2014 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS 2014)
September 14-18, 2014, Chicago, IL, USA

Robust Solution of Prioritized Inverse Kinematics Based On Hestenes-Powell Multiplier Method

Tomomichi Sugihara

***Abstract*— A robust numerical solution of the prioritized inverse kinematics is proposed. It is based on the augmented Lagrangian function and Hestenes-Powell's multiplier method. It originates the weighted inverse kinematics and only requires a small modification including an accumulation of the error of the high-priority constraint at each step of iteration and an estimation of Lagrange's multiplier. Hence, it is preferable to the conventional method, which is accompanied with an explicit complex computation of the kernel space, from the viewpoint of both the implementation cost and the computation cost per step. A drawback is that the proposed method becomes slow in some situations since Lagrange's multiplier linearly converges, while the joint displacements superlinearly converge. In some unlucky situations, it requires more computation cost in total than the conventional method. However, the proposed method is robust even in cases where the high-priority constraint is unsatisfiable. In fact, the proposed method solely succeeded in all the tested cases including unsolvable ones.**

I. Introduction

Inverse kinematics is a basic computation which maps required movements of a robot in the task space, the position and attitude of effectors in particular, to joint displacements. The required movements might be posed regardless of whether it is achievable or not by the real robot. It means that in some cases the problem becomes unsolvable, namely, any combinations of joint displacements cannot satisfy the requirements on the motion. The prioritized inverse kinematics [1] is an idea to resolve this problem by classifying the requirements into ones that have to be strictly satisfied and the other ones that do not and by minimizing the residual error of the latter as non-binding tasks based on the purpose and physical constraints. The former is called the high-priority constraint and the latter the low-priority constraint.

Inverse kinematics is a root-finding of complex nonlinear simultaneous equations. It is impossible in general to find the solution analytically except for particular mechanisms [2]–[5]. In many situations, one relies on a numerical solution, which is based on the differential inverse kinematics. The prioritized inverse kinematics has also been discussed mainly in the context of the differential inverse kinematics [1], [6]. As well-known in the field of robot kinematics, the infinitesimal displacements of joints are linearly mapped to the infinitesimal displacements of effectors [7]. The idea is to minimize the error of the low-priority constraint in the kernel space of the above map from the joint to the effector, namely, a non-empty subset of joint displacements which does not yield any movement of the effector, and to satisfy the high-priority constraint in the image space.

This approach has the following three problems.

I) The solvability of the problem, namely, the achievability of the high-priority constraint is not guaranteed as well as the ordinary inverse kinematics since it might be posed without any concern with the real robot and situation. It is more difficult to judge if the problem is solvable or not in advance. As Sugihara [8] pointed out, the least-square-error solution necessarily approaches to the singular point in unsolvable cases, so that the computation will be bankrupted in the worst case. It is unacceptable over a situation where the strictness of the computation is lost in real operations of robots.

II) To get the least-square-error solution utilizing the kernel space is a heavy computation in which multiplications of matrices and inverse matrices have to be directly handled.

III) The computation originally aims to solve nonlinear simultaneous equations, in other words, to find zeros of a manifold. It is not essential to discuss the image space and the kernel space of the tangential hyperplane of the manifold strictly in each step of iteration. The complicated computation does not pay off.

This paper proposes a novel numerical solution of the prioritized inverse kinematics based on Hestenes-Powell multiplier method [9], [10], which is a method for quadratic programmings with equality constraints. The method simultaneously finds the joint displacements and its adjoint variables, which is also known as Lagrange's multiplier, in an iterative way. It can be implemented by a slight modification of an inverse kinematics solver based on Levenberg-Marquardt method (LM method) proposed by Sugihara [8]. It does neither require a heavy computation of the pseudoinverse matrix nor the kernel space in each step, but rapidly converges to a solution which satisfies the high-priority constraint as long as it exists. A remaining problem is that the convergence is slowed down in principle when the problem is unsolvable since Lagrange's multiplier linearly converges, while the joint displacement superlinearly converges. However, it is the most preferable to the conventional methods in terms of the robustness. Even in the case that the high-priority constraint is not achievable, it finds the least-square-error solution. Comparative studies with the conventional methods showed that only the proposed method marked 100 % success rate.

This work was supported by 16th Research grant promoted by Kayamori Foundation of Informational Science Advancement.

T. Sugihara is with Department of Adaptive machine systems, Graduate school of Engineering, Osaka University, 2-1 Yamadaoka, Suita, Osaka, Japan. zhidao at ieee.org

II. REVIEW: PRIORITIZED INVERSE KINEMATICS

Suppose the required movement of a robot in the task space is represented by a combination of desired positions and attitudes of several links. The inverse kinematics comes down to a root-finding of the following nonlinear simultaneous equation:

$$\boldsymbol{e}(\boldsymbol{q}) = \boldsymbol{0}, \tag{1}$$

where $\boldsymbol{q} = [q_1 \ q_2 \ \cdots \ q_n]^{\mathrm{T}} \in \mathbb{R}^n$ is the joint displacement vector,

$$\boldsymbol{e}(\boldsymbol{q}) \equiv \left[\boldsymbol{e}_1^{\mathrm{T}}(\boldsymbol{q}) \ \ \boldsymbol{e}_2^{\mathrm{T}}(\boldsymbol{q}) \ \ \cdots \ \ \boldsymbol{e}_N^{\mathrm{T}}(\boldsymbol{q})\right]^{\mathrm{T}} \tag{2}$$

$$\boldsymbol{e}_i(\boldsymbol{q}) \equiv \begin{cases} {}^d\boldsymbol{p}_i - \boldsymbol{p}_i(\boldsymbol{q}) & \text{(positional constraint)} \\ \boldsymbol{a}({}^d\boldsymbol{R}_i \boldsymbol{R}_i^{\mathrm{T}}(\boldsymbol{q})) & \text{(attitude constraint)} \end{cases}, \tag{3}$$

$\boldsymbol{p}_i(\boldsymbol{q}) \in \mathbb{R}^3$ and ${}^d\boldsymbol{p}_i(\boldsymbol{q}) \in \mathbb{R}^3$ are the i'th point of interest on the robot body and its desired value, respectively, $\boldsymbol{R}_i(\boldsymbol{q}) \in SO(3)$ and ${}^d\boldsymbol{R}_i(\boldsymbol{q}) \in SO(3)$ are the attitude of the i'th link of interest and its desired value, respectively, and $\boldsymbol{a}(\boldsymbol{R}) \in \mathbb{R}^3$ for an arbitrary $\boldsymbol{R} \in SO(3)$ means the equivalent angle-axis vector defined as the rotation axis multiplied by the rotation angle. In any type of the constraints, $\boldsymbol{e}_i(\boldsymbol{q})$ measures the error from the desired value.

As noted in the introduction, ${}^d\boldsymbol{p}_i$ and ${}^d\boldsymbol{R}_i$ can be given without any concern with geometric and kinematic constraints of the real robot, so that the solvability of the equation (1) is not guaranteed in general. Let us classify $\boldsymbol{e}(\boldsymbol{q})$ into $\boldsymbol{e}_S(\boldsymbol{q})$, which is required to be zero as the high-priority constraint, and $\boldsymbol{e}_W(\boldsymbol{q})$, which is required to be minimized as much as possible as the low-priority constraint. The prioritized inverse kinematics is mathematically formalized as a quadratic programming with an equality constraint as

$$E \equiv \frac{1}{2}\boldsymbol{e}_W^{\mathrm{T}} \boldsymbol{W}_W \boldsymbol{e}_W \rightarrow \text{min.} \quad \text{subject to} \quad \boldsymbol{e}_S = \boldsymbol{0}, \tag{QP1}$$

where $\boldsymbol{e}_W = \boldsymbol{e}_W(\boldsymbol{q})$, $\boldsymbol{e}_S = \boldsymbol{e}_S(\boldsymbol{q})$, and $\boldsymbol{W}_W$ is a positive-definite weighting matrix, which is usually given as a diagonal matrix.

Let us define the Lagrange function with respect to the above problem as

$$L \equiv \frac{1}{2}\boldsymbol{e}_W^{\mathrm{T}} \boldsymbol{W}_W \boldsymbol{e}_W + \boldsymbol{\lambda}^{\mathrm{T}} \boldsymbol{e}_S, \tag{4}$$

where $\boldsymbol{\lambda}$ is the Lagrange's multiplier. The optimal solution of the problem (QP1) satisfies

$$\left(\frac{\partial L}{\partial \boldsymbol{q}}\right)^{\mathrm{T}} = -\boldsymbol{J}_W^{\mathrm{T}} \boldsymbol{W}_W \boldsymbol{e}_W - \boldsymbol{J}_S^{\mathrm{T}} \boldsymbol{\lambda} = \boldsymbol{0} \tag{5}$$

$$\left(\frac{\partial L}{\partial \boldsymbol{\lambda}}\right)^{\mathrm{T}} = \boldsymbol{e}_S = \boldsymbol{0}, \tag{6}$$

where

$$\frac{\partial \boldsymbol{e}_W}{\partial \boldsymbol{q}} \simeq -\boldsymbol{J}_W \tag{7}$$

$$\frac{\partial \boldsymbol{e}_S}{\partial \boldsymbol{q}} \simeq -\boldsymbol{J}_S \tag{8}$$

are used. $\boldsymbol{J}_W$ and $\boldsymbol{J}_S$ are the basic Jacobian matrices [11] which map the infinitesimal joint displacement to the corresponding positions and attitudes of links, respectively. The problem comes down to a root-finding of the above simultaneous equation.

The family of gradient methods which originate from Newton-Raphson's method first linearizes Eqs. (5) and (6) around $\boldsymbol{q} = \boldsymbol{q}_k$ at the k'th step of iteration as

$$\boldsymbol{J}_{Wk}^{\mathrm{T}} \boldsymbol{W}_W \boldsymbol{J}_{Wk} \Delta\boldsymbol{q}_k + \boldsymbol{J}_{Sk}^{\mathrm{T}} \boldsymbol{\lambda}_k = \boldsymbol{J}_{Wk}^{\mathrm{T}} \boldsymbol{W}_W \boldsymbol{e}_{Wk} \tag{9}$$

$$\boldsymbol{J}_{Sk} \Delta\boldsymbol{q}_k = \boldsymbol{e}_{Sk}, \tag{10}$$

where $\boldsymbol{J}_{Wk} = \boldsymbol{J}_W(\boldsymbol{q}_k)$, $\boldsymbol{J}_{Sk} = \boldsymbol{J}_S(\boldsymbol{q}_k)$, $\boldsymbol{e}_{Wk} = \boldsymbol{e}_W(\boldsymbol{q}_k)$ and $\boldsymbol{e}_{Sk} = \boldsymbol{e}_S(\boldsymbol{q}_k)$. The above equations are packed to

$$\begin{bmatrix} \boldsymbol{J}_{Wk}^{\mathrm{T}} \boldsymbol{W}_W \boldsymbol{J}_{Wk} & \boldsymbol{J}_{Sk}^{\mathrm{T}} \\ \boldsymbol{J}_{Sk} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{q}_k \\ \boldsymbol{\lambda}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_{Wk}^{\mathrm{T}} \boldsymbol{W}_W \boldsymbol{e}_{Wk} \\ \boldsymbol{e}_{Sk} \end{bmatrix}. \tag{11}$$

By solving the above equation and updating $\boldsymbol{q}$ as

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \Delta\boldsymbol{q}_k, \tag{12}$$

it is expected to asymptotically converge to the solution of Eqs. (5) and (6). Note that $\boldsymbol{\lambda}_k$ is directly acquired in each step.

The problem here is that the size and rank of $\boldsymbol{J}_{Sk}$ and $\boldsymbol{J}_{Wk}$ can be any integer values in general, and accordingly, neither the regularity nor the positive-definiteness of the coefficient matrix of the left side of Eq. (11) is not guaranteed. Although Moore-Penrose's inverse matrix (MP-inverse) provides a formal solution of $\Delta\boldsymbol{q}_k$, it causes serious numerical instability near the singular point due to ill-posedness. The prioritized inverse kinematics deals with unsolvable equations in nature, and thus, the solution is often found around the singular point. Hence, the iteration usually fails to converge with this approach.

Now, one may know that Eq. (11) can be solved without explicitly computing $\boldsymbol{\lambda}$. Let us define $\boldsymbol{A}^{\#}$ for an arbitrary matrix $\boldsymbol{A}$ as the weighted MP-inverse, hereafter. The general solution of Eq. (10) is represented as

$$\Delta\boldsymbol{q}_k = \boldsymbol{q}_{Sk} + \boldsymbol{N}_{Sk} \boldsymbol{y}, \tag{13}$$

where $\boldsymbol{q}_S \equiv \boldsymbol{J}_S^{\#} \boldsymbol{e}_S$, $\boldsymbol{q}_{Sk} \equiv \boldsymbol{q}_S(\boldsymbol{q}_k)$, $\boldsymbol{N}_S \equiv \boldsymbol{J}_S^{\#} \boldsymbol{J}_S - \boldsymbol{1}$ is the basis matrix of the kernel space of $\boldsymbol{J}_S$, $\boldsymbol{N}_{Sk} \equiv \boldsymbol{N}_S(\boldsymbol{q}_k)$ and $\boldsymbol{y}$ is an arbitrary $n \times 1$ vector. When treating $\boldsymbol{y}$ as a new design variable, the following equation is satisfied at the optimum of the problem (QP1):

$$\left(\frac{\partial E}{\partial \boldsymbol{y}}\right)^{\mathrm{T}} = \boldsymbol{0}$$

$$\Leftrightarrow \quad \tilde{\boldsymbol{J}}_W^{\mathrm{T}} \boldsymbol{W}_W \tilde{\boldsymbol{J}}_W \boldsymbol{y} = \tilde{\boldsymbol{J}}_W^{\mathrm{T}} \boldsymbol{W}_W (\boldsymbol{e}_W - \boldsymbol{J}_W \boldsymbol{q}_S), \tag{14}$$

where $\tilde{\boldsymbol{J}}_W = \boldsymbol{J}_W \boldsymbol{N}_S$. Then, let

$$\boldsymbol{y} = \tilde{\boldsymbol{J}}_{Wk}^{\#} (\boldsymbol{e}_{Wk} - \boldsymbol{J}_{Wk} \boldsymbol{q}_{Sk}), \tag{15}$$

where $\tilde{\boldsymbol{J}}_{Wk} \equiv \tilde{\boldsymbol{J}}_W(\boldsymbol{q}_k)$. We get $\Delta\boldsymbol{q}_k$ by putting this into Eq. (13). The above method was proposed by Nakamura, Hanafusa and Yoshikawa [1]. Obviously, it requires heavy computations such as multiplications of matrices and MP-inverse matrices. It should be noted that Eq. (15) is a solution of the following linear simultaneous equations:

$$\tilde{\boldsymbol{J}}_{Wk}\boldsymbol{y} = \boldsymbol{e}_{Wk} - \boldsymbol{J}_{Wk}\boldsymbol{q}_{Sk}, \tag{16}$$

which can be solved with $O(n^2)$ computation complexity without computing MP-inverse matrix $\tilde{\boldsymbol{J}}_{Wk}^{\#}$ as well as $\boldsymbol{q}_{Sk}$, even though they are based on MP-inverse. However, the computation of the coefficient matrix $\tilde{\boldsymbol{J}}_{Wk}$ itself requires an explicit computation of MP-inverse $\boldsymbol{J}_{Sk}^{\#}$ since it includes $\boldsymbol{N}_{Sk}$, which is with $O(n^3)$ computation complexity. The problem of numerical instability around the singular point still exists. Also, an implementation of complex computation is often accompanied with bugs.

Nakamura and Hanafusa [12], and Chiaverini [13] proposed to use the singularity-robust inverse matrix (SR-inverse) instead of MP-inverse in Eq. (15). It is supported by a fact that SR-inverse does not affect the kernel space. It is also formally possible to use SR-inverse in Eq. (13). Those ideas help to reduce the risk of numerical instability but are with almost the same computation complexity. The set of $\boldsymbol{q}$ which satisfies the high-priority constraint forms a manifold, and the inverse kinematics is to find a zero of the manifold. The discussion should be made in terms of the behavior of $\boldsymbol{q}$ on the manifold rather than that on the tangential hyperplane of $\boldsymbol{q}$. In this sense, it is not essential to discuss the image space and the kernel space of the tangential hyperplane.

To summarize, the above method is mathematically rational as a solution to the prioritized *differential* inverse kinematics, but is too complex as that to the prioritized inverse kinematics, so that its cost for implementation and computation does not pay off.

III. Robust solution of prioritized inverse kinematics based on Multiplier method and LM method

The multiplier method is an iterative solution for a quadratic programming with an equality constraint, which was proposed independently by Hestenes [9] and Powell [10]. It simultaneously updates the design variable and its adjoint variable, namely, Lagrange's multiplier, and approaches to the optimum point on the manifold corresponding to the equality constraint. In this method, the following extended Lagrange function is used instead of Eq. (4):

$$L' \equiv \frac{1}{2}\boldsymbol{e}^{\mathrm{T}}\boldsymbol{W}_E\boldsymbol{e} + \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{e}_S, \tag{17}$$

where

$$\boldsymbol{e} \equiv \begin{bmatrix} \boldsymbol{e}_S \\ \boldsymbol{e}_W \end{bmatrix}, \qquad \boldsymbol{W}_E \equiv \begin{bmatrix} \boldsymbol{1} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{W}_W \end{bmatrix}. \tag{18}$$

This means that the quadratic programming (QP1) is equivalently modified as

$$E' \equiv \frac{1}{2}\boldsymbol{e}_S^{\mathrm{T}}\boldsymbol{e}_S + \frac{1}{2}\boldsymbol{e}_W^{\mathrm{T}}\boldsymbol{W}_W\boldsymbol{e}_W \to \text{min.} \quad \text{subject to} \quad \boldsymbol{e}_S = \boldsymbol{0}. \tag{QP2}$$

The following equations are satisfied at the optimum of the problem (QP2):

$$\left(\frac{\partial L'}{\partial \boldsymbol{q}}\right)^{\mathrm{T}} = -\boldsymbol{J}_W^{\mathrm{T}}\boldsymbol{W}_W\boldsymbol{e}_W - \boldsymbol{J}_S^{\mathrm{T}}(\boldsymbol{e}_S + \boldsymbol{\lambda}) = \boldsymbol{0} \tag{19}$$

$$\left(\frac{\partial L'}{\partial \boldsymbol{\lambda}}\right)^{\mathrm{T}} = \boldsymbol{e}_S = \boldsymbol{0}. \tag{20}$$

One may find that it is equivalent with Eqs. (5) and (6) with a notice to Eq. (20), namely, the optimum of (QP2) is also that of the original problem (QP1). Furthermore, a comparison of Eqs. (5) and (19) inplicates that, when $\boldsymbol{q}$ which minimizes L' for a given $\boldsymbol{\lambda}$ is acquired, the corresponding $\boldsymbol{e}_S + \boldsymbol{\lambda}$ is closer to Lagrange's multiplier at the optimum than $\boldsymbol{\lambda}$. Refer the original papers for detailed discussions.

Based on the above, we have an updating rule of the multiplier method standing upon LM method as

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \left(\boldsymbol{J}_k^{\mathrm{T}}\boldsymbol{W}_E\boldsymbol{J}_k + \boldsymbol{W}_{Nk}\right)^{-1}\boldsymbol{J}_k^{\mathrm{T}}\boldsymbol{W}_E\boldsymbol{e}'_k \tag{21}$$

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \boldsymbol{e}_{Sk}, \tag{22}$$

where

$$\boldsymbol{J}_k \equiv \begin{bmatrix} \boldsymbol{J}_{Sk} \\ \boldsymbol{J}_{Wk} \end{bmatrix}, \qquad \boldsymbol{e}'_k \equiv \begin{bmatrix} \boldsymbol{e}_{Sk} + \boldsymbol{\lambda}_k \\ \boldsymbol{e}_{Wk} \end{bmatrix}. \tag{23}$$

$\boldsymbol{W}_{Nk}$ is the damping factor in LM method. For the readers' reference, the original method is a tandem iteration which alternates the unconstrained optimization of L' for a given λ_k and the update of λ_k, while the proposed method simultaneously updates $\boldsymbol{q}_k$ and $\boldsymbol{\lambda}_k$ in every step.

The above updating rule is only different from that in the weighted inverse kinematics based on LM method [8] at that it accumulates the residual of the high-priority constraint by Eq. (22) and uses $\boldsymbol{e}'_k$ instead of $\boldsymbol{e}_k$. Namely, the computation cost in each step of iteration and the implementation cost are almost the same with that of the weighted inverse kinematics. It is also expected that it robustly minimizes the norm of $\boldsymbol{e}_S$ even in unsolvable cases where any $\boldsymbol{q}$ does not satisfy the equality constraint. A drawback is that $\boldsymbol{\lambda}_k$ linearly converges, while $\boldsymbol{q}_k$ superlinearly converges.

Constraints on attitude should be carefully handled. Since the error of attitude is defined by the angle-axis representation, it is inadequate to sum it up for the error accumulation. In the author's implementation, the following computation is employed for a high-priority constraint on attitude $\boldsymbol{e}_i$:

$$\boldsymbol{\epsilon}_i = \boldsymbol{\epsilon}(\boldsymbol{e}_i) \tag{24}$$

$$\boldsymbol{\lambda}_{i,k+1} = \frac{\boldsymbol{\lambda}_{i,k} + \boldsymbol{\epsilon}_i}{\|\boldsymbol{\lambda}_{i,k} + \boldsymbol{\epsilon}_i\|} \tag{25}$$

$$\boldsymbol{e}'_i = \boldsymbol{e}(\boldsymbol{\lambda}_{i,k}), \tag{26}$$

where $\boldsymbol{\lambda}_{i,k}$ is a value of the adjoint variable corresponding to $\boldsymbol{e}_i$ at the k'th step of iteration, $\boldsymbol{\epsilon}(\boldsymbol{e})$ for an arbitrary angle-axis vector $\boldsymbol{e}$ is a function which converts $\boldsymbol{e}$ to a unit quaternion for the Euler parameters, and $\boldsymbol{e}(\boldsymbol{\epsilon})$ for an arbitrary Euler parameters $\boldsymbol{\epsilon}$ is a function which converts $\boldsymbol{\epsilon}$ to an angle-axis vector.

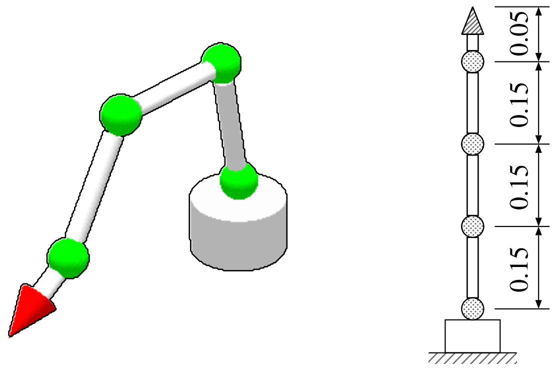


Fig. 1. Kinematics model of the tested redundant manipulator comprising five links and four spherical joints

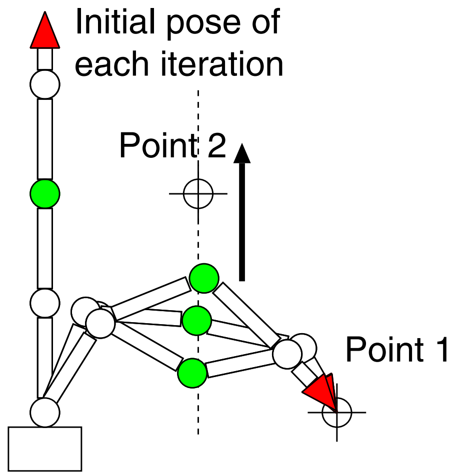


Fig. 2. Test 1: prioritized IK with solvable constraint, where the target of Point 1 (highly-prioritized constraint) is fixed while that of Point 2 (lowly-prioritized constraint) moves upward as to be out of reach.

IV. Evaluation

The proposed method was evaluated particularly from the viewpoint of robustness and the computation time with a kinematic model of a manipulator depicted in Fig. 1. This model is the same with the one used in Sugihara [8], which comprises 5 links serially connected by four spherical joints. The total degree-of-freedom is 12. The distance between joint axes are uniformly 0.15[m], and that from the center of the last joint axis to the endpoint is 0.05[m]. An inherent redundancy of the manipulator necessitates a numerical solution of the inverse kinematics. In addition, many singular points are eccentically located within its workspace, so that it favors for measuring the robustness of the algorithm.

The following two tests were conducted.

Test 1: remain the endpoint (Point 1) at position $(0, 0.4, 0)$ and approach the center of the third joint axis (Point 2) as close to $(0, 0.2, 0.005i)$ as possible, where i is an integer value from 0 to 100 so that the desired z-value of Point 2 varies from 0 to 0.5 (the maximum height of the manipulator), as illustrated in Fig. 2.

Test 2: remain the desired position of Point 2 at $(0, 0.2, 0.5)$ and move the desired position of Point 1 at $(0, 0.4+0.004i, 0)$, where i is an integer value from 0 to 100 so that the desired y-value of Point 1 varies from 0.4 to 0.8 (farther than the maximum length of the manipulator), as illustrated in Fig. 3.

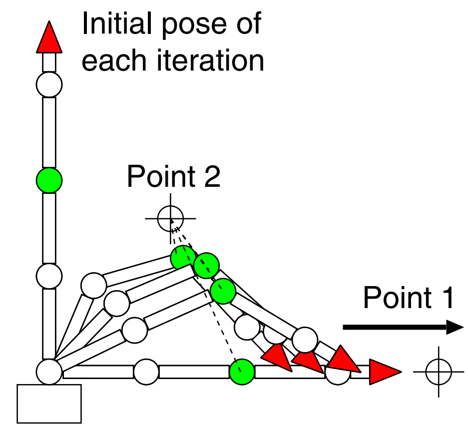


Fig. 3. Test 2: prioritized IK with unsolvable constraint, where the target of Point 1 (highly-prioritized constraint) moves forward as to be out of reach, while that of Point 2 (lowly-prioritized constraint) is fixed.

The following 6 cases were examined for comparison.

- $w = 1.0$: weighted inverse kinematics proposed by Sugihara [8], where the weight on the high-priority constraint is set for 1.0
- $w = 10.0$: same with the above, where the weight on the high-priority constraint is set for 10.0
- $w = 100.0$: same with the above, where the weight on the high-priority constraint is set for 100.0
- HYN-MP: prioritized inverse kinematics proposed by Nakamura, Hanafusa and Yoshikawa [1], where SR-inverse is used in Eq. (15)
- HYN-SR: same with the above, where SR-inverse is used in Eqs. (13) and (15)
- Proposed: the proposed method

An efficient way to compute MP-inverse and the basis matrix of its kernel space, which is necessary for HYN-MP, is shown in Appendix. For the readers' reference, HYN-MP is adopted in Yamane et al. [14]. The weight on the low-priority constraint was set for 1.0. $\boldsymbol{q}$ was initialized to $\boldsymbol{q}_0 = \boldsymbol{0}$ in every computation. Note that it is the singular point. Iteration was terminated if any of the three condition was satisfied.

i) Absolute values of all components of $\Delta\boldsymbol{q}_k$ are less than $\varepsilon = 1.0 \times 10^{-12}$
ii) Abstolute value of deviation of $\|\boldsymbol{e}_k\|$ from the previou step is less than $\delta = 1.0 \times 10^{-12}$
iii) The total number of iteration was over 10,000 times

Fig. 4 shows the result of Test 1. (a) is the error of the high-priority constraint. In the weighted inverse kinematics, it decreases as the weight increases. However, no matter how large the weight is increased, it increases as the desired position of Point 2 goes further. The weighted inverse kinematics cannot guarantee to lower the error than a certain threshould with respect to an arbitrarily given desired value. It reaffirms that the weighted inverse kinematics solver is not sufficient for the prioritized inverse kinematics and another approach is required for the problem. The proposed method succeeded to lower the error of the high-priority constraint than the threshould as well as the conventional Nakamura, Hanafusa and Yoshikawa's method. (b) is the error of the low-priority constraint. It shows that the weighted inverse kinematics sacrifices the strictness of the high-priority constraint, and as a side-effect, the error of the low-priority constraint is lowered. It is also expected that the error asymptotically approaches to that of the prioritized inverse kinematics. In terms of the computation time, (c) shows that the proposed method is faster than the other methods in the cases where the error of the low-priority constraint is small, but becomes slower as the error of the low-priority constraint increases. It is because the computation time per step of the proposed method is short but it requires more steps of iteration due to its linearly convergent property.

Fig. 5 shows the result of Test 2. (a)(b) presents a fact that the conventional Nakamura, Hanafusa and Yoshikawa's method becomes numerically unstable when the high-priority constraint is given near the singular point or out of the achievable range. Since HYN-MP and HYN-SR do not differ from each other in terms of this property, one should know that it cannot be resolved by replacing some MP-inverse for SR-inverse in the computation process. On the other hand, the proposed method solely succeeded in the all cases, so that it is obviously favorable from the viewpoint of robustness. In this case, the computation time of the proposed method is also shorter as (c) shows, though its poor-convergence still remains as a problem.

V. Conclusion

A novel solution to the prioritized inverse kinematics based on Hestenes-Powell's multiplier method was proposed. The pros and cons are summarized as follows.

I) While the conventional method explicitly manipulates the kernel space and hence is accompanied with a heavy computation, the proposed method has almost the same implementation cost and the computation cost in each step with that of the weighted inverse kinematics as it only adds the accumulation of the error of the high-priority constraint and the estimation of Lagrange's multiplier.

II) As well as in the (general) inverse kinematics, the solvability of the problem, namely, the existence of the solution which satisfies the high-priority constraint, is not guaranteed and it is more difficult to judge if the problem is solvable in advance in the prioritized inverse kinematics. The conventional method does not have any countermeasure against this issue and often fails. On the other hand, the proposed method is robust, being unconcerned with the solvability of the problem, as long as the original weighted inverse kinematics solver is robust. In fact, the proposed method solely succeeded in all the tested cases including unsolvable ones.

III) In the proposed method, Lagrange's multiplier linearly converges, while the joint displacement superlinearly converges. It makes the method slow particularly when the high-priority constraint is unachievable, so that it requires more computation cost in total than the conventional method in some situations. It is the future work to improve the converegence performance.

The prioritized inverse kinematics is an important computation particularly for a robot which leverages a large number of degrees-of-freedom and keeps contacts at many points with the environment to perform e.g. a humanoid robot. The authors are undertaking a work to apply the proposed method to designing a lively motion of a humanoid robot which positively uses the limits of movements such as knee-stretched postures but robustly keeps the supporting region and satisfies the dynamical constraint. It will be presented in the nearest conference.

References

[1] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task Priority Based Redundancy Control of Robot Manipulators," *International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.

[2] D. L. Pieper, "The Kinematics of Manipulators under Computer Control," Ph.D. dissertation, Stanford University, 1968.

[3] M. Raghavan and B. Roth, "Kinematic Analysis of the 6R Manipulator of General Geometry," in *Proceedings of The Fifth International Symposium on Robotics Research*, 1991, pp. 263–269.

[4] ——, "Solving Polynomial Systems for the Kinematic Analysis and Synthesis of Mechanisms and Robot Manipulators," *Transaction of the ASME, Special 50th Anniversary Design Issue*, vol. 117, pp. 71–79, 1995.

[5] S. Sasaki, "Feasibility Studies of Kinematic Problems in the Case of a Class of Redundant Manipulators," *Robotica*, vol. 13, pp. 233–241, 1995.

[6] A. A. Maciejewski and C. A. Klein, "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments," *International Journal of Robotic Research*, vol. 4, no. 3, pp. 109–117, 1985.

[7] D. E. Whitney, "Resolved Motion Rate Control of Manipulators and Human Prostheses," *IEEE Transactions on Man-Machine Systems*, vol. 10, no. 2, pp. 47–53, 1969.

[8] T. Sugihara, "Solvability-Unconcerned Inverse Kinematics by the Levenberg-Marquardt Method," *IEEE Transaction on Robotics*, vol. 27, no. 5, pp. 984–991, 2011.

[9] M. R. Hestenes, "Multiplier and Gradient Methods," *Journal of Optimization Theory and Applications*, vol. 4, pp. 303–320, 1969.

[10] M. J. D. Powell, "A Method for Nonlinear Constraints in Minimization Problems," in *Optimization*, R. Fletcher, Ed. Academic Press, 1969, pp. 283–298.

[11] O. Khatib, "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation," *The International Journal of Robotics and Automation*, vol. RA-3, no. 1, pp. 43–53, 1987.

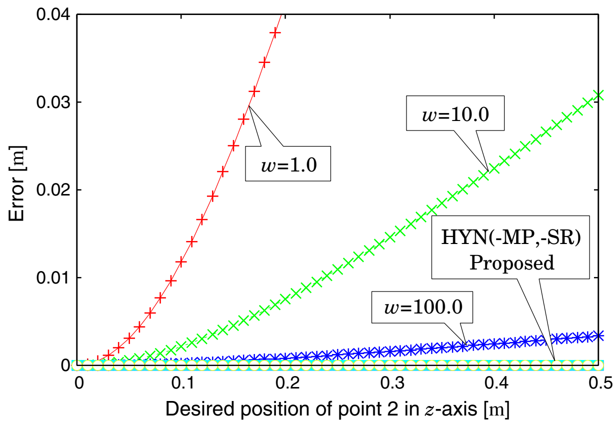
[12] Y. Nakamura and H. Hanafusa, "Inverse Kinematic Solutions with Singularity Robustness for Robot Manipulator Control," *Journal of Dynamic Systems, Measurement and Control*, vol. 108, pp. 163–171, 1986.

[13] S. Chiaverini, "Singularity-Robust Task-Priority Redundancy Resolution for Real-Time Kinematic Control of Robot Manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, 1997.

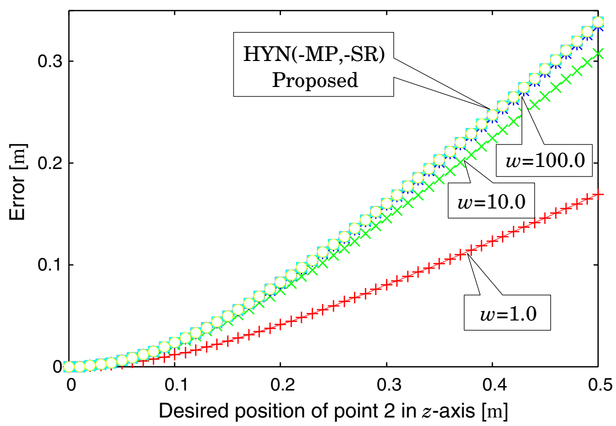
[14] K. Yamane and Y. Nakamura, "Natural Motion Animation through Constraining and Deconstraining at will," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 3, pp. 352–360, 2003.

Appendix

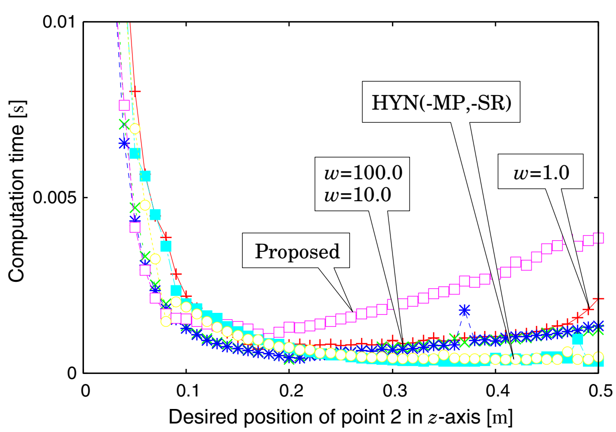
MP-inverse matrix and Basis matrix of the kernel space using LQ decomposition



(a) Error at Point 1

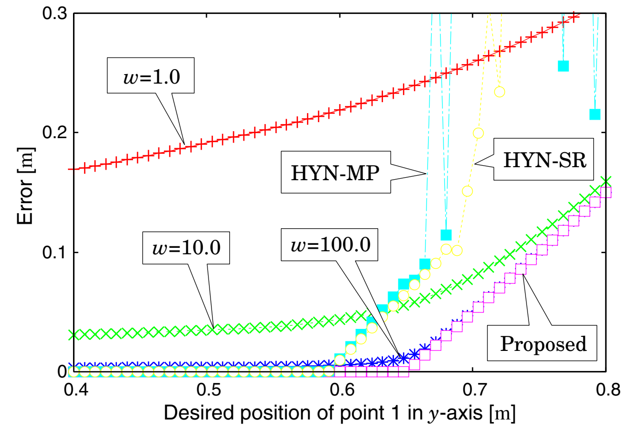


(b) Error at Point 2

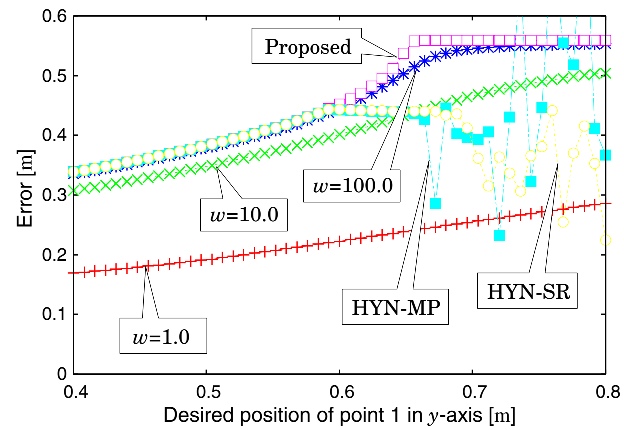


(c) Computation time

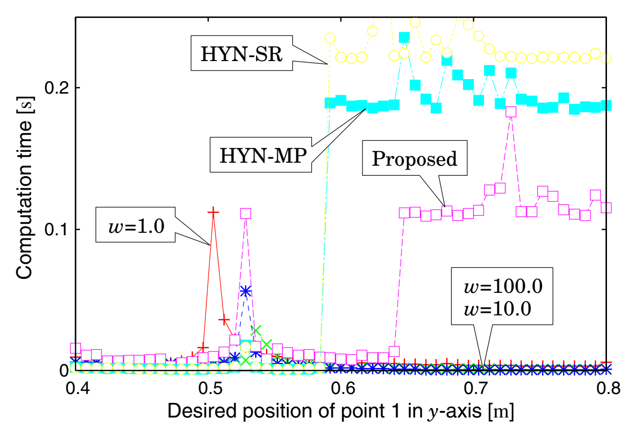
Fig. 4. Result of Test 1



(a) Error at Point 1



(b) Error at Point 2



(c) Computation time

Fig. 5. Result of Test 2

Let us decompose a matrix $\boldsymbol{A}$ into the following LQ form:

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{Q}, \tag{27}$$

where $\boldsymbol{L}$ and $\boldsymbol{Q}$ are guaranteed to be column-full-rank and row-full-rank, respectively. $\boldsymbol{Q}$ is an orthonormal matrix. The MP-inverse matrix, or more strictly, the weighted MP-inverse matrix of $\boldsymbol{A}$, $\boldsymbol{A}^{\#}$ is computed as

$$\boldsymbol{A}^{\#} = \boldsymbol{Q}^{\mathrm{T}}(\boldsymbol{L}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{L})^{-1}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{W}, \tag{28}$$

where $\boldsymbol{W}$ is a positive-definite weighting matrix, which is usually chosen as a diagonal matrix. The basis matrix of the kernel space of $\boldsymbol{A}$, $\boldsymbol{N}$, is also computed as

$$\begin{aligned}\boldsymbol{N} &= \boldsymbol{A}^{\#}\boldsymbol{A} - \boldsymbol{1} = \boldsymbol{Q}^{\mathrm{T}}(\boldsymbol{L}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{L})^{-1}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{L}\boldsymbol{Q} - \boldsymbol{1} \\ &= \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q} - \boldsymbol{1}.\end{aligned} \tag{29}$$

If $\boldsymbol{Q}$ is full-rank, $\boldsymbol{N} = \boldsymbol{O}$ from the orthonormality.