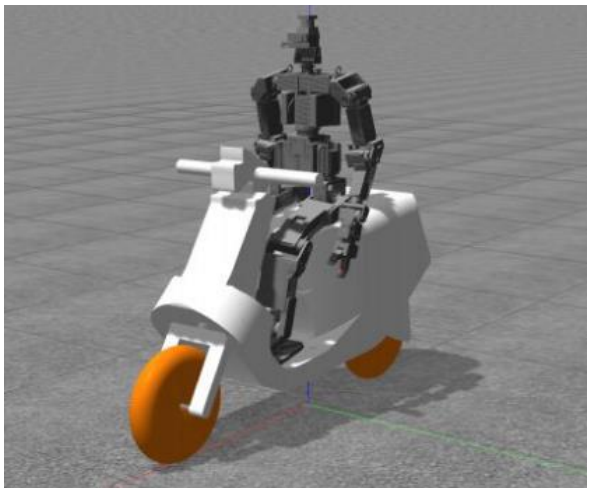
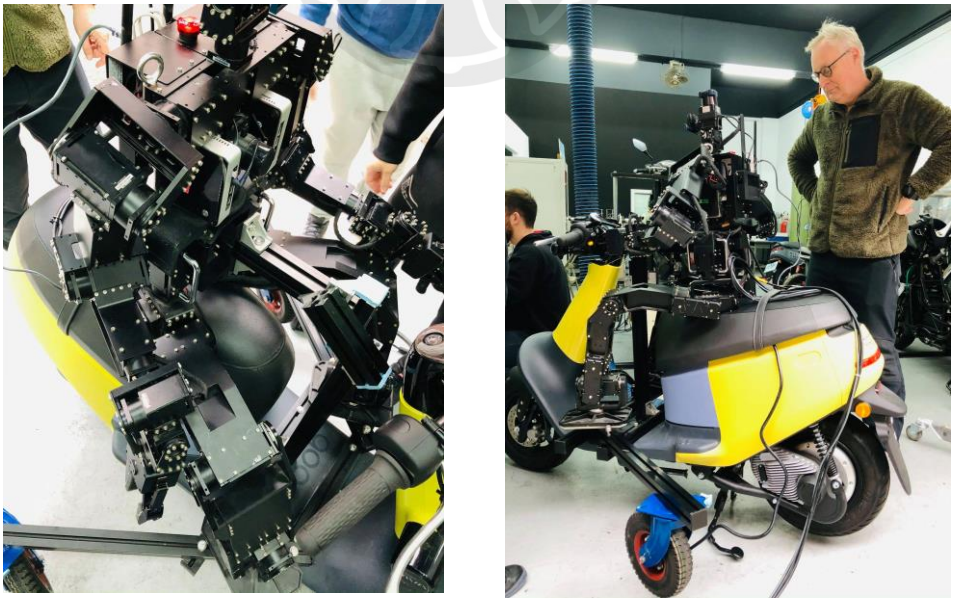


Figure 3.5: The 3D robot-scoot system model in Gazebo simulator.

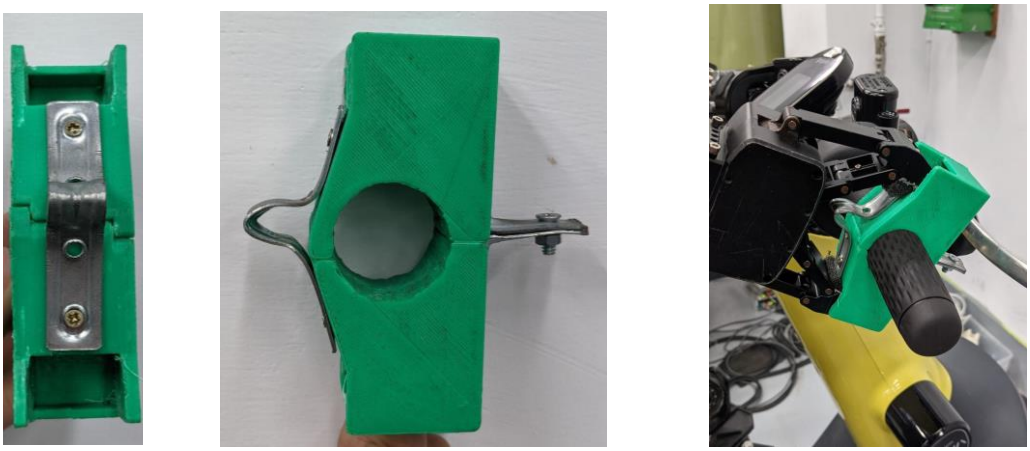
In order to have a safety experiment in real test, the robot THORMANG3 operating Gogoro VIVA scooter should be with some security measure. Thus, the robot THORMANG3 fix to Gogoro VIVA scooter with profiles and added the training wheels to prevent the robot-scoot falling down, show as Figure 3.6.



(a) (b)

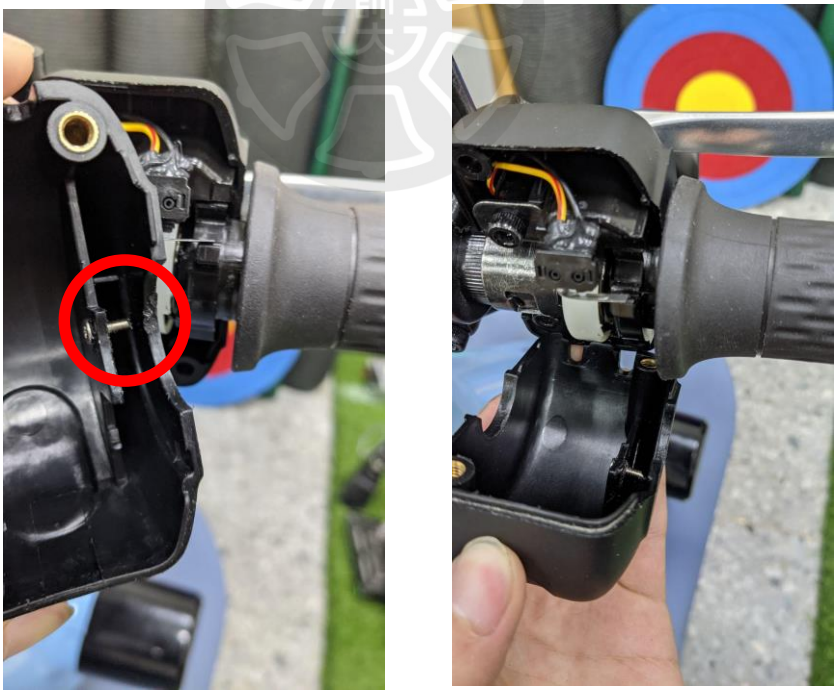
Figure 3.6: (a) Fixed robot to scooter. (b) Added the training wheels.

In Figure 3.7, a 3D support equipment for robot operating the throttle is used to suit the actual situation that the scooter throttle is hard to accuracy operate for robot gripper due to the shape is easy to slack. In Figure 3.8, a screw apply in throttle system to exhibit the angle which the throttle can turn, so the acceleration will be limited for safety.



(a) (b) (c)

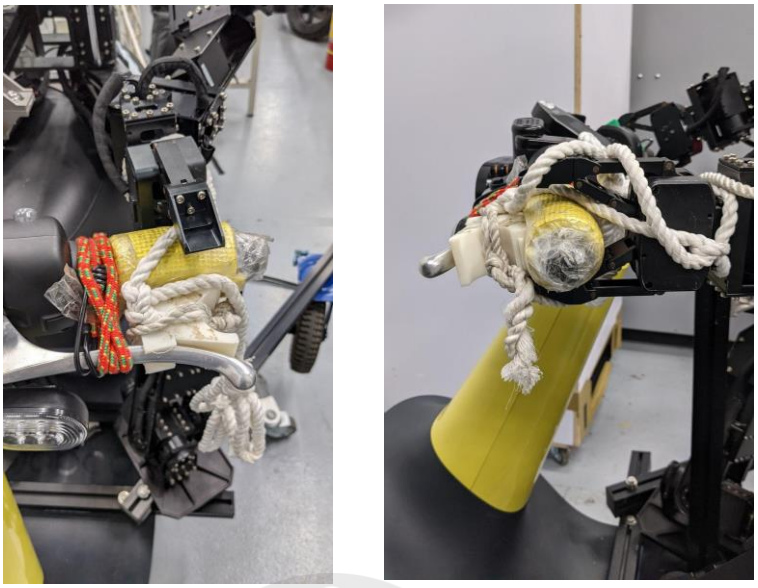
Figure 3.7: (a) Front view of 3D support equipment. (b) Side view of 3D support equipment. (c) 3D support equipment installed on the throttle.



(a) (b)

Figure 3.8: (a) Applied a screw in throttle system. (b) After applied the screw, closed the case then the throttle angle will be exhibited.

Furthermore, an emergency system is very important and necessary. Thus, a simple pattern is applied to tie up the breaker with elastic rope and put the blocker between the breaker. Once the blocker has been pulling out, the rope will tightly tie up the breaker, which means it will switch on the breaker, and stop the scooter. Figure 3.9 is our emergency system.



(a) (b)

Figure 3.9 :(a) Top view of emergency system. (b) Side view of emergency system.

## 3.4 PID Balance Controller

PID controllers are a commonly used in control filed. A PID controller is developed by following the control low Equation (2.2) from two-wheeled vehicles dynamics. Then, the PID controller experiment was evaluated using Gazebo simulator, which is a popular simulation environment for robotics. In Figure 3.10, a rear view of robot-scoot model is presented, and the tilt angle is represented by $\Phi$.

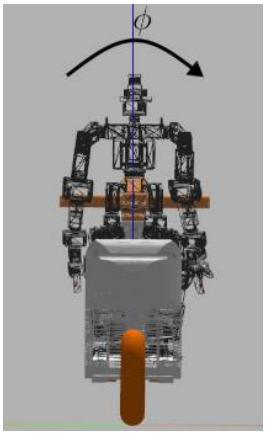



Figure 3.10: Rear view of robot-scoot model.

The balance control is a simple controller based on Equation (2.1). The error signal can be calculated from the difference between a desired tilt angle and the current tilt angle:

$$Error = \Phi - \Phi_{ref} \tag{3.2}$$

The tilt angle data of the robot ($\Phi$) came from IMU sensor is read at a rate of 250 Hz.

In the simulation, the reference angle is defined to be 0 which is a up right angle for robot-scoot system. However, in the real robot-scooter system, there could exist an uncertain error between the tilt angle computed by the IMU and the robot initial position, thus the computed roll angle for feedback is $\Phi - \Phi_b$ instead of θ, where $\Phi_b$ is an offset, [30].

In order to obtain a good performance, different gains were explored. The gains are presented in Table 4.1 in the Experimental Results chapter.

## 3.5 Target Position of Manipulate E-Scooter

In our main goal, we have to define a work space for steering the Gogoro head bar after designed an accuracy and low iteration IK. The Figure 3.11 to Figure 3.14 are the measured space for steering Gogoro head bar.

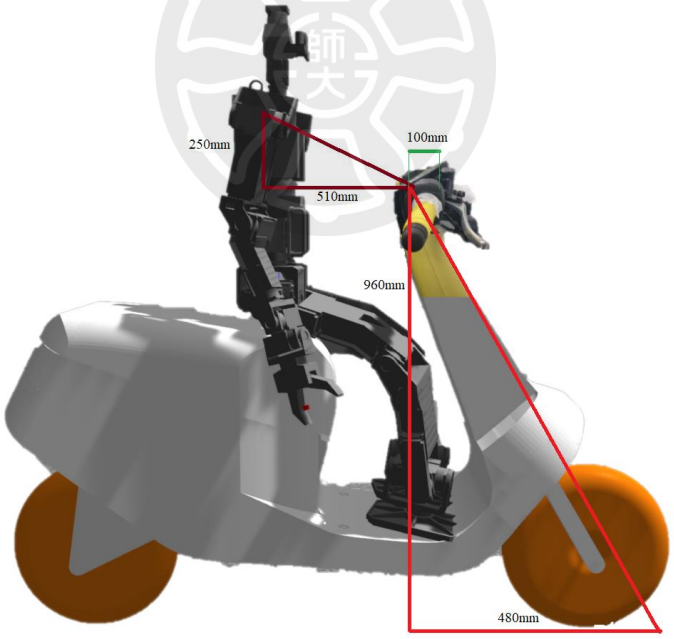



Figure 3.11: Side view (x-z plan) for measured Thormang3 and Gogoro system.

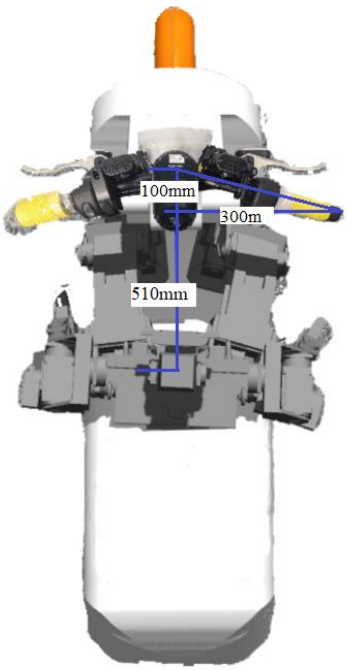



Figure 3.12: Up view (x-y plan) for measured Thormang3 and Gogoro system.


Figure 3.13: Up view (x-y plan) for measured Gogoro head bar.

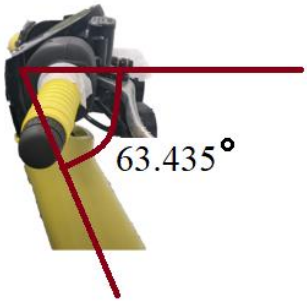



Figure 3.14: Side view (x-z plan) for measured Gogoro head bar.

In Figure 3.11, it shows the x position is 510 mm and z position is -250 mm from thormang3 shoulder, and Figure 3.12 is upper view for robot-scoot. Figure 3.13 is closer view for steering head bar shows y position should below 300 mm, and shows head bar angle is 18.435° to vertical, thus the robot gripper yaw should be 18.435° to the head bar. and Figure 3.14 shows the head bar is 63.435° to the ground, and it effects to compute the height point (z-axis) when the steering.

Therefore, we defined a 0° steering position with x_start_pos is 510 mm, y_start_pos is 248 mm, z_start_pos is -250 mm, head_bar_extend_angle is 18.435 deg, desired_steering_angle = θ.

Then the target position Equation to control the Gogoro steering can be computed as below.

$$left\_x\_target = x\_start\_pos + y\_start\_pos * \sin(\theta) \quad (3.3)$$

$$right\_x\_target = x\_start\_pos - y\_start\_pos * \sin(\theta) \quad (3.4)$$

$$left\_y\_target = y\_start\_pos * abs(\sin(\theta)) \quad (3.5)$$

$$right\_y\_target = -y\_start\_pos * abs(\sin(\theta)) \quad (3.6)$$

$$z\_target = z\_start\_pos + y\_start\_pos * \sin(head\_bar\_extend\_angle) * \cos(90 \pm \theta) * \sin(63.435°) \quad (3.7)$$

The Equation (3.3) is the x axis position for the left arm, and the Equation (3.4) is the x axis position for the right arm. The Equation (3.5) is the y axis position for the left arm, and the Equation (3.6) is the y axis position for the right arm. The Equation (3.7) is the z axis position for the left arm and the right arm.

## 3.6 Forward Kinematics

To build the robot arms kinematics, the specification from the robot THORMANG3 is important. In Figure 3.15 shows THORMANG3 joint link specification[3], and the Table 3.3 shows the coordinate distance relationship between each joint. Therefore, Denavit-Hartenberg or similar method can be used to build the robot arms kinematics. Our arms kinematics D-H parameters are show in Table (4.1) and Table (4.2).

[3] https://emanual.robotis.com/docs/en/platform/thormang3/references/#physical-properties

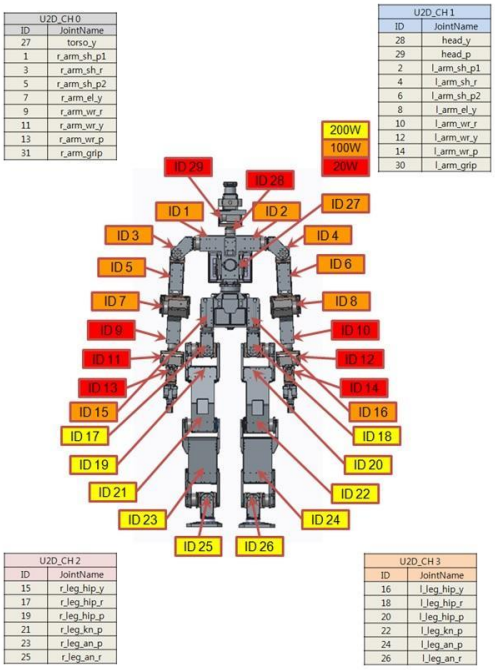



Figure 3.15: THORMANG3 joint link specification

| L_Arm | x | y | z | R_Arm | x | y | z |
|---|---|---|---|---|---|---|---|
| 27-2 | 0 | 152 | 160 | 27-1 | 0 | -152 | 160 |
| 2-4 | 57 | 60 | -39 | 1-3 | 57 | -60 | -39 |
| 4-6 | -57 | 33 | -0 | 3-5 | -57 | -33 | 0 |
| 6-8 | 30 | 187 | 57 | 5-7 | 30 | -187 | 57 |
| 8-10 | 171 | 30 | -57 | 7-9 | 171 | -30 | -57 |
| 10-12 | 39 | 0 | 45 | 9-11 | 39 | 0 | 45 |
| 12-14 | 45 | -45 | -45 | 11-13 | 45 | 45 | -45 |
| 14-30 | 87.7 | -58.3 | 0 | 13-31 | 87.7 | -31.7 | 0 |
| 30-30_2 | 0 | -26.6 | 0 | 31-31_2 | 0 | -26.6 | 0 |

Table 3.3: The coordinate distance relationship between each joint.

## 3.7 Jacobian Pseudo Inverse Kinematic

In nowadays, manipulator in humanoid robot are principally using inverse kinematic (IK). In [31], a comprehensive survey of humanoid robot motion planning based IK are reviewed, and Jacobian pseudo IK methods are widely used in solving target position problem of motion planning [32]. The Jacobian pseudo inverse Equation shows as below:

$$\Delta\theta = J^T(JJ^T)^{-1}e \tag{3.8}$$

Where Jacobian matrix J is described in Equation (2.4), and $\Delta\theta$ is the delta joint value that can close to the desired target position.

## 3.8 Inverse Kinematics using Adapted PSO

However, there has infinite solution for IK, and we expect to find a optimal solution that can manipulate the scooter steering as efficiency as it can. The PSO algorithm is a famous method for optimization, but different from IK calculation the PSO algorithm start with random initial position instead of current position. In [35], it adapted the PSO algorithm to IK by setting the initial position to current position and this algorithm will reset the particles to avoid the solution get stuck with the local solution in every 10 steps. And the Figure 3.16 shows the flow chat of adapted PSO algorithm.

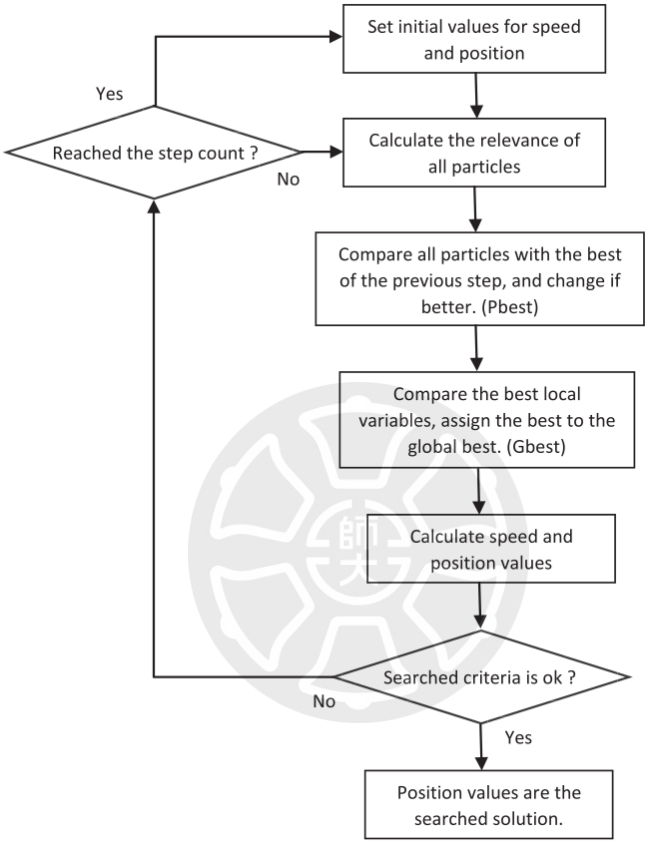



Figure 3.16: The flow chart of adapted PSO algorithm.

## 3.9 Hybrid Inverse Kinematics using Jacobian Pseudo and PSO

The adapted PSO algorithm generates a random initial speed that does not searching efficiency and the sensitivity to noise, and the $pbest_i - x_i(t)$ from the Equation (2.7) pulls the particles to previous position which is not a proper way to solve the IK problems. To improve the adapted PSO algorithm to IK problem, we use Jacobian pseudo method to calculate the initial speed, using $\Delta\theta$ in Equation (3.8) to the initial speed, and change the $pbest_i - x_i(t)$to a random vector that will explore more, the $pbest_i - x_i(t)$becomes to normal distribution with μ is 0 and σ is 0.001 , $N(0,0.001)$.

To find a optimal solution that can manipulate the scooter steering as efficiency as it can. The fitness function should not only compare to the distance error for IK but also takes efficiency into account. Equation (3.9) measures the efficiency:

$$E = (d \cdot s) \sum_{joints} (F_{joint} \cdot s) \tag{3.9}$$

The E is the value of efficiency, d is the vector between target position and current position, the s is the displacement with changing position, and F is the movement of each joint link with joint angular velocity in small degrees.

The sensitivity value is relative to the joint value effects to EE, the more error from the EE means the kinematic pose is less resistance to noise which is used the white noise N(0,0.01) in Equation (3.10).

$$sensitivity = \frac{1}{t} \times \sum_t \sum_{joints} |EE(joint + noise) - EE(joint)| \tag{3.10}$$

Therefore, the fitness function combines the efficiency and the sensitive and shows below:

$$fitness = \frac{1}{\frac{1}{|d|} + E} + sensitivity \tag{3.11}$$

And the best population is selected by the smallest value from fitness value.

## 3.10 Jacobian Pseudo Inverse Kinematic with Momentum

Although the adapted PSO algorithm can have chance to find optimal solution, it spend too much time to compute the kinematic model, the more population size it has the more computation time it cost. From the PSO algorithm behavior in Equation (2.7), the speed elements contain a random speed of momentum and a best moving direction to find the optimal solution. Refer to PSO algorithm, we apply the momentum to Jacobian pseudo IK method.

Therefore, in Equation (3.12), a new method of Jacobian pseudo IK acquires momentum vector with previous value and will keep accumulation.

$$V_{i+1} = V_i + \Delta\theta, V_0 = 0 \tag{3.12}$$

But if the accumulate momentum $V_i$ will lead to overshot, which cause the error from next position to target position be bigger, then reset momentum vector $V_i$ to zero. The Figure 3.17. illustrates the reset situation when momentum velocity cause an overshot in 3rd step.

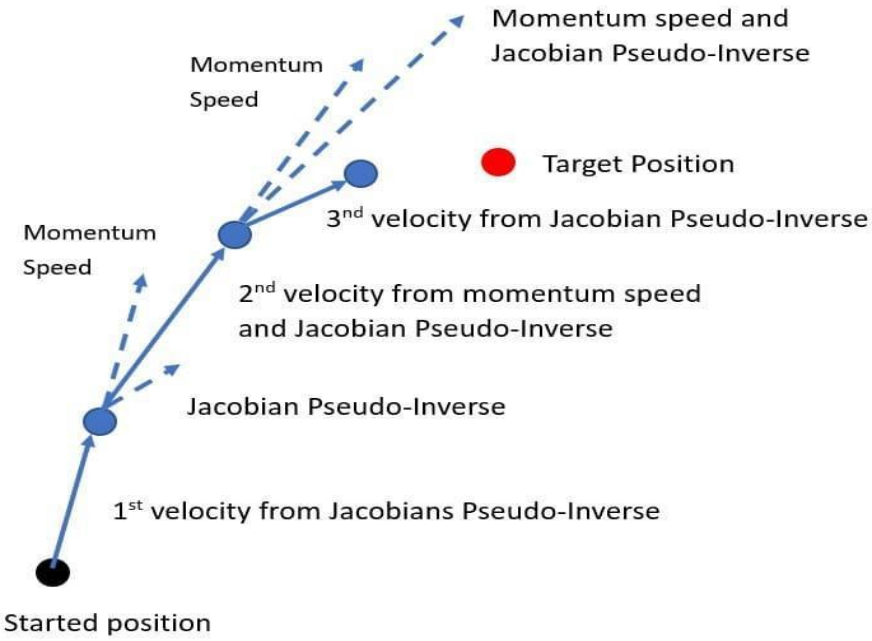



Figure 3.17: The momentum rests to zero when momentum velocity causes an overshot in 3rd step.

# Chapter 4. Experiment and Results

## 4.1 Experiment Result for Thormang3 Forward Kinematics

Building a kinematics module uses the information from Table 3.3 and Figure 4.1, which has specific physical properties for Thormang3 and the graphic for D-H representation of kinematics. Therefore, a Thormang3 left arm D-H parameters shows in Table 4.1, where $\theta_1$ to $\theta_7$ are corresponding to ID 2,4,6,8,10,12,14 in Figure 3.11, and a 3D kinematics module presents in Figure 4.2 and the Thormang3 right arm D-H parameters shows in Table 4.2. The result of kinematic module for Thormang3 both arm shows in Figure 4.2.

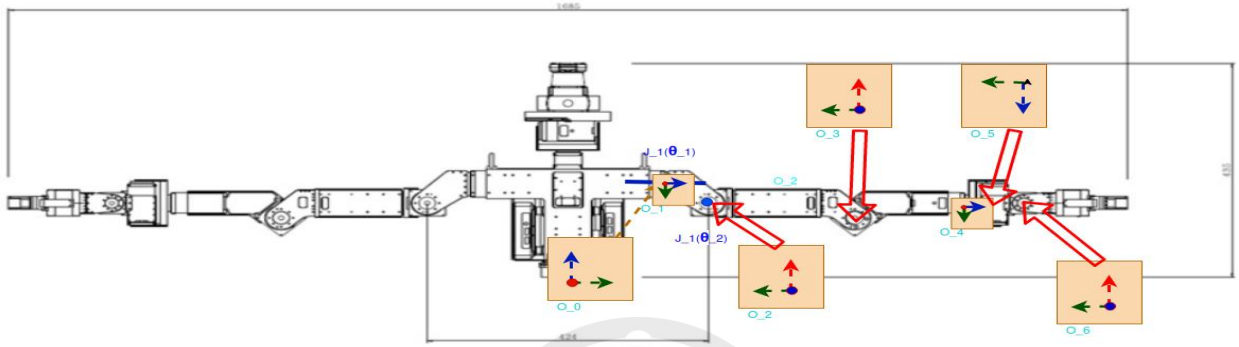


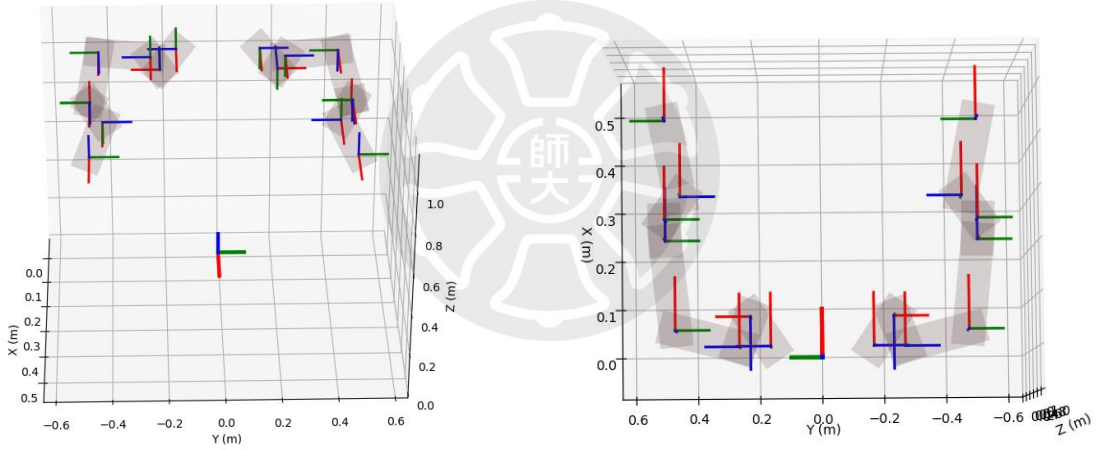
Figure 4.1 : Graphic for D-H representation of kinematics.

| | θ | *d* | *a* | α |
|---|---|---|---|---|
| 1 | 0 | 1053.5 mm | 0 mm | -π/2 |
| 2 | 0 | 152 mm | 0 mm | 0 |
| 3 | $\theta_1$ | 60 mm | 57 mm | -π/2 |
| 4 | 0 | 39 mm | 0 mm | 0 |
| 5 | -π/2 | 0 mm | 0 mm | π/2 |
| 6 | $\theta_2$ | 57 mm | 33 mm | π/2 |
| 7 | -π/2 | 0 mm | 0 mm | -π/2 |
| 8 | $\theta_3$ | 187 mm | 30 mm | π/2 |
| 9 | 0 | 57 mm | 0 mm | -π |
| 10 | $\theta_4$ | 57 mm | 171 mm | π/2 |
| 11 | 0 | 30 mm | 0 mm | 0 |
| 12 | -π/2 | 0 mm | 0 mm | -π/2 |
| 13 | $\theta_5$ | 39 mm | 45 mm | π/2 |
| 14 | π/2 | 0 mm | 0 mm | -π/2 |
| 15 | $\theta_6$ | 45 mm | 45 mm | -π/2 |
| 16 | 0 | 45 mm | 0 mm | 0 |
| 17 | $\theta_7$ | -45 mm | 145 mm | -π/2 |

Table 4.1: Thormang3 left arm D-H parameters.

| | θ | *d* | *a* | α |
|---|---|---|---|---|
| 1 | 0 | 1053.5 mm | 0 mm | π/2 |
| 2 | 0 | 152 mm | 0 mm | 0 |
| 3 | $\theta_1$ | 60 mm | 57 mm | π/2 |
| 4 | 0 | 39 mm | 0 mm | 0 |
| 5 | π/2 | 0 mm | 0 mm | -π/2 |
| 6 | $\theta_2$ | 57 mm | 33 mm | -π/2 |
| 7 | π/2 | 0 mm | 0 mm | π/2 |
| 8 | $\theta_3$ | 187 mm | 30 mm | -π/2 |
| 9 | 0 | 57 mm | 0 mm | π |
| 10 | $\theta_4$ | 57 mm | 171 mm | -π/2 |
| 11 | 0 | 30 mm | 0 mm | 0 |
| 12 | π/2 | 0 mm | 0 mm | π/2 |
| 13 | $\theta_5$ | 39 mm | 45 mm | -π/2 |
| 14 | -π/2 | 0 mm | 0 mm | π/2 |
| 15 | $\theta_6$ | 45 mm | 45 mm | π/2 |
| 16 | 0 | 45 mm | 0 mm | 0 |
| 17 | $\theta_7$ | -45 mm | 145 mm | π/2 |

Table 4.2: Thormang3 right arm D-H parameters.

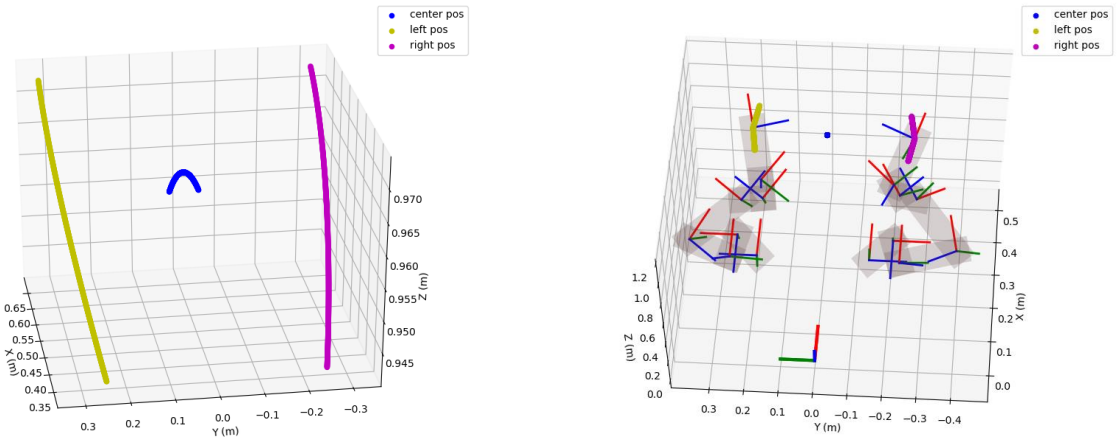



(a) (b)

Figure 4.2: Thormang3 kinematics module with (a) front view and (b) upper view by python matplotlib.

## 4.2 Experiment Result for Target position

After building the Thormang3 kinematic arm module, a target position for manipulating the Gogoro scooter is necessary to IK. Using the definition from Equation (3.6) to (3.10), the target of steering position for both arms can be computed and are Shown in Figure 4.3.




(a) (b)

Figure 4.3 : Target position for Thormang3 manipulate the Gogoro steering head bar.

## 4.3 Experiment Result for Jacobian Pseudo Inverse Kinematics

The results of the Jacobian pseudo IK are applied to the target position and convert to the steering angle. Starting the initial position with random degrees of -18 ~ 0 steering,and then steers the angle. Figure 4.4 shows the average iteration in 10 different random initial steering angle and Figure 4.5 shows the average efficient value in 10 different random initial steering angle for moving different steering angle. In Figure 4.4 and 4.5, the x axis is steering commands for steering degrees, y axis is the average iteration of computational. Thus, if the steering command is moving 5 degrees, the robot moves 5 degrees from the random initial steering angle, and the vertical line is in -1 to +1 standard deviation for 10 times different test.

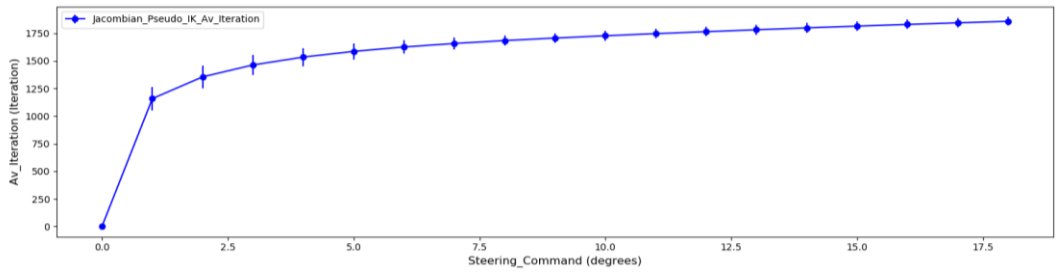



Figure 4.4 : The average iteration for Jacobian pseudo IK in 10 different initial steering angle.

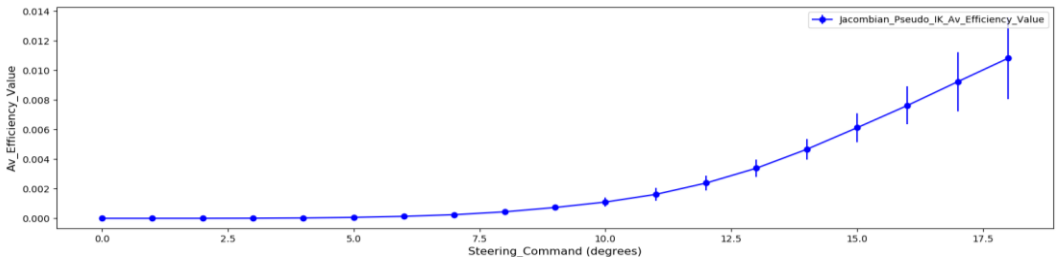



Figure 4.5 : The average value of efficiency for Jacobian pseudo IK in 10 different initial steering angle.

## 4.4 Experiment Result for PSO Inverse Kinematics

For PSO algorithm, the parameters used to the PSO IK are shown in Table 4.3 and evolve the results of different population size in Table 4.4. From the results, the proper population size is 20 due to increasing more population size is not getting lower iteration (generation) to find the solution for steering 15 degrees but using more time to calculation.

| | |
|---|---|
| Max Iteration | 8000 |
| Acceptance | 0.001 |
| W | 0.5 |
| C1 | 2 |
| C2 | 2 |

Table 4.3: Parameters for PSO IK.

| | PSO_IK |
|---|---|
| N_Pop = 10 | Iteration : 2892 |
| | Time : 22.0655 s |
| N_Pop = 20 | Iteration : 1835 |
| | Time : 27.7120 s |
| N_Pop = 30 | Iteration : 2061 |
| | Time : 45.9731 s |

Table 4.4: Result of different population size of PSO IK to steering 15 degrees.

With the parameters from Table 4.3 and the population size of 20, the average PSO IK generation result of steering to different steering angle and the average efficient value for 5 times in different random initial steering poses are shown in Figure 4.6 and Figure 4.7, and the vertical line is in -1 to +1 standard deviation for 5 times different test.

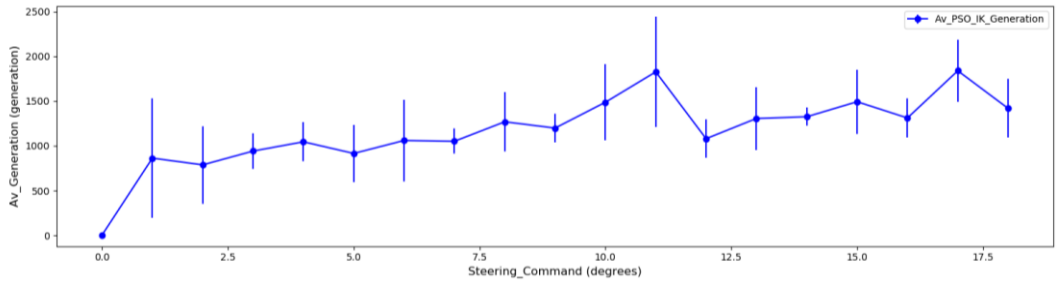



Figure 4.6 : The average generation time for PSO IK in 5 different steering angle.

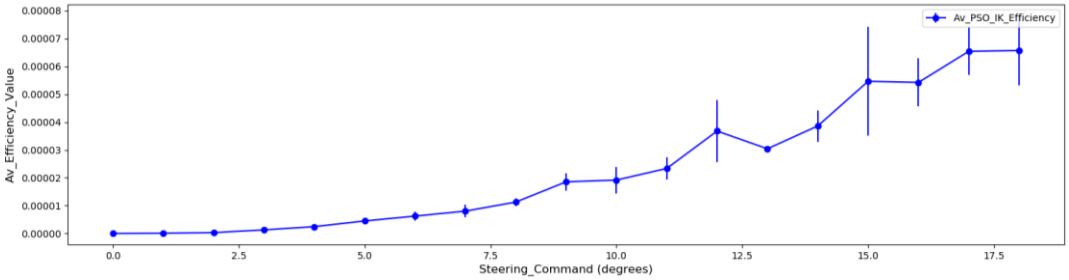



Figure 4.7 : The average value of efficiency for PSO IK in 5 different steering angle.

And the pose solution from PSO IK is very similar to Jacobian pseudo IK, the Figure 4.8 shows the both solution together and the most of the poses are overlapping to each other.

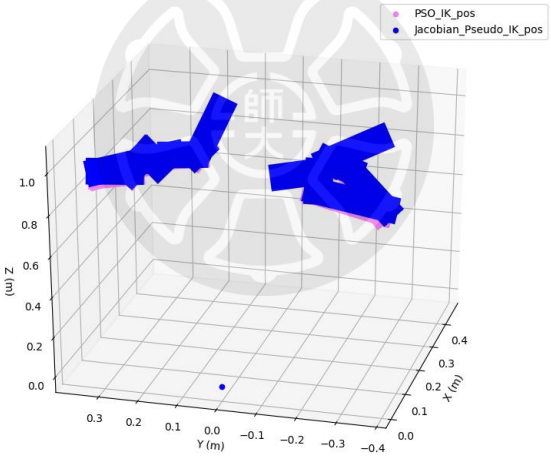



Figure 4.8 : The  PSO IK and Jacobian pseudo IK  pose in 3D graph.

## 4.5 Experiment Result for Hybrid Inverse Kinematics with Jacobian Pseudo and PSO

As mention in Section 3.10, the hybrid IK replaces PSO random initial speed to Jacobian pseudo $\Delta\theta$ in Equation (3.8) and replaces $pbest_i - x_i(t)$ to the random speed $N(0,0.001)$ for exploration, and applies the efficiency and  sensitivity to the fitness function. The parameters are shown in Table 4.5 for hybrid IK and the population size is 20.

| | |
|---|---|
| Max Iteration | 8000 |
| Acceptance | 0.001 |
| C2 | 2 |

Table 4.5: Parameters for hybrid IK.

And then, the average hybrid IK generation results of steering to different steering angle and the average efficient value are shown in Figure 4.9 and Figure 4.10, and the vertical line is in -1 to +1 standard deviation for 5 times different test.

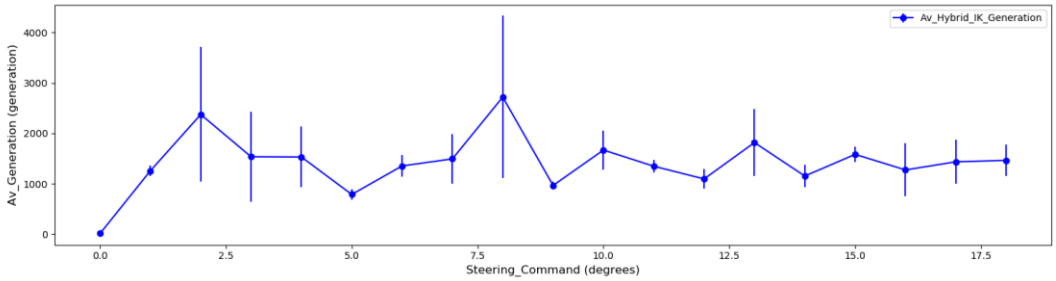



Figure 4.9 : The average generation time for hybrid IK in 5 different steering angle.

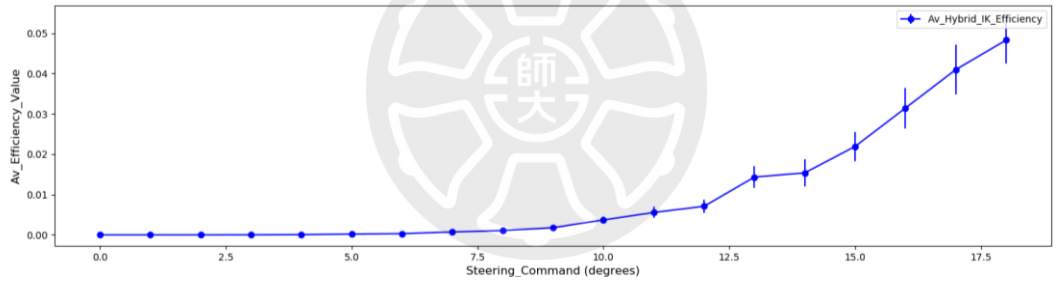



Figure 4.10 : The value of efficiency for hybrid IK in 5 different steering angle.

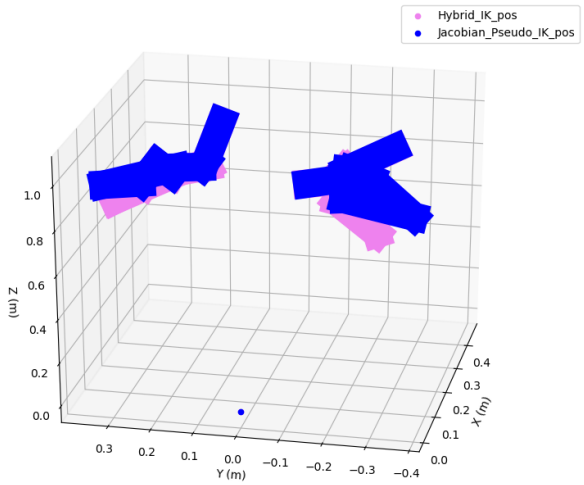



Figure 4.11 : The hybrid IK and Jacobian pseudo IK pose in 3D graph.

## 4.6 Experiment Result for Jacobian Pseudo Inverse Kinematics with Momentum

While the PSO IK and hybrid IK are using the population size but it computes the fitness in every iteration with forward kinematic which cause the time cost in computing increasing a lot. Thus, we try to apply a concept of the momentum from PSO behavior to Jacobian pseudo IK. And the Jacobian pseudo IK average iteration results in 10 different initial steering angle and the average efficient value in 10 different initial steering angle are shown in Figure 4.12 and Figure 4.13, and the vertical line is in -1 to +1 standard deviation for 10 times different test. Figure 4.14 shows the 3D graph to compare Jacobian pseudo IK with momentum with normal Jacobian pseudo IK, the solution of steering pose has same result.

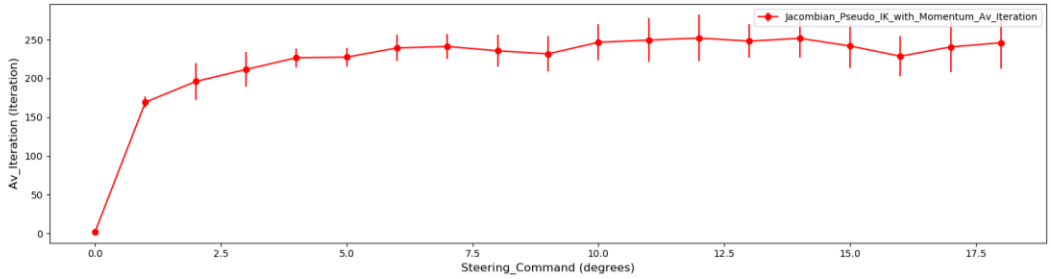



Figure 4.12 : The average iteration for Jacobian pseudo IK with momentum in 10 different initial steering angle.

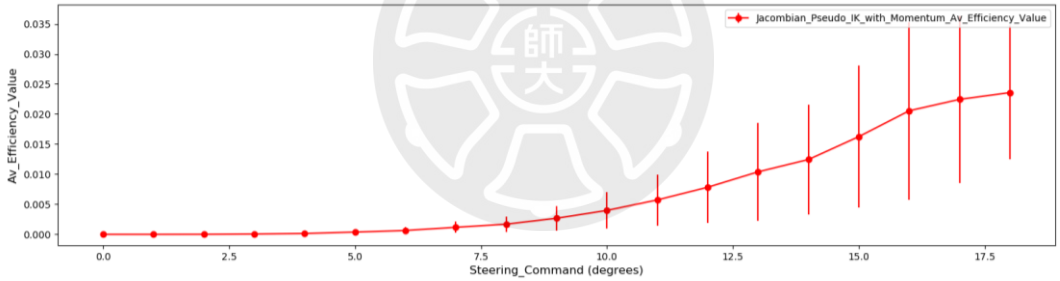



Figure 4.13 : The average value of efficiency for Jacobian pseudo IK with momentum in 10 different initial steering angle.

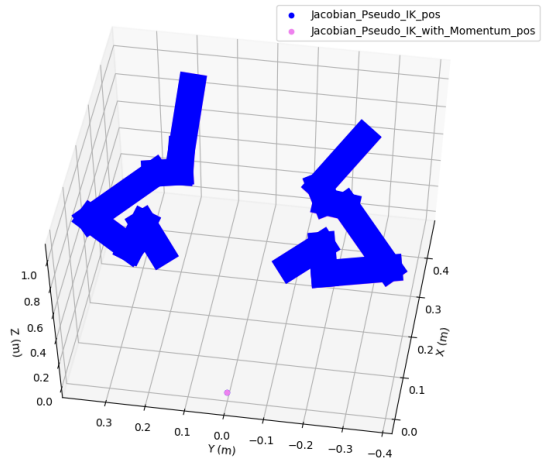



Figure 4.14 : The Jacobian pseudo IK and Jacobian pseudo IK with momentum pose in 3D graph.

## 4.7 Comparing Jacobian Pseudo IK and Jacobian Pseudo IK with Momentum

This section shows the difference between both method when applied momentum or not. Figure 4.15 combines both method of average iteration results together, and we can observe that the Jacobian pseudo IK applied the momentum, red line in Figure 4.15, is much faster than original method and average iteration is always lower than 300.

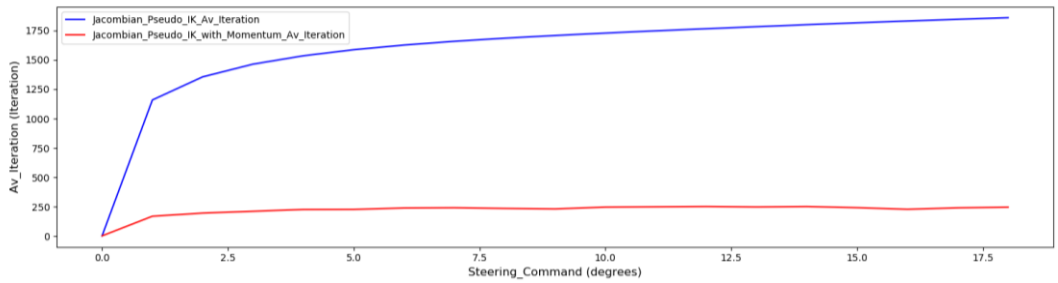



Figure 4.15 : Comparing the results of generation from Jacobian pseudo IK and Jacobian pseudo IK with Momentum.

Moreover, Figure 4.16 combines both method of average efficient results, and the Jacobian pseudo IK applied the momentum is more efficiency than the original Jacobian pseudo IK.

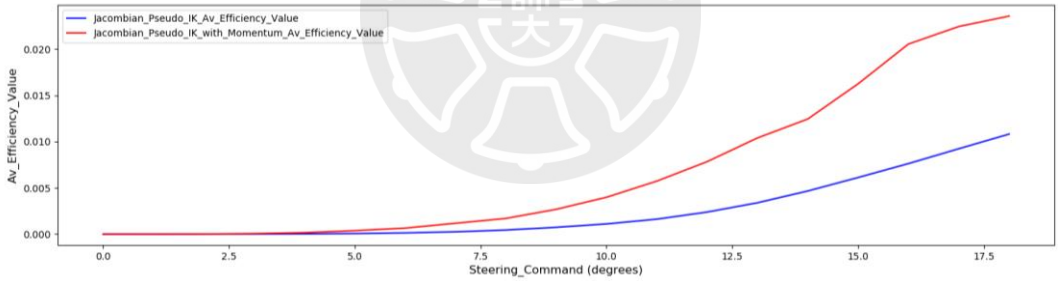
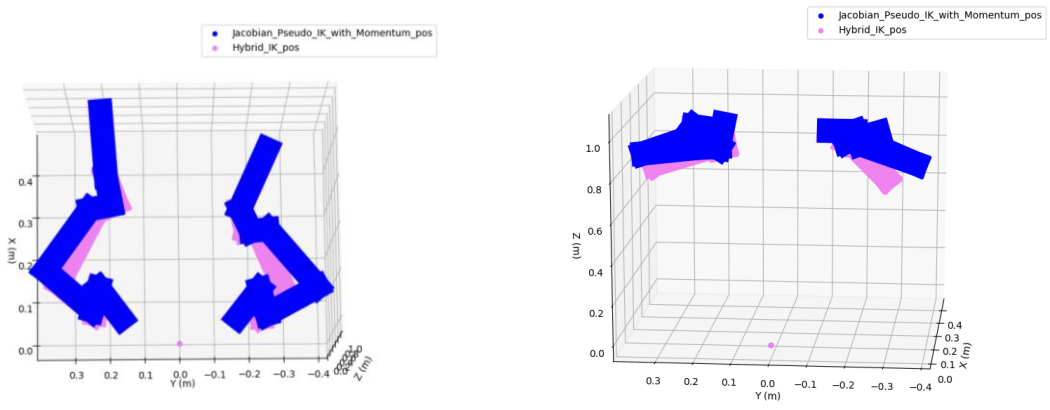



Figure 4.16 : Comparing the results of efficiency from Jacobian pseudo IK and Jacobian pseudo IK with Momentum.

## 4.8 Evaluating Jacobian Pseudo IK with momentum to Optimal Solution from Hybrid IK

However, the result of average efficiency from hybrid IK in Figure 4.10, the average efficiency value goes to 0.04, which is better than Jacobian pseudo IK with Momentum and the solution of Jacobian pseudo IK with Momentum can be optimized. From the hybrid IK solution in Table 4.6 and 4.7 and the Jacobian pseudo IK solution in Table 4.8 and 4.9 , we observers a phenomenon that the end effector from elbow link is closer to the robot torso, in other words, the joint value from raw direction of shoulder is lower, and the Figure 4.17 shows both poses result in 3D plot.




(a) (b)

Figure 4.17 : Comparing Hybrid IK pose and Jacobian pseudo IK with momentum pose.

Therefore, we limit the joint value from raw direction of shoulder into 70 ~ 90 degrees, Figure 4.18 shows the average iteration is about 300 iteration, and Figure 4.19 shows the result of average efficiency by using Jacobian pseudo IK with Momentum and the limitation, and the highest average efficiency reaches to 0.04.


Figure 4.18 : The average iteration for Jacobian pseudo IK with momentum and limitation in different steering angle.

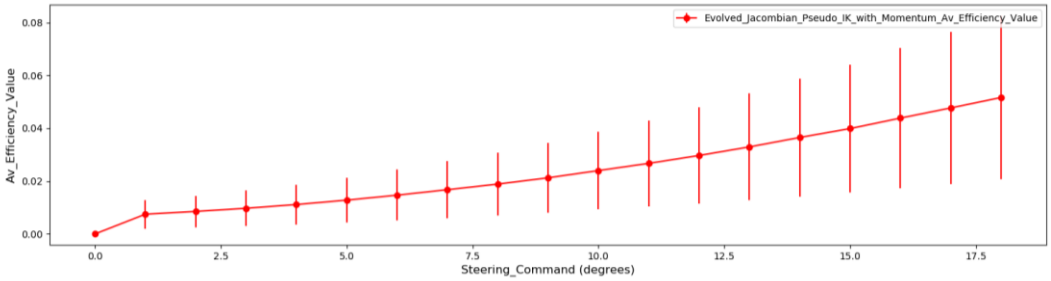



Figure 4.19 : The average value of efficiency for Jacobian pseudo IK with momentum and limitation in different steering angle.

The unit in Table 4.6 to 4.9 show different solution to Hybrid IK results and Jacobian pseudo IK with momentum results for both robot arms from 0~18 degrees steering poses and joints value (in degrees).

| | l_sh_p1 | l_sh_r | l_sh_p2 | l_el_y | l_wr_r | l_wr_y | l_wr_p |
|---|---|---|---|---|---|---|---|
| 0 deg | -4.9591 | 75.0369 | 17.3148 | 17.1884 | 166.5985 | 12.8096 | -35.7219 |
| 1 deg | -5.7619 | 74.2756 | 17.5296 | 16.5587 | 165.5319 | 12.6039 | -34.7842 |
| 2 deg | -6.2814 | 74.1330 | 17.0797 | 15.9324 | 165.0822 | 12.4751 | -33.0006 |
| 3 deg | -7.3620 | 74.0536 | 17.0717 | 14.7565 | 164.6339 | 12.2482 | -31.5310 |
| 4 deg | -8.0181 | 73.3795 | 17.1492 | 14.0090 | 163.6786 | 11.8261 | -30.2803 |
| 5 deg | -8.5281 | 73.8361 | 16.4397 | 13.3520 | 163.8458 | 11.9956 | -28.2130 |
| 6 deg | -9.3710 | 73.2584 | 16.5409 | 12.2276 | 163.0147 | 11.4141 | -26.8414 |
| 7 deg | -10.5188 | 71.9706 | 17.4415 | 11.3351 | 161.3061 | 10.8966 | -26.3775 |
| 8 deg | -11.0096 | 72.4910 | 16.7678 | 10.3805 | 161.6341 | 10.8183 | -24.1862 |
| 9 deg | -11.9952 | 71.7001 | 17.2285 | 9.3165 | 160.5067 | 10.2702 | -23.1251 |
| 10 deg | -13.0952 | 71.5476 | 17.3280 | 7.9880 | 159.9677 | 9.9597 | -21.6312 |
| 11 deg | -13.8534 | 71.3082 | 17.4526 | 6.8850 | 159.4494 | 9.4488 | -20.1520 |
| 12 deg | -14.9889 | 70.5092 | 18.0422 | 5.6985 | 158.2469 | 8.9363 | -19.1453 |
| 13 deg | -15.6909 | 70.6151 | 17.6827 | 4.5171 | 158.1303 | 8.6401 | -17.1524 |
| 14 deg | -16.5480 | 70.7223 | 17.5695 | 3.2131 | 157.8836 | 8.2439 | -15.3874 |
| 15 deg | -17.5700 | 70.3111 | 17.7459 | 1.8685 | 157.1614 | 7.7800 | -13.7979 |
| 16 deg | -18.2445 | 70.7056 | 17.1856 | 0.5836 | 157.3380 | 7.5476 | -11.6076 |
| 17 deg | -18.9958 | 71.4111 | 16.5995 | -0.6774 | 157.8075 | 7.4823 | -9.4730 |
| 18 deg | -20.2184 | 70.6195 | 17.4568 | -2.3010 | 156.5997 | 6.6738 | -8.3489 |

Table 4.6 :Left arm joint value from Hybrid_IK solution.

| | r_sh_p1 | r_sh_r | r_sh_p2 | r_el_y | r_wr_r | r_wr_y | r_wr_p |
|---|---|---|---|---|---|---|---|
| 0 deg | 9.4393 | -74.9336 | -0.9781 | -33.9568 | -125.7544 | -50.7314 | 54.5485 |
| 1 deg | 8.8112 | -75.9336 | -1.4998 | -34.6415 | -126.3622 | -49.9309 | 56.2223 |
| 2 deg | 8.0284 | -76.8741 | -1.5765 | -35.6525 | -126.8260 | -49.4020 | 57.6850 |
| 3 deg | 7.2110 | -77.5150 | -2.1975 | -36.3687 | -127.3273 | -48.1734 | 59.2765 |
| 4 deg | 6.4830 | -78.2130 | -2.5352 | -37.3893 | -127.4624 | -47.4532 | 61.1325 |
| 5 deg | 5.5811 | -79.9844 | -2.5562 | -38.1519 | -129.3970 | -47.1958 | 61.5358 |
| 6 deg | 4.8497 | -80.7031 | -3.1465 | -38.8878 | -130.1405 | -46.1657 | 62.9055 |
| 7 deg | 4.0353 | -81.3877 | -3.8125 | -39.3863 | -131.0594 | -44.9137 | 64.1009 |
| 8 deg | 3.2260 | -82.4280 | -4.1914 | -40.1742 | -132.1538 | -44.1512 | 65.1238 |
| 9 deg | 2.4803 | -82.4259 | -5.2532 | -40.6686 | -132.4397 | -42.3435 | 66.8917 |
| 10 deg | 1.8688 | -82.2861 | -6.1040 | -41.1115 | -132.5624 | -40.6553 | 68.5677 |
| 11 deg | 1.1484 | -84.0339 | -6.5614 | -41.6298 | -134.7398 | -40.1749 | 68.9566 |
| 12 deg | 0.4542 | -84.9037 | -7.2430 | -41.8547 | -136.1901 | -38.9526 | 69.6434 |
| 13 deg | -0.0171 | -83.380 | -8.8405 | -42.2734 | -135.1650 | -36.2068 | 72.5327 |
| 14 deg | -0.6808 | -84.1517 | -9.5109 | -42.4853 | -136.5362 | -34.9876 | 73.2774 |
| 15 deg | -1.4750 | -84.7510 | -10.3312 | -42.9413 | -137.8050 | -33.6669 | 74.2629 |
| 16 deg | -2.1590 | -85.7046 | -11.1677 | -42.9558 | -139.6726 | -32.2902 | 74.7879 |
| 17 deg | -2.9055 | -86.2482 | -11.6776 | -43.2211 | -140.8367 | -31.0344 | 75.5011 |
| 18 deg | -3.3982 | -86.4329 | -12.7357 | -43.3674 | -141.8128 | -29.4147 | 76.8187 |

Table 4.7 :Right arm joint value from Hybrid_IK solution.

| | l_sh_p1 | l_sh_r | l_sh_p2 | l_el_y | l_wr_r | l_wr_y | l_wr_p |
|---|---|---|---|---|---|---|---|
| 0 deg | -11.1546 | 59.8779 | 30.4015 | 17.0224 | 151.8043 | 6.5728 | -49.4664 |
| 1 deg | -11.8403 | 59.5896 | 30.4635 | 16.5414 | 151.1692 | 6.4191 | -48.0837 |
| 2 deg | -12.5804 | 59.2460 | 30.5220 | 15.7653 | 150.4411 | 6.0561 | -46.5277 |
| 3 deg | -13.3256 | 58.9393 | 30.6370 | 14.9272 | 149.7561 | 5.6443 | -44.9766 |
| 4 deg | -14.0891 | 58.6588 | 30.7838 | 14.0485 | 149.0889 | 5.2207 | -43.4185 |
| 5 deg | -14.8711 | 58.3980 | 30.9545 | 13.1332 | 148.4313 | 4.7899 | -41.8476 |
| 6 deg | -15.6716 | 58.1555 | 31.1479 | 12.1788 | 147.7815 | 4.3494 | -40.2609 |
| 7 deg | -16.4899 | 57.9307 | 31.3630 | 11.1864 | 147.1382 | 3.9001 | -38.6576 |
| 8 deg | -17.3260 | 57.7235 | 31.5996 | 10.1554 | 146.5011 | 3.4416 | -37.0372 |
| 9 deg | -18.1806 | 57.5355 | 31.8593 | 9.0824 | 145.8705 | 2.9719 | -35.3985 |
| 10 deg | -19.0528 | 57.3662 | 32.1407 | 7.9698 | 145.2456 | 2.4930 | -33.7418 |
| 11 deg | -19.9439 | 57.2176 | 32.4458 | 6.8131 | 144.6267 | 2.0025 | -32.0662 |
| 12 deg | -20.8533 | 57.0896 | 32.7734 | 5.6144 | 144.0131 | 1.5020 | -30.3716 |
| 13 deg | -21.7808 | 56.9829 | 33.1238 | 4.3730 | 143.4045 | 0.9918 | -28.6581 |
| 14 deg | -22.7280 | 56.8999 | 33.4988 | 3.0838 | 142.8010 | 0.4692 | -26.9243 |
| 15 deg | -23.6943 | 56.8405 | 33.8975 | 1.7481 | 142.2017 | -0.0643 | -25.1703 |
| 16 deg | -24.6799 | 56.8059 | 34.3201 | 0.3647 | 141.6062 | -0.6092 | -23.3959 |
| 17 deg | -25.6858 | 56.7977 | 34.7675 | -1.0698 | 141.0139 | -1.1669 | -21.6000 |
| 18 deg | -26.7120 | 56.8172 | 35.2400 | -2.5571 | 140.4241 | -1.7375 | -19.7822 |

Table 4.8 : Left arm joint value from Jacobian pseudo IK with momentum solution.

| | r_sh_p1 | r_sh_r | r_sh_p2 | r_el_y | r_wr_r | r_wr_y | r_wr_p |
|---|---|---|---|---|---|---|---|
| 0 deg | 13.0472 | -51.069 | -16.2511 | -32.4362 | -105.3284 | -27.9216 | 65.9101 |
| 1 deg | 12.6304 | -51.8249 | -16.4182 | -33.0243 | -105.8751 | -27.1406 | 67.6095 |
| 2 deg | 11.9903 | -52.5501 | -16.6320 | -33.8778 | -106.4939 | -26.2639 | 69.4334 |
| 3 deg | 11.3506 | -53.2519 | -16.8721 | -34.6952 | -107.1842 | -25.3633 | 71.1825 |
| 4 deg | 10.7277 | -53.9467 | -17.1558 | -35.4738 | -107.9503 | -24.4438 | 72.8769 |
| 5 deg | 10.1265 | -54.6363 | -17.4851 | -36.2117 | -108.7891 | -23.5083 | 74.5233 |
| 6 deg | 9.5467 | -55.3195 | -17.8592 | -36.9095 | -109.6969 | -22.5574 | 76.1257 |
| 7 deg | 8.9876 | -55.9946 | -18.2768 | -37.5674 | -110.6703 | -21.5914 | 77.6869 |
| 8 deg | 8.4513 | -56.6609 | -18.7357 | -38.1832 | -111.7047 | -20.6123 | 79.2096 |
| 9 deg | 7.9343 | -57.3161 | -19.2339 | -38.7590 | -112.7977 | -19.6189 | 80.6962 |
| 10 deg | 7.4407 | -57.9605 | -19.7688 | -39.2912 | -113.9435 | -18.6142 | 82.1491 |
| 11 deg | 6.9661 | -58.5921 | -20.3382 | -39.7824 | -115.1406 | -17.5966 | 83.5707 |
| 12 deg | 6.5125 | -59.2107 | -20.9392 | -40.2309 | -116.3841 | -16.5681 | 84.9632 |
| 13 deg | 6.0783 | -59.8156 | -21.5694 | -40.6371 | -117.6713 | -15.5286 | 86.3290 |
| 14 deg | 5.6654 | -60.4070 | -22.2252 | -41.0001 | -118.9975 | -14.4805 | 87.6706 |
| 15 deg | 5.2705 | -60.9843 | -22.9049 | -41.3213 | -120.3616 | -13.4224 | 88.9896 |
| 16 deg | 4.8966 | -61.5482 | -23.6040 | -41.5994 | -121.7575 | -12.3579 | 90.2889 |
| 17 deg | 4.5436 | -62.0987 | -24.3190 | -41.8350 | -123.1817 | -11.2885 | 91.5706 |
| 18 deg | 4.2081 | -62.6360 | -25.0486 | -42.0292 | -124.6341 | -10.2129 | 92.8362 |

Table 4.9 :Right arm joint value from Jacobian pseudo IK with momentum solution.

## 4.9 Comparing Sensitivity to Optimal Solution and Original Solution

In the original solution from section 4.6, the Figure 4.20 shows the sensitive value for original solution, and the Figure 4.21 shows the sensitive value for optimal solution from section 4.8. Although the original solution is not efficiency as the results from section 4.8, it is more resistance to the noise. With the hundred times of testing, the average sensitive value for original solution is 7.677 e-05 in Figure 4.20, and the average sensitive value for optimal solution is 9.669 e-05 in Figure 4.21.

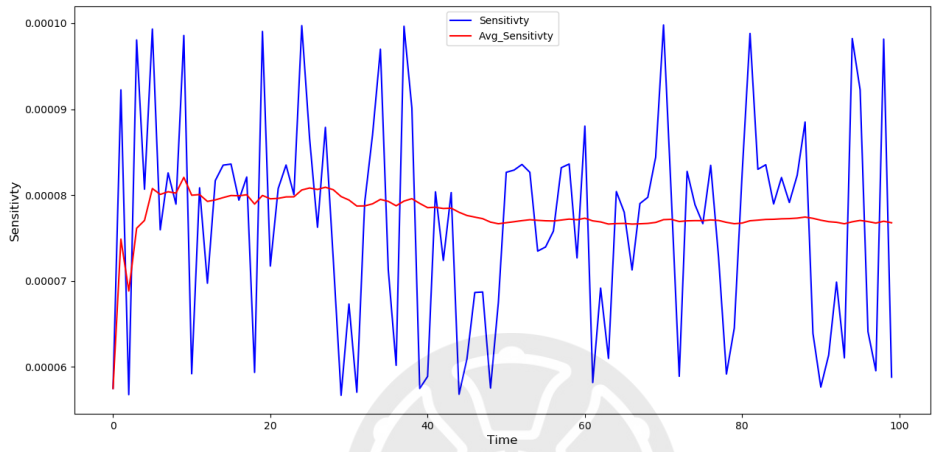



Figure 4.20 : The sensitive value and average value for original solution.

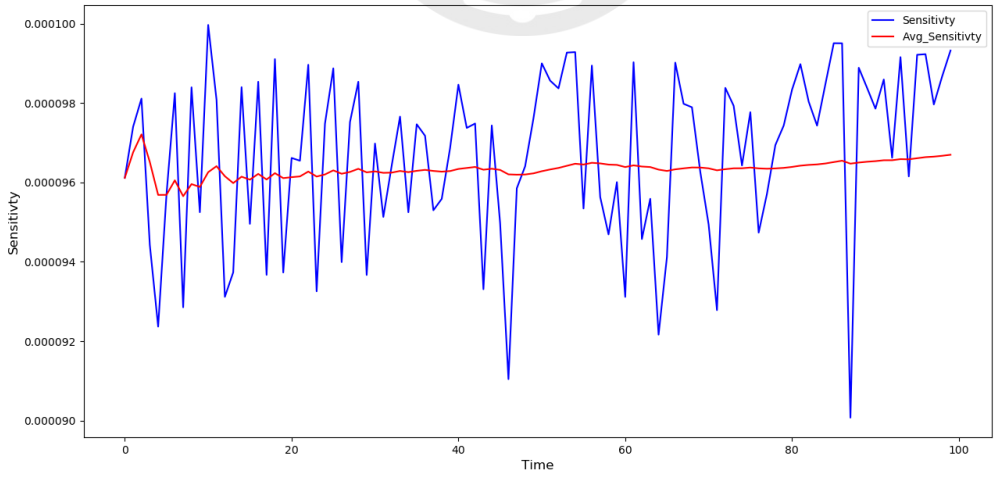



Figure 4.21 : The sensitive value and average value for optimal solution.

## 4.10 Experiment Results in Real Robot-Scooter System

In this section, the fast IK in C++ version and PID controller are described and discussed, and comparing result of PID in simulation with reality result. The last part discusses the actuator restriction which should be satisfied in real robot-scoot system.

### 4.10.1 Experiment Results for Jacobian Pseudo Ik with Momentum in C++

Second, the methodology apply in section 3.10 with the proposed kinematics module from Jacobian pseudo IK with momentum. In the experiment, the proposed method is 2 to 10 times faster than original Jacobian pseudo-inverse method. Within the computer specification in Table 4.10, a C++ version of real time computation are shows in Figure 4.22, the x axis is distance in cm with different cases and y axis is real time cost in ms from different distance x cm.

| | |
|---|---|
| CPU (Processor) | AMD Ryzen 9 3900X (24) @ 3.800GHz |
| GPU (Graphics) | NVIDIA GeForce RTX 2080 SUPER (8GB) |
| RAM (Memory) | 16GB |

Table 4.10: Relevant specification of the computer used to compute IK.

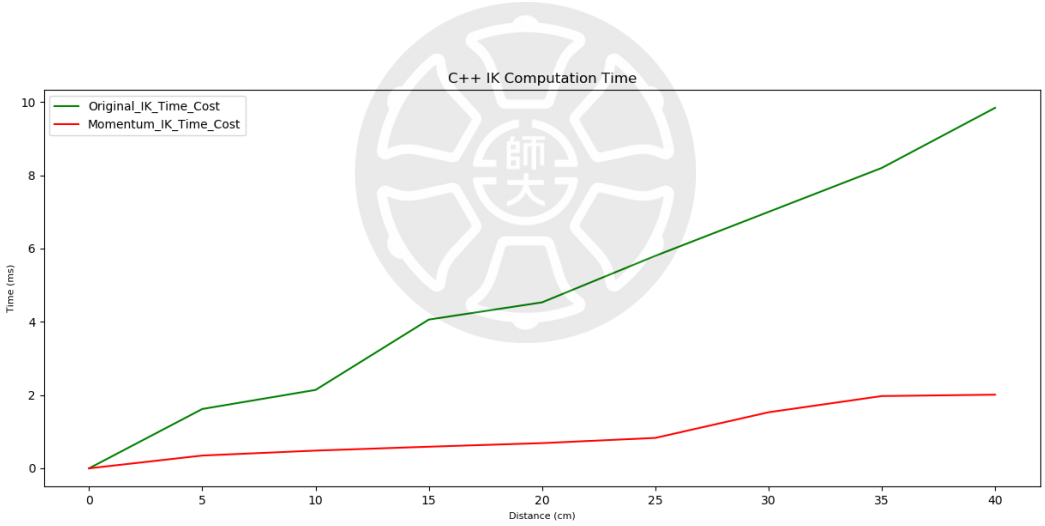



Figure 4.22 : C++ version of Evaluating real time cost from Jacobian pseudo IK with momentum and original Jacobian pseudo-inverse method.

### 4.10.2 PID Balance Controller in Simulation Results

In the Gazebo simulation system, we had the simulation results from our last project leader [34]. There has three different PID controller parameters for low speed (10 rad/s), middle speed (20 rad/s) and high speed (40 rad/s), where the parameters are shown in Table 4.11.

| System Velocity | P | I | D |
|---|---|---|---|
| 10 rad/s (2.5km/h) | 8.0 | 0.04 | 4.0 |
| 20 rad/s ( 5km/h ) | 2.0 | 0.04 | 4.0 |
| 40 rad/s (10km/h) | 1.0 | 0.04 | 2.0 |

Table 4.11: PID parameters for balance control in Gazebo simulation system in [34].

The results of low speed (10 rad/s ) PID balance control shows in Figure 4.23 and 4.24. Where P gain is 8, I gain is 0.04 and D gain is 4. The x axis is the iteration time, each interval time is 0.1 second and y axis is radian unit.

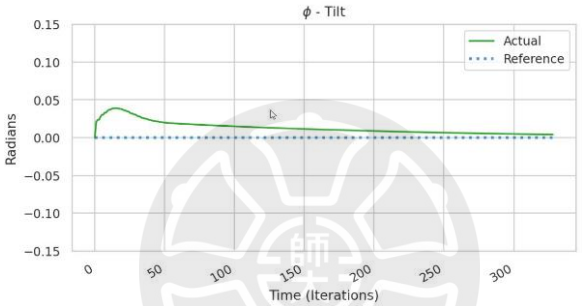



Figure 4.23: Actual tilt during low speed PID balance control.

(a)

(b)

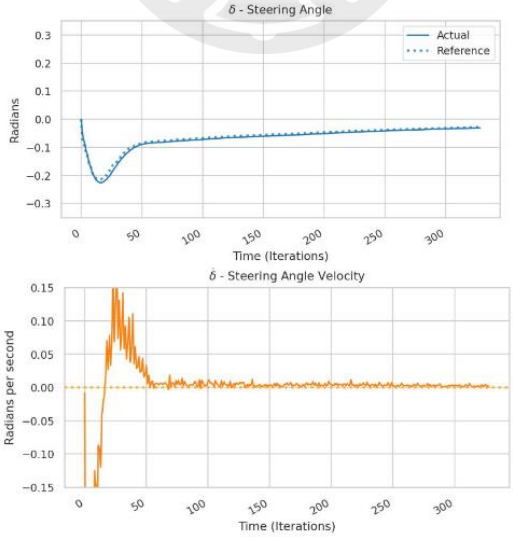



Figure 4.24: (a)Steering commands during low speed PID control and (b) low speed steering velocity.

And the results of middle speed (20rad/s) PID control shows in Figure 4.25 and 4.26. Where P gain is 2, I gain is 0.04 and D gain is 4. The x axis is the iteration time, each interval time is 0.1 second and y axis is radian unit.

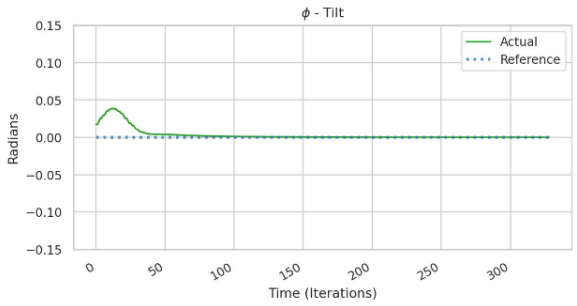



Figure 4.25: Actual tilt during middle speed PID balance control.

(a)

(b)

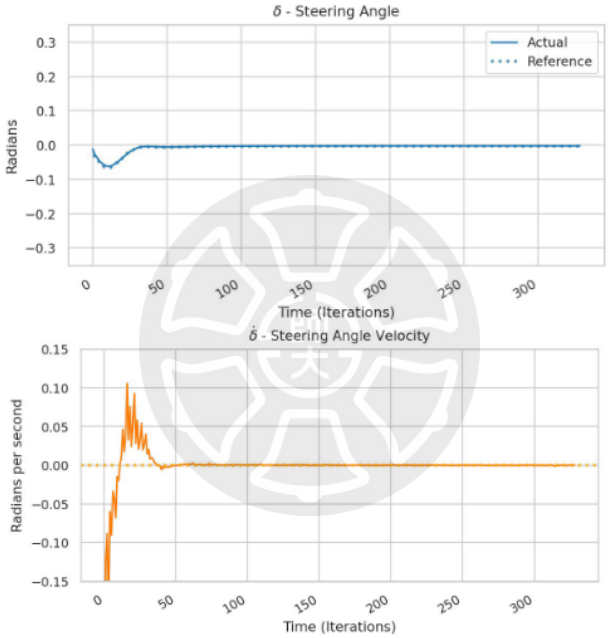



Figure 4.26 : (a)Steering commands during middle speed PID control and (b) middle speed steering velocity.

The last results of high speed (40rad/s) PID control shows in Figure 4.27 and 4.28. Where P gain is 1, I gain is 0.04 and D gain is 2. The x axis is the iteration time, each interval time is 0.1 second and y axis is radian unit.

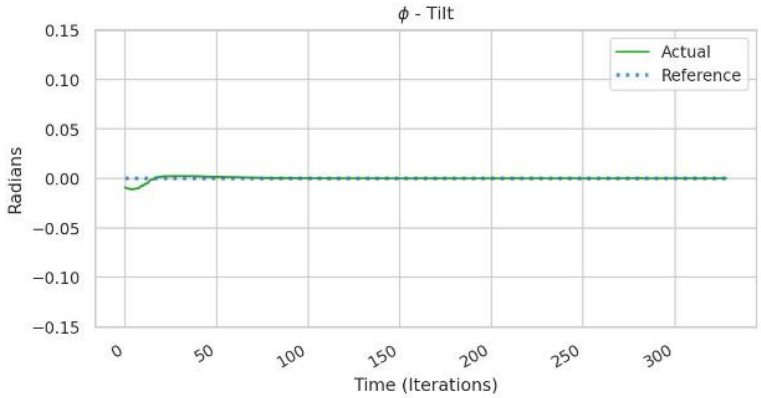



Figure 4.27 : Actual tilt during high speed PID balance control.

(a)

(b)

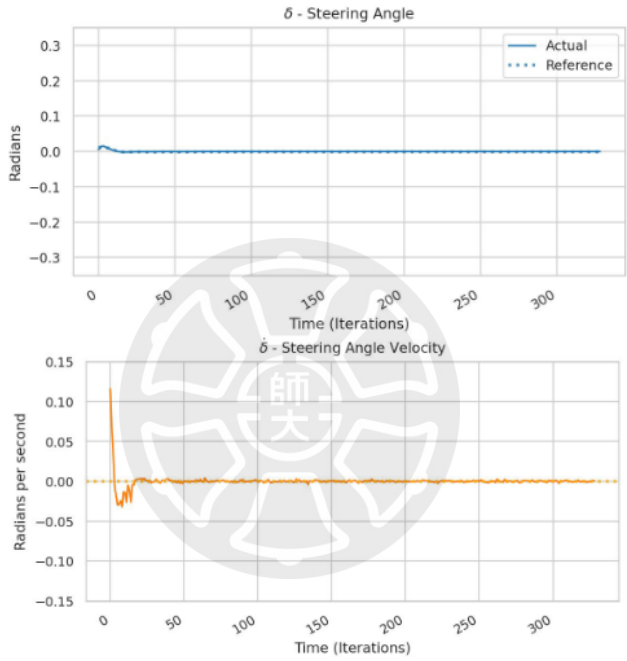



Figure 4.28 : (a)Steering commands during high speed PID control and (b) middle speed steering velocity.

### 4.10.3 PID Balance Controller in Real Robot Results

In the real Thormang3-Gogoro system, however, the speed is not a constant speed in simulation for real testing. The upper bound of speed is about 12 km/hr and the first three seconds will gas from 0 km/hr to the upper limitation speed. Then the speed decreased after three seconds gassed. The results show in Figure 4. 29 and 4.30. Where P gain is 1.2, I gain is 0.04 and D gain is 2. The x axis is the time iteration with 0.1 seconds and the y axis is radian unit.

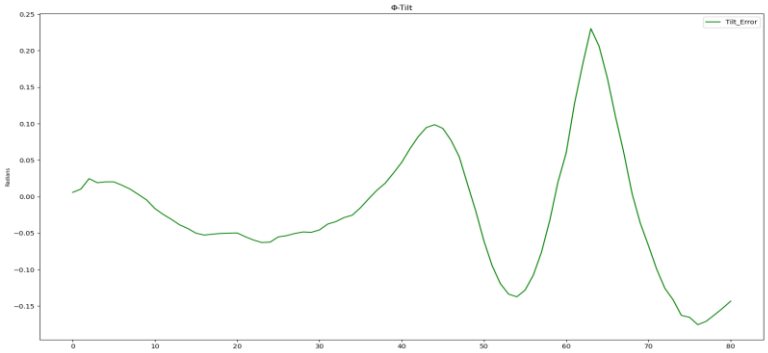



Figure 4.29: Actual tilt during real Thormang3-Gogoro system PID balance control.

(a)

(b)

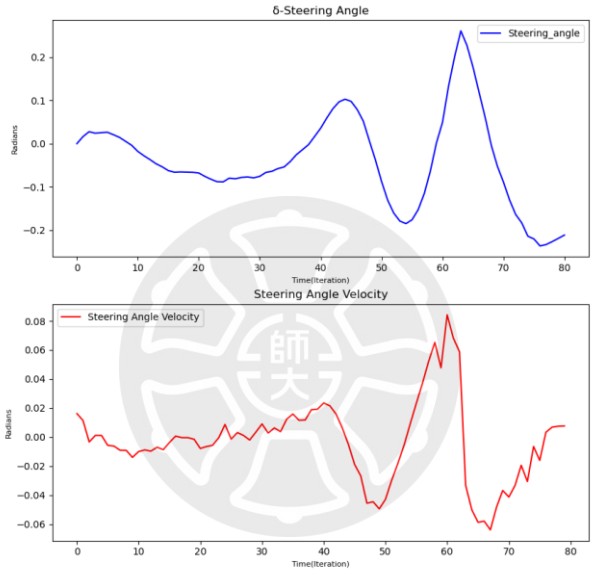



Figure 4.30: (a)Steering commands during real Thormang3-Gogoro system PID control and (b) real Thormang3-Gogoro system steering velocity.

The results of balance control diverged caused by decreasing speed after three seconds gassed and PID controller parameters are very sensitive to the speed that the Thormang3-Gogoro system is easily fall down to one side when the speed is out of the expected value.

The link of real robot-scoot test video :

https://drive.google.com/file/d/1TsbdT0Cgv_L7hKs6Sr95EFoK6S_Wk7aY/view?usp=sharing

### 4.10.4 Actuator's Restrictions on Balance Control

In Figure 4.31 shows Thormang3 actuator of PH54-100-S500-R[4] to its arm servo. The maximum speed for the actuator is 29.2 rev/min. And the IMU sensor in thormang3 system transmits the data in 250 Hz. Therefore, the constrain from the actuator is 3.057817 rad/s (0.003 rad /ms).

From the result of Section 4.10.2 and 4.10.3, we can observe the maximum velocity in simulation is 0.15 rad/ 100 ms,(0.0015 rad/ms) in Figure 4.24 and the maximum velocity in real test is 0.08 rad/ 100 ms (0.0008 rad/ms) in Figure 4.30.

Figure 4.31: The actuator PH54-100-S500-R used in Thormang3 arms.

The Figure 4.32 presents the maximum changing in joint value in y axis and difference delta steering command in x axis. We consider the maximum velocity 0.15 rad/100 ms (0.0015 rad/ms) in simulation which is lager than real test, and the 0.0015 rad of steering command will need 0.00247 rad/ms joint revolution which is totally under the restrictions 0.003 rad /ms.

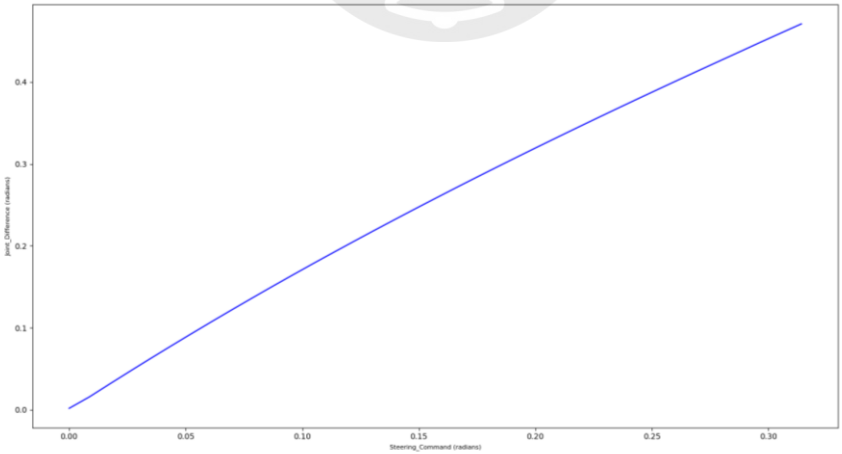



Figure 4.32 : The maximum changing of joint value in different delta steering angle.

[4] https://emanual.robotis.com/docs/en/dxl/pro/h54-100-s500-r/

# Chapter 5. Conclusion and Future Work

## 5.1 Conclusion

The main objective of this thesis is to propose a fast IK algorithm that can reduce computational cost while maintaining acceptable performance and high efficiency of motion planning. Therefore, there are four different IK algorithms to implement on Thormang3 for the motion of manipulating the Gogoro E-scooter steering.

First, the Jacobian pseudo IK is used a newton method to approximate the error to linear. This method is the most general proposed method for IK. Second, the PSO IK method randomly search a IK problem and converge to an acceptable error. Third, the hybrid IK using Jacobian pseudo IK and PSO is moving more direct to the target position and applies the efficiency and sensitivity value for exploring the optimal solution to efficiency motion and operating the two-wheeled electric scooter steering. Fourth, the Jacobian pseudo IK with momentum greatly reduces the iteration and the computational time from original Jacobian pseudo IK. Although Jacobian pseudo IK with momentum has same result of ending pose but the moving process is more efficiency. Then we combine the optimal solution from the hybrid IK and the Jacobian pseudo IK with momentum, a high efficiency motion and fast IK is successful developed. In the results, we have 0.04 value of average efficiency and a fast IK that lower than 3 ms computational time, and it is also satisfied to the robot servo restriction.

## 5.2 Future Work

In the Future, we believe the human robot Thormang3 can pass the scooter license test , and the test includes the balancing test, U turn and  two-stage left turn. The next step to achieve the goal is to develop a speed controller that can obtain the current scooter speed by robot vision and let Thormang3 control the two-wheeled electric scooter speed during the driving. This will be a exciting research that combining RL agent from the previous work [34], which required the speed value from speed detection, and Jacobian pseudo IK with momentum to do the balancing control and path trajectory for driving in the reality.

# References


[1] Kajita, Shuuji, et al. Introduction to humanoid robotics. Vol. 101. Springer Berlin Heidelberg, 2014.

[2] Alom, Md Zahangir, et al. "The history began from alexnet: A comprehensive survey on deep learning approaches." arXiv preprint arXiv*:1803.01164* (2018).

[3] Hirai, Kazuo, et al. "The development of Honda humanoid robot." Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146). Vol. 2. IEEE, 1998.

[4] Vukobratović, Miomir, Branislav Borovac, and Kalman Babković. "Contribution to the study of anthropomorphism of humanoid robots." International Journal of Humanoid Robotics 2.03 (2005): 361-387.

[5] Singhania, S.; Kageyama, I.; Karanam, V.M. Study on Low-Speed Stability of a Motorcycle. Appl. Sci. **2019**, *9*, 2278.

[6] Garziad, M., Saka, A. (2019). Influence of rider on the stability and control of two wheeled vehicles. Journal Européen des Systèmes Automatisés, Vol. 52, No. 5, pp. 515-520.

[7] Ngoc Kien Vu, Hong Quang Nguyen, "Balancing Control of Two-Wheel Bicycle Problems", Mathematical Problems in Engineering, vol. 2020, Article ID 6724382, 12 pages, 2020.

[8] Y. Tanaka and T. Murakami, "Self sustaining bicycle robot with steering controller," in Proceedings of International Workshop on Advanced Motion Control, pp. 193–197, Kawasaki, Japan, March 2004.

[9] C.-K. Chen and T.-K. Dao, "Speed-adaptive roll-angle-tracking control of an unmanned bicycle using fuzzy logic," Vehicle System Dynamics, vol. 48, no. 1, pp. 133–147, 2010.

[10] C. Huang, Y. Tung, and T. Yeh, "Balancing control of a robot bicycle with uncertain center of gravity," in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 5858–5863, Singapore, May 2017.

[11] S. Vatanashevanopakorn and M. Parnichkun, "Steering control based balancing of a bicycle robot," in Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics, pp. 2169–2174, Karon Beach, Phuket, Thailand, December 2011.

[12] S. Lee and W. Ham, "Self-stabilizing strategy in tracking control of unmanned electric bicycle with mass balance," in Proceedings of the IEEE International Conference on Intelligent Robots and Systems, pp. 2200–2205, Lausanne, Switzerland, October 2002.

[13] L. Keo, K. Yoshino, M. Kawaguchi, and M. Yamakita, "Ex-perimental results for stabilizing of a bicycle with a flywheel balancer," in Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, May 2011.

[14] M. Yamakita, A. Utano, and K. Sekiguchi, "Experimental study of automatic control of bicycle with balancer," in Proceedings of the International Conference of Intelligent Robots and Systems, pp. 5606–5611, Beijing, China, October 2006.

[15] Sikander, A., Prasad, R. Reduced order modelling based control of two wheeled mobile robot. J Intell Manuf **30,** 1057–1067 (2019).

[16] D. Karnopp, "Tilt control for gyro-stabilized two-wheeled vehicles," Vehicle System Dynamics, vol. 37, no. 2, pp. 145–156, 2002.

[17] Rodriguez-Rosa, David & Payo-Gutierrez, Ismael & Castillo Garcia, Fernando Jose & Gonzalez-Rodriguez, Antonio & Perez-Juarez, Sergio. (2017). Improving Energy Efficiency of an Autonomous Bicycle with Adaptive Controller Design. Sustainability. 9. 866. 10.3390/su9050866.

[18] Y. Tanaka and T. Murakami, "Self sustaining bicycle robot with steering controller," The 8th IEEE International Workshop on Advanced Motion Control, 2004. AMC *'04.*, 2004, pp. 193-197.

[19] K. J. Astrom, R. E. Klein and A. Lennartsson, "Bicycle dynamics and control: adapted bicycles for education and research," in IEEE Control Systems Magazine, vol. 25, no. 4, pp. 26-47, Aug. 2005.

[20] Zhang, Dianmu and B. Hannaford. “IKBT: Solving Symbolic Inverse Kinematics with Behavior Tree.” J. Artif. Intell. Res. 65 (2019): 457-486.

[21] Aristidou, A., Lasenby, J., Chrysanthou, Y. and Shamir, A. (2018), Inverse Kinematics Techniques in Computer Graphics: A Survey. Computer Graphics Forum, 37: 35-58.

[22] S. Farzan and G. N. DeSouza, "From D-H to inverse kinematics: A fast numerical solution for general robotic manipulators using parallel processing," 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 2507-2513.

[23] Ben Kenwright (2012) Inverse Kinematics – Cyclic Coordinate Descent (CCD),Journal of Graphics Tools, 16:4, 177-217.

[24] S. Phaniteja, P. Dewangan, P. Guhan, A. Sarkar and K. M. Krishna, "A deep reinforcement learning approach for dynamically stable inverse kinematics of humanoid robots," 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2017, pp.

[25] Ahmed R. J. Almusawi, L. Canan Dülger, Sadettin Kapucu, "A New Artificial Neural Network Approach in Solving Inverse Kinematics of Robotic Arm (Denso VP6242)", Computational Intelligence and Neuroscience, vol. 2016, Article ID 5720163, 10 pages, 2016.

[26] H. Ananthanarayanan and R. Ordóñez, "Real-time Inverse Kinematics of redundant manipulator using a hybrid (analytical and numerical) method," 2013 16th International Conference on Advanced Robotics (ICAR), 2013, pp. 1-6.

[27] Z. Al-Qurashi and B. Ziebart, "Hybrid Algorithm for Inverse Kinematics Using Deep Learning and Coordinate Transformation," 2019 Third IEEE International Conference on Robotic Computing (IRC), 2019, pp. 377-380.

[28] S. An and D. Lee, "Prioritized Inverse Kinematics with Multiple Task Definitions," 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 1423-1430, doi: 10.1109/ICRA.2015.

[29] Khaleel, H. (2018). Inverse Kinematics Solution for Redundant Robot Manipulator using Combination of GA and NN. Al-Khwarizmi Engineering Journal, 14(1), 136-144.

[30] Chun-Feng Huang, Yen-Chun Tung, Hao-Tien Lu & T.-J. Yeh (2018) Balancing control of a bicycle-riding humanoid robot with center of gravity estimation, Advanced Robotics, 32:17, 918-929.

[31] G. Tevatia and S. Schaal, "Inverse kinematics for humanoid robots," Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), 2000, pp. 294-299 vol.1.

[32] Karpińska, J., and Tchoń, K. (April 12, 2012). "Performance-Oriented Design of Inverse Kinematics Algorithms: Extended Jacobian Approximation of the Jacobian Pseudo-Inverse." ASME. J. Mechanisms Robotics. May 2012; 4(2): 021008.

[33]J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95 - International Conference on Neural Networks, 1995, pp. 1942-1948 vol.4.

[34] Guilherme Christmann and Jacky Baltes. "Balance and Steering Control of a Humanoid Robot on an Electric Scooter ",Airiti Library.2021, dio:10.6345/NTNU202100111

[35] Nizar Rokbani, Adel.M. Alimi, Inverse Kinematics Using Particle Swarm Optimization, A Statistical Analysis, Procedia Engineering, Volume 64, 2013, Pages 1602-1611, ISSN 1877-7058

# Autobiography

Chien, Lin received his B. App. Sc degree in Electrical Engineering from Fu Jen Catholic University, Taiwan in August 2017. After graduation, he has worked for two years at tian lei sheng guang company as a sound engineer. He is in charge of layout and tuning parameters for speaker sound. Then in August 2019, he continuous his Master's Degree in the Electrical Engineering Department at National Taiwan Normal University and joined the Educational Robotics Centre laboratory which focused on humanoid robots. He has been worked on the humanoid adult-size robots since 2019 and has obtained 1st place at FIRA RoboWorld Cup 2020 adult-size humanoid robot competitions. Also, got 1st place in IROS Humanoid Robot Application Challenge - Robot Magic and Music, International Competition on Intelligent Humanoid Robotics. His research interest includes robotics, computer vision and robot-human interaction.

# Academic Achievement

1. 1st place, 2020 Humanoid Robot Application Challenge - Robt Magic and Music, online competition.
2. 1st place , International Competition on Intelligent Humanoid Robotics 2020 - HuroCup Adult-Sized & Kid-Sized, Taiwan.
3. Top 20% rank (40/200) 2020-2 FROM IP TO IPO Program - Startup Competition (FITI 創新創業激勵計畫), Taiwan.