Forward and Inverse Kinematic of a Manipulator Simulator Software Using Unity Engine

Handry Khoswanto[1], Kevin Nathanael Sugiharto[2], Iwan Njoto Sandjaja[2] and Thiang[1]
[1]*Electrical Engineering Program, Petra Christian University, Surabaya, Indonesia*.
[2]*Informatics Engineering Program, Petra Christian University, Surabaya, Indonesia*.
handry@petra.ac.id

***Abstract*—Robot manipulator is a technology that is widely used in various applications. Because of the widespread use of robot manipulators, students of Electrical Engineering need to learn about robotics. However, we faced problems that are the price of the robot is relatively expensive and lack of tools to help students for solving mathematical calculations in robotics. The aim of this research is to develop application software for solving the problems that we faced. By using this application software, students can learn about robotics more effective. The application was designed using Unity engine and C# language. The application simulates the movement of robot manipulator either using forward or inverse kinematic equation. The manipulators are limited to 2 and 3 degree of freedoms (DOF). A similar application, RoboAnalyzer, is used for comparing the results of the designed application software. The application has been tested and the results show that mathematical calculation done by the application is approximately same as the result of calculation using mathematic equation. The simulator can simulate the movement of the robot well enough. However, the inverse kinematic calculation can only output one result of all possible results that are resulted by the RoboAnalyzer application.**

***Index Terms*—Forward Kinematic; Inverse Kinematic; Manipulator; Robotic; Simulator.**

I. INTRODUCTION

Robot manipulator is a technology that is widely used in all life applications. Robot manipulator is a mechanical system that consists of links and joints that is able to produce controlled movement in order to move, lift, and manipulate objects. Robot manipulator has been applied widely in the industrial world such as welding process by using welding arm robots [1]. Because of the widespread use of robot manipulators, it is important for students of Electrical Engineering who are interested in robotics, learn about robotics.

Generally and specifically, learning robotics is to learn about the use of the arm robot. This is quite difficult to be done because the movement of robots is calculated based on many numbers of trigonometric equations. Other than that, the price of the robot is relatively high so that it inhibits student to learn about robotics.

Based on these problems, it needs a robot simulation application software that can simulate the movement of robots and this application can support the class of learning robotics so that the student can learn robotics well. This application software must be able to show a simulation of the movement of the robot from the initial axis to a particular destination axis with an animated movement of the arm robot. The animation in this application is designed in 3D to simplify and reduce the cost of learning.

An existing similar application, RoboAnalyzer, is a three-dimensional application that simulates many types of robots. RoboAnalyzer is developed by IIT Dehli, India. This application can simulate robot manipulator with DOF of one to seven. RoboAnalyzer software has two choices of arm types, i.e., rotate and lamar [2].

In this research, a robot simulator software has been designed where the robot can only use two or three rotate arms. Users can see simulation of the movement of robots and the mathematical calculations are displayed in the program so that user is easier to learn about robot manipulator.

II. MATERIAL AND METHODS

A. Simulator Development

In this research, application software for robot manipulator simulation was developed in Windows OS platform. The designed application software can simulate the movement of the robot manipulator in 3D graphic, based on forward and inverse kinematic calculation. This designed application is limited for 2 or 3 DOF robot manipulator with all joint types are revolute (2R or 3R) where the degree of freedom (DOF) determines the possibility of the movement of the mechanism [3]. User can set the length of each link freely. This application can show the forward and inverse kinematic equation.

The designed application was built by using C# programming language. C# is a programming language designed for various applications running on the .NET Framework. C# is a simple, exceptional, safe, and object-oriented language. The various innovations in C# are enabling rapid application development while still maintaining the expression and elegance of the C language style.

For 2D or 3D simulation of movement of the robot manipulator, this application uses Unity engine [4]. Unity engine is a multipurpose game engine that supports 2D and 3D graphics, drag-and-drop functionality and scripting using C#. It is an ultimate game development platform. The unity application comes with an asset store that allows the application developer to have a choice either creating or not creating the user interface component of the designed application. Unity is equipped with good technology that enables programmers to develop high-quality applications. Unity application can support the latest OS technologies like Windows, Mac, iOS, Android, VR, etc. The programming languages used in Unity are C# and javascript [4].

In this application, authors use the Unity engine to draw the robot shape, draw the robot movement, and draw the line trajectory of robot movement.

Testing result of the designed application software is then compared to the result of a similar application, called RoboAnalyzer. RoboAnalyzer is an application-based three-dimensional model that can be used for teaching and learning robotic concepts. It is a product that is developed at Mechatronics Laboratory, Departement of Mechanical Engineering at IIT Delhi, India. Virtual Robot Module which is part of RoboAnalyzer has developed an application where it has joint and Cartesian movements. It was created from a COM server, using one that integrates VRM and Matlab, Ms. Excel, and other applications that have a COM view.[2]

B. Robot Manipulator

Robot manipulator is a robot that is often used in industrial activities. The robot is divided into two sections:

1. **Arm and Body** - arms and body of the robot are used to move and position parts or tool within the workspace of the robot. They can be formed by three joints which are connected by a large link.
2. **Wrist** - This wrist is used to orient parts or equipment in the workspace of the robot. It can consist of two or three compact joints.

Robot manipulator is built by a sequence of links and joints combination. Links are a rigid body that connects the joints or axes. Axes are movable components of the robot that causes relative motion between adjoining links. Mechanical joints that are used to build manipulators consist of five principal types. Two types of the joints are linear, in which the relative motion between adjacent links is non-rotational. Three types of the joints are the rotary type, in which relative motion involves rotation between links. The arm-and-body of the manipulator is built based on one of four configurations. Each structure of manipulator provides different workspace and is suitable for different applications. The four configurations are:

1. **Gantry** - This robot configuration has linear joints which are installed overhead. They are also called as a Cartesian and square robot.
2. **Cylindrical** – This robot configuration is named Cylindrical because of the shape of the workspace. A cylindrical robot is built by using a linear/prismatic joint which is connected base using a revolute joint.
3. **Polar**- The base joint of a polar robot allows for twisting and the joints are a combination of rotary and linear types. The workspace created by this configuration is spherical.
4. **Joined-Arm** - This is the most popular configuration of industrial robots. The arm connects with a twisting joint, and the links within it are connected with rotary joints. It is also called an articulated robot [5].

The configuration type of robot manipulator used in this research is joined arm manipulator and it is limited to 2 or 3 DOF manipulator. Figure 1 and 2 show the robot modeling in 2 DOF and 3 DOF coordinate system respectively.

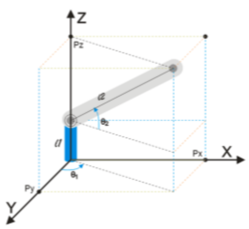

Figure 1: Robot model in 2 DOF coordinate system

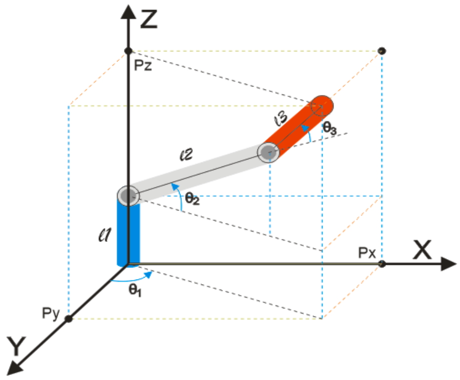

Figure 2: Robot model in 3 DOF coordinate system

C. Forward and Inverse Kinematic

Kinematics is the branch of classical mechanics that describes the motion of points, bodies (objects), and systems of bodies (groups of objects) without considering the mass of each of the forces that caused the motion [6]. Kinematics consists of two branches, i.e., forward kinematic and inverse kinematic. Both forward and inverse kinematics are closely related because of the mutually related calculations between the joint positions and the absolute value of the Cartesian coordinate frame. The relationship between the both forward and inverse kinematics can be illustrated in Figure 3.

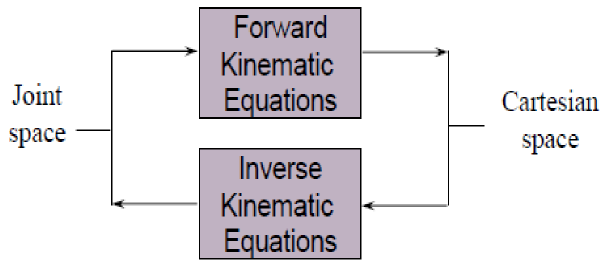

Figure 3: Relationship between forward and inverse kinematic

In this application, both methods, forward and inverse kinematic are used to calculate the position of the robot manipulator for drawing movement for the robot.

a. Forward Kinematic

Forward kinematic is the static geometrical problem of computing the position and orientation of the End Effector of the manipulator for a given set of joint angles. Equations 1, 2, and 3 are a forward kinematic equation for a 2 DOF manipulator. These equations were derived based on 2 DOF robot model shown in Figure 1.

$$P_x = \mathrm{Sin}(\theta_1) * \left(L_2 * Cos(\theta_2)\right) \quad (1)$$

$$P_y = \mathrm{Cos}(\theta_1) * \left(L_2 * Cos(\theta_2)\right) \quad (2)$$

$$P_z = L_1 + L_2 * Sin(\theta_2) \quad (3)$$

where: P_x, P_y, P_z = position x, y, z of the end effector of manipulator
θ_1 , θ_2 = angle of joint 1 and 2
L_1, L_2 = length of link/arm 1 and 2

Equations 4, 5, and 6 are a forward kinematic equation for a 3 DOF manipulator. These equations were derived based on 3 DOF robot model shown in Figure 2 [7].

$$P_x = \mathrm{Sin}(\theta_1) * \left(L_2 * Cos(\theta_2) + L_3 * Cos(\theta_2 + \theta_3)\right) \quad (4)$$

$$P_y = \mathrm{Cos}(\theta_1) * \left(L_2 * Cos(\theta_2) + L_3 * Cos(\theta_2 + \theta_3)\right) \quad (5)$$

$$P_z = L_1 + L_2 * Sin(\theta_2) + L_3 * Sin(\theta_2 + \theta_3) \quad (6)$$

where: P_x, P_y, P_z = position x, y, z of the end effector of manipulator
$\theta_1, \theta_2, \theta_3$ = angle of joint 1, 2, and 3
L_1, L_2, L_3 = length of link/arm 1, 2, and 3

b. Inverse Kinematic

Inverse kinematics is the static geometrical problem of calculating all possible sets of joint angles that could be used to attain the given position and orientation of End Effector. Based on 2 DOF robot model shown in Figure 1, inverse kinematic equations were derived and shown in Equation 7 and 8.

$$\theta_1 = Atan2(P_y, P_x) \quad (7)$$

$$\theta_2 = \mathrm{Asin}\left(\frac{P_z - L_1}{L_2}\right) \quad (8)$$

where: P_x, P_y, P_z = position x, y, z of the end effector of manipulator
θ_1 , θ_2 = angle of joint 1 and 2
L_1, L_2 = length of link/arm 1 and 2

Equation 9 to 13 are an inverse kinematic equation for a 3 DOF manipulator. These equations were derived based on 3 DOF robot model shown in Figure 2 [7].

$$r = \pm\sqrt{(P_x^2 + P_y^2)} \quad (9)$$

$$D = \pm\sqrt{((P_z - l_1)^2 - r^2)} \quad (10)$$

$$\theta_1 = Atan2(P_y, P_x) \quad (11)$$

$$\theta_3 = \mathrm{Acos}\left(\frac{D^2 - L_2^2 - L_3^2}{2L_2L_3}\right) \quad (12)$$

$$\theta_2 = Atan2(r, P_z - L_1) - Atan2(L_2 + L_3 * Cos\theta_3, L_3 * Sin\theta_3) \quad (13)$$

where: P_x, P_y, P_z = position x, y, z of the end effector of manipulator
$\theta_1, \theta_2, \theta_3$ = angle of joint 1, 2, and 3
L_1, L_2, L_3 = length of link/arm 1, 2, and 3

III. Analysis and Design

The simulator application has 3 DOF options. Generally, there are five steps for using the designed application. The first step is to select the desired DOF of the robot. In the designed application, DOF is only limited to two or three DOF.

The second step is to input the desired length of the link/arm according to the desired DOF. The length of the link/arm in the simulator is in centimeter unit.

The third step is to input the angle of each joint/DOF and number of the counter-steps.

The fourth step is that the program will do the calculation based on the angle input of each joint and then move the robot and generate the trajectory, and display the calculation result.

The fifth step is that the simulator movement stops after the robot reaches its destination angle.

IV. Results and Discussion

The designed application software has been tested and it can illustrate the forward and inverse kinematic. Here is an example of the forward kinematic testing results of two DOF robot using the designed application software and comparison software, RoboAnalyzer. Table 1 shows the calculation results. Figure 4 and 5 show the results using RoboAnalyzer and the designed Application software respectively.

Table 1
Forward Kinematic Calculation Result of 2 DOF Robot

Angle 1	Angle 2	X	Y	Z
90	0	3.000	0.000	5.000

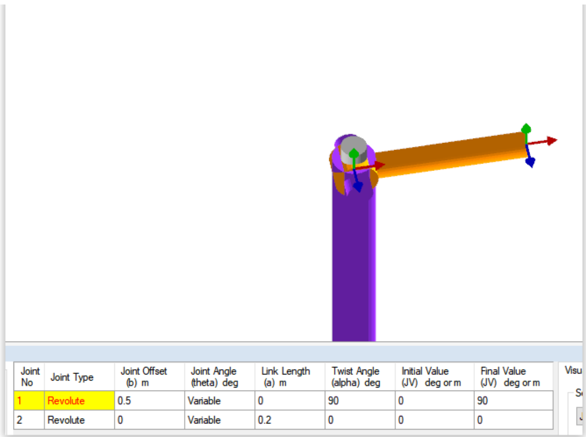

Figure 4: Forward kinematic result of 2 DOF robot using RoboAnalyzer

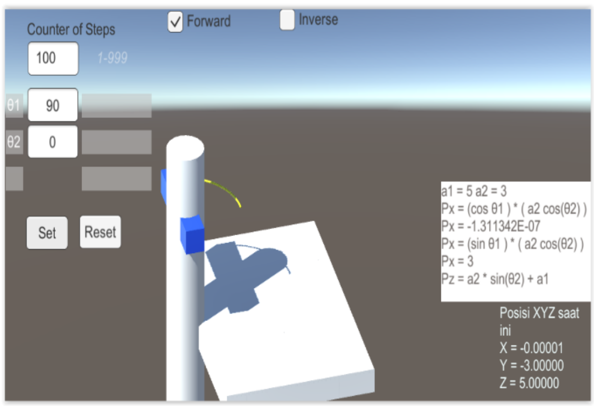

Figure 5: Forward kinematic result of 2 DOF robot using designed application software

Table 2 shows the experimental result of inverse kinematic calculation of 2 DOF robot and Figure 6 shows its simulation results.

Table 2
Experimental Result of Inverse Kinematic Calculation of 2 DOF Robot

Angle 1	Angle 2	X	Y	Z
90	0	3.000	0.000	5.000

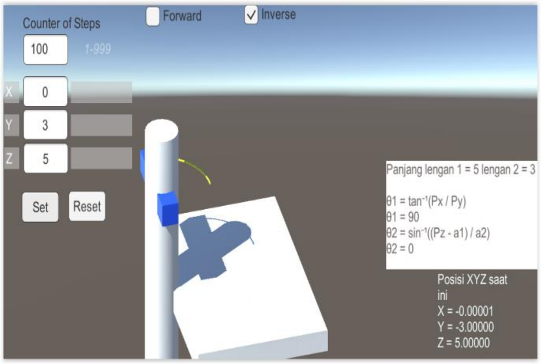

Figure 6: Simulation result of inverse kinematic calculation of 2 DOF robot

Next, here is an example of the experimental results of 3 DOF robot. Experiments were done using forward and inverse kinematic methods. Table 3 shows calculation result of forward kinematic. Figure 7 and figure 8 show the simulation results of 3 DOF robot using RoboAnalyzer and the designed application software.

Table 3
Forward Kinematic Calculation Result of 3 DOF Robot

Angle 1	Angle 2	X	Y	Z
90	90	0.000	0.000	9.000

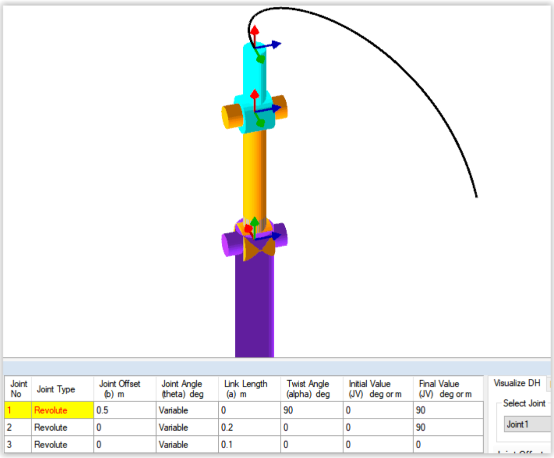

Figure 7: Forward kinematic simulation result of 3 DOF robot using RoboAnalyzer

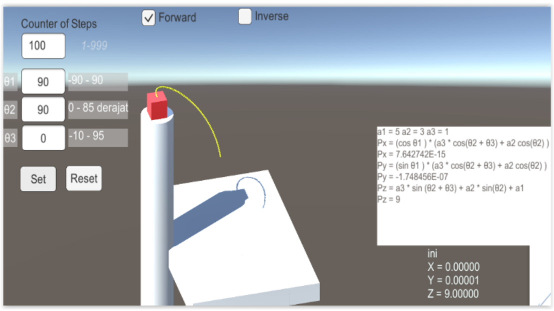

Figure 8: Forward kinematic simulation result of 3 DOF robot using RoboAnalyzer

Table 4 shows the inverse kinematic calculation result of 3 DOF robot using the designed application and figure 9 shows the inverse kinematic simulation result of 3 DOF robot using the designed application.

Table 4
Inverse Kinematic Calculation Result of 3 DOF robot

Angle 1	Angle 2	X	Y	Z
90	0	3.000	0.000	5.000

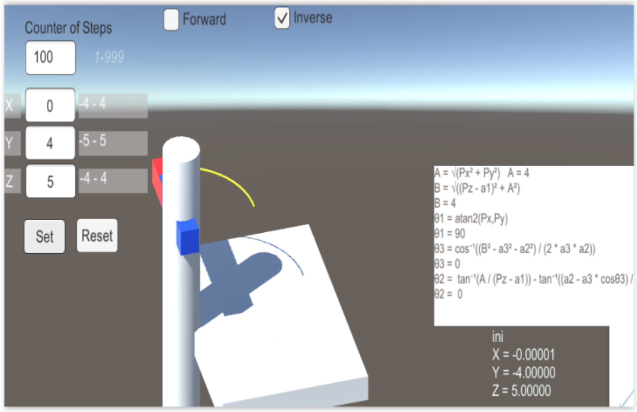

Figure 9: Inverse kinematic simulation result of 3 DOF robot using the designed application

V. Conclusion

Based on the testing results, it can be concluded that the designed application software can do the forward and inverse kinematic calculation, simulate the movement of the robot. Simulation of movement of the robot's end-effector produces trajectory line. All inverse kinematic results of the designed application, either 2 DOF or 3 DOF can only result in one answer of some possible answers, while RoboAnalyzer can result in two to four possible calculation answers. In the designed application, rotation view simulator is not provided on Android OS. For further development, the application can be improved so that the robot looks like the actual or real robot. The application also can be developed so that it can give results not only one answer but also all other possible answers of inverse kinematics calculation. The application can also be developed for running on Android OS.

References

[1] Cucu. *Robot manipulator*, 2012. Retrieved November 12, 2016 from http://www.robotic-explorer-bandung.com/artman2/publish/in-house-training/Robot_Manipulator.shtml
[2] Roboanalyzer. *RoboAnalyzer*. Retrieved March 12, 2017 from http://www.roboanalyzer.com.
[3] Naveenagrawal. .*Degree of Freedom*, 2009. Retrieved Februari 26, 2017 from http://www.brighthubengineering.com/machine-design/6634-kinematics-design-of-mechanisms-degrees-of-freedom/.
[4] Unity. *Unity*, 2016. Retrieved November 10, 2016 from https://www.unity3d.com.
[5] RobotWorx. *What is robot manipulator*, 2016. Retrieved November 10, 2016 from https://www.robots.com/faq/show/what-is-a-robot-manipulator.
[6] KBBI. *Kamus Besar Bahasa Indonesia (KBBI)*.Retrieved Juli 26, 2017 from http://kbbi.web.id/kinematika.
[7] Sadegh, Hossein & Zarabadipour, Hassan. *Modeling, Simulation and Position Control of 3DOF Articulated Manipulator*. Indonesian Journal of Electrical Engineering and Informatics, 2014. 2. 132-140. 10.11591/IJEEI.v2i3.119.