International Journal of Control, Automation, and Systems (2014) 12(5):1102-1109
DOI 10.1007/s12555-013-0102-0

ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Scaled Jacobian Transpose based Control for Robotic Manipulators

An Yong Lee, Jongguk Yim, and Youngjin Choi*

Abstract: This paper presents a scaled Jacobian transpose based control method for robotic manipulators as a modification of a conventional Jacobian transpose based method. The proposed method has several advantages such as it shows faster convergence and better tracking performance than the conventional method, furthermore, it does not have any singularity problem similar to the conventional method. The scaled Jacobian transpose is obtained by collecting each pseudoinverse of the column vector of the Jacobian matrix. The proposed method performs a given task well under singular configurations while minimizing the task error. Finally, a few comparative studies with the conventional method are provided to show the effectiveness of the proposed method through simulations.

Keywords: Jacobian, robotic manipulator, singularity.

1. INTRODUCTION

Inverse kinematics is still one of the important issues for robotic manipulation [1-3]. Especially redundant manipulators have more joint DoFs (degrees of freedom) for natural behavior like human arms than the minimum number of DoFs required to perform a task specified in the task space. The remaining joint DoFs are referred to as redundancy. Although both position and orientation of end-effector of the redundant manipulator can be calculated using the forward kinematics, it does not have a unique inverse kinematic solution at the level of joint configuration. As an alternative, a differential (or rate) inverse kinematics and the Euler integration method have been utilized to achieve inverse kinematics solutions. Since the differential inverse kinematics requires the pseudoinverse of Jacobian matrix, however, we often meet the kinematic singularity problems whenever the Jacobian matrix loses rank. Indeed the singularity can be classified as kinematic singularity, algorithmic singularity, and representation singularity in [4-7]; the kinematic singularity is commonly caused when Jacobian matrix loses the rank, the algorithmic singularity is mainly caused when two or more tasks are applied to one manipulator system, and the representation singularity happens when extracting three Euler angles from orientation matrix. The singularity avoidance has been achieved through the careful planning with respect to the desired trajectory such a way to keep it out of singular configurations [8].

For set-point regulation control, it was proven in [5] that Jacobian transpose based (in short JT-based) method provides a stable solution without causing any singularity problems. When the JT-based method is applied to the trajectory tracking control, however, we know that it does not always provide a reliable trajectory tracking performance because it was devised only for the set-point regulation control. Also since most robotic tasks are composed of trajectory tracking such as moving an object, grasping, and de-burring, and welding, the trajectory tracking control is more important than the set-point regulation control for a variety of robotic manipulations.

For trajectory tracking control, the Jacobian pseudoinverse based (JP-based) method has been utilized with a concept of closed-loop inverse kinematics (CLIK) in [4,5,7-10] but the JP-based method includes typical singularity problems. In order to alleviate the singularity phenomenon of JP-based method, a damped least-squares (DLS) and its variants have been proposed in [4,11,12]. The performance of these methods depends greatly on the selection of dampening parameter value, for example, a large dampening parameter provides the singularity robustness to the JP-based method at the cost of accuracy of task execution [9].

The goal of this paper presents a novel method to solve the differential inverse kinematics of robotic manipulators. The method to be suggested in the paper does not contain the kinematic singularity. For this, all of the

Manuscript received March 11, 2013; revised February 12, 2014; accepted April 8, 2014. Recommended by Associate Editor Seul Jung under the direction of Editor Hyouk Ryeol Choi.

This work was supported in part by the Global Frontier R&D Program on “Human-centered Interaction for Coexistence” funded by the National Research Foundation of Korea grant funded by the Korean Government (MSIP) (NRF-2012M3A6A3057080), in part by the National Research Foundation (NRF-2013R1A1A 2010192), and in part by the BK21 Plus Program (Future-oriented innovative brain raising type, 22A20130012806) funded by the Ministry of Education (MOE, Korea) and National Research Foundation of Korea (NRF), Korea.

An Yong Lee is with the Department of Electrical, Electronic, Control and Instrumentation Engineering, Hanyang University, Korea (e-mail: dragonlay@hanyang.ac.kr).

Jongguk Yim is with the Research Institute of Engineering and Technology, Hanyang University, Korea (e-mail: jgyim@hanyang.ac.kr).

Youngjin Choi is with the Department of Electronic Systems Engineering, Hanyang University, Hanyangdaehakro 55, Sangrok-gu, Ansan 426-791, Korea (e-mail: cyj@hanyang.ac.kr).

* Corresponding author.

© ICROS, KIEE and Springer 2014

pseudoinverses of each column vector of the Jacobian matrix are again collected into one matrix. Since it has the form of Jacobian transpose multiplied with the scaling factors, it is referred to as a scaled Jacobian transpose based (SJT-based) method in this paper. Actually the idea of the Jacobian partitioning had been utilized in [13], in which the exactness of the inverse kinematics was achieved through the iterative and sequential null space projections of the column vector of the Jacobian. The SJT-based method, however, compensates for the error by using the CLIK. Now we assume that each link system of the manipulator pursues the realization of the desired motion. It is formulated as an optimization problem for each link system and its solutions consist of the core element of the SJT-based method.

This paper is organized as follows; Section 2 provides literature survey and research motivation; Section 3 proposes the SJT-based method as a main result of the paper; Section 4 suggests several simulation results for comparison with the conventional JT-based method, and finally, Section 5 draws the conclusion of the paper.

2. PRELIMINARIES

Most tasks of a robotic manipulator are described as functions of time in 6-dimensional task space. Also the desired task velocity is obtained by taking time derivative with respect to the desired trajectory given as a function of time. As aforementioned this paper deals with the differential inverse kinematics to obtain the desired joint configurations of the robotic manipulator. Let us define the 6-dimensional task velocity of the end-effector as follows:

$$\dot{x} \triangleq \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix}, \tag{1}$$

where $\dot{x} \in \Re^6$ means the task velocity, $\dot{p}_e \in \Re^3$ and $\omega_e \in \Re^3$ imply a translational velocity and an angular velocity of end-effector of robotic manipulator with respect to the base frame, respectively. Also the relation between the task velocity and n-dimensional joint velocity is expressed as follows:

$$\dot{x} = J(q)\dot{q}, \tag{2}$$

where $\dot{q} \in \Re^n$ implies the joint velocity and $J(q)$ is a geometric (or manipulator) Jacobian defined as the following form:

$$J(q) \triangleq \begin{bmatrix} z_1 \times (p_e - p_1) & \cdots & z_n \times (p_e - p_n) \\ z_1 & \cdots & z_n \end{bmatrix} \in \Re^{6\times n} \tag{3}$$

in which z_i is the z-directional unit vector of i th coordinate frame with respect to the base frame, p_i is the position vector of i th coordinate origin with respect to the base, for $i = 1, 2, \cdots, n,$ and p_e is the position vector of the end-effector with respect to the base. Here note that $n > 6$ in the case of the redundant manipulator.

2.1. JP-based method

The differential relation of (2) can be utilized for solving the inverse kinematics. For a given desired task $x_d(t)$ and $\dot{x}_d(t)$, the aim of the inverse kinematics is to find feasible joint configuration $q_d(t)$ and joint velocity $\dot{q}_d(t)$ in order to reproduce the desired task after applying forward kinematics to the manipulator system. By applying Moore-Penrose pseudoinverse to the Jacobian matrix, we arrive at the differential inverse kinematics as follows:

$$\dot{q}_d(t) = J^+(q)\dot{x}_d(t), \tag{4}$$

where $J^+(q) = J^T(q)[J(q)J^T(q)]^{-1}$ denotes the Moore-Penrose right pseudoinverse of $J(q)$, if $J(q)$ has full row rank. Indeed, the above right pseudoinverse is easily derived from a simple optimization; for instance, minimizing $\| \dot{q}_d \|$ subject to the constraint of $\dot{x}_d = J(q)\dot{q}_d$ brings (4). Also since a numerical integration is required to obtain $q_d(t)$ from $\dot{q}_d(t)$, the integration error is inevitably caused while integrating it numerically. As an alternative, the closed-loop inverse kinematics (CLIK) has been adopted to compensate for the numerical error as follow:

$$\dot{q}_r(t) = J^+(q)\{\dot{x}_d(t) + K[x_d(t) - x(t-\Delta t)]\}, \tag{5}$$

where $K \in \Re^{6\times 6}$ is the gain matrix having only positive diagonal elements and $x(t-\Delta t)$ is the forward kinematics solution obtained at the previous time step. The sampling time step Δt is normally equal to the control sampling time. The above equation (5) is called as the conventional JP-based control method.

Now let us define a small disturbance to be $\delta(t) \triangleq x(t-\Delta t) - x(t),$ and if (5) is applied to the differential forward kinematics $\dot{x}(t) = J(q)\dot{q}_r(t),$ then we have the following equation:

$$\dot{x}(t) = J(q)J^+(q)\{\dot{x}_d(t) + K[x_d(t) - x(t-\Delta t)]\}. \tag{6}$$

Here since $J(q)J^+(q) = I$ with full rank $J(q)$, above (6) can be modified as following linear system equation:

$$\dot{e}(t) + Ke(t) = K\delta(t), \tag{7}$$

where $e(t) \triangleq x_d(t) - x(t)$ means a task error. Since K is the positive constant gain matrix, we can conclude that the linear system of (7) is stable and also bounded for the bounded disturbance $\delta(t)$. Finally the conventional JP-based method can be summarized as follows:

$$\dot{q}_r(t) = J^+(q)\{\dot{x}_d(t) + K[x_d(t) - x(t-\Delta t)]\}, \tag{8}$$
$$q_r(t) = q_r(t-\Delta t) + \dot{q}_r(t)\Delta t. \tag{9}$$

However, the above JP-based method of (8) and (9) contains the kinematic singularities whenever the corresponding Jacobian loses rank.

2.2. JT-based method

The JT-based method has been derived from the modification of the JP-based method as follows:

$$\dot{q}_r(t) = J^T(q)\{\dot{x}_d(t) + K[x_d(t) - x(t-\Delta t)]\}, \quad (10)$$

$$q_r(t) = q_r(t-\Delta t) + \dot{q}_r(t)\Delta t \quad (11)$$

in which the $J^+ = J^T(JJ^T)^{-1}$ of the JP-based method was replaced with the J^T because the both of J^+ and J^T span the same range space as well as the JT-based method does not require the matrix inversion. As we can see in (10), the JT-based method does not have any singularities, but it does not guarantee the error convergence in the case of the trajectory tracking. When the JT-based method is utilized, the selection of the gain matrix becomes important issue to improve the tracking performance.

On the other hand, if we consider the set-point regulation, then the task is just defined as the set-point x_s to be regulated, not as the trajectory to be tracked. Thus the JT-based method of (10) can be reduced to the following form:

$$\dot{q}_r(t) = J^T(q)K[x_s - x(t-\Delta t)]. \quad (12)$$

For the given set-point regulation task, if (12) is utilized with full rank $J(q)$, then $x(t) \to x_s$ according as $t \to \infty$, namely, the globally asymptotic convergence for an equilibrium point x_s is guaranteed as proven in [5]. However, this property of the error convergence is not valid for the trajectory tracking task.

Although the JT-based method of (10) does not cause the kinematic singularity problem, it would have a relatively significant error when it is applied to the trajectory tracking tasks. In the next section, we propose a new scaled Jacobian transpose based (SJT-based) method; it is singularity-free and it shows better tracking performance than the JT-based method.

3. SCALED JACOBIAN TRANSPOSE BASED (SJT-BASED) METHOD

For the development of the SJT-based method, we assume that each link of the manipulator pursues the realization of the desired task motion. For this, the Jacobian matrix is partitioned column by column and then the optimization is solved for each link system. Let us first consider i th column vector of the Jacobian of (3) as follows:

$$J_i(q) \triangleq \begin{bmatrix} z_i \times (p_e - p_i) \\ z_i \end{bmatrix} \in \Re^6 \quad \text{for} \quad i = 1, \cdots, n, \quad (13)$$

where J_i is an i th partitioned Jacobian for i th link system. Then the differential forward kinematics can be decomposed by using the partitioned Jacobians as follows:

$$\dot{x}_d = \sum_{i=1}^{n} J_i(q)\dot{q}_{d,i}. \quad (14)$$

Here we can know that the contribution of i th link system is $J_i(q)\dot{q}_{d,i}$ toward the realization of the desired task.

3.1. Derivation of SJT-based method

For given desired task motion $\dot{x}_d$, we are able to make the optimization with respect to the i th link system from the assumption that each link system pursues the realization of the desired task motion as follows:

$$\min_{\dot{q}_{d,i}} \frac{1}{2}\left\| \dot{x}_d - J_i(q)\dot{q}_{d,i} \right\|^2 \quad \text{for} \quad i = 1, \cdots, n. \quad (15)$$

Indeed since the above is a simple unconstrained optimization problem, the solution can be easily obtained through following procedures; first, let us take Lagrangian as follows:

$$L_i \triangleq \frac{1}{2}\left(\dot{x}_d - J_i(q)\dot{q}_{d,i}\right)^T \left(\dot{x}_d - J_i(q)\dot{q}_{d,i}\right) \quad (16)$$

for $i = 1, \cdots, n.$ Second, differentiate above with respect to $\dot{q}_{d,i}$ to find the minimum of the Lagrangian:

$$\frac{\partial L_i}{\partial \dot{q}_{d,i}} = -J_i^T(q)\dot{x}_d + J_i^T(q)J_i(q)\dot{q}_{d,i} = 0, \quad (17)$$

then solving above equation provides the minimum solution as follows:

$$\dot{q}_{d,i} = \alpha_i(q)J_i^T(q)\dot{x}_d \quad \text{for} \quad i = 1, \cdots, n, \quad (18)$$

where $\alpha_i(q) = [J_i^T(q)J_i(q)]^{-1} \in \Re$ must be a scalar variable which is called as a scaling factor in the paper. Indeed (18) implies the contribution of i th link system of the manipulator to achieve the desired task motion. The inverse of the scaling factor is always a positive value larger than or equal to 1 as proven in the following equations:

$$\begin{aligned} J_i^T(q)J_i(q) &= [z_i \times (p_e - p_i)]^T[z_i \times (p_e - p_i)] + z_i^T z_i \\ &= \| z_i \times (p_e - p_i) \|^2 + 1 \geq 1 \end{aligned} \quad (19)$$

because of $z_i^T z_i = 1.$ Thus, the scaling factor exists as a positive value within $0 < \alpha_i(q) \leq 1$ in the case of the spatial manipulator.

Remark 1: In addition, we notice that the scaling factor of the partitioned Jacobian must be also positive even in the case of planar manipulator. The partitioned Jacobian of the planar manipulator cannot have zero vector because the direction vector z_i is always perpendicular to the workspace plane of the manipulator. Thus the scaling factor $\alpha_i(q)$ always exists as the positive value in the case of the planar manipulator.

Finally let us collect (18) according to the sequence of $i = 1, \cdots, n,$ then we have

$$\begin{bmatrix} \dot{q}_{d,1} \\ \vdots \\ \dot{q}_{d,n} \end{bmatrix} = \begin{bmatrix} \alpha_1(q) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \alpha_n(q) \end{bmatrix} \begin{bmatrix} J_1^T(q) \\ \vdots \\ J_n^T(q) \end{bmatrix} \dot{x}_d(t),$$

$$\dot{q}_d(t) = D(q)J^T(q)\dot{x}_d(t), \quad (20)$$

where

$$D(q) \triangleq \begin{bmatrix} \alpha_1(q) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \alpha_n(q) \end{bmatrix} \in \Re^{n\times n},$$

$$J(q) = [J_1(q) \,|\, J_2(q) \,|\cdots|\, J_n(q)] \in \Re^{6\times n}$$

in which $D(q)$ is the scaling matrix having positive diagonal elements. Now if the CLIK is applied to (20), then we have

$$\dot{q}_r(t) = D(q)J^T(q)\{\dot{x}_d(t) + K[x_d(t) - x(t-\Delta t)]\}, \quad (21)$$

$$q_r(t) = q_r(t-\Delta t) + \dot{q}_r(t)\Delta t. \quad (22)$$

This is referred to as a 'scaled Jacobian transpose based (SJT-based) method' in this paper because it becomes equal to the JT-based method except the multiplication of the scaling matrix $D(q)$. The SJT-based method does not contain the kinematic singularities because the positive scaling matrix always exists. The next section discusses about the convergence property of the task error of the SJT-based method.

3.2. Convergence of task error

For given SJT-based method of (21), let us apply the Lyapunov indirect method to examine the property of the error convergence. First take the Lyapunov function candidate as follows:

$$V = \frac{1}{2}e^T Ke > 0, \quad (23)$$

where K is the gain matrix having positive diagonal elements and $e = x_d - x$ is the error vector. Second if we take the time derivative using the forward kinematics $\dot{x} = J(q)\dot{q}_r$ and (21), then we have

$$\begin{aligned} \dot{V} &= e^T K\dot{e} \\ &= e^T K[\dot{x}_d - J\dot{q}_r] \\ &= e^T K[I - JDJ^T]\dot{x}_d(t) - e^T KJDJ^T K[x_d(t) - x(t-\Delta t)] \\ &= e^T K[I - JDJ^T]\dot{x}_d - e^T KJDJ^T Ke + e^T KJDJ^T K\delta, \end{aligned} \quad (24)$$

where we should note that all the terms contain the matrix $J(q)D(q)J^T(q)$. This common term has the following property:

$$\begin{aligned} J(q)D(q)J(q)^T &= \alpha_1 J_1 J_1^T + \cdots + \alpha_n J_n J_n^T \\ &= \sum_{i=1}^{n} \alpha_i J_i J_i^T \geq 0. \end{aligned} \quad (25)$$

Here $\alpha_i J_i J_i^T = (J_i^T J_i)^{-1} J_i J_i^T \in \Re^{6\times 6}$ is a rank-one matrix, for $i = 1, 2\cdots, n$. In addition, since $\sqrt{\alpha_i} J_i$ must be a normalized column vector, we can define the matrix M composed of the normalized column vectors as follows:

$$M = \left[\sqrt{\alpha_1} J_1 |\cdots| \sqrt{\alpha_n} J_n\right] = J\sqrt{D} \in \Re^{6\times n}. \quad (26)$$

Now Naimark's Theorem in [14] gives us some insight to provide the sufficient condition for the convergence. If we can find a $(n-6)\times n$ matrix N so that the following matrix can be $n\times n$ unitary:

$$H = \begin{bmatrix} M \\ N \end{bmatrix} \in \Re^{n\times n}, \quad (27)$$

then the non-square matrix M is an isometry, i.e., $MM^T = I$, in other words.

$$JDJ^T = I. \quad (28)$$

In the above if the sufficient condition of Naimark's Theorem is satisfied, the time derivative of Lyapunov function candidate of (24) is reduced to:

$$\begin{aligned} \dot{V} &= -e^T K^2 e + e^T K^2 \delta \\ &= -e^T K^2 e + \frac{1}{2\gamma^2} e^T K^2 e - \frac{1}{2}\left\| \frac{1}{\gamma} Ke - \gamma K\delta \right\|^2 \\ &\quad + \frac{\gamma^2}{2}\delta^T K^2 \delta \\ &\leq -\left(1 - \frac{1}{2\gamma^2}\right) e^T K^2 e + \frac{\gamma^2}{2}\delta^T K^2 \delta. \end{aligned} \quad (29)$$

Here since an arbitrary constant γ can be chosen as $\gamma > 1/\sqrt{2}$, we can conclude that the task error of the SJT-based method is bounded or goes to zero only if the disturbance is bounded or goes to zero under the sufficient condition of Naimark's Theorem. Since the above sufficient condition of Naimark's Theorem implies that the $\{\sqrt{\alpha_i} J_i\}$ and $\{\sqrt{\alpha_k} J_k\}$ should be orthogonal each other for different i and k combinations, however, it is difficult to be satisfied for general joint configurations.

Remark 2: Naimark's Theorem might be further simplified in the case of non-redundant robot manipulator, $n = 6$. This property of

$$JDJ^T = \sum_{i=1}^{6} \alpha_i J_i J_i^T = I$$

is always true if and only if $\{\sqrt{\alpha_i} J_i\}$ is orthonormal basis.

On the other hand, if we consider the set-point regulation control, i.e., $\dot{x}_d = 0$ and $x_d = x_s$ as the constant to be regulated, the time derivative of Lyapunov function of (24) is reduced directly to:

$$\dot{V} = -e^T K_2 e + e^T K_2 \delta,$$

where $K_2 \triangleq KJ(q)D(q)J^T(q)K \geq 0$ and even $K_2 > 0$ if the $J(q)$ has full rank. Also we can know that

$$\dot{V} \leq -\left(1 - \frac{1}{2\gamma^2}\right) e^T K_2 e + \frac{\gamma^2}{2}\delta^T K_2 \delta. \quad (30)$$

In the case of the regulation control, we conclude that the task error of the SJT-based method is bounded or goes to zero only if the disturbance is bounded or goes to zero without the sufficient condition of Naimark's Theorem.

In summary if the matrix H of (27) according to Naimark's Theorem can be found to be a unitary matrix, then we can know that the proposed SJT-based method achieves the boundedness of the task error only if the disturbance is bounded. In the case of the regulation control, we arrive at the same conclusion without the sufficient condition of the Naimark's Theorem. The next section suggests a motion ellipsoid analysis in order to show another advantage of the proposed method.

3.3. Motion ellipsoid analysis

For simplicity, the planar two-link manipulator system is considered to compare the differences between motion ellipsoids of the conventional JT-based and the proposed SJT-based. The motion ellipsoid is obtained by using the singular value decomposition (SVD) of the transformation T. For instance, the geometrical interpretation of the SVD of $T = U\Sigma V^T$ is illustrated in Fig. 1. The major and minor axis of the ellipsoid are determined by two singular values σ_{max} and $\sigma_{\min}$.

For given typical Jacobian with unit lengths of the planar two-link system as follows:

$$J(q) = \begin{bmatrix} -\sin(q_1) - \sin(q_1 + q_2) & -\sin(q_1 + q_2) \\ \cos(q_1) + \cos(q_1 + q_2) & \cos(q_1 + q_2) \end{bmatrix}$$

first the JT-based method makes use of the following transformation:

$$T_1 = J^T(q). \tag{31}$$

Now if the input vector of unit magnitude, $|u| = 1$, is applied to the transformation of (31), then the change phases of the motion ellipsoids are depicted as shown in Fig. 2(a). The postures of the manipulator shown in the Fig. 2(a) are numbered from 0 to 4, where 0th posture corresponds to $q = [q_1, q_2]^T = [\frac{\pi}{2}, -\frac{\pi}{2}]^T$ and finally 4th posture corresponds to the singular configuration of $q = [\pi, 0]^T$. As we can see in the Fig. 2(a), the motion ellipsoid approaches the long line segment, in other words, the major axis increases accordingly as the posture approaches near singular configuration. The bigger the major axis, the more sensitive to the inputs. In other words, if the input vector with unit magnitude is aligned with the initial rotation matrix V^T, then the sensitivity of the output increases as the major axis is large. This is one of the disadvantages of the JT-based method.

Second the SJT-based method utilizes the following transformation:

$$T_2 = D(q)J^T(q). \tag{32}$$

Then, the change phases of the motion ellipsoids are obtained as suggested in Fig. 2(b). As we can see in Fig. 2(b), the major axis of the motion ellipsoid keeps a

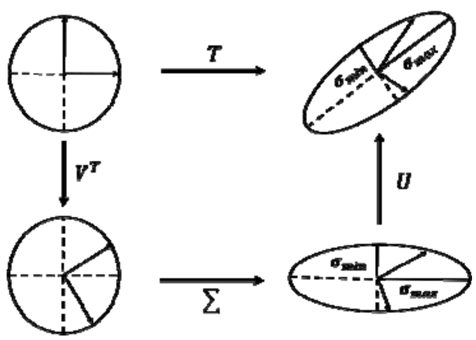
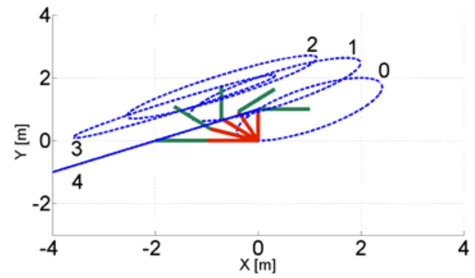
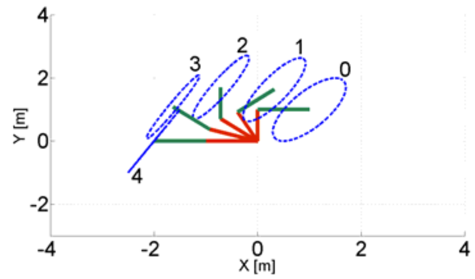


Fig. 1. Geometrical interpretation of singular value decomposition (SVD: $T = U\Sigma V^T$): an initial rotation V^T, scaling Σ along the coordinate axis and a final rotation U.

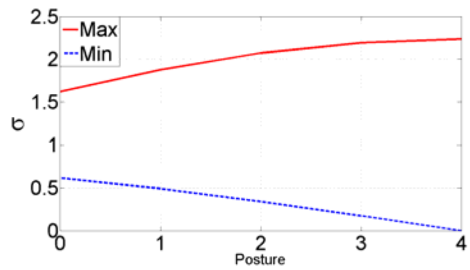


(a) Transformation T_1 of the JT-based.

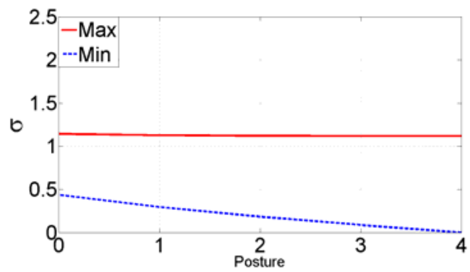


(b) Transformation T_2 of the SJT-based.

Fig. 2. Changes of motion ellipsoids according to the joint configurations.



(a) Transformation T_1 of the JT-based.



(b) Transformation T_2 of the SJT-based.

Fig. 3. Singular values σ_{max} and σ_{min} according to 4 postures, where 4th posture corresponds to the rank deficiency of the Jacobian matrix.

constant value even though the minor axis of the motion ellipsoid becomes zero due to the singularity. Thus we know that the transformation T_2 of the SJT-based method is less sensitive to the input variation. This is one of the advantages of the SJT-based method over the conventional JT-based method.

In detail, the variations of both major and minor axis are suggested in Fig. 3 according to four postures. Although the minor axis denoted by σ_{min} have the similar values between the transformations T_1 and T_2, the major axis denoted by σ_{max} shows big difference between the both as shown in Fig. 3. Indeed the major axis of the transformation T_2 are not nearly changed even though the Jacobian loses rank. Thus we can conclude that the SJT-based method using the transformation T_2 is more robust to the input variations than the conventional JT-based. The next section shows several simulations to verify the robustness of the proposed method.

4. SIMULATION RESULTS

For the simulation studies, let us consider a simple three-link planar manipulator composed of three revolute joints as shown in Fig. 4. The link lengths of the manipulator are set to be $l_1 = 0.3$ m, $l_2 = 0.4$ m, $l_3 =$ 0.28 m and $gripper = 0.2$ m. The simulator was written using the MFC and the OpenGL. Here three kinds of tasks are given as follows. The first task is to track the line trajectory within the workspace as shown in Fig. 5(a). The second task is to track the line trajectory from the inside of the workspace to the outside in order to confirm the robustness against kinematic singularity as shown in Fig. 5(b). The third task is to track the circular trajectory with radius 0.194 m, it is devised to confirm the robustness of the proposed method at near the singularity as shown in Fig. 5(c).

Since the tracking performances of both the conventional JT-based and the proposed SJT-based are affected by the size of the gain matrix K, first the gain matrix $K = kI$ is set to have the same positive diagonal element $k > 0$, second the available gain ranges of k according to the

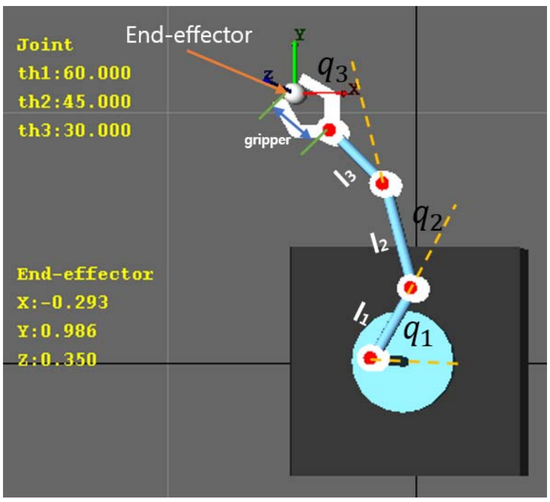
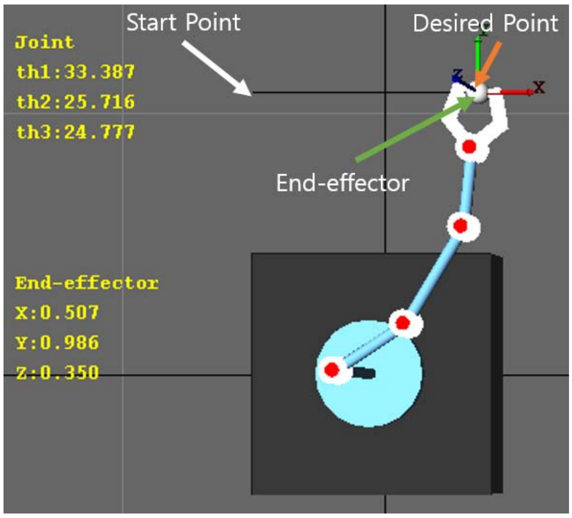
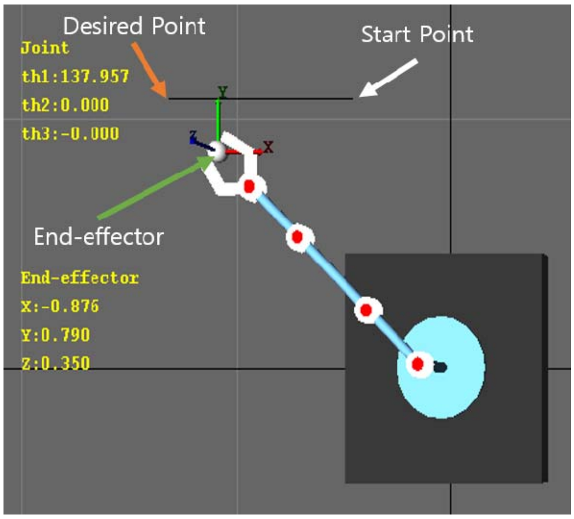


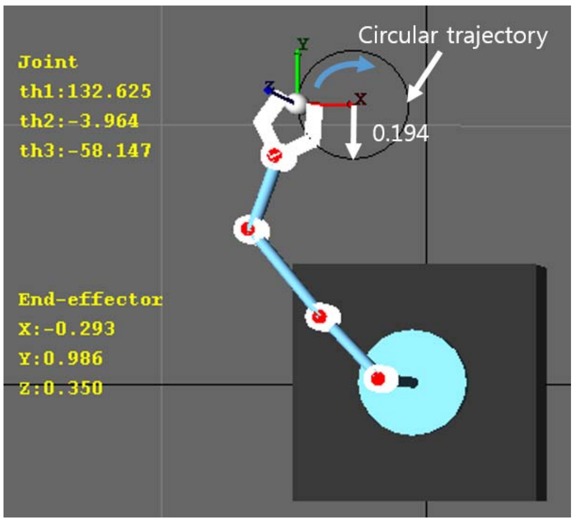
Fig. 4. 3 DoFs planar manipulator.



(a) First task within the workspace, from the start point $x_d(0) = [-0.293, 0.986]^T$ to $x_d(t_f) = [0.507, 0.986]^T$.

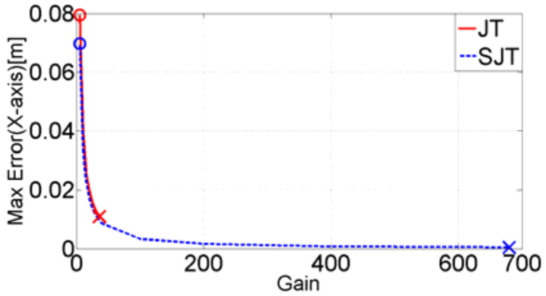


(b) Second task outside the workspace, from $x_d(0) = [-0.293, 0.986]^T$ to $x_d(t_f) = [-1.092, 0.986]^T$.

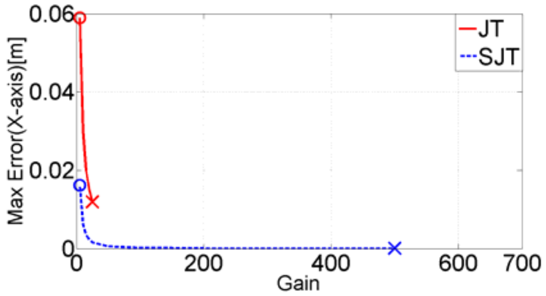


(c) Third task of circular trajectory having nearby singularity.

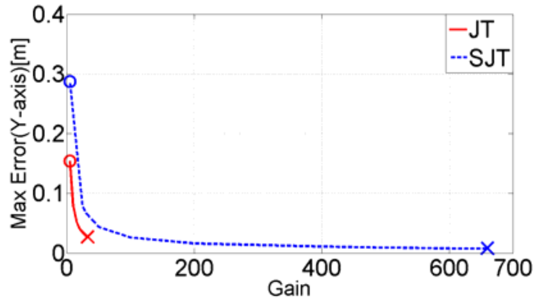
Fig. 5. Three desired tasks to verify the robustness and performance of the proposed SJT-based method.



(a) During the first task execution.



(b) During the second task execution.



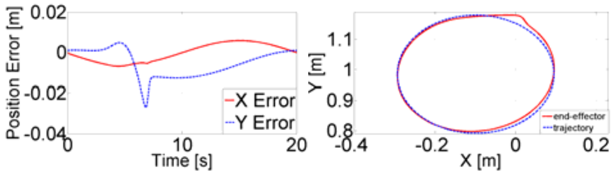
(c) During the third task execution.

Fig. 6. Maximum task errors according to the gain changes.

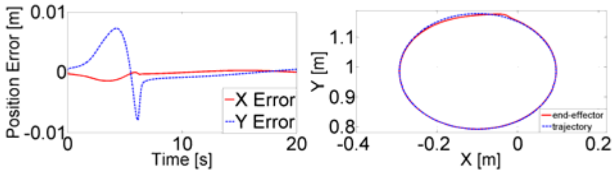
task executions are determined by increasing it until the unstable phenomenon occurs. In Fig. 6(a), the maximum errors during the first task execution were plotted according to the gain changes. In the JT-based method, we could not increase the gain k over 36 due to instability, but the SJT-based method was able to increase the gain k to 680. As proven in Section 3.3, since the major axis of the JT-based becomes very sensitive to the input, we could not increase the gain. However, in the SJT-based method, the gain could be increased enough because it is less sensitive than the JT-based. Thus we could reduce the error greatly by using the SJT-based as shown in Fig. 6(a).

In the second task outside the workspace, the maximum errors were obtained as shown in Fig. 6(b), in which it was plotted as the maximum error before going outside the workspace of the manipulator. The available gain range was 25 in the JT-based, but it was 500 in the SJT-based. Also we could confirm the better tracking performance and robustness of the proposed method through Fig. 6(b).

In the third circular task including the boundary of the singularity, the maximum errors were plotted in Fig. 6(c) according to the changes of the gain. The available gain



(a) JT-based method (k = 33).



(b) SJT-based method (k = 660).

Fig. 7. Position errors and actual joint configurations during third task execution.

range was 33 in the JT-based, but it was 660 in the SJT-based. Also we could confirm the proposed method shows the better tracking performance than the conventional JT-based. In summary, the three tasks could be performed stably within the specific gain ranges, but the SJT-based method was able to bring the better tracking performance.

Finally, the simulation results during third task execution are given in Fig. 7, where the best gains (33 for the JT-based and 660 for the SJT-based) were utilized to compare the position errors of both methods. The maximum errors appear when it passes by a near singularity as we can see in Fig. 7, but the error of the proposed method is reduced to less than half the error of the conventional method.

5. CONCLUSIONS

This paper has presented the SJT-based method for robotic manipulations with two aims; the first is to reduce the task error and the second is to alleviate the singularity phenomena. Clearly the proposed method shows better tracking performance and robustness property against singularity. In addition, the convergence of task error and the motion ellipsoid property were suggested to show these advantages. Finally, a few comparative studies with the conventional method were provided to show the effectiveness of the proposed method through simulations.

REFERENCES

[1] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 722-728, 1985.

[2] I. D. Walker and S. I. Marcus, "Subtask performance by redundancy resolution for redundant robot manipulators," *IEEE Trans. on Robotics and Automation*, vol. 4, no. 3, pp. 350-354, 1988.

[3] T. Ho, C.-G. Kang, and S. Lee, "Efficient closed-form solution of inverse kinematics for a specific six-DOF arm," *International Journal of Control, Automation, and Systems*, vol. 10, no. 3, pp. 567-

573, 2012.

[4] Y. Choi, "Singularity-robust inverse kinematics using Lagrange multiplier for redundant manipulators," *Trans. of the ASME, Journal of Dynamic Systems, Measurement and Control,* vol. 130, no. 5, Article no. 051009, September 2008.

[5] C. de Wit, G. Bastin, and B. Siciliano, *Theory of Robot Control*, Springer, 1996.

[6] L. Sciavicco, B. Siciliano, L. Villani, and G. Oriolo, *Robotics Modelling, Planning and Control*, Springer, 2009.

[7] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. on Robotics and Automation*, vol. 13, no. 3, pp. 398-410, 1997.

[8] R. H. Taylor, "Planning and execution of straight-line manipulator trajectories," *IBM Journal of Research and Developments*, vol. 23, pp. 424-436, 1979.

[9] L. Sciavicco and B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," *IEEE Journal of Robotics and Automation*, vol. 4, no. 4, pp. 403-410, August 1988.

[10] W. A. Wolovich and H. Elliott, "A computational technique for inverse kinematics," *Proc. 3rd IEEE Conference on Decision and Control*, pp. 1359-1363, 1984.

[11] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *Trans. of the ASME, Journal of Dynamic Systems, Measurement and Control*, vol. 108, pp. 163-171, September 1986.

[12] C. W. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped leastsquares method," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 93-101, 1986.

[13] A. Turretta, G. Casalino, and A. Sorbara, "Distributed control architecture for self-reconfigurable manipulators," *International Journal of Robotics Research*, vol. 27, no. 3-4, pp. 481-504, 2008.

[14] http://en.wikipedia.org/

An Yong Lee received his B.S. degree in Mechanical and Control Instrumentation from Sunmoon University, Korea in 2006, and his M.S. degree in Mechatronics Engineering from Hanyang University, Korea in 2008. He is working toward a Ph.D. degree at Hanyang University. His research interests include manipulator control and trajectory planning.

Jongguk Yim received his B.S. and M.S. degrees in Precision Mechanical Engineering from Hanyang University, Seoul, Korea, in 1994 and 1996, respectively, and his Ph.D. degree in Mechanical Engineering from Hanyang University, Seoul, Korea in 2013. Now he is a post doctoral fellow in Research Institute of Engineering and Technology of Hanyang University since 2013. His research interests include nonlinear control, robust control, and robotics.

Youngjin Choi received his B.S. degree in Precision Mechanical Engineering from Hanyang University, Seoul, Korea, in 1994, and his M.S. and Ph.D. degrees in Mechanical Engineering from POSTECH, Pohang, Korea, in 1996 and 2002, respectively. Since 2005, he is an associate professor at the department of Electronic Systems Engineering of Hanyang University, Ansan, Korea. From 2002 to 2005, he was a senior research scientist at the intelligent robotics research center of the Korea Institute of Science and Technology (KIST). From 2011 to 2012, he was a visiting researcher at the University of Central Florida (UCF), USA. From 2010 to 2014, he was the associate editor of the IEEE Transactions on Robotics. His research interests include biorobotics, control theory, humanoid walking, rehabilitation robot, and dual-arm manipulation.