Short Papers

Solvability-Unconcerned Inverse Kinematics by the Levenberg–Marquardt Method

Tomomichi Sugihara, *Member, IEEE*

***Abstract*—A robust numerical solution to the inverse kinematics is proposed based on the Levenberg–Marquardt (LM) method, where the squared norm of residual of the original equation with a small bias is used for the damping factor. A rather simple idea remarkably improves the numerical stability and the convergence performance, even in unsolvable cases. Discussion is done through an investigation of the condition number of the coefficient matrix. Comparison tests with conventional methods show that only the proposed method succeeds in all cases. It frees operators from being careful about the target position-orientation assignment of effectors so that it facilitates easy robot motion designs and remote operations.**

***Index Terms*—Kinematics, numerical ill-posedness, singularity, redundant robots.**

I. Introduction

Inverse kinematics (IK) is one of the fundamental computations in robotics, but it still has some mathematical difficulties. The objective of the problem is to find a robot configuration that satisfies specified constraints about the position and orientation of a link of interest (the end effector in typical cases). As Pieper [1] indicated, it comes down to the root finding of simultaneous multidimensional polynomial equations. Except for some special mechanisms [1]–[4], it is hard to solve the problem analytically so that numerical solutions are utilized in many cases.

The Newton–Raphson (NR) method is often used for numerical solution of IK. It requires the basic Jacobian matrix [5], where its efficient computation methods are already available [6], [7]. The NR method, however, succeeds if and only if the function of the original equation has the inverse function and the original equation is solvable. From the viewpoint of the kinematics, the former condition means that the robot is required to be nonredundant, namely, the number of motion constraints is the same with the degree of freedom (DOF) of the robot, and the robot does not pass through the singular points during the movement from the initial configuration to the final configuration. The latter means that the desired position and orientation of the link of interest needs to be within the workspace of the robot. Because of those limitations, the global convergence is not guaranteed on the NR method so that it largely depends on the initial value.

Manuscript received August 22, 2010; revised February 6, 2011; accepted April 22, 2011. Date of publication June 7, 2011; date of current version October 6, 2011. This paper was recommended for publication by Associate Editor C. Torras and Editor G. Oriolo upon evaluation of the reviewers' comments. This work was supported in part by the Grant-in-Aid for Young Scientists (A) #22680018, the Japan Society for the Promotion of Science and in part by "The Kyushu University Research Superstar Program," based on the budget of Kyushu University allocated under President's initiative. This paper is an English version of a Japanese paper published in the *Journal of the Robotics Society of Japan*, vol. 29, no. 3, pp. 269–277. An earlier version of this work was presented at the 2009 IEEE-RAS International Conference on Humanoid Robots, Paris, France.

The author is with the Graduate School of Engineering, Department of Adaptive Machine Systems, Osaka University, 565-0871 Suita, Japan (e-mail: zhidao@ieee.org).

Digital Object Identifier 10.1109/TRO.2011.2148230

The problem of singularity is strongly related to the property of the basic Jacobian matrix, which has been discussed mainly in the studies of the differential inverse kinematics (DIK) [6], [8]– [12]. Whitney [6] proposed to solve this problem by the use of Moore–Penrose's generalized inverse matrix (MP-inverse matrix). The MP-inverse matrix that can be computed regardless of the rank of the original Jacobian matrix, minimizes the residual of the constraint equation by the minimum deviation of configuration. Nakamura and Hanafusa [10] pointed out that this method cannot resolve the problem that the configuration chatters in the vicinity of the singular points and proposed the singularity-robust inverse matrix (SR-inverse matrix) into which the damping factor was introduced. Wampler [11] also proposed a similar method, and referred that it was involved in a framework of the Levenberg–Marquardt (LM) method [13], [14].

The problem of solvability is as severe as that of singularity. A more difficulty is that it is hard to know if the equation is solvable or not in advance in most situations—it is burdensome for operators to assign carefully the goal position and orientation of the link of interest to be within the workspace in order to guarantee the solvability in the motion design and remote control of large-scale structure-varying kinematic chains [15] such as humanoid robots. A rational idea is to substitute the minimization of residual for the above root finding problem in IK. Several studies that are based on this idea used the steepest descent (SD) method [16], [17] and the Variable metric (VM) method [18], [19]. However, the convergence speed of the former is slow, while the latter is less reliable since it frequently falls into local minima. Reasonably rapid solutions are in a class of gradient methods that utilizes DIK at each step of iterations. The LM method shows a high stability of computation among them. Although the convergence performance of the LM method depends on the choice of the damping factor, this issue has not been sufficiently discussed so far.

This paper proposes a way to choose the damping factor of the LM method for IK by which the computation becomes robust against the problems of both singularity and solvability and fast convergent. Robustness and convergence performance are achieved by a rather simple method to use the squared norm of residual for the damping factor. A similar method was proposed by Chan and Lawrence [20], which has a problem that the computation becomes unstable if the goal position and orientation of the link of interest is set near the singular points. The proposed method resolved this problem by slightly biasing the damping factor. The performance was evaluated on a 12-DOF redundant manipulator model with comparisons to conventional methods, where only the proposed method marked 100% success in all the tests.

II. Review: Minimization Approach to Inverse Kinematics

This section summarizes and reviews the conventional numerical methods, particularly the ones based on minimization approaches for IK.

The robot kinematics is represented by a set of algebraic constraint equations on the joint configuration vector $\boldsymbol{q} = [q_1\, q_2 \ldots q_n]^{\mathrm{T}} \in \mathbb{R}^n$, where n is the number of joints. Limitation of the joint motion range is not taken into account in this paper. Let us define the residual of ith constraint equation $\boldsymbol{e}_i \in \mathbb{R}^3$ as

$$\boldsymbol{e}_i(\boldsymbol{q}) \equiv \begin{cases} {}^d\boldsymbol{p}_i - \boldsymbol{p}_i(\boldsymbol{q}) & \text{(position constraint)} \\ \boldsymbol{a}({}^d\boldsymbol{R}_i \boldsymbol{R}_i^{\mathrm{T}}(\boldsymbol{q})) & \text{(orientation constraint)} \end{cases} \quad (1)$$

where $\boldsymbol{p}_i \in \mathbb{R}^3$ and ${}^d\boldsymbol{p}_i \in \mathbb{R}^3$ are the position and its desired value of the link of interest, respectively, and $\boldsymbol{R}_i \in SO(3)$ and ${}^d\boldsymbol{R}_i \in SO(3)$ are the orientation and its desired value of the link of interest, respectively. In addition, $\boldsymbol{a}(\boldsymbol{R}) \in \mathbb{R}^3$ for an arbitrary $\boldsymbol{R} \in SO(3)$ means the equivalent angle-axis vector, the way to compute which is described in Appendix A. The constraints include ones due to tasks and the others due to mechanisms such as closed loops, while they are not distinguished hereafter. Let the number of all the constraints $3\,m$ and define the residual vector $\boldsymbol{e}(\boldsymbol{q}) \in \mathbb{R}^{3m}$ as

$$\boldsymbol{e}(\boldsymbol{q}) \equiv [\, \boldsymbol{e}_1^{\mathrm{T}}(\boldsymbol{q}) \quad \boldsymbol{e}_2^{\mathrm{T}}(\boldsymbol{q}) \quad \dots \quad \boldsymbol{e}_m^{\mathrm{T}}(\boldsymbol{q}) \,]^{\mathrm{T}}. \tag{2}$$

IK comes down to solving the following nonlinear equation:

$$\boldsymbol{e}(\boldsymbol{q}) = \boldsymbol{0}. \tag{3}$$

The conventional IK that is based on the NR method tries to reach $\boldsymbol{q} = \boldsymbol{q}^*$ which satisfies Eq. (3), from a certain initial value $\boldsymbol{q}_0$ by the following updating rule:

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k - \nabla \boldsymbol{e}(\boldsymbol{q}_k)^{-1} \boldsymbol{e}_k \tag{4}$$

where k is the step of iteration, and $\boldsymbol{e}_k \equiv \boldsymbol{e}(\boldsymbol{q}_k)$. The basic Jacobian matrix $\boldsymbol{J}(\boldsymbol{q})$ is also available instead of $\nabla \boldsymbol{e}$ as

$$\nabla \boldsymbol{e}(\boldsymbol{q}_k) \simeq -\boldsymbol{J}_k \tag{5}$$

where $\boldsymbol{J}_k \equiv \boldsymbol{J}(\boldsymbol{q}_k)$. The above formulation implies the following three assumptions:

1) The number of constraints and the DOF of the robot are the same, i.e., $n = 3\,m$.
2) $\boldsymbol{J}_k$ is always regular.
3) Equation (3) is solvable.

Even if one of them is violated, the iteration will be bankrupted.

In order to discuss the IK solver that will not be bankrupted in unsolvable cases, let us replace the original equation (3) with the following minimization problem (QP):

$$E(\boldsymbol{q}) \equiv \frac{1}{2} \boldsymbol{e}^{\mathrm{T}} \boldsymbol{W}_E \boldsymbol{e} \to \min. \tag{QP}$$

where $\boldsymbol{W}_E = \mathrm{diag}\{w_{E,i}\} (w_{E,i} > 0 \text{ for } \forall i = 1 \sim 3m)$ is the weighting matrix in order to reflect the priority level of each constraint and also to absorb the difference of the physical metric between each constraint. Note that this is not an equivalent replacement but implies that the safety is prioritized over finding the true solution by avoiding an abnormal computation for practical applications.

Here, we point out the following important fact. When a gradient method is applied to the minimization problem, $\nabla E = \boldsymbol{0}^{\mathrm{T}}$ is satisfied at the convergent point. From the definition of $E(\boldsymbol{q})$,

$$\nabla E = \boldsymbol{e}^{\mathrm{T}} \boldsymbol{W}_E \nabla \boldsymbol{e} \simeq -\boldsymbol{e}^{\mathrm{T}} \boldsymbol{W}_E \boldsymbol{J}, \tag{6}$$

and thus, the convergent point is a singular point if it is not the solution of (3). Therefore, the problem of singularity is unavoidable if the original equation is unsolvable.

The NR method is still available to the problem (QP) in a slightly different form from (4). By differentiating (6), we get

$$\begin{aligned} \nabla^2 E &= \nabla \boldsymbol{e}^{\mathrm{T}} \boldsymbol{W}_E \nabla \boldsymbol{e} + \sum_{i=1}^{n} \frac{\partial \nabla \boldsymbol{e}}{\partial q_i} \boldsymbol{W}_E \boldsymbol{e} \\ &\simeq \boldsymbol{J}^{\mathrm{T}} \boldsymbol{W}_E \boldsymbol{J} - \sum_{i=1}^{n} \frac{\partial \boldsymbol{J}}{\partial q_i} \boldsymbol{W}_E \boldsymbol{e} \end{aligned} \tag{7}$$

where (5) is applied. $\dfrac{\partial \boldsymbol{J}}{\partial q_i}$ is called *manipulator Hessian* [21]–[23].[1] The updating rule of the NR method that is based on (7) is given as

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \left(\boldsymbol{J}_k^{\mathrm{T}} \boldsymbol{W}_E \boldsymbol{J}_k - \sum_{i=1}^{n} \frac{\partial \boldsymbol{J}_k}{\partial q_i} \boldsymbol{W}_E \boldsymbol{e}_k \right)^{-1} \boldsymbol{g}_k \tag{8}$$

$$\boldsymbol{g}_k \equiv \boldsymbol{J}_k^{\mathrm{T}} \boldsymbol{W}_E \boldsymbol{e}_k. \tag{9}$$

This method was proposed by Deo and Walker [24]. However, the manipulator Hessian requires a large computational cost, and moreover, it does not guarantee the decrease of E since the coefficient matrix is not necessarily positive definite so that it is not still globally convergent. Now, our aim is not to know the exact curvature of E but to know the descent direction of E. Hence, it is better to use an appropriate positive definite matrix instead of $\nabla^2 E$.

If the latter term of (7) is omitted, it is equivalent with the Gauss–Newton (GN) method [25], where the updating rule is defined as

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + (\boldsymbol{J}_k^{\mathrm{T}} \boldsymbol{W}_E \boldsymbol{J}_k)^{-1} \boldsymbol{g}_k \tag{10}$$

which is equivalent with the usage of the weighted norm-minimizing generalized inverse matrix of $\boldsymbol{J}_k$ instead of $\nabla \boldsymbol{e}(\boldsymbol{q}_k)^{-1}$ in (4). This method was proposed by Anderson and Angeles [26] as a solution of DIK. Although the updating rule (10) is not available if $\boldsymbol{J}_k$ is not full-rank, namely, at the singular points, this is resolved if the MP-inverse matrix is used instead [6]. The problem is rather that the coefficient matrix becomes ill-posed, and the computation becomes unstable in the vicinity of the singular points as pointed out by Nakamura and Hanafusa [10]. In [24], a method to use (8) only near the singular points and (10) in the other area is proposed. It requires to judge if the robot configuration is near the singular points every step of iteration, and thus, is disadvantageous in terms of computational cost.

The LM method [13], [14] resolves the above problem and is defined by the following updating rule:

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \boldsymbol{H}_k^{-1} \boldsymbol{g}_k \tag{11}$$

$$\boldsymbol{H}_k \equiv \boldsymbol{J}_k^{\mathrm{T}} \boldsymbol{W}_E \boldsymbol{J}_k + \boldsymbol{W}_N \tag{12}$$

where $\boldsymbol{W}_N = \mathrm{diag}\{w_{N,i}\} (w_{N,i} > 0 \text{ for } \forall i = 1 \sim n)$ is called the damping factor. The solution of IK using the LM method was proposed by Goldenberg *et al.* [27]. The LM method does not require the computation of the manipulator Hessian and guarantees the decrease of E since $\boldsymbol{H}_k$ is always regular and positive definite, and the increment term of (11) is always oriented to the descent direction. This is the simplest form of Tikhonov's regulation [28], and is interpreted that the following mixed minimization problem is solved in each step of iterations [10]:

$$\frac{1}{2} \boldsymbol{r}_k^{\mathrm{T}} \boldsymbol{W}_E \boldsymbol{r}_k + \frac{1}{2} \Delta \boldsymbol{q}_k^{\mathrm{T}} \boldsymbol{W}_N \Delta \boldsymbol{q}_k \to \min. \tag{13}$$

where $\Delta \boldsymbol{q}_k \equiv \boldsymbol{q}_{k+1} - \boldsymbol{q}_k$, and $\boldsymbol{r}_k \equiv \boldsymbol{e}_k - \boldsymbol{J}_k \Delta \boldsymbol{q}_k$. Hence, $\boldsymbol{q}_k$ converges to a certain value even in redundant cases, taking the minimum deviation.

To summarize this section, the minimization approach with the LM method is more preferable than the other methods from the viewpoint of numerical robustness in unsolvable cases, which necessarily causes a convergence to singular points, even on redundant robots.

[1] This is a third-order tensor. It should not be confused with ordinary Hessian matrix or its determinant.

III. Improved Error Damping Least Square Method in Inverse Kinematics

In the LM method, the choice of the damping factor $\boldsymbol{W}_N$ largely affects the convergence performance. Many methods to choose $\boldsymbol{W}_N$ have been proposed such that they use a constant value [11], the measure of manipulability [10], [29], [30], suppression of the minimum singular value of the coefficient matrix under a threshold [31], [32], suppression of the condition number of the coefficient matrix under a threshold [33], [34], explicit limitation of the increment of the configuration vector [35]–[37], the adjoint vector of the residual of higher priority constraints [38], and so forth. They were considered to keep better local numerical conditions from the viewpoint of DIK, and the discussion from the viewpoint of the convergence performance of iteration was almost out of scope. On the other hand, Chan and Lawrence [20] proposed the error damped least-squares method that defines the damping factor as

$$\boldsymbol{W}_N = \lambda E_k \mathbf{1} \tag{14}$$

where λ is a constant coefficient and does not have a significant effect; $\lambda = 1$ is acceptable, $E_k \equiv E(\boldsymbol{q}_k)$, and $\mathbf{1}$ is the $n \times n$ identity matrix. If the original equation (3) is solvable, E_k quadratically converges to 0 while $\boldsymbol{q}_k$ converges to the solution, and thus, the iteration is expected to be superlinearly convergent [39]. However, if the solution is near one of the singular points, the coefficient matrix will become numerically ill-posed as it approaches to the solution.

We propose to modify it as

$$\boldsymbol{W}_N = E_k \mathbf{1} + \overline{\boldsymbol{W}}_N \tag{15}$$

where $\overline{\boldsymbol{W}}_N = \mathrm{diag}\{\overline{w}_{N,i}\}$, and $(\overline{w}_{N,i} > 0$ for $\forall i = 1 \sim n)$ are small biasing values. Suppose all $\overline{w}_{N,i}$ are equally $\overline{w}_N$, namely, $\overline{\boldsymbol{W}}_N = \overline{w}_N \mathbf{1}$ for simplicity, and the singular value decomposition is conducted to $\boldsymbol{W}_E^{1/2} \boldsymbol{J}_k$ as

$$\boldsymbol{W}_E^{1/2} \boldsymbol{J}_k = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^{\mathrm{T}} \tag{16}$$

where $\boldsymbol{W}_E^{1/2}$ means $\mathrm{diag}\{\sqrt{w_{N,i}}\}$, $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthonormal matrices, and $\boldsymbol{\Sigma} = \mathrm{diag}\{\sigma_i\}$ is a square matrix in which the singular values are diagonally assigned in descent order, i.e., $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$. Since

$$\begin{aligned} \boldsymbol{H}_k &= \boldsymbol{V} \boldsymbol{\Sigma} \boldsymbol{U}^{\mathrm{T}} \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^{\mathrm{T}} + (E_k + \overline{w}_N) \mathbf{1} \\ &= \boldsymbol{V} \left\{ \boldsymbol{\Sigma}^2 + (E_k + \overline{w}_N) \mathbf{1} \right\} \boldsymbol{V}^{\mathrm{T}} \end{aligned} \tag{17}$$

the condition number of $\boldsymbol{H}_k$ κ is

$$\kappa = \frac{\sigma_1^2 + E_k + \overline{w}_N}{\sigma_n^2 + E_k + \overline{w}_N}. \tag{18}$$

It states the following facts.

1) If Eq. (3) is solvable and the solution is far from the singular points, $\boldsymbol{W}_N$ quadratically converges to $\overline{\boldsymbol{W}}_N$. If $\overline{\boldsymbol{W}}_N$ is sufficiently small, the iteration is superlinearly convergent as well as the Chan *et al*'s method.
2) If Eq. (3) is solvable and the solution is near one of the singular points, κ converges to $\dfrac{\sigma_1^2 + \overline{w}_N}{\overline{w}_N}$. Therefore, $\overline{w}_N$ is necessary to prevent $\boldsymbol{H}_k$ from degeneracy.
3) If Eq. (3) is unsolvable, κ becomes close to 1 as E_k increases. It suggests an interesting fact that the larger the norm of $\boldsymbol{e}_k$ during the iteration, the more stable the computation. This property is also expected from an equation $\|\boldsymbol{H}_k^{-1} \boldsymbol{g}_k\| \simeq \dfrac{1}{\|\boldsymbol{e}_k\|}$.

Although there still remains room for discussion about how to choose $\overline{w}_N$, it only matters in the aforementioned case 2 where the solution is near one of the singular points. As stated in the case 1, a smaller value shows a better convergence. How small value can be accepted depends on σ_1. Although it is not easy to estimate σ_1, we have an empirical knowledge that the computation sufficiently robustly converges with $\overline{w}_N \simeq 1.0 \times 10^{-1} l^2 \sim 1.0 \times 10^{-3} l^2$ at least for the characteristic length, i.e., the length of a typical link, $l = 0.1 \sim 100$ [m].

In order to guarantee the global convergence of the LM method in general, it is known that a linear search algorithm such as the Moré–Thuente method [40] should be combined. In the following section, however, it will be empirically shown that the proposed method has sufficient convergence performance without a linear search.

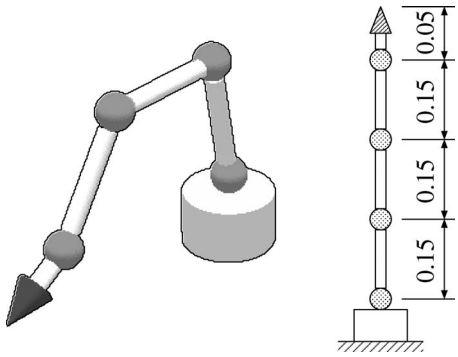

Fig. 1. Kinematics model of the tested manipulator comprising five links and four spherical joints. The lengths of the three links are all 0.15 [m], while the end-effector's length is 0.05 [m].

TABLE I
Parameters for LM(NH), LM(MWM), and LM(Chi)

method	max.	threshold	recital
LM(NH)	0.1	2.0×10^{-4}	manipulability
LH(MWM)	0.001	1.0×10^{-2}	condition number
LH(Chi)	0.1	1.0×10^{-2}	minimum singular value

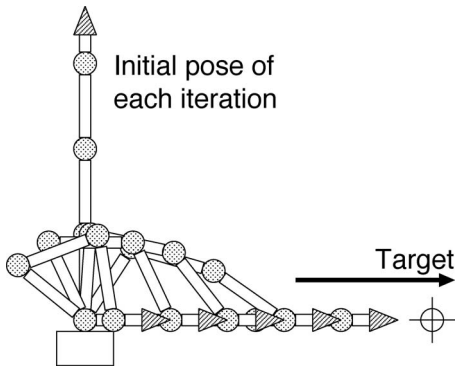

Fig. 2. In the first test, the target position of the endpoint is moved from 0.1 to 1.0 [m] step-by-step.

IV. Evaluations

The proposed method was evaluated in terms of the convergence performance and the computation time with a redundant manipulator model as shown in Fig. 1. This model has 12 DOF, in which five links are serially connected by four spherical joints. The link lengths are all 0.15[m], and the distance from the center of rotation of the effector's joint to the endpoint is 0.05 [m]. The effector position means

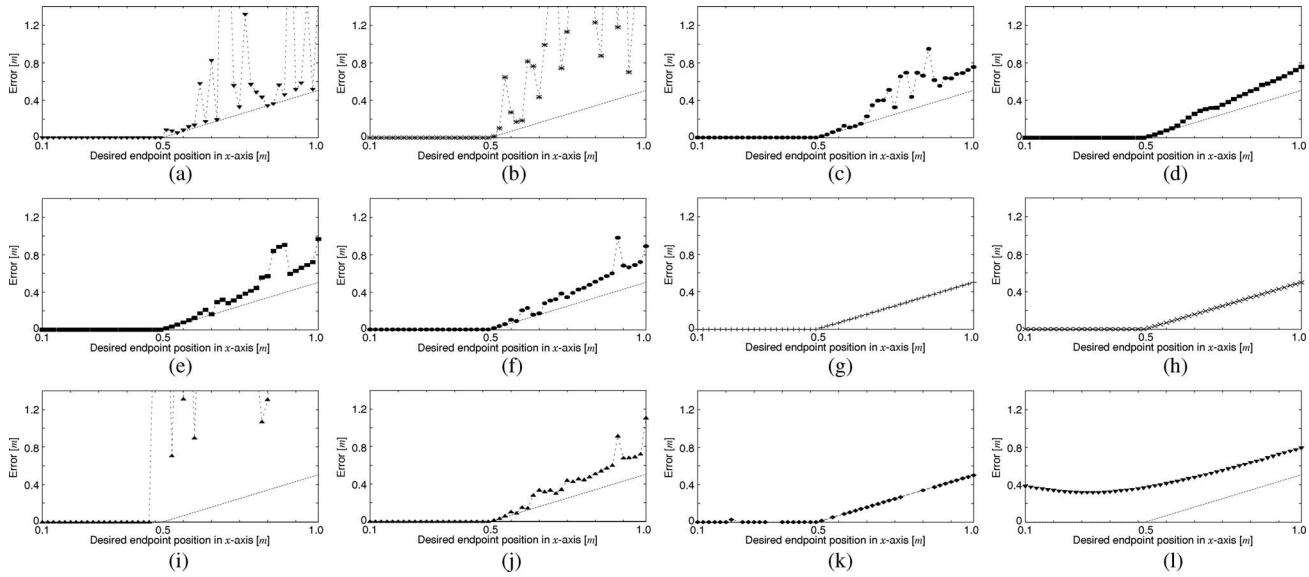

Fig. 3. Error comparison between SD, GN, LM(λ =const.), LM(NH), LM(CL), LM(proposed), LM(MWM), LM(Chi), VM, and VM(MT). All methods except for LM(CL) and LM(proposed) failed to get the minimum solution particularly in unsolvable range. VM made the results diverged in some cases. All results of VM(MT) were captured by local minima. (a) SD method, (b) GN method, (c) LM($\lambda = 0.001$), (d) LM($\lambda = 0.01$), (e) LM($\lambda = 0.1$), (f) LM(NH), (g) LM(CL), (h) LM(proposed), (i) LM(MWM), (j) LM(Chi), (k) VM, and (l) VM(MT).

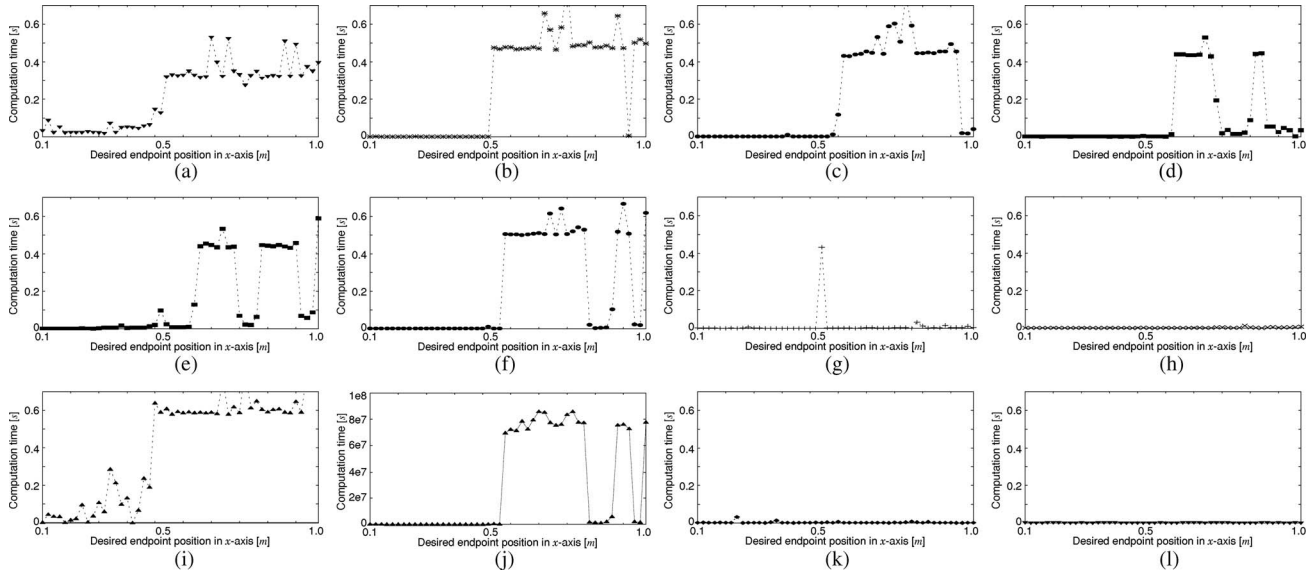

Fig. 4. Computation time comparison between SD, GN, LM(λ =const.), LM(NH), LM(CL), LM(proposed), LM(MWM), LM(Chi), VM, and VM(MT). Except for the proposed method, all methods failed to terminate iterations in some cases. Particularly, LM(Chi) consumed incomparably long time. Although the result of VM(MT) showed fast convergence, they were all captured by local minima. See Fig. 3. (a) SD method, (b) GN method, (c) LM($\lambda = 0.001$), (d) LM($\lambda = 0.01$), (e) LM($\lambda = 0.1$), (f) LM(NH), (g) LM(CL), (h) LM(proposed), (i) LM(MWM), (j) LM(Chi), (k) VM, and (l) VM(MT).

the endpoint position hereafter. The following 12 methods were also evaluated for comparison.

1) SD: the steepest descent method.
2) GN: the GN method with a weighted MP-inverse matrix.
3) LM($\lambda = 0.001, 0.01, 0.1$): the LM method with a constant λ for $\boldsymbol{W}_N = \lambda \mathbf{1}$ (three sets).
4) LM(NH): the LM method with Nakamura and Hanafusa's method [10].
5) LM(MWM): the LM method with Mayorga–Wong–Milano's method [34].
6) LM(Chi): the LM method with Chiaverini's method [31].
7) LM(CL): the LM method with Chan and Lawrence's method [20].
8) LM(proposed): the proposed version of the LM method.
9) VM: variable metric method based on the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula without a linear search.
10) VM(MT): the variable metric method based on the BFGS formula with More-Thuente method [40] for a linear search.

The updating rule for SD was

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k - \frac{E_k}{\boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{g}_k} \boldsymbol{g}_k \tag{19}$$

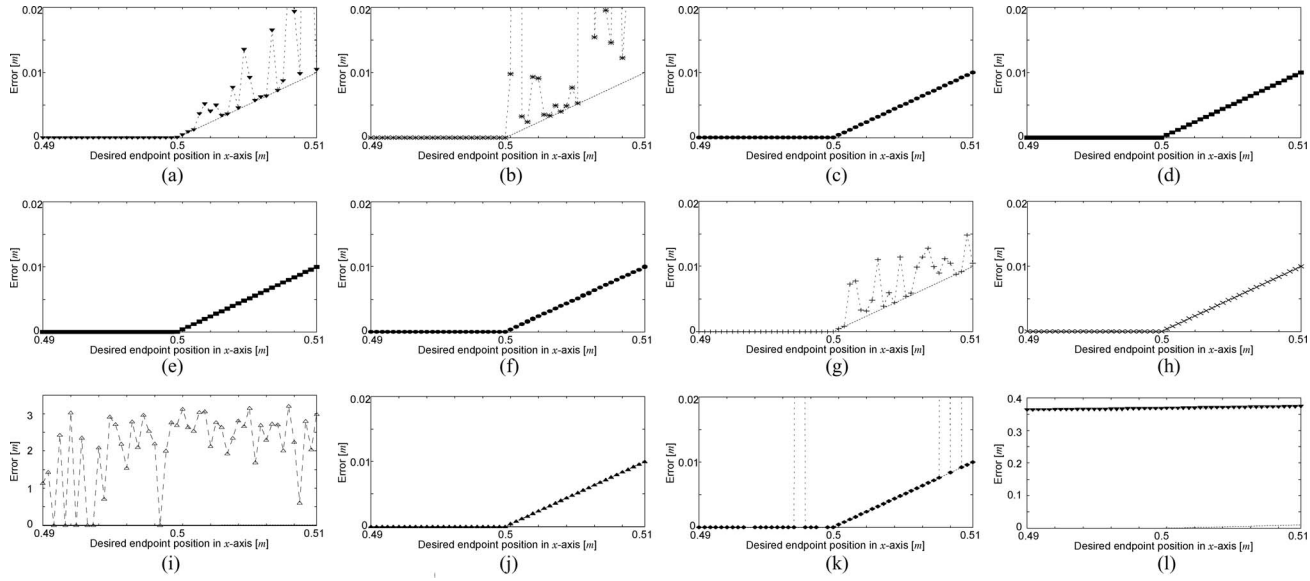

Fig. 5. Error comparison between SD, GN, LM(λ =const.), LM(NH), LM(CL), LM(proposed), LM(MWM), LM(Chi), VM, and VM(MT) near the boundary of the solvable range. Note that the ranges of the vertical axes of (i) LM(MWM) and (l) VM(MT) are different from the other graphs as well as from the corresponding parts in Fig. 3 due to their incomparably lower accuracy. While LM(CL) showed unstable behavior, the proposed method succeeded in all cases. Although LM(Chi) showed successful results, it consumed long time for computations. See Fig. 6. (a) SD method, (b) GN method, (c) LM($\lambda = 0.001$), (d) LM($\lambda = 0.01$), (e) LM($\lambda = 0.1$), (f) LM(NH), (g) LM(CL), (h) LM(proposed), (i) LM(MWM), (j) LM(Chi), (k) VM, and (l) VM(MT).

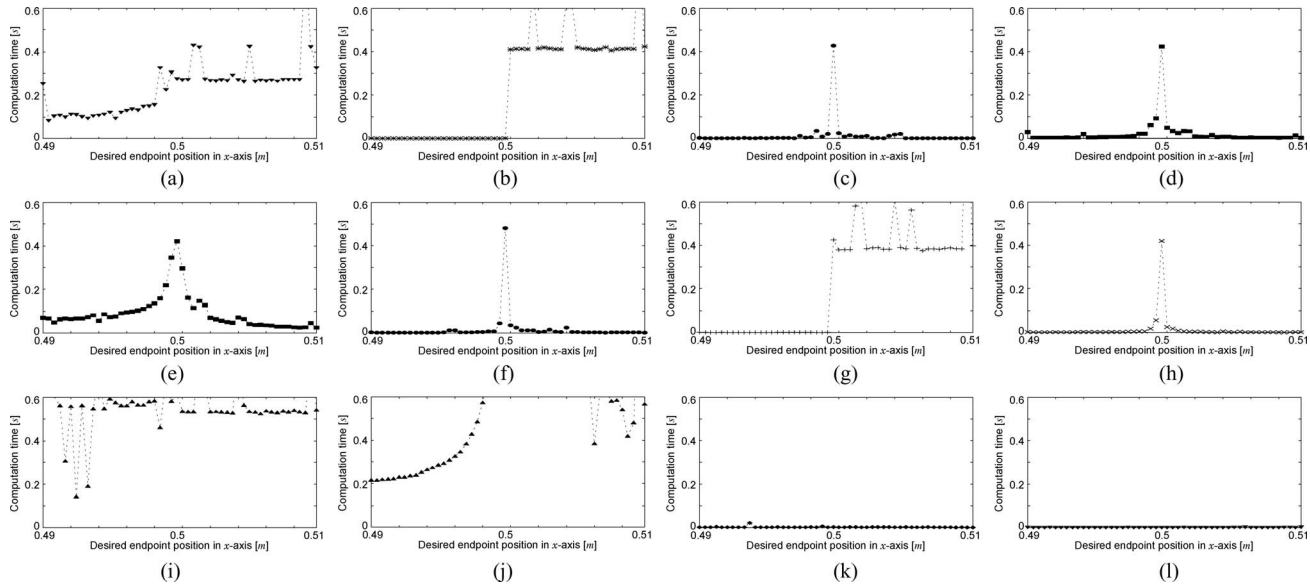

Fig. 6. Computation time comparison between SD, GN, LM(λ =const.), LM(NH), LM(CL), LM(proposed), LM(MWM), LM(Chi), VM, and VM(MT) near the boundary of the solvable range. Although the proposed method took long time exactly at the boundary, the result was correct. In addition, any other methods did not show better performance than the proposed method. (a) SD method, (b) GN method, (c) LM($\lambda = 0.001$), (d) LM($\lambda = 0.01$), (e) LM($\lambda = 0.1$), (f) LM(NH), (g) LM(CL), (h) LM(proposed), (i) LM(MWM), (j) LM(Chi), (k) VM, and (l) VM(MT).

which was based on the estimated minimizer in accordance with the approximated $E(\boldsymbol{q})$ by a quadratic function around $\boldsymbol{q}_k$.

MP-inverse solution was implemented with *LQ* decomposition, which was faster than the singular value decomposition. See Appendix B, for details.

LM(NH), LM(MWM), and LM(Chi) are implemented according to each paper. All these methods need the maximum damping factor and a threshold as design parameters, the choice of which is not trivial. We designed them as shown in Table I through some trials and errors for the evaluations in this section; it was hard to find another set of those values that presented better results in terms of the computation time and accuracy at least in the tests.

$\overline{w}_N = 1.0 \times 10^{-3}$ was chosen for the proposed method. For all iterations, $\boldsymbol{W}_E = \boldsymbol{1}$ was adopted, and the initial value was reset for $\boldsymbol{q}_0 = [0\ 0\ 0\ 0\ 0\ 0]^{\mathrm{T}}$ in every test—note that it is a singular point. The iteration was terminated when one of the following conditions were satisfied.

a) All components of $\Delta\boldsymbol{q}_k$ are less than $\epsilon = 1.0 \times 10^{-12}$.

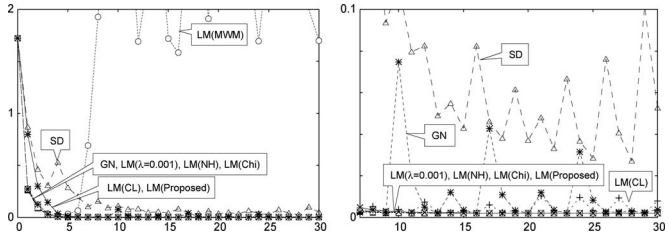

Fig. 7 History of weighted-squared residual in iterations (Left: From 0 to 300 step. Right: From 7 to 300 step).

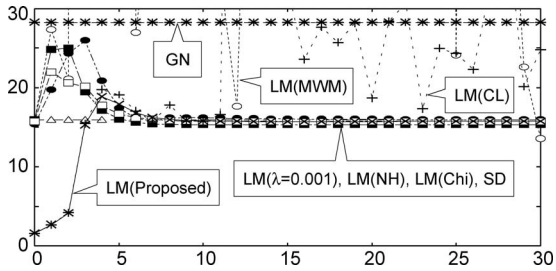

Fig. 8 History of condition number in iterations.

b) The deviation of $\|\boldsymbol{e}_k\|$ from the previous step was less than $\delta = 1.0 \times 10^{-12}$.

c) The number of iteration steps was over $10\,000$.

In the first test, IK was tried to be solved, in which the desired effector position and orientation were set for

$$ {}^{d}\boldsymbol{p} = \begin{bmatrix} x_d \\ 0 \\ 0 \end{bmatrix}, \quad {}^{d}\boldsymbol{R} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \tag{20} $$

as depicted in Fig. 2. Fifty cases where x_d was set for the value from 0.1 to 1.0 equally divided by 50 were examined. Since the maximum extension length of the manipulator was 0.5[m], the problem became unsolvable when $x_d \geq 0.5$. Figs. 3 and 4 show the norms of residual at termination and the computation times, respectively, for the aforementioned 12 methods. The dotted line in Fig. 3 means the minimum norm of residual from the desired position and orientation in theory. Namely, if the point for the norm of residual at termination is on this dotted line, the computation is regarded to be successful. Only LM(CL) and LM(proposed) succeeded in all the cases, while the convergence performance was degraded near $x_d = 0.5$ in LM(CL) as shown in Fig. 4(g). SD, GN, LM($\lambda = 0.001, 0.01, 0.1$), LM(NH), LM(MWM), and LM(Chi) frequently failed in unsolvable cases. Although the success rate of VM was comparatively high in these cases, it diverged in some cases, and more seriously, it was not clear under what condition it diverged. All the results of VM(MT) were captured at local minima and never reached the minimizer.

The second test was to solve IK for fifty cases in which x_d was set for the value from 0.49 to 0.51 equally divided by 50 in order to investigate the behavior around $x_d \simeq 0.5$, which is the limitation of solvability. Figs. 5 and 6 show the norms of residual at termination and the computation times, respectively. LM($\lambda = 0.001, 0.01, 0.1$), LM(NH), LM(Chi), and LM(proposed) succeeded in all the cases. Note that any of those methods cannot avoid the degradation of convergence performance exactly at $x_d = 0.5$, as shown in Fig. 6(c)–(f) and (h). On the other hand, LM(CL) frequently failed in this test as predicted in the previous section. VM still randomly failed, and VM(MT) was captured at local minima in all the cases. Fig. 7 shows a series of changes of the evaluation function E_k in the iteration process of the tested methods

TABLE II
RESULT OF RANDOM TEST $\times 1000$

method	rate of success	ave. computation time[μs]
LM(proposed)	1000/1000	2227
LM(CL)	935/1000	2267
LM($\lambda = 0.1$)	277/1000	19265
LM($\lambda = 0.01$)	276/1000	4161
LM($\lambda = 0.001$)	255/1000	1819
LM(NH)	240/1000	2197
GN	211/1000	417
SD	211/1000	52431
LM(MWM)	191/1000	99305
VM	1/1000	1776
VM(MT)	0/1000	–

The configuration was initialized for the home position every time. The target position and orientation of the effector were set within 1.2×1.2×1.2[m] cubic area.

for $x_d = 0.502$, which is near the singular point, where the results of LM($\lambda = 0.01, 0.1$) and VM are not plotted. It matches with an expectation that SD is linearly convergent, LM(CL) and LM(proposed) are superlinearly convergent, and the others are quadratically convergent. Although LM(CL) on the left side of the same figure seems to converge, the right side of the same figure, in which the changes of E_k after the seventh step of the iteration are plotted, exposes that the method does not converge in the final stage of computation. Fig. 8 shows the change of the condition number κ in the iteration process of each method. One can find that it converges to about 16 in LM(proposed) as well as the other successful methods, while it fluctuates from about ten to hundreds in LM(CL) so that the convergence performance is degraded.

In the last test, IK was examined 1000 times for the desired positions and orientations that were randomly sampled in a $1.2 \times 1.2 \times 1.2[\mathrm{m}^3]$ cubic area whose center was at the center of rotation of the first joint. The rates of success and the average computation times at successes are listed in Table II in the order of higher rate of success. LM(Chi) was excluded from the comparison methods since its average computation time was by far the longest in all the results. If the gap between the norm of residual by each method and the minimum value of them by the 12 methods was less than 1.0×10^{-6}, it was regarded to be a success, since it was unable to know the true solution of each case in the test. It is noteworthy that only the proposed method marked 100% success rate. The success rates of the other methods except for LM(CL) were all under 50%. Particularly, VM(MT) failed in all the cases. Despite the success rate of VM was comparatively high in the previous test, it succeeded in only one case in this test. It means that the variable metric method in general is not reliable.

About the computation time, it stands to reason that the GN method was the fastest when succeeded. However, the GN method was five times as fast as the proposed method—not remarkably faster than the proposed. The proposed method was also competitive with the other comparatively successful methods which include LM(CL), LM($\lambda = 0.001$), and LM(NH). Namely, the proposed method does not sacrifice the computation time instead of high rate of success.

One can conclude that the proposed method is the most practical among the 12 methods.

V. APPLICATION

The proposed IK solution was implemented on a motion editing software for humanoid robots as an application. It utilizes an intuitive pin-and-drag interface proposed by Yamane and Nakamura [41]. Fig. 9 shows some screenshots of it. A difficulty of such a software is that the relationship between the number of constraints and the DOF of the

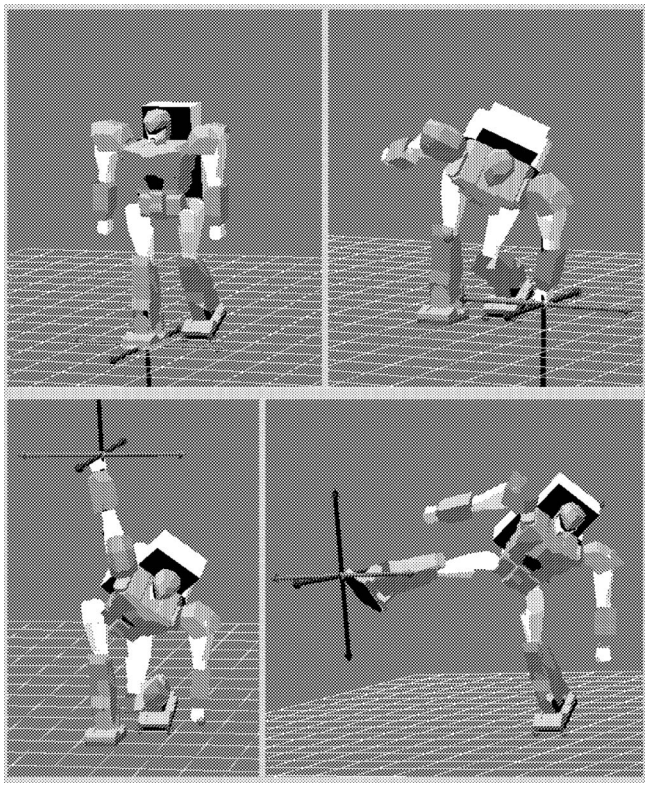

Fig. 9 Snapshots of a humanoid robot motion editor.

robot frequently varies. Particularly, in the overconstraint cases where the number of constraints is more than the robot DOF, the problem will necessarily become unsolvable. Hence, a solvability-unconcerned IK solver such as the proposed method is crucial.

The number of constraints in each scene in Fig. 9 were 12 (position and orientation of the right foot and positions of two points of the left toe) for the top left, 15 (position of the left hand in addition) for the top right, 18 (position of the right hand in addition) for the bottom left, and 18 (position and orientation of the right foot, positions of two points of the left toe and position and orientation of the trunk) for the bottom right, respectively. In the top two situations, the robot converged to a certain posture that satisfied the specified constraints in spite of redundancy. In the bottom-left situation, the goal position of the right hand was located out of reach. Namely, this is a result of an unsolvable case, and the robot posture in the figure is a singular point. During the editing process of the bottom-right posture, the goal positions of the trunk and the right foot were repeatedly drawn away from each other so that the robot frequently fall into singular points. In spite of that, the robot posture was stably edited; it was neither captured at the singular points nor oscillated.

In the software developed by Yamane *et al.*, the robot configuration tracks the goal by a resolved motion rate from the velocity of the link of interest which compensates the error of position and orientation. Thus, the convergence performance depends on both the potential of DIK solver and the velocity of the link of interest. On the other hand, in the author's software, iterative computations are conducted in every step, which achieves a sufficiently interactive operation for a 26-DOF anthropomorphic robot model [42]. The weights on the pinned points and the dragged point were 1.0 and 0.1, respectively. The prioritized DIK that is based on the null space [43] was not utilized. In order to cope with the limitation of the joint motion range, the software adopts a simple technique that truncates the joint displacement value by its limit.

VI. Conclusion

A numerical IK solution which is based on the LM method was proposed. It is robust against the problems of singularity, redundancy, and even solvability. A simple technique that defines the damping factor as the squared norm of residual with a slight bias does not require a heavy computation of the measure of manipulability, the minimum singular value of the coefficient matrix, and so forth. In spite of that, it is superior to the conventional methods in terms of the rate of success. In fact, it has never failed in the author's experiences. It is expected that it superlinearly converges in theory, while it was also empirically evaluated to be sufficiently fast in practice. Although there is room for discussion about how to choose the biasing value, it marked robust convergence performance when it was set for $1.0 \times 10^{-1} \sim 1.0 \times 10^{-3}$ times as large value as the characteristic length of the robot, as noted in Section III. This fact provides an advantage to the conventional methods that do not have a clear policy to decide the design parameters and are sensitive to those parameters.

The proposed method is expected to be a fundamental technique to build software tools, particularly, remote operation systems and motion editors for a large-scale robot such as humanoid robots, which enables easy and safe handling of a variety of motion for novice users.

Appendix A

Conversion from an Orientation Matrix to the Equivalent Angle-Axis Vector

Let us define the following vector with respect to an orientation matrix $\boldsymbol{R} = \{r_{ij}\} \in SO(3)(i = 1 \sim 3, j = 1 \sim 3)$:

$$\boldsymbol{l} \equiv \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}. \tag{21}$$

If $\boldsymbol{R}$ is not a diagonal matrix, $\boldsymbol{l}$ is not the zero vector, and an angle-axis vector $\boldsymbol{a}$ which is equivalent to $\boldsymbol{R}$ is computed as

$$\boldsymbol{a}(\boldsymbol{R}) \equiv \frac{\operatorname{atan2}\left(\|\boldsymbol{l}\|, r_{11} + r_{22} + r_{33} - 1\right)}{\|\boldsymbol{l}\|} \boldsymbol{l}. \tag{22}$$

If $\boldsymbol{R}$ is a diagonal matrix, there are four cases where $(r_{11}, r_{22}, r_{33}) = (1, 1, 1)$, $(1, -1, -1)$, $(-1, 1, -1)$ or $(-1, -1, 1)$. If two of (r_{11}, r_{22}, r_{33}) are -1, $\boldsymbol{a}$ is defined as

$$\boldsymbol{a} \equiv \frac{\pi}{2} \begin{bmatrix} r_{11} + 1 \\ r_{22} + 1 \\ r_{33} + 1 \end{bmatrix}. \tag{23}$$

if $(r_{11}, r_{22}, r_{33}) = (1, 1, 1)$, then $\boldsymbol{a} \equiv \boldsymbol{0}$.

Luh *et al.* [44] proposed an idea to use $\boldsymbol{l}$ in (21) rather than $\boldsymbol{a}$ directly for the orientation error evaluation, which wrongly copes with the above four cases that $\boldsymbol{R}$ is diagonal. The method which is described in this section solves the problem.

Appendix B

Moore–Penrose's-Inverse Solution with LQ Decomposition

Let us decompose the basic Jacobian matrix $\boldsymbol{J}_k$ into the following multiplication form of two matrices $\boldsymbol{L}$ and $\boldsymbol{Q}$:

$$\boldsymbol{J}_k = \boldsymbol{L}\boldsymbol{Q}, \tag{24}$$

where $\boldsymbol{L}$ is a column-full-rank lower triangle matrix, and $\boldsymbol{Q}$ is an orthonormal matrix. This operation is called LQ decomposition, which is available for an arbitrary $\boldsymbol{J}_k$.

MP-inverse matrix of $\boldsymbol{J}_k$, $\boldsymbol{J}_k^*$, is computed from $\boldsymbol{L}$ and $\boldsymbol{Q}$ as

$$\boldsymbol{J}_k^{\#} = \boldsymbol{Q}^{\mathrm{T}}(\boldsymbol{L}^{\mathrm{T}}\boldsymbol{W}_E\boldsymbol{L})^{-1}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{W}_E. \tag{25}$$

MP-inverse solution can be efficiently implemented in such a way that an equation $\boldsymbol{L}^{\mathrm{T}}\boldsymbol{W}_E\boldsymbol{L}\boldsymbol{v} = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{W}_E\boldsymbol{e}_k$ is solved first for $\boldsymbol{v}$ by the Gauss elimination method, and then, $\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{v}$.

REFERENCES

[1] D. L. Pieper, "The kinematics of manipulators under computer control," Ph.D. dissertation, Stanford Univ., Stanford, CA, 1968.

[2] M. Raghavan and B. Roth, "Kinematic analysis of the 6R manipulator of general geometry," in *Proc. 5th Int. Symp. Robot. Res.*, 1991, pp. 263–269.

[3] M. Raghavan and B. Roth, "Solving polynomial systems for the kinematic analysis and synthesis of mechanisms and robot manipulators," *Trans. ASME, Spec. 50th Anniversary Design Issue*, vol. 117, pp. 71–79, 1995.

[4] S. Sasaki, "Feasibility studies of kinematic problems in the case of a class of redundant manipulators," *Robotica*, vol. 13, pp. 233–241, 1995.

[5] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *Int. J. Robot. Autom.*, vol. RA-3, no. 1, pp. 43–53, 1987.

[6] D. E. Whitney, "The mathematics of coordinated control of prosthetic arms and manipulators," *Trans. ASME, J. Dyn. Syst., Meas., Control*, vol. 94, no. 4, pp. 303–309, 1972.

[7] D. E. Orin and W. W. Schrader, "Efficient computation of the Jacobian for robot manipulators," *Int. J. Robot. Res.*, vol. 3, no. 4, pp. 66–75, 1984.

[8] M. Uchiyama, "A study of computer control of motion of a mechanical arm (1st report, calculation of coordinative motion considering singular points)," *Trans. Jpn. Soc. Mech. Eng., Series C*, vol. 45, no. 391, pp. 314–322, 1979.

[9] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1985, pp. 722–728.

[10] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *Robot. Manuf. Autom. PED*, vol. 15, pp. 193–204, 1985.

[11] C. W. Wampler, II, " Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-16, no. 1, pp. 93–101, Jan. 1986.

[12] D. N. Nenchev, Y. Tsumaki, and M. Uchiyama, "Singurality-consistent behavior of telerobots: Theory and experiments," *Int. J. Robot. Res.*, vol. 17, no. 2, pp. 138–152, 1998.

[13] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Q. Appl. Math.*, vol. 2, pp. 4164–4168, 1944.

[14] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Ind. Appl. Math.*, vol. 11, no. 2, pp. 431–441, 1963.

[15] Y. Nakamura and K. Yamane, "Dynamics computation of structure-varying kinematic chains and its application to human figures," *IEEE Trans. Robot. Autom.*, vol. 16, no. 2, pp. 124–134, Apr. 2000.

[16] W. A. Wolovich and H. Elliot, "A computational technique for inverse kinematics," in *Proc. 23rd IEEE Conf. Decis. Control*, 1984, pp. 1359–1363.

[17] A. Balestrino, G. de Maria, and L. Sciavicco, "Robust control of robotic manipulators," in *Proc. 9th IFAC World Congr.*, 1984, vol. 5, pp. 2435–2440.

[18] L. T. Wang and C. C. Chen, "A combined optimization method for solving the inverse kinematics problem of mechanical manipulators," *IEEE Trans. Robot. Autom.*, vol. 7, no. 4, pp. 489–499, Nov. 1991.

[19] J. Zhao and N. I. Badler, "Inverse kinematics positioning using nonlinear programming for highly articulated figures," *ACM Trans. Graph.*, vol. 13, no. 4, pp. 313–336, 1994.

[20] S. K. Chan and P. D. Lawrence, "General inverse kinematics with the error damped pseudoinverse," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1988, pp. 834–839.

[21] E. D. Pohl and H. Lipkin, "A new method of robotic motion control near singularities," in *Proc. 5th IEEE Int. Conf. Adv. Robot.*, 1991, pp. 405–410.

[22] K. A. O'Neil, Y. Chen, and J. Seng, "Removing singularities of resolved motion rate control of mechanisms, including self-motion," *IEEE Trans. Robot. Autom.*, vol. 13, no. 5, pp. 741–751, Oct. 1997.

[23] A. Hourtash, "The kinematic hessian and higher derivatives," in *Proc. 2005 IEEE Int. Symp. Comput. Intell. Robot. Autom.*, 2005, pp. 169–174.

[24] A. S. Deo and I. D. Walker, "Adaptive non-linear least squares for inverse kinematics," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1993, pp. 186–193.

[25] J. E. Dennis, Jr. and R. B. Schnabel, "*Numerical Methods for Unconstrained Optimization and Nonlinear Equations*," Soc. Ind. Appl. Math., 1993.

[26] K. Anderson and J. Angeles, "Kinematic inversion of robotic manipulators in the presence of redundancies," *Int. J. Robot. Res.*, vol. 8, no. 6, pp. 80–97, 1989.

[27] A. A. Goldenberg, B. Benhabib, and R. G. Fenton, "A complete generalized solution to the inverse kinematics of robots," *IEEE J. Robot. Autom.*, vol. RA-1, no. 1, pp. 14–20, Mar. 1985.

[28] G. H. Golub, P. C. Hansen, and D. P. O'Leary, "Tikhonov regularization and total least squares," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 1, pp. 185–194, 1999.

[29] T. Yoshikawa, "Manipulability of robotic mechanisms," *Int. J. Robot. Res.*, vol. 4, no. 2, pp. 3–9, 1985.

[30] L. Kelmar and P. Khosla, "Automatic generation of kinematics for a reconfigurable modular manipulator system," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1988, pp. 663–668.

[31] S. Chiaverini, "Estimate of the two smallest singular values of the Jacobian matrix: Applications to damped least-squares inverse kinematics," *J. Robot. Syst.*, vol. 10, pp. 991–1008, 1988.

[32] S. Chiaverini, B. Siciliano, and O. Egeland, "Review of damped least-squares inverse kinematics with experiments on an industrial robot manipulator," *IEEE Trans. Control Syst. Technol.*, vol. 2, no. 2, pp. 123–134, Jun. 1994.

[33] A. A. Maciejewski and C. A. Klein, "Numerical filtering for the operation of robotic manipulators through kinematically singular configurations," *J. Robot. Syst.*, vol. 5, no. 6, pp. 527–552, 1988.

[34] R. V. Mayorga, A. K. C. Wong, and N. Milano, "A fast procedure for manipulator inverse kinematics evaluation and pseudoinverse robustness," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 4, pp. 790–798, Jul./Aug. 1992.

[35] A. A. Maciejewski and C. A. Klein, "The singular value decomposition: Computation and applications to robotics," *Int. J. Robot. Res.*, vol. 8, pp. 63–79, 1989.

[36] A. S. Deo and I. D. Walker, "Robot subtask performance with singularity robustness using optimal damped least squares," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1992, pp. 434–441.

[37] S. R. Buss and J. Kim, "Selectively damped least squares for inverse kinematics," *J. Graph. Tools*, vol. 10, no. 3, pp. 37–49, 2005.

[38] Y. Choi, "Singularity-robust inverse kinematics using Lagrange multiplier for redundant manipulators," *Trans. ASME, J. Dyn. Syst., Meas., Control*, vol. 130, no. 5, pp. 051009-1–051009-7, 2008.

[39] N. Yamashita and M. Fukushima, "On the rate of convergence of the Levenberg–Marquardt method," *Computing*, vol. 15, pp. 237–249, 2001.

[40] J. J. Moré and D. J. Thuente, "Line search algorithms with guaranteed sufficient decrease," *ACM Trans. Math. Softw.*, vol. 20, no. 3, pp. 286–307, 1994.

[41] K. Yamane and Y. Nakamura, "Natural motion animation through constraining and deconstraining at will," *IEEE Trans. Vis. Comput. Graph.*, vol. 9, no. 3, pp. 352–360, Jul./Sep. 2003.

[42] T. Sugihara, K. Yamamoto, and Y. Nakamura, "Hardware design of high performance miniature anthropomorphic robots," *Robot. Auton. Syst.*, vol. 56, pp. 82–94, 2007.

[43] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. Robot. Res.*, vol. 4, no. 3, pp. 109–117, 1985.

[44] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "Resolved acceleration control of mechanical manipulators," *IEEE Trans. Automat. Control*, vol. AC-25, no. 3, pp. 468–474, Jun. 1980.