KINEMATICS ANALYSIS AND IMPLEMENTATION OF THREE DEGREES OF FREEDOM ROBOTIC ARM BY USING MATLAB

Hasan Dawood Salman[1] aldulaimi_hasan@yahoo.com

Mohsin Noori Hamzah[1] 20066@uotechnology.edu.iq

Sadeq Hussein Bakhy[1] sadeqbakhy@yahoo.com

[1] **Mechanical Engineering Department, University of Technology, Iraq**

ABSTRACT

The kinematics modeling of the robot arm plays an important role in robot control. This paper presents the kinematic model of a three-degree of freedom articulated robot arm, which is designed for picking and placing an application with hand gripper, where a robot has been manufactured for that purpose. The forward kinematic model has been presented in order to determine the end effector's poses using the Denavit-Hartenberg (DH) convention. For inverse kinematics, an algebraic solution based on trigonometric formulas mixed with geometric method was adopted for a 3 DOF modular manipulator taking into account the existence of a shoulder offset. MATLAB software was used as a tool to simulate and implement the motional characteristics of the robot arm, by creating a 3D visual software package under designing a Graphical User Interface "GUI" with a support simulation from robotic Toolbox (Rtb 10.3). Finally, an electronic interfacing circuit between the GUI program and the robot arm was developed using Arduino microcontroller to control the robot motion. The presented work can be applicable for learning the reality interface design methodology of the other kinds of robot manipulators and achieve a suitable solution for the motional characteristics.

Keywords: Robot, Forward kinematics, Inverse kinematics, D-H parameters, MATLAB, Arduino.

التحليل الحركي والتنفيذ لذراع روبوتي ذو ثلاث درجات حريه بأستخدام الماتلاب

حسن داود سلمان **محسن نوري حمزه** **صادق حسين باخي**

الخلاصة

النمذجة الكينماتيكية لذراع الروبوت تلعب دورا مهما في التحكم بالروبوت. يقدم هذا البحث النموذج الحركي ليد روبوت مفصلية ذو ثلاث درجات حرية، حيث تم تصميم و تصنيع الروبوت لغرض الالتقاط و النقل. تم تقديم الكينماتيكا الأماميه من أجل تحديد موقع النهايه المؤثره باستخدام طريقه دانفينت- هارتنبيرغ. بالنسبة للكينماتيكا العكسية، تم اعتماد حل جبري يعتمد على الصيغ المثلثية الممزوجة بالطريقة الهندسية للمناورذو ثلاث درجات من الحريه مع مراعاة وجود إزاحة الكتف. تم استخدام برنامج الماتلاب كأداة لاختبار وتنفيذ الخصائص الحركية لذراع الروبوت، من خلال إنشاء حزمة برامج مرئية ثلاثية الأبعاد تم تصميمها بأستخدام واجهة المستخدم الرسومية مع دعم المحاكاة بواسطه التولبوكس الروبوتية. وأخيرًا، تم تطوير دائرة ربط إلكترونية بين ذراع الروبوت وبرنامج واجهة المستخدم الرسومية باستخدام متحكم االأردوينو للتحكم في حركة الروبوت. يمكن أن يكون العمل المقدم قابلاً للتطبيق لتعلم منهجية تصميم الربط الواقعي للأنواع الأخرى من المناور الروبوتيه وتحقيق حل مناسب للخصائص الحركية.

Received : 20 - 4 - 2020
Accepted : 30 - 6-2020

INTRODUCTION

Robot manipulator is one of the interested fields in educational, medical and industrial applications. Understanding the complexity of the field of robot manipulator and their applications requires the knowledge of their kinematics, dynamics and control. Kinematics analysis is considered as a key in the study of the industrial manipulators' behavior and it is the basis of the robot motion control (Junfeng et al., 2011). Therefore, modeling and analysis in any robotics system is a decisive step before using it in these circumstances to work with high accuracy (Clothier & Shang, 2010). Many literature have discussed the kinematic analysis of the industrial robot. Abbas (2013), derived the forward kinematic of a AL5B Robot arm with 5 DOF and proposed a software package for testing the motional characteristics by using MATLAB/Simulink and AutoCAD. Mohammed and Sunar (2015), presented the forward kinematics model of a 4DOF robotic arm for both the product of exponentials formula and the DH method, the inverse kinematics problem was provided using an algebraic solution based on trigonometric formulas, and the simulation of the robot arm was executed using the MATLAB program. Also, the modeling analysis of a 6 DOF robotic manipulators using the MATLAB software for their simulation was achieved by Singh et al. (2015), Lin and Min (2015) and Barakat et al. (2017). Furthermore, A.R. Akkar and Najim A-Amir (2016) analyzed the kinematics of a 6 DOF robot arm and accomplished the kinematics arm by using LabVIEW. Momani et al. (2016), proposed genetic algorithm (GA) for solving the inverse kinematics problem of robot manipulators by optimizing the motion planning of redundant manipulators based on the concept of minimizing the accumulative path deviation in the absence of any obstacles in the workspace. Alwan and Rashid (2018), suggested a method for solving the kinematic model of a three-link robot manipulator with six DOF by dividing the inverse kinematics into two sub problems; arm for the position solving, and wrist for the orientation solving to achieve a suitable solution for tracking trajectories and simulating the result with a MATLAB robotics toolbox. Gupta (2018), presented a geometric approach to compute the inverse kinematics for a 3 DOF robotic arm and developed the arm by integrating the software setup with the hardware setup to implement it using microcontroller (Arduino). In this paper, a forward kinematic model will be analyzed based on the DH convention, and inverse kinematic solutions will be presented in algebraic solution with a geometric method and validation the results by testing and implementing the actuators of a robot arm using "GUI" in MATLAB and introducing the suitable solution in a real interface.

ROBOT DESCRIPTION

The robot manipulator was designed and fabricated based on theoretical and practical concepts relating to a robot. Where, the robot manipulator was designed based on Solidworks program, and then aluminum and hard Polylactic acid (PLA) were used to build joints, actuators holders and links. The different components of the robot were separately built and then assembled. The robot manipulator was manufactured from three links articulated, which contacted by three revolute joints as $R\perp R\parallel R$, and a hand griper was used as an end effector, as shown in figure (1). To actuate the robot manipulator, the stepper motors were used for joints and gripper, since the stepper motor is an appropriate actuator for accurate positioning. Furthermore, each stepper motor has a reduction gear box to maximize the provided torques and achieves the high precision robot manipulator motion. In addition, Microstepping drivers were used to realize more accurate controlling and smooth motion.

ROBOT KINEMATICS

Robot kinematics can be described as the connection between neighboring links by considering the four link parameters where, the definitions of the four parameters are (**Mark W. Spong 2013)**:

a_i is the link length from Z_i to Z_{i+1} axes along the X_i axis.
α_i is the twist angle from Z_i to Z_{i+1} axes about the X_i axis.
d_i is the link offset from X_i to X_{i+1} axes along the Z_i axis
θ_i is the joint angle from X_i to X_{i+1} axes about the Z_i axis.

The definition of mechanisms by means of these quantities is a convention usually called the Denavit-Hartenberg (DH) notation. The DH convention matrix comes from the product of four matrices, as in equation (1) (**Mark W. Spong 2013)**:

$$T_i^{i-1} = \text{Rot z } \theta_i .\ \text{Trans z,d}_\text{i} .\ \text{Trans x,a}_\text{i} .\ \text{Rot x,}\alpha_\text{i} \quad (1)$$

Considering each of these transformations, equation (1) can be written as:

$$T_i^{i-1} = \begin{pmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

Where, C = cos, and S = sin.

Forward Kinematic

The forward kinematics of a robot is a method to determine the desired position and orientation of the manipulator's end-effector as a function of the joint variables. However, the homogenous transformation matrix based on the DH parameters is commonly used in the kinematics analysis to specify the poses of the end-effector with respect to the base coordinate. Table 1 presents the D-H parameters of the robotic arm corresponding to the assigned frames shown in figure (2). By substituting the D-H parameters of table(1) into equation (2), the transformation matrices can be obtained as:

$$T_1^0 = \begin{pmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3)\ , \qquad T_2^1 = \begin{pmatrix} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & \text{d2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4)$$

$$T_3^2 = \begin{pmatrix} C_3 & -S_3 & 0 & a_3C_3 \\ S_3 & C_3 & 0 & a_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

The forward kinematics solution of the end-effector with respect to the base coordinate identified as T_3^0 is obtained from the product of these matrices, as depicted in equation (6).

$$T_3^0 = T_1^0\, T_2^1 T_3^2 \quad (6)$$

That leads to:
$$T_3^0 = \begin{pmatrix} C_1C_{23} & -C_1S_{23} & -S_1 & C_1(a_3C_{23} + a_2C_2) - S_1d_2 \\ S_1C_{23} & -S_1S_{23} & C_1 & S_1(a_3C_{23} + a_2C_2) + C_1d_2 \\ S_{23} & C_{23} & 0 & a_3S_{23} + a_2S_2 + d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (7)$$

Where, C_1=cos q_1 , S_1= sin q_1 , $C_{23} = \cos q_2 \cos q_3 - \sin q_2 \sin q_3$
and $S_{23} = \cos q_2 \sin q_3 + \sin q_2 \cos q_3$

The first three columns in the matrix represent the orientation of the end-effector, and the fourth column represents the position of the end-effector which can be written as:

$$P_X = C_1(a_3C_{23} + a_2C_2) - S_1d_2 \tag{8}$$

$$P_Y = S_1(a_3C_{23} + a_2C_2) + C_1d_2 \tag{9}$$

$$P_Z = a_3S_{23} + a_2S_2 + d_1 \tag{10}$$

Invers Kinematic

The inverse kinematics analysis determines the set of the joint variable values for any desired position and orientation of the end-effector, where the algebraic solution and geometry approach are applied to calculate the inverse kinematics equation. The kinematic equation of the robot arm in equation (7) can be written as:

$$T_3^0 = T_1^0 T_2^1 T_3^2 = \begin{pmatrix} r_{11} & r_{12} & r_{13} & P_X \\ r_{21} & r_{22} & r_{23} & P_Y \\ r_{31} & r_{32} & r_{33} & P_Z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{11}$$

To solve for q_1, the restatement of equation (11) via multiplying each side by $[T_1^0]^{-1}$ yields (Renfrew, 2004):

$$[T_1^0]^{-1}\ T_3^0 = T_3^1 \tag{12}$$

Where, T_3^1 can be solved in forward kinematics. By inverting T_1^0, equation (12) can be writing as:

$$\begin{pmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & 1 & -d_1 \\ -S_1 & C_1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{11} & r_{11} & P_X \\ r_{11} & r_{11} & r_{11} & P_Y \\ r_{11} & r_{11} & r_{11} & P_Z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} C_{23} & -S_{23} & 0 & a_3C_{23} + a_2C_2 \\ S_{23} & C_{23} & 1 & a_3S_{23} + a_2S_2 \\ 0 & 0 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{13}$$

Equating the (3, 4) elements from both sides of equation (13) yields:

$$-S_1P_X + C_1P_Y = d_2 \tag{14}$$

The trigonometric substitutions can be used to solve equation (14) as :

$$P_X = \rho \cos\phi$$
$$P_Y = \rho \sin\phi \tag{15}$$

Where, $\rho = \sqrt{P_X{}^2 + P_Y{}^2}$, $\phi = \text{Atan2}(P_Y, P_x)$

Substituting equation (15) into equation (14), one can obtain:

$$\text{Cosq}_1\ \text{Sin}\phi - \text{Cos}\phi\ \text{Sinq}_1 = \frac{d_2}{\rho} \tag{16}$$

OR $$\text{Sin}(\phi - q_1) = \frac{d_2}{\rho} \tag{17}$$

Hence, $$\text{Cos}(\phi - q_1) = \pm\sqrt{1 - \frac{d_2{}^2}{\rho^2}} \tag{18}$$

And so, $$\phi - q_1 = \text{Atan2}\left(\frac{d_2}{\rho}, \pm\sqrt{1 - \frac{d_2{}^2}{\rho^2}}\right) \tag{19}$$

Thus, the solution of q_1 can be obtained as:

$$q_1 = \text{Atan2}(P_Y, P_x) - \text{Atan2}\left(d_2, \pm\sqrt{P_X{}^2 + P_y{}^2 - d_2{}^2}\right) \tag{20}$$

Note that two possible solutions of q_1 related to the minus or plus sign in (20) have been found. These correspond to the right and left of the arm configuration. To solve for q_3, the matrix elements of the (1,4) and the (2,4) are equated from the both sides of equation (13), that yields:

$$C_1P_x + S_1P_Y = a_3C_{23} + a_2C_2 \tag{21}$$

$$P_Z - d_1 = a_3S_{23} + a_2S_2 \tag{22}$$

By squaring the equations (21) and (22) and adding the result, with substituting equation (14), one gets:

$$P_x{}^2 + P_Y{}^2 - d_2{}^2 + (P_Z - d_1)^2 = a_2{}^2 + a_3{}^2 - 2a_2a_3C_3 \tag{23}$$

That leads to: $C_3 = \dfrac{P_x{}^2 + P_Y{}^2 - d_2{}^2 + (P_Z - d_1)^2 - a_2{}^2 - a_3}{2a_2a_3} = D$

And, $S_3 = \pm\sqrt{1 - \text{D}^2}$

That leads to: $q_3 = \text{Atan2}\left(\pm\sqrt{1-\text{D}^2}, D\right)$ (24)

The two possible solutions for q_3 correspond to the elbow-down and elbow-up configuration. Geometry approach is used to solve q_2 via considering the two upper links, as revealed in figure (3). The value q_2 can be written as:

$$q_2 = \text{Atan2}(\text{r}, \text{s}) - \text{Atan2}(a_2 + a_3C_3, a_3S_3) \tag{25}$$

Since $\text{r}^2 = P_X{}^2 + P_y{}^2 - d_2{}^2$, $s = P_Z - d_1$

That leads : $q_2 = \text{Atan2}\left(\sqrt{P_X{}^2 + P_y{}^2 - d_2{}^2}, P_Z - d_1\right) - \text{Atan2}(a_2 + a_3C_3, a_3S_3)$ (26)

Where, the solution of q_2 is associated to the value of q_3. In general, there are four solutions for inverse kinematics which correspond to right arm-elbow down, right arm-elbow up, left arm-elbow down and left arm-elbow up. Where, the perfect solution is that minimizes the required motion of each joint to obtain same position.

SOFTWARE SETUP

In the software part, the implementation program and simulation environment for manipulator motion were developed under designing a "GUI" using MATLAB. The program can control the manipulator based on the forward and inverse kinematics analysis with a support simulation from robotics Toolbox (Rtb 10.3). The main screen of the GUI program consists of different windows of the arm control, as shown in figure (4). The main two functions of this window are the forward and the inverse kinematics, this allows the user to determine the position of the end-effector by entering the desired angles in the edit boxes in term of forward kinematics. Also, finding the joint variables is by entering the desired end-effector position by moving the angle slider or by the edit boxes in term of inverse kinematics. Another solution of the inverse kinematics can be obtained by activating the function of (Left Angle/Upper Elbow) to enforce the robot for this configuration, which is preferred in the picking and placing application. If the user enters wrong position or angle, a dialog warning message will appear. The simulation of robot arm for each state is evinced in the right side of the window; which represents the motional characteristics of the robot arm. Moreover, another function can display the required number and direction of pulses(J1, J2 and J3) for actuators to send via serial port to Arduino. In addition, a smart hand gripping option is added for the feature of work.

ACTUATORS IMPLEMENTATION

The bipolar winding stepper motors as shown in figure (5) (type NEMA 17) were used to actuate the joints of the robot arm. The rotational angle of stepper motors can be divided into a number of steps, which proportion to the number of digital pulses. Stepper motor typically has 200 steps per revolution with a step size of 1.8°. Microstepping technique was used to drive the rotor smoothly and increase the position resolution by increasing the number of required pulses. This was achieved by changing the magnetic field due to the different current levels in coils. The digital Microstepping drivers A4988 were selected to meet the current requirements of the stepper motors used in the arm. Where, this driver does an interface between the stepper motor and the controller when it receives a signal from the Adriano and then sends it to the stepper motor in the desired form (voltage and current rating of the motor) to control the value of the step angle. The used digital driver A4988 has ability to divide the full rotation of stepper motor from ratio 1/1 up to 1/16 (i.e. 200 pulse/rev with step size 1.8° to 3200 pulse/rev with step size 0.11250). The driver A4988 has selector pins (MS1, MS2, and MS3), which allow to select the step resolutions. Table (2) lists the final step size of actuators due to Microstepping drivers A4988 and the reduction by gear ratio and pulley-belt system. Where, the required steps for each joint are calculated as:

$Ji\ (pul)$ = ROUND (q_i (deg.) * J total (pulses/rev) /360̊) (27)

Where, the round function is used to get the integer value of pulses. Again, the actual joint variable can be calculated from the sent pulses as:

qi (deg.) = $Ji\ (pul)$* 360̊ / J total (pulses/rev) (28)

This may be applied in the forward kinematic equation to calculate the actual end effecter position. The CNC-shield was used in combination with these drivers. Where, the shield is a great option which already includes capacitors and presents an easy way to select the Microstepping resolution. Also, it makes wiring much easier. The microcontroller receives its input data from the processing software in the form of a string, which contains the value of the position of end-effector or the values of the angles. Since, the microcontroller decodes the string and passes it to A4988 drivers through digital pins. Figure (6) manifests the way of connecting the stepper motor with A4899 driver and Arduino.

RESULTS AND DISCUSSION

The program in MATLAB were developed to calculate the theoretical work of the forward and inverse kinematics based on the DH parameters and the equations which are solved above. The Input data of DH parameter in MATLAB program are listed in table (3). Arbitrary configurations were used to the validation of the results by testing and implementing the actuators of the forward and inverse kinematics using MATLAB "GUI" with a support simulation Rtb10.3. In the forward kinematics mode, the software takes the values of angles by user and passes them to the processing software to compute the position of end-effector theoretically, and then counts the number of step by converting the rotational actuator angle to the require pulses, by which the motors must be moved. Two arbitrary proposed configurations will be experienced and implemented, where the first one represents the home position, which is taken as a reference to compute the actuators pulses of the new configuration to avoid the cumulative error. The output results of the two cases are listed in table (4), and the simulation and implementation configurations are demonstrated in figure(7). For validation the inverse kinematics mode, the coordinates of the pose end effector with respect to the base coordinate is entered as [px, py, pz] = [390, 80, 300], which is used as an input data. The output results reveal four different sets of the inverse kinematics solutions of robotic arm which are listed in table (5), and figure (8) depicts the simulation and implementation of the left elbow up solution and the left elbow down solution. Since the stepper motors based on the integer number of pulses, the results viewed a small difference between the theoretical and actual pose depending on the step size that can be treated by adding the value to the new configuration calculation. Also, the simulation and implementation of actuator confirmed that there was no theoretical error exists.

CONCLUSIONS

Kinematics analysis of a 3 DOF robotic arm was derived using the DH convention. GUI in MATLAB with a support simulation RTB was developed to simulate and implement the motional characteristics of the robot arm. The actual implementation for the actuators was valid by the development of an electronic interfacing circuit between the GUI program and the robot arm using hardware configuration (Microstepping driver and Microcontroller). The results manifested that theoretical analysis supported the actual implementation for the actuators and the small difference between the theoretical and actual pose depending on the step size of actuator that can be treated by adding the value to the new configuration calculation. In addition, a smart hand gripping option is added for the feature of work.

Table (1): The D-H parameter for the robot arm

i	α_i (deg)	a_i(mm)	d_i(mm)	θ_i
1	-90	0	d_1	q_1
2	0	a_2	d_2	q_2
3	0	a_3	0	q_3

Table(2): Actuators step size

Joint No.	Pul/rev	Micro stepping	Gear Ratio	pulley Ratio	Total Pul/rev	Step size (deg.)
M1	200	1/16	1/1	1/4	12800	0.0281
M2	200	1/8	1/26.851	-	42962	0.00838
M3	200	1/8	1/5	1/3	24000	0.015

Table (3): DH parameters of robot arm

i	α_i(deg)	a_i (mm)	d_i(mm)	θ_i	Range (deg.)
1	-90	0	170	q_1	[-120,120]
2	0	197	10	q_2	[-20 , 120]
3	0	258	0	q_3	[-120,120]

Table(4): Forward kinematics

Cases	Position	Theoretical	Actual	Abs. Different
Case1 [q_1,q_2,q_3] = [0,-90,90]	Px (mm)	258	258	0
	Py (mm)	10	10	0
	Pz (mm)	367	367	0
Case 2 [q_1,q_2,q_3] =	Px (mm)	398.3923	398.3570	0.035329
	Py (mm)	-218.4649	-218.5304	0.065428

[-30,5,-5]	Pz (mm)	187.1697	187.1793	0.009597

Table (5): Inverse kinematics with input data [px, py, pz] = [390, 80, 300]

Cases	**Joint variable**	**Theoretical (deg.)**	**Actual (deg.)**	**Offset (deg.)**
Case1 Left Elbow Up	q_1	10.1529	10.1531	0.0020
	q_2	44.6560	44.6585	0.0025
	q_3	-46.5354	46.5301-	0.0053
Case 2 Left Elbow Down	q_1	10.1529	10.1531	0.0020
	q_2	-8.4781	-8.4758	0.0023
	q_3	-46.5354	46.5301-	0.0053
Case3 Right Elbow up	q_1	-166.9685	-166.9665	0.0020
	q_2	44.6560	44.6585	0.0025
	q_3	-46.5354	46.5301-	0.0053
Case 4 Right Elbow Down	q_1	-166.9685	-166.9665	0.0020
	q_2	-8.4781	-8.4758	0.0023
	q_3	-46.5354	46.5301-	0.0053

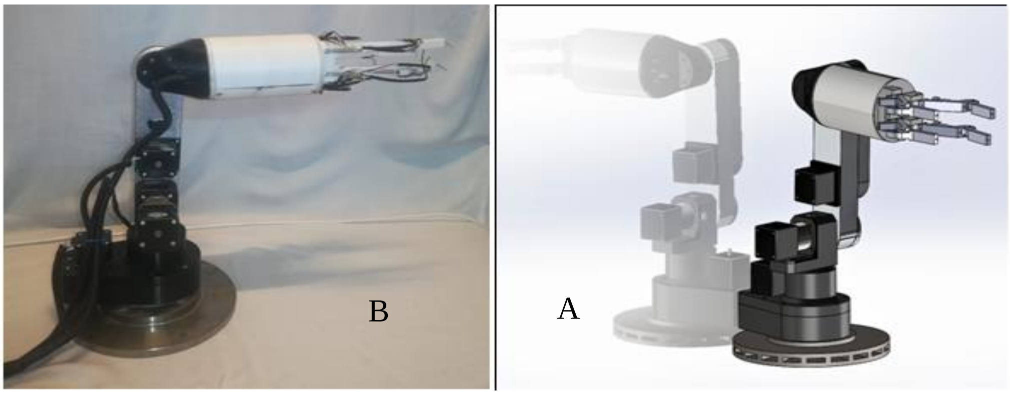

Fig. (1): 3DOF robotic arm: (A) Designed based on Solidworks program, (B) Build via aluminum and hard Polylactic acid (PLA)

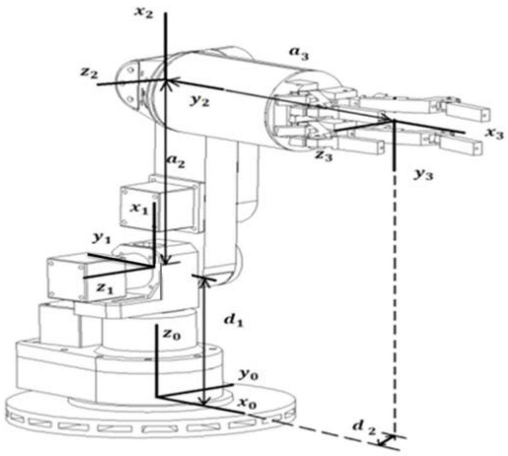

Fig. (2): The robotic arm with the assigned frames

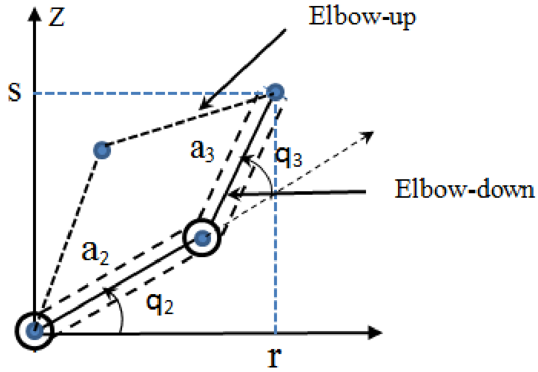

Fig. (3): Solution for q_2

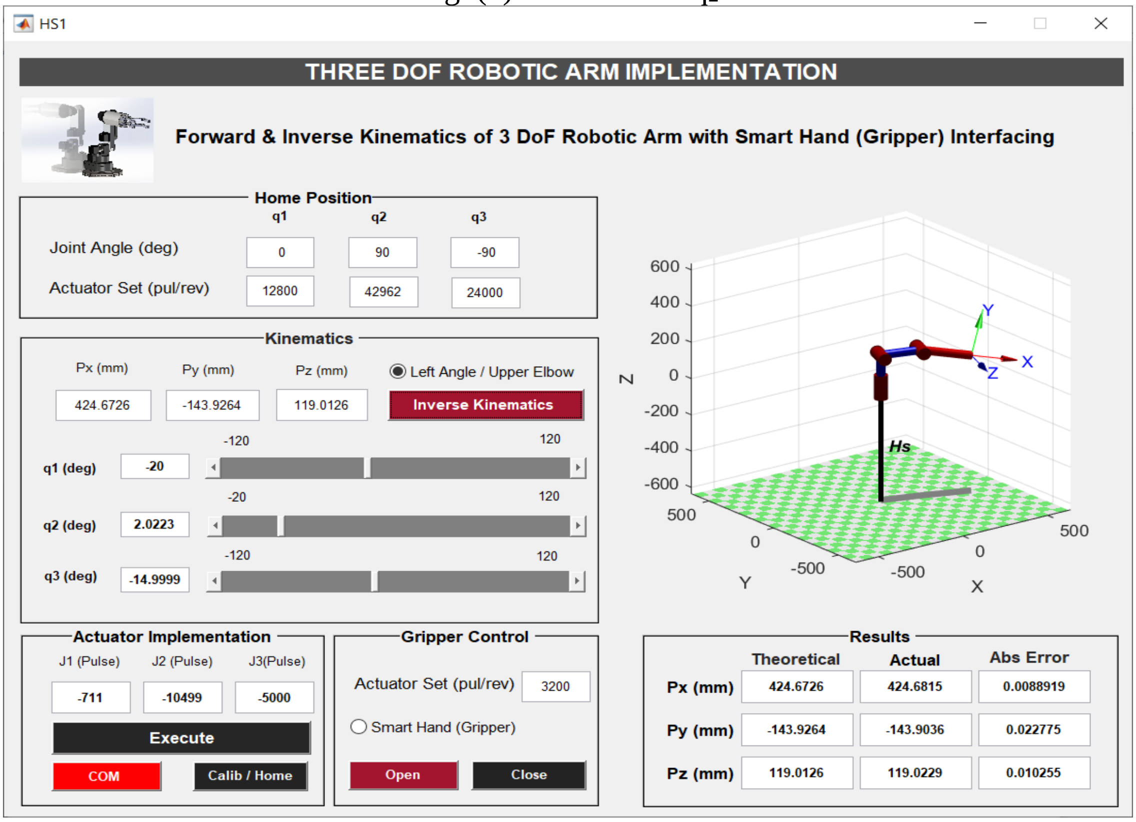

Fig. (4): The GUI in Matlab for implementation and simulation robotic arm.

Fig.(5) Stepper motor(type NEMA 17)

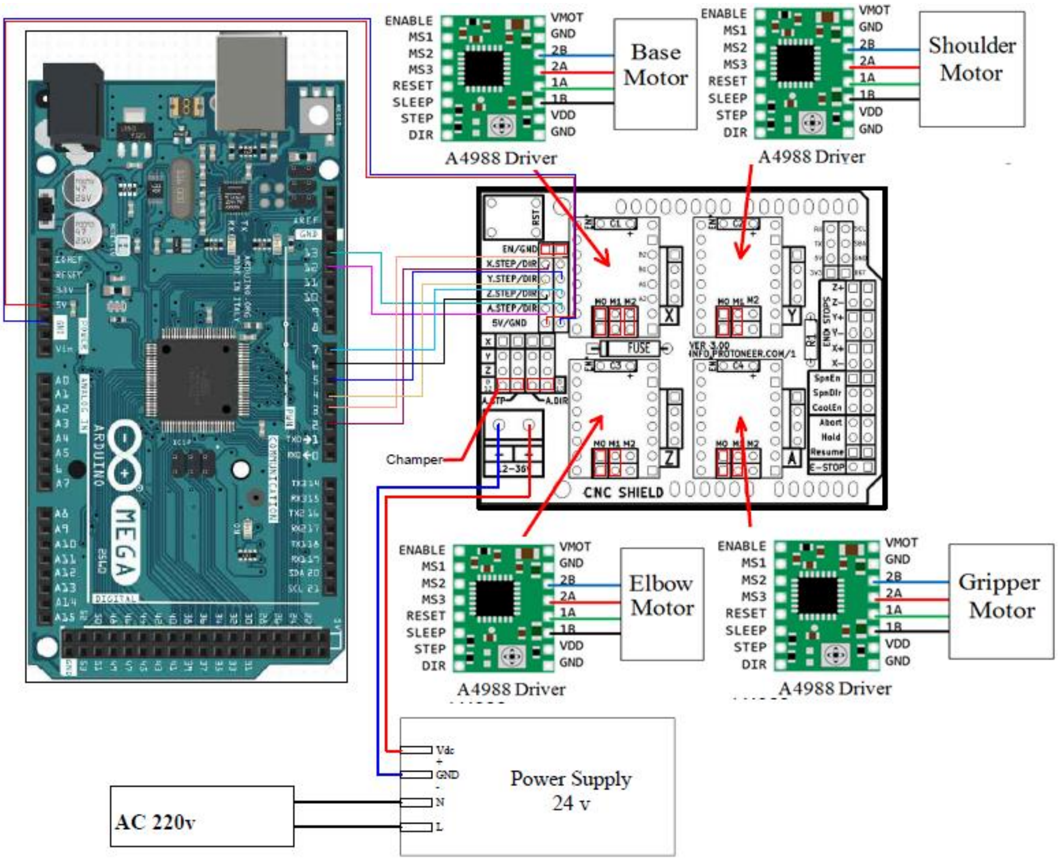

Fig. (6): The wiring diagram for connected components of CNC shield, A4988 driver, Arduino and stepper motor.

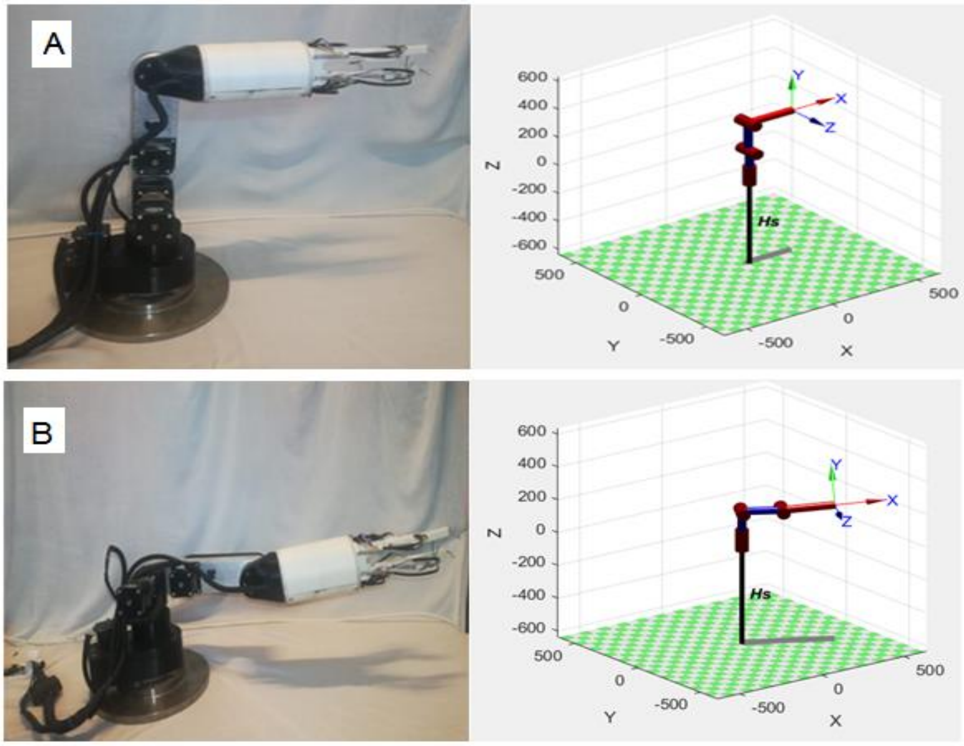

Fig. (7):Forward kinematics: A) Home position (case1) B) $[q_1, q_2, q_3]$ = [-30, 5, -5] (case 2)

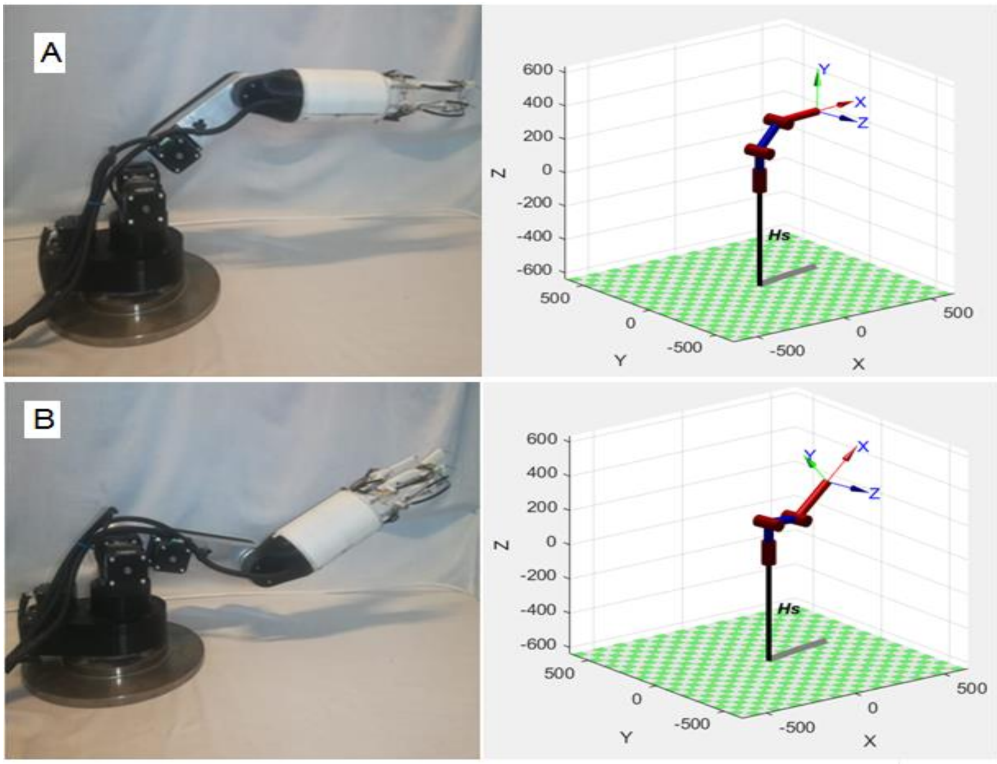

Fig.(8): Inverse kinematics: (A) Left elbow up solution, (B) Left elbow down solution

REFERENCES

A.R. Akkar, H., & Najim A-Amir, A. (2016). Kinematics Analysis and Modeling of 6 Degree of Freedom Robotic Arm from DFROBOT on Labview. Research Journal of Applied Sciences, Engineering and Technology, 13(7), 569–575.

Abbas, T. F. (2013). Forward Kinematics Modeling of 5-DOF Stationary Articulated Robots. Engineering & Technology Journal ISSN, 16816900(3), 500–513.

Alwan, H. M., & Rashid, Z. H. (2018). Kinematic Analysis and Simulation of Three Link (Open Chain) Robot Manipulator with Six DOF. Journal of Engineering and Applied Sciences, 13, 1829–1834.

Barakat, A. N., Gouda, K. A., & Bozed, K. A. (2017). Kinematics analysis and simulation of a robotic arm using MATLAB. 4th International Conference on Control Engineering and Information Technology, CEIT 2016, December, 16–18.

Clothier, K. E., & Shang, Y. (2010). A Geometric Approach for Robotic Arm Kinematics with Hardware Design, Electrical Design, and Implementation. Journal of Robotics, 2010, 1–10. https://doi.org/10.1155/2010/984823

Gupta, A. (2018). A Geometric Approach to Inverse Kinematics of a 3 DOF Robotic Arm. International Journal for Research in Applied Science and Engineering Technology, 6(1), 3524–3530. https://doi.org/10.22214/ijraset.2018.1491

Junfeng, S., Shangxing, Z., & Xuejun, Z. (2011). Analysis of Motion Control with Trajectory Planning for HP20 Manipulator. 2011 Third International Conference on Measuring Technology and Mechatronics Automation, 1, 471–473.

Lin, Y., & Min, H. (2015). Inverse kinematics of modular manipulator robot with shoulder offset based on geometric method mixed with analytical method algorithm. 2015 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE-CYBER 2015, 61175094, 1198–1203.

Mark W. Spong, Seth Hutchinson, and M. V. (2013). Robot Modeling and Control 1st ed (Vol. 9). JOHN WILEY & SONS, INC.

Mohammed, A. A., & Sunar, M. (2015). Kinematics modeling of a 4-DOF robotic arm. Proceedings - 2015 International Conference on Control, Automation and Robotics, ICCAR 2015, May, 87–91. https://doi.org/10.1109/ICCAR.2015.7166008

Momani, S., Abo-Hammour, Z. S., & Alsmadi, O. M. K. (2016). Solution of inverse kinematics problem using genetic algorithms. Applied Mathematics and Information Sciences, 10(1), 225–233. https://doi.org/10.18576/amis/100122

Renfrew, A., Book Review: Introduction to Robotics: Mechanics and Control. International Journal of Electrical Engineering & Education, , vol. 41, no. 4, pp. 388–388, 2004.

Singh, E. H., Dhillon, N., & Ansari, E. I. (2015). Forward and inverse Kinematics Solution for Six DOF with the help of Robotics tool box in matlab. International Journal of Application or Innovation in Engineering & Management (IJAIEM), 4(3), 17–22.