Journal of Engineering Science and Technology
Vol. 14, No. 4 (2019) 1948 - 1959

IMPLEMENTATION OF AN ANALYTICAL INVERSE KINEMATICS SIMULATION METHOD FOR A 5-DOF PARALLEL MANIPULATOR

HENDRIKO HENDRIKO*, NUR KHAMDI, JAJANG JAENUDIN

Mechatronics Department, Politeknik Caltex Riau,
Jl. Umbansari, Rumbai, Pekanbaru 28265, Provinsi Riau, Indonesia
*Corresponding Author: hendriko@pcr.ac.id

Abstract

The simplicity of the kinematical calculation represents one of the main reasons for choosing a parallel manipulator. Similar to the case of a serial robot, the majority of studies concerning parallel robots use the Jacobian-based method to define the position and the posture of the end-effector. However, although the kinematical aspect of a parallel robot is simple, it will still prove costly if the inverse kinematics are calculated using the Jacobian-based method. Therefore, in this study, an analytical-based methodology was developed for calculating the inverse kinematics of a parallel robot. A simulation program based on the proposed algorithm was developed using Matlab. The simulation program, which is referred to as an Analytical Inverse Kinematics Simulation (AIKS), could be used to calculate the rotation of every actuator in order to achieve the required position. To ensure the implementability of the method, one test was performed using a complex trajectory. The result showed that the AIKS could be used to generate the degree of rotation of every actuator so as to achieve the required positioning of the end-effector. Further, to check the accuracy of the proposed model, two verification mechanisms were used. The results demonstrated that the proposed method is accurate. Moreover, the comparison test showed that the proposed method is cheaper in terms of the computational cost than the Jacobian method.

Keywords: 5-DOF, Analytical method, Inverse kinematics, Parallel manipulator.

1. Introduction

The level of interest in studying robotics technology has increased considerably over the past six decades. Many studies have attempted to make robots that can be used in various applications. According to the structural topologies, a robot can be classified as either a serial or a parallel robot. A robot will be categorized as being serial if its kinematic structure takes the form of an open-loop chain, while it will be categorized as being parallel if the chain is a closed loop. Gough and Whitehall [1] and Stewart [2] introduced the parallel manipulator in 1962 and 1965 respectively. Their architecture was termed the Gough-Steward platform manipulator. Since approximately the year 2000, numerous articles have described new architectures for parallel robots.

The parallel manipulator offers several benefits when compared to the serial manipulator, such as the high degree of accuracy, the high stiffness in almost all the setup configurations, and the high payload capacity. These benefits represent reasons why parallel manipulator is currently widely used in the manufacturing industry, flight simulators, etc. However, the parallel manipulator also exhibits several drawbacks, such as the limited workspace. Cox and Tesar [3] analysed comprehensively the differences between the parallel and serial manipulators.

The production of realistic motions in a robot remains a challenge. Many parallel robots have been developed and studied with regard to the appropriate implementations, such as the DELTA robot [4], 3-UPU [5], and 3-PRC [6]. These robots all feature a parallel manipulator with three DOF (Degrees of Freedom). Only a few robots featuring parallel manipulators with five or more DOF have been reported in the literature. Yet, manipulators with five or more DOF can offer the most flexibility in terms of the orientation of the end-effector.

The problems associated with inverse kinematics have been widely discussed in the last few years. Indeed, different approaches have been recommended for overcoming the issue of inverse kinematics in numerous applications. Mostly, researchers have proposed the use of numerical methods. For instance, Zhao and Badler [7] used a local minimum from a set of non-linear equations to overcome an inverse kinematics problem concerning Cartesian space. In terms of the numerical-based approach, the method most widely used to determine an inverse kinematics problem is the Jacobian method, which uses a matrix of partial derivatives of the connecting system relative to the posture of the end-effectors. Many studies have been conducted to develop the Jacobian method for inverse kinematics calculations, such as the Jacobian transpose, Lavenberg-Marquardt Algorithm (LMA), which is also known as the Damped Least Squares (DLS) and other derivatives [8, 9].

Researchers agree that the Jacobian method provides accurate results, although it also has disadvantages due to the long computation time, the complex matrix algorithm, and the problems of singularity [10]. These limitations occur because the Jacobian method solves the problem of inverse kinematics using a fully numerical approach.

Other studies have proposed different techniques for solving the problem of inverse kinematics, such as the Sequential Monte Carlo Method (SMCM) [11] and the particle filtering approach [12]. These two methods use a statistical-based approach rather than a matrix. Although such methods aim to avoid the use of a complex matrix calculation, the statistical approach still involves a long computational time. To overcome this problem, a hybrid-based methodology, which is a combination of the analytical and numerical methods, has been proposed [13, 14]. This method has been proven to significantly reduce the calculation time when compared to the fully numerical-based method. Many studies by Kiswanto et al. [15-17], Henriko [18] and Henriko et al. [19] have been performed to develop the analytical method in relation to various applications, as well as to compare its computational time to that of other methods. The results have shown that the analytical-based approach is able to reduce the computational time to a remarkable extent.

Therefore, Hendriko et al. [20] developed an analytical-based methodology called Analytical Inverse Kinematic Simulation (AIKS) in the present paper. The study was aimed to develop AIKS so that it could be used to determine the inverse kinematics problems associated with a 5-DOF parallel manipulator. In this study, the construction of a 3-DOF parallel manipulator was extended by attaching two more actuators onto the mobile platform. Hence, the end-effector can be oriented to arbitrary direction because it can be rotated about z-axis and x-axis.

2. Analytical Inverse Kinematics

Inverse kinematics is an approach for defining the posture of a robot by estimating every individual DOF required to perform a given task. The design of the 5-DOF parallel manipulator was developed as detailed in Fig. 1(a). The construction is comprised of several main parts, including a permanent platform that is fixed at the base of the robot and a mobile platform that can move freely in space. The end-effector was attached to the mobile platform. Three actuators are used to move the three links of the mobile platform, which are represented by points a_1, a_2, and a_3. The displacement of these points is intended to actuate the mobile platform. The actuator moves the mobile platform by moving points a_1, a_2, and a_3 up and down about the z-axis. Meanwhile, the rotational motion of the end-effector under the mobile platform is actuated by two other actuators. These two other actuators rotate the end-effector about both the z-axis and the y-axis. To ensure that the mobile platform could move freely due to the displacement of points a_1, a_2, and a_3, fisheye bearings were used at points b_1, b_2, and b_3.

In order to calculate the inverse kinematics of the end-effector, a suitable variable for mapping the coordinate frames were required. Therefore, several coordinate frames, as depicted in Figs. 1(a) and (b), were used to define the posture of the end-effector. The coordinate frames comprised: 1) Global Coordinate Frame (GCF), which was set up at the centre of the permanent platform, as can be seen in Fig. 1(a), 2) Local Coordinate Frame (LCF), which represents the initial position of the end-effector, as presented in Fig. 1(b), and 3) Mobile Coordinate Frame (MCF), which is considered at point E, as illustrated in Fig. 1(b). The GCF is a fixed system, which is denoted by X, Y, Z, while the LCF and MCF are represented by the basis vectors x, y, z and u, v, y respectively.

The z-axis of the GCF is perpendicular to the base, while the y-axis is parallel with line $\overline{b_1\ b_2}$, as shown in Fig. 1(a). The kinematics algorithm was intended to determine the movement of the links of the mobile platform, as represented by points $a_1\,(x_{a1}, y_{a1}, z_{a1})$, $a_2\,(x_{a2}, y_{a2}, z_{a2})$, and $a_3\,(x_{a3}, y_{a3}, z_{a3})$, the rotation angle about the z-axis (ε) and the rotation angle about the y-axis (λ). In determining both the coordinates and the orientation of the end-effector, three points must be given at the beginning, namely points D' (x_{D}', y_{D}', z_{D}'), E' (x_{E}', y_{E}', z_{E}') and F' (x_{F}', y_{F}', z_{F}'). In the initial position, points E and F are parallel with the x-axis. Then, all the points in Fig. 1 could be defined using the three given points. Finally, the rotation angle of the actuator required to move the manipulator to the expected position and orientation could be determined.

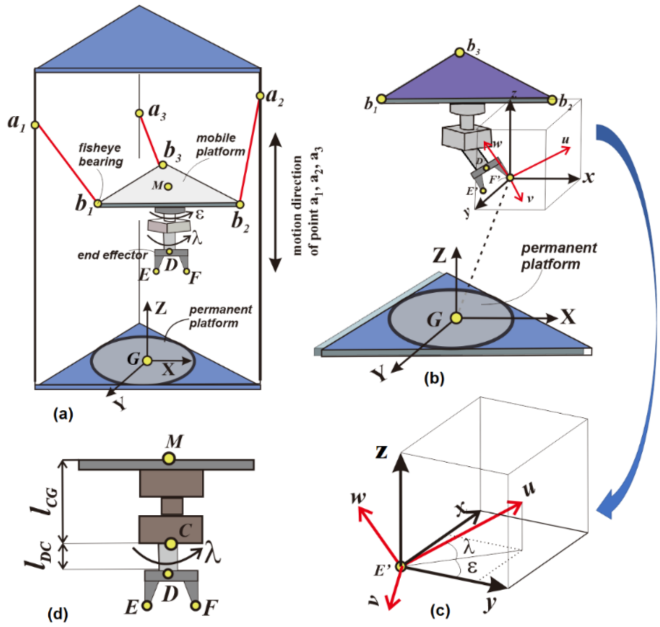

Fig. 1(a) Design of 5-DOF parallel manipulator,
(b) Coordinate frame, (c) Angle of rotation of end-effector,
and (d) Distance between links of end-effector.

The posture of the end-effector with respect to the LCF is presented in Fig. 1(d). It was determined using the orientation of the end-effector relative to both the x-axis (ε) and the z-axis (λ). They were calculated as follow:

$$\varepsilon = \sin^{-1}\left(\frac{y_{F'}-y_{E'}}{x_{F'}-x_{E'}}\right) \tag{1}$$

$$\lambda = sin^{-1}\left(\frac{z_{F'}-z_{E'}}{\sqrt{(x_{F'}-x_{E'})^2+(y_{F'}-y_{E'})^2}}\right) \tag{2}$$

The variables ε and λ denote the angle of rotation required by Actuators 1 and 2, respectively, to rotate the end-effector to the designed position. The mapping tool [T] was used to convert the coordinate frame from the MCF to the LCF, including the rotation of the manipulator about the y-axis and the z-axis. It was determined by:

$$[T] = Rot(Z,\varepsilon).Rot(Y,\lambda) \tag{3}$$

$$[T] = \begin{bmatrix} cos\,\varepsilon & -\sin\varepsilon & 0 \\ sin\,\varepsilon & cos\,\varepsilon & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\,\lambda & 0 & sin\,\lambda \\ 0 & 1 & 0 \\ sin\,\lambda & 0 & cos\,\lambda \end{bmatrix} \tag{4}$$

Based on the construction of the manipulator in Fig. 1(d), point C', which is located at the elbow of the end-effector, was determined by:

$$C'\ (x_{C'}\ ; y_{C'}\ ; z_{C'}\ ) = [T]\ D\ (x_D; y_D; z_D) + (0; 0;\ l_{CD}) \tag{5}$$

where l_{CD} is the distance between points C and D, as presented in Fig. 1(d).

With reference to Fig. 1(a), point M' ($x_{M'}$; $y_{M'}$; $z_{M'}$), which is located at the centre of the mobile platform, was defined as follows:

$$M'(x_{M'}; y_{M'}; z_{M'}) = C'(x_{C'}; y_{C'}; z_{C'}) + (0; 0; +l_{DG}) \tag{6}$$

where l_{DC} is the distance between points C and D, as presented in Fig. 1(c). Then, all the points on the mobile platform were calculated as follows:

$$b'_1(x_{b'_1}; y_{b'_1}; z_{b'_1}) = M'(x_{M'}; y_{M'}; z_{M'}) + (-r.\sin\alpha\ ;\ r.\cos\alpha\ ;\ 0) \tag{7}$$

$$b'_2\left(x_{b'_2}; y_{b'_2}; z_{b'_2}\right) = M'\left(x_{M'}; y_{M'}; z_{M'}\right) + (0;\ -r;\ 0) \tag{8}$$

$$b'_3\left(x_{b'_3}; y_{b'_3}; z_{b'_3}\right) = M'(x_{M'}; y_{M'}; z_{M'}) + (r.\sin\alpha\ ;\ r.\cos\alpha\ ;\ 0) \tag{9}$$

where r denotes the distance between the connecting points (b'_1, b'_2, b'_3) point M. An equilateral triangle was used as the shape of the mobile platform, as can be seen in Fig. 2(a). Points (b'_1, b'_2, and b'_3) $b'_1, b'_2,$ and b'_3 are the corner points on the mobile platform, hence, they can move translational about all the axes. Different to the b' points, points a_1, a_2, and a_3 were set up at the column, as presented in Fig. 1(a), which means that they are only able to move along the Z-axis. Therefore, the X-axis and Y-axis of a'_1, a'_2, and a'_3 and a'_3 in the GCF were determined as follows:

$$a'_1(x_{a'_1}; y_{a'_1}) = b'_1(x_{b'_1}; y_{b'_1}) + \left((R-r).\sin\alpha\ ;\ (R-r).\cos\alpha\right) \tag{10}$$

$$a'_2(x_{a'_2}; y_{a'_2}) = b'_2\left(x_{b'_2}; y_{b'_2}\right) + (0;\ (r-R)) \tag{11}$$

$$a'_3(x_{a'_3}; y_{a'_3}) = b'_3\left(x_{b'_3}; y_{b'_3}\right) + ((R-r).\sin\alpha\ ;\ (R-r).\cos\alpha) \tag{12}$$

Finally, the Z-axis of a'_1, a'_2, and a'_3 was defined by:

$$z_{a'_n} = \pm\sqrt{L^2 - (x_{b'_n} - x_{a'_n})^2 - (y_{b'_n} - y_{a'_n})^2} \tag{13}$$

where n refers to the corner number (n = 1, 2, 3) and L denotes the length of the connecting link between point a and point b. The motion of points a'_1, a'_2, and a'_3 , and a'_3 was activated by Actuator 3, Actuator 4, and Actuator 5, respectively. In order to achieve the expected end-effector position and orientation, the required angle of rotation of the actuator (ϑ) was calculated as follows:

$$\vartheta_n = \frac{z_{a'_n} - z_{a_n}}{2.\pi.r_s} \times 360 \qquad (14)$$

where r_s denotes the radius of the slider wheel required to change the position of points a'_1, a'_2, and a'_3 up and down.

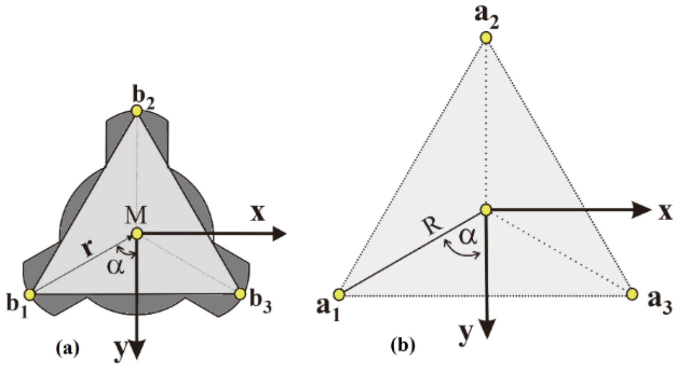

Fig. 2. Robot's platform: (a) Mobile platform and (b) Upper platform.

3. Implementation and Discussion

All the algorithm results in this study have been used to develop a program using Matlab. The program is referred to as an Analytical Inverse Kinematics Simulation (AIKS). The AIKS was examined in terms of its ability to calculate the variables needed to move the end-effector to the required location and orientation. The variables needed by the actuators are ε, λ, ϑ_1, ϑ_2, *and* ϑ_3. Finally, the accuracy of the method was verified using two verification methods. First, by comparing the coordinates of the end-effector using CAD software, and second, by comparing the data generated experimentally using a physical parallel manipulator.

3.1. Angle of rotation calculation

In order to check the applicability of the developed method, one test was conducted. In this test, the end-effector was designed to travel on a complex trajectory, as shown in Figs. 3(a) and (b). Figure 3(a) shows the trajectory of two points on the end-effector, namely point "*a*" and point "*b***.**" Meanwhile**,** Fig. 3(b) presents the orientation and coordinates of the end-effector while following the trajectory. Due to the complex trajectory, the posture of the end-effector was dynamically changed. The rotation of Actuator 1 about the *z*-axis (ε) and the rotation of Actuator 2 about the *y*-axis (λ) are presented in Fig. 3(b). These two variables were given as part of the designed trajectory and orientation of the end-effector. Using the AIKS, the angles of rotation of Actuator 3 (ϑ_1), Actuator 4 (ϑ_2) and Actuator 5 (ϑ_3), could be generated, and the results are presented in Fig. 4(a). The angles of rotation were calculated based on the position of the upper link of the mobile platform on the *z*-axis, a'_1, a'_2, and a'_3. The position of the upper link during the end-effector's travel along the trajectory is presented in Fig. 4(b). From these graphs, it can be seen that the end-effector moved in a very dynamic fashion.

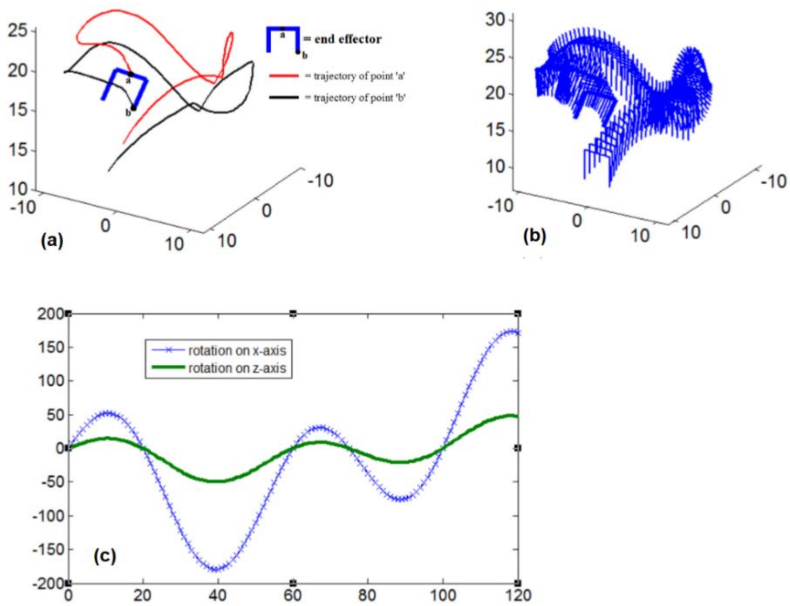

Fig. 3(a) End-effector trajectory, (b) Moving progression of end-effector and (c) Angle of rotation of motor.

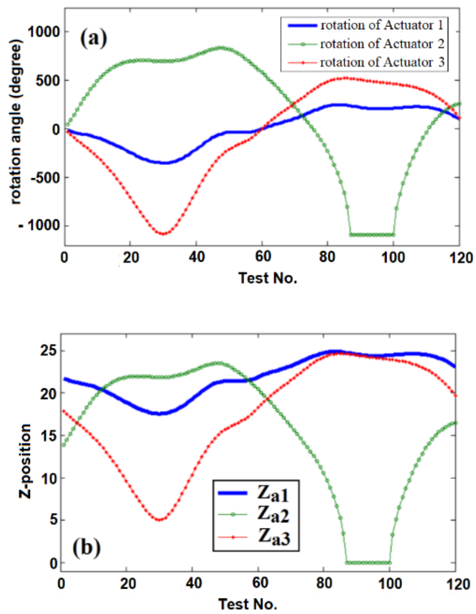

Fig. 4(a) Angle of rotation of motors 3, 4, and 5, and (b) *z*-axis of upper platform.

3.2. Model verification

Although the test showed that the proposed method could be implemented to calculate the inverse kinematics of a 5-DOF parallel manipulator, the accuracy of the method still required verification. In this study, the verification was performed twice.

First, the accuracy of the method was tested by comparing the coordinates of the upper link on the *z*-axis, which were obtained using the AIKS, with the coordinates measured using Siemens-NX. The second verification method involved comparing the calculated data with the data measured experimentally using a physical parallel manipulator.

In the present study, the parallel manipulator was constructed as presented in Fig. 5(a), while the details of the end-effector were as shown in Fig. 5(b). The coordinates of points a'_1, a'_2, and a'_3 on the parallel manipulator were measured using the method shown in Fig. 5(c). In this test, 60 measurements were taken during the movement of the end-effector along the designed trajectory. The trajectory of the end-effector, as shown in Fig. 3(b), was used to verify the accuracy of the proposed method.

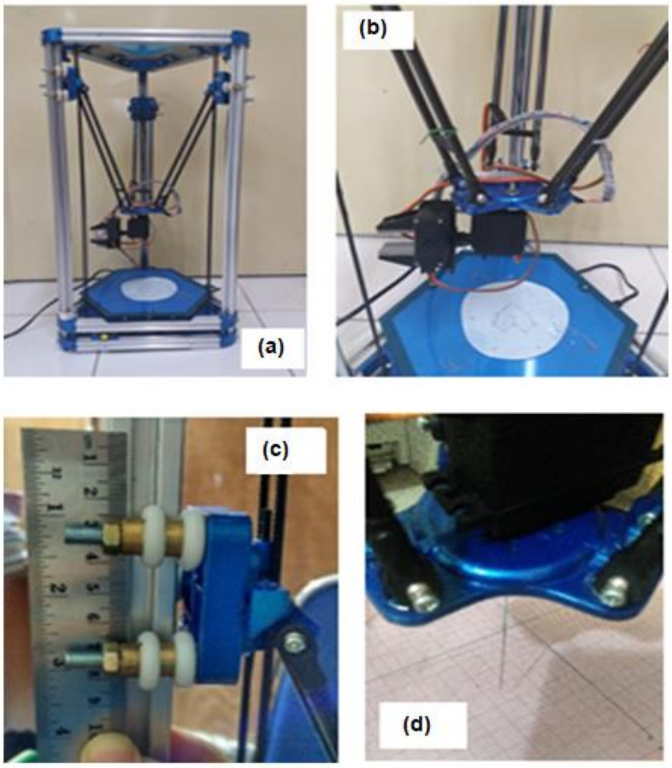

Fig. 5. (a) Construction of parallel manipulator, (b) End-effector, (c) Method used to verify position of mobile platform on *z*-axis and (d) Method used to verify position of end-effector on *x*-axis and *y*-axis.

The results are presented in Fig. 6. The graph in Fig. 6 shows that all the points measured using the physical parallel manipulator very nearly coincided with the designed trajectory generated using the program simulation. This test demonstrated that the developed method is accurate.

The experimental verification was performed by comparing the coordinates of the end-effector while following the designed trajectory, as shown in Fig. 7. In this test, the accuracy of the manipulator in terms of following the trajectory along the two axes, namely the *x*-axis and the *y*-axis, was tested. The method used to measure the coordinate of the end-effector is depicted in Fig. 5(d). The trajectory that was obtained by the physical manipulator was compared with the designed trajectory, as shown in Fig. 7. The deviation was calculated using a statistical method known as the Mean Squared Error (MSE), and the result was 1.01%. Once again, the result shows that the deviations were relatively small. Based on the series of verification test, it can be concluded that the results both prove that the developed method is accurate and demonstrate the precision of the developed parallel manipulator.

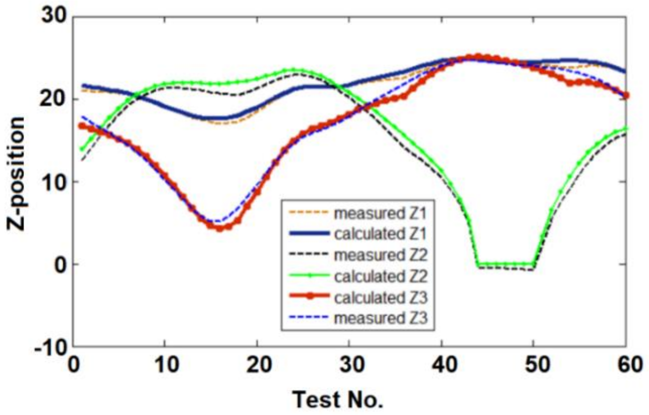

Fig. 6. Model verification using an experimental test.

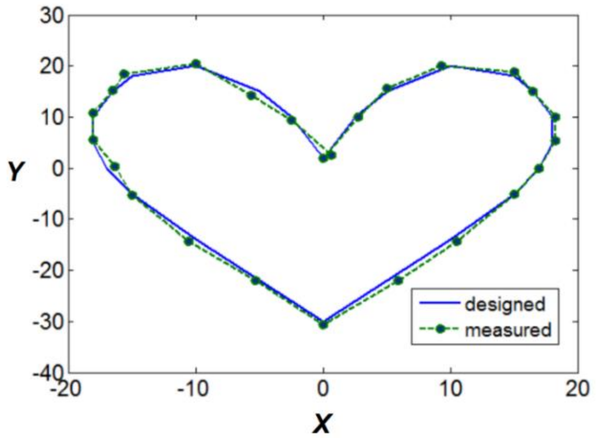

Fig. 7. Verification results concerning accuracy of robot in terms of following programmed trajectory.

3.3. Computational time

The main advantage of the analytical-based approach is the shorter computational time when compared to the numerical-based approach. Therefore, to check the efficiency of the developed method, the computational time of the AIKS was compared to that of the Jacobian method. As discussed in the introduction, the Jacobian method is widely used in calculating the inverse kinematics of robots. For comparison purposes, the trajectory and orientation of the manipulator, as shown in Fig. 3, were used. The angles of rotation of the actuators needed to achieve the designed trajectory and orientation were calculated using both the AIKS and the Jacobian method. In this test, the end-effector travelled along 120 different locations and orientations. The computational time of the proposed method in terms of calculating the inverse kinematics of the robot's motion in two consecutive locations was recorded. The result, as presented in Fig. 8, was the average value of three-time measurements for every motion and every method. From the graph in Fig. 8, it is clear that the computational time of the AIKS is significantly shorter than that of the Jacobian approach. The AIKS took only 1.87 s to calculate the inverse kinematics of the assigned tasks, while the Jacobian method took 29.93 s.

The results showed that the analytical method was computationally more efficient than the numerical method. In this study, the AIKS method was proven to be applicable in supporting inverse kinematics calculations. This means that the proposed method contributes to simplifying the work needed to calculate the inverse kinematics of a parallel manipulator. This represents the main advantage of the proposed method when compared to other methods based on the numerical and statistical approaches.

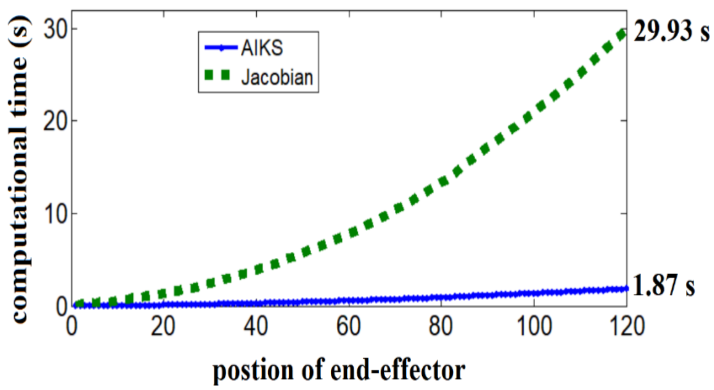

Fig. 8. Computational times of AIKS and Jacobian method.

4. Conclusions

In this research study, a method based on the analytical approach was developed to determine the inverse kinematics of a parallel manipulator. The algorithm was developed for a robot with five DOF, and it was used to develop a simulation program in Matlab. The simulation program was referred to as an Analytical Inverse Kinematics Simulation (AIKS) method. Based on the method's implementation, several concluding remarks can be offered.

- One test was conducted to check the applicability of the developed method in terms of generating the angle of rotation required by the actuator to achieve the desired position. The results demonstrated that the method could be used to provide the required data.

- The verification test proved that both the proposed method and the physical 5-DOF parallel manipulator were accurate.
- Moreover, the comparison test proved that the proposed method was computationally more efficient than the Jacobian method. This is the main advantage of the proposed method when compared to other approaches based on numerical methods.

Acknowledgement

The authors wish to thank the Indonesian Ministry of Research Technology and Higher Education for the financial support provided through the fundamental research grant scheme.

Nomenclatures

r_s	Radius of slider wheel, m
T	Mapping tool

Greek Symbols

α	Angle between *R* and *Y*-axis (Fig. 2), deg.
ε	Rotation angle of actuator 1 (Fig. 1), deg.
λ	Rotation angle of actuator 2 (Fig. 1), deg.
ϑ_n	Rotation angle of actuator on mobile platform (Fig. 1), deg.

Abbreviations

AIKS	Analytical Inverse Kinematics Simulation
CAD	Computer-Aided Design
DOF	Global Coordinate Frame
GCF	World Health Organization
LCF	Local Coordinate Frame
MCF	Mobile Coordinate Frame

References

1. Gough, V.E.; and Whitehall, S.G. (1962). Universal tyre test machine. *Proceedings of the 9th International Technology Congress FISITA*. London, United Kingdom, 117-137.
2. Stewart, D. (1965). A platform with six degrees of freedom: A new form of mechanical linkage which enables a platform to move simultaneously in six degrees of freedom developed by Elliot-Automation. *Aircraft Engineering and Aerospace Technology*, 38(4), 30-35.
3. Cox, D.J.; and Tesar. D. (1981). The dynamic modeling and command signal formulation for parallel multi-parameter robotic devices. *Internal Report CIMAR*, University of Florida, Gainesville, Florida.
4. Pierrot, F.; Reynaud, C.; and Fournier, A. (1990). DELTA: A simple and efficient parallel robot. *Robotica*, 8(2), 105-109.
5. Di Gregorio, R.; and Parenti-Castelli, V. (1998). A translational 3-DOF parallel manipulator. *Advances in Robot Kinematics: Analysis and Control*, 49-58.

6. Li, Y.; and Xu, Q. (2006). Kinematic analysis and design of a new 3-DOF translational parallel manipulator. *Journal of Mechanical Design*, 128(4), 729-737.
7. Zhao, J.; and Badler, N.I. (1994). Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics (TOG)*, 13(4), 313-336.
8. Matikainen, V.; Backman, J.; and Visala, A. (2014). Cartesian control of an advanced tractors rear hitch-damped least-squares solution. *Proceedings of the 19th World Congress of International Federation of Automatic Control (IFAC)*. Cape Town, South Africa, 47(3), 11565-11570.
9. Hu, B. (2014). Formulation of unified Jacobian for serial-parallel manipulators. *Robotics and Computer-Integrated Manufacturing*, 30(5), 460-467.
10. Aristidou, A.; and Lasenby, J. (2009). Inverse kinematics: A review of existing techniques and introduction of a new fast iterative solver. *Technical Report.* University of Cambridge, Department of Engineering.
11. Courty, N.; and Arnaud, E. (2008). Inverse kinematics using sequential Monte Carlo methods. *Proceedings of the International Conference on Articulated Motion and Deformable Objects*. Puerto de Andratx, Mallorca, Spain, 1-10
12. Hecker, C.; Raabe, B.; Enslow, R.W.; Deweese, J.; Maynard J.; and Prooijen, K.v. (2008). Real-time motion retargeting to highly varied user-created morphologies. *ACM Transactions on Graphics (TOG)*, 27(3), 1-11.
13. Olsen, A.L.; and Petersen, H.G. (2011). Inverse kinematics by numerical and analytical cyclic coordinate descent. *Robotica*, 29(4), 619-626.
14. Kenwright, B. (2012). Inverse kinematics-cyclic coordinate descent (CCD). *Journal of Graphics Tools*, 16(4), 177-217.
15. Kiswanto, G.; Hendriko, H.; and Duc, E. (2014). An analytical method for obtaining cutter workpiece engagement during a semi-finish in five axis milling. *Computer-Aided Design*, 55, 81-93.
16. Kiswanto, G.; Hendriko, H.; and Duc, E. (2015). A hybrid analytical and discrete based methodology for determining cutter workpiece engagement in five-axis milling. *International Journal of Advanced Manufacturing Technology*, 80(9-12), 2083-2096.
17. Kiswanto, G.; Hendriko, H.; and Duc, E. (2014). Analytical method for obtaining cut geometry of helical toroidal cutter during semi-finish in 5-axis milling. *Applied Mechanics and Materials*, 541-542, 780-784.
18. Hendriko. (2015). Mathematical model for chip geometry calculation in five-axis milling. *Jurnal Teknologi*, 77(23), 107-112.
19. Hendriko, H.; Kiswanto, G.; Istiyanto, J.; and Duc, E. (2017). Implementation of analytical boundary simulation method for cutting force prediction model in five-axis milling. *Machining Science and Technology*, 22(1), 163-179.
20. Hendriko, H; Nurkhamdi; Jaenudin, J; Muthahar, I.M. (2018). Analytical based invers kinematics method for 5-axis delta robot. *International Journal of Materials, Mechanics and Manufacturing*, 6(4), 264-267.