Kinematics Analysis of Humanoid Robot Arm

Nouran Ayman Mosbah Ali

October 6 University, Egypt, nouranayman197@gmail.com

Supervisor: Abdelrady Okasha Elnady, Associate Professor

Head of Mechatronics Department, Faculty of Engineering, October 6 University, Egypt, rady_nady@yahoo.com

Abstract– Humanoid robots have been fascinating people ever since the invention of robots. Kinematics of a robot arm deals with the geometry of motion with respect to fixed reference coordinate system without regard to the effect of forces. This paper aims to derive the forward kinematic (FK) equations by two techniques and solve the inverse kinematic (IK) equations of 7 DOF of humanoid robot arm. Denavit and Hartenberg (D-H) introduced a matrix representation of a link in order to describe the spatial geometry of a robotic manipulator. Because of this representation, one can easily understand the kinematics of robot arm. This work aims to implement D-H parameters of 7 DOF humanoid arm. The inverse Kinematics problem and obtaining its solution is one of the most important problems in robotics. It is quite complex, due to its non-linear formulations and having multiple solutions. The forward kinematics of 7DOF is calculated using the derived equations and compared with that obtained using roboanalyzer software and Peter Croke robotic MATLAB toolbox. The IK is obtained by iterative technique using newton raphson method.

Keywords- Humanoid robot arm, (DH), Forward Kinematics, inverse kinematics, Jacobian, RoboAnalyzer.

I. Introduction

The development of humanoid robots is a well-known and interesting research field in robotics. Humanoid robots are generally human-like and use bipedal locomotion. Existing humanoid robots have different capabilities, allowing them to be one of the most capable robots in various applications such as rescue, education, assisting, and entertainment. But the real challenge is the kinematics analysis of robot with higher DOF [1]. Kinematics studies the motion of bodies without consideration of the forces or moments that cause the motion. Robot kinematics refers the analytical study of the motion of a robot manipulator. Two types of kinematic analysis to study the movement of the robot namely; forward kinematics and inverse kinematics. Forward kinematics means the position of the end-effector of is given in terms of joint variables. Inverse kinematic means the joint variables are given in terms of the end effector position [2]. Denavit & Hartenberg (1955) showed that a general transformation between two joints requires four parameters. These parameters known as the Denavit-Hartenberg (DH) parameters have become the standard for describing robot kinematics [3]. A MATLAB program is one of the most programs used in many applications such as image processing, optimization, matrices, technical computing, etc. Robotic toolbox in MATLAB is used in determining the position of the end effector of the robotic arm and in simulating the movement of this robotic arm depending on the DH parameters[4]. In this paper forward kinematic (FK) equations have been derived by two techniques the first technique using Denavit and Hartenberg (D-H) transformation matrix and the second using (U_V) matrices and compared with that obtained using roboanalyzer software and Peter Croke robotic MATLAB toolbox. Closed form or analytical solution is the simplest solution for inverse kinematics. But, it is not possible to obtain the analytical relation between joint variables and robot end effector position and orientation for all robot configurations, especially in case of high degrees of freedom manipulator due to its non-linear behavior ([5]-[7]). So in this paper the numerical solution for inverse Kinematics (IK) is chosen. The inverse Kinematics have been solved by iterative technique Newton-Raphson method and simulated by the MATLAB.

II. Design Methodology

The design of adaptive robotic mechanical systems can be approached using the general formulation now proposed. This methodology is based on the use of the basic principles of mechanics, i.e., the fundamental laws of statics, Newton's laws of mechanics including their extension to rotating bodies (Euler's equations) as well as energy-based formulations of these principles. The methodology can be divided into four main steps.

Step 1: Definition of the Variables. The first step is to analyze the robotic mechanical system under study in order to determine the configuration variables and the design parameters. The vector of configuration variables, noted θ contains the motion variables (joint variables or Cartesian variables) used to describe the motion of the mechanical system.

Step 2: Mathematical Expression of the Fundamental Mechanical Properties. Once the vector of configuration variables and the vector of design parameters are defined, it is possible to use them to develop the mathematical expressions associated with the mechanical properties that represent the adaptiveness of the mechanical system.

Step 3: Derivation of the Conditions for Adaptiveness. Using the mathematical expression developed in the preceding step, it is now possible to derive the conditions under which the desired properties can be achieved for any value of the configuration variables.

Step 4: Derivation of the design rules

The latter equation is a function of the design parameters only.

It can be thought of as a set of constraints which define a subspace of the design space in which the mechanical properties.

III. Kinematics analysis of humanoid robot

Kinematics is the study of the motion of mechanical points, bodies and systems without consideration of their associated physical properties and forces acting on them. The study is often referred to as the geometry of motion, and it models these motions mathematically using algebra. The design procedure of robot arm is shown in Fig.1.

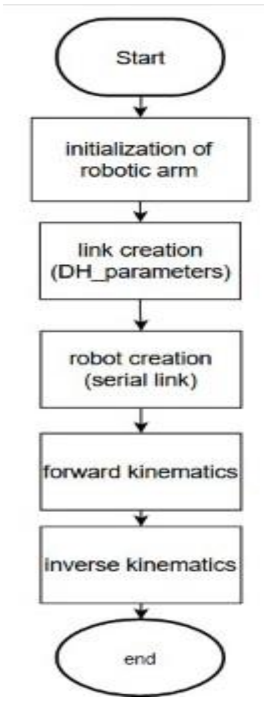

Fig.1: Design procedure of robot arm

Fig. 2 shows a schematic representation of forward and inverse kinematics.

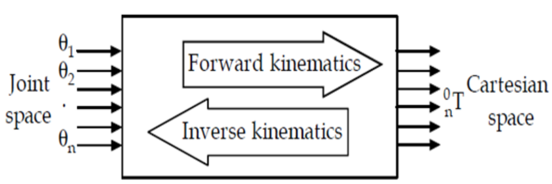

Fig. 2: Schematic representation of forward and inverse kinematics.

A. Forward kinematics

Forward Kinematics problem is concerned with the relationship between the individual joints of the robot manipulator and the position and orientation of the tool or end-effector [8]. Stated more formally, forward kinematics problem is to determine the position and orientation of the end-effector, given the values of joint variables of a robot. The joint variables are the angles between the links in the case of revolute or rotational joints, and the link extension in the case of prismatic or sliding joints. There are two methods to solve the Forward Kinematics problem; namely geometric (graph) solution and algebraic solution.

Geometric solution has been used in simple problems and Algebraic solution in the complex problems. Humanoid Robot Arm with 7-Degree of Freedom is considered as complex problem so algebraic solution is the best choice to solve the problem. The forward kinematic (FK) equations are obtained by two techniques using transformation matrix of Denavit and Hartenberg (D-H) and UV_matrices. Forward kinematics of 7DOF is calculated using the derived equations and compered with that obtained using roboanalyzer software and Peter Croke robotic MATLAB toolbox. Fig.3 shows the designed and manufactured humanoid robot arm

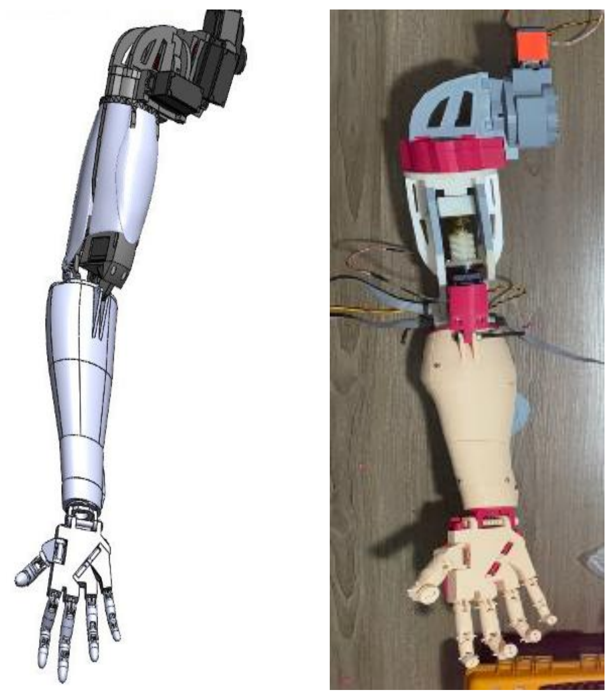

Fig. 3: Design of humanoid robot arm

1) Free body diagram of humanoid robot arm:

The free body diagram of the designed and manufactured robot arm is given in Fig.4. The lengths L1, L2, L3 and L4 are 9.65 cm, 28.774 cm, 28.04 cm, and 19.94 cm respectively.

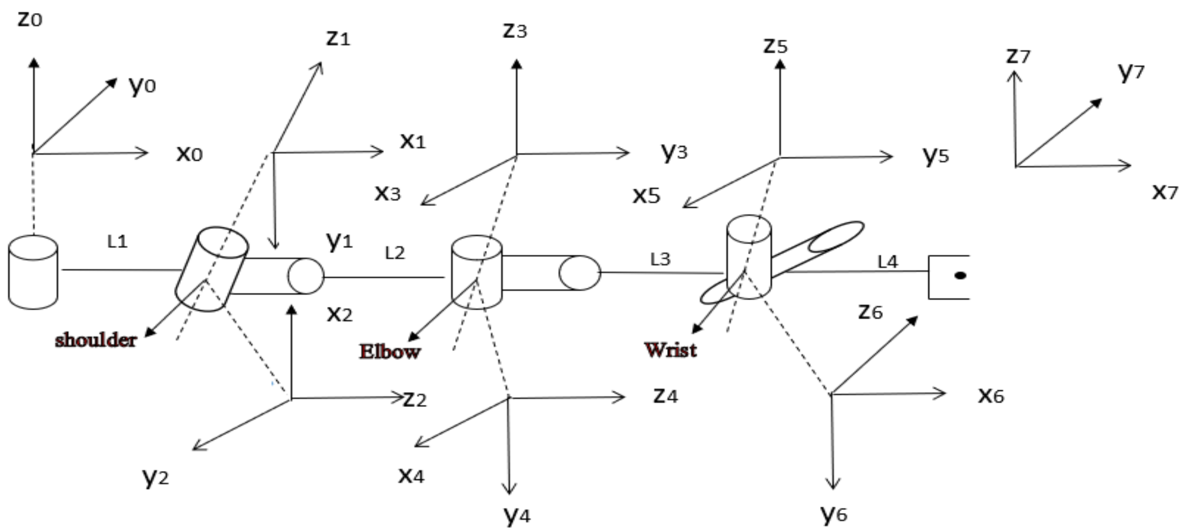

Fig. 4: Free body diagram of humanoid robot arm

2) Forward kinematics using Denavit and Hartenberg (D-H) transformation matrix:

The design parameters of DH convention are four parameters, two for link and two for joint. DH parameters are shown in Fig. 5. The present robot arm has 7 DOF and its DH parameters are listed in Table (1).

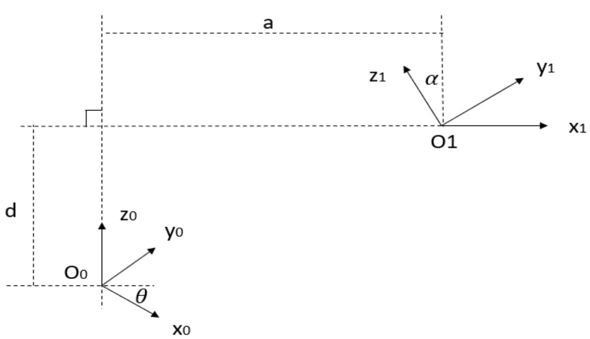

Fig. 5: The DH frame.

The transformation matrix between two successive links is the multiplication of four matrices.

$$T_i^{i-1} = R_{z.\theta i}\, T_{z.di}\, T_{x.ai}\, R_{x.\alpha i}.$$

$$T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The general transformation matrix between two successive links is given as

$$T_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Where c represents cos and s for sin.

TABLE I

DH TABBLE FOR HUMANOID ROBOT ARM

Joint#	Joint angle $\boldsymbol{\theta}$	Twist angle $\boldsymbol{\alpha}$	Link length a_i (cm)	Offset d_i (cm)	range
1	θ_1	$-\pi/2$	9.65	0	$-\pi/2 : \pi/2$
2	$\theta_2 - \pi/2$	$-\pi/2$	0	0	$-\pi/2 : \pi/2$
3	$\theta_3 + \pi/2$	$\pi/2$	0	28.774	$-\pi/2 : \pi/2$
4	θ_4	$-\pi/2$	0	0	$-\pi/2 : \pi/2$
5	θ_5	$\pi/2$	0	28.04	$-\pi/2 : \pi/2$
6	$\theta_6 + \pi/2$	$-\pi/2$	0	0	$-\pi/2 : \pi/2$
7	θ_7	$\pi/2$	19.94	0	$-\pi/2 : \pi/2$

The global transformation matrix T_7^0 is given as

$$T_7^0 = T_1^0\, T_2^1\, T_3^2\, T_4^3\, T_5^4\, T_6^5\, T_7^6. \quad (3)$$

New position = Transformation matrix x Home position

$$\begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix} = T_7^0 \begin{bmatrix} x_h \\ y_h \\ z_h \\ 1 \end{bmatrix} \quad (4)$$

Home position for the arm (cm) is $[86.33 \quad 0 \quad 0 \quad 1]$

From equation (4), the coordinates of the end effector are given by equations (5), (6), and (7).

X=9.65* $C\theta_1 - 28.7* C\theta_1 * S\theta_2$ −(997*$C\theta_7 * (S\theta_6 * (S\theta_4 * (S\theta_1 * S\theta_3 + C\theta_1 * C\theta_2 * C\theta_3)$+$C\theta_1 * C\theta_4 * S\theta_2$)- $C\theta_6 * (C\theta_5 * (C\theta_4 * (S\theta_1 * S\theta_3$+ $C\theta_1 * C\theta_2 * C\theta_3$)- $C\theta_1 * S\theta_2$* $S\theta_4$)+ $S\theta_5 * (C\theta_3 * S\theta_1 - C\theta_1 * C\theta_2 * S\theta_3$))))/50- (997* $S\theta_7 * (S\theta_5 * (C\theta_4 * (S\theta_1 * S\theta_3 + C\theta_1 * C\theta_2 * C\theta_3$)- $C\theta_1 * S\theta_2$* $S\theta_4$)- $C\theta_5 * (C\theta_3 * S\theta_1 - C\theta_1 * C\theta_2 * S\theta_3$)))/50.-(701* $S\theta_4$*($S\theta_1 * S\theta_3 + C\theta_1 * C\theta_2 * C\theta_3$))/25-(701* $C\theta_1 * C\theta_4 * S\theta_2$))/25. (5)

Y=9.65* $S\theta_1$-28.7* $S\theta_1 * S\theta_2$ +(997*$C\theta_7 * (S\theta_6 (S\theta_4 * (C\theta_1 * S\theta_3 - C\theta_2 * C\theta_3 * S\theta_1$)- $C\theta_4 * S\theta_1 * S\theta_2$) - $C\theta_6 * (C\theta_5 * (C\theta_4 * (C\theta_1 * S\theta_3 - C\theta_2 * C\theta_3$* $S\theta_1$)+ $S\theta_1$* $S\theta_2$* $S\theta_4$)+ $S\theta_5 * (C\theta_1 * C\theta_3 + C\theta_2 * S\theta_1 * S\theta_3$))))/50+(701* $S\theta_4 * C\theta_1 * S\theta_3 - C\theta_2 * C\theta_3 * S\theta_1$))25+(997* $S\theta_7 * (S\theta_5 * (C\theta_4 * (C\theta_1 * S\theta_3 - C\theta_2 * C\theta_3 * S\theta_1) + S\theta_1$* $S\theta_2$* $S\theta_4$)- $C\theta_5$*($C\theta_1 * C\theta_3 + C\theta_2 * S\theta_1 * S\theta_3$)))/50 -(701*+ $C\theta_4 * S\theta_1 * S\theta_2$))/25. (6)

Z=(997* $S\theta_7 * (S\theta_5(C\theta_2 * S\theta_4 + C\theta_3 * C\theta_4 * S\theta_2$)+ $C\theta_5 * S\theta_2 * S\theta_3$))/50-(701* $C\theta_2$* $C\theta_4$)/25-28.7* $C\theta_2$- (997*$C\theta_7 * (C\theta_6 * (C\theta_5 * (C\theta_2$* $S\theta_4$+ $C\theta_3 * C\theta_4 * S\theta_2$)- $S\theta_2$* $S\theta_3$* $S\theta_5$)+ $S\theta_6$* ($C\theta_2$* $C\theta_4$- $C\theta_3 * S\theta_2 * S\theta_4$))))/50+(701* $C\theta_3 * S\theta_2 * S\theta_4$)/25. (7)

3) Forward kinematics using UV_matrices

Using the same notations in the previous section, the global transformation matrix K_7^0:

$$K_7^0 = \begin{bmatrix} R_7^0 & d_7^0 \\ 0 & 1 \end{bmatrix}$$

R_7^0 is the rotation matrix of the end effector frame to the ground frame and given in equation (8)

R_7^0=$U_1 V_1 x\ U_2 V_2\ x\ U_3 V_3 x U_4 V_4\ x U_5 V_5\ x\ U_6 V_6 x\ U_7 V_7$. (8)

The position vector of the origin of the end effector frame to the origin of the ground frame is given by equation (9).

$$d_7^0 = d_1^0 + d_2^1 + d_3^2 + d_4^3 + d_5^4 + d_6^5 + d_7^6. \quad (9)$$

where

$$d_{10}=U_1 S_1. \quad (10)$$
$$d_{21}=U_1 V_1 U_2 S_2. \quad (11)$$
$$d_{32}=U_1 V_1 U_2 V_2 U_3 S_3. \quad (12)$$
$$d_{43}=U_1 V_1 U_2 V_2 U_3 V_3 U_4 S_4. \quad (13)$$
$$d_{54} = U_1 V_1 U_2 V_2 U_3 V_3 U_4 V_4 U_5 S_5. \quad (14)$$
$$d_{65}=U_1 V_1 U_2 V_2 U_3 V_3 U_4 V_4 U_5 V_5 U_6 S_6. \quad (15)$$
$$d_{76}=U_1 V_1 U_2 V_2 U_3 V_3 U_4 V_4 U_5 V_5 U_6 V_6 U_7 S_7. \quad (16)$$

The matrices $U_i . V_i$ *and* S_i are given by equations 17-18.

$$U_i = \begin{bmatrix} C\theta_i & -S\theta_i & 0 \\ S\theta_i & C\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (17)$$

$$V_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i \\ 0 & S\alpha_i & C\alpha_i \end{bmatrix} \quad (18)$$

$$S_i = \begin{bmatrix} a_i \\ 0 \\ d_i \end{bmatrix} \quad (18)$$

New position = Transformation matrix x Old position

$$\begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix} = K_7^0 \begin{bmatrix} x_h \\ y_h \\ z_h \\ 1 \end{bmatrix} \quad (19)$$

Home position for the arm (cm): $[86.33 \quad 0 \quad 0 \quad 1]$
From equation (19), the coordinates of the end effector are given by equations (20), (21), and (22).

X= -(997*$C\theta_7 * (S\theta_6(* S\theta_4 * (S\theta_1 * S\theta_3 + C\theta_1 * C\theta_2 * C\theta_3$)+ $C\theta_1 * C\theta_4 * S\theta_2$)- $C\theta_6 * (C\theta_5 * (C\theta_4 * (S\theta_1 * S\theta_3$+ $C\theta_1 * C\theta_2 * C\theta_3$)- $C\theta_1 * S\theta_2$* $S\theta_4$)+ $S\theta_5 * (C\theta_3 * S\theta_1 - C\theta_1 * C\theta_2 * S\theta_3$))))/50-(997* $S\theta_7 * (S\theta_5 * (C\theta_4 * (S\theta_1 * S\theta_3 + C\theta_1 * C\theta_2 * C\theta_3$)- $C\theta_1 * S\theta_2$* $S\theta_4$)- $C\theta_5 * (C\theta_3 * S\theta_1 - C\theta_1 * C\theta_2 * S\theta_3$)))/50. (20)

Y=(997*$C\theta_7 * (S\theta_6(S\theta_4 * (C\theta_1 * S\theta_3 - C\theta_2 * C\theta_3 * S\theta_1$)- $C\theta_4 * S\theta_1 * S\theta_2$)- $C\theta_6 * (C\theta_5 * (C\theta_4 * (C\theta_1 * S\theta_3 - C\theta_2 * C\theta_3$* $S\theta_1$)+ $S\theta_1$* $S\theta_2$* $S\theta_4$)+ $S\theta_5 * (C\theta_1 * C\theta_3 + C\theta_2 * S\theta_1 * S\theta_3$))))/50+ (997* $S\theta_7 * (S\theta_5 * (C\theta_4 * (C\theta_1 * S\theta_3 - C\theta_2 * C\theta_3 * S\theta_1) + S\theta_1$* $S\theta_2$* $S\theta_4$)- $C\theta_5$*($C\theta_1 * C\theta_3 + C\theta_2 * S\theta_1 * S\theta_3$)))/50. (21)

Z= (997* $S\theta_7 * (S\theta_5(C\theta_2 * S\theta_4 + C\theta_3 * C\theta_4 * S\theta_2$)+ $C\theta_5 * S\theta_2 * S\theta_3$))/50- (997*$C\theta_7 * (C\theta_6 * (C\theta_5 * (C\theta_2$* $S\theta_4$+ $C\theta_3 * C\theta_4 * S\theta_2$)- $S\theta_2$* $S\theta_3$* $S\theta_5$)+ $S\theta_6$* ($C\theta_2$* $C\theta_4$- $C\theta_3 * S\theta_2 * S\theta_4$))))/50 (22)

B. Inverse kinematics

Inverse Kinematics (IK) is used to determine the required joints variables of the robotic arm to achieve the specified position and orientation of its end-effector [9]. There are two kinds of solutions for inverse kinematics;

closed form solutions (analytical expression) and Numerical solutions.

The humanoid robot arm with 7 DOF is a redundant robot which has multiple solutions so the numerical solution is selected to solve its inverse kinematics problem. The selected technique depends on error minimization.

A manipulator is solvable if all the sets of joint variables can be found corresponding to a given end-effector location. The necessary conditions are

- Tool point within the workspace, to avoid singularity.
- $n \geq 6$, to have any arbitrary of orientation tool.
- Tool orientation is such that none of the joint limitations are violated [10].

1) Inverse and Pseudo-inverse Jacobian:

In this section, a method for inverse kinematics based on the numerical estimation of the inverse Jacobian at the current pose of the end-effector is introduced. Let the current location is $\vec{r}$ and the target location is $\vec{t}$ as shown in Fig. 6.

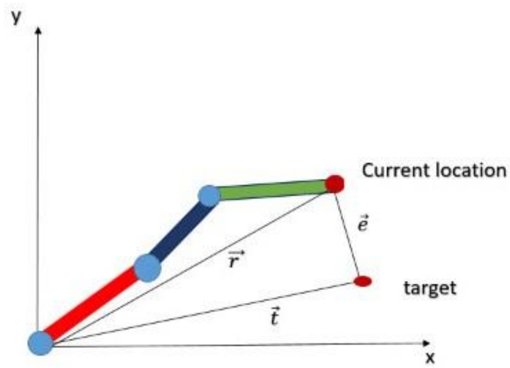

Fig. 6: Basic concept for inverse kinematics

The end-effector position $\vec{r}$ which is obtained from forward kinematics is highly nonlinear function of the joint variables $\vec{q}$. The function $f(\vec{q})$ is linearized at the target position as

$$\vec{f}(\overrightarrow{q_d}) \approx \vec{f}(\overrightarrow{q_0}) + J(\overrightarrow{q_0})(\overrightarrow{q_d} - \overrightarrow{q_0}) . \qquad (23)$$

So the target position can be written as

$$\vec{t} \approx \vec{r}_0 + J(\overrightarrow{q_0})\, \Delta q. \qquad (24)$$

The error between the target and current locations is

$$e = J\Delta\vec{q} . \qquad (25)$$

The change in the joint angles

$$\Delta q = j^{-1} e . \qquad (26)$$

where

$\overrightarrow{r_0} = f(\overrightarrow{q_0})$ current location.

$\vec{t} = f(\overrightarrow{qd})$ target location.

$\vec{e} = \vec{t} - \overrightarrow{r_0}$ error between the two locations.

$J(\overrightarrow{q_0})$ Jacobian Matrix at current location.

Δq: Change in the joint variable

The linearization concept is shown in fig.7.

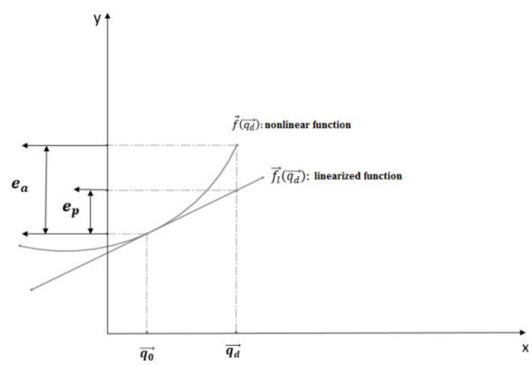

Fig. 7: Linearization of $f(q)$ at certain location

Where:

$\overrightarrow{q_0}$: inetial guess for joint variable.

$\overrightarrow{q_d}$: desired value for joint variable.

e_a:actual error.

e_p: approximate error.

The end-effector position $x_e.y_e.z_e$: is function of 7-joint variables $\theta_1.\theta_2 \dots \theta_7$, the Jacobian matrix can be written as

$$J = \begin{bmatrix} \frac{\delta x_e}{\delta\theta_1} & \frac{\delta x_e}{\delta\theta_2} & \dots & \frac{\delta x_e}{\delta\theta_7} \\ \frac{\delta y_e}{\delta\theta_1} & \frac{\delta y_e}{\delta\theta_2} & \dots & \frac{\delta y_e}{\delta\theta_7} \\ \frac{\delta z_e}{\delta\theta_1} & \frac{\delta z_e}{\delta\theta_2} & \dots & \frac{\delta z_e}{\delta\theta_7} \end{bmatrix} \qquad (27)$$

Since the humanoid robot arm has 7 DOF, Jacobian matrix is rectangular matrix of order 3 X 7. If the matrix is rectangular matrix, its inverse is computed using pseudo inverse. The left inverse $((J^T J)^{-1} J^T)$ is used when the number of rows (m) greater than the number of columns (n). The right inverse $(J^T(J^TJ)^{-1})$) is used when the number of rows (m) less than the number of columns (n) which is the present case.

$J \in R^{m*n}$ m < n rank(J)=m.

There are infinite solutions for Δq, for all possible solutions, the smallest $\Delta\vec{q}$ is selected.

$$\Delta\vec{q} = J^{\dagger}\vec{e}\,. \qquad (28)$$

Where:

$J^{\dagger}$: right pseudo inverse.

2) Algorithm for Solving inverse kinematics:

Using Newton-Raphson method, approximate values for the joint variables are calculated. Assume initial values for the joint variable and calculate the initial current position. Calculate error between the initial position and target position then follow the following procedure.

Step 1: Calculate $\Delta\vec{q} = J^T(JJ^T)^{-1}\vec{e}$

Step 2: update joint variable $\vec{q_c} = \vec{q_c} + \Delta\vec{q}$

Step 3: update $\vec{r}(q_c)$

Step 4: calculate error again $\vec{e} = \vec{t} - \vec{r_0}(\vec{q_c})$.

Step 5: check the relative error, if it is less than 10^{-5} stop, otherwise repeat from step 1 again

$$\varepsilon = \| \vec{e} \| < 10^{-5}$$

The maximum number of iteration is set 1000. If it reaches 1000 iteration, there is no solution for the inverse kinematics.

IV Results and Discussions

A. Results and simulation of forward kinematics

The forward kinematics of 7DOF is calculated using the derived equations in section 3.1.2 and section 3.1.3 and compered with that obtained using Roboanalyzer software and Peter Croke robotic MATLAB toolbox.

The position coordinates of the end effector for 7 cases (Configurations) are presented in Table 2 to Table 8. Also, the robot configuration using Matlab and Roboanalyzer are shown in Fig.8 to Fig.21.

Case 1:

θ_1= 0, θ_2= -90, θ_3= 90, θ_4= 0, θ_5= 0, θ_6= 90, θ_7= 0

TABBLE.II

END EFFECTOR POSITION CORDINATE_CASE1

	DH solution	UV solution	Peter corke	FK using robo analyzer
X	86.33	86.33	86.33	86.33
Y	0	0	0	0
Z	0	0	0	0

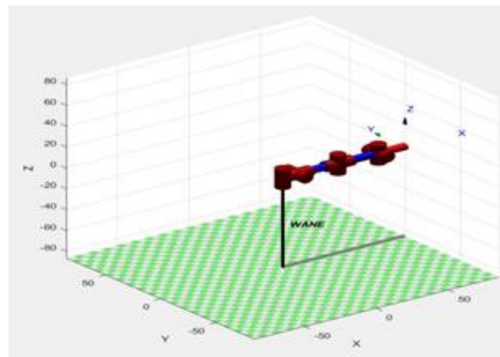

Fig. 8:case1 MATLAB simulation

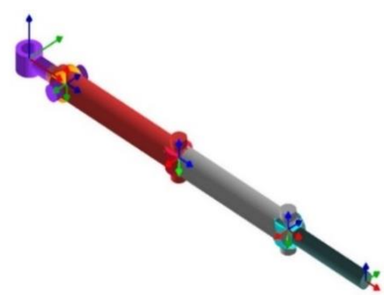

Fig. 9: Case 1 3D model simulation

Case 2:

θ_1= -90, θ_2= -60, θ_3= -30, θ_4= 0, θ_5= 30, θ_6= 60, θ_7= 90

TABBLE.III

END EFFECTOR POSITION CORDINATE_CASE2

	DH solution	UV solution	Peter corke	FK using roboanalyzer
X	-19.94	-19.94	-19.94	-19.94
Y	-58.788	-58.788	-58.788	-58.8489
Z	-28.37	-28.40	-28.370	-28.405

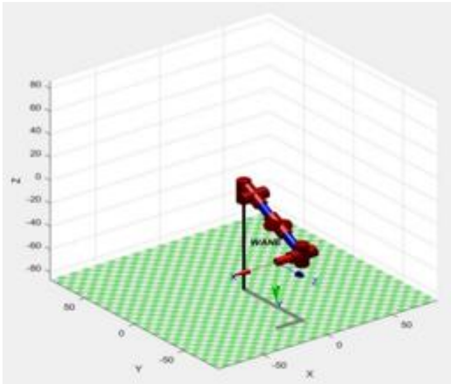

Fig. 10: Case2 Matlab simulation

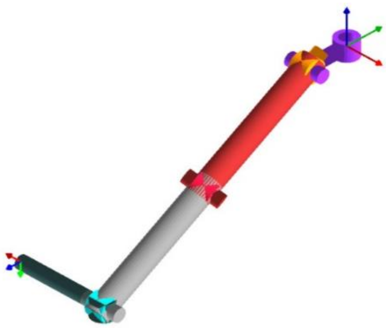

Fig. 11: Case 2 3D model simulation

Case 3:

θ_1= -30, θ_2= 0, θ_3= -60, θ_4= 90, θ_5= 60, θ_6= -90, θ_7 = 30;

TABBLE.IV

END EFFECTOR POSITION CORDINATE_CASE3

	DH solution	UV solution	Peter corke	FK using robo analyzer
X	1.521	1.521	1.521	1.5213
Y	-14.527	-14.527	-14.527	-14.527
Z	-20.065	-20.065	-20.066	-20.135

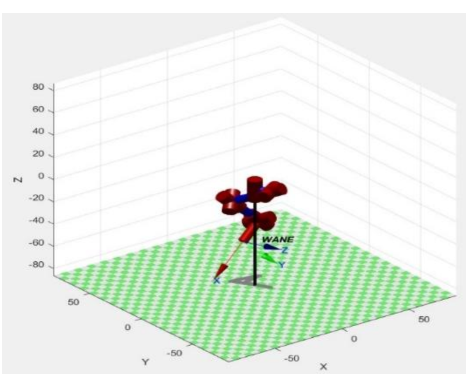

Fig. 12: Case3 Matlab simulation

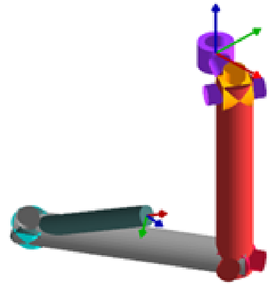

Fig.13: Case 3 3D model simulation

Case 4: θ_1= 30, θ_2= -60, θ_3= 0, θ_4= 90, θ_5= 60, θ_6= -90, θ_7= -30

TABBLE.V

END EFFECTOR POSITION CORDINATE_CASE4

	DH solution	UV solution	Peter corke	FK using robo analyzer
X	29.20	29.25	29.201	29.2537
Y	22.615	22.645	22.615	22.6468
Z	-27.99	-28.03	-27.995	-28.0305

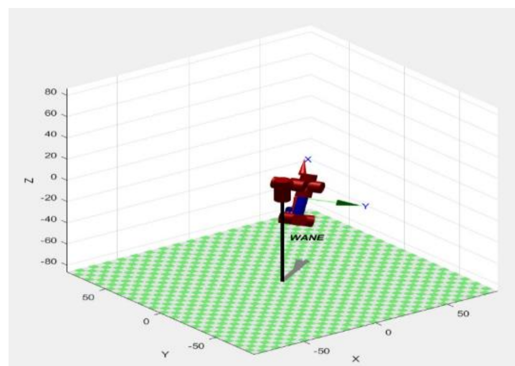

Fig.14: Case4 Matlab simulation

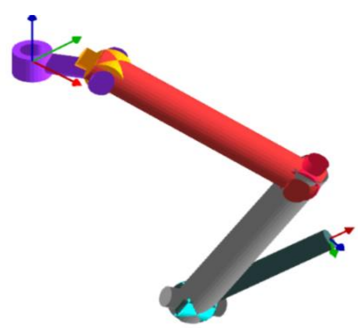

Fig. 15: Case 4 3D model simulation

Case 5: θ_1= 0, θ_2= 90, θ_3= -60, θ_4= 60, θ_5= -90, θ_6= -30, θ_7= 30

TABBLE.VI

END EFFECTOR POSITION CORDINATE_CASE5

	DH solution	UV solution	Peter corke	FK using robo analyzer
X	-37.387	-37.457	-37.287	-37.4517
Y	-2.759	-2.759	-2.760	-2.759
Z	18.861	18.861	18.862	18.8618

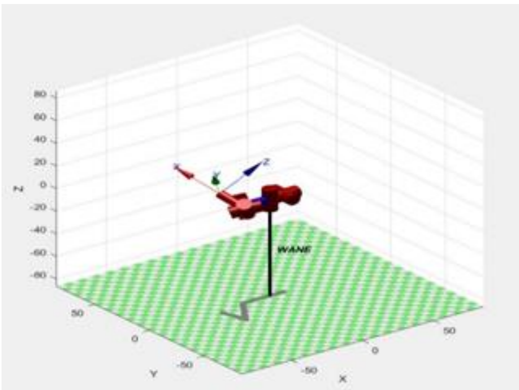

Fig.16: Case5 Matlab simulation

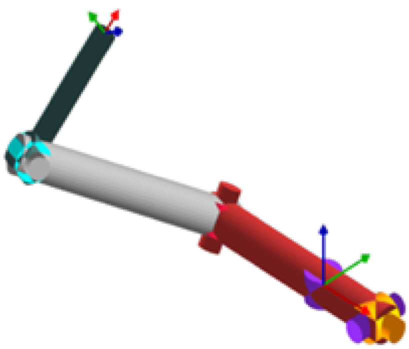

Fig.17: Case 5 3D model simulation

Case 6:

θ_1= 90, θ_2= 60, θ_3= -90, θ_4= -30, θ_5= 30, θ_6= -60, θ_7= 0;

TABBLE.VII

END EFFECTOR POSITION CORDINATE_CASE6

	DH solution	UV solution	Peter corke	FK using robo analyzer
X	-12.863	-12.863	-12.863	-12.8632
Y	-17.052	-17.112	-17.052	-17.1129
Z	-21.172	-21.207	-21.173	-21.207

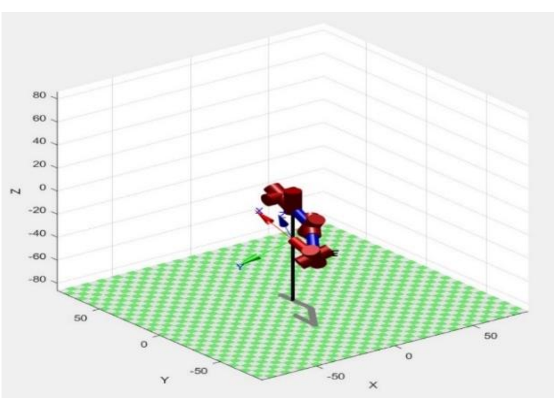

Fig.18: Case6 Matlab simulation

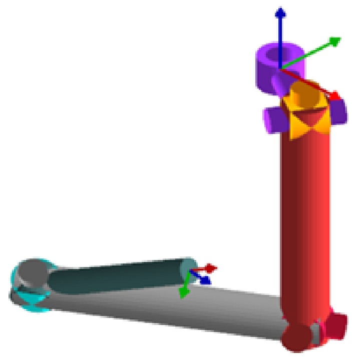

Fig.14: Case 6 3D model simulation

Case 7:

θ_1= -60, θ_2= 60, θ_3= -30, θ_4= -90, θ_5= 90, θ_6= 0, θ_7 = 30;

TABBLE.VIII

END EFFECTOR POSITION CORDINATE_CASE7

	DH solution	UV solution	Peter corke	FK using robo analyzer
X	-4.530	-4.530	-4.530	-4.53
Y	5.9766	5.9766	5.9766	5.976
Z	-47.873	-47.877	-47.843	-47.877

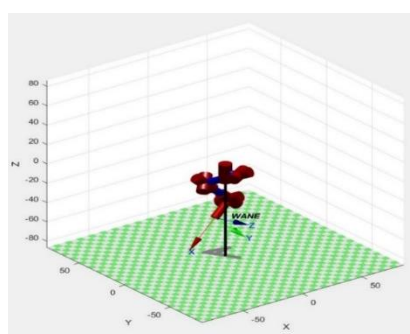

Fig.20: Case7 Matlab simulation

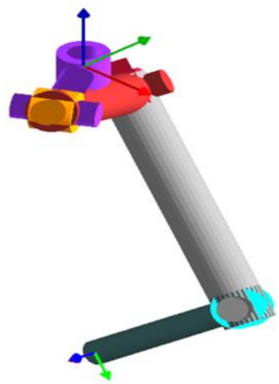

Fig.21: Case 7 3D model simulation

B. Result of inverse kinematics

Inverse kinematics of 7 DoF robot arm discussed in section 3.2 is calculated. The results for five different cases are presented in Table 9

TABBLE.IX

END EFFECTOR POSITION CORDINATE_CASE8

Cases	End-Effector Desired Position (cm)	Joint Angles (degree)	End-Effector Actual Position (cm)	Relative error= $\left\|\frac{Desired - Actual}{Desired}\right\| * 100\%$
1	p_x= -37.457 p_y= -2.759 p_z= 18.861	θ_1=41.1993 θ_2= 81.4092 θ_3= 58.1782 θ_4=44.0594 θ_5=71.1285 θ_6= -4.2767 θ_7= 82.1326	p_x= -37.4570 p_y= -2.7590 p_z= 18.8610	3.761 x 10^{-8} 4.75 x 10^{-7} 2.828 x 10^{-6}
2	p_x= -25.0268 p_y= -1.213 p_z= 7.236	θ_1= -7.1897 θ_2 = 18.0251 θ_3 =-17.6146 θ_4 =106.0547 θ_5 =59.4423 θ_6 =-261.358 θ_7 =-52.1622	p_x= -25.0268 p_y= -1.213 p_z= 7.236	1.189 x 10^{-6} 3.88 x 10^{-7} 9.69 x 10^{-7}
3	p_x= 3.6541 p_y=-19.2130 p_z= 8.2360	θ_1=-1.0431 θ_2= -98.6014 θ_3=13.8213 θ_4=216.9353 θ_5=-23.2666 θ_6= -107.4704 θ_7= 109.7496	p_x= 3.6541 p_y= -19.213 p_z= 8.2360	1.395 x 10^{-9} 3.095 x 10^{-9} 7.346 x 10^{-9}
4	p_x= -6.5897 p_y= 14.213 p_z= 1.236	θ_1=-24.3368 θ_2= 10.9158 θ_3= 1.8683 θ_4=108.8973 θ_5= -87.3327 θ_6= -291.8013 θ_7= -87.3327	p_x= -6.5897 p_y= 14.2130 p_z= 1.2360	4.240 x 10^{-7} 2.350 x 10^{-7} 3.18 x 10^{-8}
5	p_x= 61.3598 p_y= 4.2130 p_z= 13.2360	θ_1=-5.0702 θ_2=-119.9461 θ_3 = 33.7745 θ_4= -4.1809 θ_5=-100.6893 θ_6=-783.2957 θ_7=-66.0657	p_x= 61.3598 p_y= 4.2130 p_z= 13.2360	9.72 x 10^{-9} 1881 x 10^{-8} 1.829 x 10^{-8}

V. CONCLUSIONS

In this paper, forward kinematic (FK) equations have been derived by two techniques the first technique using Denavit and Hartenberg (D-H) transformation matrix and the second using (UV) matrices. The results are compared with that obtained using roboanalyzer software and Peter Croke robotic MATLAB toolbox for different cases. The results indicated that the two techniques were effectively. The Inverse kinematics analysis of 7 DOF robotic arm is solved using Newton-Raphson method iterative technique. The relative error is calculated for different iteration numbers for different cases. Results show that the method is highly convergent and stable. The inverse kinametics results have been simulated by MATLAB. The benefit of this method is find solution for IK problem in the complex cases such as (7DOF) redundant robot.

ACKNOWLEDGMENT

First and foremost, I would like to thank God Almighty for giving me the strength, knowledge, ability and opportunity to undertake this research study and to persevere and complete it satisfactorily. Without his blessings, this achievement would not have been possible. The success of this step required a lot of guidance and assistance from Assoc. Prof. Abdelrady Okasha and I would like to thank him for giving me all support and guidance.

REFERENCES

[1] S. S. Kamlesh and R. Mishra, "Advanced path simulation of a 5R robotic arm for CT Guided Medical Procedures," *Mater. Today Proc.*, vol. 5, no. 2, pp. 6149–6156, 2018.

[2] H. M. Al-Khafaji and M. J. Jweeg, "Solving the inverse kinematic equations of elastic robot arm utilizing neural network," *Al-Khwarizmi Eng. J.*, vol. 13, no. 1, pp. 13–25, 2017.

[3] S. Kucuk and Z. Bingul, *Robot Kinematics: Forward and Inverse Kinematics*, no. December. 2006.

[4] D. Sivasamy, M. D. Anand, and K. A. Sheela, "Robot forward and inverse kinematics research using MATLAB," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2S3, pp. 29–35, 2019.

[5] J. Zhao and N. I. Badler, "Inverse kinematics positioning using nonlinear programming for highly articulated figures," *ACM Trans. Graph.*, vol. 13, no. 4, pp. 313–336, 1994.

[6] T. P. Singh, P. Suresh, and S. Chandan, "Forward and inverse kinematic analysis of robotic manipulators," *Int. Res. J. Eng. Technol.*, vol. 4, no. 2, pp. 1459–1468, 2017.

[7] H. M. Al-Khafaji and M. J. Jweeg, "Solving the inverse kinematic equations of elastic robot arm utilizing neural network," *Al-Khwarizmi Eng. J.*, vol. 13, no. 1, pp. 13–25, 2017

[8] W. I. M. AL-Tameemi and W. M. H. Hadi, “Kinematics analysis of 5250 lab-volt 5-DoF robot arm,” *Eng. Tech. J.*, vol. 32, no. 9 Part A, pp. 2196–2204, 2014.

[9] S. Dereli and R. Köker, “A meta-heuristic proposal for inverse kinematics solution of 7-DOF serial robotic manipulator: quantum behaved particle swarm algorithm,” *Artif. Intell. Rev.*, vol. 53, no. 2, pp. 949–964, 2020.

[10] S. Dereli and R. Köker, “IW-PSO approach to the inverse kinematics problem solution of a 7-DOF serial robot manipulator,” *Sigma J Eng Nat Sci*, vol. 36, no. 1, pp. 77–85, 2018.