Interactive Kinematics Simulator for the 5 DOF Virtual Industrial Robot

Hwa Jen YAP†
Department of Mechanical Engineering
Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia
Tel: +603-7967-5240, Email: hjyap737@um.edu.my / jyap737@yahoo.com

Shirley THO
Department of Mechanical Engineering
Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia
Email: shirley524@gmail.com

Siow Wee CHANG
Bioinformatics Program
Institute of Biological Sciences, Faculty of Science, University of Malaya, 50603 Kuala Lumpur, Malaysia.
Email: siowwee@um.edu.my

Zahari TAHA
Innovative Manufacturing, Mechatronics and Sports Lab
Faculty of Manufacturing Engineering, University Malaysia Pahang, 26600 Pekan, Pahang, Malaysia
Email: zaharitaha@ump.edu.my / ztr.motion@gmail.com

Abstract. Virtual industrial robot kinematics simulator provides the user with an interactive 3D graphical representation of manipulator. The virtual simulator allows users to simulate the movement and position of the robot arm before operating the actual robot. This paper describes the control of a virtual robot in virtual environment. The system is designed to provide users with an interative and easy way to control the robot manipulator without the requirement of programming skills. The normal computer screen was set to be touch-screen by using infrared LED pen and Wiimote as the input device. Users can draw the robot arm by moving the infrared LED pen on screen and the Wiimote will receive the signal of the pen's movement. In this system, users will be requested to define the link and joint parameters by drawing the 5 DOF robot model on screen. Following this, inverse kinematics is automatically calculated in real time when the user inputs the final position of the end effector. The results of inverse kinematics will be shown in 3D robot model. This system allows users to design a robot arm easily and visualize its performance before building a real robot arm.

Keywords: interactive; degree-of-freedom (ODF); inverse kinematics; virtual robot

1. INTRODUCTION

In general, robots can be categorized into two types: service robots and industrial robots. Service robots are referred as robots which are free to move with a certain intelligence level. These robots include the wheels-based mobile robot, unmanned-aerial vehicle (UAV), humanoid robot and so on. Conversely, industrial robots are referred as machines which are constructed similar to human arm and located at a fixed point.

According to the definition by International Standards Organization, an industrial robot is an automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes (ISO, 1994). Industrial robots are mostly used in the manufacturing environment, which the tasks are carried out repeatedly until overwritten with a new program to perform new tasks (Groover, 1986). The typical operations for industrial robots are welding, material handling, assembly etc. Nowadays, it can be used in cooperative surgical system (Kasi et al., 2014).

In industrial robotics, the study of inverse kinematic involves determining the set of joint variables to achieve a desired final position of the tool frame. There are two main ways to find the angles of each joint. The first method uses a graphical method to solve the problem. Normally, the robot arms will be drawn and the relationship between joint angles and robot arm parameters will be determined using trigonometry equations. The second method uses a formula to determine the joint axis geometry. The equations for joint angles involve the relationship between individual joints of the robot manipulator and the position and orientation of end effector. This method normally involves a set of nonlinear coupled algebraic equations. There are no general algorithms which may be employed to solve the equations. The equations exist for special manipulator geometries, for instance, decoupled manipulators and more generally when the degree of the characteristic polynomial is less or equal to four (Mavroidis, 1994).

There are many methods used to generate the equations for robot joint angles. Denavit-Hartenberg (D-H) is the simplest method to solve the inverse kinematics problem (Devanit and Hartenberg, 1995). Initially, a coordinate frame for all links with respect to other links is assigned. Following this, the link parameters table is built and a 4x4 homogeneous transformation matrix is derived to describe the position and orientation of each link relative to its neighbouring link. The forward kinematics matrix of the robot manipulator is then retrieved and the inverse kinematics equations are solved.

An overview of robotics simulation shows that the simulation can be applied to kinematics and dynamics of industrial robots (Žlajpah, 2008). In 1998, there was a robot interface which incorporates virtual reality and allows the manipulation of any type of robot structure, using an automatic kinematic generator (Fluckiger, 1998). The robots are described by using Kleinfinger notation and tree structure in a text-based Robot Arm File Format (RAFF) by specifying a series of joints, their parameters and their hierarchy. Furthermore, web-based robot simulators were also developed by researchers (Rohrmeier, 2000; VRL, 2002). VR-based robot programming and simulation is developed to teach the robot path and generate robot commands automatically (Yap et al., 2008; Yap et al., 2014; Tahriri et al., 2015). Augmented Reality (AR) based industrial robot is used in the modular-based automated factory operation (Pai et al., 2016). Dynamic simulation using new techniques are also presented, such as Fuzzy Adaptive Differential Equation (FADE) (Al-Dabbagh et al., 2014). On the other hand, the axiomatic design principles in analysing the ergonomics design parameter of a virtual environment being discussed (Taha et al., 2014). However, the kinematics for a specific robot needs to be obtained before the robot path can be simulated.

2.PROBLEM STATEMENTS AND OBJECTIVES

Most of the programs in the current market are capable of solving forward kinematics only. Although some programs are capable of solving inverse kinematics problems, the solutions are limited to specific robot arms. An operator is required to manually calculate the robot joint angles. The robot inverse kinematic equations are complicated and are time-intensive to be solved manually.

The mouse is the most common input device for personal computers and it provides rapid and exact pointing. However, the mouse is not considered as a natural and intuitive way to interact with graphical user interface (GUI) because the cursor is moved in *xy*-plane whereas the mouse is moved in *xz*-plane. For example, the user needs to move the mouse forward (*z*-axis) in order to move the cursor on the screen upward (*y*-axis). Therefore, users may be confused when they attempt to use a mouse for the first time. In ergonomics studies, the best design is making the control and display in the same direction. This means that the control must move upward in order to move the cursor on screen upward, rather than in the forward direction. Furthermore, control interfaces are complicated and require skilled operators to perform an operation. The use of mouse or controller as an input device is not an intuitive way for users, particularly novices and elderly. The operator needs to press the +/- button to control the movement of link/angle of joint (VRL, 2002). Therefore, the ultimate goal is to improve the human-computer interface while developing virtual simulator for an industrial robot. The developed system should provide the user with a natural, user-friendly and interactive way to design the robot arm without requiring robotic skills. On top of these, the system should be able to generate the robot kinematics automatically and simulate the robot arm movement in 3D graphical simulation.

3. SYSTEM ARCHITECTURE

In this research, infrared pen is used as the input device or 3D digitizer. The pen is used to select menus, input the parameters, and interact with virtual objects and so on. Figure 1 shows the system flow of the design plane of the project. In the beginning, users have to define the type of joints through the control interface and define the length of link using the infrared pen. This step will be continued until users convert the design plane into the task performance plane. The link and joint parameters defined by user will be used to build the virtual robot model. In the task performance plane, users have to define the position of the end effector. The input data will be transmitted into the program and executed. The link parameter table will be shown on the screen. If the position of the end effector is out of the working range, the program will show a warning message. Otherwise the robot arm's position will be simulated.

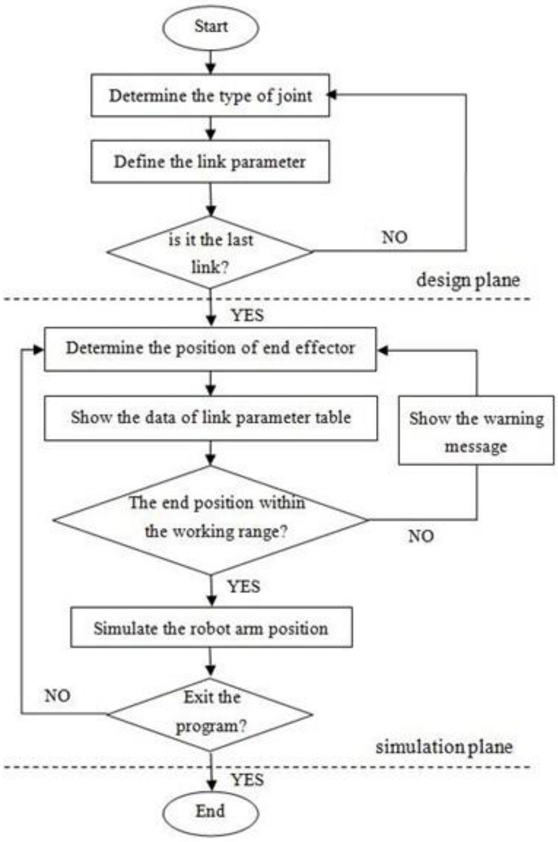

Start
Determine the type of joint
Define the link parameter
is it the last link?
NO
design plane
YES
Determine the position of end effector
Show the data of link parameter table
Show the warning message
The end position within the working range?
NO
YES
Simulate the robot arm position
NO
Exit the program?
simulation plane
YES
End

Figure 1: System flow of design plane of the project

Tests are carried out to locate the optimum distance of the infrared camera from the screen, whereby the monitor screen should be placed within the detection volume of the camera (Johnny, 2010). It is found that the minimum distance from the infrared camera to the computer monitor screen and projection screen is 580 mm and 3150 mm, respectively. Moreover, the maximum distance is 5000 mm for both cases.

In general, there is no optimum location for the infrared camera. The camera can be placed at any location as long as it is within the minimum and maximum distance. However, farther distance will increase the probability of disturbances by other infrared sources outside the screen, as well as obstacles between infrared camera and screen. Therefore, in this research, the camera is located at a minimum distance, with the angle focused at the centre of screen. The performance of the system under lit conditions is the same as in dark conditions. The lighting will not affect the accuracy and sensitivity because the system detects the infrared signal, rather than a normal light wave.

4. KINEMATIC MODELLING

The robot kinematics showed in this research is inverse kinematics. For inverse kinematics, a set of equations are developed, which relate to a particular configuration of the robot. Normally, the robot configuration is referred as the link and joint parameters, which are defined by the user. However, in this research, the robot configuration is referred as a 5 degree-of-freedom robot arm. The Denavit-Hartenberg (D-H) and trigonometry methods are used to find the joint angles in order to reach the desired end effector's position

4.1 Denavit-Hartenberg (D-H) method

There are no general solutions or equations for the D-H method. Therefore, for general cases, generic steps are used to solve inverse kinematics using the Denavit-Hartenberg (D-H) method:

- Step-1: Assign coordinate frames to all links with respect to other links
- Step-2: Build the link parameter table
- Step-3: Generate homogeneous transformation matrix
- Step-4: Obtain the forward kinematic matrix
- Step-5: Solve the inverse kinematics problems to obtain the rotation angles.

Thus, the generic equation for a 5-DOF robotic arm can be written as:

$$\begin{aligned} P_x &= f(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5) \\ P_y &= f(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5) \\ P_z &= f(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5) \end{aligned} \tag{1}$$

In this paper, the coordinate frame for 5-DOF robot arm can be represented in Figure 2. The transformation matrixes for all 5-joints are developed accordingly.

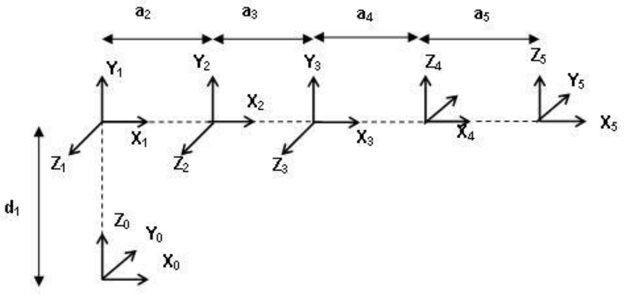


Figure 2: Assigned coordinate frame for 5-DOF robot arm

$$\text{Joint 1}: A_1 = A_0^1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(2a)}$$

$$\text{Joint 2}: A_2 = A_1^2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(2b)}$$

$$\text{Joint 3}: A_3 = A_2^3 = \begin{bmatrix} C_3 & -S_3 & 0 & a_3C_3 \\ S_3 & C_3 & 0 & a_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(2c)}$$

$$\text{Joint 4}: A_4 = A_3^4 = \begin{bmatrix} C_4 & 0 & -S_4 & a_4C_4 \\ S_4 & 0 & C_4 & a_4S_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(2d)}$$

$$\text{Joint 5}: A_5 = A_4^5 = \begin{bmatrix} C_5 & -S_5 & 0 & a_5C_5 \\ S_5 & C_5 & 0 & a_5S_5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(2e)}$$

where $S_i = \sin\theta_i$ and $C_i = \cos\theta_i$

The total transformation can be obtained through the matrix multiplication of these matrixes equation (2a - 2e).

$$A_0^5 = A_1A_2A_3A_4A_5 = \begin{bmatrix} a11 & a12 & a13 & Px \\ a21 & a22 & a23 & Py \\ a31 & a32 & a33 & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Therefore,

$$\begin{aligned} P_x &= [C_1 \times C_{23} \times C_4 \times (a_4 + (a_5 \times C_5))] - [C_1 \times S_{23} \times S_4 \times (a_4 + (a_5 \times C_5))] \\ &\quad - (a_5 \times S_1 \times S_5) + [C_1 \times ((a_2 \times C_2) + (a_3 \times C_{23}))] \\ P_y &= [S_1 \times C_{23} \times C_4 \times (a_4 + (a_5 \times C_5))] - [S_1 \times S_{23} \times S_4 \times (a_4 + (a_5 \times C_5))] \quad (4) \\ &\quad + (a_5 \times C_1 \times S_5) + [S_1 \times ((a_2 \times C_2) + (a_3 \times C_{23}))] \\ P_z &= [S_{23} \times C_4 \times (a_4 + (a_5 \times C_5))] + [C_{23} \times S_4 \times (a_4 + (a_5 \times C_5))] \\ &\quad + (a_2 \times S_2) + (a_3 \times S_{23}) + a_1 \end{aligned}$$

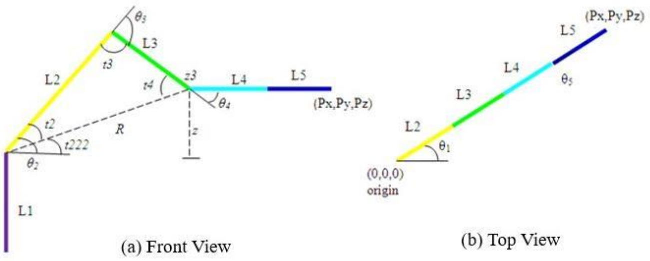
The solutions above can be further simplified as following:

$$\begin{aligned} S_{23} &= S(\theta_2 + \theta_3) = S\theta_2C\theta_3 + C\theta_2S\theta_3 \\ C_{23} &= C(\theta_2 + \theta_3) = C\theta_2C\theta_3 - S\theta_2S\theta_3 \end{aligned} \quad (5)$$

4.2 Trigonometry Method

There is no general equations suitable for all cases, as each robot model has its own inverse kinematics equations. Not all robot models can be solved by the D-H method. In this research, a 5-DOF robot model is selected for testing purposes. The inverse kinematics equations for this model are generated by the trigonometry method.

4.2.1 Clamping Task



(a) Front View

(b) Top View

Figure 3: Robot model during clamping

Figure 3 shows the front/top view of the robot model during clamping, which the end effector (L5) is always faced forward.

(a) For the location of end effector when $Px \geq 0$

$$\begin{aligned} x3 &= Px - (L4 + L5) \times \cos(\theta_1) \\ y3 &= Py - (L4 + L5) \times \sin(\theta_1) \\ z &= Pz - L1 \\ R &= \sqrt{(x3)^2 + (y3)^2 + (z)^2} \\ t3 &= \cos^{-1}\left(\frac{L2 \times L2 + L3 \times L3 - R \times R}{2 \times L2 \times L3}\right) \\ \theta_3 &= t3 - 180 \\ \theta_5 &= 0 \\ t2 &= \cos^{-1}\left(\frac{L2 \times L2 + R \times R - L3 \times L3}{2 \times R \times L2}\right) \\ t222 &= \sin^{-1}\left(\frac{z}{R}\right) \\ t4 &= 180 - (t2 + t3) \end{aligned} \quad (6)$$

If the x-coordinate of joint-3 been calculated is larger than 0,

$$\begin{aligned} \theta_2 &= t2 + t222 \\ \theta_4 &= t4 - t222 \end{aligned} \quad (7)$$

else,

$$\begin{aligned} \theta_2 &= 180 - t222 + t2 \\ \theta_4 &= t4 - (180 - t222) \end{aligned} \quad (8)$$

(b) For the location of end effector when $Px < 0$

$$x3 = Px + (L4 + L5) \times \cos(\theta_1)$$
$$y3 = Py + (L4 + L5) \times \sin(\theta_1)$$
$$z = Pz - L1$$
$$R = \sqrt{(x3)^2 + (y3)^2 + (z)^2}$$
$$t3 = \cos^{-1}\left(\frac{L2 \times L2 + L3 \times L3 - R \times R}{2 \times L2 \times L3}\right)$$
$$\theta_3 = 180 - t3 \tag{9}$$
$$\theta_5 = 0$$
$$t2 = \cos^{-1}\left(\frac{L2 \times L2 + R \times R - L3 \times L3}{2 \times R \times L2}\right)$$
$$t222 = \sin^{-1}\left(\frac{z}{R}\right)$$
$$t4 = 180 - (t2 + t3)$$

If the *x*-coordinate of joint-3 been calculated is larger than 0,

$$\begin{aligned} \theta_2 &= 180 - (t2 + t222) \\ \theta_4 &= t222 - t4 \end{aligned} \tag{10}$$

else,

$$\begin{aligned} \theta_2 &= 90 - (t2 + t222) \\ \theta_4 &= 90 + t222 - t4 \end{aligned} \tag{11}$$

4.2.2 Drilling Task

Figure 4 shows the front/top view of the robot model during drilling, and now the end effector (L5) is always faced downward. The mathematic models for this solution:

$$\theta_1 = \tan^{-1}\left(\frac{Py}{Px}\right)$$
$$x3 = Px$$
$$y3 = Py$$
$$z3 = Pz + (L4 + L5)$$
$$z = z3 - L1$$
$$R = \sqrt{(x3)^2 + (y3)^2 + (z)^2} \tag{12}$$
$$\theta_5 = 0$$
$$t3 = \cos^{-1}\left(\frac{L2 \times L2 + L3 \times L3 - R \times R}{2 \times L2 \times L3}\right)$$
$$t2 = \cos^{-1}\left(\frac{L2 \times L2 + R \times R - L3 \times L3}{2 \times R \times L2}\right)$$
$$t222 = \sin^{-1}\left(\frac{z}{R}\right)$$
$$t4 = 180 - (t2 + t3)$$

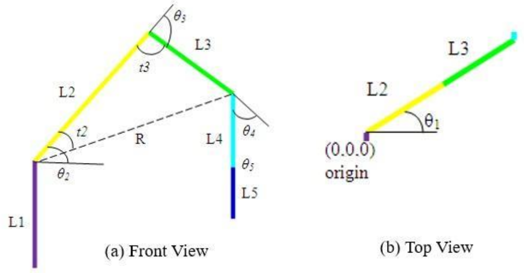


Figure 4: Robot model during drilling

For the location of end effector when $Px \geq 0$

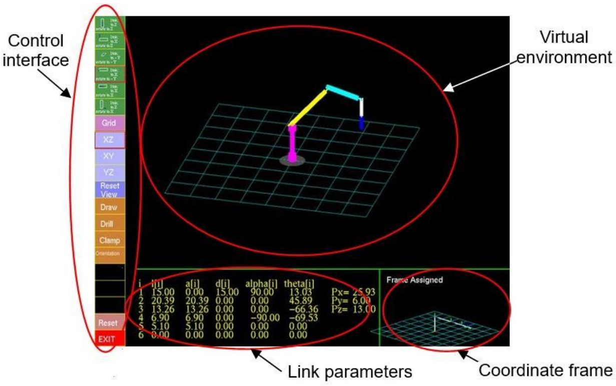
$$\begin{aligned} \theta_2 &= t2 + t222 \\ \theta_3 &= t3 - 180 \\ \theta_4 &= (t4 - t222) - 90 \end{aligned} \tag{13}$$

else,

$$\begin{aligned} \theta_2 &= 180 - t3 \\ \theta_3 &= 180 - (t2 + t222) \\ \theta_4 &= 90 - (t4 - t222) \end{aligned} \tag{14}$$

5. USER CONTROL INTERFACE

Figure 5 shows the user control interface for the developed virtual robot system. The control interface is designed to be more user-friendly, in order to suit novice users. It is divided into four sections:



```
i  l[i]   a[i]   d[i]   alpha[i] theta[i]
1  15.00  0.00   15.00  90.00    13.03     Px= 25.93
2  20.39  20.39  0.00   0.00     45.89     Py= 6.00
3  13.26  13.26  0.00   0.00     -66.36    Pz= 13.00
4  6.90   6.90   0.00   -90.00   -69.53
5  5.10   5.10   0.00   0.00     0.00
6  0.00   0.00   0.00   0.00     0.00
```

Figure 5: Interface of the program

(a) Control interface

The control interface is located at the left-hand side of the program, and is used to determine the type of robot link/joint, viewing plane, tasks and other basic control buttons. The user can select the type of joint which is able to rotate about a certain axis. Therefore, the frame assigned can be directly used in inverse kinematics calculation. Figure 6 shows the example robot model & its type of joint is listed in Table-1.

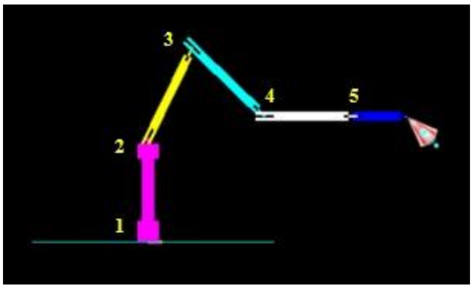
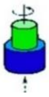
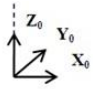
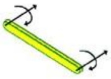
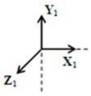
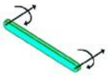
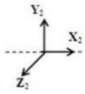
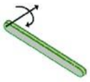
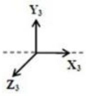
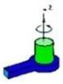
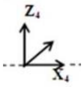


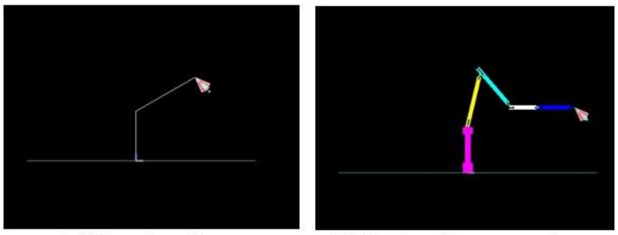
Figure 6: Example of Robot Model

Table 1. Robot joint and frame

Joint	Type of Joint	Frame Assigned	Icon
1			link: in Z rotate in Z
2			link: in X rotate in - Y
3			link: in X rotate in - Y
4			link: in X rotate in - Y
5			link: in X rotate in Z

(b) Virtual Environment – Drawing/Task Performance

When the interface functions as a drawing plane (Figure 7a), the user is requested to define the link and joint parameters by drawing. After the last parameter is defined, the drawing plane will transform into the task performance plane. The users are requested to define the position of the end effector. The program will automatically solve the inverse kinematics problems and simulate the final position of the robot model, as shown in Figure 7b.



(a) Drawing plane: define link and joint parameters

(b) Task performance plane: performance clamping process

Figure 7: Virtual environment

(c) Link Parameters

The bottom section of the interface shows the link parameters. The first column shows the link number for the robot model. The second through fifth columns display the values of the link parameters. The *Px*, *Py* and *Pz* values show the coordinate of the end effector's final position. If there are parameters which are out of the range, the colour of the text will change from yellow to red as a warning signal to the user.

(d) Coordinate Frame

The coordinate frames assigned are based on some basic rules. The blue line represents the *z*-axis of the joint, which is the rotating axis. The red and green lines represent the *x*-axis and *y*-axis of the joint, respectively.

6. RESULTS

The results are verified to ensure that the generated data is accurate according to the developed D-H and trigonometry methods. The verification process is performed by replacing the displayed values into mathematical model (forward kinematics), using Laptop based configuration as shown in Figure 8.

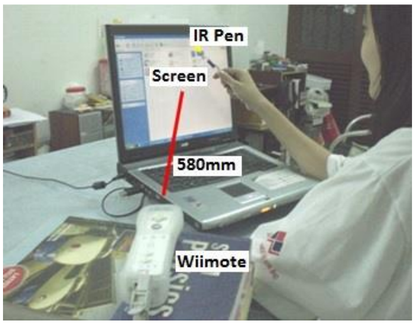
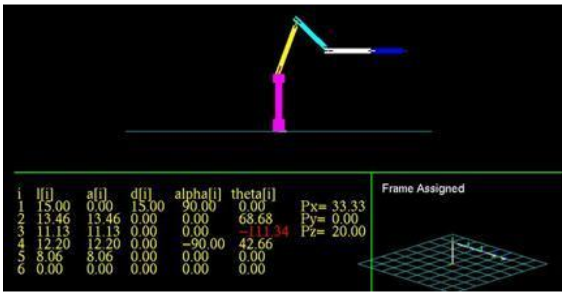


Figure 8. Experiment Setup

6.1 Case-1: Clamping

In clamping situation, the end effector of the robot is assumed to be face-forward, as shown in Figure 9. Therefore, Link-5 (blue colour) is always placed horizontally. The link parameters are tabulated in Table 2.



```
i  l[i]   a[i]   d[i]   alpha[i]  theta[i]
1  15.00  0.00   15.00  90.00     0.00        Px= 33.33
2  13.46  13.46  0.00   0.00      68.68       Py= 0.00
3  11.13  11.13  0.00   0.00      -111.34     Pz= 20.00
4  12.20  12.20  0.00   -90.00    42.66
5  8.06   8.06   0.00   0.00      0.00
6  0.00   0.00   0.00   0.00      0.00
```

Figure 9. Example for clamping robot model

Table 2. Link parameters

Joint	Link length *a* (unit)	Link angle *α* (degree)	Joint length *d* (unit)	Joint angle *θ* (degree)
1	0	90	*d*1= 15.00	θ_1
2	*a*2 = 13.46	0	0	θ_2
3	*a*3 = 11.13	0	0	θ_3
4	*a*4 =12.20	-90	0	θ_4
5	*a*5 = 8.06	0	0	θ_5

Following this, the angle obtained from inverse kinematics (Figure 9) is replaced into equation (5),

$$S_{23} = \sin(68.68 - 111.34) = -0.6777$$
$$C_{23} = \cos(68.68 - 111.34) = 0.7354$$

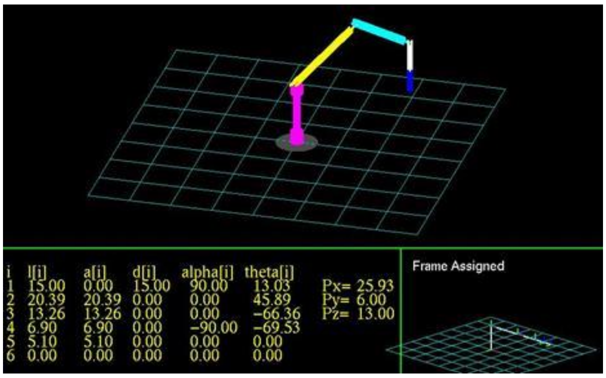
Thus, from equation (4):

$$\begin{aligned} P_x &= (0.5408 \times 20.26) - (-0.4592 \times 20.26) - 0 + 13.0786 \\ &= 33.3386 \text{ unit} \\ P_y &= 0 \text{ unit} \\ P_z &= (-0.4983 \times 20.26) + (0.4983 \times 20.26) + 12.5389 - 7.5422 + 15 \\ &= 19.9967 \text{ unit} \end{aligned}$$

As shown in Figure 9, the values for the end effector's positions (33.33, 0.00, 20.00) is same as the manual calculation using forward kinematics. The data verification proves that the program can generate very accurate results for inverse kinematics.

6.2 Case-2: Drilling

Conversely, the end effector of the robot is assumed to be faced downward in the drilling situation, in which Link-5 is always faced downward (Figure 10)



```
i  l[i]   a[i]   d[i]   alpha[i]  theta[i]
1  15.00  0.00   15.00   90.00     13.03     Px= 25.93
2  20.39  20.39  0.00    0.00      45.89     Py= 6.00
3  13.26  13.26  0.00    0.00     -66.36     Pz= 13.00
4  6.90   6.90   0.00   -90.00    -69.53
5  5.10   5.10   0.00    0.00      0.00
6  0.00   0.00   0.00    0.00      0.00
```

Figure 10. Example for drilling robot model

Similar to case-1, the angles from the drilling parameters obtained from Figure 10 are replaced into equation (5),

$$S_{23} = \sin(45.89 - 66.36) = -0.3497$$
$$C_{23} = \cos(45.89 - 66.36) = 0.9369$$

Thus, from equation (4):

$$\begin{aligned} P_x &= (0.3192 \times 12) - (0.31922 \times 12) - 0 + 25.9302 \\ &= 25.9302 \text{ unit} \\ P_y &= (0.0739 \times 12) - (0.0739 \times 12) + 0 + 6.000 \\ &= 6.000 \text{ unit} \\ P_z &= (-0.1223 \times 12) - (0.8777 \times 12) + 14.6401 - 4.6370 + 15 \\ &= 13.0021 \text{ unit} \end{aligned}$$

The forward kinematics shows that the system can generate very accurate results for inverse kinematics. The position of the end effector from the system (Figure 10) same as the results calculated by using forward kinematics. The data verification proves that the program can generate accurate results for inverse kinematics.

7. CONCLUSION

The developed Virtual Industrial Robot Kinematics Simulator can be used to visualize the motion of robots and the working range. It can be used in educational and training purposes as it can demonstrate the solution for inverse kinematics problems. The simulator provides the user with a natural, user-friendly and intuitive way to design robot arms without the need of robotic skills. The users are not required to draw the robot models in CAD software in order to observe the performance of the designed robots. From this research, the users are simply requested to define the joints and link parameters to build virtual 3D robot models. Moreover, the user-friendly control interface allows not only experts or trained operators, but novice users to take advantages of controlling the robots. This program is simple compared with other programs in the market, and the input method is intuitive.

Furthermore, this system allows user to simulate the robot's motion in virtual 3D world. This enables users to visualize the performance of robot, such as work envelope, clearances and reaches. Presently, this system can be used to solve the inverse kinematics robots with different link parameters. This system provides a simple way to solve the inverse kinematics problem and generate results in graphical format.

ACKNOWLEDGMENTS

Thank you to the Department of Mechanical Engineering, Faculty of Engineering, University Malaya, for providing the necessary facilities to support this study. This work was supported by the Ministry of Higher Education of Malaysia under Grant No. GA012-2014.

REFERENCES

Al-Dabbagh, R.D., Kinsheel, A., Mekhilef, S., Baba, M.S., Shamshirband, S. (2014) System Identification and Control of Robot Manipulator Based on Fuzzy Adaptive Differential Evolution Algorithm. *Advances in Engineering Software*, **78**, pp. 60-66.

Devanit, J., Hartenberg, R.S. (1955) A Kinematics Notation for Lower-Pair Mechanisms Based on Matrices. *Transaction of the ASME Journal of Applied Mechanics*, **22**, 215-221.

Fluckiger, L. (1998) A Robot Interface using Virtual Reality and Automatic Kinematic Generator. *Proceedings of the 29th International Symposium on Robotics*, Birmingham, UK, 123-126,

Groover, M.P. (1986) *Industrial Robotics: Technology, Programming and Application*. McGraw-Hill, New York

ISO (1994). No. 8373:1994. *Manipulating industrial robots – vocabulary*.

Johnny, C.L. (2010) Low-cost multi-point interactive whiteboards using the Wiimote. http://johnnylee.net/projects/wii/.

Kasi, V., Mekhilef, S., Ghazilla, R.A.R., Ahmad, N. (2014) Robotic System Development for Cooperative Orthopedic Drilling Assistance. *Advances in Mechanical Engineering*, **2014**, Article ID 437485.

Mavroidis, C., Ouezdou, F.B., Bidaud, P. (1994) Inverse Kinematics of Six Degree of Freedom General and Special Manipulators using Symbolic Computation. *Robotica*, **12**, 421-430

Pai, Y.S., Yap, H.J., Dawal, S.Z.D., Ramesh, S., Phoon, S.Y. (2016). Virtual Planning, Control and Machining for a Modular-Based Automated Factory Operation in an Augmented Reality Environment. Scientific Reports, 6, 27380.

Rohrmeier, M. (2000) Web Based Robot Simulation using VRML. *Proceedings of the Winter Simulation Conference*, Orlando, FL, USA,1525-1528.

Taha, Z., Soewardi, H., Dawal, S.Z.M. (2014). Axiomatic Design Principles in Analysing the Ergonomics Design Parameter of a Virtual Environment. *International Journal of Industrial Ergonomics*, **44**, 368–373.

Tahriri, F., Mousavi, M., Yap, H.J., Dawal, S.Z.M., Taha, Z. (2015) Optimizing the Robot Arm Movement Time using Virtual Reality Robotic Teaching System. *International Journal of Simulation Modelling*, **14**, 28-38.

VRL (2002) Virtual Reality Laboratory - Robots in Virtual Manufacturing. http://www-vrl.umich.edu/sel_prj/robots/index.html

Yap, H.J., Taha, Z., Lee, J.V. (2008) VR-based Robot Programming and Simulation System for an Industrial Robot. *International Journal of Industrial Engineering: Theory, Applications and Practice*, **15**, 314-322.

Yap, H.J., Taha, Z., Dawal, S.Z.M., Chang, S.W. (2014) Virtual Reality Based Support System for Layout Planning and Programming of an Industrial Robotic Work Cell. *PLoS ONE*, **9**, e109692.

Žlajpah, L. (2008) Simulation in Robotics. *Mathematics and Computers in Simulation*, **79**, 879-897.