Taylor & Francis
Taylor & Francis Group

Advanced Robotics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tadr20

Numerical method for inverse kinematics using an extended angle-axis vector to avoid deadlock caused by joint limits*

Masanori Sekiguchi & Naoyuki Takesue

To cite this article: Masanori Sekiguchi & Naoyuki Takesue (2021) Numerical method for inverse kinematics using an extended angle-axis vector to avoid deadlock caused by joint limits*, Advanced Robotics, 35:15, 919-926, DOI: 10.1080/01691864.2021.1928545

To link to this article: https://doi.org/10.1080/01691864.2021.1928545

ADVANCED ROBOTICS
2021, VOL. 35, NO. 15, 919–926
https://doi.org/10.1080/01691864.2021.1928545

FULL PAPER

OPEN ACCESS

Numerical method for inverse kinematics using an extended angle-axis vector to avoid deadlock caused by joint limits*

Masanori Sekiguchi and Naoyuki Takesue

Graduate School of Systems Design, Tokyo Metropolitan University, Tokyo, Japan

ABSTRACT
In this paper, a new inverse kinematics strategy is proposed for avoiding deadlock caused by joint limits. The proposed method is a combination of the conventional numerical method for inverse kinematics and an extended angle-axis vector. The extended angle-axis vector, instead of Euler angles, represents the orientation error. In the proposed method, the rotation direction of the end effector is changed by using the extended angle-axis vector to avoid deadlock. This paper describes the specific parameters included in the extended angle-axis vector that are necessary to change the rotation direction. The effectiveness of the proposed method was verified through simulations using a 9-degree-of-freedom robot. The proposed method is expected to be useful for solving inverse kinematics problems, which remain analytically unresolved.

ARTICLE HISTORY
Received 17 March 2020
Revised 12 August 2020
Accepted line on March 2020. April 2021

KEYWORDS
Inverse kinematics; extended angle-axis vector; joint limits; manipulator; deadlock

1. Introduction

To automate work using an articulated robot, such as an industrial robot arm, it is necessary to generate joint displacement commands that can achieve the required motions. In most cases, the requirements given to the robot during the teaching process are assigned in task space. Therefore, inverse kinematics is required to convert a point in task space to a corresponding point in joint space. Inverse kinematics methods can be divided into two categories: analytical methods for obtaining solutions by geometrical consideration or algebraic transformation [1–3], and numerical methods that perform iterative calculations [4]. If the robot mechanism meets certain conditions, the inverse kinematics problem can be solved analytically; otherwise, numerical methods are used.

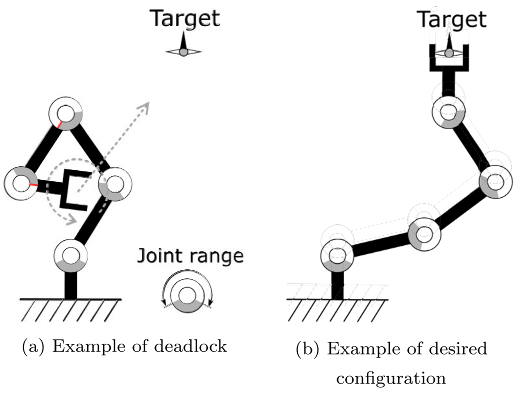
Numerical methods for inverse kinematics based on the Newton–Raphson method or Gauss–Newton method have two main problems: the solution does not converge to a constant value in some cases, and it becomes numerically unstable in the singular configuration [5]. These problems can be solved by using the methods shown in [6, 7]. However, the problem of deadlock (local minimum) due to the limitation of the joint range of motion remains [8]. Figure 1 shows an example of deadlock using a 4-degree-of-freedom (DOF) redundant planar robot. In the numerical method, the calculation result does not converge to the solution when it falls into the deadlock configuration shown in Figure 1(a) during the iterative calculation. This deadlock can be prevented by changing the rotation direction of the end effector. In Figure 1, the end effector has two rotation directions, clockwise and counterclockwise. The direction is selected depending on whether the domain of the orientation error is $[\,0,\ 2\pi\,]$ or $[-2\pi,\ 0\,]$. If a clockwise rotation of the end effector is specified, the deadlock shown in Figure 1(a) does not occur.

In planar robots, the domain of the orientation error can be easily changed because the orientation of the end effector can be represented by one variable. However, when the orientation of the end effector is expressed by three variables, such as Euler angles, the domain of the orientation error depends on the range of the inverse trigonometric function and cannot be changed easily. In addition to the above problems, Euler angles have the problem that gimbal lock and continuity of values are not guaranteed. To solve the deadlock problem, the authors have developed an extended angle-axis vector, which is a new method for expressing orientation errors [9]. In this paper, through application of the extended angle-axis vector, a new numerical inverse kinematics method is proposed for avoiding deadlock.

CONTACT Masanori Sekiguchi sekiguchi-masanori@ed.tmu.ac.jp

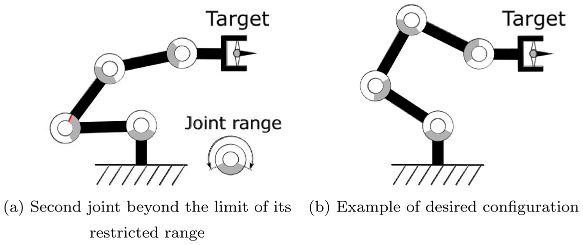
*Portions of this work were previously presented at the 25th Robotics Symposia which was held on-line on March 2020.

Supplemental data for this article can be accessed here. https://doi.org/10.1080/01691864.2021.1928545



(a) Example of deadlock (b) Example of desired configuration

Figure 1. Deadlock caused by joint limits.



(a) Second joint beyond the limit of its restricted range (b) Example of desired configuration

Figure 2. Joint limit avoidance using redundancy.

As shown in Figure 2, the problem of exceeding joint limitations can be solved by using redundancy. The posture shown in Figure 2(b) can be obtained by calculating the differential inverse kinematics, including the orthogonal projection operator to the null space, and several concrete calculation methods have been proposed [10–17]. These are similar to the method applied in this study in that they consider joint limits, but the content is different. The problem of exceeding the joint range of motion can be solved by using the inverse kinematics method described in Section 2. However, locked joints, that is, joints that have reached their limit, cause the deadlock shown in Figure 1. Thus, in this study, a new inverse kinematic strategy was developed to avoid deadlock.

2. Numerical inverse kinematics method

In this section, the numerical inverse kinematics method presented in [6, 7] is briefly described. This section provides the foundation for the proposed method described later.

Let $\boldsymbol{q}_k \in \mathbb{R}^n$ be the joint displacement at iteration k. $\boldsymbol{p}_\mathrm{e}(\boldsymbol{q}_k) \in \mathbb{R}^3$ and $\boldsymbol{R}_\mathrm{e}(\boldsymbol{q}_k) \in SO(3)$ are the position and orientation of the end effector, respectively. Let ${}^\mathrm{d}\boldsymbol{p}_\mathrm{e} \in \mathbb{R}^3$ and ${}^\mathrm{d}\boldsymbol{R}_\mathrm{e} \in SO(3)$ be the desired position and orientation, respectively. Then, the error vector $\boldsymbol{e}_k$ is given by

$$\boldsymbol{e}_k = \begin{bmatrix} {}^\mathrm{d}\boldsymbol{p}_\mathrm{e} - \boldsymbol{p}_\mathrm{e}(\boldsymbol{q}_k) \\ \boldsymbol{\phi}(\boldsymbol{R}_k) \end{bmatrix} \in \mathbb{R}^6, \tag{1}$$

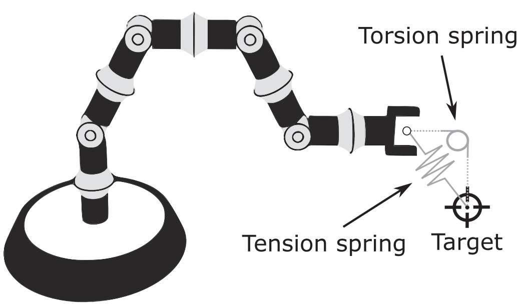


Figure 3. Virtual tension spring and virtual torsion spring.

where $\boldsymbol{R}_k := {}^\mathrm{d}\boldsymbol{R}_\mathrm{e}\boldsymbol{R}_\mathrm{e}^\mathrm{T}(\boldsymbol{q}_k)$ and $\boldsymbol{\phi}(\boldsymbol{R}_k)$ is a function that transforms the rotation matrix $\boldsymbol{R}_k$ into an angle-axis vector (see Appendix A). The angle-axis vector represents the orientation error. As shown in Figure 3, a virtual tension spring and a virtual torsion spring are installed with respect to the error vector $\boldsymbol{e}_k$. Then the elastic energy V_k of the virtual spring is expressed as

$$V_k = \frac{1}{2}\boldsymbol{e}_k^\mathrm{T}\boldsymbol{K}\boldsymbol{e}_k \in \mathbb{R}, \tag{2}$$

where $\boldsymbol{K} \in \mathbb{R}^{6\times 6}$ is the spring constant matrix. The equation for updating the joint displacement to minimize V_k is expressed as

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \boldsymbol{D}_k^{-1}\boldsymbol{J}_k^\mathrm{T}\boldsymbol{K}\boldsymbol{e}_k, \tag{3}$$

$$\boldsymbol{D}_k = \boldsymbol{J}_k^\mathrm{T}\boldsymbol{K}\boldsymbol{J}_k + \left(\frac{1}{2}V_k + \delta\right)\boldsymbol{I}_n, \tag{4}$$

where $\boldsymbol{I}_n \in \mathbb{R}^{n\times n}$ is the identity matrix and $\delta \in \mathbb{R}$ is a small positive constant. The constant δ prevents $\boldsymbol{D}_k$ from becoming irregular when $V_k \simeq 0$ holds. Use of the Cholesky decomposition [18–21] allows the cost of computing the inverse matrix to be reduced, because $\boldsymbol{D}_k$ is a positive definite symmetric matrix.

If the ith joint exceeds the range of motion during the iterative calculation, the following procedure is required:

(1) Reset the joint displacement to the limit of the joint range.
(2) Then, replace the ith column of the Jacobian matrix $\boldsymbol{J}_k$ with a zero vector.

By performing the above two steps, the problem that the joint displacement exceeds the range of motion can be solved. However, deadlock may still occur. Therefore, the next section describes a method for avoiding deadlock by using the extended angle-axis vector.

3. Inverse kinematics using extended angle-axis vector

This section describes the proposed numerical inverse kinematics method using the extended angle-axis vector, which can specify the rotation direction of the end effector in advance. The proposed method can be applied to a robot whose error vector includes the orientation. Note that redundancy is not related to the application scope of the proposed method.

3.1. Application of extended angle-axis vector

In the proposed method, the orientation error vector in Equation (1) is replaced with

$$\boldsymbol{e}_k = \begin{bmatrix} {}^{\mathrm{d}}\boldsymbol{p}_{\mathrm{e}} - \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{q}_k) \\ \widehat{\boldsymbol{\phi}}(\boldsymbol{R}_k, \boldsymbol{\sigma}_k) \end{bmatrix} \in \mathbb{R}^6, \qquad (5)$$

where $\widehat{\boldsymbol{\phi}}(\boldsymbol{R}_k, \boldsymbol{\sigma}_k)$ is a function that transforms the rotation matrix $\boldsymbol{R}_k$ into an extended angle-axis vector [9] (see Appendix B). The extended angle-axis vector includes the following parameters:

$$\boldsymbol{\sigma}_k = \begin{bmatrix} \sigma_{1,k} \\ \sigma_{2,k} \\ \sigma_{3,k} \end{bmatrix}. \qquad (6)$$

The $\boldsymbol{\sigma}_k$ parameter translates the range of the orientation error and changes the rotation direction of the end effector when the numerical inverse kinematics method is applied. Moreover, if $\boldsymbol{\sigma}_k = \mathbf{0}$, the extended angle-axis vector is equivalent to the angle-axis vector.

3.2. Update equation for $\boldsymbol{\sigma}_k$

The $\boldsymbol{\sigma}_k$ component $\sigma_{i,k}$ $(i = 1, 2, 3)$ is updated according to the extended angle-axis vector component $\widehat{\phi}_i(\boldsymbol{R}_k, \boldsymbol{\sigma}_k)$:

$$\sigma_{i,k+1} = \begin{cases} 0 & \left(\left|\widehat{\phi}_i(\boldsymbol{R}_k, \boldsymbol{\sigma}_k)\right| < \pi\right) \\ \sigma_{i,k} & (\text{otherwise}). \end{cases} \qquad (7)$$

Appendix C shows an example of the numerical calculation.

Each axes x, y and z has two types of rotation direction (clockwise or counterclockwise). Therefore, the main rotation direction of the end effector has eight (2^3) options, and the following eight initial values of $\boldsymbol{\sigma}_k$ are prepared:

$$\boldsymbol{\sigma}_0 = \begin{cases} [\ \pi \quad \pi \quad \pi\ ]^{\mathrm{T}} & (\text{case1}) \\ [\ \pi \quad \pi \quad -\pi\ ]^{\mathrm{T}} & (\text{case2}) \\ [\ \pi \quad -\pi \quad \pi\ ]^{\mathrm{T}} & (\text{case3}) \\ [\ \pi \quad -\pi \quad -\pi\ ]^{\mathrm{T}} & (\text{case4}) \\ [\ -\pi \quad \pi \quad \pi\ ]^{\mathrm{T}} & (\text{case5}) \\ [\ -\pi \quad \pi \quad -\pi\ ]^{\mathrm{T}} & (\text{case6}) \\ [\ -\pi \quad -\pi \quad \pi\ ]^{\mathrm{T}} & (\text{case7}) \\ [\ -\pi \quad -\pi \quad -\pi\ ]^{\mathrm{T}} & (\text{case8}). \end{cases} \qquad (8)$$

The rotation direction of the end effector is determined according to the above initial values.

In the proposed method, deadlock can be prevented by the simple principle that the numerical calculation is carried out again after resetting all joint displacements to the initial configuration and changing the rotation direction of the end effector. It is impossible to know in advance which $\boldsymbol{\sigma}_0$ is effective for avoiding deadlock. Hence, cases 1–8 are executed in order, i.e. the maximum number of trial calculations is eight.

4. Evaluation by simulations

In this section, the effectiveness of the proposed method is shown through a simulation experiment that reveals the difference between using and not using the extended angle-axis vector. Figure 4 shows the 9-DOF robot model used in the simulations. The joint range of motion was set to $[-90^\circ, 90^\circ]$ for all joints. Most non-redundant robots are designed so that inverse kinematics can be solved analytically. Therefore, a 9-DOF robot, whose inverse kinematics cannot be solved analytically, was used in this experiment.

In preparation for the comparative experiment, 100 targets were generated from 100 postures. As shown in Figure 5, 100 postures were generated by randomly setting joint displacement from the range of motion. In Figure 6, the target position and orientation are represented using a sphere and three orthogonal pointers, respectively. The center of the sphere is the target position, and the three pointers represent the target orientation.

Table 1 shows the success rate when 100 inverse kinematics problems were solved using the conventional method described in Section 2 and the proposed method. When the iterative calculation result produced $V_k = 0$, it was regarded as a success, and otherwise it was regarded as a failure. In the conventional method, failure due to deadlock occurred in 10 out of 100 tests. In contrast, all

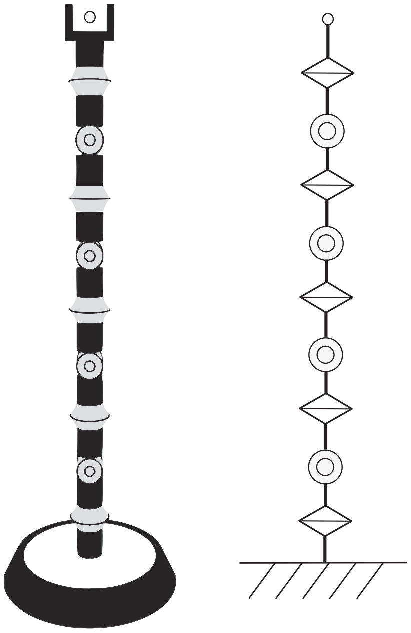


Figure 4. 9-DOF robot used in simulations.

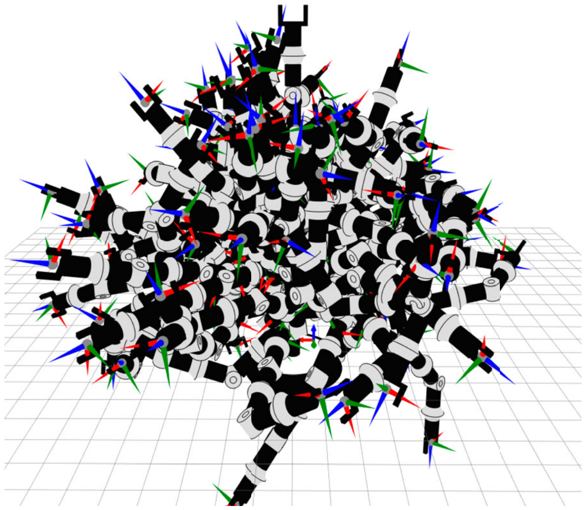


Figure 5. 100 postures generated by randomly setting $q_i \in [-90^\circ, 90^\circ]$ ($i = 1, 2, \ldots, 9$).

tests with the proposed method were successful because even if the first numerical calculation results fell into a deadlock, it can be avoided by changing the rotation direction of the end effector. In this experiment, 10 examples succeeded in calculating inverse kinematics by using only the proposed method. Three of these examples are shown in Figure 7.

If all cases of $\boldsymbol{\sigma}_0$ are executed, multiple inverse kinematic solutions may be obtained. However, in this experiment, the calculation was terminated when joint displacements with $V_k = 0$ were first obtained.

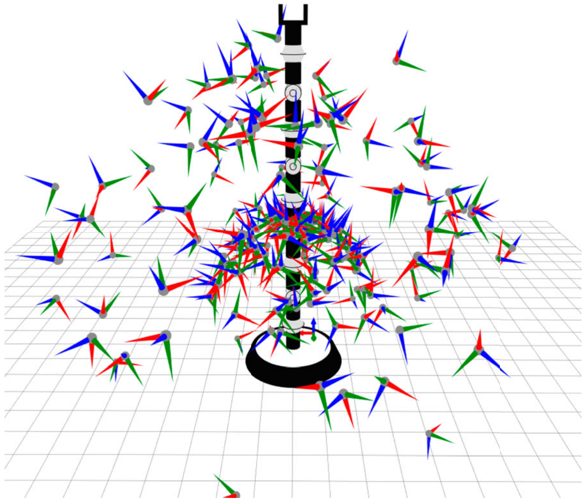


Figure 6. 100 targets generated by randomly setting postures.

Table 1. Comparison of outcomes with and without the proposed method. 100 inverse kinematics problem was generated from the targets shown in Figure 6.

	Proposed	Conventional
Success	100 times	90 times
Failure	0 times	10 times

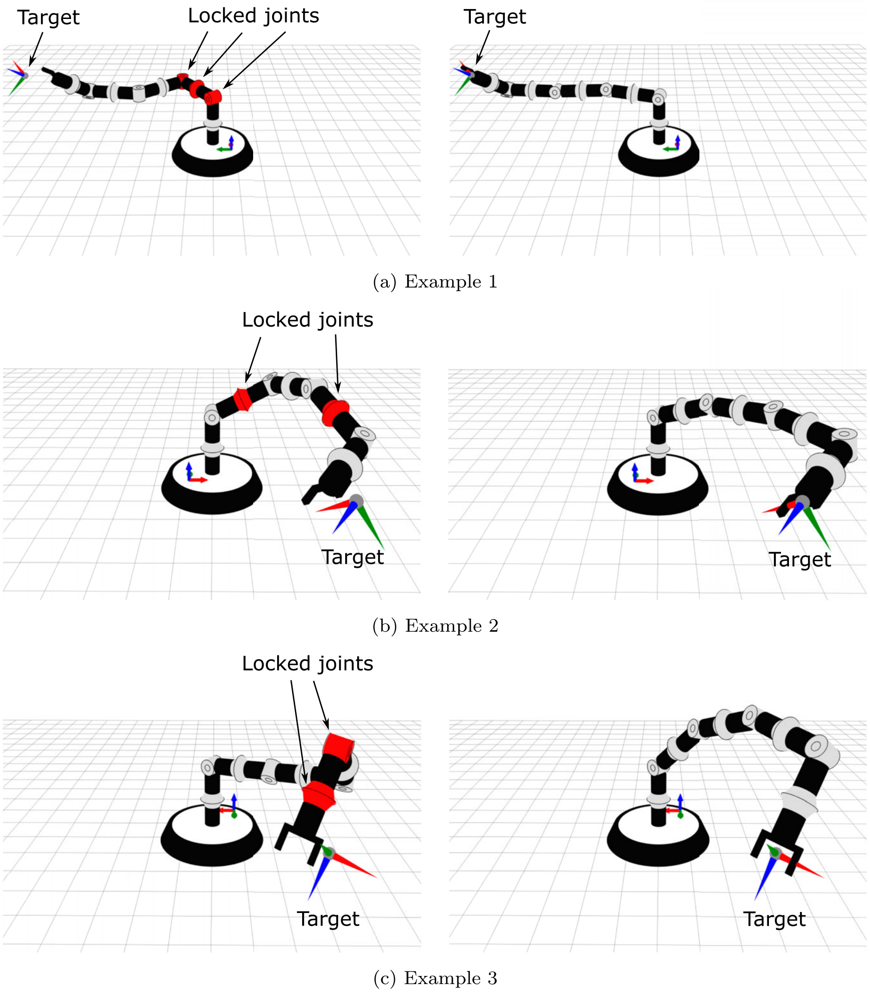
5. Discussion

This section describes remaining issues of the proposed method, differences from a method using the random initial configuration, and selection method of inverse kinematics solution.

5.1. Remaining issues

This section discusses whether the proposed method always succeeds in avoiding deadlock. The conclusion is that even when the proposed method is used, it is not always possible to find an inverse kinematics solution for the reasons described below. Thus, there is room for improvement on these points.

One of the reasons is that the extended angle-axis vector cannot represent an orientation error whose magnitude exceeds 2π. In addition, under certain conditions, deadlock cannot be avoided only by changing the rotation direction of the end effector. An example is shown in Figure 8. To avoid the deadlock shown in Figure 8(a), it is necessary to turn back the second joint from the lower right to the upper left, as shown in Figure 8(b). However, such joint folding is not possible even when the proposed method is used.

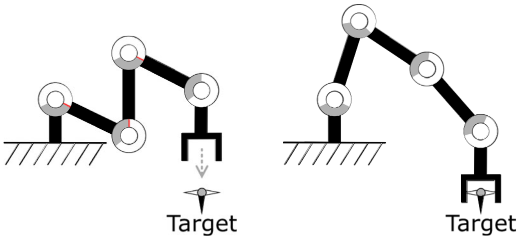


(a) Example 1

(b) Example 2

(c) Example 3

Figure 7. Inverse kinematics solutions obtained by conventional method (left) and proposed method (right).



(a) Example of unresolved deadlock

(b) Example of desired configuration

Figure 8. Example of deadlock that cannot be solved by the proposed method.

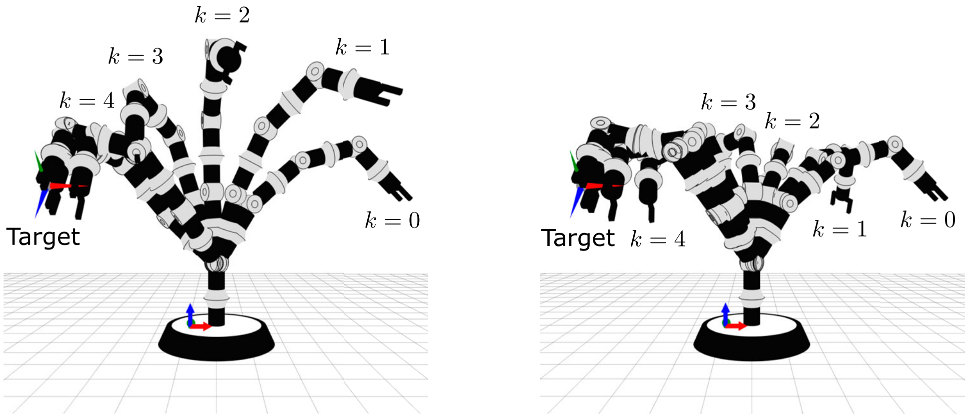
5.2. Comparison with a method using the random initial configuration

Another way to avoid deadlock due to joint range of motion is to randomly change the initial configuration and rerun the iteration. This method is an effective means for non-redundant robots whose inverse kinematic solutions are uniquely determined. However, in a redundant robot, changing the initial configuration randomly causes problems. Since the inverse kinematic solutions of a redundant robot is not unique, randomly generating the initial configuration leads to unwanted changes in joint angle. On the other hand, such a problem does not occur if the rotation direction of the end effector is changed instead of the initial configuration as in the proposed method.

5.3. How to select an inverse kinematics solution

Multiple inverse kinematics solutions may be obtained by changing the rotation direction of the end effector. Two examples are shown in Figure 9. Hence, this section presents one of the simple methods to uniquely determine the inverse kinematics solution.

Suppose two inverse kinematics solutions $\boldsymbol{q}_{f1}$ and $\boldsymbol{q}_{f2}$ are obtained. In this case, the inverse kinematics solution can be uniquely determined by comparing the two norms



(a) Example 1

(b) Example 2

Figure 9. Multiple inverse kinematics solutions obtained by changing the rotation direction of the end effector.

$||\boldsymbol{q}_{f1} - \boldsymbol{q}_0||$ and $||\boldsymbol{q}_{f2} - \boldsymbol{q}_0||$. The one with the smaller norm is regarded as the inverse kinematics solution.

6. Conclusion

In this paper, a numerical inverse kinematics method using the extended angle-axis vector has been proposed to solve the deadlock problem caused by the limitation of the joint range of motion. The simulation results showed that the proposed method could solve several inverse kinematics problems that could not be solved by conventional method. The proposed method is expected to be useful for solving inverse kinematics problems, which remain analytically unresolved. However, as discussed in Section 5, it remains an issue that even when the proposed method is used, it is not always possible to avoid deadlock.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This paper is based on results obtained from a project, JPNP18002, subsidized by the New Energy and Industrial Technology Development Organization (NEDO).

Notes on contributors

Masanori Sekiguchi received a BE and ME degrees from Tokyo Metropolitan University in 2017 and 2019, respectively. Since 2019, he has been a PhD student at the Department of Systems Design, Tokyo Metropolitan University. His research interests include kinematics, motion planning and industrial applications.

Naoyuki Takesue received BE and ME degrees from The University of Electro-Communications, respectively, in 1995 and 1997. In 2000, he received a PhD in Engineering from Osaka University. He joined Osaka University in 2000 and then joined Nagoya Institute of Technology in 2003. He joined Tokyo Metropolitan University in 2008 as an associate professor. His research interests include mechanism design, motion control of manipulator, industrial applications, physically-assistive robots, and aerial/aquatic/mobile robots.

ORCID

Naoyuki Takesue http://orcid.org/0000-0002-8029-5480

References

[1] Tian X, Xu Q, Zhan Q. An analytical inverse kinematics solution with joint limits avoidance of 7-DOF anthropomorphic manipulators without offset. J Franklin Inst. 2021;358(2):1252–1272.

[2] Yu C, Jin M, Liu H. An analytical solution for inverse kinematic of 7-DOF redundant manipulators with offset-wrist. IEEE Int Conf Mechatron Autom. 2012;92–97.

[3] Guo W, Li R, Cao C, et al. Kinematics analysis of a novel 5-DOF hybrid manipulator. Int J Autom Technol. 2015;9(6):765–774.

[4] Deo AS, Walker ID. Overview of damped least-squares methods for inverse kinematics of robot manipulators. J Int Rob Syst. 1995;14(1):43–68.

[5] Goldenberg A, Benhabib B, Fenton R. A complete generalized solution to the inverse kinematics of robots. IEEE J Rob Autom. 1985;1(1):14–20.

[6] Sekiguchi M, Takesue N. Stable numerical solution of inverse kinematics in singular posture and unsolvable problem based on minimization of elastic energy of virtual spring. J Rob Soc Jpn. 2018;36(9):645–653.(in Japanese).

[7] Sugihara T. Solvability-unconcerned inverse kinematics by Levenberg-Marquardt method. IEEE Trans Robot. 2011;27(5):984–991.

[8] Takesue N, Sekiguchi M. Cooperative and robotized system and expansion of kinematics. J Rob Soc Jpn. 2019;37(10):919–922. (in Japanese).

[9] Sekiguchi M, Takesue N. A method for calculating angle-axis vector at singular points and a proposal of extended

angle-axis vector. J Rob Soc Jpn. 2019;37(8):726–734.(in Japanese).

[10] Ito M, Shibata M. Non-delayed visual servo control admitting joint range of motion maximally. IEEJ Trans Ind Appl. 2012;132(5):588–595.

[11] Liegeois A. Automatic supervisory control of the configuration and behavior of multibody mechanisms. IEEE Trans Syst Man Cybern. 1977;7(12):868–871.

[12] Zghal H, Dubey RV, Euler JA. Efficient gradient projection optimization for manipulators with multiple degrees of redundancy. IEEE Int Conf Rob Autom. 1990;2:1006–1011.

[13] Marchand E, Chaumette F, Rizzo A. Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing. Proc IEEE/RSJ Int Conf Int Rob Syst. 1996;3:1083–1090.

[14] Tasai M-J. Workspace Geometric Characterization and Manipulability of Industrial Robots, Ph.D. dissertation, Mechanical Engineering, Ohio State University, 1986.

[15] Chan TF, Dubey RV. A weighted Least-Norm solution based scheme for avoiding joint limits for redundant joint manipulators. IEEE Trans Rob Autom. 1995;11(2):286–292.

[16] Ito M, Kawatsu K, Shibata M. Kinematic control of redundant manipulators for admitting joint range of motion maximally. IEEE J Ind Appl. 2017;6(4):278–285.

[17] Wan J, Wu H, Ma R, et al. A study on avoiding joint limits for inverse kinematics of redundant manipulators using improved clamping weighted least-norm method. J Mech Sci Technol. 2018;32(3):1367–1378.

[18] Krishnamoorthy A, Menon D. Matrix inversion using cholesky decomposition. 2013 Signal Processing: Algorithms, Architectures, Arrangements, and Applications, IEEE. 2013;70–72.

[19] Bojanczyk AW, Brent RP, van Dooren P, et al. A note on downdating the cholesky factorization. SIAM J Sci Stat Comput. 1987;8(3):210–221.

[20] Higham NJ. Cholesky factorization. WIREs Comput Stat. 2009;1(2):251–254.

[21] Li J, Widlund OB. FETI-DP BDDC, and block cholesky methods. Int J Numer Methods Eng. 2006;66(2):250–271.

Appendices

Appendix 1. Angle-axis vector

For any rotation matrix $\boldsymbol{R} = [r_{ij}] \in SO(3)$, the vector $\boldsymbol{\ell}$ is defined as

$$\boldsymbol{\ell} := \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} \in \mathbb{R}^3. \tag{A1}$$

Then, the angle-axis vector is expressed as

$$\boldsymbol{\phi}(\boldsymbol{R}) = \begin{cases} \dfrac{\operatorname{atan2}\big(\|\boldsymbol{\ell}\|,\ \operatorname{tr}\boldsymbol{R} - 1\big)}{\|\boldsymbol{\ell}\|}\boldsymbol{\ell} & (\|\boldsymbol{\ell}\| \neq 0) \\ \boldsymbol{0} & (\|\boldsymbol{\ell}\| = 0 \text{ and } \operatorname{tr}\boldsymbol{R} = 3) \\ \text{see below} & (\|\boldsymbol{\ell}\| = 0 \text{ and } \operatorname{tr}\boldsymbol{R} = -1), \end{cases} \tag{A2}$$

where $\operatorname{tr}\boldsymbol{R} = r_{11} + r_{22} + r_{33}$. If $\|\boldsymbol{\ell}\| = 0$ and $\operatorname{tr}\boldsymbol{R} = -1$, the rotation matrix $\boldsymbol{R}$ is symmetrical. The angle-axis vector where $\|\boldsymbol{\ell}\| = 0$ and $\operatorname{tr}\boldsymbol{R} = -1$ can be calculated as follows according to the sign of the non-diagonal component of the matrix $\boldsymbol{R}$:

(1) $r_{21} = r_{12} \geq 0,\ r_{32} = r_{23} \geq 0,\ r_{13} = r_{31} \geq 0$

$$\boldsymbol{\phi}(\boldsymbol{R}) = \pm\frac{\sqrt{2}\pi}{2}\begin{bmatrix} \sqrt{r_{11}+1} \\ \sqrt{r_{22}+1} \\ \sqrt{r_{33}+1} \end{bmatrix}, \tag{A3}$$

(2) $r_{21} = r_{12} \geq 0,\ r_{32} = r_{23} < 0,\ r_{13} = r_{31} \leq 0$

$$\boldsymbol{\phi}(\boldsymbol{R}) = \pm\frac{\sqrt{2}\pi}{2}\begin{bmatrix} \sqrt{r_{11}+1} \\ \sqrt{r_{22}+1} \\ -\sqrt{r_{33}+1} \end{bmatrix}, \tag{A4}$$

(3) $r_{21} = r_{12} < 0,\ r_{32} = r_{23} \leq 0,\ r_{13} = r_{31} \geq 0$

$$\boldsymbol{\phi}(\boldsymbol{R}) = \pm\frac{\sqrt{2}\pi}{2}\begin{bmatrix} \sqrt{r_{11}+1} \\ -\sqrt{r_{22}+1} \\ \sqrt{r_{33}+1} \end{bmatrix}, \tag{A5}$$

(4) $r_{21} = r_{12} \leq 0,\ r_{32} = r_{23} \geq 0,\ r_{13} = r_{31} < 0$

$$\boldsymbol{\phi}(\boldsymbol{R}) = \pm\frac{\sqrt{2}\pi}{2}\begin{bmatrix} -\sqrt{r_{11}+1} \\ \sqrt{r_{22}+1} \\ \sqrt{r_{33}+1} \end{bmatrix}. \tag{A6}$$

The derivation of the angle-axis vector is described in [9].

Appendix 2. Extended angle-axis vector

The extended angle-axis vector $\widehat{\boldsymbol{\phi}}(\boldsymbol{R}, \boldsymbol{\sigma}) \in \mathbb{R}^3$ is represented by the linear sum of the angle-axis vector $\boldsymbol{\phi}(\boldsymbol{R})$ and the correction term ${}^{\mathrm{c}}\boldsymbol{\phi}(\boldsymbol{\sigma})$ as

$$\widehat{\boldsymbol{\phi}}(\boldsymbol{R}, \boldsymbol{\sigma}) = \boldsymbol{\phi}(\boldsymbol{R}) + {}^{\mathrm{c}}\boldsymbol{\phi}(\boldsymbol{\sigma}). \tag{A7}$$

The method for calculating the correction term ${}^{\mathrm{c}}\boldsymbol{\phi}(\boldsymbol{\sigma})$ is described below. The following variables are calculated using the angle-axis vector component $\phi_i(\boldsymbol{R})$ $(i = 1, 2, 3)$:

$$\theta_i = \operatorname{sign}\big(\phi_i(\boldsymbol{R})\big) \cdot \|\boldsymbol{\phi}(\boldsymbol{R})\| \quad (i = 1, 2, 3), \tag{A8}$$

$$\boldsymbol{n}_x = \frac{1}{\|\boldsymbol{\phi}(\boldsymbol{R})\|}\begin{bmatrix} |\phi_1(\boldsymbol{R})| \\ 0 \\ 0 \end{bmatrix}, \tag{A9}$$

$$\boldsymbol{n}_y = \frac{1}{\|\boldsymbol{\phi}(\boldsymbol{R})\|}\begin{bmatrix} 0 \\ |\phi_2(\boldsymbol{R})| \\ 0 \end{bmatrix}, \tag{A10}$$

$$\boldsymbol{n}_z = \frac{1}{\|\boldsymbol{\phi}(\boldsymbol{R})\|}\begin{bmatrix} 0 \\ 0 \\ |\phi_3(\boldsymbol{R})| \end{bmatrix}, \tag{A11}$$

Where, if $\|\boldsymbol{\phi}(\boldsymbol{R})\| = 0$, $\boldsymbol{n}_x$, $\boldsymbol{n}_y$, and $\boldsymbol{n}_z$ are defined as the fundamental vectors.

The correction term ${}^{\mathrm{c}}\boldsymbol{\phi}(\boldsymbol{\sigma}) \in \mathbb{R}^3$ can be obtained by the following equations:

$$ {}^{\mathrm{c}}\boldsymbol{\phi}(\boldsymbol{\sigma}) = {}^{\mathrm{c}}\theta_1 \boldsymbol{n}_x + {}^{\mathrm{c}}\theta_2 \boldsymbol{n}_y + {}^{\mathrm{c}}\theta_3 \boldsymbol{n}_z, \tag{A12} $$

$$ {}^{\mathrm{c}}\theta_i = \begin{cases} 2\pi & \begin{pmatrix} \sigma_i > 0 \quad \text{and} \\ -\pi \le \theta_i \le -\pi + \sigma_i \end{pmatrix} \\ -2\pi & \begin{pmatrix} \sigma_i < 0 \quad \text{and} \\ \pi + \sigma_i \le \theta_i \le \pi \end{pmatrix} \\ 0 & \text{(otherwise).} \end{cases} \tag{A13} $$

Appendix 3. Example calculation of the extended angle-axis vector

An example of the calculation of the extended angle-axis vector is shown in Table A1. In addition, ${}^{\mathrm{d}}\boldsymbol{R}_{\mathrm{e}}$ and four types of rotation matrix $\boldsymbol{R}_{\mathrm{e}}(\boldsymbol{q}_k)$ corresponding to Table A1 are shown in Figure A1.

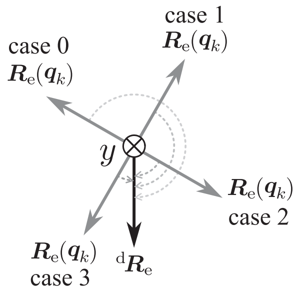

case 1
$\boldsymbol{R}_{\mathrm{e}}(\boldsymbol{q}_k)$
case 0
$\boldsymbol{R}_{\mathrm{e}}(\boldsymbol{q}_k)$
y
$\boldsymbol{R}_{\mathrm{e}}(\boldsymbol{q}_k)$
case 2
$\boldsymbol{R}_{\mathrm{e}}(\boldsymbol{q}_k)$
case 3
${}^{\mathrm{d}}\boldsymbol{R}_{\mathrm{e}}$

Figure A1. ${}^{\mathrm{d}}\boldsymbol{R}_{\mathrm{e}}$ and four types of rotation matrix $\boldsymbol{R}_{\mathrm{e}}(\boldsymbol{q}_k)$ corresponding to Table A1.

Table A1. Example calculation of the angle-axis vector $\boldsymbol{\phi}(\boldsymbol{R}_k)$, correction term ${}^{\mathrm{c}}\boldsymbol{\phi}(\boldsymbol{\sigma})$, and extended angle-axis vector $\widehat{\boldsymbol{\phi}}(\boldsymbol{R}_k, \boldsymbol{\sigma}_k)$.

Case	$\boldsymbol{R}_k = {}^{\mathrm{d}}\boldsymbol{R}_{\mathrm{e}}\boldsymbol{R}_{\mathrm{e}}^{\mathrm{T}}(\boldsymbol{q}_k)$	$\boldsymbol{\sigma}_k$	$\widehat{\boldsymbol{\phi}}(\boldsymbol{R}_k, \boldsymbol{\sigma}_k)$		$\boldsymbol{\phi}(\boldsymbol{R}_k)$		${}^{\mathrm{c}}\boldsymbol{\phi}(\boldsymbol{\sigma}_k)$
0	$\begin{bmatrix} \cos\left(-\frac{4}{6}\pi\right) & 0 & \sin\left(-\frac{4}{6}\pi\right) \\ 0 & 1 & 0 \\ -\sin\left(-\frac{4}{6}\pi\right) & 0 & \cos\left(-\frac{4}{6}\pi\right) \end{bmatrix}$	$\begin{bmatrix} 0 \\ \pi \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ \frac{8}{6}\pi \\ 0 \end{bmatrix}$	$=$	$\begin{bmatrix} 0 \\ -\frac{4}{6}\pi \\ 0 \end{bmatrix}$	$+$	$\begin{bmatrix} 0 \\ 2\pi \\ 0 \end{bmatrix}$
1	$\begin{bmatrix} \cos\frac{5}{6}\pi & 0 & \sin\frac{5}{6}\pi \\ 0 & 1 & 0 \\ -\sin\frac{5}{6}\pi & 0 & \cos\frac{5}{6}\pi \end{bmatrix}$	$\begin{bmatrix} 0 \\ \pi \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ \frac{5}{6}\pi \\ 0 \end{bmatrix}$	$=$	$\begin{bmatrix} 0 \\ \frac{5}{6}\pi \\ 0 \end{bmatrix}$	$+$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$
2	$\begin{bmatrix} \cos\frac{2}{6}\pi & 0 & \sin\frac{2}{6}\pi \\ 0 & 1 & 0 \\ -\sin\frac{2}{6}\pi & 0 & \cos\frac{2}{6}\pi \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ \frac{2}{6}\pi \\ 0 \end{bmatrix}$	$=$	$\begin{bmatrix} 0 \\ \frac{2}{6}\pi \\ 0 \end{bmatrix}$	$+$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$
3	$\begin{bmatrix} \cos\left(-\frac{1}{6}\pi\right) & 0 & \sin\left(-\frac{1}{6}\pi\right) \\ 0 & 1 & 0 \\ -\sin\left(-\frac{1}{6}\pi\right) & 0 & \cos\left(-\frac{1}{6}\pi\right) \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -\frac{1}{6}\pi \\ 0 \end{bmatrix}$	$=$	$\begin{bmatrix} 0 \\ -\frac{1}{6}\pi \\ 0 \end{bmatrix}$	$+$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$