3D Control System of Arm Robot Prototype for Skin Cancer Detection

Dessy Novita
Department of Electrical Engineering
Padjadjaran University
Bandung, Indonesia
d.novita@unpad.ac.id

Muhamad Yerri Suyud Hasim
Department of Electrical Engineering
Padjadjaran University
Bandung, Indonesia
doeki.yerri@gmail.com

Bambang Mukti Wibawa
Department of Electrical Engineering
Padjadjaran University
Bandung, Indonesia
b.mukti.wibawa@unpad.ac.id

Arjon Turnip
Department of Electrical Engineering
Padjadjaran University
Bandung, Indonesia
turnip@unpad.ac.id

2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS) | 978-1-6654-2482-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/AIMS52415.2021.9466061

***Abstract*— Arm robot has a lack of control systems that depend on desired control for assistive medical. Our laboratory robotics & artificial intelligent at Padjadjaran University created skin cancer detection of arm robot with dark flow framework to identify skin cancer in real-time. The implementation of the arm robot was for increasing the accuracy, precision, and stability. The main purpose of this paper was to control an arm robot for skin cancer detection that is capable to scan the whole body skin to localize the skin cancers by driving the manipulator in circular or elliptical skimming. To initiate the communication with the arm robot which used Dynamixel as the actuators, we applied USB2Dynamixel as the communicator. SMPS2Dynamixel was used to supply the power into servo motors. 3D Control system software has designed, and it had some features such as; forward kinematic movement, inverse kinematic movement, and 3D simulation to help user visualize the position of the arm robot. Control software was built in MATLAB GUI environment and 3D simulation adapted Peter Corke Robotics Toolbox.**

***Index Terms* — Arm robot, Dynamixel, MATLAB GUI, Dynamixel, forward kinematic, inverse kinematic, 3D Simulation.**

I. Introduction

Arm robot is a mechanical robot to be able to move as human hands do. In its development, the arm robot is becoming more sophisticated with more degrees of freedom (DOF), and capabilities that exceed the human arm. To be called a modern arm robot, a robotic arm system consists of at least three main parts; the mechanical structure (manipulator), actuator, and control system. Arm robots can be classified based on their functional, which are arm robots, medical arm robots, industrial arm robots [1]. Arm robots for applications in medical and healthcare have been widely applied such as the da Vinci robot. It became a well-known the applied robot by Intuitive surgical for minimally invasive surgery [2] and needs to develop new procedures, kinematic and dynamic structure, design and construction [3] also to improve the rigidity of links manipulator by continuum robots for performanced [4]. Robot-assisted pelvic lymphnode dissection (RAPLND) has been used to treat patients's skin malignancy at Guy's Hospital for Merkel cell carcinoma by using a four-arm da Vinci robot. Data for patients were collected that consist of age, indication, disease pathology, site of primary tumour, previous treatment, comorbidities, operative time, complications, length of hospital stay, the histopathologically reported number of lymphnodes and nodal disease involvement. RAPLND need to expand advance technological by cross-specialty collaboration and to minimize the technique of the morbidity of a pelvic lymphadenectomy [5,6].

Furthermore, our laboratory robotics & artificial intelligent at University of Padjadjaran has begun a research on skin cancers diagnoses arm robot to identify the image of skin cancer tissue [7] and also developed a controller by adding automatic gain controller by extremum seeking control [8, 9, 10, 11]. Skin cancers divide on malignant melanoma and non-melanoma whose main types are basal cell carcinoma and squamous cell carcinoma [12, 13, 14, 15]. In 2018, Skin cancer diagnoses reported about 22 percent of melanoma and about 78 percent of non-melanoma [15, 16]. Melanoma is abnormal skin cells from the melanin-producing cells that our skin has color such as dark. When some cells growth out of control and DNA damage form a mass of cancerous cells [17].

This paper describes how to build the system starting from the hardware needed until the software. The conventional method in robotics defines joint kinematics relation for computational applications by Denavit-Hartenberg (DH) notation. The DH representation of the robotic joint's computational flexibility makes it the logical choice for describing the complex, virtual five-links manipulator. This paper was organized to find computational application of the arm robot by finding Denavit-Hartenberg parameters, to build hardware system, to develop the software and also to control the system.

II. Hardware Analysis

The main feature of this robot is capable to scan the whole body skin, to localize the skin cancers by driving the manipulator in circular or elliptical skimming [7]. Arm robot implementation was applied to increase the accuracy, precision and stability during scanning process. The arm robot was created using Dynamixel servo motor as the actuator, but it had a lack of control system. Skin cancer diagnoses of arm robot is shown on Fig.1 and dimension of robot describe on detail on Table 1. The arm robot had 5 degrees of freedom which was containing 5 servos.

There were 2 pieces of Dynamixel RX 28, 1 piece of Dynamixel RX 64, 1 piece of Dynamixel EX 106+, and 1 piece of Dynamixel Pro M42. Dynamixel servo motors had an all-in-one system where the servo was equipped with an encoder, intelligent controller, DC motor, bearing, and horn in a servo with a variety of exceptional features. The difference type servos on robot causes new problem which was different protocol programming. We used Dynamixel SDK for MATLAB to solve the issue.

The length of the arm is obtained the length from the center of rotation of the servo at a joint to the center of rotation of the next servo. Based on the datasheet for each Dynamixel servo used, the results of manipulator were measurements, and experiments in determining the safe range for each joint. We summarized range of motion (ROM) each joint of the robot in Table 2. Each joint of arm robot depicts on Fig.2 that consist of 5 joints : joint-1 waist, joint-2 shoulder, joint-3 elbow, joint-4 wrist 1, and joint-5 wrist 2. Before carrying out the test, we made a robot arm holder intending to make the Robot Arm rigid. The material was used a multiplex with a thickness of 15 mm that is depicted in Fig.3.

TABLE 1
MANIPULATOR LENGTH DIMENSION MEASUREMENT RESULTS

Joint	*Waist*	*Shoulder*	*Elbow*	*Wrist (1)*	*Wrist (2)*
Symbol	L_0	L_1	L_2	L_3	L_4
Length Link	115 mm	552 mm	295 mm	80 mm	40mm

TABLE 2
RANGE OF MOTION (ROM)

Joint	Waist	Shoulder	Elbow	Wrist 1	Wrist 2
Symbol	θ1	θ2	θ3	θ4	θ5
RoM Servo capability	300°	360°	251°	300°	300°
RoM Manipulator capability	360°	180°	320°	200°	200°
RoM Safe	300°	80°	240°	180°	180°

Denavit-Hartenberg (DH) is a method to describe the relationship between the links (arms) which in this case is assumed to be a rigid body. There are 4 parameters in this method, namely [18]:

a) Link length (a_i) is the distance from z_i to z_{i+1} measured along x_i.
b) Link twist (α_i) is the angle between z_i and z_{i+1} measured around x_i.
c) Link offset (d_i) is the distance from x_{i-1} to x_i measured along z_i.
d) Joint angle (θ_i) is the angle between x_{i-1} to x_i measured around z_i.

The Denavit-Hartenberg parameters for arm robot are described on detail in Table 3. Referring to the schematic diagram in Fig.4, Intel NUC acts as the main brain of the skin cancer diagnoses of arm robot control system.

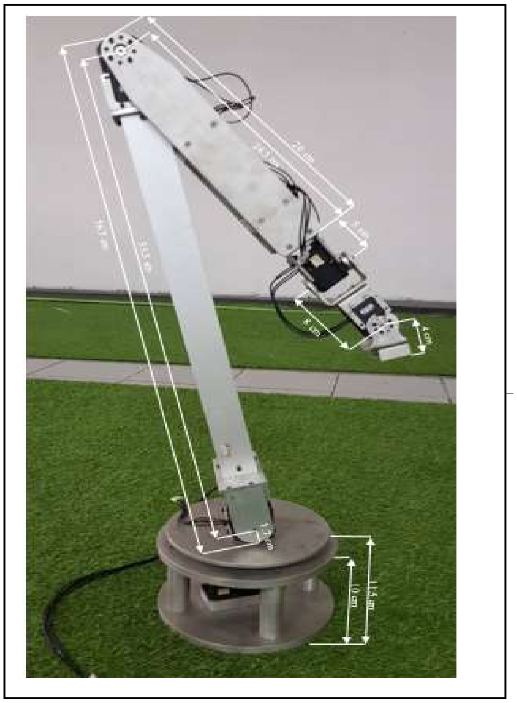


Fig. 1. Skin cancer of arm robot with dimension

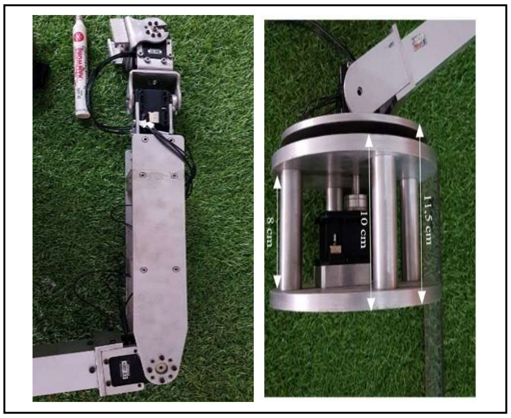


Fig. 2. Arm Robot’s Joints

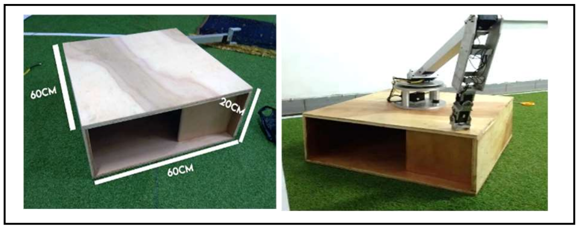


Fig. 3. The result of pooling the table with the manipulator

TABLE 3
DENAVIT HARTENBERG PARAMETER

i	θ_i	d_i	a_i	αi
1	θ_1	L1	0	90
2	θ_2	0	L2	0
3	θ_3	0	L3	-90
4	θ_4	0	L4	90
5	θ_5	0	L5	0

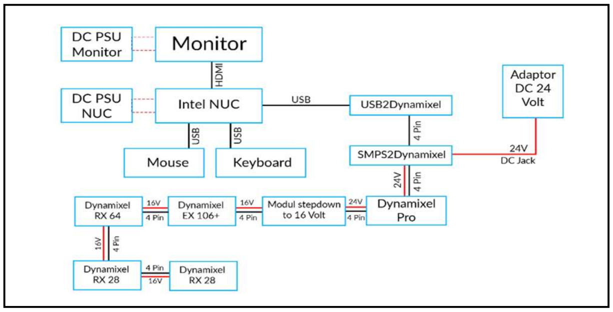


Fig. 4. Schematic diagram for hardware environment

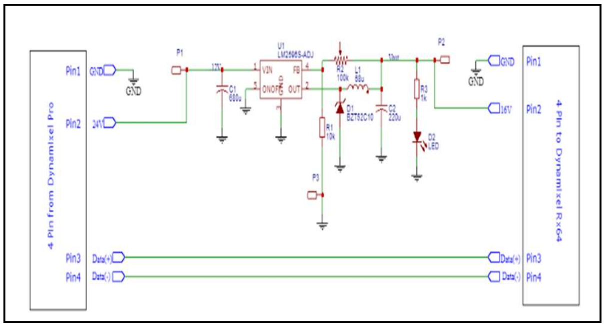


Fig. 5. Wiring diagram for stepdown module.

To drive a servo, a power supply are needed. Intel NUC does not provide enough power. We had 24 Volt DC through SMPS2Dynamixel to power the servos. The first servo was Dynamixel PRO which needed 24 Volt power, but the Dynamixel EX 106+, Dynamixel RX 64, and Dynamixel RX 28 required only 16 Volt DC voltage. We created stepdown modul using the LM2596. The wiring diagram for the LM2596-based stepdown module is shown in Fig.5.

III. Software Environtment

Robotics Toolbox is a unified library or collection of functions and classes to represent 2D and 3D simulations of a simple Robot Arm. This toolbox compiled by Peter Corke also provides artificial functions to make it easier for users to carry out forward and inverse kinematics calculations [20]. The transformation matrix describes the position or vector of a system against the coordinate system by considering rotation, translation, scaling, and perspective. The transformation matrix contains important information such as end effector position, joint angle, and the relationship between joints and links. This information can also be used in forward-kinematics and inverse-kinematics. Transformation matrix arm 5 joints described the relationship between coordinate joint to n-joints from forward kinematic on (1) [1].

$$T_5 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^0A_1 * {}^1A_2 * {}^2A_3 * {}^3A_4 * {}^4A_5 \tag{1}$$

Transition matrix for relative position and orientation of links is given (2)

$$^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & \alpha_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & \alpha_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

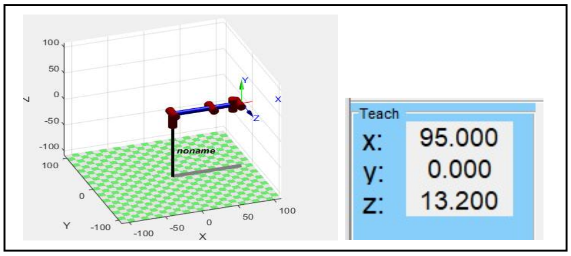
Referring to (2), we got the transformation matrix of each joint. After declaring the variables contained in the A_i matrix, the next step is to declare the arm length variable mentioned in Table 1 and input the Denavit-Hartenbergh manipulator's notation into the A_i matrix using the “subs” MATLAB function. After entering the script into the Command Window, the variables saved in the workspace. The matrix A_1 to A_5 were on (3), where C is cos and S is sin. To find the transformation matrix $A_{1\text{-}5}$ in (1), we multiplied the transformation matrix A_1 to A_5 and named it as At. The transformation matrix equations look simpler when we entered the joint variable for each joint about 0 degrees.

$$^0A_1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ; \quad ^1A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^2A_3 = \begin{bmatrix} C_3 & 0 & -S_3 & a_3C_3 \\ S_3 & 0 & C_3 & a_3S_3 \\ 0 & -1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} ; \quad ^3A_4 = \begin{bmatrix} C_4 & 0 & S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^4A_5 = \begin{bmatrix} C_5 & 0 & -S_5 & 0 \\ S_5 & 0 & C_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

These results have information that the end effector is in the position X = 95 cm, Y = 0 cm, Z = 66/5 or 13.2 cm from the centre coordinate of the arm robot. To make sure the results obtained were valid, we tried the same plot with the same parameters in the SerialLink feature on Fig.6. The Robotis Dynamixel SDK was a Software Development Kit that provided Dynamixel control functions using a communication package. The API in the library allowed users to control all Robotis Dynamixel platforms such as the AX, RX, EX, MX, XL, XM, XH, PRO-L, PRO-M, PRO-H and PRO + series. With the Robotis Dynamixel SDK, it allowed us to control Dynamixel with both protocol 1, protocol 2, and both packets at the same time. Robotis Dynamixel SDK can be used on windows operating systems, Linuxm and MacOS. To control the servo simultaneously with the simulator and the commands that have been given, we should know the position in bits / steps. We did this by reading the position capabilities of each servo using the built-in software, namely Dynamixel wizard 2. The results were obtained according to the Table 4. We found out the angular position value on the simulator and manipulator at the same angle. To change or convert the angle, we had used “Mapfun Add Ons”. Flowchart of program is shown on Fig.7.



Teach

x: 95.000

y: 0.000

z: 13.200

Fig. 6. Result of SerialLink method with 0 degrees for every joint

TABLE 4
POSITION CALIBRATION

No	Servo	Simulator			Manipulator	
		Range of Motion (Safe)	Pos Min	Pos Max	Pos Min	Pos Max
1	RX 64	300°	-150°	150°	0	1023
2	PRO	80°	-40°	40°	-30000	30000
3	EX106	240°	-150°	90	0	3800
4	RX28 (1)	180°	-90°	90°	800	200
5	RX28 (2)	180°	-90°	90°	200	800

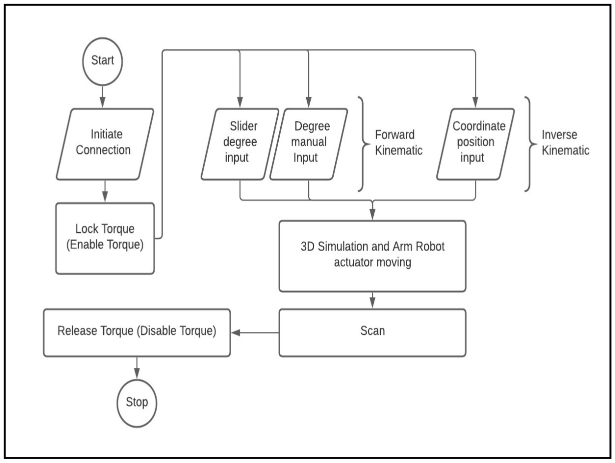

Start
Initiate Connection
Lock Torque (Enable Torque)
Slider degree input
Degree manual Input
Forward Kinematic
Coordinate position input
Inverse Kinematic
3D Simulation and Arm Robot actuator moving
Scan
Release Torque (Disable Torque)
Stop

Fig. 7. Flow chart program

IV. DESIGN USER INTERFACE OF CONTROL SYSTEM

After succeeding with the hardware, the next step was designed an integrated application software. We made some features and implemented it into one application. After going through various experiments, an integrated application has done and interface of control system shown as Fig.8. In the connection panel, we created an input box, a pop up menu, 3 buttons and a message holder on Fig.9(a). The input box allowed the user to change the communication port.

In this case, the communication port must be the USB port that connected to the arm robot. There was also pop-up menu that contained the baud rate options in communicating. The connect button initiated communication between the integrated application and the ar robot. If the communication port did not match connected to the arm robot then status text would change to "failed to connect, try to replug the usb2dyamixel". And if successful the message changed to "connected to COM X with XX baudrate". The Disconnect button cut off the communication between the manipulator and the integrated application. If this button was pressed according to the work flow chart then the torque in the manipulator disabled.

The Load button read the angular position of each joint from the arm robot and display it in the integrated application. After this button is pressed, the 3D simulation read the joint manipulator angle and created a 3D Arm Robot simulation according to the real manipulator conditions.

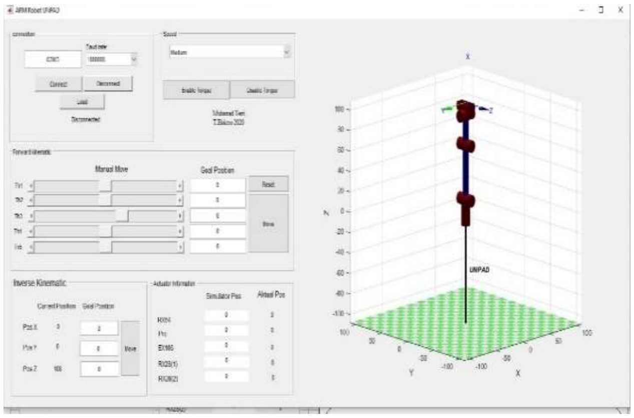
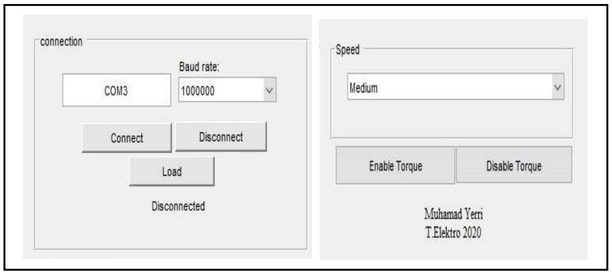


Fig. 8. Interface of control system


connection

COM3

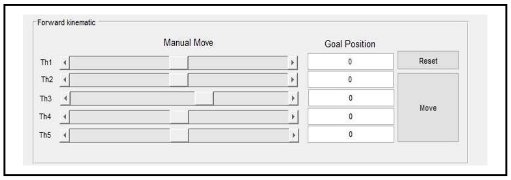
Baud rate: 1000000

Disconnected

Speed: Medium

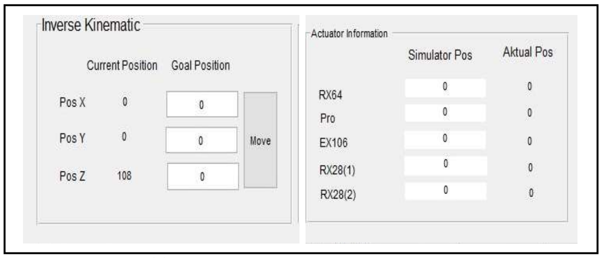
Muhamad Yerri
T.Elektro 2020

Fig. 9. (a) Connection Panel Interface, (b)Speed index and Torque Panel Interface


Forward kinematic

	Manual Move	Goal Position
Th1		0
Th2		0
Th3		0
Th4		0
Th5		0

Fig. 10. Forward Kinematic Panel Interface


Inverse Kinematic

	Current Position	Goal Position
Pos X	0	0
Pos Y	0	0
Pos Z	108	0

Actuator Information

	Simulator Pos	Aktual Pos
RX64	0	0
Pro	0	0
EX106	0	0
RX28(1)	0	0
RX28(2)	0	0

(a) (b)

Fig. 11. (a) Inverse Kinematic Panel Interface, (b) Infomation Panel Interface

In the panel on Fig.9 (b), there was a pop-up menu with a choice of 3-speed options, namely; slow, medium, fast. The speed in question is the speed of the servo movement for the simulator and manipulator. We obtained these parameters through experimental results. The Enable Torque button prompts each servo to activate its torque. This results in the servo being stiff have a resistance to other forces entering to affect the servo position. The servo only moved if it given commands by the integrated application. The Disable Torque button command was each servo to disable their torque so the user can move the manipulator freely.

In Fig.8, the panel had 3 types of controllers, the first was the slider for each joint, the second was the move button, and the last was the reset button. The slider control allowed the user to move the simulator and knew the angle of each joint without having to move the actual manipulator. When the position of the slider changes, the 3D simulation moved accordingly, and the angle information displayed in the “Post Simulator” section of the actuator information panel. Besides, the end effector information also updated whenever there was movement in the 3D simulator. The Move button moved the angle of each joint based on user input on "Goal Position". The input was 0 if the user did not change it. The limit for each joint has been applied to the script so that the simulator and manipulator only moved within the actual range of motion. The reset button on this panel had the function of changing each joint in the simulator to return to their 0 position. After pressing this button, this information updated the simulator position in the information panel and moved the slider to the value position equal to 0.

Forward kinematic panel interface is shown in Fig. 10. In Fig.11 (a), the Inverse Kinematics panel, there was information about the position of the end effector to the centre coordinate of the robot. In this panel, there was also an input box for the desired x, y and z positions. When the Move button on this panel was pressed, the application calculated the angle that required for each joint to reach the x, y and z positions according to the input box. On this occasion, we used the inverse kinematics model calculation function provided by Peter Corke. In the panel on Fig.11 (b), there were two blocks of information, the angle degree of the simulator shown in ”Simulator Pos” and the angle of the manipulator shown in ”Actual Pos”.

V. Accuracy test

Forward Kinematic had been chosen as an input to test the accuracy. Inverse Kinematic would take a time longer to calculate the degree of movement and there were some coordinates to reach by the arm robot. In Inverse Kinematics input, when the coordinates were outside the zone, the system would remind us to change the coordinate.

We moved the robot arm from one position to another randomly point and repeated 10 times. Every point position had different input degree of each joints. We did two experiments, the first was to check the accuracy of each joint. The second was to check Arm Robot multi joints movement simultaneously. We made a laser module and Laser plotting for projection as shown on Fig.12, and provided a millimeter block as a projection screen media. After every single move, we marked the laser projection on screen media with the real marker. After we got the physical marker on millimeter block, we were sampling the physic one to the digital one. After we got the digital sampling, we were going to findout standard deviation of the data sampling as (4),

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} \tag{4}$$

where σ is population standard deviation, N is the size of population, x_i is each value from the population and μ is population mean. We measured the distance between the laser and the millimeter block media as Fig.13, so we can measure the α with (5) and (6):

$$\text{Tan}\ \alpha = \frac{\text{Standard Deviation}}{\text{Distance}} \tag{5}$$

$$\alpha = \text{Tan}^{-1}\left(\frac{\text{Standard Deviation}}{\text{Distance}}\right) \tag{6}$$

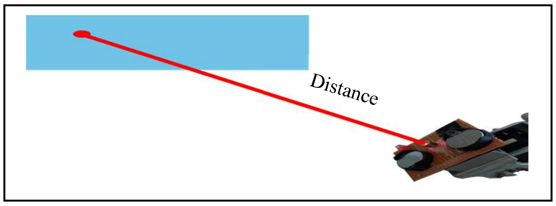


Fig. 12. Laser plotting

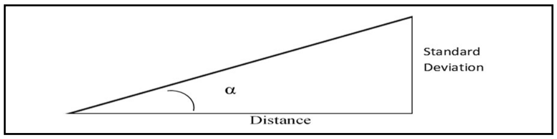


Fig.13 Measured Laser and millimeter block

VI. Result

The 3D Control System Software was running smoothly without any problem. The 3D simulation and the manipulator arm robot moving simultaneously. We need a measurement to determine whether our system was successfully built or not. So, we did the accuracy test for every joint and find out the precision with multi joint movement. Before we tested the manipulator simultaniously, we tested every single joint by moving one joint from one position input to another position. For Dynamixel RX 64 on Fig.14, we tested with 6 positions which were -30, -20, -10, 0, 10 , and 20 as inputs. First, we moved the servo with -30 as an input, then we moved the servo to position 20. There was no pattern, we moved randomly untill the position get marked 10 times. For single joint movement test we used normal speed. The results of experiments describe joint-1 RX 64 on Fig.15 and Table 5, joint-2 Dynamixel Pro on Fig.16 and Table 6, joint-3 Dynamixel EX 106 on Fig.17 and Table 7, joint-4 RX 28 (1) on Fig.18 and Table 8 and , joint-4 RX 28 (2) on Fig.19 and Table 9.

1) Joint-1: Dynamixel RX 64

(Distance mean ≈ 82 cm)

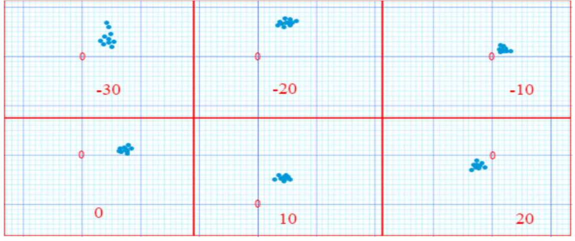


Fig. 14. Sampling for RX 64 accuracy test

Table 5
RX 64 Test Result

Input (°)	Standard Deviation (cm)	Error (°)	Error mean (°)
-30	0.1098	0.0767	0.073
-20	0.1777	0.1242	
-10	0.0778	0.0543	
0	0.0821	0.0573	
10	0.0913	0.0638	
20	0.0882	0.0616	

2) *Joint 2: Dynamixel PRO*

(Distance mean ≈ 95 cm)

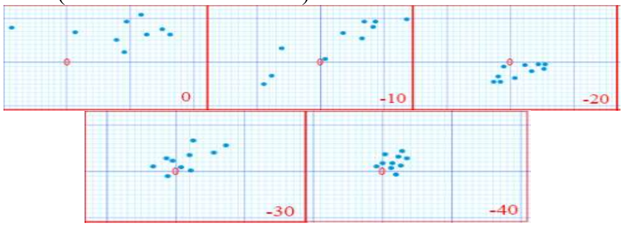


Fig. 15. Sampling for Dynamixel PRO accuracy test

TABLE 6
DYNAMIXEL PRO TEST RESULT

Input (°)	Standard Deviation (cm)	Error (°)	Error mean (°)
0	0.8009	0.4830	0.3477
-10	1.0069	0.6072	
-20	0.3920	0.2364	
-30	0.4487	0.2706	
-40	0.2343	0.1413	

3) *Joint 3: Dynamixel EX 106*

(Distance mean ≈ 90 cm)

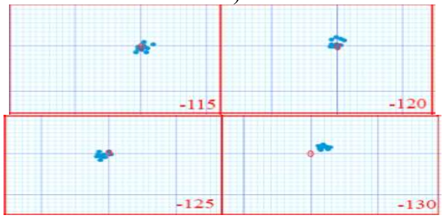


Fig. 16. Sampling for EX106 accuracy test

TABLE 7
DYNAMIXEL EX106 TEST RESULT

Input (°)	Standard Deviation (cm)	Error (°)	Error mean (°)
-115	0.1505	0.0958	0.0771
-120	0.1517	0.0966	
-125	0.1222	0.0778	
-130	0.0603	0.0384	

4) *Joint 4: Dynamixel RX28 (1)*

(Distance mean ≈ 85 cm)

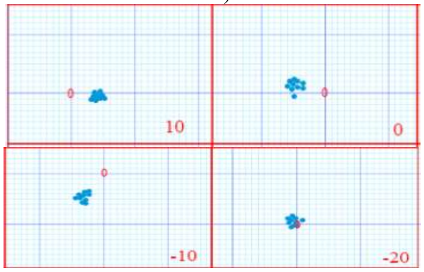


Fig. 17. Sampling for RX28(1) accuracy test

TABLE 8
DYNAMIXEL RX28(1) TEST RESULT

Input (°)	Standard Deviation (cm)	Error (°)	Error mean (°)
10	0.1045	0.0704	0.0818
0	0.1306	0.0880	
-10	0.1071	0.0722	
-20	0.1432	0.0965	

5) *Joint 4: Dynamixel RX28 (2)*

(Distance mean ≈ 85 cm)

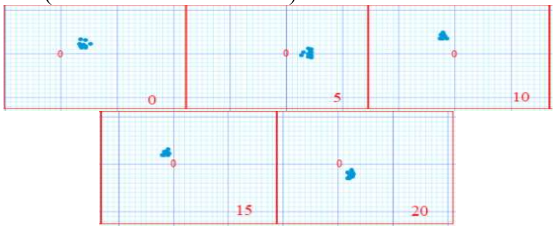


Fig. 18. Sampling for RX28(2) accuracy test

TABLE 9
DYNAMIXEL RX28(2) TEST RESULT

Input (°)	Standard Deviation (cm)	Error (°)	Error mean (°)
0	0.0906	0.0611	0.0601
5	0.1467	0.0988	
10	0.0596	0.0401	
15	0.0757	0.0510	
20	0.0733	0.0494	

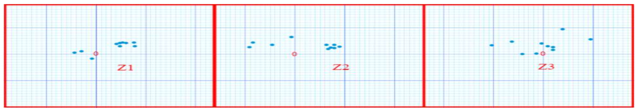


Fig.19. Digital marker sampling for Z1,Z2,Z3

TABLE 10
COMPARASION BETWEEN DATASHEET RESOLUTION WITH ERROR MEAN

Servo	Datasheet resolution (°)	Error mean (°)
Rx 64	0.29	0.073
Pro	1.368	0.3477
EX 106	0.06	0.071
RX 28 (1)	0.29	0.0818
Rx 28 (2)	0.29	0.0601

Multi joints movement test can not measure the accuracy using standard deviation (4) because there were more than one joint who caused the error. We did the multi joints movement to determine the precision of the manipulator system. We used slow speed for the movement speed. We had 3 points of position known as Z_1, Z_2, and Z_3 that were shown on Fig. 19, we can determine the precision of Z_1 was about ± 0,65 cm, Z_2 about ±1,1 cm, and Z_3 about ± 1 cm with the distance between the laser and media was more than 80 cm. Table 10 shows comparison between servo's resolution according to datasheet with the accuracy error mean for every servo. Dynamixel PRO result as shown on Fig.15 had bigger precision than multi joints movement that involved every joints to move. It is proved that movement speed of manipulator determine the precision. Dynamics analysis and PID tuning are needed to increase the accuracy and the precision of the manipulator.

VII. CONCLUSION

The computational application of the arm robot with the Denavit-Hartenberg parameters created hardware environment system, and developed the software to control the system with forward kinematics movement, inverse kinematics movement, and 3D simulation features. The system is worked out without any problem.

ACKNOWLEDGMENT

The authors would like to express gratitude for Universitas Padjadjaran, Rector of UNPAD, Directorate of Research, Community Service and Innovation of Universitas Padjadjaran, Ministry of Research, Technology and Higher Education of the Republic of Indonesia (RISTEKDIKTI) for research funding. Department of Electrical Engineering, Universitas Padjadjaran, Fellow research teams and students who participated in this research at Electrotechnic, Electric Power and Communication Technology Laboratory.

REFERENCES

[1] M. E. Moran, "Evolution of Arm Robots," National Center for Biotechnology Information, 2007. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4247431/. [Accessed 6 November 2019].

[2] Harvey, Crag and Dragone, Rocco.: Surgical Robotics: The Evolution of a Medical Technology. In: Medicaldesignbriefs.com. (2016). [Online]. https://www.medicaldesignbriefs.com/component/content/article/mdb/features/articles/25006?start=1 [accessed: November, 13th, 2019].

[3] Mianowski K.: Design study of mechanical robot-manipulators for medical applications. In: MeSRob, (2016).

[4] Jessica B.K, D., Caleb, R., Howie, C.: Continuum Robots for Medical Applications: A Survey. In: IEEE Transactions on Robotics. 31(6) : 1261-1280, (2015).

[5] A.D. MacKenzie Ross, P. Kumar, B.J. Challacombe, P. Dasgupta, J.L.C. Geh, The addition of the surgical robot to skin cancer management, Annals of the Royal College Surgeons of Englands, 95: 70-72, (2013).

[6] William S, David S. F, James J, David K.O, Robot-assisted laparoscopic transperitoneal pelvic lymphadenectomy and metastasectomy for melanoma: initial report of two cases. In: Journal Robotic Surgery, 4: 129-132, (2010).

[7] D.Novita, A.Abdurrochman and A.Sholahuddin, "Control Prototype Manipulator Robot for Skin Cancer Therapy", book of chapter Springer international publication at Cyber-Physical, Computer and Automation System (CPCAS) chapter in Advances in Intelligent Systems and Computing Vol.1291 on Springer 2021, https://www.springer.com/gp/book/9789813340619#aboutAuthors

[8] D.Novita, S.Yamamoto.: Extremum Seeking for Dead-Zone Compensation and Its Application to a Two-Wheeled Robot. In: Jounal of Automation and Control Engineering, August,; 3 (4) : 265-269, (2015). doi: 10.12720/joace.3.4.265-269.

[9] M. Chan, K. Kong, M. Tomizuka, : Automatic controller gain tuning of a multiple joint robot based on modified extremum seeking control. In: The 18th IFAC world congress, Milano (Italy) ,4131-4136, August 28 - September 2, (2011).

[10] N. J. Killingsworth, M. Krstic.: PID tuning using extremum seeking. In: IEEE Control System Magazine, 70-79, February (2006).

[11] K. Kong, K. Inaba, M. Tomizuka, Real-time nonlinear programming by amplitude modulation. Proceedings of DSCC2008 ASME Dynamic Systems and Control Conference, Ann Arbor, Michigan, USA, 1-8, October 20-22, (2008).

[12] R. Gordon. : Skin Cancer : An Overview of Epidemiology and Risk Factors. Seminars in Oncology Nursing, 9: 160-9, (2013).

[13] V.Lazareth.: Management of Non-Melanoma Skin Cancer. Seminars in Oncology Nursing, 29: 182-94, (2013).

[14] Almansour, Ebtihal and Jaffar, M.Arfan.: Classification of Dermoscopic Skin Cancer Images Using Color and Hybrid Texture Features. In: IJCSNS International Journal of Computer Science and Network Security, 16(4), April, (2016)

[15] Word cancer research fund/ American Institute for Cancer Research, Food, nutrition, physical activity and prevention of cancer: a global perspective, Skin cancer report 2019, Washington, DC:AICR, 2019, available from wcrf.org/about-the-report.

[16] Ferlay J, Ervik M, Lam F, et al 2018, Global Cancer Observatory: Cancer Today, available from https://gco.larc.fr/today.

[17] Mayo Clinic: Skin Cancer. In https://www.mayoclinic.org/diseases-conditions/skin-cancer/symptoms-causes/syc-20377605

[18] V. N. Iliukhin, K. B. Mitkovskii, D. A. Bizyanova, and A. A. Akopyan, "The Modeling of Inverse Kinematics fr 5 DOF Manipulator," Procedia Eng., vol. 176, pp. 498–505, 2017, doi: 10.1016/j.proeng.2017.02.349.

[19] "Robot Arm Terminology," what when how, 2010. [Online]. Available: http://what-when-how.com/computer-graphics-and-geometric-modeling/transformations-and-the-graphics-pipelinebasic-computer-graphics-part-5/. [Accessed 14 November 2019].

[20] P. C. "MATLAB toolboxes: robotics and vision for students and teacher," IEEE Robotics and Automation Magazine, vol. 14, pp. 16-17, December 2007.