ENERGY ENGINEERING AND CONTROL SYSTEMS

Volume 8, Number 2, 2022

Synthesis of Control Algorithm for Position of Six–Axis Manipulator

Maxym Vihuro[a], Andriy Malyar[a,b,*]

[a]*Lviv Polytechnic National University, 12 Stepan Bandera Street, Lviv, 79013, Ukraine*
[b]*Bydgoszcz University of Science and Technology, Prof. S. Kaliskiego Avenue, 7, Bydgoszcz, 85-796, Poland*

Received: October 15, 2022. Revised: November 05, 2022. Accepted: November 14, 2022.

© 2022 The Authors. Published by Lviv Polytechnic National University.

Abstract

The paper formulates the inverse kinematic problem for the robotic manipulator with six degrees of freedom. For the solution if this problem, the geometric method combined with the Denavit–Hartenberg transformation was applied. The Denavit–Hartenberg method offers the advantage of reducing the number of coordinates that determine the special position of the solid body from six to four. This method provides for an accurate positioning of the working tool. The inverse kinematic problem was solved. This problem aims at calculating the rotating angle of each axis. The geometric solution of the problems for each of the axes is presented. Based on the calculation data, the algorithm of determining the rotating angles of the robotic manipulator was developed. This algorithm was implemented in the Matlab environment. The control flow chart of the algorithm is presented and its operation is described. The paper offers an example of solving the inverse kinematic problem using the developed algorithm. The calculation results were verified and shown to be consistent with the preset position, which confirms the adequacy of the developed model.

Keywords: robotic manipulator; working tool; coordinate system; travel path; Denavit–Hartenberg method; inverse kinematics; generalized coordinates.

1. Introduction

Industrial enterprises are actively automating the manufacturing process on a global scale. The leading plants in all the industries have replaced manual labour with robotic manipulators and other robotized complexes. The demand for industrial robots is increasing annually, which is giving rise to numerous studies in this field. Ukraine’s industry also needs reformation of the machine building complex to be competitive in terms of goods quality and price, which is why domestic research and development of domestic robotic equipment for complete automation of the manufacturing process are important.

In this field, the synthesis of the control algorithm for the movement of the manipulator is important as it determines the accuracy and speed of operation and, consequently, the quality of the products. Therefore, the aim is to study the robotic manipulator and compute the necessary movements of its links.

2. Formulation of the research problem

One of the tasks in creating a control system for the robot is developing the control algorithm [1]. To do this, the inverse kinematic problem (IKP) is solved. It consists in determining the variable parameters of the manipulator for the known geometric parameters of the links, which ensure the preset positions and orientation of the gripper relative to the absolute coordinate system. The IKP consists in calculating a set of generalized coordinates of the manipulator

[*] Corresponding author. Email address: andrii.v.maliar@lpnu.ua

This paper should be cited as: M. Vihuro, A. Malyar. Synthesis of control algorithm for position of six–axis manipulator. *Energy Engineering and Control Systems*, 2022, Vol. 8, No. 2, pp. 118 – 126. https://doi.org/10.23939/jeecs2022.02.118

for the preset position coordinates and orientation of the final coordinate system connected with the gripper or working tool [2], [3].

This problem is more complicated than the forward kinematic problem (FKP) as it can lead to the uncertainty of solution, i.e. different configurations of the robot can correspond to one and the same set of working tool special positions. Besides, IKP solution significantly depends on the manipulator design, which rules out the possibility to develop the only method of solving IKP in general for all robot types.

To develop the algorithm of computing the rotating angles of the six–axis robotic manipulator, the following tasks need to be fulfilled:

- bridging the coordinate system with the links;
- determination of Denavit–Hartenberg parameters;
- building homogeneous transformation matrices;
- solving the inverse kinematic problem by applying the geometric method using trigonometric formulae.
- developing the algorithm for finding the rotating angles for the robotic links based on the IKP solution.

The first three steps were discussed in [4].

The six–axis manipulator design is widely used due to the functional features and kinematic analysis options. Research of robotic manipulators is of topical importance, which is evidenced by a large number of papers in this field [5]–[7].

The kinematic problems were solved in [8], [9]. Quite often, researches focus on robots with up to three links. There are several ways of solving the inverse kinematic problem: mathematical [11], using neural networks [12] or geometric [13]. These techniques are combined with the Denavit–Hartenberg method and Euler-Lagrange method [9], [10], [14], [15]. The geometric (analytical) method of solving IKP relies on finding analytical expressions in an explicit form using trigonometric functions, taking into consideration the kinematic scheme of the manipulator.

As the recent studies show, solving the inverse kinematic problem with further development of the optimized control algorithm is a topical task, which will expedite the improvement in positioning control systems for robotic manipulators.

3. Solving the inverse kinematic problem

The paper describes the solution of the inverse kinematic problem and control algorithm for the six-axis welding robotic manipulator (Fig.1).

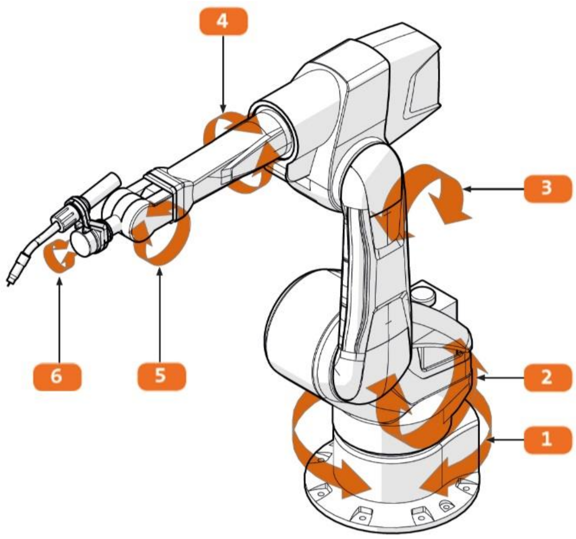


Fig.1. The welding robotic manipulator with six degrees of freedom (1–6).

This inverse kinematic problem aims at calculating the angle θ_i for each joint. The angle θ_i is the rotating angle of the manipulator axis. The initial data is the Denavit–Hartenberg constant parameters, which depend on the robot design

(d_i, α_i, a_i), and matrix T_0^6 of the final coordinates and working tool orientation. The detailed review of the Denavit–Hartenberg method was made in the previous paper [3]. The calculation of angles θ_i will start from the first joint.

To find the angle θ_1 depicted in Fig.2 we need to consider the coordinates of the position vector p_{0x}^4, p_{0y}^4 of the transformation matrix T_0^4 of the fourth link of the robot (1). To make transformation from the matrix

$$T_0^6 = \begin{pmatrix} n_{x6} & o_{x6} & a_{x6} & p_{x6} \\ n_{y6} & o_{y6} & a_{y6} & p_{y6} \\ n_{z6} & o_{z6} & a_{z6} & p_{z6} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

to the coordinate data, the equation (2) is used.

$$T_0^4 = \begin{pmatrix} n_{0x}^4 & o_{0x}^4 & a_{0x}^4 & p_{0x}^4 \\ n_{0y}^4 & o_{0y}^4 & a_{0y}^4 & p_{0y}^4 \\ n_{0z}^4 & o_{0z}^4 & a_{0z}^4 & p_{0z}^4 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \quad (1)$$

$$P_0^4 = P_0^6 - A_0^6 \cdot d_6 = \begin{pmatrix} p_{0x}^6 \\ p_{0y}^6 \\ p_{0z}^6 \\ 1 \end{pmatrix} - \begin{pmatrix} a_{0x}^6 \\ a_{0y}^6 \\ a_{0z}^6 \\ 0 \end{pmatrix} \cdot d_6 = \begin{pmatrix} p_{0x}^4 \\ p_{0y}^4 \\ p_{0z}^4 \\ 1 \end{pmatrix}, \quad (2)$$

where P_0^6 is the working tool position vector; A_0^6 is the working tool approach vector; d_6 is the distance between the axes x_5 and x_6 along the axis z_5.

The vector P_0^4 describes the robotic manipulator position without considering the wrist. The vectors p_{0x}^4, p_{0y}^4 suffice to determine the angle θ_1.

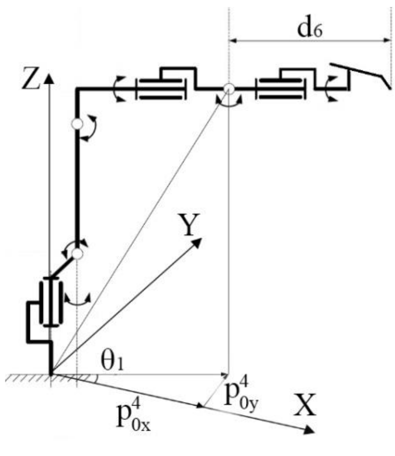


Fig. 2. The geometric image for finding the angle θ_1.

Then,

$$\tan(\theta_1) = \frac{p_{0y}^4}{p_{0x}^4} \quad \rightarrow \quad \theta_1 = \arctan\left(\frac{p_{0y}^4}{p_{0x}^4}\right). \quad (3)$$

This solution is correct for the robot's rotating angles θ_1 ranging from $\frac{\pi}{2}$ to 0 and to $-\frac{\pi}{2}$.

For the rotating angles equalling $\frac{\pi}{2}, \pi, -\frac{\pi}{2}$, the solution will be as follows:

$$\theta_1 = \pi + \arctan\left(\frac{p_{0y}^4}{p_{0x}^4}\right) \quad (4)$$

for the angles ranging from $\frac{\pi}{2}$ to π;

$$\theta_1 = -\pi + \arctan\left(\frac{p^4_{0y}}{p^4_{0x}}\right) \tag{5}$$

for the angles ranging from π to $-\frac{\pi}{2}$.

This differentiation is aimed at reducing the travel path of the robot to a desired coordinate.

The angles $\theta_2 \ldots \theta_6$ are calculated in a similar way.

4. The algorithm for computing the rotating angles of the robotic manipulator

The rotating angles of all the axes of the manipulator are equal to the calculated theta angles multiplied by the reduction rate of the corresponding reduction gears. This algorithm was developed in the Matlab environment and is the solution of IKP. Based on the developed algorithm, the flow chart was built, which is shown in Fig.3, Fig.4.

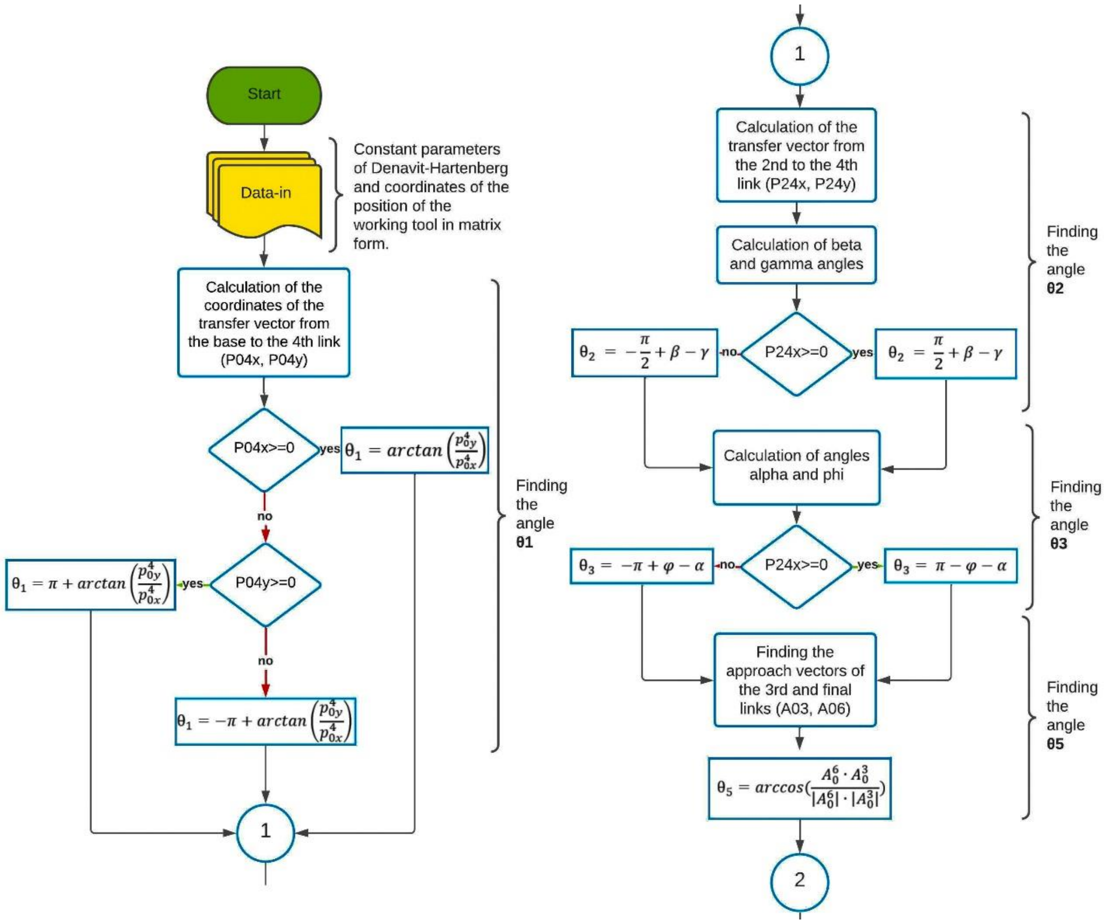
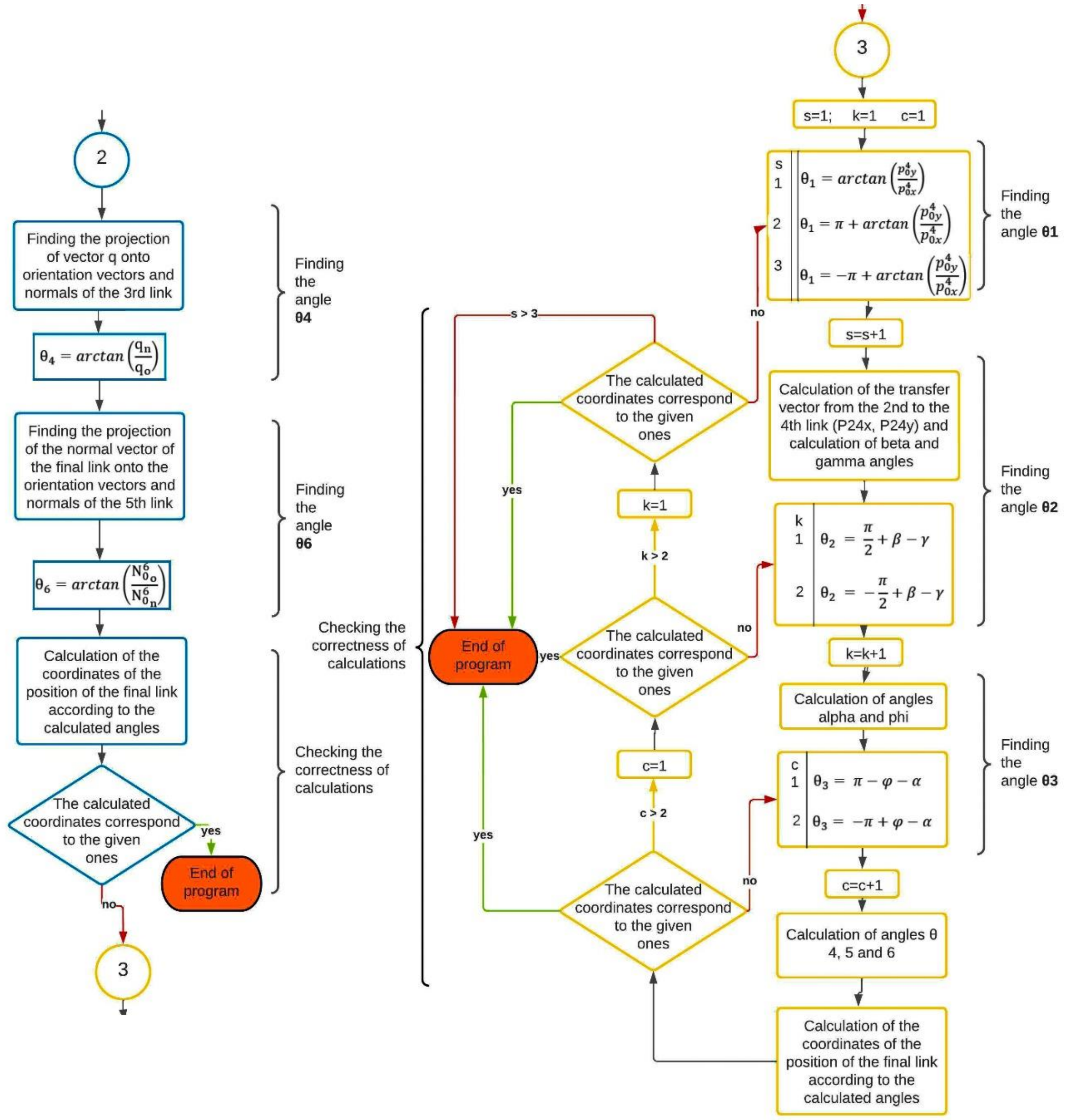


Fig.3. The flow chart of the algorithm for computing the rotating angles of the robotic manipulator. Start of the blue subprogram.

The algorithm and flow chart are split into two subprograms, the blue one and the yellow one. This was done due to the uncertainty of the solutions. In other words, different configurations of the robot can correspond to one and the same spacial position of the working tool. In some cases, the geometric method produces an incorrect result.

The blue part of the chart implements the algorithm that tries solution equations based on certain signs. These signs are the location of the position vector in certain quadrants of the coordinate system. This needs to be done only for the first three theta angles.

Upon completing the calculation of the coordinates of the positions of the end link according to the calculated angles, the check-up with the matrix T_0^6 is run in the blue subprogram. If the result of the check-up is negative, the yellow part of the scheme is actuated.



a) *b)*

Fig.4. The flow chart of the algorithm for computing the rotating angles of the robotic manipulator: *a)* end of the blue subprogram; *b)* yellow subprogram.

In the yellow subprogram, the algorithm goes through 12 different solutions for theta angles. This is done using loops. After each loop, the coordinates of the final position of the end link are checked up according to the calculated theta angles.

The full loop of the yellow part of the algorithm:

1) calculating θ_1 using the equation *s*;

2) *s*=*s*+1;

3) calculating parameters for evaluating θ_2;

4) calculating θ_2 using the equation *k*;

5) *k*=*k*+1;

6) calculating parameters for evaluating θ_3;

7) calculating θ_3 using the equation *c*;

8) $c = c$+1;

9) calculating parameters for evaluating θ_4, θ_5 and θ_6 and determining these angles;

10) calculating positions coordinates for the end link based on the calculated angles θ_i;

11) verification of the coordinates;

12) YES – the calculation is completed, angles θ_i are determined;

NO – calculation θ_3 using the equation *c*;

c > the number of equations – proceeds starting from step 4, *c* is zeroed;

13) *k* > the number of equations – proceeds starting from step 4, *k* is zeroed;

14) *s* > the number of equations – the preset coordinates cannot be reached or there has been a failure.

This division into two subprograms is the result of the optimization of the program as a whole. The blue subprogram is better optimized as it has fewer calculations but is more vulnerable. The yellow subprogram has a large number of calculations but it finds the correct result at any uncertainties. If the blue subprogram succeeds, the yellow one is not actuated.

5. An example of solving the inverse kinematic problem

As an example, the angles θ_i (of the generalized coordinates) of all the links and travel path of the working tool will be calculated. The initial data are the coordinates of the working tool position in the final point *B* with coordinates *X,Y,Z* (432; 1576; 1572), written as a homogeneous transformation matrix T_0^6 taken from the results of solving the forward kinematic problem and constant Denavit–Hartenberg parameters (*d, α, a*), recorded in Table 1.

Table 1. Constant parameters of the coordinate system of the manipulator links.

Joint	*d* (mm)	*α*	*a* (mm)
1	637	–90°	249
2	0	0°	650
3	0	–90°	100
4	815	90°	0
5	0	–90°	0
6	499	0°	0

$$T_0^6 = \begin{pmatrix} 0.5 & 0 & 0.87 & 432 \\ -0.84 & 0.26 & 0.48 & 1576 \\ -0.22 & -0.97 & 0.13 & 1572 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \qquad (6)$$

This matrix is the result of solving the forward kinematic problem. As a result of calculations of the algorithm for determining the rotating angle of the robotic manipulator (Fig.3, Fig.4), θ_i angles were obtained. For the working tool to reach the point *B*, the manipulator must turn the axes by the angles presented in Table 2.

Table 2. The rotating angles of the manipulator.

Joint number	θ by IKP	θ by FKP
1	90°	90°
2	30°	30°
3	−45°	– 45°
4	−90°	90°
5	60°	– 60°
6	180°	0°

The comparison of the results of solving IKP with the initial FKP data reveals disparities between the rotating angles. This stems from the uncertainty of the solutions. Despite that, having verified the matrices T_0^6, calculated based on the obtained angles θ_i and resulting from the forward kinematic problem (6), it is obvious that they are identical. The result of solving IKP is correct. The calculation of the travel path of the manipulator elements was carried out in the Matlab environment. The resulting curves are presented in Fig.5 – Fig.7.

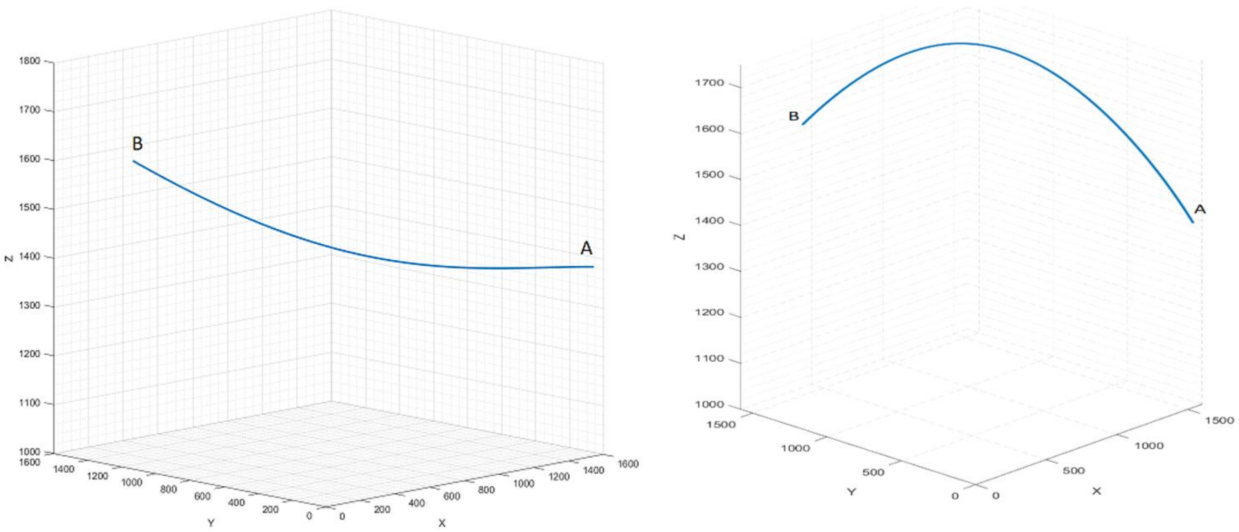


Fig.5.The travel path of the manipulator working tool from point *A* to point *B* in the coordinates *X*,*Y*,*Z*.
Left: IKP–based calculation; Right: FKP–based calculation.

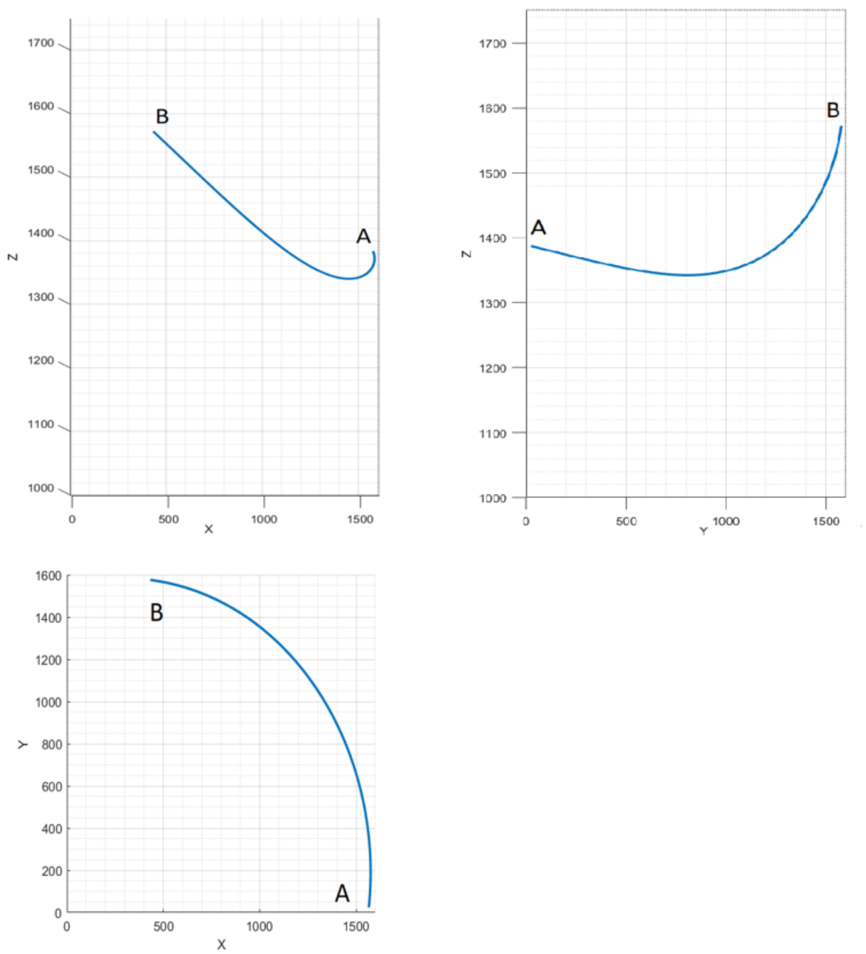


Fig.6. The travel path of the manipulator working tool from point *A* to point *B* in different planes. IKP-based calculation.

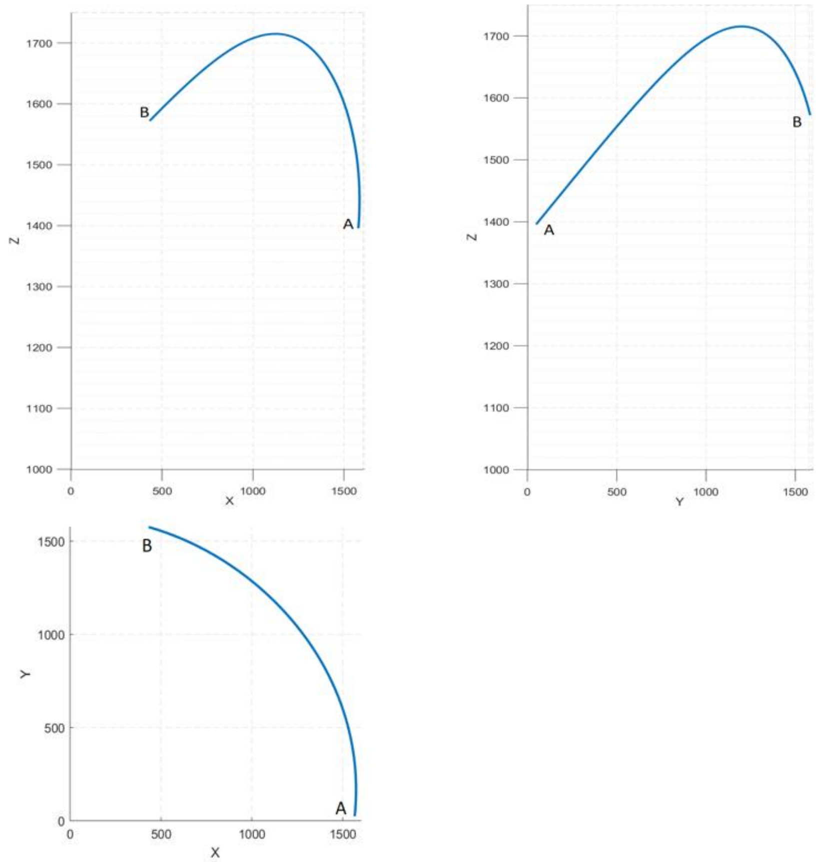


Fig.7. The travel path of the manipulator working tool from point *A* to point *B* in different planes. FKP-based calculation.

The comparison of the travel paths of the tool in Fig.5 when solving FKP and IKP, confirms the uncertainty of the solutions as the travel paths are different.

6. Conclusion

The paper solves the inverse kinematic problem for the six-axis manipulator. The study resulted in the algorithm for computing the rotating angles of the manipulator links in order to achieve the desired position.

The calculations produced several variants of the working instrument reaching the final position. The disparities were demonstrated in the calculated trajectories, which corroborate the statement about the uncertainty of the solutions of the inverse kinematic problem.

This algorithm consists of two components, which makes it possible to take into consideration the specificity of the IKP solution and to obtain accurate rotating angles of the robotic manipulator axes for the known preset spatial position and direction of the working tool. The examples of solving the forward and inverse kinematic problems were given and the trajectories of the working tool were built and compared, proving the correctness of the solutions.

The obtained algorithm can be used for creating control systems for robotic manipulators according to the requirements of the technological process.

References

[1] V. Fedak, F. Durovsky, R. Uveges, K. Kyslan, and M. Lacko (2015) Implementation of Robot Control Algorithms by Real-Time Control System. *International Journal of Engineering Research in Africa*, vol.18, 112–119. DOI:10.4028/www.scientific.net/JERA.18.112

[2] B.Siciliano, L.Sciavicco, L.Villani, G.Oriolo (2010) Robotics: Modelling, Planning and Control. Springer Science&Business Media. 632 p.

[3] Matthew T. Mason (2001) Mechanics of Robotic Manipulation. MIT Press. 272 p.

[4] Vihuro M., Malyar A. (2021) Solving the forward kinematics problem for a welding manipulator with six degrees of freedom. *Electrical Power and Electromechanical Systems*, no.1. 27-34. https://doi.org/10.23939/sepes2021.01.027

[5] C. Smith and H. Christensen (2009) Robot manipulators. *IEEE Robotics&Automation Magazine*, vol.16, no.4, 75-83. DOI: 10.1109/MRA.2009.934825

[6] G. C. Fernandez, S. M. Gutierrez, E. S. Ruiz, F. M. Perez and M. C. Gil. (2012) Robotics the New Industrial Revolution, *IEEE Technology and Society Magazine*, vol. 31, no. 2, 51-58 DOI: 10.1109/MTS.2012.2196595

[7] Blatnický M, Dižo J, Gerlici J, Sága M, Lack T, Kuba E. (2020) Design of a robotic manipulator for handling products of automotive industry. *International Journal of Advanced Robotic Systems*, no.17(1). DOI:10.1177/1729881420906290

[8] Fu, K., Gonzalez, R., Lee, K. Robotics: Translated from English. Moscow: Mir Publishers, 1989. 624 p. https://www.studmed.ru/fu-k-gonsales-r-li-k-robototehnika_8855f0f7adb.html. (in Russian)

[9] Dmitrieva I.S, Levchenko D.O. Research of kinematic model of manipulative robot // System technology. 2015. No. 3 (98). pp. 57-62. https://journals.nmetau.edu.ua/issue/download/49-24-PB-2.pdf.

[10] Burdakov, S.F., Dyachenko, V.A., Timofeev, A.N. (1986) Design of industrial robotic manipulators and robotic centers. Moscow: Vysshaya Shkola Publishers. 264 p. https://www.studmed.ru/burdakov-sf-dyachenko-vatimofeev-an-proektirovanie-manipulyatorov-promyshlennyh-robotov-i-robotizirovannyh-kompleksov_f9dfe59d37d.html (in Russian)

[11] Colomé A., Torras C. (2012) Redundant inverse kinematics: Experimental comparative review and two enhancements // Intelligent Robots and Systems (IROS). 2012. pp. 5333-5340. DOI: 10.1109/IROS.2012.6385672.

[12] Duka A. V. (2014) Neural network based inverse kinematics solution for travel path tracking of a robotic arm. *Procedia Technology*, vol. 12. 20-27. DOI:10.1016/j.protcy.2013.12.451.

[13] K. Tokarz and S. Kieltyka (2010) Geometric approach to inverse kinematics for arm manipulator. *Proceedings of International Conference on Systems*, vol. II, ISSN 1792-4235, ISBN 978-960-474-214-1, 682-687

[14] Denavit J. and Hartenberg R. S. (1955) A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. *Journal of Applied Mechanics*, vol. 22, 215–221. DOI:10.1115/1.4011045.

[15] Marushchak, Ya.Yu. Kushnir, A.P. (2016) The Mathematical Model of the Mechanism of Electrodes Movement of Arc Steel-Melting Furnaces on the Basis of Denavit-Hartenberg Presentation. *Electrotechnical and Computer Systems Journal*, no 22 (98). 20-27. DOI: 10.15276/eltecs.22.98.2016.03.

Синтез алгоритму керування положенням шестиосьового маніпулятора

Максим Вігуро[a], Андрій Маляр[a,b]

[a]*Національний університет «Львівська політехніка», вул. Степана Бандери, 12, Львів, 79013, Україна*
[b]*Бидгощський технічний університет, алея проф. С. Каліскєго, 7, Бидгощ, 85-796, Польща*

Анотація

В статті формулюється обернена задача кінематики для розглянутого робота маніпулятора з шістьма ступенями. Для розв’язку даної задачі використано геометричний метод у поєднанні з перетвореннями Денавіта та Хартенберга. Перевагою методу Денавіта–Хартенберга є зменшення кількості координат, що визначають тіло в просторі, з шести до чотирьох. Даний метод забезпечує точне позиціювання робочого інструменту. Проведено розрахунок оберненої задачі кінематики. Метою даної задачі є розрахунок кута повороту кожної з осей. Наведено геометричний розв'язок задач для кожної з осей. На основі даних розрахунків розроблено алгоритм визначення кутів повороту робота маніпулятора. Даний алгоритм реалізований в програмному середовищі Matlab. Наведено блок схеми алгоритму та описано його роботу. Продемонстровано приклад розв’язку оберненої задачі кінематики за допомогою розробленого алгоритму. Проведено перевірку результатів розрахунків, які збіглись із заданим наперед положенням, що свідчить про адекватність створеної моделі.

Ключові слова: робот маніпулятор; робочий інструмент; система координат; траєкторія руху; метод Деневіта–Хартенберга; обернена задача кінематики; узагальнені координати.