

PAPER • OPEN ACCESS

Simulation of Mitsubishi RV-M1 Robotic Arms by Using MATLAB® for High School Teaching

To cite this article: D Prabowo *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **598** 012104

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

Simulation of Mitsubishi RV-M1 Robotic Arms by Using MATLAB® for High School Teaching

D Prabowo¹, M Wiannastiti² and R Hedwig¹

¹Computer Engineering Department, Faculty of Engineering, Bina Nusantara University, 9 K.H. Syahdan, Jakarta 11480, Indonesia.

²School of Computer Science, Bina Nusantara University. 9 K.H. Syahdan, Jakarta 11480, Indonesia.

rinda@binus.edu

Abstract. Since most of high school students in Jakarta love robots very much, they join Engineering Club at school. Unfortunately, only few high schools have a real robot arm for students to learn in their school labs. The purpose of this research is to create a small simulation program based on MATLAB® and Robotic Toolbox 9.10 which resembles Mitsubishi RV-M1 robotics arm. The simulation is a 3D view and 5 degree of freedom which its end-effectors position is as close as the real one. It is hoped the results of this research study can be used as a base for further development of human controlled robotics arms simulation.

1. Introduction

Since 2014, robotic has become one of the popular subjects in most high schools in Jakarta and the number of engineering clubs has increased. Four engineering clubs from different schools asked Computer Engineering Department of Bina Nusantara University (BINUS) to give free robotics classes with about 20 students in a class regularly. We, the Engineering Department agreed and took it as a part of our community development projects. At the beginning, we brought the only RV-M1 robot to schools but since the number of schools required Robotic classes increased and none had their own robot, one RV-M1 robot was not enough to use for those classes. Besides, the robot was also used internally at BINUS for its department students' extracurricular activities.

The idea of building the simulation program is based on the reason that most of the students learn sensor, control, dynamic, and kinematics as the profound study about manipulator movement in space with the function of time regardless its force and momentum [1]. The movement of robot arm uses kinematic study and students learn its connection between its position and orientation of manipulator links. The manipulator consists of links and joints in the form of revolute or prismatic. The manipulator, which is divided into direct kinematic and inverse kinematic, comes in the form of an arm, fingers, and a leg [2].

In this paper, we focus on direct kinematic since it is a kinematic analysis which involves the forward transformation equation to get coordinate position of x, y, and z, as well as its orientation. [3] To support the discussion of direct kinematic analysis, inverse kinematic which involves the inverse transformation is used. This kinematic is used to get the angle and position orientation of each joints which determine the final position of the arm robot. The idea of this research is to give a free simulation application for High Schools so that they can practice about robot arm in the class.



To overcome the situation and fulfil the need of learning robotics in high school, we build a simulation program to study the kinematics of robot arm in a computer. The simulation is similar to both PUMA 560 and Mitsubishi RV-M1 Move Master. We use the last model since it is the robot available in Computer Engineering laboratory. Moreover, this simulation is comparable to the real model of RV-M1 during the test. Therefore, this simulation can be used and applied in high schools as a substitute of real RV-M1 robot. The simulation uses 5 degree of freedom (DOF) with forward kinematic where its analysis uses Denavit-Hartenberg (DH) convention [4][5]. Based on this study, it is hoped we are able to present a simple simulation for students. The research question of this experiment will be “does the arm robot simulation works according to the real arm robot?”

2. System Design and Algorithm

Before designing the system, it is important to understand the kinematic problem faced in simulated robot based on its joints' characteristics. There is $n+1$ links for a n -joint robot in its arm. Each joint connects two links [6][7]. The limitation of arm movement is derived from the rotation and the translation boundaries. If it is a rotation joint, we use a revolute joint, otherwise a prismatic joint. The rotation joint boundary is defined in two values; negative rotation (against right hand rules) and positive rotation (according to right hand rules) [8]. In the meantime, the prismatic joint boundary is also defined into two values; positive translation (along with positive z axes) and negative translation (along with negative z axes) [9]. We can calculate the kinematic movement from the joint boundary of simulation parameter.

Figure 1 shows the model and parameter of RV-M1 which is used as the model of our simulation in this research [10]. We use DH in direct forward kinematics to develop homogenous matrix as follow:

$${}^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\cos\theta_i \sin\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \sin\alpha_i & -\cos\theta_i \cos\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

A is a homogenous matrix that has information on 3D simulation graphic whereas each value in the A is different for each joint. Each joint depends on link characteristic as shown in table 1 [11]. We design the system based on this parameters as shown in figure 2.

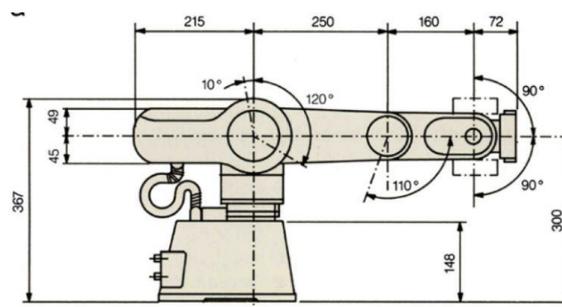


Figure 1. The parameters of RV-M1[10]

Table 1. DH Parameters for RV-M1 [11] with symbols referring to equation (2).

Links	a_n (mm)	α_n (rad)	θ_n (rad)	d_n (mm)
1	0	pi/2	θ_1	152
2	250	0	θ_2	0
3	160	0	θ_3	0
4	72	pi/2	θ_4	0
5	107	θ_5	0	0

The input of the system is in the form of angle's value for each link which later is fitted in current DH parameter. The next, we develop homogenous transform matrix for each joint which calculates and finds the position of joint link and end-effector (EE).

The flowchart of the whole system can be seen in figure 3 where in the beginning the program runs graphical user interface (GUI) and initializes the first position of robot arm. In the GUI, a user can easily manipulate the data parameters of robot arm. The first position of robot arm's angles is set at 0, 0, 0, 0, 0 (5 DOF) and this continually does whenever the system resets. Later, after setting the beginning position of robot arm, the program executes kinematics program by entering the parameter in the calculation. If there is any data change, the system will update GUI display according to the changes. The display is in 3D (x, y, z axes) of the latest position of RV-M1 EE.

By using equation

$${}^{i-1}A_i = T_{x,a}T_{x,\alpha}T_{z,\theta}T_{z,d} \quad (2)$$

Where

- ${}^{i-1}A_i$ - transform matrix of i th joint to i th-1 joint.
- $T_{x,a}$ - translation matrix of x axes, value of an a .
- $T_{x,\alpha}$ - rotation matrix of x axes, value of degree of α .
- $T_{z,\theta}$ - rotation matrix of z axes, value of degree of Θ .
- $T_{z,d}$ - translation matrix of z axes, value of a d .

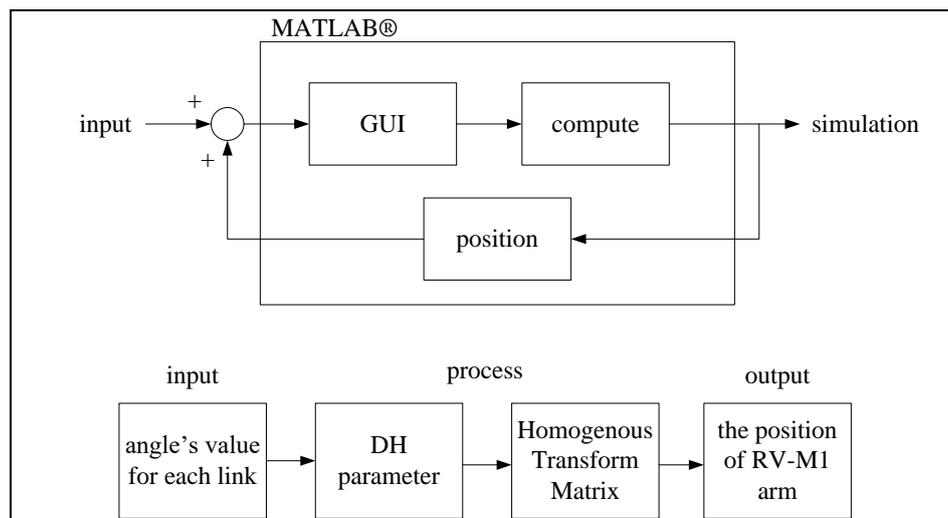


Figure 2. System block diagram

and homogeneous transform matrix (equation 1) we can determine

$${}^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i \cos\alpha_{i-1} & \cos\theta_i \cos\alpha_{i-1} & -\sin\alpha_{i-1} & -\sin\alpha_{i-1} d_i \\ \sin\theta_i \sin\alpha_{i-1} & \cos\theta_i \sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

in which we can input in DH parameter in table 1. Therefore, we get the value of matrix transform of each joint link. Thus we can determine the EE position.

Since we can manipulate matrices and its counting in Matrix Laboratory of MATLAB®, we try to combine it with Toolbox: Robotics Toolbox 9.10 from Peter Corke [12][13][14]. There are functions such as rotational matrix counting, translation matrix counting, robot arm frame developing and others in the toolbox which is favorable to build a 3D simulation of RV-M1 robot arm. The flowchart of kinematics programming can be seen in figure 4.

In this function, we set the DH parameter from the robot arm where alpha, a, d, theta variables store α , a, d, and Θ value respectively. Other variables such as bon, and bin, are used to store links, and start the loop. Deg is used as the constant number and Trans is used as transformation matrix. This function gives EE value which is displayed in a frame with 3D axes. The simple graphical user interface (GUI) designed is presented in figure 5 and the display of the RV-M1 simulation is shown in figure 6 and 7 with different point of view. The flowchart of the GUI and the detailed program can be accessed freely by requesting to corresponding author.

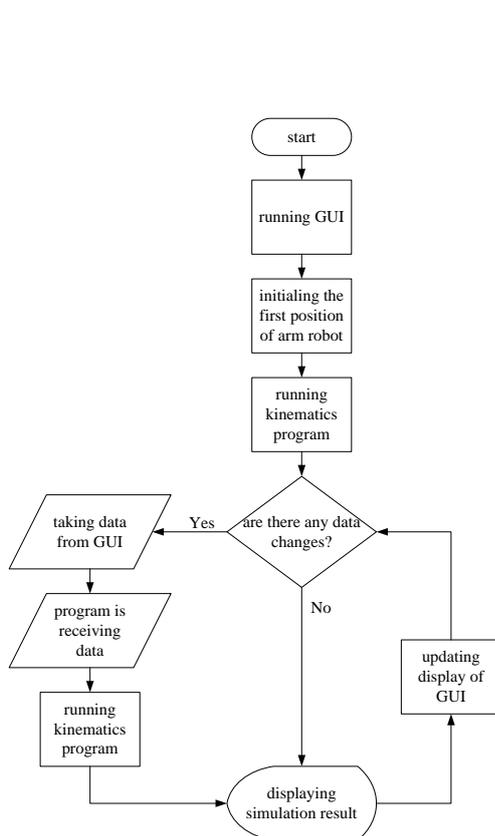


Figure 3. Flowchart of the system

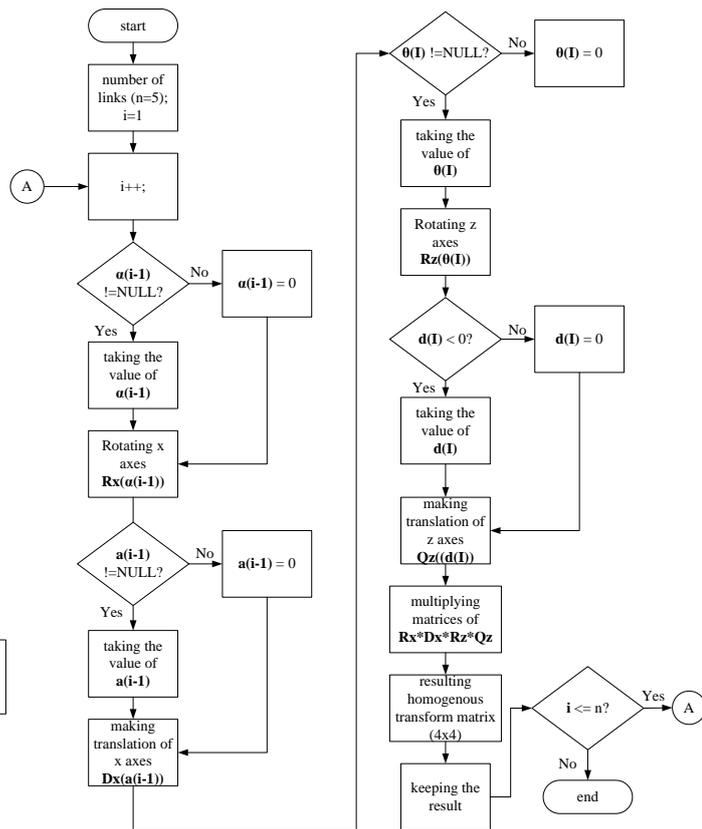


Figure 4. Flowchart of kinematics programming

The function of the flowchart is shown as follows

```

function [e_x, e_y, e_z] = trans_for(A, B, C, D, E )
%create transformation matrix for function

bon = 5; %Jumlah Link
bin = 2; %Inisialisasi Link
% (i) = [x, 1, 2, 3, 4, 5]
alpha= [0, -90, 0, 0, -90, 0];
a= [0, 0, 0.250, 0.160, 0.072, 0.075];
d= [0,0.152,0,0,0,0];
theta = [0, A, B, C, D, E];

deg = pi/180;
Trans = eye;
%Links
while(bin<=bon+1)
    P = trotx((pi/180)*alpha(bin-1),deg);
    Q = [ 1 0 0 a(bin-1); 0 1 0 0; 0 0 1 0; 0 0 0 1];
    R = trotx(theta(bin)*(pi/180),deg);
    S = [ 1 0 0 0; 0 1 0 0; 0 0 1 d(bin); 0 0 0 1];
    Trans = Trans *(P*Q*R*S);
    bin = bin+1;
end

mat = [0 0 0 1] * Trans';
e_x = mat(1);
e_y = mat(2);
e_z = mat(3);

% Detailed explanation goes here

mdl_rvm1
figure(1)
rvml.plot([A*deg B*deg C*deg D*deg E*deg]);
title('RV-M1 arm')

end

```

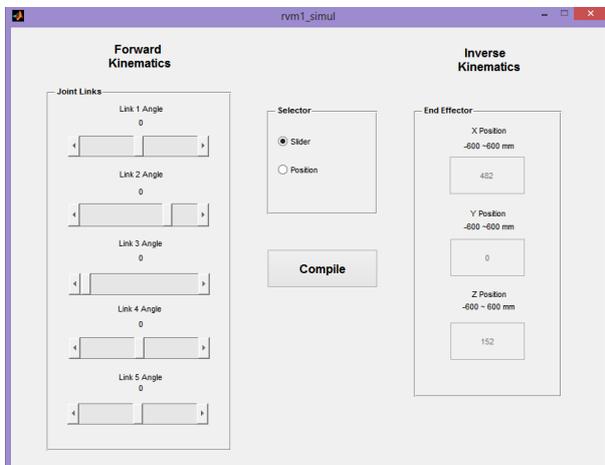


Figure 5. GUI program display

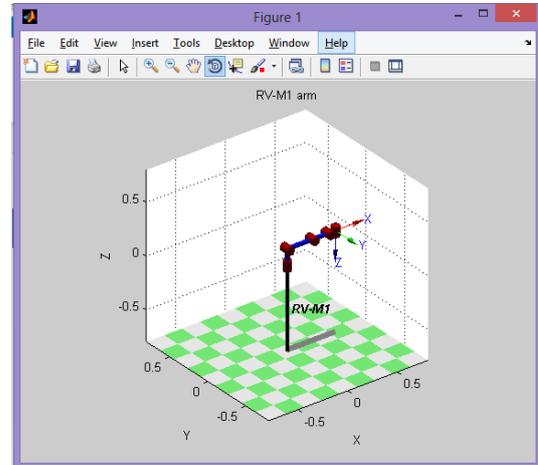


Figure 6. 3D frame of RV-M1 simulation

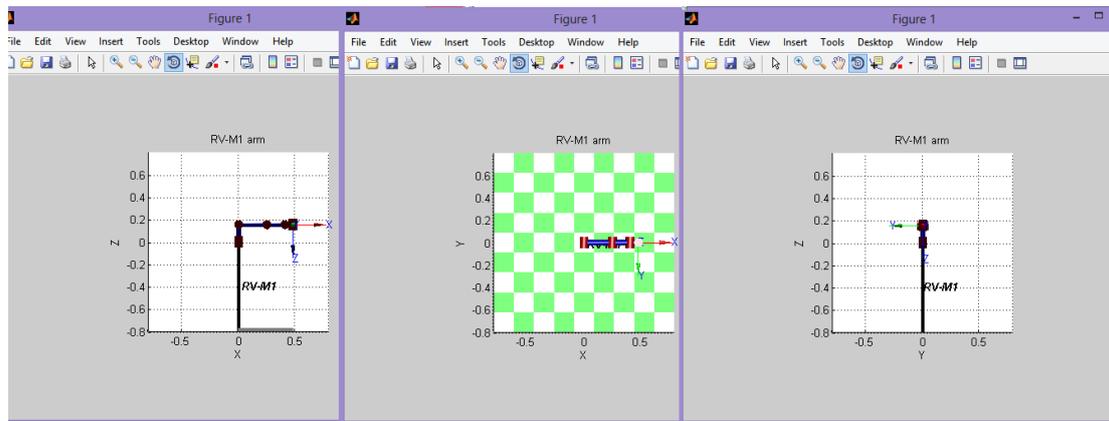


Figure 7. Frame display of robot arm simulation in x-z axes, x-y axes, and y-z axes, respectively

3. Result and Discussion

After several experiments, we found two results. First, the result of comparison between the real RV-M1 robot arm movement with the one simulated in the program. Table 2 shows the differences between the simulation movement result and the real RV-M1 movement result. From the table 2 and 3, it can be seen that the displacement between the actual robot arm movement and simulation result is not significant when the Theta was moved from one degree to another degree. Therefore, the simulation can be used for studying since it will resemble the real RV-M1 robot.

The second, the result of consistency of EE movement in simulation by using DH parameter and matrix transforms in 2 axes (X, Y). In this experiment, we compared the simulation result with the actual RV-M1. The EE was moved in 2 positions; A (100, -250) and B (100, -400) where the simulation was run 20 times for the same position. The validation process between the simulator and the real robot arm is shown in figure 8 and 9. The comparison result of validation process is shown in table 3. We found out that the simulation successfully represented the movement position of actual RV-M1. Thus, the simulation can be used in the classroom for the student without using the real RV-M1 robot arm.

From these two experiments, we conclude that the EE position movement simulated in the program with the actual movement of RV-M1 is 82% to 99% of accuracy. In this case, we do not compare the algorithm we used with others since we only used the algorithm from Peter Corke [8] and modified it with the condition of RV-M1 in the lab.

Table 2. The differences displacement value between actual EE from RV-M1 and simulation EE from the simulation program in degree.

No	Theta					Data						Differences value		
						Actual EE			Simulation of EE					
	1	2	3	4	5	X	Y	Z	X	Y	Z	x	y	z
1	0	0	0	0	0	500	0	150	482	0	152	18	0	2
2	90	0	0	0	0	0	500	150	0	482	152	0	18	2
3	90	-90	0	0	0	0	0	650	0	0	634	0	0	26
4	90	-90	90	0	0	50	250	400	0	232	402	50	18	2
5	90	-90	90	90	0	50	150	350	0	160	330	50	10	20
6	90	-90	90	90	90	50	150	350	0	160	330	10	0	20
7	0	-90	90	90	90	150	0	350	160	0	330	22	0	8
8	0	0	90	90	90	200	0	0	178	0	-8	12	0	20
9	0	0	0	90	90	400	0	100	410	0	80	10	0	20
10	0	0	0	0	90	500	0	150	482	0	152	18	0	2
11	130	0	0	0	90	-350	350	150	-310	369	152	40	19	2
12	130	-90	0	0	90	0	0	600	0	0	634	0	0	34
13	130	-90	90	0	90	-150	-150	400	-149	178	402	1	28	2
14	130	-90	90	-90	90	-150	150	450	-103	123	474	47	27	24
15	45	0	0	0	90	350	300	100	340	340	153	10	40	50
16	45	0	0	90	90	300	300	100	290	290	80	10	10	20
17	45	0	90	90	90	100	100	0	128	128	-8	28	28	8
18	45	-90	90	90	90	100	100	350	113	113	330	13	13	20
19	45	-90	90	-90	90	100	100	500	113	113	474	13	13	26
20	45	0	90	-90	90	200	250	0	228	228	-8	28	22	8
Average												19	12.3	15.8

**Figure 8.** The actual position of RV-M1 on B position

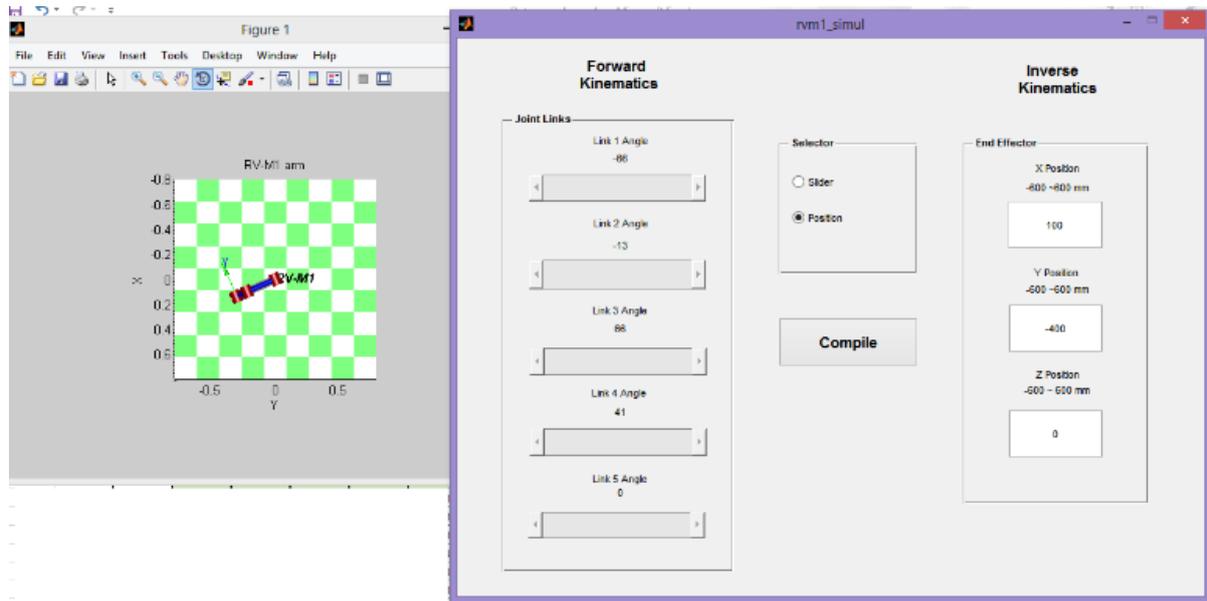


Figure 9. Position of B in the simulation

Table 3. The differences of position between simulation result and actual result in degree

NO	Experiment											
	Actual				Simulation				Differences			
	Position of A		Position of B		Position of A		Position of B		Position of A		Position of B	
	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
1	100	250	100	-400	100	250	100	-400	0	0	0	0
2	100	250	100	-400	100	250	100	-400	0	0	0	0
3	100	250	100	-400	100	250	100	-400	0	0	0	0
4	100	250	100	-400	100	250	100	-400	0	0	0	0
5	100	250	100	-400	100	250	100	-400	0	0	0	0
6	100	250	100	-400	100	250	100	-400	0	0	0	0
7	100	250	100	-400	100	250	100	-400	0	0	0	0
8	100	250	100	-400	100	250	100	-400	0	0	0	0
9	100	250	100	-400	100	250	100	-400	0	0	0	0
10	100	250	100	-400	100	250	100	-400	0	0	0	0
11	100	250	100	-400	100	250	100	-400	0	0	0	0
12	100	250	100	-400	100	250	100	-400	0	0	0	0
13	100	250	100	-400	100	250	100	-400	0	0	0	0
14	100	250	100	-400	100	250	100	-400	0	0	0	0
15	100	250	100	-400	100	250	100	-400	0	0	0	0
16	100	250	100	-400	100	250	100	-400	0	0	0	0
17	100	250	100	-400	100	250	100	-400	0	0	0	0
18	100	250	100	-400	100	250	100	-400	0	0	0	0
19	100	250	100	-400	100	250	100	-400	0	0	0	0
20	100	250	100	-400	100	250	100	-400	0	0	0	0

4. Conclusion

The simulation of RV-M1 robot is made in the form of .EXE program which is easy for the schools to directly install the program in the computer without the necessity of installing MATLAB®. Therefore, this program is user friendly and can be used freely by students in the school lab for a robot simulation subject. The experiment result can be accessed via YouTube at <https://www.youtube.com/watch?v=H1UixZTwOko>.

Even though this experiment is simple, and many similar programs are available in the internet, this program still gives an advantage, that is easier to install, and schools can get this coding of the program for free. In the future, the research will be continued by adding the dynamic model, as well as increasing the degree of freedom of robot arm.

References

- [1] J.P. Merlet. 1993 *IEEE Journal on Robotics and Automation* **9** 842-46.
- [2] R.N. Jazar. 2010 *Theory of Applied Robotics: Kinematics, Dynamics, and Control* (New York: Springer).
- [3] M.W. Spong, S. Hutchinson, M. Vidyasagar. 2004 *Robot Dynamics and Control 2nd edition* [Online] Available at: http://smpp.northwestern.edu/savedLiterature/Spong_Textbook.pdf [Accessed 2 May 2017].
- [4] J. Chen, L.M. Chao. 1987 *IEEE Journal on Robotics and Automation* **3** 539-45.
- [5] H.A. Basher. 2007 Modeling and simulation of flexible robot manipulator with a prismatic joint. In: *Proceedings 2007 IEEE SoutheastCon*. Richmond USA.
- [6] Anonymous 1996 *Industrie-Roboter: Anwender-Handsbuch RV-M1* (Ratingen: Mitsubishi Electric Europe GMBH).
- [7] R. Kumar, P. Kalra, N.R. Prakash. 2011 *Robot. CIM-Int. Manuf.* **27** 994-1000.
- [8] P. Corke. 2011 *Robotics Toolbox* [Online] Available at: <http://petercorke.com/wordpress/toolboxes/robotics-toolbox> [Accessed 3 May 2017].
- [9] P. Corke. 2013 *Robotics, Vision and Control: Fundamental Algorithms in MATLAB* (Berlin: Springer).
- [10] P.I. Corke. 1996 *IEEE Robot. Autom. Mag.* **3** 24-32.
- [11] P.I. Corke. 2007 *IEEE T. Robot.* **23** 590-94.
- [12] E.J. Haug. 1989 *Computer Aided Kinematics and Dynamics of Mechanical Systems, Vol 1: Basic Methods* (Massachusetts: Allyn and Bacon).
- [13] R. Boulic, D. Thalmann. 1992 *Comput. Graph. Forum.* **11** 189-202.
- [14] C.R. Rocha, C.P. Tonetto, A. Dias. *Robot. 2011 CIM-Int Manuf.* **27** 723-28.