

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/279201859>

Kinematics Modeling of a 4-DOF Robotic Arm

Conference Paper · May 2015

DOI: 10.1109/ICCAR.2015.7166008

CITATIONS

63

READS

57,873

2 authors, including:



[Amin A. Mohammed](#)

King Fahd University of Petroleum and Minerals

11 PUBLICATIONS 198 CITATIONS

SEE PROFILE

Kinematics Modeling of a 4-DOF Robotic Arm

Amin A. Mohammed and M. Sunar

Mechanical Engineering Department
King Fahd University of Petroleum & Minerals
Dhahran, Saudi Arabia
e-mail: mehmetts@kfupm.edu.sa

Abstract—This work presents the kinematics model of an RA-02 (a 4 DOF) robotic arm. The direct kinematic problem is addressed using both the Denavit-Hartenberg (DH) convention and the product of exponential formula, which is based on the screw theory. By comparing the results of both approaches, it turns out that they provide identical solutions for the manipulator kinematics. Furthermore, an algebraic solution of the inverse kinematics problem based on trigonometric formulas is also provided. Finally, simulation results for the kinematics model using the Matlab program based on the DH convention are presented. Since the two approaches are identical, the product of exponential formula is supposed to produce same simulation results on the robotic arm studied.

Keywords-Robotics; DH convention; product of exponentials; kinematics; simulations

I. INTRODUCTION

Nowadays, robotics is a rich area of research, in terms of their kinematics, dynamics and control. Kinematics, in particular, plays a significant role in robotics and especially for the study of industrial manipulators' behavior. Therefore, a decisive step in any robotics system is the analysis and modeling of the manipulator kinematics [1]. It can be divided generally into forward and inverse kinematics. The former refers to direct finding of the end effector (EE) position and posture for the given joint coordinates, on the contrary, the latter is the determination of joint variables in terms of the EE position and posture [2]. In other words, it is the process of obtaining a configuration space that corresponds to a given work space. It has several disciplines of applications including the computer graphics [3].

While the forward kinematics is straightforward, the inverse kinematics is not an easy task and it usually has multiple solutions. Numerous techniques and their combinations could be used for solving and analyzing the inverse kinematics problem. However each method beside its advantages possesses some handicaps. Therefore, it is advantageous to apply more than one method and to integrate them with additional procedures [3]. Several approaches have been investigated and compared by Aristides *et al* [4]. Jasjit Kaur *et al* [5] have analyzed and simulated a robotic arm having three links-manipulator, where the inverse kinematics results in two solutions, out of which the best one has been chosen with the usage of the genetic algorithm. They have concluded that the robotic arm movement optimization can be achieved by the genetic

algorithms in a practical and effective way. A geometric approach to solve the unspecified joint angles needed for the autonomous positioning of a robot arm is presented in [1]. Based on few assumptions concerning the working environment of the manipulator and the kind of manipulation required, in addition to the basic trigonometry, an easier solution has been proposed. The introduced methodology has been tested using five degrees of freedom robotic arm compounded to an i-Robot. By carrying out the mechanical design, all components have been assembled together in addition to two infrared sensors to detect the object and to find its position. Finally, the electrical design was done, and the proposed method was tested and justified. However, this approach lacks its generality due to the abundant assumptions considered, and what is more is that the last link is restricted to just two orientations. In what we introduce here is that the last link could have any arbitrary orientation, which makes the analysis more general to different configurations.

Ge and Jin [6] proposed an algorithm of a redundant robot kinematics based on product of exponentials (PE), where the forward and inverse kinematics problems were addressed. The inverse kinematics problem was decomposed into several sub problems and by studying them separately, a solution of the inverse kinematic problem was proposed. Kinematics and interactive simulation system modeling for robot manipulators is given in [7], where the forward and inverse kinematics of a robotic arm called Katana450 are studied by the exponential product method, then an interactive simulation system is built based on three-merit virtual scene modeling techniques proposed for robot manipulators. These techniques are OpenGL, VRML and another one based on Visual C++, Lab View and Matlab platform. A robot simulator, which is introduced by Lodes [8] and is based on Matlab, enables the modeling of any robotic arm provided that the corresponding DH parameters are known. Links visualization in terms of shape and size was made after some considerations. The program was shown to adequately model several example robots.

This paper focuses on modeling and simulation of the RA-02 robotic arm, a small special robot with 4 revolute joints [9]. An excellent survey of robotics kits for post-secondary education is given in [10]. The rest of this paper is organized as follows: Section II below presents the kinematics analysis based on the DH convention and exponentials formulation. The inverse kinematics solution

based on the algebraic method is given in section III. In section IV, the manipulator simulation procedure and results are presented. Section V concludes this work.

II. MANIPULATOR KINEMATICS

The DH convention and product of exponentials often used in kinematics analysis of robot manipulators and other mechanical structures. The below section utilizes both methods to drive the kinematics model of the robot. To simplify the analysis, the effect of the gripper is ignored making the robotic arm a four degree of freedom, a four-joint spatial manipulator. The treatment of this manipulator is analogue to that of PUMA 560 after removing the wrist and adding one more revolute joint.

A. DH Convention.

Denavit-Hartenberg (DH) convention is commonly used in the kinematics analysis of the robotic manipulator [2]. It is based on attaching a coordinate frame at each joint and specifying four parameters known as DH parameters for each link, and utilizing these parameters to construct a DH table. Finally, a transformation matrix between different coordinate frames is obtained. The major objective is to control both the position and orientation of the EE or the gripper in its work space. We will first derive the relationship between the joint variables and the position, and orientation of the gripper, using the DH method [2]. The robotic arm under study is shown below in Fig. 1 (RA-02A of Images SI Inc [9]) and DH parameters in Fig. 2.



Figure 1. The robot arm.

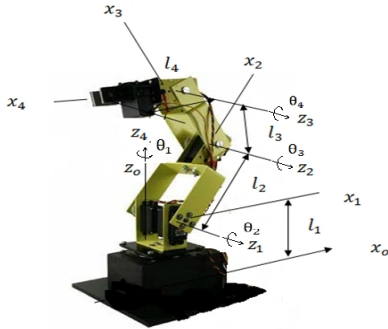


Figure 2. DH parameters.

Various rotary and linear motions of the robotic arm are shown below in Fig 3. Each motion is made possible by a servomotor (or just a servo) at the corresponding location.

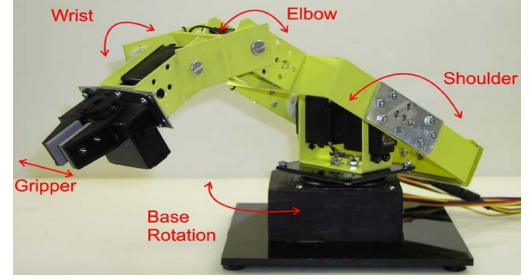


Figure 3. Robotic arm with 5 independent motions [9].

B. Forward Kinematics

As Fig. 2 depicts a coordinate frame attached to every link in order to find its configuration in the neighboring frames using the rigid motion formula. To do so a DH table is needed as follows:

TABLE I. DH PARAMETERS.

Frame (i)	a_i	a_i	d_i	θ_i
1	0	90	l_1	θ_1
2	l_2	0	0	θ_2
3	l_3	0	0	θ_3
4	l_4	0	0	θ_4

By applying the Denavit-Hartenberg (DH) notations [2] for the joints coordinates, the DH-table can be constructed as listed above in Table II. The link lengths as shown in Fig. 2 are $l_1 = 11.5\text{ cm}$, $l_2 = 12\text{ cm}$, and $l_3 = l_4 = 9\text{ cm}$.

C. Forward Transformation Matrices

Once the DH table is ready, the transformation matrices are easy to find. Generally, the matrix of transformation from the frame B_i to the frame B_{i-1} for the standard DH method is given by [2]:

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

Then the individual transformation matrices for $i = 1, 2, 3$, and 4 can be easily obtained. Thereafter, the complete transformation 0T_4 is found from:

$${}^0T_4 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 \quad (2)$$

Upon the determination of, 0T_4 we can find the global coordinates of the end effector. The tip point of the arm is at the origin of frame B_4 (Fig. 2), i.e. it is at $[0 \ 0 \ 0 \ 1]^T$. So, its position in the global frame becomes:

$${}^0r_p = {}^0T_4 {}^4r_p = {}^0T_4 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} r_{14} \\ r_{24} \\ r_{34} \\ 1 \end{bmatrix} = \begin{bmatrix} dx \\ dy \\ dz \\ 1 \end{bmatrix} \quad (3)$$

Obviously, this is the last column of the transformation matrix 0T_4 . After simplifications using trigonometric formulas, the previous equation becomes:

$$dx = c\theta_1 [l_2 c\theta_2 + l_3 c(\theta_2 + \theta_3)] + l_4 c\theta_1 c(\theta_2 + \theta_3 + \theta_4) \quad (4)$$

$$dy = s\theta_1 [l_2 c\theta_2 + l_3 c(\theta_2 + \theta_3)] + l_4 s\theta_1 c(\theta_2 + \theta_3 + \theta_4) \quad (5)$$

$$dz = [l_1 + l_2 s\theta_2 + l_3 s(\theta_2 + \theta_3)] + l_4 s(\theta_2 + \theta_3 + \theta_4) \quad (6)$$

where s stands for \sin and c stands for \cos . In addition, dx , dy and dz are the global end effector coordinates.

Moreover, the end effector orientation is:

$$\phi = \theta_2 + \theta_3 + \theta_4 \quad (7)$$

The DH parameters are akin to the configuration of the robot. For different manipulator structures, the kinematics equations are not unique. Moreover, kinematics equations of the manipulator based on the DH convention provide some singularity making the equations difficult to solve or unsolvable in some cases. In addition, in the DH convention, the common normal is not defined properly when axes of two joints are parallel. In this case, the DH method has a singularity, where a little change in the spatial coordinates of the parallel joint axes can create a huge misconfiguration in representation of the DH coordinates of their relative position. In the following section, an alternative to the DH convention, the PE is presented.

D. Product of Exponentials

Aside from the DH convention, another method is the so-called product of exponentials (PE). Using the reality that the motion of each joint is generated by a twist accompanied with the joint axis, a more geometric representation of the kinematics can be acquired. Remember that if ξ is a twist, according to reference [11], the forward kinematics is given by:

$$g_{st}(\theta) = e^{\hat{\xi}_1 \theta_1} e^{\hat{\xi}_2 \theta_2} \dots e^{\hat{\xi}_n \theta_n} g_{st}(0) \quad (8)$$

The above equation is called the product of exponential formula for the robot forward kinematics, where $g_{st}(0)$ is

the final configuration of the robot and $e^{\hat{\xi}_n \theta_n}$ is a matrix exponential given by [7]:

$$e^{\hat{\xi}_n \theta_n} = \begin{bmatrix} e^{\theta_n \omega_n^{\wedge}} & (I - e^{\theta_n \omega_n^{\wedge}})(\omega_n \times v_n) + \theta_n \omega_n \omega_n^T \theta_n \\ 0 & 1 \end{bmatrix} \quad (9)$$

For a prismatic joint the twist ξ_i is given by: $\xi_i = \begin{bmatrix} v_i \\ 0 \end{bmatrix}$,

and for a revolute joint by: $\xi_i = \begin{bmatrix} -\omega_i \times q_i \\ \omega_i \end{bmatrix}$,

where $\omega_i \in R^3$ is a unit vector in the direction of the axis of the twist, $q_i \in R^3$ is any point on the axis, and $v_i \in R^3$ is a unit vector directing in the translation direction. In this case, the twist ξ 's for different links of the robot are given by:

$$\xi_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \xi_2 = \begin{bmatrix} L_1 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} \quad \xi_3 = \begin{bmatrix} L_1 \\ 0 \\ -L_2 \\ 0 \\ -1 \\ 0 \end{bmatrix} \quad \xi_4 = \begin{bmatrix} L_1 \\ 0 \\ 0 \\ -(L_2 + L_3) \\ -1 \\ 0 \end{bmatrix}$$

Moreover, $g_{st}(0)$ is the initial configuration of the robot given by

$$g_{st}(0) = \begin{bmatrix} 1 & 0 & 0 & l_2 + l_3 + l_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The forward kinematics map of the manipulator has the form:

$$g_{st}(\theta) = e^{\hat{\xi}_1 \theta_1} e^{\hat{\xi}_2 \theta_2} e^{\hat{\xi}_3 \theta_3} e^{\hat{\xi}_4 \theta_4} g_{st}(0) = \begin{bmatrix} R(\theta) & p(\theta) \\ 0 & 1 \end{bmatrix} \quad (10)$$

By expanding terms in the product of exponentials formula, Eq. 10 yields

$$R(\theta) = \begin{bmatrix} \cos(\theta_2 + \theta_3 + \theta_4) \cos \theta_1 & -\sin \theta_1 & -\sin(\theta_2 + \theta_3 + \theta_4) \cos \theta_1 \\ \cos(\theta_2 + \theta_3 + \theta_4) \sin \theta_1 & \cos \theta_1 & -\sin(\theta_2 + \theta_3 + \theta_4) \sin \theta_1 \\ \sin(\theta_2 + \theta_3 + \theta_4) & 0 & \cos(\theta_2 + \theta_3 + \theta_4) \end{bmatrix} \quad (11)$$

$$p(\theta) = \begin{bmatrix} \cos \theta_1 (L_3 \cos(\theta_2 + \theta_3) + L_2 \cos \theta_2 + L_4 \cos(\theta_2 + \theta_3 + \theta_4)) \\ \sin \theta_1 (L_3 \cos(\theta_2 + \theta_3) + L_2 \cos \theta_2 + L_4 \cos(\theta_2 + \theta_3 + \theta_4)) \\ L_1 + L_3 \sin(\theta_2 + \theta_3) + L_2 \sin \theta_2 + L_4 \sin(\theta_2 + \theta_3 + \theta_4) \end{bmatrix} \quad (12)$$

It is clear that the above equation (12) is identical to those in equations (4) through (6). Thus, the kinematics modeling using the DH convention and product of exponentials agrees with each other.

III. INVERSE KINEMATICS

We end up with a set of four nonlinear equations with four unknowns. Solving these equations algebraically, known as the inverse kinematics, requires that we need to know the joint variables $\theta_1, \theta_2, \theta_3$, and θ_4 for a given EE position $[dx, dy, dz]$ and orientation ϕ . We get from equations (4) to (7), by dividing, squaring, adding and using some trigonometric formulas:

$$\theta_1 = \tan^{-1} \left(\frac{dy}{dx} \right) \quad (13)$$

$$\theta_2 = \tan^{-1} \left(c, \pm \sqrt{r^2 - c^2} \right) - \tan^{-1} (a, b) \quad (14)$$

$$\theta_3 = \cos^{-1} \left(\frac{A^2 + B^2 + C^2 - l_2^2 - l_3^2}{2l_2l_3} \right) \quad (15)$$

where

$$a = l_3 \sin \theta_3, b = l_2 + l_3 \cos \theta_3, c = dz - l_1 - l_4 \sin \phi, \text{ and } r = \sqrt{a^2 + b^2}.$$

In addition

$$A = (dx - l_4 c \theta_1 c \phi), B = (dy - l_4 s \theta_1 c \phi), \text{ and } C = (dz - l_1 - l_4 s \phi)$$

Having determined $\theta_1, \theta_2, \text{ and } \theta_3$, we can then find θ_4 from the EE orientation of ϕ as follows:

$$\theta_4 = \phi - \theta_2 - \theta_3 \quad (16)$$

IV. SIMULATION OF THE ROBOTIC ARM

Since robotics is a multidisciplinary subject, its study requires skills from different fields of knowledge, not easily available. Thus, simulation has been recognized as a suitable tool that combines all these features together enabling the user of a direct visualization of different kinds of motion that a robot may perform, making the role of simulation very important in robotics [12]. Using the robotics toolbox together with the Matlab software [13]-[15], the kinematics of a robotic arm can be simulated and analyzed based on the DH convention described before. The toolbox takes a conventional approach to represent the kinematics and dynamics of serial-link robotic arms. Besides, it provides several functions and routines, which are handy for the simulation and scrutiny of robotic manipulators, like kinematics, dynamics and trajectory creation.

Based on the aforementioned toolbox, many configurations of the robot manipulator are easily visualized. In order to simulate the robotic manipulator, first we erect a vector of *Link* objects and insert our four groups of DH parameters as in Table I. Then, these are passed to the constructor *SerialLink*, which is the key step to utilize the toolbox. A detailed procedure can be found in [15]. In order to generate the plot four joint variables are needed. In the first case all joint variables are entered in the form of zeros (row vector), such as $[\theta_1 \ \theta_2 \ \theta_3 \ \theta_4] = [0 \ 0 \ 0 \ 0]$. Fig. 4 shows rest or home position of the robot, where all joints variables are zeros. In the toolbox, each revolute joint is resembled by a small cylinder. Since we have four revolute joints, four cylinders can be seen in Figs. 4-7.

A second interesting configuration is that when all links are standing upright as Fig. 5 shows. The corresponding input in this case is given by: $[0 \ \pi/2 \ 0 \ 0]$, where all the angles are defined in radian by default. To test the functionality of all the joints, other two configurations are selected as given by the vectors of joint variables

$[-\pi/2 \ 0 \ -\pi/2 \ 0]$ and $[\pi \ \pi/4 \ -\pi/2 \ \pi/4]$, which are depicted in Figs. 6 and 7, respectively.

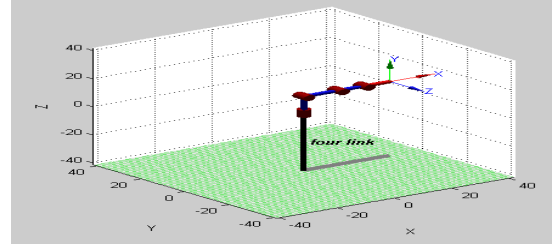


Figure 4. Home position.

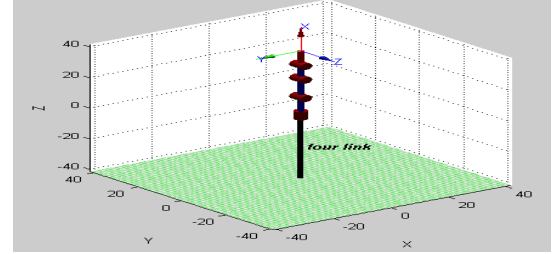


Figure 5. Upright position.

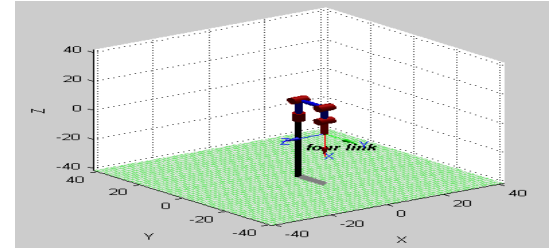


Figure 6. Left-down position.

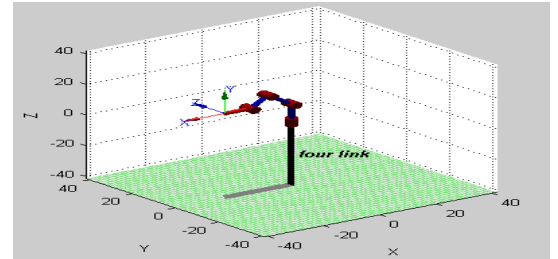


Figure 7. All joints are given angles.

V. CONCLUSION

Kinematics model of a 4 degree-of-freedom robotic arm is presented using both the DH method and product of exponentials formula. It is proven that both approaches provide the same solution for the robot manipulator under study. In addition, the simulation of the robot manipulator is carried out using the Matlab software via the robotics toolbox, through which several positions of the manipulator are realized based on the DH convention. Although results of the product of exponentials formula are not given, they are expected to be same as those of the DH convention.

ACKNOWLEDGMENT

Authors gratefully acknowledge the support provided by King Fahd University of Petroleum & Minerals through the NSTIP project 09-ELE786-04.

REFERENCES

- [1] K. E. Clothier and Y. Shang, "A Geometric Approach for Robotic Arm Kinematics with Hardware Design, Electrical Design, and Implementation," *J. Robot.*, 2010.
- [2] R. N. Jazar, *Theory of Applied Robotics*. Boston, MA: Springer US, 2010.
- [3] "Barinka, Lukas, and Roman Berka. 'Inverse kinematics-basic methods.' Web.< <http://www.cescg.org/CESCG-2002/LBarinka/paper.pdf> (2002)."
- [4] A. Aristidou and J. Lasenby. "Inverse kinematics: a review of existing techniques and introduction of a new fast iterative solver," 2009.
- [5] J. Kaur and V. K. Banga, "Simulation of Robotic Arm having three link Manipulator," *International Journal of Research in Engineering and Technology (IJRET)*, vol. 1, no. 2, March, 2012, ISSN: 2277-4378, 2012.
- [6] X. F. Ge and J. T. Jin, "The algorithm of redundant robotic kinematics based on exponential product," *Appl. Mech. Mater.*, vol. 58-60, pp. 1902-1907, Jun. 2011.
- [7] X. Xiao, Y. Li, and H. Tang, "Department of electromechanical engineering, university of Macau, Macao SAR, China," *IEEE International Conference on Information and Automation (ICIA)*, 2013, pp. 1177-1182.
- [8] A. Lodes, "Robot Simulator in MATLAB," unpublished.
- [9] "<http://www.imagesco.com/kits/robotic-arm.html>."
- [10] M. Ruzzenente, M. Koo, K. Nielsen, L. Grespan, and P. Fiorini, "A review of robotics kits for tertiary education," in *Proceedings of International Workshop Teaching Robotics Teaching with Robotics: Integrating Robotics in School Curriculum*, 2012, pp. 153-162.
- [11] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC press, 1994.
- [12] L. Žlajpah, "Simulation in robotics," *Math. Comput. Simul.*, vol. 79, no. 4, pp. 879-897, Dec. 2008.
- [13] P. I. Corke and others, "A computer tool for simulation and analysis: the Robotics Toolbox for MATLAB," in *Proc. National Conf. Australian Robot Association*, 1995, pp. 319-330.
- [14] P. I. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Berlin: Springer, 2011.
- [15] P. Corke, *Robotics, Vision and Control*, vol. 73. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.