

Axis manipulation to solve Inverse Kinematics of Hyper-Redundant Robot in 3D Space

Sheldon Ijau Winston, Annisa Jamali *

Department of Mechanical and Manufacturing Engineering, Faculty of Engineering, Universiti Malaysia Sarawak, Malaysia

Abstract

A solution based on inverse kinematics is required for the robot's end effector, also known as its tip, to reach a target. Current methods for solving the inverse kinematics solution for a hyper-redundant robot in three 3D are generally complex, difficult to visualise, and time intensive. The condition requires development of new algorithms for solving inverse kinematics more quickly and efficiently. In this study, an axis manipulation using a geometrical approach is used. Initially, a general algorithm for a 2-m-link hyper-redundant robot in 3D is generated. Then, the method employed a repetitive basic inverse kinematics solution of a two-link robot on virtual links. Next, the virtual links are generated using a specific geometric proposition. Finally, the 3D solution is generated by rotating about the global z-axis. This method reduces the mathematical complexity required to solve the inverse kinematics solution for a 2-m-link robot. In addition, this method can manage variable link manipulators, thereby eliminating singularity. Numerical simulations of hyper redundant models in 3D are presented. This new geometric approach is anticipated to enhance the performance of hyper-redundant robots, enabling them to be of more significant assistance in fields such as medicine, the military, and search and rescue.

Keywords:

3D Space;
Hyper-redundant;
Inverse kinematic;
Robot;
Variable length links;

Article History:

Received: August 22, 2022
Revised: September 16, 2022
Accepted: September 20, 2022
Published: September 24, 2022

Corresponding Author:

Annisa Jamali,
Mechanical and Manufacturing
Engineering Department,
Universitas Malaysia Sarawak,
Malaysia
Email: jannisa@unimas.my

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license



INTRODUCTION

A robot with several actuated degrees of freedom exceeding the minimum number of actuated degrees needed to perform a task is known as a hyper-redundant robot (HRR.). A redundant degree of freedom has been recognised to improve manipulator performance in complex and unstructured environments tremendously. The various abilities of hyper-redundant manipulators, such as obstacle avoidance, manoeuvrability in inaccessible areas, and reconfigurable joints in the event of a failed joint, are the factors that make the application of HRR. in a variety of fields extremely advantageous. This is demonstrated by the fact that HRR has been utilised in various fields. For instance, HRR is used in military operations such as deploying missiles and area surveillance [1].

Furthermore, in case of a disaster such as hurricanes or earthquakes, the HRR is very useful in performing search and rescue of trapped victims rescue [2]. HRR may also be used to inspect bridges, which saves the time of inspection, inspecting nuclear power plants and spacecraft, and cleaning the machine, such as inside the steam pipes, vessels, and various narrow spaces [3, 4, 5]. Thus, to ensure the system's efficacy, it is essential to continue researching the algorithm for a hyper-redundant manipulator capable of handling the previously described complex scenario.

Inverse Kinematics (IK) is a crucial algorithm for redundant manipulator analysis, which is essential for robot control. IK is used to determine the actuator's variables to determine the end effector's position and orientation. In order to solve an inverse kinematics problem, it can be classified into three groups: an algebraic approach, an iterative approach, and a geometric approach. The algebraic strategy includes Jacobian, Grobner bases, the product of exponential

[6], analytical inverse kinematics [7][8], conformal geometric algebra [9], and generalised analytical methods [10]. In the iterative approach, several strategies have been proposed, such as Artificial Neural Network [11][12], iterative algorithm [13], genetics algorithm, ANFIS, Fuzzy Logic, evolution algorithm (EA) [14][15], and iterative algorithm [16][17]. The geometric approach offers a simpler calculation such as repetitive basic inverse kinematics of a two-link [18], the angular deflection of the upper platform concerning the lower platform [19], a non-iterative geometric approach for inverse kinematics [20], and a modified modal method [21][22].

Typically, the algebraic method is burdened by a challenging symbolic expansion and a heavy computational load. In addition, although this method provides several potential solutions with different arm configurations, there is no clear indication for choosing the optimal solution. In contrast, the iterative method requires significant computation time, and it takes considerable time to arrive at a helpful inverse kinematics (IK) solution. Furthermore, error minimisation is always a research topic due to the many possible solutions offered by this technique. Finally, the geometric approach uses geometric heuristics to find the angle to take advantage of the manipulator's unique structure. The geometric approach, therefore, provides a direct solution.

This paper proposes a new geometrical approach to solve the inverse kinematics of HRR in 3D using Axis manipulation after employing a repetitive basic inverse kinematics solution of a two-link robot on virtual links. First, virtual links are generated using a particular geometric proposition. Then, the 3D solution is generated by rotating the global z-axis.

MATERIAL AND METHOD

This section elaborated on the details of the algorithm. The first section delves into the specifics of the general inverse kinematics solution for 2-link. Following is the complete algorithm for solving the inverse kinematics of a hyper-redundant robot with multiple links.

General Inverse Kinematics of 2- link

The development of the inverse kinematics algorithm for HRR is based on the repetitive computation of the basic 2-link inverse kinematics solution along the HRR [Figure 1](#) shows a primary elbow-down 2-link manipulator. [Figure 1](#) shows two angles, θ_1 for the angle between x_0 and link a_1 , and θ_2 for the angle between link a_1 and a_2 . Let the position of the end effector, $P = (x, y)$, and origin, $O = (0, 0)$. The calculations for obtaining the angles are as follows.

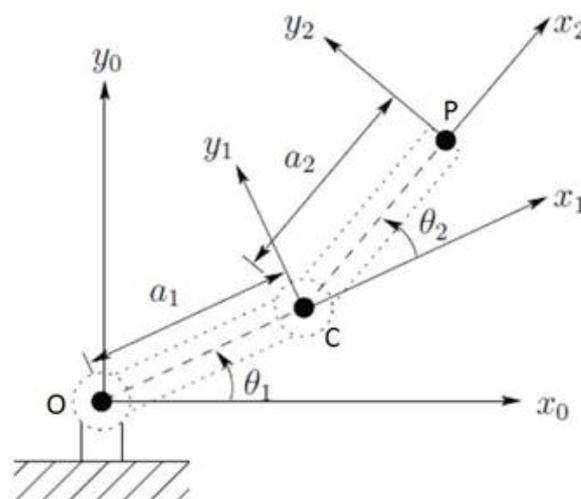


Figure 1. Basic Elbow Down 2-Link Manipulator

$$x = a_1 \cos \theta_1 + a_2 \cos (\theta_1 + \theta_2) \quad (1)$$

$$y = a_1 \sin \theta_1 + a_2 \sin (\theta_1 + \theta_2) \quad (2)$$

To find the value of θ_2 , the cosine rule is used resulting.

$$x^2 + y^2 = a_1^2 + a_2^2 - 2a_1a_2 \cos OCP \cos \theta_2 = -\cos OCP \quad (3)$$

$$x^2 + y^2 = a_1^2 + a_2^2 + 2a_1a_2 \cos \theta_2 \quad (4)$$

$$\therefore \theta_2 = \cos^{-1} \frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1a_2} \quad (5)$$

To find the value of θ_1 , the value from θ_2 and the cosine rule are used, thus resulting.

$$\therefore \theta_1 = \tan^{-1} \frac{y^2}{x^2} \pm \tan^{-1} \frac{a_2 \sin \theta_2}{a_1 + a_2 \cos \theta_2} \text{ (elbow up and down)} \quad (6)$$

The basic inverse kinematics solution for the 2-link manipulator was used to develop the inverse kinematics solution for HRR in 3D Space.

Geometric Approach Algorithm

Following are the steps of the proposed geometrical algorithm.

Step 1: Define the number of links - First, the number of links of the HRR is defined. If the number of links is other than 2m, then the algorithm will round down the number of links to the lowest 2-m, and then the excess links are locked to the neighbour link.

Step 2: Define the lengths of each link- Secondly, the length of each link in the HRR is defined. If the HRR has constant link lengths, then the length of each link is only defined once. If the HRR has various link lengths, then the length of each link must be defined. The length of each link can be stored in a matrix form as follows.

$$L = [l_1 \ l_2 \ l_3 \ \dots \ l_{2m}] \quad (7)$$

The methods used in completing the research are written in this section. The method includes research chronological, including research design, research procedure (in the form of algorithms, Pseudocode, or other), instruments, and analysis techniques used in solving problems. The description of the course of research should be supported by references so that the explanation can be accepted scientifically [13].

Step 3: Define the desired end effector location. Then, the desired location of the end effector is then defined. Let the coordinate of the desired end effector location as (x, y, z). Then, the whole length of the HRR (sum of all the lengths with origin at x = y = z = 0 is rested along the x-axis.

Step 4: Compute the angle between the plane of desired end effector location to the x-z plane- Then, the 2D inverse kinematics solution is then carried out on the HRR on the x-z plane. Before the algorithm's execution, an angle from the robot's 2D plane to the x-z plane is calculated using the formula in (8).

$$\theta_{xz} = \tan^{-1} \frac{y}{x} \quad (8)$$

The coordinate of the desired end effector location is projected onto the x-z plane by using these formulae, resulting in the projected end effector's location on the x-z plane as (xxz, 0, zxz):

$$x_{xz} = \frac{x}{\cos \theta_{xz}}, \ y_{xz} = 0, \ z_{xz} = z \quad (9)$$

Step 5: Compute the 2D inverse kinematics solution on the x-z plane- The whole length of the HRR is divided into two parts, making it a temporary basic 2-link robot. The end effector is

then moved to the desired end effector location. The robot's elbow location is then determined using the cosine rule. An example of a 4-link HRR was chosen from Figure 2 to explain this algorithm. \mathbf{b} is the location of the elbow, and \mathbf{c} is the location of the desired end effector projected onto the x-z plane. The length C is simply the total length of the first half of the link, and length A is simply the total length of the second half of the link. The length B is determined by using as in (10):

$$B = \sqrt{x_{xz}^2 + z_{xz}^2} \quad (10)$$

The angle a_1 and a_2 are determined as in (11):

$$a_1 = \cos^{-1} \frac{A^2 - B^2 - C^2}{-2BC}, \quad a_2 = \tan^{-1} \frac{z_{xz}}{x_{xz}} \quad (11)$$

Hence, the location of the elbow \mathbf{b} on the x-z plane can then be determined by using the following:

$$b_x = C \cos(a_1 + a_2), \quad b_z = \frac{C}{\cos(a_1 + a_2)} \quad (12)$$

After the elbow's location is determined, the elbow will move to the centroid of the HRR. The centroid of the elbow can be determined by using this formula:

$$x_{centroid} = \frac{O_x + b_x + c_x}{3} \quad (13)$$

The result from the centroid formula should make the position of the HRR to be as shown in Figure 3.

Moving the elbow to the centroid would create two smaller 2-link robots with elbows \mathbf{d} and \mathbf{e} , as shown in Figure 3. Then the location of each elbow is then calculated similarly to calculating the location of the elbow for Figure 4. Next, the zig-zag shape of the HRR will then be turned into a coil shape. The method for making it a coil shape is to place the elbows that are at the centroids diagonal opposite the line $\overline{\mathbf{de}}$ to \mathbf{g} .

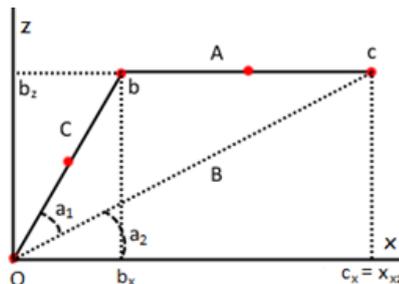


Figure 2. Determination of the Elbow Location

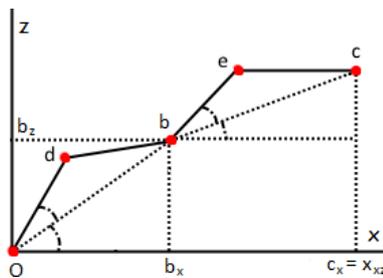


Figure 3. Location of the Elbow at the Centroid of the HRR.

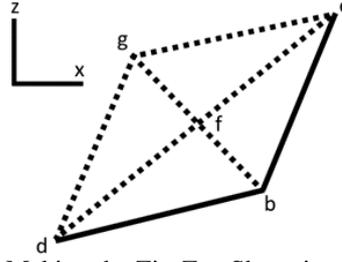


Figure 4. Making the Zig-Zag Shape into a Coil Shape

The coiling algorithm first calculates the angle between the line. These are the equations that calculate the angle:

$$\overline{de} = (e_x - d_x)\hat{i} + (e_z - d_z)\hat{k}, \quad \overline{dg} = (g_x - d_x)\hat{i} + (g_z - d_z)\hat{k}, \quad \cos \theta_{coil} = \frac{\overline{de} \cdot \overline{dg}}{|\overline{de}| |\overline{dg}|} \quad (14)$$

$$\overline{df} = \overline{dg} \cdot \cos \theta_{coil} \quad (15)$$

Finding the coordinate f from Figure 4 is necessary for finding the location of g. The following is the method for finding the coordinate of f and then g, and Figure 5 shows the final result of the coiling algorithm:

$$f_x = \frac{|\overline{df}|}{|\overline{de}|}(e_x - d_x) + d_x, \quad f_y = \frac{|\overline{df}|}{|\overline{de}|}(e_y - d_y) + d_y \quad (16)$$

$$g_x = 2f_x - b_x, \quad g_y = 2f_y - b_y \quad (17)$$

The calculations that were shown so far are only for the end effector position in the positive z-direction, in which the configuration is the elbow-up configuration. For the negative z-direction, the elbow-down configuration is used instead. The calculations for the elbow down configuration are similar to the elbow up configuration.

Step 6: Rotate the 2D plane to the desired end effector location- After the computation of the 2D inverse kinematics solution, then the HRR is rotated to the desired end effector plane, with the end effector's location at (x, y, z) by the following formula:

$$x = x_{xz} \cdot \cos \theta_{xz}, \quad y = y, \quad z = z_{xz} \quad (18)$$

Step 7: Display the HRR- After all the computation, the visual representation of the final position of the HRR in 3D is then displayed.

The flowchart shown in Figure 6 summarised the proposed algorithm.

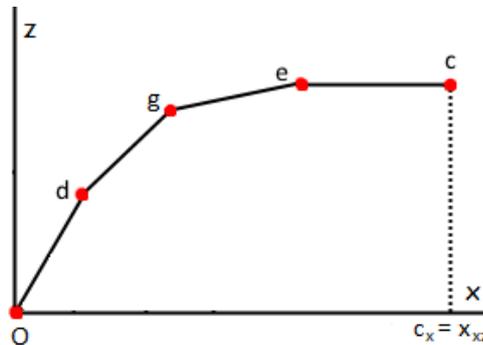


Figure 5. Coil-Shaped H.R.R.

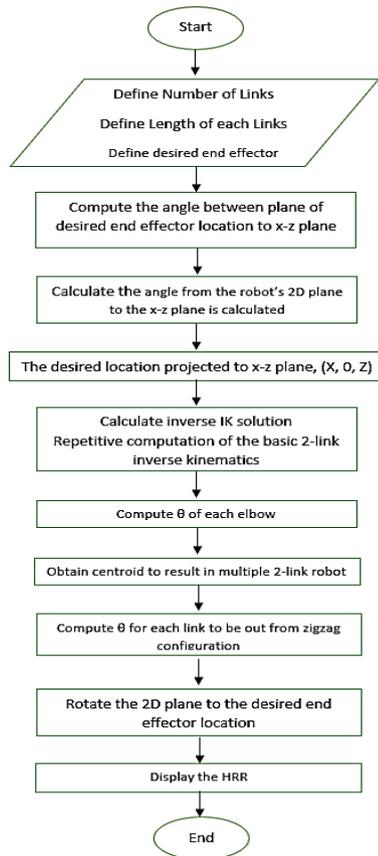


Figure 6. Flowchart of the proposed algorithm

RESULTS AND DISCUSSION

Simulations were carried out with different target positions at four other quadrants for both positive z-direction and negative z-direction. For each quadrant, three tests were carried out for constant link lengths with various units, and one was carried out for various link lengths. Figure 7 (a) shows the simulation results for the proposed algorithm for the end effector in a positive z-direction for the first quadrant. The top left section is the 4-link HRR, with each link having a length of 1 unit and the desired end effector location at (2, 1, 1). The top right corner is the 8-link HRR, with each link having a length of 1.1 units and the desired end effector location at (5.5, 4.3, 2). The bottom left section is the 16-link HRR, with each link having a length of 0.5 units and the desired end effector location at (3, 4, 1). Finally, the bottom right section is the 4-link HRR with various link lengths (one unit, two units, 1 unit, and two units, respectively, from the origin to the end effector) with the desired end effector location at (3, 2, 1).

Meanwhile, Figure 7 (b) shows the simulation results for the proposed algorithm for the end effector in the negative z-direction for the first quadrant. Similarly, three tests were carried out for constant link lengths, and one test was carried out for various link lengths. The top left section is the 4-link HRR, with each link having a length of two units and the desired end effector location at (5, 3, -1). The top right corner is the 8-link HRR, with each link having a length of 1.1 units and the desired end effector location at (6, 4.1, -1). The bottom left section is the 16-link HRR, with each link having a length of 0.5 units and the desired end effector location at (5, 1, -2). Finally, the bottom right section is the 4-link HRR with various link lengths (1.1 unit, 1.2 units, 1.3 unit, and 1 unit, respectively, from the origin to the end effector) with the desired end effector location at (2.5, 1.3, -0.5).

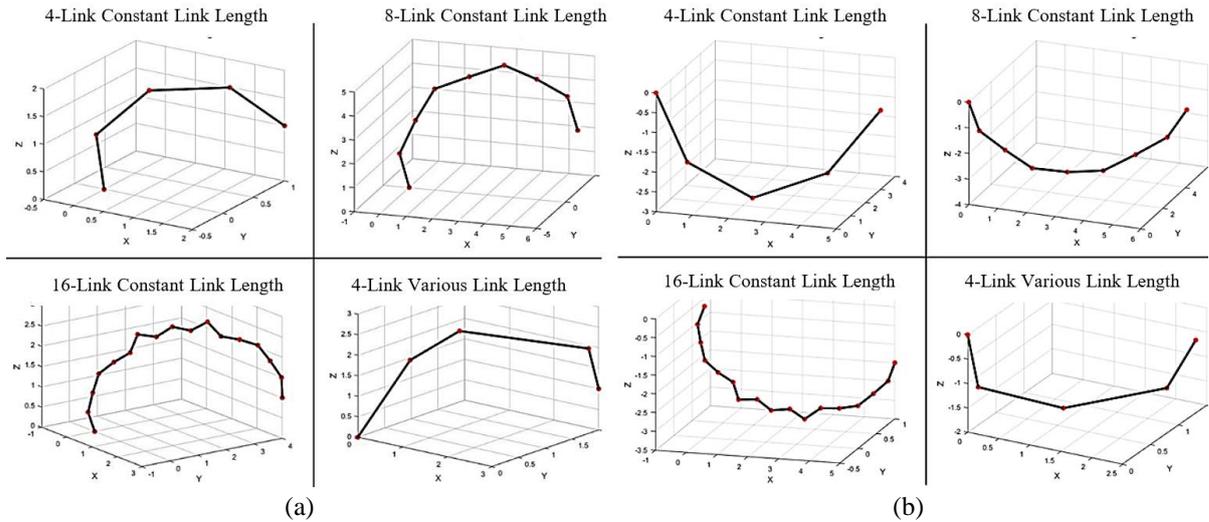


Figure 7. First Quadrant Results for End Effector in (a) Positive z-Direction (b) Negative z-Direction

Likewise, the same tests were carried out for other quadrants. All the results for the different quadrants were tabulated in Table 1.

Figure 8 (a)-(b), Figure 9 (a)-(b) and Figure 10 (a)-(b) represent the results for the 2nd, 3rd, and 4th quadrants, respectively. From the results, it shows that the proposed algorithm was able to be successfully implemented in each quadrant and both positive and negative z-direction in 3D Space. The simulation also accommodated the elbow up and elbow down configuration. Besides that, the simulation was also able to include various link lengths. However, the simulation was only limited to 2-m-links. The results from the simulation show that the constant link length of 4-link HRR, various link lengths of 4-link HRR, and constant link length of 8-link HRR were consistently able to form a coil shape in every test. The various link length of 8-link HRR was able to form a coil shape for certain link lengths and end effector locations, such as in Figure 8(a)- (b). The constant link length of 16-link HRR was only able to partially coil the shape at the two ends of the robot for every test. The constant link length of 4-link HRR, various link length 4-link HRR, and constant link length of 8-link HRR were able to avoid singularity better than the various link length of 8-link HRR and constant link length of 16-link HRR. The various link length of 8-link HRR and constant link lengths of 16-link HRR have the potential not to be able to reach a desired end effector location because the zig-zag shape increases the likelihood of singularity to occur. Singularity is the inability of the end effector to move in a certain direction, no matter how the robot moves. There were also some limitations when the desired end effector location gets closer to the origin, which increases the likelihood of singularity.

Table 1. Simulation Results for different quadrants in the positive and negative z-direction with various links

		Target position in 3D			
		Top left section 4-link HRR	Top right section 8-link HRR	Bottom left section 16-link HRR	Bottom right section m-link HRR. with various link length
1st Quadrant	(+ ve z)	(2, 1, 1)	(5.5, 4.3, 2)	(3, 4, 1)	(3, 2, 1)
	(- ve z)	(5, 3, -1)	(6, 4.1, -1)	(5, 1, -2)	(2.5, 1.3, -0.5)
2nd Quadrant	(+ ve z)	(-1, 3, 0.2)	(-1.1, 0.9, 1.2).	(-12, 2, -5)	(-6, 4, 1)
	(- ve z)	(-1, 3, -0.2)	(-10, 4, -4)	(-7, 4, -2)	(-6, 2, -4)
3rd Quadrant	(+ ve z)	(-2, -2, 0.5)	(-7, -4, 5)	(-10, -7, 4)	(-1, -3, 1)
	(- ve z)	(-3, -4, -2)	(-10, -6, -4)	(-10, -7, -5)	(-3, -1, -1)
4th Quadrant	(+ ve z)	(5, -4, 2)	(8, -6, 5)	(20, -11, 5)	(7, -5, 4)
	(- ve z)	(4, -2, -1)	(6, -7, 5)	(10, -10, -2)	(6, -5, -3)

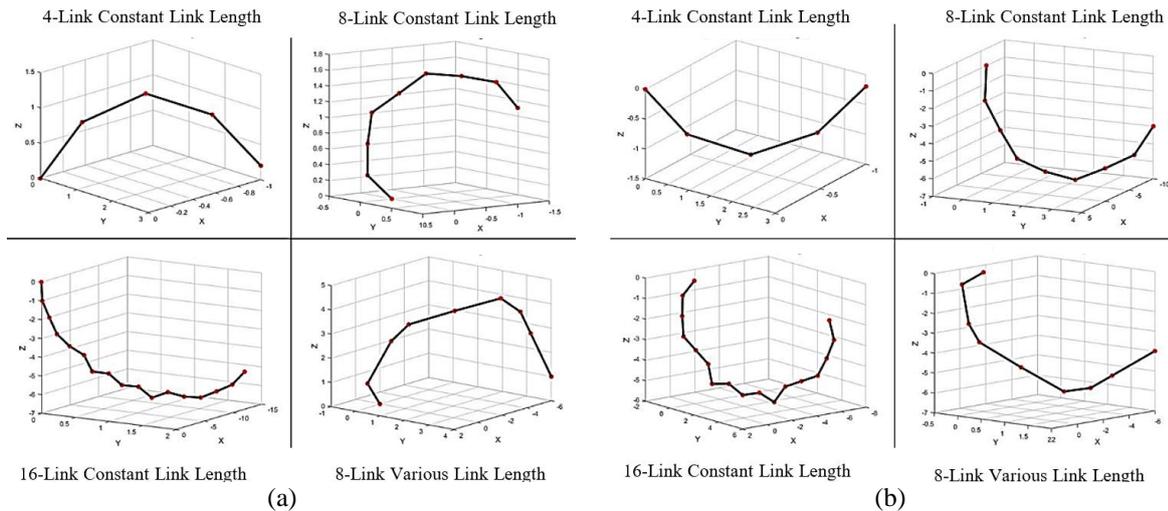


Figure 8. Second Quadrant Results for End Effector in (a) Positive z-Direction (b) Negative z-Direction

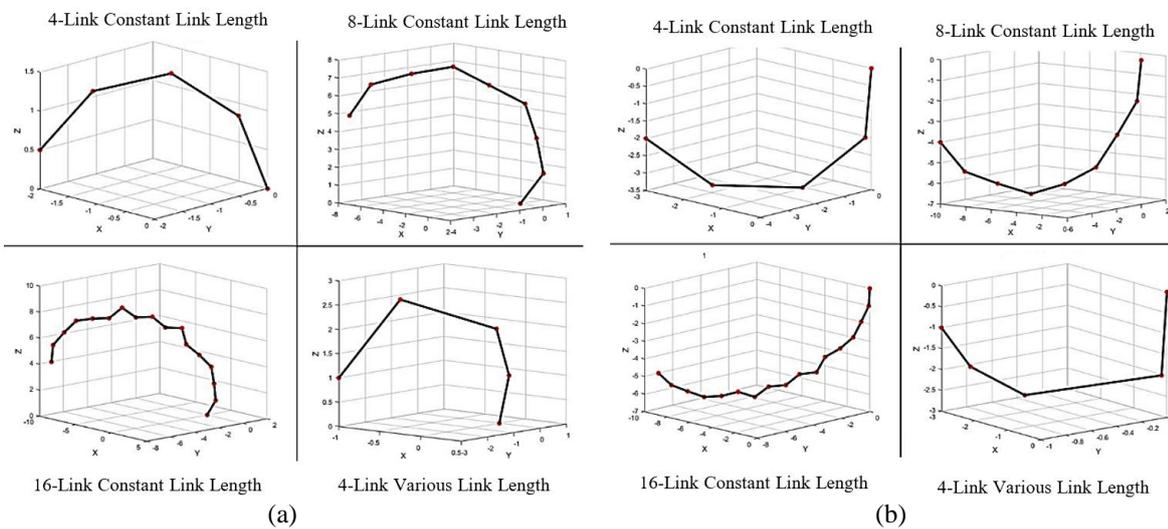


Figure 9. Third Quadrant Results for End Effector in (a) Positive z-Direction (b) Negative z-Direction

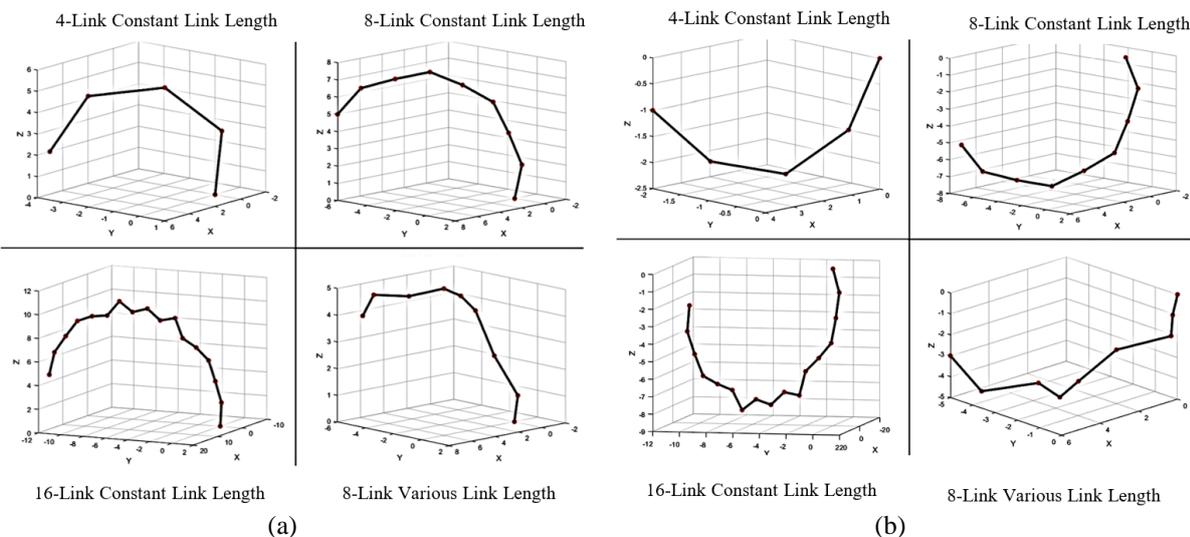


Figure 10: Fourth Quadrant Results for End Effector in (a) Positive z-Direction (b) Negative z-Direction

Besides that, any number of links that are not 2-m will not be able to be used for the calculations. Instead, the links would have to be forced to be locked to another link in the simulation, so the HRR becomes a 2-m number of links, for example, a 5-link HRR will have one of its links locked to another link to become one larger link, so it turns into a 4-link HRR. The advantage of this algorithm is there is no need to choose a solution from many different solutions. Instead, this algorithm provides a single solution. Furthermore, at the lower number of links, this algorithm can prevent singularity from occurring.

CONCLUSION

The algorithm for inverse kinematics of HRR in 3D was developed by further expanding the previous research, which was only studied for 2D. The algorithm for this research includes rotating the solved 2D inverse kinematics solution around the z-axis to the desired end effector location. The algorithm was simulated successfully in MATLAB. The inverse kinematics solution of the HRR was validated by comparing the results of this research to the previous research. Then the result was repeated for different quadrants to further show the algorithm's effectiveness. The HRR can have a continuous curve configuration for constant link length HRR of 8 links and below, a zig-zag configuration for a greater number of links, and certain HRR with various link lengths. The HRR was applied to every octant of the 3D Space, with both elbows up and elbows down configuration for positive z-direction and negative z-direction, respectively.

ACKNOWLEDGMENT

This research was supported by Universiti Malaysia Sarawak through the Small Grant Scheme (Id No. SGS/02(S115)/897/2012(28). The authors wish to thank Universiti Malaysia Sarawak (UNIMAS) and the Minister of Education Malaysia (MOE) for funding and providing facilities to conduct this research.

REFERENCES

- [1] J. Peng, W. Xu, F. Wang, Y. Han and B. Liang, "A Hybrid Hand-Eye Calibration Method for Multilink Cable-Driven Hyper-Redundant Manipulators," in *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-13, 2021, Art no. 5010413, doi: 10.1109/TIM.2021.3078523
- [2] Z. Huang et al., "Impact of caudal fin geometry on the swimming performance of a snake-like robot," *Ocean Engineering*, vol. 245, ID: 110372, 2022, doi: 10.1016/j.oceaneng.2021.110372
- [3] X. Yang et al., "The snake-inspired robots: a review," *Assembly Automation*, vol. 42, no. 4, pp. 567-583, 2022, doi: 10.1108/AA-03-2022-0058
- [4] A. Martín-Barrio, J. J. Roldán-Gómez, I. Rodríguez, J. del Cerro & A. Barrientos, "Design of a Hyper-Redundant Robot and Teleoperation Using Mixed Reality for Inspection Tasks," *Sensors*, vol. 20, no. 8, pp. 2181, 2020, doi: 10.3390/s20082181
- [5] L. Liao and J. Luo, "Design and implementation of a snake-like robot with biomimetic 2-D gaits," *International Journal of Applied Science and Engineering*, vol. 18, no. 5, pp. 1-8, 2021, doi: 10.6703/IJASE.202109_18(5).001
- [6] L. Tang, J. Wang, Y. Zheng, G. Gu, L. Zhu & X. Zhu, "Design of A Cable-Driven Hyper-Redundant Robot with Experimental Validation," *International Journal of Advanced Robotic Systems*, vol. 14, no. 5, 2017, doi: 10.1177/172988141773445
- [7] J. Oh, H. Bae and J. -H. Oh, "Analytic Inverse Kinematics Considering the Joint Constraints and Self-Collision for Redundant 7DOF Manipulator," *2017 First IEEE International Conference on Robotic Computing (IRC)*, 2017, pp. 123-128, doi: 10.1109/IRC.2017.46
- [8] L. Zhao, J. Zhao, H. Liu & D. Manocha, "Collision-Free Kinematics For Hyper-Redundant Manipulators In Dynamic Scenes Using Optimal Velocity Obstacles," *International Journal of Advanced Robotic Systems*, vol. 18, no. 1, pp. 1-17, 2021, doi: 10.1177/1729881421996148
- [9] Z. Iklima, M. I. Muthahhar, A. Khan & A. Zody, "Self-learning of delta robot using Inverse Kinematics and Artificial Neural Networks," *SINERGI*, vol. 25, no. 3, pp. 237-244, 2021, doi: 10.22441/sinergi.2021.3.001

- [10] T. Sun, S. F. Yang, T. Huang & J. S. Dai, "A Generalised and Analytical Method To Solve Inverse Kinematics Of Serial And Parallel Mechanisms Using Finite Screw Theory," *Mechanisms and Machine Science*, vol. 50, pp. 602–608, 2018, doi: 10.1007/978-3-319-60867-9_69
- [11] K. K. Dash, B. B. Choudury & S. K. Senapati, "An Inverse Kinematic Solution of a 6-DOF Industrial Robot Using ANN," *Indian Journal of Scientific Research*, vol. 16, no. 2, pp. 103-106, 2017
- [12] B. Dato, M. Gleizes & F. Migeon, "Cooperative Neighborhood Learning: Application to Robotic Inverse Model," *13th International Conference on Agents and Artificial Intelligence (ICAART 2021)*, Feb 2021, Online Streaming, France, 2021, pp. 368-375
- [13] F. Yuan, Y. Zhao, Y. Zhang, X. Zhang & X. Lu, "An iterative algorithm for inverse displacement analysis of Hyper-redundant elephant's trunk robot," *Robotica*, vol. 40, no. 10, pp. 3539-3556, 2022, doi: 10.1017/S026357472200039X
- [14] C. J. Lin, C. S. Chen & S. K. Yu, "A GPU-based evolution algorithm for motion planning of a redundant robot," *International Journal of Robotics and Automation*, vol. 2, no. 2, pp. 45-57, 2017, doi: 10.15406/iratj.2017.02.00015
- [15] V. V. M. J. S. Chembuly & H. K. Voruganti, "An efficient approach for inverse kinematics and redundancy resolution of spatial redundant robots for cluttered environment," *SN Applied Sciences*, vol. 2, no. 6, ID: 1012, 2020, doi: 10.1007/s42452-020-2825-x
- [16] J. Li, H. Yu, N. Shen, Z. Zhong, Y. Lu & J. Fan, "A novel inverse kinematics method for 6-DOF robots with non-spherical wrist," *Mechanism and Machine Theory*, vol. 157, ID: 104180, 2021, doi: 10.1016/j.mechmachtheory.2020.104180
- [17] S. Xie, L. Sun, G. Chen, Z. Wang and Z. Wang, "A Novel Solution to the Inverse Kinematics Problem of General 7R Robots," in *IEEE Access*, vol. 10, pp. 67451-67469, 2022, doi: 10.1109/ACCESS.2022.3184451
- [18] A. Jamali, R. Khan and M. M. Rahman, "A new geometrical approach to solve inverse kinematics of hyper redundant robots with variable link length," *2011 4th International Conference on Mechatronics (ICOM)*, 2011, pp. 1-5, doi: 10.1109/ICOM.2011.5937183
- [19] M. M. Billah & M. R. Khan, "Smart Tendon Actuated Flexible Actuator," *Journal of Robotics*, vo. 2015, no. 2, pp. 1-10, 2015, doi: 10.1155/2015/295410
- [20] O. M. Omisore, S. Han, L. Ren, N. Zhang, K. Ivanov, A. Elazab & L. Wang, "Non-iterative geometric approach for inverse kinematics of redundant lead-module in a radiosurgical snake-like robot," *Biomedical Engineering Online*, vol. 16, no. 1, pp. 93, 2017, doi: 10.1186/s12938-017-0383-2
- [21] M. N. Muftah, W. L. Xuan and A. 'A. M. Faudzi, "ARX, ARMAX, Box-Jenkins, Output-Error, and Hammerstein Models for Modeling Intelligent Pneumatic Actuator (IPA) System," *Journal of Integrated and Advanced Engineering (JIAE)*, vol. 1, no. 2, pp. 81-88, 2021, doi: 10.51662/jiae.v1i2.18
- [22] W. Xu, Z. Mu, T. Liu, & B. Liang, "A Modified Modal Method for Solving the Mission-oriented Inverse Kinematics of Hyper-redundant Space Manipulators for On-orbit Servicing," *Acta Astronautica*, vol. 139, pp. 54–66, 2017, doi: 10.1016/j.actaastro.2017.06.015