

PAPER • OPEN ACCESS

2D and 3D robot arms simulation using GeoGebra

To cite this article: D U Suwarno and R C E Putri 2024 *J. Phys.: Conf. Ser.* **2684** 012002

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



PRIME
PACIFIC RIM MEETING
ON ELECTROCHEMICAL
AND SOLID STATE SCIENCE

HONOLULU, HI
Oct 6–11, 2024

Abstract submission deadline:
April 12, 2024

Learn more and submit!

Joint Meeting of

The Electrochemical Society
•
The Electrochemical Society of Japan
•
Korea Electrochemical Society

2D and 3D robot arms simulation using GeoGebra

D U Suwarno* and R C E Putri

Electrical Engineering Department, Sanata Dharma University, Yogyakarta, 55282
Indonesia

*E-mail: joko_unt@usd.ac.id

Abstract. Student studying robot arms requires mathematical knowledge and programming skills to simulate robot arm movements. Students often have difficulty understanding the mathematical equations of robotic arms and how to simulate them. The author tries to solve this problem using GeoGebra by constructing a 2D robot arm and a 3D robot arm and performing rotation operations on the robot arm. The robot arm consists of an upper arm and a lower arm and can be rotated independently. The simulations that have been carried out are in the form of a 2D robot arm and a 3D robot arm with an R, R, R configuration. With the help of GeoGebra, students are helped to simulate a robot arm with an RR (revolute revolute) joint configuration.

1. Introduction

To study robotic arms requires basic knowledge of coordinate systems, both Cartesian coordinates and polar coordinates. The robot arm consists of an upper arm and a lower arm with an RR (Revolute Revolute) joint configuration. The robot arm is controlled by the angle of the upper arm and the angle of the lower arm to obtain the desired end effector point. To find out the coordinates of the arm and the final position of the end effector, a simulator is needed. The robot arm mechanism is an application of mechanics with rotational and translational operations. Studying robot mechanisms requires visualization of arms that can be changed quickly in angle and arm length. The difficulty experienced by students is how to build and turn on the robot arm. GeoGebra offers several representations of mathematical functions, GeoGebra is used to help students create graphs of mathematical functions. Animating the movement of a robotic arm requires basic knowledge such as forming segments or joints and performing rotational operations on the arm.

The simulator robot using GeoGebra was done by Homa [1]. The robot built is an arm robot with revolute and prismatic configurations for STEM (Science, Mathematics, Technology and Engineering) education. Homa creates activities for students to study robotics, manipulate robot arms, position objects in a plane, apply mathematics, and solve inverse problems. The use of GeoGebra as validation of robot arm movements was also carried out by Wardhana [2]. The robot arm model used is an R-R-R robot arm.

GeoGebra is used in STEM education which includes topics of trigonometry, calculus, geometry-based design, programming, and visualization. Robotics has a lot to do with trigonometry, calculus, and programming (sequence of steps) so GeoGebra is suitable for robotics problems.

In this paper, the authors will discuss robot arm rotation using GeoGebra. GeoGebra is a web-based application for mathematical problems and makes visualization easier. This work aims to show how dynamic geometry concepts can be applied to subjects using GeoGebra software.



2. Theoretical background

2.1. Kinematics robot arm

The robot arm consists of an upper arm and a lower arm. The upper arm is connected to the base and the lower arm is the end effector. The point connecting the base, upper arm, and lower arm joints and the end effector forms a triangle. The movement of the RR robot arm requires basic knowledge of the properties of triangles. The visualization of the robot arm is shown as in figure 1, where Link 1 and Link 2 will form an arbitrary triangle. The joint of the robot arm can be a Revolute Joint. Point A is the base point which is the revolute joint, B is the revolute joint, and C is the end effector of the robot arm. Robot movement manipulation is carried out to move the end effector at the endpoint of the system. The position of the end effector can be determined by changing the angle of the upper arm and the angle of the lower arm.

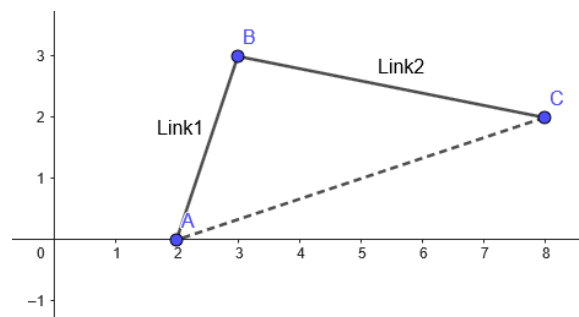


Figure 1. 2D robot arm model with R R joints.

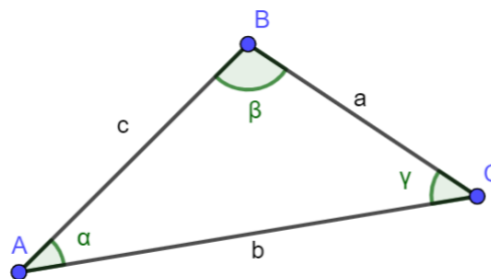


Figure 2. The cosine rule.

To calculate the position of a point or angle of a triangle, the cosine rule is used as shown in figure 2. The cosine rule is used to calculate the sides of a triangle for opposite angles. The side length of the triangle is calculated simultaneously as follows:

$$a^2 = b^2 + c^2 - 2bc \cos(\alpha)$$

$$b^2 = a^2 + c^2 - 2ac \cos(\beta)$$

$$c^2 = a^2 + b^2 - 2ab \cos(\gamma)$$

Triangular angles are calculated using the following equation:

$$\cos(\alpha) = \frac{b^2 + c^2 - a^2}{2bc}$$

$$\cos(\beta) = \frac{a^2 + c^2 - b^2}{2ac}$$

$$\cos(\gamma) = \frac{a^2 + b^2 - c^2}{2ab}$$

In the textbook, rotation operations on the robot arm use a rotation matrix as equation (1). Matrix rotation 2D:

$$Rot(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (1)$$

2.2. GeoGebra

GeoGebra is an application for mathematical calculations, statistics, and visualization of mathematical functions. GeoGebra is Open Source (Open Source Software/Free Software where the license for the program is provided free of charge to its users. Users can use this program for various purposes, both for learning and for modifying programs.

2.3. Manipulated robot arm with GeoGebra

Generally, the movement of robots in 3D uses cosine rules which are then converted to cartesian coordinates. The 3D rotation matrix describes the position and orientation of the frame on the axis of space assigned to each link. In addition to using cosine rules, and manipulation of rotation, 3D robots on GeoGebra can use cartesian and polar coordinates.

The 3D coordinate system (spherical coordinates, sphere) requires parameters R, alpha angle, and beta angle. The alpha angle is the horizontal angle, while the beta angle is the vertical angle. The position of a robotic arm, several components make up its unique position. This unique position consists of an angular shape formed by the revolute joint of Link G1 against the base forming θ_1 , Link G2 against the base forming θ_2 . The next unique position is the unique position formed from the presence of prismatic joints that will cause a translational process in the arm $L1_x L1_y$ dan $L2_x L2_y$ shown in figure 3.

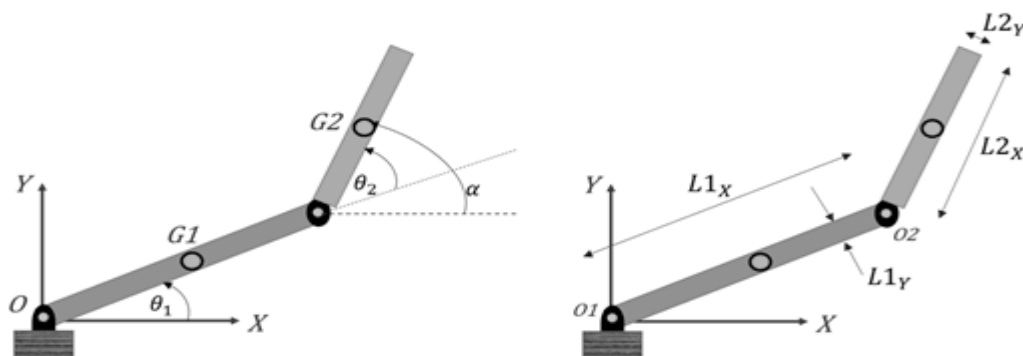


Figure 3. An example of a robotic arm calculation from a textbook [3, 4].

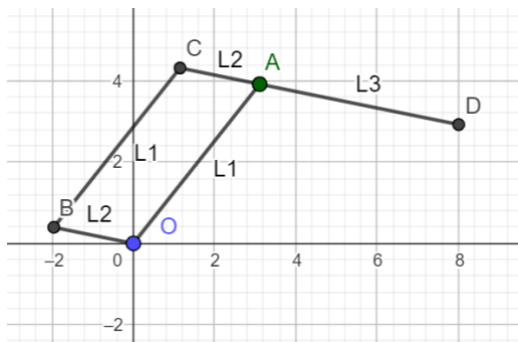
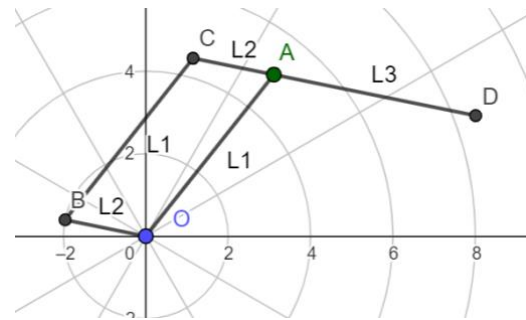
The simple form of the above system is visualized with GeoGebra. Visualizing the system with GeoGebra with syntax as shown in table 1.

Table 1. Command to create points using GeoGebra.

Protocol Construction	Description
$A=(3,4)$	Point on Cartesian Coordinate
$B=(5;30^\circ)$	Point on Polar Coordinate, angle in degree
$B=(5;\pi/6)$	Point on Polar Coordinate, angle in radian
$C=(2,3,4)$	3D Point on Cartesian coordinate
$D=(5;30^\circ;45^\circ)$	A 3D point on Polar Coordinates, Longitude, and Latitude angle

3. Methods

In this study, a robot arm with an R-R (revolute-revolute) configuration, an independent Revolute was used. The arm robot consists of two main arms, namely the upper arm and lower arm, which can be controlled separately. The next step is to make construction in GeoGebra carry out forward kinematic calculations and compare the Cartesian coordinate system and polar coordinates as shown in figure 4 and figure 5.

**Figure 4.** The robot arm on Cartesian grid.**Figure 5.** The robot arm on polar grid.

The syntax of the parallel link system above can be visualized with GeoGebra. Visualizing the system with GeoGebra can be done with syntax as shown in table 2.

Table 2. Protocol construction for robotics arm 2D on GeoGebra.

Protocol Construction	Description
$L1=5$	Upper arm length (5 unit radius)
$L2=2$	Link to manipulate the lower arm
$\alpha=\text{slider}(0,\pi)$	angle range for upper arm L1
$A=(L1;\alpha)$	Joint coordinate for Upper Arm (in polar coordinate)
$\beta=\text{slider}(0,\pi)$	angle range for lower arm L2 and L3
$B=(L2;\beta)$	Joint coordinate for lower arm manipulator L2 (in polar coordinate)
$C=B+A$	Join for lower arm manipulator
$D=A+(L1;-\beta)$	Link for end effector
$\text{Segment}(O, A)$	Link for upper arm L1
$\text{Segment}(O, B)$	Link for arm manipulator L2
$\text{Segment}(C, D)$	Link for lower arm L2, L3

4. Results and discussion

Visualization of movement manipulation on the robotic arm using a parallel system can be visualized in 3D. An example of syntax for GeoGebra 3D application is shown in table 3. The visualization syntax of the robotic arm is shown in figure 6.

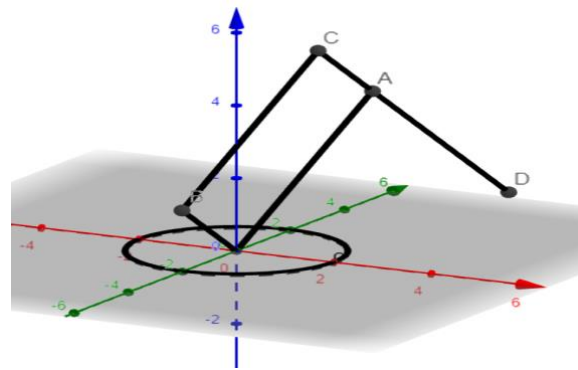


Figure 6. The robot arm construction in 3D using GeoGebra.

Table 3. GeoGebra protocol construction for 3D robot arm configuration R, R, R.

Protocol Construction	Description
$L1=5$	Upper arm length (5 unit radius)
$L2=2$	Link to manipulate the lower arm
$\theta=\text{slider}(0,\pi)$	angle for the robot arm base, ranging from 0 to 180 degrees
$\alpha=\text{slider}(0,\pi)$	angle for upper arm L1
$A=(L1;\theta;\alpha)$	Joint coordinat for Upper Arm (in 3D polar coordinate)
$\beta=\text{slider}(0,\pi)$	angle for lower arm L2 and L3
$B=(L2;\theta; \beta)$	Joint coordinat for lower arm manipulator L2 (in 3D polar coordinate)
Segment(O, A)	Link for upper arm L1
Segment(O, B)	Link for arm manipulator L2
Segment(C, D)	Link for lower arm L2, L3

Commands using polar coordinates do not require the sine and cosine functions. The sine and cosine commands are required when converting polar coordinates to Cartesian coordinates. The coordinate position of Point D (end effector) is calculated using the cosine rule. Changes in manipulating the movement of the robot arm at the end effector point can be done by changing the value of α to change the angle of link A. Meanwhile, changing the value of β to change the angle of link B. The calculation of the position and angle of the end effector is shown in figure 7.

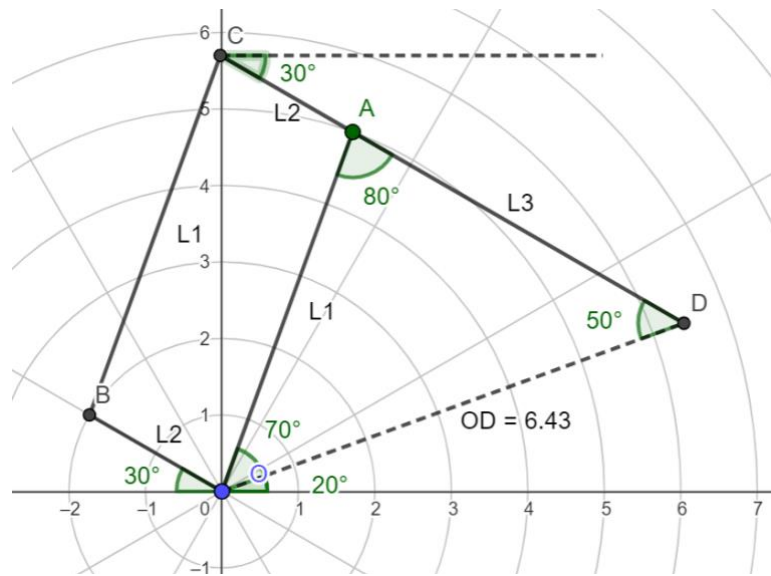


Figure 7. The end effector position and the angle of the robot arm.

Given $L_1 = 5$ cm ; $L_3 = 5$ cm; the angle of the upper arm robot $\alpha = 70^\circ$, the angle of the lower arm robot $\beta = 30^\circ$, end effector position can be calculated using following equation:

$$OD = \sqrt{L_1^2 + L_3^2 - 2L_1L_3\cos(\gamma)} = \sqrt{5^2 + 5^2 - 2 \times 5 \times 5 \times \cos(\angle 80^\circ)} = 6.43 \text{ cm.}$$

An end effector angle calculate using following equation:

$$\angle ADO = 180^\circ - \angle \beta - \angle \alpha = 180^\circ - 30^\circ - 70^\circ = 80^\circ,$$

$$\angle AOD = \angle DOA = \frac{180^\circ - 80^\circ}{2} = 50^\circ.$$

An end effector angle is $\angle OD = \angle \alpha - \angle DOA = 70^\circ - 50^\circ = 20^\circ$.

Through simulation and visualization of the robot arm using GeoGebra, users can validate equations with simulation results. Validation that can be shown includes the angles of the robot arm, the position of the end effector, and the working space of the robot arm [5]. Robotic arm simulation using GeoGebra is a combination of mathematics, programming, and engineering skills that can become a topic in STEM education [6, 7]. Geogebra supports STEM learning. In Mathematics: GeoGebra is used to visualize mathematical concepts, such as graphs, functions, and geometry. It helps students in understanding fundamental mathematical concepts in STEM subjects. In Science: GeoGebra can also be used in natural sciences to visualize relevant data, calculations and mathematical models. This can help students understand the scientific principles underlying science topics. In Technology: GeoGebra is technology-based software that can be used in STEM contexts. It teaches students the technology skills necessary to operate this math software. In engineering, GeoGebra can be used to design and visualize mathematical models related to engineering problems. It helps students and engineers in designing and testing engineering solutions. Thus, GeoGebra is a very useful tool in STEM education, as it helps students and teachers in understanding and teaching the mathematics and science concepts underlying STEM subjects.

5. Conclusion

The conclusions obtained are that 2D and 3D robot arm simulation using GeoGebra was successful. Robot arm movement is carried out using a slider facility to change the angle values for the base, upper arm, and lower arm. The GeoGebra construction protocol is given as an example that students can carry

out to make it easier to simulate the movement of a robot arm. GeoGebra is suitable for simulating mathematical equations and can be used in STEM education for students.

References

- [1] Homa A I R 2019 *Acta Sci.* **21** 178–191
- [2] Wardhana A S, Ashari M and Suryoatmojo H 2019 *Sol. Energy* **194** 436–449
- [3] Corke P I 1996 *Visual Control of Robots: High-performance Visual Servoing* (Taunton UK: Research Studies Press)
- [4] Craig J 2005 *Introduction to Robotics: Mechanics and Control* (Pearson Education International)
- [5] Ljubica D 2009 *Int. J. Emerg. Technol. Learn.* **4** 51–54
- [6] Lavicza Z et al. 2018 *Mathematics learning through arts, technology, and robotics: Multi-and transdisciplinary STEAM approaches* East Asia Regional Conference on Mathematics Education (National Taiwan Normal University)