DESIGN AND IMPLEMENTATION OF 6 DOF ROTARIC ROBOT USING INVERSE KINEMATICS METHOD

Fransisko Limanuel[1], Calvin Susanto[2], Ferry Rippun[3]
[1,2,3] Electrical Engineering Department
Atma Jaya Catholic University of Indonesia
Jakarta, Indonesia
e-mail: [1]fransisko.limanuelsl@gmail.com, [2]calvinsusreal@gmail.com,
[3]ferry.rippun@atmajaya.ac.id*

ABSTRAK

Paper ini akan membahas tentang perhitungan *inverse kinematic* yang akan dipakai untuk mengendalikan robot artikulasi 6-DOF. Robot ini terdiri dari 6 smart servo Dynamixel MX-28 dengan mikrokontroller OpenCM 9.04. Robot artikulasi telah disederhanakan menjadi 4-DOF dikarenakan tidak adanya halangan pada area kerja dan tidak perlu melakukan pergerakan yang khusus. Metode perhitungan menggunakan persamaan titik potong antara bola dengan garis, sehingga dapat memudahkan penentuan titik pada perhitungan *inverse kinematic*. Pengujian menggunakan metode pengulangan untuk menemukan kesalahan absolut, kesalahan relatif, dan deviasi standar. Pengujian dilakukan dengan menentukan 3 titik yang akan dituju dan akan diulang 10 kali untuk setiap titik, lalu akan dihitung jauhnya posisi yang dihasilkan dengan posisi yang diinginkan. Hasil pengujian menunjukkan bahwa robot memiliki nilai toleransi 5,4%.

Kata kunci: 6-DOF manipulator, *Articulated robot*, *inverse kinematics* dan *forward kinematics*, Dynamixel MX-28, OpenCM 9.04

ABSTRACT

This paper will discuss the calculation of inverse kinematic which will be used to control the 6-DOF articulated robot. This robot consists of 6 Dynamixel MX-28 smart servo with OpenCM 9.04 microcontroller. The articulated robot has been simplified to 4-DOF because there are no obstacles in the work area and no special movements are required. The calculation method uses the intersection point equation between the ball and the line, so that it can make it easier to determine the point in calculating the kinematic inverse. Tests using the repetition method to find the absolute error, relative error and standard deviation. The test is carried out by determining 3 points to be aimed at and will be repeated 10 times for each point, then the resulting position will be calculated from the desired position. The test results show that the robot has a tolerance value of 5.4%.

Keywords*: 6-DOF manipulator, Articulated robot, inverse kinematics and forward kinematics, Dynamixel MX-28, OpenCM 9*

INTRODUCTION

Robot is a machine designed to assist or replace the role of humans in carrying out several tasks such as doing heavy task or daily tasks. Because the need for robot development has been increasing, the Faculty of Engineering, Atma Jaya University formed *Pusat Robotik dan* Internet of Things (PATRIOT) which will become the basis of a robot learning center. The team started the development of a robotic arm as an early stage. The robot arm in question is a robot arm with 6 degrees of freedom (DOF) called ROTARIC (Rotary Articulated Arm Robot with Interchangeable Component).

The movement of the robot can be adjusted automatically and can be controlled by humans. In the movement of the manipulator robot usually uses kinematics system. It can be divided into forward and inverse kinematics. Forward kinematics uses rotation angle of each joint to finding the end-effector position, while inverse kinematics uses the position of end-effector to finding the value of rotation angles.

This paper discuss design and implement a 6-DOF robotic arm using inverse kinematics method. With restrictions there are no obstacles in the work space.

This paper is structured as follows: The basis theory contains the definition of a robot, basic understanding of forward and inverse kinematics, specifications of servo and microcontroller used. Followed by designing the kinematic model, inverse kinematic and user interface. After that, analyze whether the relationship between the inverse kinematic model made with the forward kinematic. And ends with a conclusion.

THEORETICAL BASIS

A. Robot

The word robot comes from the Czech language, robota, which means worker, became popular when a Czech writer, Karl Capek, made a show of a comedy he wrote in 1921 entitled RUR (Rossum's Universal Robot)[7].The Robotics Institute of America (RIA) defines a robot as a re-programmable multi-functional manipulator designed to move materials, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of task.

The RIA recognizes four classes of robot: Handling devices with manual control; Automated handling devices with predetermined cycles; Programmable, servo-controlled robots with continuous of point-to-point trajectories; Robots capable of Type C specifications which also acquire information from the environment for intelligent motion[8].

The robot to be designed is an arm robot. Robot arm has several types in terms of the range and requirements needed. Broadly speaking, there are several types of Robot Arm : Gantry Robot/Cartesian Robot; Cylindrical Robot; Spherical Robot; Selective Compliant Assembly Robot Arm (SCARA); Articulated robot[1]. In this research, the robot to be designed is an articulated robot.

B. Forward and inverse kinematics

Kinematics is the study of motion without regard to the forces causing the motion[5]. Forward kinematics refers to the use of the kinematic equations of a robot to compute the position of the end-effector from specified values for the joint parameters[6]. For example the following is the forward kinematic for 2 DOF.

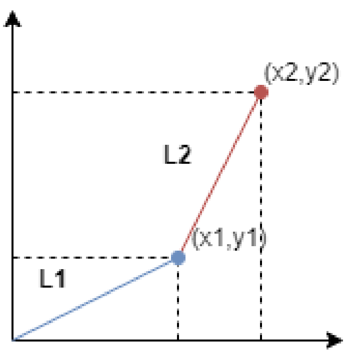


Fig.1. 2 DOF example

The forward kinematic solution for Fig.1 is

$$x_2 = L_1C_1 + L_2C_2 \qquad (1)$$

$$y_2 = L_1S_1 + L_2S_2 \qquad (2)$$

Where L_i = joint length, $C_i = \cos\theta_i$, $S_i = \sin\theta_i$.

The inverse kinematic uses the inverse of the forward kinematic method, to calculate the joint coordinates corresponding to a given situation of the end-effector[3];. However, the reverse operation is, in general, much more challenging[2], because there are many choices of angles to reach a certain position.

C. Smart Servo Dynamixel MX-28

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration [9]. Servo is a device with unidirectional communication, which means that the control unit (Arduino, Raspberry Pi or other control unit) gives position commands to the servo with a Pulse Width Modulation (PWM) signal. In an arm robot, feedback from the servo to the control unit is required. Therefore, a smart servo is needed. Smart servo does not use PWM to control position, but uses serial communication which allows two-way communication. Serial communication on a smart servo has different usage protocols depending on the manufacturer. In this design, the smart servo used is the type MX-28 from Robotis. Here are the MX-28 specifications[10]:

Table 1. MX-28 specification

Item	Specifications
Baud Rate	8,000 [bps] ~ 4.5 [Mbps]
Control Algorithm	PID control
Resolution	4096 [pulse/rev]
Operating Mode	Joint Mode (0 ~ 360 [°]) Wheel Mode (Endless Turn)
Weight	MX-28AR/AT : 77 [g], MX-28R/T : 72 [g]
Gear Ratio	193 : 1
Stall Torque	2.3 [Nm] (at 11.1 [V], 1.3 [A] 2.5 [N.m] (at 12 [V], 1.4 [A]) 3.1 [Nm] (at 14.8 [V], 1.7 [A])
Feedback	Position, Temperature, Load, Input Voltage, etc

D. OpenCM 9.04

OpenCM 9.04 is a series of microcontroller boards released by the company ROBOTIS which are Open Source. The specifications of the OpenCM 9.04 hardware are as follows[4]:

Table 2. OpenCM 9.04 specification

Item	Specification
CPU	STM32F103CB (ARM Cortex-M3)
Input Voltage	5V~16V (5V USB; 12V DXL Port; 7,4V XL-Series Port)
I/O	26 GPIO
Timer	4 (16bit)
ADC	10 (12bit)
3 Pin TTL	4 buah
USB	1 mikro USB Tipe B 2.0

DESIGN METHOD

A. Kinematic Model

The 6-DOF arm robot discussed in this paper has six rotary joints: the first, third and fifth stand vertically, and the second, fourth and sixth is horizontal are illustrated in Fig.3.(a). Table 3 shows the specifications of the manipulator.

Table 3. Specification of the length of joint

Joint	Length (cm)
1	15.5
2	12
3	10.5
4	10.5
5	6.5
6	6.8

B. Inverse Kinematic

Inverse Kinematic is a kinematic calculation method by knowing the length of each joint arm and the final position of the end-effector. With no obstacles in moving the robot arm, the configuration of the robot arm can be simplified into a 4 DOF robot arm as in the configuration in Fig.3.(b). This configuration consists of one vertical standing joint and 3 parallel horizontal joints where the third and fifth joints are assumed to be constant, not rotating. The length of joint 2 and 3 will be joined to become joint 23 and the length of joint 4 and 5 will be joined into joint 45.

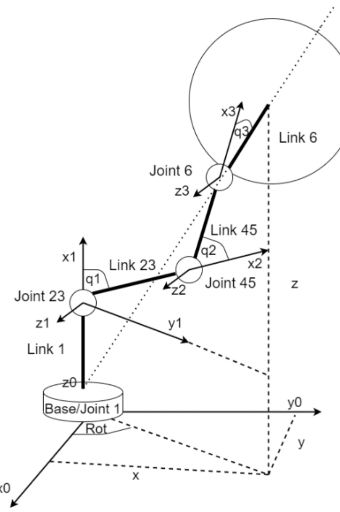
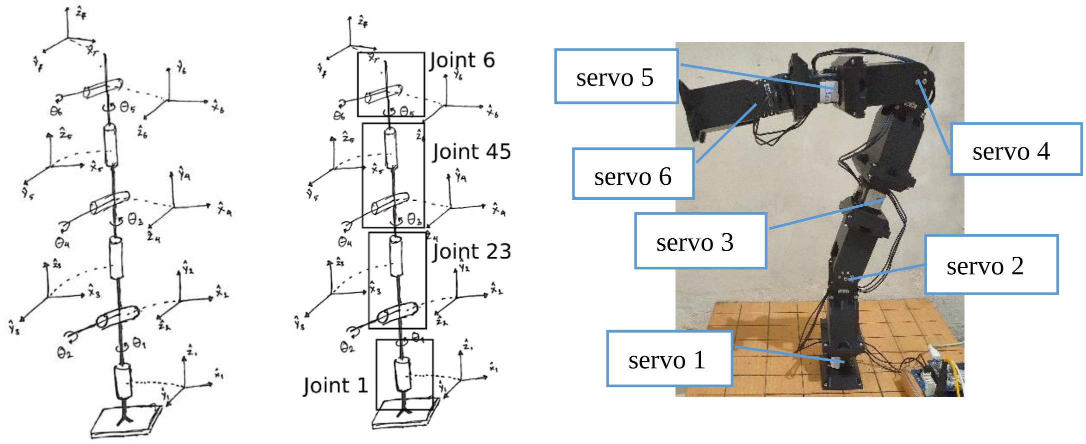


Fig.2. Calculate the inverse kinematic for 4-DOF

To calculate the inverse kinematic 4-DOF can be simplified by using the intersection equation between the sphere and the line which is illustrated in Fig.2. The result of the calculation is the point from link 45 so that it becomes 3-DOF.



(a) 6-DOF configuration (b) 4-DOF configuration (c) ROTARIC robot stucture

Fig.3. Manipulator design

The sphere and line equations are used because there are many choices of angles to reach a point. The equations for the intersection of circles and lines is as follows:

$$(1 + m_1^2 + m_1^2 m_2^2)x^2 - (2\text{a} + 2\text{b}m_1 + 2cm_1 m_2)x + (a^2 + b^2 + c^2 - r^2) = 0 \quad (3)$$

m_1 = the slope of the line on the y and x axes
m_2 = the slope of the line on the z and y
x = intersection point
r = the radius of the sphere
a = the center of the sphere in the x plane
b = the center of the sphere in the y plane
c = the center of the sphere in the z plane

The result of the quadratic equation will get two intersection points, the intersection point to be chosen is the intersection point which has the closest value to the point (0, 0, 0) and will be the end point of joint 45. The angle of joint 6 to point (0, 0, 0) will be the same as the slope value because joint 6 is parallel to the line equation used. The y and z points can be determined by the equation (5) and (6).

$$y = m_1 x \quad (4)$$

$$z = m_2 m_1 x \quad (5)$$

After getting the final position of link 45, the calculation becomes the inverse kinematic calculation of 3-DOF which is illustrated in the Fig.4.

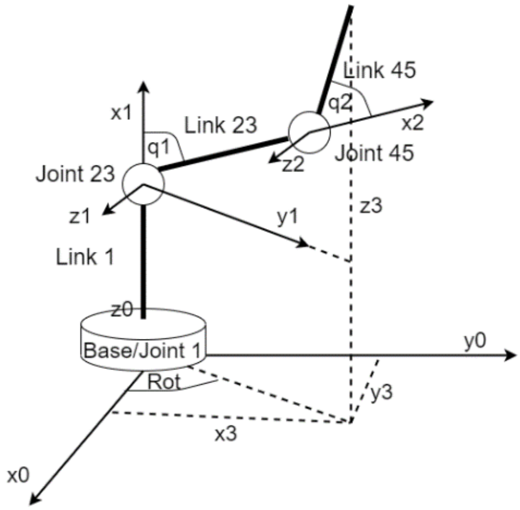
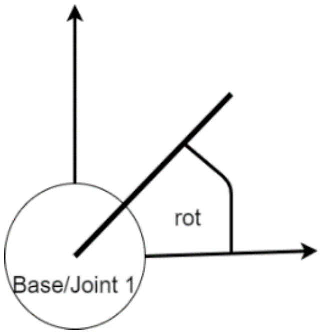
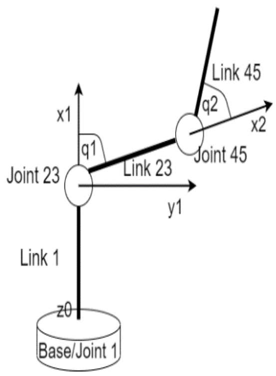


Fig.4. Calculate the inverse kinematic for 3-DOF

Calculating the inverse kinematic 3-DOF can be simplified by dividing it into two sides, the side seen from above and the side seen from the side as shown in Fig.5.



(a) Top view



(b) Side view
Fig.5. Division of 3-DOF

From Fig.5.(a), the equation for calculating the angle of joint 1 is as follows:

$$rot = atan2(y, x) \quad (6)$$

To calculate 3-DOF seen from the side can use the following trigonometric calculations:

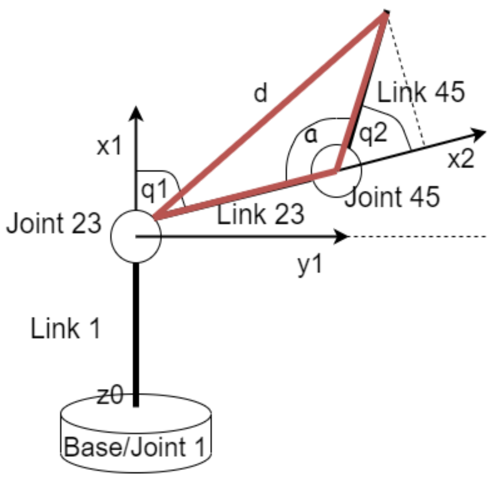


Fig.6. Trigonometric equations for calculating q2

$$d^2 = a_1^2 + a_2^2 - 2a_1a_2 \cos q_2 \quad (7)$$

$$d = \sqrt{x^2 + y^2 + (z - a_0)^2} \quad (8)$$

$$\cos \alpha = \frac{a_1^2 + a_2^2 - d^2}{2a_1a_2} \quad (9)$$

Where a_0 is the length of link 1, a_1 is the length of link 23 and a_2 is the length of link 45. Since the total angle of α plus q_2 is 180 degrees, the value of q_2 is:

$$\cos q_2 = \frac{d^2 - a_1^2 - a_2^2}{2a_1a_2} \quad (10)$$

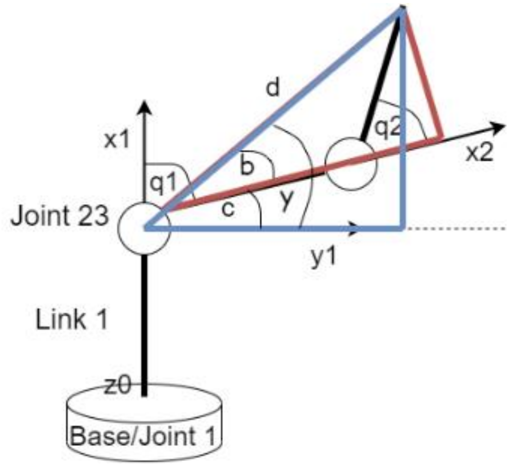


Fig.7. Trigonometric equations for calculating q1

To calculate q1, required the value of b, c and γ. Calculating the value of b, the length of joint 45 can be projected against joint 23 to create a right triangle with the base a1 + a2 cos q2, height a2 sin q2, and hypotenuse d.

$$b = tan^{-1} \frac{a_2 \sin q2}{a_1 + a_2 \cos q2} \quad (11)$$

$$\gamma = tan^{-1} \frac{z_3 - a_0}{\sqrt{x_3^2 + y_3^2}} \quad (12)$$

$$c = \gamma - \mathrm{b} \quad (13)$$

$$q_1 = \gamma - \mathrm{b} \quad (14)$$

Since the angle of connection 4 to point (0,0) will be the same as the slope value, the angle of connection 6 to connection 45 will be

$$q_3 = tan^{-1} \frac{z}{\sqrt{x_3^2 + y_3^2}} - q_1 - q_2 \quad (15)$$

Fig.8. Explaination of GUI part

C. User interface

The user interface is created using the Visual Studio program in the c# language. Fig.8 shows the designed user interface .

Parts of the user interface sequentially as follows:

1. The connection section, this section is for connecting the software to the hardware via the port on the computer used with the selected data transfer rate.
2. The command section, this section shows the commands sent from software to hardware.
3. Coordinate input part, this part as the desired coordinate input. After the coordinates are selected, then press the "go" button, the program will calculate the angle needed to reach these coordinates.
4. Angle result section, the calculation result will be displayed in this section. Once the corner is displayed it can be sent to the hardware by pressing the "WriteIKAngle" button. Then the angle obtained will be changed in the range 0-4095.
5. The length of the joint, this part is the input for the variable length of each link robot.
6. Graphical section, this section displays a simulation of the movement of the robot that will be run. This section is divided into three parts, namely the display on the x, z plane, then the y, z plane and finally the x, y plane.

ANALYSIS DATA

A. Smart Servo MX-28 torque test

Servo torque testing is carried out to determine the availability of torque. The test is carried out by installing the servo at a height of 1.5 meters, then attaching a steel rod with a length of 20 centimeters to the servo horn. The servo speed will be set at (100,200,500).

Steel rod are marked every 5 centimeters. A rope attached to each mark, then tie a bottle of water on the

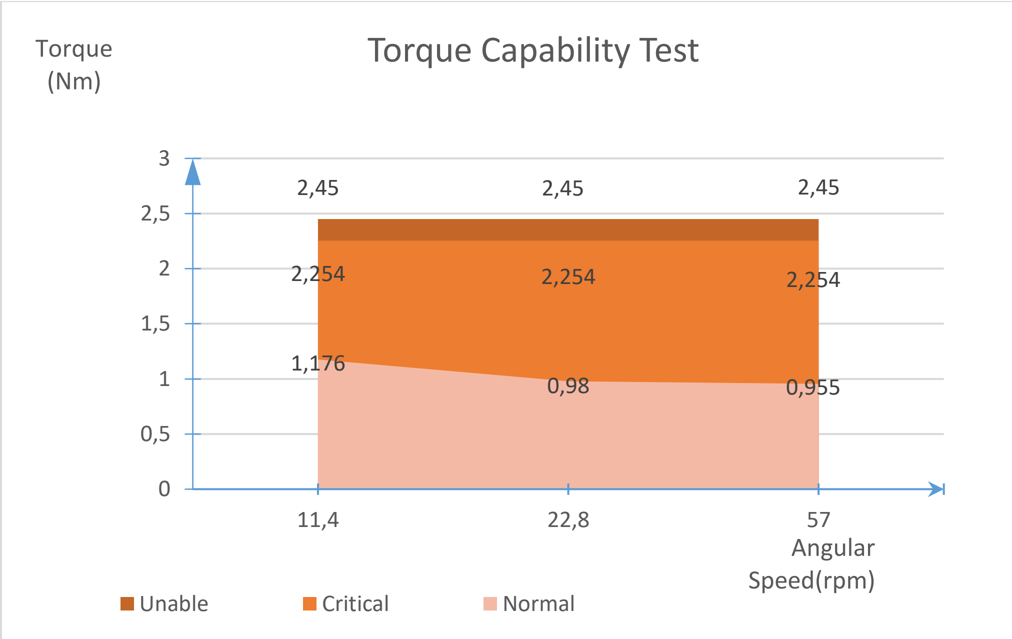


Fig.9. Torque capability test graph

rope as a variable weight, leaving it hanging above the ground.

Slowly change the angle of the servo to 180 degrees, if the servo can move and survive then the weight and length variables are in the "normal" phase, if the servo can only survive or move then it will enter the "critical" phase, and if the servo cannot survive and move then it will enter "unable" phase.

The test is carried out repeatedly by changing the mark on the steel rod, load weight and servo speed. The test results are shown in the graph in Fig.9.

Based on this torsion test, the current structure and construction of the robot is limited to the torque at the 2 servo, namely at the 2nd joint to the end-effector which weighs 0.7 kg and 0.45 m, resulting in a downward torque of about 1.54Nm which is not sufficient for the MX -28 servo to handle on the move. As a result of this problem, the resultant length from connection axis 2 to the end point must not exceed 0.3 meters and use an angular speed of 11.4 rpm.

B. Repeatability test results on ROTARIC

This test is carried out using software that is made so that the robot accepts the specified angle. The robot is placed in a box with each side representing the xy, zy, and xz planes with a resolution of 5 cm. After the robot moves to the place specified in the software, the end-effector will be measured using a box with an accuracy of 1 cm. The experiment was carried out by determining the desired position and then measured repeatedly ten times. The position to be tested is the first point (6, 1, 61), the second point (5, -10, 50) and the last point (1, 5, 60). Then the absolute error, relative error, standard deviation is calculated.

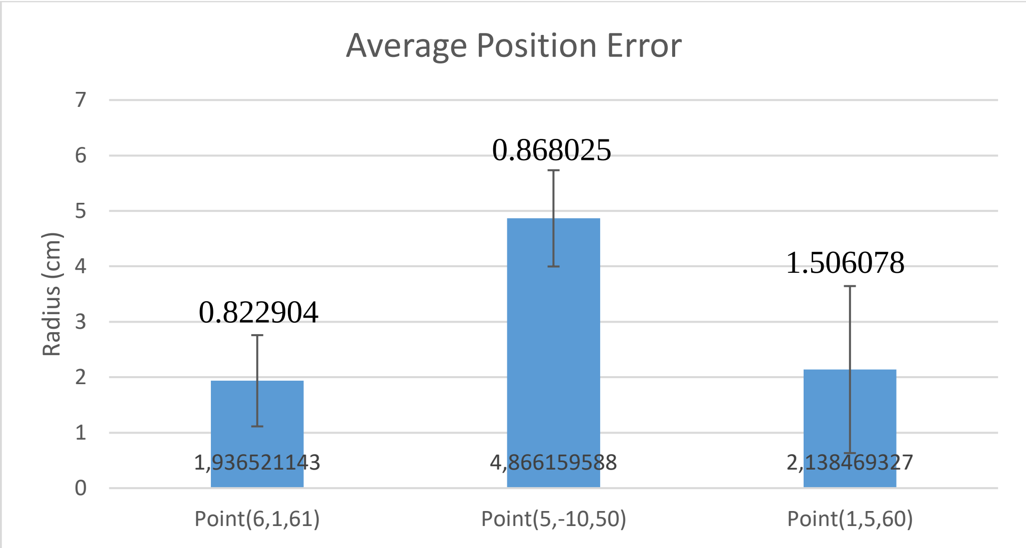


Fig.10. Graphic average position error

Fig.10 shows the average error resulting from position measurement, for position (6, 1, 61) it has an error of 1.936521143 with a standard deviation of 0.822904, for positions (5, -10, 60) it has an error of 4.866159588 with a standard deviation of 0.868025 and for the position (1, 5, 60) it has an error of 2.138469327 with a standard deviation of 1.506078. From the test, it is found that the largest average error is 4.866159588 cm.

The error obtained is quite large because the measurement tool at the time of testing has an accuracy of 5 cm so that an error of 5 cm has moved 1 scale on the measuring instrument. Error because servo 2 is unable to maintain its position due to the large torque to hold that position. The average error of the test is 5.4%.

CONCLUSION

In this paper, the inverse kinematic used has 2 calculation methods, which is the intersection of the ball with the line and the inverse kinematic 2 DOF.

The tolerance of the tool is quite high, namely 5.4%, this is due to the fact that the servo cannot maintain its position due to the large torque to hold the position, especially in servo2 because the torque capacity is 1.176Nm while the torque load on servo2 is 1.54Nm

ACKNOWLEDGEMENT

This research was supported by Universitas Katolik Indonesia Atma Jaya funded by Fakultas Teknik Unika Atma Jaya

REFERENCES

[1] Aparnathi, Rajendra. (2014). *The Novel of Six axes Robotic Arm for Industrial Applications.* IAES International Journal of Robotics and Automation (IJRA). 3. 10.11591/ijra.v3i3.4892.

[2] Donald L. Pieper. *The kinematics of manipulators under computer control.* PhD thesis, Stanford University, Department of Mechanical Engineering, October 24, 1968.

[3] K. Bouzgou and Z. Ahmed-Foitih, *Geometric modeling and singularity of 6 DOF Fanuc 200IC robot*, 4th Int. Conf. Innov. Comput. Technol.

INTECH 2014 3rd Int. Conf. Futur. Gener. Commun. Technol. FGCT 2014, pp. 208–214, 2014.

[4] OpenCM9,04. *Spesification*. (https://emanual.robotis.com/docs/en/parts/controller/opencm904/,accessed 26-11-2020)

[5] P. Marothiya and S. Saha, “Robot inverse kinematics and dynamics algorithms for windows,” Recent Trends Manuf., pp. 229–237, 2003.

[6] Paul, Richard (1981). *Robot manipulators: mathematics, programming, and control : the computer control of robot manipulators*. MIT Press, Cambridge,Massachusetts. ISBN 978-0-262-16082-7

[7] Pitowarno. Endra. 2006. *ROBOTIKA : Desain, Kontrol, dan Kecerdasan Buatan.* Yogyakarta : Andi Offset.

[8] Robot Institute of America: *NBS/RIA Robotics Research Workshop : proceedings of the NBS/RIA Workshop on Robotic Research, held at the National Bureau of Standards in Gaithersburg, MD, on November 13-15, 1979 /* (Washington, D.C. : U.S. Dept. of Commerce, National Bureau of Standards : For sale by the Supt. of Docs., U.S. G.P.O., 1981),

[9] Sawicz, D. 2012. Hobby Servo Fundamentals.

[10] Slatour. *Smart Servo: The Difference Between Smart And Regular* Servos. (https://www.robotshop.com/community/blog/show/smart-servo-the-difference-between-smart-and-regular-servos, accessed 26-11-2020).

.