Comparison of the Efficiency of Numerical Algorithms for Solving the Inverse Kinematics Problem of Redundant Serial-Link Robots

Xuan Bien Duong[1(✉)], Anh My Chu[1], Chi Hieu Le[2], Tien Anh Nguyen[1], Minh Duc Vu[1], and Anh Le[1]

[1] Le Quy Don Technical University, Hanoi, Vietnam
duongxuanbien@lqdtu.edu.vn
[2] Faculty of Science and Engineering, University of Greenwich, London, UK

Abstract. This paper presents the results of the comparison of the use efficiency between two numerical algorithms which are the Adjust of Generalized Vector (AGV) and the Closed-Loop Inverse kinematics (CLIK) method when solving the inverse kinematics problem (IK) for the redundant serial-link robots. AGV method is developed based on the Taylor expansion method but adjusted to improve the accuracy of the IK solution. CLIK is a numerical method that is built as a closed-loop kinematics control system with the position error of the end-effector point in the workspace. The industrial SCARA robots with 4 degrees of freedom (DOF) including rotational and translational joints and PA10-7C industrial robots with 7-DOF are used to illustrate the effectiveness of the algorithms. The comparison results are expressed through the obtained joint variables values, the position error of the end-effector point in the workspace, and the deviation values of joint variables between the two methods. The results of this study allow evaluating the effectiveness of these methods for solving the IK and serve as the basis for choosing a suitable method for each specific robot configuration. However, this article has not considered the effectiveness of the methods mentioned above for other types of robots.

Keywords: Inverse kinematics · Effective algorithm · Redundant robots · Closed-loop feedback

1 Introduction

The characteristics of the redundant robot create the distinctive of the IK. It means that will exist countless options for the configuration of the robot in response to a task in the workspace, or in other words, the system of inverse kinematics equations has countless solutions. This is the advantage of this type of robot because we can choose an optimal result among a multitude of results on the basis of applying quality criteria such as avoidance of limited joints, joint velocity, avoid obstacles, the configuration degradation and is quite useful for minimizing energy consumption [1–6]. However, due to the multitude of solutions to choose from, choosing a viable option that fits the robot's physical configuration is a huge challenge. Research on effective solutions to solve the

B. T. Long et al. (Eds.): MMMS 2020, LNME, pp. 1000–1008, 2021.
https://doi.org/10.1007/978-3-030-69610-8_132

IK problem for redundant robots is the top concern because it is the foundation to analyze the dynamics problem and design the control system.

There are many numerical algorithms which are developed to solve the IK such as Jacobian Transpose [7], Pseudoinverse [8], DLS [9], Quasi-Newton and Conjugate gradient [10, 11], Closed-loop inverse kinematics (CLIK) [12–15]. The Quick IK algorithm [7] to improve the performance solving IK from Jacobian Transpose method due to a great deal of loops. The IK was solved by using CLIK method with the velocity and acceleration constraints [13, 14]. CLIK method is used in [15] for welding robot 6-DOFs combining with a positioner to track a complex 3D curve. The parallel genetic algorithm is used to solve the IK of Puma 500 robot in [16]. The elimination technique is presented in [17] to reduce the complexity of the inverse kinematics formulation based on analytical method. The new solution method is introduced in [18] to avoid joint limitation, avoid singularities and avoid obstacles for redundant robots. The AGV method has been presented and developed in [19, 20, 21]. This numerical method has been applied quite effectively in solving the IK for the redundant robots with 4DOF and 5DOF [19], 6 DOF [20, 21] with small position error in the joint-space and the workspace. It is noted that all of the joints in these studies are rotational joints.

This paper focuses on comparing the efficiency of solving the IK for the redundant serial-link robots between the AGV and CLIK algorithm because of their outstanding compared to the other numerical methods. The highlight of this study is shown through the simulation results of the IK for SCARA robot with 4DOF combining a translational joint and robot PA10-7C with 7DOF [22]. The kinematics equations of these robots are established based on the homogeneous matrix transformation D-H method. The IK solving algorithms are performed on MATLAB/SIMULINK software with kinematics parameters of robots in reality.

2 Materials and Methods

2.1 Kinematics Modeling

Based on the D-H method, a local D-H matrix $\mathbf{H}_i$ for the link i was established as follows [23, 24]. For a robot consisting of n serial of links, the position of the end-effector point on link n relative to the fixed coordinate system is determined as

$$\mathbf{D}_n = \mathbf{H}_1\mathbf{H}_2..\mathbf{H}_n \tag{1}$$

Define the vector $\mathbf{x}$ which includes the first three elements of 4^{th} column of $\mathbf{D}_n$ matrix describes position of the end-effector point (E point) relative to the fixed coordinate system. In essence, the vector $\mathbf{x}$ represents the relationship between the end-effector point position and the joint variable values. Therefore, the forward kinematics equations are determined as $\mathbf{x} = f(\mathbf{q})$ and the end-effecter point velocity is inferred as $\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$. The Jacobian matrix is $\mathbf{J}(\mathbf{q})$ with size $3 \times n$. The inverse kinematics equations are determined as $\mathbf{q}(t) = f^{-1}(\mathbf{x}(t))$. Once the values of $\mathbf{q}$ have been determined from Eq. (4), the joints velocity are calculated as $\dot{\mathbf{q}} = \mathbf{J}^+(\mathbf{q})\dot{\mathbf{x}}$ with $\mathbf{J}^+(\mathbf{q})$ is the pseudo-inverse matrix of $\mathbf{J}(\mathbf{q})$ and defined as [23, 24]. The joints acceleration is

inferred as $\ddot{\mathbf{q}} = \mathbf{J}^{+}(\mathbf{q})(\ddot{\mathbf{x}} - \dot{\mathbf{J}}\dot{\mathbf{q}})$. After calculating the value of vector $\mathbf{q}$, the value of vectors $\dot{\mathbf{q}}, \ddot{\mathbf{q}}$ can be determined.

2.2 The Algorithms for the IK Solving

2.2.1 Algorithm for Adjusted of Generalized Vector

Assume that the robot works in the period from $t = 0$ to $t = t_f$. Divide the time interval $[0 \div t_f]$ into N equal segments. The value of each segment is $\Delta t = t_f/N$. The next time step can be calculated as $t_{k+1} = t_k + \Delta t; k = 0 \div (N-1)$. Operate Taylor's expansion of $\mathbf{q}(t_{k+1})$ near the value of $\mathbf{q}(t_k)$, the values of generalized joints position vector are given as [19, 20].

$$\mathbf{q}(t_{k+1}) = \mathbf{q}(t_k + \Delta t) = \mathbf{q}(t_k) + \dot{\mathbf{q}}(t_k)\Delta t + \frac{1}{2}\ddot{\mathbf{q}}(t_k)(\Delta t)^2 + \ldots \tag{2}$$

The result given (2) is quite rough at $t = t_{k+1}$. Determine a better approximation according to the formula $\mathbf{q}(t_{k+1}) = \tilde{\mathbf{q}}(t_{k+1}) + \Delta\mathbf{q}(t_{k+1})$ by calculating the value of $\Delta(t_{k+1}) = \mathbf{J}^{+}(\tilde{\mathbf{q}}(t_{k+1}))[\mathbf{x}(t_{k+1}) - f(\tilde{\mathbf{q}}(t_{k+1}))]$, $\tilde{\mathbf{q}}(t_{k+1}) = \tilde{\mathbf{q}}(t_{k+1}) + \Delta\mathbf{q}(t_{k+1})$. Operate the loop until $\|\Delta(t_{k+1})\| \leq \varepsilon$ with ε is the allowable error vector then take $\mathbf{q}(t_{k+1}) = \tilde{\mathbf{q}}(t_{k+1})$. So, the value of generalized position joints vector $\mathbf{q}$ has been found. Specific contents of this algorithm were clearly described in [19, 20].

2.2.2 CLIK Algorithm

The general IK solution can be described as [7].

$$\dot{\mathbf{q}} = \mathbf{J}^{+}(\mathbf{q})\dot{\mathbf{x}} + (\mathbf{I} - \mathbf{J}^{+}(\mathbf{q})\mathbf{J}(\mathbf{q}))\dot{\mathbf{q}}_0 \tag{3}$$

where, $\mathbf{I}$ is the unit matrix with size $n \times n$ and $\mathbf{q}_0$ is the initial value. The closed-loop algorithm is used based on the path error $\mathbf{e} = \mathbf{x}_d - \mathbf{x}$ in workspace between the desired and actual path ($\mathbf{x}_d$ and $\mathbf{x}$). The generalized CLIK algorithm can be expressed by [7].

$$\dot{\mathbf{q}} = \mathbf{J}^{+}(\mathbf{q})(\dot{\mathbf{x}}_d + \mathbf{K}_p(\mathbf{x}_d - \mathbf{x})) \tag{4}$$

The IK solution of redundant robot based on closed-loop algorithm is given as [7, 15].

$$\dot{\mathbf{q}} = \mathbf{J}^{+}(\mathbf{q})(\dot{\mathbf{x}}_d + \mathbf{K}_p(\mathbf{x}_d - \mathbf{x})) + (\mathbf{I} - \mathbf{J}^{+}(\mathbf{q})\mathbf{J}(\mathbf{q}))\dot{\mathbf{q}}_0 \tag{5}$$

Where, $\mathbf{K}_p$ is a symmetric positive definite matrix. This algorithm was clearly described in [7, 15].

2.3 Numerical Simulation Results and Discussion

Consider the kinematics model of SCARA robot with 4-DOF as shown in Fig. 1 and PA10-7C robot with 7-DOF as shown in Fig. 2.

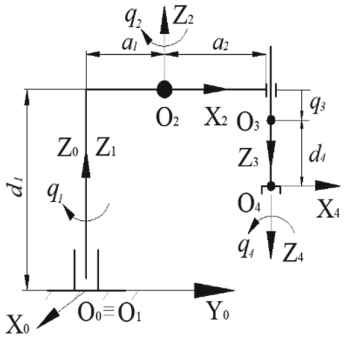

Fig. 1. Kinematics model of SCARA robot $\mathbf{q} = [\,q_1 \quad q_2 \quad q_3 \quad q_4\,]^T$, q_1, q_2, q_4 in radian unit and q_3 in meter unit.

Table 1. D-H parameters SCARA

Links	D-H parameters			
	θ_i	d_i	a_i	α_i
1	q_1	d_1	a_1	0
2	q_2	0	a_2	π
3	0	q_3	0	0
4	q_4	d_4	0	0

$\mathbf{q}_{\max} = [\,2.5 \quad 2.5 \quad 0.4 \quad 6.2\,]^T$,
$\dot{\mathbf{q}}_{\max} = [\,6.5 \quad 6.5 \quad 0.3 \quad 31.4\,]^T$,
$d_1 = 0.42; a_1 = 0.22; a_2 = 0.4(m)$,
$a_2 = 0.4; d_4 = 0.09(m)$

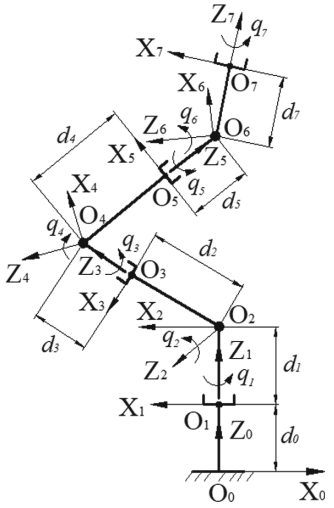

Fig. 2. Kinematics model of PA10-7C robot $\mathbf{q} = [\,q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5 \quad q_6 \quad q_7\,]^T$

Table 2. D-H parameters PA10-7C

Links	D-H parameters			
	θ_i	d_i	a_i	α_i
1	q_1	$d_0 + d_1$	0	$-\pi/2$
2	q_2	d_2	0	$\pi/2$
3	q_3	d_3	0	$-\pi/2$
4	q_4	d_4	0	$\pi/2$
5	q_5	d_5	0	$-\pi/2$
6	q_6	0	0	$\pi/2$
7	q_7	d_7	0	0

Where, $d_0 = 0.21; d_1 = 0.1; d_2 = 0.21; d_3 = 0.13(m)$, $d_4 = 0.28; d_5 = 0.21; d_7 = 0.07(m)$
$\dot{\mathbf{q}}_{\max} = [\,1 \quad 1 \quad 2 \quad 2 \quad 6.3 \quad 6.3 \quad 6.3\,]^T$ (rad/s)

The fixed coordinate system is $(OXYZ)_0$ and $(OXYZ)_i$, ($i = 1 \div 4$ with SCARA and $i = 1 \div 7$ with PA10-7C) are the local coordinate systems attached link i. Table 1 and Table 2 describe the kinematic parameters according to the D-H rule [23, 24] of two robots, respectively. Accordingly, the transformation $\mathbf{H}_i$ homogeneous matrices are determined. A given trajectory in the workspace for SCARA robot is described as

$$x_E = 0.3 + 0.08\cos(0.2t)(m); y_E = 0.3 + 0.08\sin(0.2t)(m); z_E = 0.1\cos(0.2t)(m)$$

A given trajectory in the workspace for PA10-7C robot is presented as

$$x_E = 0.2 + 0.4\cos(0.2t)(m); y_E = 0.2 + 0.4\cos(0.2t)(m); z_E = 0.65(m)$$

The given position error value in AGV method is $\varepsilon = 10^{-3}(m)$. The values of coefficient in CLIK method for the SCARA robot and the PA10-PC robot are $k_p = 50$ and $k_p = 100$. The numerical simulation results of the SCARA are shown in Fig. 3, 4, 5, 6, 7, 8, 9, 10, 11 and Fig. 12 and of the PA10-7C robot are described in Fig. 13, 14, 15, 16, 17, 18, 19 and Fig. 25 after applying the AGV and CLIK algorithm to solve the IK problem. Figure 12 and Fig. 25 show the models of both robots in 3D by using the values of joint variables that are obtained from the inverse kinematics problem. These models confirm the reliability of the algorithms.

With SCARA robot, the position of joint 1, joint 2 and translational joint 3 are presented in Fig. 3, Fig. 4 and Fig. 5, respectively. The deviation values between AGV and CLIK is shown in Fig. 6. The maximum deviation value is $10^{-2}(m)$ in joint 2. However, when using the joint variable values to find the position of the end-effector point in the workspace through the forward kinematics problem (Fig. 7, Fig. 8, and Fig. 9), there appears to be quite a difference large in the position error of the end-effector point (AGV with Fig. 10, CLIK with Fig. 11). This difference lies in the nature of the algorithms. With the CLIK method, the position error values continuously changed according to the closed-loop trajectory control system. On the contrary, the position error is calculated specifically in each time step in the AGV method. According to the AGV algorithm, the step i is performed until the position error value smaller than the preset error value, then step $i + 1$ can be performed.

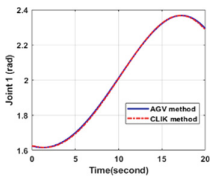

Fig. 3. Joint 1

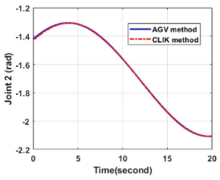

Fig. 4. Joint 2

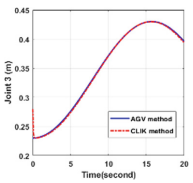

Fig. 5. Joint 3

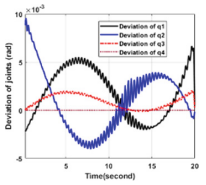

Fig. 6. Deviation

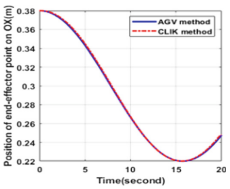

Fig. 7. E point in OX

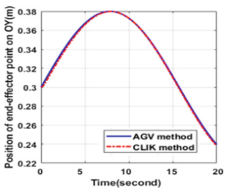

Fig. 8. E point in OY

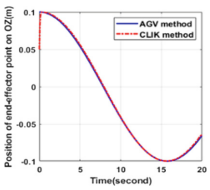

Fig. 9. E point in OZ

For the PA10-7C robot, the values of the joints are depicted in Fig. 13, 14, 15, 16, 17 and Fig. 18. The large differences between the two robots are the translational joint of the SCARA robot and the number of DOF. The number of kinematic equations is also different. This explains the difference in the results of the IK between the AGV and CLIK methods.

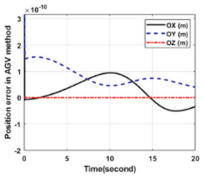

Fig. 10. Error position of E point with AGV

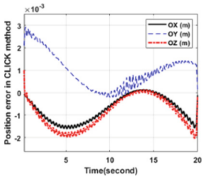

Fig. 11. Error position of E point with CLIK

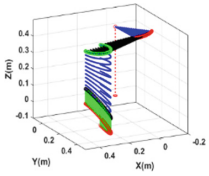

Fig. 12. SCARA model in MATLAB

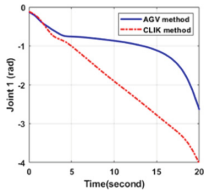

Fig. 13. Joint 1 position

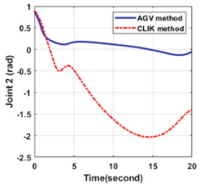

Fig. 14. Joint 2 position

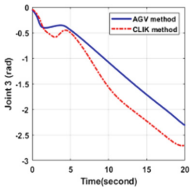

Fig. 15. Joint 3 position

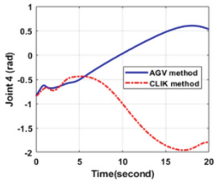

Fig. 16. Joint 4 position

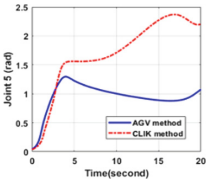

Fig. 17. Joint 5 position

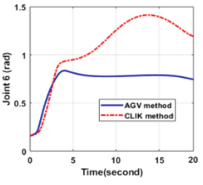

Fig. 18. Joint 6 position

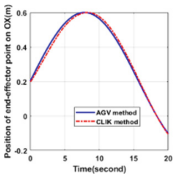

Fig. 19. E point in OX

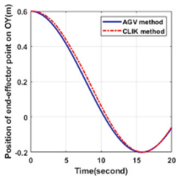

Fig. 20. E point in OY

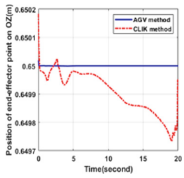

Fig. 21. E point in OZ

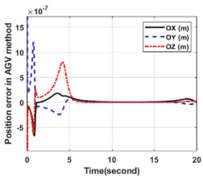

Fig. 22. Error position with AGV

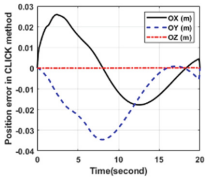

Fig. 23. Error position of E point with CLIK

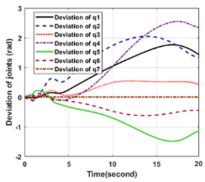

Fig. 24. Joints deviation (AGV and CLIK)

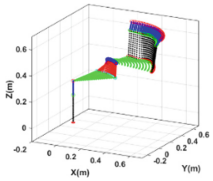

Fig. 25. PA10-7C model in MATLAB

The deviation value of the joint variables is shown in Fig. 24. The end-effector point position is shown in Fig. 19, Fig. 20, and Fig. 21. The position errors of the end-effector point of the two methods are described in Fig. 22 and Fig. 23. The position error of the CLIK method is higher than that of the AGV method. The cause has been considered similar to the case of the SCARA robot.

In general, both techniques have the same level of complexity when calculating Jacobian matrices. However, the CLIK algorithm needs to find a suitable K_p coefficient to ensure the minimum position error. To find this coefficient, it can be to use the automatic adjustment module of MATLAB/SIMULINK software. This technique is easier in designing a position controller. However, given the trajectory in the workspace is a curve, this method is difficult to apply, it is necessary to use a support optimization algorithm to ensure the position error is within the allowable limit. The AGV method takes more computation time than the CLIK method because this algorithm performs for each specific time step. Based on the above evaluation, it can be said that the AGV method is more effective in solving the IK for the redundant serial-link robots than the CLIK method with the given path of the end-effector point in the workspace is a curve.

3 Conclusion

Through building the kinematics equations of the redundant serial-link robots including the SCARA robot with 4DOF combining a translational joint and the PA10-7C robot with 7DOF, two methods AGV and CLIK are applied to solve the inverse kinematics problem of these redundant robots. The efficiency of these methods was compared

through numerical simulation results, including the position and deviation of the joint variables, the position, and position error of the end-effector point in the workspace. The results of this study show that both methods have successfully applied to solve the IK of redundant serial-link robots. However, the AGV method is more efficient than the CLIK method for this robot type because the simulation results show that the AGV method gives smaller position errors, higher reliability through each calculation step, and does not depend on finding the coefficient K_p suitable in the CLIK method. This study has not covered IK for other types of robots such as parallel robots, mobile robots.

Acknowledgement. This work was supported by a Research Environment Links, ID 528085858, under the Newton Fund partnership. The grant is funded by the UK Department for Business, Energy and Industrial Strategy and delivered by the British Council.

References

1. Locke, R.C.O, Patel, R.V.: Optimal remote center-of-motion location for robotics-assisted minimally-invasive surgery. In: 2007 IEEE International Conference on Robotics and Automation, pp. 1990–1905 (2007)
2. Chiaverini, S., Oriolo, G., Walker, I.D.: Kinematically Redundant Manipulators. Springer Handbook of Robotics, pp. 245–268. Springer, Heidelberg (2007)
3. Wang, J., Hu, X., Zhang, B.: Real-time motion planning of kinematically redundant manipulators using recurrent neural networks. In: Yu, W. (ed.) Recent Advances in Intelligent Control Systems, pp. 169–193. Springer, London (2009)
4. Maciejewski, A.A., Klein, C.A.: Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. Int. J. Robot. Res. **4**, 109–116 (1985)
5. Klanke, S., Lebedev, D., Haschke, R., Steil, J., Ritter, H.: Dynamic path planning for a 7-DOF robot arm. In: Proceedings of the 2006 IEEE/RSJ, pp. 3879–3884 (2006)
6. Llama, M.A., Wism, A.Z., Castañon, W.Z., Hernandez, R.G.: Force and position fuzzy control: a case study in a Mitsubishi PA10-7CE robot arm. Advanced Topics on Computer Vision, Control and Robotics in Mechatronics, pp. 165–194. Springer, Cham (2018)
7. Lian, S., Han, Y., Wang, Y., Bao, Y., Xiao, H., Li, X., Sun, N.: Accelerating inverse kinematics for high-DOF robots. In: Proceedings of the 54th Annual Design Automation Conference 2017, Austin, USA (2017)
8. Yoshikawa, T.: Dynamic manipulability of robot manipulators. J. Robot. Syst. **2**, 113–124 (1985)
9. Wampler, C.W.: Manipulator inverse kinematic solutions based on vector formulations and damped least squares methods. IEEE Trans. Syst. Man Cybern. **16**, 93–101 (1986)
10. Wang, L.C.T., Chen, C.C.A.: Combined optimization method for solving the inverse kinematics problem of mechanical manipulators. IEEE Trans. Robot. Automation **7**, 489–499 (1991)
11. Zhao, J., Badler, N.I.: Inverse kinematics positioning using nonlinear programming for highly articulated figures. ACM Trans. Graph. **13**, 313–336 (1994)
12. Sciavicco, L., Siciliano, B.: A solution algorithm to the inverse kinematic problem for redundant manipulators. J. Robot. Autom. **4**, 403–410 (1988)

13. Antonelli, G., Chiaverini, S., Fusco, G.: Kinematic control of redundant manipulators with online end-effector path tracking capability under velocity and acceleration constraints. In: IFAC Robot Control, Austria, pp. 183–188 (2000)
14. Wang, J., Li, Y., Zhao, X.: Inverse kinematics and control of a 7 dof redundant manipulator based on the closed loop algorithm. Int. J. Adv. Robot. Syst. **7**, 1–10 (2010)
15. My, C.A., Bien, D.X., Tung, H.B., Hieu, L.C., Cong, N.V., Hieu, T.V.: Inverse kinematic control algorithm for a welding robot-positioner system to trace a 3D complex curve. In: International Conference on Advanced Technologies for Communications (ATC), pp. 319–323 (2019)
16. Aguilar, O.A., Huegel, J.C.: Inverse kinematics solution for robotic manipulators using a CUDA-based parallel genetic algorithms. In: Mexican International Conference on Artificial Intelligence 2011, Part 1, pp. 490–503 (2011)
17. Husty, M.L., Pfurner, M., Schrocker, H.P.: A new and efficient algorithm for the inverse kinematics of a general serial 6R manipulator. Mech. Machine Theory **42**, 66–81 (2007)
18. Kelemen, M., Virgala, I., Liptak, T., Mikova, L., Filakovsky, F., Bulej, V.: A novel approach for an inverse kinematics solution of a redundant manipulator. Appl. Sci. **8**, 2–20 (2018)
19. Khang, N.V., Dien, N.P., Vinh, N.V., Nam, T.H.: Inverse kinematic and dynamic analysis of redundant measuring manipulator BKHN-MCX-04. Vietnam J. Mech. VAST **32**, 15–26 (2010)
20. Bien, D.X., My, C.A., Hieu, L.C., Tung, B.H.: Inverse Kinematics Analysis of Welding Robot IRB 1520ID Using Algorithm for Adjusting the Increments of Generalized Vector, the 5th RICE2020, Springer. ISSN: 2194-5358 (2020)
21. Bien, D.X., Tuan, P.A., Nhat, D.D., Hiep, D.X., Nghia, T.K., Anh, M.N.: Optimize the feed rate and determine the joints torque for industrial welding robot TA 1400 based on kinematics and dynamics modeling. Int. J. Mech. Eng. Robot. Res. **9**, 1335–1340 (2020)
22. http://www.solutions4u-asia.com/. Accessed Sept 2020
23. Spong, M., Hutchinson, W., Vidyasagar, S.M.: Robot Modeling and Control, First edition. Springer, New York (2001)
24. Lewis, F., Dawnson, L., Abdallah, D.: Robot Manipulator Control Theory and Practice - Second edition, Marcel Dekker INC, New York, USA (2004)