Design Of 4DOF 3D Robotic Arm to Separate the Objects Using a Camera

Akhmad Fahruzi[1]*, Bimo Satyo Agomo[2], Yulianto Agung Prabowo[3]
[1,2,3]Department of Electrical Engineering, Institut Teknologi Adhi Tama Surabaya, Indonesia
[1]fahruziakhmad@itats.ac.id; [2]bimosatyoagomo@gmail.com; [3]agungp@itats.ac.id
*Corresponding author

ABSTRACT

Nowadays robotic arm is widely used in various industries, especially those engaged in manufacturing. Robotic arms are usually used to perform jobs such as picking up and moving goods from their place of origin to the location desired by the operator. In this study, a 3d 4 DOF (Degree of Freedom) robotic arm. The prototype was made to move goods with random coordinates to places or boxes whose coordinates were determined in advance. The robot can know the coordinates of the object to be taken or moved. The arm robot prototype design is completed with a camera connected to a computer, where the camera is installed statically (fixed position) above the robot's work area. The camera functions like image processing to detect the object's position by taking the coordinates of the object. Then the object coordinates will be input into inverse kinematics that will produce an angle in every point of the servo arm so that the position of the end effector on the robot arm can be founded and reach the intended object. From the results of testing and analysis, it was found that the error in the webcam test to detect object coordinates was 2.58%, the error in the servo motion test was 12.68%, and the error in the inverse kinematics test was 7.85% on the x-axis, the error was 6.31% on the y-axis and an error of 12.77% on the z-axis. The reliability of the whole system is 66.66%.

Keywords: Robotic arm, image processing, inverse kinematics.

Article History

Received : April, 27th 2021	Revised : May, 28th 2021	Accepted : May, 29th 2021

I. INTRODUCTION

Robots are said to be automatic equipment made to replace human functions. Robots are referred to as programmable multi-functional manipulators intended to perform certain tasks [1]. With the increasing activity of the manufacturing industry, robotic arms were created to help various industries to perform tasks or jobs rather than using human labor. Robots are generally used to perform unsafe, dangerous, highly repetitive, and unpleasant tasks [2]. Robots can be performed as material handling, assembly, arc welding, resistance welding, tool loading and unloading tools, painting and spraying, and others [3] [4]. For example, robotic arms are widely used in assembly or packing lines by lifting small objects in a repetitive motion that humans cannot do for long periods. Seeing the ability of an arm robot that is better than humans, the Arm Robot is made with three joints and a gripper to clamp the objects.

. In previous researchers, the inverse kinematic in the robot arm has been discussed, but the operator still inputs more on the theory or mathematical approach, the accuracy of the joint robot movement [5][6][7], and the input coordinates on the xyz axis. In this study, the robot arm, which is made on a prototype scale, has the task of being able to move objects based on their red and green colors to a certain place. The robot uses four servo motors as an arm robot actuator and a gripper as a clamp or object gripper. So that the robot can distinguish the color of the object and know the coordinates of the object on the xyz axis automatically, a camera is added in this research. The camera is installed right above the robot's work area, which is positioned unchanged. Objects that have red or green color will be captured by the camera and then processed by the computer with a visual basic program. Search for object coordinates, and kinematic inverse calculations are done on a computer through a visual basic program then the results will be sent via serial to the Arduino Uno Microcontroller. Arduino Microcontroller will control the servo motor on each joint of the robot arm with the angle value according to the calculation of the inverse kinematic.

II. METHOD

A. Robotic Arm Model Design

For making the robotic arm, the critical thing to be searched is the angles of each joint. The robotic arm prototype made in this study has three links, as seen in Fig. 1, with the length of each link arm respectively 10,10,16 cm.

 E - ISSN : 2686-6269

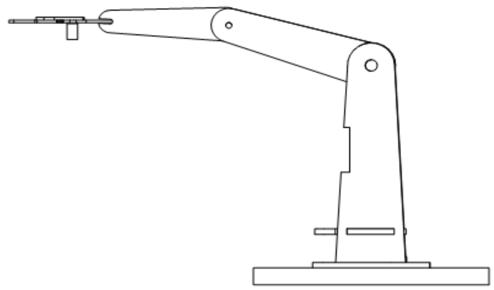

Fig. 1. Design Of 3 Links Robotic Arm

B. Inverse Kinematic

Based on the elbow manipulator diagram in Fig. 2, the target was located at the coordinates (x_c, y_c, z_c) while the angles we're searching for are θ_1, θ_2 and θ_3.

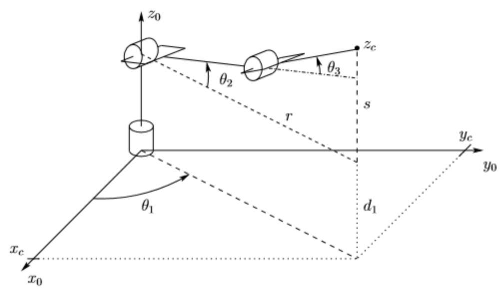

Fig. 2. Elbow Manipulator

To determine the angle (θ_1, θ_2, θ_3) as shown in Fig. 2, a method is needed, called kinematic inverse. Inverse kinematics is a method for determining the angular variables in each joint of the robotic arm. The robot's movement on the "end effector" can reach the object targeted at the coordinates (x,y,z).
From Figure 2, the value of θ_1 can be Equation (1)[10][11].

$$\theta_1 = \tan^{-1}(x_c, y_c) \tag{1}$$

Where $\tan^{-1}$ (x_c, y_c) denotes the function of the two arctangent arguments.

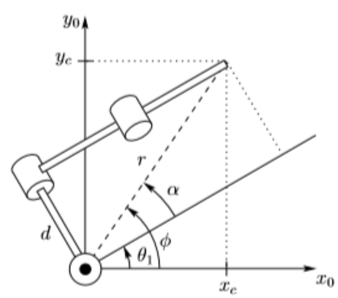

Fig. 3. Left Arm Configuration

The variable r in Fig. 3 represents the range of link length 1 and link length 2 in accordance with the angles θ_2 and θ_3 , so that the value of r is Equation (2),

$$r^2 = x_c{}^2 + y_c{}^2 - d^2 \tag{2}$$

While s is Equation (3),

$$s_2 = z_c^2 \tag{3}$$

To get the completes calculation of θ_2 and θ_3 using Equation (4).

DOI: 10.25139/ijair.v3i1.3787

E - ISSN : 2686-6269

$$\cos\theta_3 = \frac{r + s^2 - a_2{}^2 - a_3{}^2}{2a_2a_3} = \frac{x_c{}^2 + y_c{}^2 - d^2 + z_c{}^2 - a_2{}^2 - a_3{}^2}{2a_2a_3} \quad (4)$$

Because $cos^2\,\theta + sin^2\,\theta = 1$, then $sin^2\,\theta_3 = 1 - cos^2\,\theta_3$

$$sin\,\theta_3 = \sqrt{1 - cos^2\theta_3}$$

Meanwhile, to find θ_2 based on Equation (5).

$$\theta_2 = \tan^{-1}(r,s) - \text{Atan}\,(a_2 + a_3c_3, a_3s_3)$$
$$\theta_2 = \tan^{-1}\left(\sqrt{x_c{}^2 + y_c{}^2 - d^2},\ d^2\right) - \tan^{-1}(a_2 + a_3cos\theta_3, a_3sin\theta_3) \quad (5)$$

C. Forward kinematics

Before determining the kinematics of the robotic arm, the first thing that must be considered was the manufacture of Denavit-Hartenberg Parameters [9]. The D-H parameter is used to analyze the robot's movement through each link and the direction of movement of the joint on the robotic arm, as seen in Table I.

TABLE I
D-H PARAMETERS

Link	θ_i	α_i	r_i	d_i
1	θ_1	90	0	a1
2	θ_2	0	a2	0
3	θ_3	0	a3	0

Referring to Table I, the transformation of the matrix on each arm can be founded based on calculations using Equation (6).

$${}^0_3H = {}^0_1H \cdot {}^1_2H \cdot {}^2_3H \quad (6)$$

Matrix transformation on the first link based on Equation (7),

$${}^0_1H = \begin{pmatrix} & {}^0_1R & & . & {}^0_1d \\ . & . & . & . & . \\ 0 & 0 & 0 & . & 0 \end{pmatrix}$$
$${}^0_1R = \begin{pmatrix} cos\theta_1 & -sin\theta_1 & 0 \\ sin\theta_1 & cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \quad (7)$$
$${}^0_1H = \begin{pmatrix} cos\theta_1 & 0 & sin\theta_1 & 0 \\ sin\theta_1 & 0 & -cos\theta_1 & 0 \\ 0 & 1 & 0 & a_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Matrix transformation on the second link based on Equation (8),

$${}^1_2H = \begin{pmatrix} & {}^1_2R & & . & {}^1_2d \\ . & . & . & . & . \\ 0 & 0 & 0 & . & 0 \end{pmatrix}$$
$${}^1_2R = \begin{pmatrix} cos\theta_2 & -sin\theta_2 & 0 \\ sin\theta_2 & cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (8)$$
$${}^1_2H = \begin{pmatrix} cos\theta_2 & -sin\theta_2 & 0 & a_2.cos\theta_2 \\ sin\theta_2 & cos\theta_2 & 0 & a_2.sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Matrix transformation on the third link based on Equation (9),

$${}^2_3H = \begin{pmatrix} & {}^1_2R & & . & {}^1_2d \\ . & . & . & . & . \\ 0 & 0 & 0 & . & 0 \end{pmatrix} \quad (9)$$

DOI: 10.25139/ijair.v3i1.3787

E - ISSN : 2686-6269

$$^2_3R = \begin{pmatrix} cos\theta_3 & -sin\theta_3 & 0 \\ sin\theta_3 & cos\theta_3 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$^2_3H = \begin{pmatrix} cos\theta_3 & -sin\theta_3 & 0 & a_3.cos\theta_3 \\ sin\theta_3 & cos\theta_3 & 0 & a_3.sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

So that the kinematic Equation is obtained using Equation (10)-(12),

$$x = (a_3.cos\theta_3.cos\theta_1.cos\theta_2) + (a_3.cos\theta_3.cos\theta_1.-sin\theta_2) + (a_2.cos\theta_2.cos\theta_1) \quad (10)$$

$$y = (a_3.cos\theta_3.sin\theta_1.cos\theta_2) + (a_3.cos\theta_3.sin\theta_1.-sin\theta_2) + (a_2.sin\theta_2.sin\theta_1) \quad (11)$$

$$z = (a_3.cos\theta_3.sin\theta_2) + (a_3.sin\theta_3 cos\theta_2) + (a_2.sin\theta_2) + a_1 \quad (12)$$

The following calculation is an example in forwarding kinematics direction when $\theta_3 = 0^o$ $\theta_3 = 0^o$ and $\theta_3 = 0^o$. a_1=10 cm, a_2=10 cm and a_3=19 cm.

x = 19.1.1 + 19.1.1.0 + 10.1.1 = 29
y = 19.1.0.1 + 19.1.0.0 + 10.0.0 = 0
z = 19.1.0 + 19.0.1 + 10.0 + 10 = 10

D. Block Diagram

A system block diagram in Fig. 4 will be created using predetermined components. The webcam or camera in Fig. 4 is used to recognize objects that were taken, which are green and red. The computer thought a visual basic program, and the object will be searching for coordinate point value on the variable *x*,*y*, *z*-axis. The coordinate point date is used as an input parameter for finding the angles of the robot arm through the inverse kinematic method, and the value of these angles will be sent via serial communication to Arduino UNO. Arduino will control or move the servo motor based on predetermined angles.

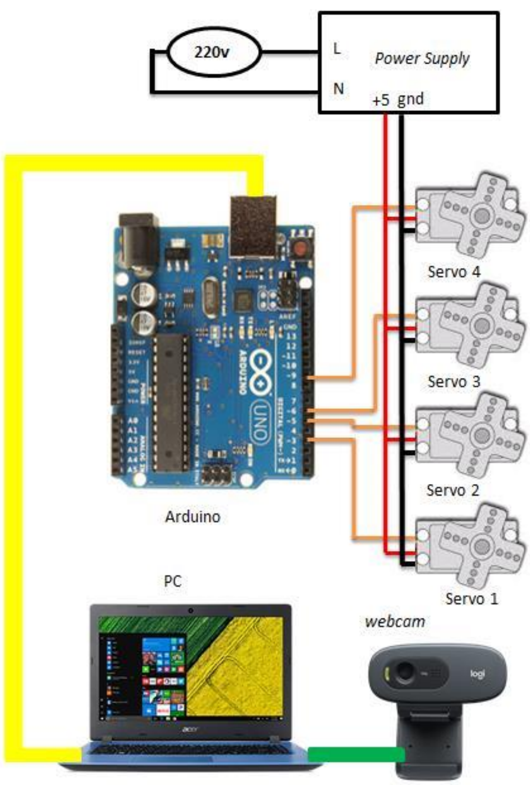

Fig. 4. Block Diagram System

E. Hardware Design

Hardware design in Figure 5, the camera was placed in the upper position to detect objects and determine the objects' coordinates. At the backside, there is a box that contains a microcontroller and a 5 volt 6 Ampere power supply as the power supply for each servo motor on the robot arm. There is a container used to put objects after the robot's arm takes them in the system.

DOI: 10.25139/ijair.v3i1.3787

E - ISSN : 2686-6269

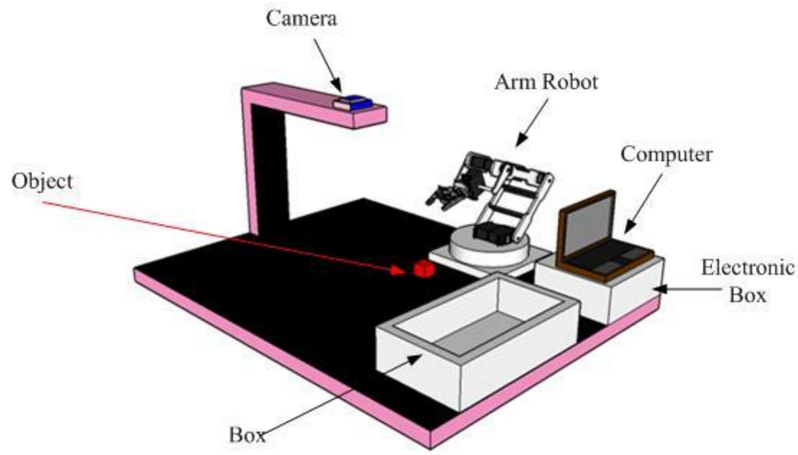

Fig. 5. Mechanical System Design

F. *Software Design*

The flowchart was used to coordinate point values on the x and y axes or (cx, cy) as shown in Fig. 6. The method used to recognize objects or objects that have green and red colors is the threshold method. Meanwhile, to find the coordinate position of the object, a mapping is carried out based on the coordinate value in pixels which is converted into cm units.

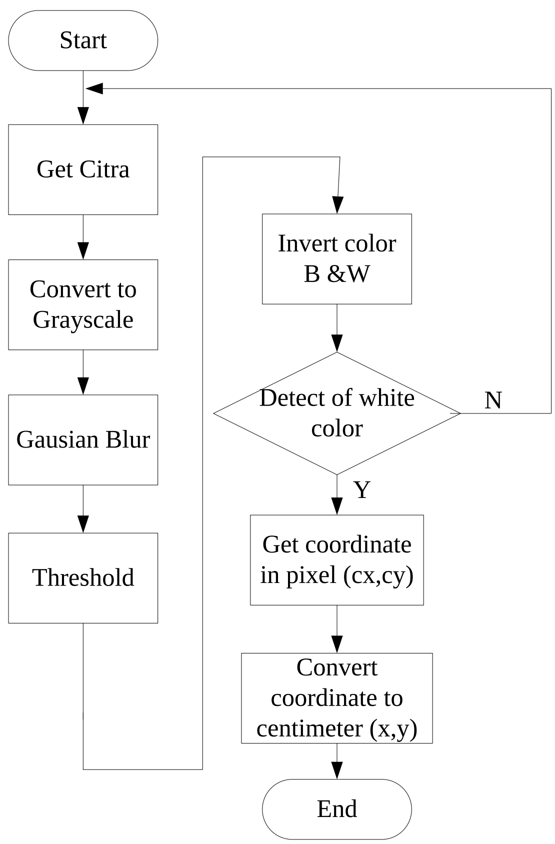

Fig. 6. The Flowchart Looks For The Coordinates Of The Object

III. RESULT AND DISCUSSION

Based on the mechanical and hardware design, the results of making the robotic arm in this study are shown in Fig.7. The performance of the arm robot has been made several tests, namely testing the reading of coordinates in Table II, servo motor Table III to Table IV, kinematic return in Table VI, and the overall result in Table VII.

DOI: 10.25139/ijair.v3i1.3787

E - ISSN : 2686-6269

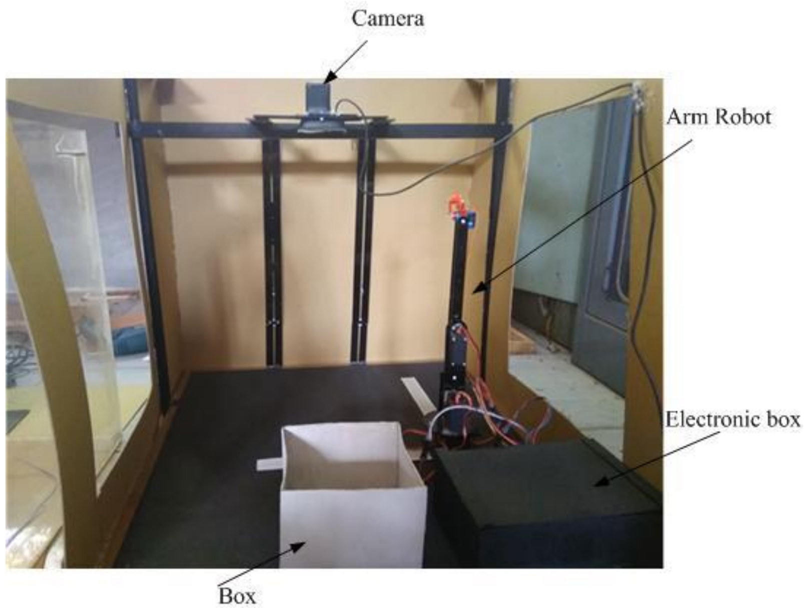

Fig.7. Mechanical design results

A. *Coordinate Reading Testing*

The camera that was placed above in Fig. 7 is used to recognize green and red objects and determine the coordinates of these objects with the result in Fig. 8.

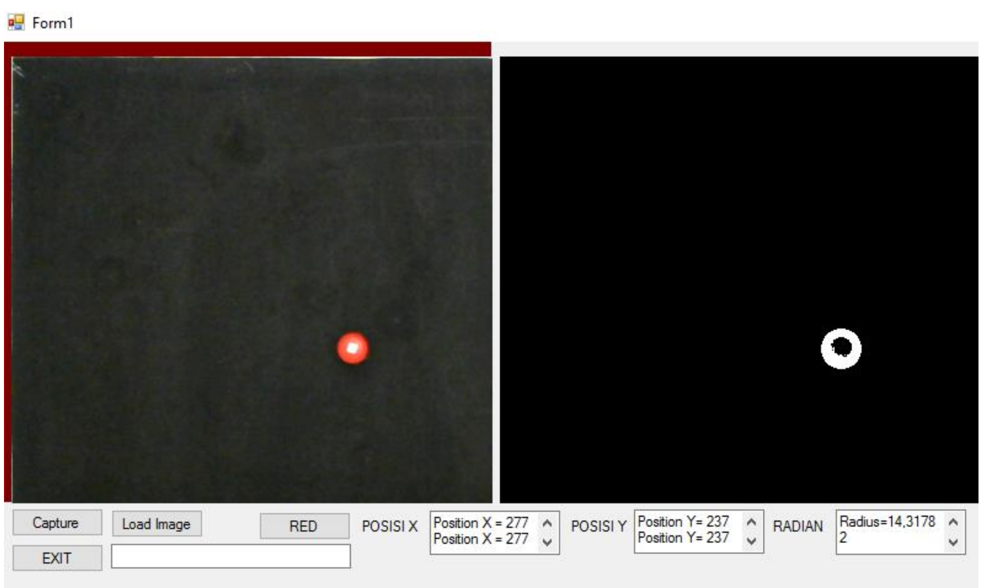

Fig. 8. Coordinates Reading in Visual Studio

Table I calculates the error while reading the coordinates carried out by the camera on the x and y axes. The test results obtained an average error of 2.58%, which can still be included in the tolerance limit. Several factors cause camera reading errors. Namely, the camera's position must not be allowed to move because when the camera moves, the coordinate points in the work area will also move so that the test results in a large error value.

TABLE I I
COORDINATE READING TESTING RESULT

Experiment	Axis	Reading Coordinate Using Camera (cm)	Actual Coordinate (cm)	Error (%)
1	X	19.66	19.8	0.71
	Y	9.21	9.3	0.97

DOI: 10.25139/ijair.v3i1.3787

E - ISSN : 2686-6269

Experiment	Axis	Reading Coordinate Using Camera (cm)	Actual Coordinate (cm)	Error (%)
2	X	18.6	18.4	1.07
	Y	8.274	8.4	1.52
3	X	20.48	20	2.34
	Y	14.73	15	1.83
4	X	16.37	16.5	0.79
	Y	13.79	14	1.52
5	X	10.504	11	4.72
	Y	9.213	9.5	3.11
6	X	8.626	9	4.33
	Y	18.016	18.1	0.33
7	X	7.922	8.3	4.77
	Y	6.279	6.5	3.51
8	X	2.875	3.1	7.82
	Y	17.195	17.4	1.19
9	X	20.246	20.5	1.25
	Y	5.575	5.8	4.03
10	X	8.274	8.4	1.52
	Y	7.101	7.4	4.21
			Error average (%)	2.58

B. Servo Motor Testing

In this test on Table III, Table IV, and Table V, four servo motors consist of three servo motors: one 6221MG-20Kg / cm servo, two MG995-10Kg / cm servo, and SG90-2.5Kg / cm. In the overall test, the servo angle obtained an average error of 12.68%. The servo caused this error with a significant difference between the input angle and the robot's angle. That way, making programs for inverse kinematics must adjust the angular error of each servo.

TABLE III
THE SERVO MOTOR TEST RESULTS ON THE BASE

Servo position	Servo type	Angel input (degrees)	Angle robot (degrees)	Error (%)
Base	MG995-10Kg/cm	20	30	50
		40	50	25
		60	70	16.66
		80	93	16.25
		100	115	15
		120	135	12.5
		140	155	10.71
		160	180	12.5
		180	205	13.88
			Error average (%)	19.16

TABLE IV
THE TEST RESULTS OF THE SERVO MOTOR ON ARM 2

Servo position	Servo type	Angel input (degrees)	Angle robot (degrees)	Error (%)
arm 2	MG995-10Kg/cm	20	25	25
		40	50	17.5
		60	70	16.66
		80	90	12.5
		100	108	8

DOI: 10.25139/ijair.v3i1.3787

E - ISSN : 2686-6269

Servo position	Servo type	Angel input (degrees)	Angle robot (degrees)	Error (%)
		120	129	7.5
		140	150	7.14
		160	170	6.25
		180	190	5.55
			Error average (%)	11.79

TABLE V
THE SERVO MOTOR TEST RESULTS ON THE ARM 1

Servo position	Servo type	Angel input (degrees)	Angle robot (degrees)	Error (%)
arm 1	6221MG-20Kg/cm	20	21	5
		40	41	2.5
		60	62	3.33
		80	75	6.25
		100	90	10
		120	110	8.33
		140	125	10.71
		160	145	9.37
		180	165	8.33
			Error average (%)	7.09

C. Inverse Kinematic Testing

From Table VI, the robot moves based on the inverse kinematics method, namely the position of the end effector to the coordinates given, but the coordinates that the robot is aiming at are still experiencing errors (errors). The average error in the inverse kinematics system is 7.85% on the x-axis, the error is 6.30% on the y-axis, and the error is 12.77% on the z-axis. This error was caused by the magnitude of the servo angle error with the input angle error that has been observed in the previous test, namely the base servo, arm servo 1, and arm servo 2. The average error of the three servos is 12.68% so that when the robot arm performs the movement at each of the given angles, the movement of the robot arm becomes less than perfect.

TABLE VI
RETURN KINEMATIC TEST RESULTS

Experiment	Coordinate input (*x*;*y*;*z*)	Robot coordinate (*x*;*y*;*z*)	Error (%)
1	15;15; 10	14.5;16; 9	3.33; 6.25; 10
2	10;20;10	9.5;19.5 ;8	5;2,5;20
3	5; 25;10	4;26;9	20;4;10
4	18 , 8, 10	17;7,5;9	5.5; 6.25;10
5	20;10;10	21;9,5;8	4.7;5; 20
6	17;11;10	15;12;9	11.7;9;10
7	18, 3, 10	17;2,7; 9	5.5;10; 10
8	21;9;10	19;8;8,5	9.5;11.1 ;15
9	3;4;10	NaN	NaN
10	9;12;10	8,5;11;9	5.5;8.3;10
		Error average (%)	7.85; 6.3; 12.77

D. Overalls Testing

The overall test is the merger of the coordinate reading program with the robotic arm reverse kinematic program. This test was conducted to determine the success rate of the robot arm when picking up and placing the objects based on the coordinates automatically, with the test results in Table VII.

The success rate in the system work is 66.67%. This was caused by various factors like camera readings that still have errors. The determination of the center point on the arm robot adjusts the condition of the camera reading. The camera's condition should not be shifted because it can be caused the result while changing the location of the work area, which results in a shift in the coordinates of the object being read. The back kinematics calculation has a slight error. The angle takes is disturbed, and the data sent to the servo motor is in the form of integer data that ignores the number behind the comma. The error rate of the servo motor from corner to corner is very large, resulting in the angle being deviated from the movement of the servo motor, which results in

DOI: 10.25139/ijair.v3i1.3787

E - ISSN : 2686-6269

the displacement of the robot arm motion less effective. The robot can move well if it can minimize the errors and use a good quality servo motor to reduce the angle reading error on the sent data.

TABLE VII
OVERALL TEST RESULTS

Experiment	Coordinate Objects			Color		Success Type	
	X	Y	Z	red	green	Yes	No
1	20.83	9.8	1	√	-	√	-
2	18.36	7.92	1	√	-	√	-
3	23.1	8.9	1	-	√	-	√
4	13.2	8.3	1	-	√	√	-
5	26.89	6.78	1	√	-	√	-
6	20.2	15.11	1	-	√	-	√
7	18.58	12.64	1	√	-	-	√
8	19.12	7.21	1	√	-	√	-
9	18.19	8.23	1	√	-	√	-
10	13.86	8.79	1	-	√	-	√
11	17.11	19.12	1	-	√	-	√
12	21.13	7.2	1	√	-	√	-
13	9.89	18.79	1	-	√	√	-
14	25.67	10.45	1	-	√	√	-
15	16.54	8.52	1	√	-	√	-
					sum	10	5
				Success Rate (%)		66.67	

IV. CONCLUSION

The results and analysis of the back kinematic system have been running, but there are still getting errors. This is due to the large errors of the servo base, arm servo 1 and 2. Arm 2 servo with an average error of the three servos, which is equal to 12.68%, so that when the robot arm moves at each given angle, the movement of the robot arm becomes less perfect. Testing and analysis of the back kinematic system have an error of 7.85% on the x-axis, an error of 6.31% on the y-axis, and 12.77% on the z-axis.

Results and analysis of the object coordinate reading system using the camera are already running. Objects can be detected as objects that are red and green on a black background. From testing and analysis, the error that occurred in the program was 2.58%.

REFERENCES

[1] Pitowarno, Endra. 2006, Robotika : *Disain, Kontrol, dan Kecerdasan Buatan*. Yogyakarta: PT Andi Offset.
[2] E. Punna, R. K. Nenavath, and B. Maloth, "*Labview Controlled Robotic ARM*."
[3] A. Elfasakhany, E. Yanez, K. Baylon, and R. Salgado. Design and development of a competitive low-cost robot arm with four degrees of freedom. Modern Mechanical Engineering, vol. 1, p. 47, 2011.
[4] C. F. Olson. Probabilistic self-localization for mobile robots. Robotics and Automation, IEEE Transactions on, vol. 16, pp. 55-66, 2000.
[5] Konietschke, R. and Hirzinger, G., 2009, May. Inverse kinematics with closed form solutions for highly redundant robotic systems. In Robotics and Automation, 2009. ICRA'09. IEEE International Conference on (pp. 2945-2950). IEEE.
[6] Achmad Zaki Rahman, Khairul Jauhuri, Dede Sumantri. Invers Kinematics dan Pengukuran Akurasi Pergerakan pada Model Robot Manipulator. Jurnal Teknik Mesin-ITI Vol. 3, No. 2, Oktober 2019.
[7] S. Gómez, G. Sánchez, J. Zarama, M. Castañeda Ramos. Design of a 4-DOF Robot Manipulator with Optimized Algorithm for Inverse Kinematics. World Academy of Science, Engineering and Technology International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering Vol:9, No:6, 2015
[8] Amin A. Mohammed and M. Sunar. Kinematics Modeling of a 4-DOF Robotic Arm. International Conference on Control, Automation and Robotics, 2015.
[9] R. N. Jazar. Theory of Applied Robotics. Boston, MA: Springer US, 2010.
[10] Guangbing Bao, Shizhao Liu, Hong Zhao, Kinematics Simulation of 4-DOF Manupulator. 2nd International Conference on Materials Science, Machinery and Energy Engineering (MSMEE), Volume 123, 2017
[11] ZHOU Fei, CHEN Fulin, SHEN Jinlong, Study of Kinematics Simulation of 4 - DOF Manipulator Based on MATLAB [J]. Machine Building and Automation, Feb 2016,45(1) : 115-119.