Prioritized Inverse Kinematics using QR and Cholesky Decompositions

Sang-ik An[1] and Dongheui Lee[2]

Abstract— **This paper proposes new methods for the prioritized inverse kinematics (PIK) by using the QR decomposition (QRD) and the Cholesky decomposition (CLD) on the purpose of separation between orthogonalization and inversion processes that are essential parts of the PIK. The distinctive approach eliminates the interference between two processes which usually induces inaccuracy and sometimes instability on the prioritized inverse solutions. Two degenerate properties of using the QRD are explained and the remedies are provided as the modified damped least-squares pseudoinverse and the numerical reconditioning. The effectiveness are examined by the kinematic simulations with the *n*-link manipulators in the two-dimensional case and the KUKA LWR in the three-dimensional case.**

I. INTRODUCTION

There are two well-known non-iterative solutions for the prioritized inverse kinematics with an arbitrary number of tasks. The first method was proposed by Nakamura [1] and Maciejewski [2], and generalized to the arbitrary number of tasks by Siciliano and Slotine [3]. In this method, the subsequent null space motions are determined to minimize the task errors of the lower priorities. Therefore, the tasks are performed accurately, while the large joint velocities are required in the vicinity of the kinematic and algorithmic singularities. To remedy this problem, the damped least-squares pseudoinverse (DLPI) was introduced by Nakamura [4] and Wampler [5] at the expenses of the task errors. The second method has been suggested by Chiaverini [6] and generalized by Baerlocher [7]. This method directly projects the joint motions of the lower-priority tasks to the null space of the higher-priority tasks, such that the solutions gradually reduce the null-space velocities near the algorithmic singularities. However, it suffers from the inaccuracy of the solutions for the lower-priority tasks, so the closed-loop inverse kinematics scheme (CLIK) developed by Chiacchio *et al.* [8] was utilized together. Later, Park [9] and Choi [10] have solved the inaccuracy problem of the Chiaverini's method by using the weighted pseudoinverse with which the joint space is transformed to the orthogonal space of tasks.

In spite of their successive advances, there are still several difficulties when using the proposed methods, especially if the number of tasks grows. The first difficulty is that the consecutive projections still generate task errors of the lower priorities because the DLPI is usually not an option in the practical cases to prevent the drastic increase of the joint velocities near the unexpected singularities. The second difficulty is on the determination of the constants for the performance and stability of the algorithms where it's better to have less tunable gains if possible. The third difficulty is the computational complexity that might be a burden of the implementation with the limited resources in the real time applications. Recently, the conventional problem has been revisited by using the QRD to improve the computational efficiency [11][12].

In this paper, we are proposing new methods for the PIK by using QR and Cholesky decompositions with which the QRD is used for the different purpose which is to separate the orthogonalization and the inversion processes of the PIK. Our approach is to orthogonalize tasks first, then to find the inverse solutions later, while the previous approaches perform both simultaneously. The greatest advantage of seperating two processes is that the orthogonalization process is not interrupted by the stabilization techniques when calculating the inverse solutions. Additionally, the blockwise matrix computations are possible for the time-efficient implementation. We will also show the numerical reconditioning method to improve the performance when the tasks are extremely many. A comparative study to the conventional methods will be given by the kinematic simulations with the *n*-link manipulators and the KUKA LWR.

The background of the PIK will be provided in section II and the proposed algorithm will be explained in section III. The comparison results of the kinematic simulation will be shown in section IV and finally the concluding remarks will be in section V.

II. BACKGROUND

A. Inverse Kinematics

The forward and inverse kinematics are to find the mapping between the joint space $\Omega_q \subset \mathbb{R}^n$ and the task space $\Omega_x \subset \mathbb{R}^m$ of mechanisms. The joint variable $\mathbf{q} \in \Omega_q$ is given by the generalized coordinate and the task variable $\mathbf{x} \in \Omega_x$ is defined from the tasks required. Then the forward kinematics can be written as:

$$\mathbf{x} = \mathbf{f}(\mathbf{q}) \tag{1}$$

where $\mathbf{f} : \Omega_q \to \Omega_x$ are usually surjective and nonlinear, thus to find the inverse of (1) is difficult. Therefore, the resolved motion rate control has been introduced by Whitney [13] with which the kinematic problems are solved in the velocity level

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

where $\dot{\bullet}$ is the time derivative of $\bullet$, $\mathbf{J} \triangleq \partial \mathbf{f}/\partial \mathbf{q} \in \mathbb{R}^{m\times n}$ is the Jacobian of $\mathbf{f}$ with respect to $\mathbf{q}$, and usually $r \triangleq \text{rank}(\mathbf{J}) \leq m \leq n$ is assumed.

[1]S. An and [2]D. Lee are with the Institute of Automatic Control Engineering, Technische Universität München, D-80290 Munich, Germany.S.An.Robots@gmail.com and dhlee@tum.de

When finding the inverse of (2), there are two important issues, singularity and redundancy. If the Jacobian is square and has full rank ($r = m = n$), the unique solution is

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\dot{\mathbf{x}} \tag{3}$$

, while, in other cases, the mechanism has singularity ($r < m$) or redundancy ($m < n$), such that the inverse solution doesn't exist or infinite solutions exist. Then, the simplest strategy could be to minimise the task error $\|\dot{\mathbf{e}}\| \triangleq \|\dot{\mathbf{x}} - \mathbf{J}\dot{\mathbf{q}}\|$ when there is singularity or the joint velocity $\|\dot{\mathbf{q}}\|$ when there is redundancy where $\|\bullet\|$ is the Euclidean norm of $\bullet$. In both cases, the inverse solution is given by

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}(\mathbf{q})\dot{\mathbf{x}} \tag{4}$$

where $\mathbf{J}^{\dagger} \triangleq \mathbf{V}\Sigma^{\dagger}\mathbf{U}^T \in \mathbb{R}^{n\times m}$ is the Moore-Penrose pseudoinverse [14] defined from the singular value decomposition (SVD) $\mathbf{J} = \mathbf{U}\Sigma\mathbf{V}^T$ and $\Sigma^{\dagger} \triangleq \mathrm{diag}(1/\sigma_i) \in \mathbb{R}^{n\times m}$ is the rectangular diagonal matrix composed of the non-zero singular values $\{\sigma_1, \cdots, \sigma_r\}$.

Whenever the Jacobian is rank deficient, the mechanism can not generate the motion in some directions, such that the inverse solutions tend to increase drastically near the singularities. Nakamura [4] and Wampler [5] have introduced the DLPI to the robotics area with which the sum of the task error and the joint velocity $\|\dot{\mathbf{x}} - \mathbf{J}\dot{\mathbf{q}}\|^2 + \lambda^2\|\dot{\mathbf{q}}\|^2$ is minimized

$$\dot{\mathbf{q}} = \mathbf{J}^{*}(\mathbf{q})\dot{\mathbf{x}} \tag{5}$$

where $\mathbf{J}^{*} \triangleq \mathbf{J}^T(\mathbf{J}\mathbf{J}^T + \lambda^2\mathbf{I}_m)^{-1} \in \mathbb{R}^{n\times m}$ is the DLPI of $\mathbf{J}$, $\mathbf{I}_m \in \mathbb{R}^{m\times m}$ is the identity matrix, and the constant $\lambda \in \mathbb{R}$ is a compromise between the accuracy of the task execution and the boundedness of the solutions.

A drawback of solving the inverse kinematics in the velocity level is that the solutions (3), (4), and (5) don't guarantee the task consistency in the position level given by (1). To overcome this problem, Chiacchio *et al.* [8] have proposed the CLIK in which the position-level task errors are compensated

$$\dot{\mathbf{q}} = \mathbf{J}^{\times}(\mathbf{q})\,(\dot{\mathbf{x}} + \mathbf{K}\mathbf{e}) \tag{6}$$

where $\mathbf{K} \triangleq \mathrm{diag}(k_i) \in \mathbb{R}^{m\times m}$ is the feedback gain, $\mathbf{e} \triangleq \mathbf{x} - \mathbf{f}(\mathbf{q}) \in \mathbb{R}^m$ is the position-level task error, and $\mathbf{J}^{\times}$ is any of $\mathbf{J}^T, \mathbf{J}^{-1}, \mathbf{J}^{\dagger}$, and $\mathbf{J}^{*}$.

B. Prioritized Inverse Kinematics

The forward and inverse kinematics can be defined more generally with k-tasks in order of higher to lower priorities

$$\begin{aligned}\dot{\mathbf{x}}_1 &= \mathbf{J}_1(\mathbf{q})\dot{\mathbf{q}}\\ \dot{\mathbf{x}}_2 &= \mathbf{J}_2(\mathbf{q})\dot{\mathbf{q}}\\ &\vdots\\ \dot{\mathbf{x}}_k &= \mathbf{J}_k(\mathbf{q})\dot{\mathbf{q}}\end{aligned} \tag{7}$$

where $\dot{\mathbf{x}}_i \in \mathbb{R}^{m_i}$ and $\mathbf{J}_i \in \mathbb{R}^{m_i\times n}$ are the velocity and the Jacobian of the i-th task, respectively. It can be compactly expressed in the form of (2) by defining $\dot{\mathbf{x}} \triangleq \begin{bmatrix}\dot{\mathbf{x}}_1^T & \cdots & \dot{\mathbf{x}}_k^T\end{bmatrix}^T \in \mathbb{R}^m$, $\mathbf{J} \triangleq \begin{bmatrix}\mathbf{J}_1^T & \cdots & \mathbf{J}_k^T\end{bmatrix}^T \in \mathbb{R}^{m\times n}$, and $m \triangleq \sum m_i$. From now, we assume that (2) and (7) denote the same equation.

Definition 1 (Task and Algorithmic Singularities)**.** Let's define $r_i \triangleq \mathrm{rank}(\mathbf{J}_i)$, then we can say that the i-th task has a task singularity if $r_i < m_i$ and the i-th and j-th tasks have an algorithmic singularity if $\mathrm{rank}(\begin{bmatrix}\mathbf{J}_i^T & \mathbf{J}_j^T\end{bmatrix}^T) < r_i + r_j$.

Definition 2 (Prioritized Inverse Kinematics)**.** Given the ordered tasks (7), the PIK is to find the joint velocity $\dot{\mathbf{q}}$ that minimizes the task error, $\|\dot{\mathbf{e}}_i\| \triangleq \|\dot{\mathbf{x}}_i - \mathbf{J}_i\dot{\mathbf{q}}\|$, on the condition that the minimum task errors of the higher priority, $\|\dot{\mathbf{e}}_j\|$, $1 \le j < i$, are not changed.

If (7) has the unique solution (3), then it is also the unique solution of the PIK. However, if not, (4) and (5) cannot be the solution of the PIK because they minimize the total task error $\|\dot{\mathbf{e}}\|$, therefore the error is distributed throughout the whole tasks. Two well-known non-iterative solutions for the PIK are shown in the next two propositions.

Proposition 1 (Nakamura and Maciejewski's Method)**.** *The recursive form* (8) *and* (9) *can be a solution of the PIK.*

$$\dot{\mathbf{q}}_i = \dot{\mathbf{q}}_{i-1} + (\mathbf{J}_i\mathbf{N}_{i-1})^{\dagger}(\dot{\mathbf{x}}_i - \mathbf{J}_i\dot{\mathbf{q}}_{i-1}), \qquad \dot{\mathbf{q}}_0 = \mathbf{0} \tag{8}$$

$$\mathbf{N}_i = \mathbf{N}_{i-1} - (\mathbf{J}_i\mathbf{N}_{i-1})^{\dagger}(\mathbf{J}_i\mathbf{N}_{i-1}), \qquad \mathbf{N}_0 = \mathbf{I}_n \tag{9}$$

Proof. See [1], [2], and [3]. □

Proposition 2 (Chiaverini, Park, and Choi's Method)**.** *The recursive form* (10) *and* (11) *can be a solution of the PIK*

$$\dot{\mathbf{q}}_i = \dot{\mathbf{q}}_{i-1} + \mathbf{N}^W_{i-1}\mathbf{J}_i^{W\dagger}\dot{\mathbf{x}}_i, \qquad \dot{\mathbf{q}}_0 = \mathbf{0} \tag{10}$$

$$\mathbf{N}^W_i = \mathbf{N}^W_{i-1} - \mathbf{N}^W_{i-1}\mathbf{J}_i^{W\dagger}\mathbf{J}_i, \qquad \mathbf{N}^W_0 = \mathbf{I}_n \tag{11}$$

where $\mathbf{J}_i^{W\dagger} \triangleq \mathbf{W}^{\dagger}\mathbf{J}_i^T(\mathbf{J}_i\mathbf{W}^{\dagger}\mathbf{J}_i^T)^{\dagger} \in \mathbb{R}^{n\times m_i}$ *and* $0 \le \mathbf{W} \triangleq \mathbf{J}^T\mathbf{J} \in \mathbb{R}^{n\times n}$.

Proof. See [6], [7], [9], and [10]. □

III. Prioritized Inverse Kinematics using QR and Cholesky Decompositions

As we discussed, the inverse solutions of both methods require significant joint velocities near the singular positions, therefore the DLPI should replace the pseudoinverse. For the first method, $(\mathbf{J}_i\mathbf{N}_{i-1})^{\dagger}$ is replaced with $(\mathbf{J}_i\mathbf{N}_{i-1})^{*}$ and, for the second method, $\mathbf{J}_i^{W\dagger}$ is replaced with $\mathbf{J}_i^{W*} \triangleq \mathbf{W}^{\diamond}\mathbf{J}_i^T(\mathbf{J}_i\mathbf{W}^{\diamond}\mathbf{J}_i^T)^{*}$ where $\mathbf{W}^{\diamond} \triangleq (\mathbf{J}^T\mathbf{J} + \delta^2\mathbf{I}_n)^{-1}$. Usually, the damping constant λ and δ are determined differently. However, once we adopt the DLPI, the null-space projection and the orthogonalization between tasks become imperfect, such that there should be interference between tasks. Therefore, the sources of the task error are not only the imperfect inversions but also the imperfect projections by the DLPI. This problem stems from the fact that both methods perform the orthgonalization and the inversion processes simultaneously.

Our strategy is to perform the orthogonalization of tasks before calculating the inverse solutions. It becomes possible thanks to the QRD of the Jacobian transpose. After the QRD is accomplished, the inversion process can be performed in the manner of the blockwise matrix computations. Therefore, the orthogonalization process is not interrupted by the inversion process and consequently the task errors can be reduced.

A. Basic Idea

The QRD of the Jacobian transpose without the pivoting is given by

$$\mathbf{J}^T = \mathbf{Q}\mathbf{R} = \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}\mathbf{R}_1 \\ \mathbf{0}\end{bmatrix} = \mathbf{Q}_1\mathbf{R}_1 \triangleq \hat{\mathbf{J}}^T\mathbf{C}^T$$

where $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_n$, $\mathbf{Q}_1 = \hat{\mathbf{J}}^T \in \mathbb{R}^{n\times m}$, $\mathbf{R} \in \mathbb{R}^{n\times m}$, and $\mathbf{R}_1 = \mathbf{C}^T \in \mathbb{R}^{m\times m}$ is an upper triangular matrix. Then, the forward kinematics can be rewritten as:

$$\dot{\mathbf{x}} = \mathbf{C}(\mathbf{q})\hat{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} \tag{12}$$

or

$$\begin{aligned}
\dot{\mathbf{x}}_1 &= \mathbf{C}_{11}^A\hat{\mathbf{J}}_1^A\dot{\mathbf{q}} = (\mathbf{C}_{11}\hat{\mathbf{J}}_1)\dot{\mathbf{q}} \\
\dot{\mathbf{x}}_2 &= \mathbf{C}_{22}^A\hat{\mathbf{J}}_2^A\dot{\mathbf{q}} = (\mathbf{C}_{21}\hat{\mathbf{J}}_1 + \mathbf{C}_{22}\hat{\mathbf{J}}_2)\dot{\mathbf{q}} \\
&\vdots \\
\dot{\mathbf{x}}_k &= \mathbf{C}_{kk}^A\hat{\mathbf{J}}_k^A\dot{\mathbf{q}} = (\mathbf{C}_{k1}\hat{\mathbf{J}}_1 + \cdots + \mathbf{C}_{kk}\hat{\mathbf{J}}_k)\dot{\mathbf{q}}
\end{aligned} \tag{13}$$

where $\mathbf{C} = \begin{bmatrix}\mathbf{C}_{11} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{k1} & \mathbf{C}_{k2} & \dots & \mathbf{C}_{kk}\end{bmatrix}$, $\hat{\mathbf{J}} = \begin{bmatrix}\hat{\mathbf{J}}_1 \\ \hat{\mathbf{J}}_2 \\ \vdots \\ \hat{\mathbf{J}}_k\end{bmatrix}$, $\mathbf{C}_{ij} \in \mathbb{R}^{m_i\times m_j}$, $\hat{\mathbf{J}}_i \in \mathbb{R}^{m_i\times n}$, $\mathbf{C}_{ij}^A = \begin{bmatrix}\mathbf{C}_{i1} & \cdots & \mathbf{C}_{ij}\end{bmatrix}$, and $\hat{\mathbf{J}}_i^A = \begin{bmatrix}\hat{\mathbf{J}}_1^T & \cdots & \hat{\mathbf{J}}_i^T\end{bmatrix}^T$. The lower triangular matrix $\mathbf{C}$ can be divided into two parts

$$\mathbf{C} = \mathbf{C}_D + \mathbf{C}_L \tag{14}$$

where $\mathbf{C}_D = \mathrm{diag}(\mathbf{C}_{ii})$ is a block diagonal matrix and $\mathbf{C}_L = \mathbf{C} - \mathbf{C}_D$ is a strictly lower triangular matrix.

Proposition 3 (Proposed Method I (Recursive Form))**.** *The recursive form* (15) *can be a solution of the PIK.*

$$\dot{\mathbf{q}}_i = \dot{\mathbf{q}}_{i-1} + \hat{\mathbf{J}}_i^T\mathbf{C}_{ii}^\dagger(\dot{\mathbf{x}}_i - \mathbf{C}_{i,i-1}^A\hat{\mathbf{J}}_{i-1}^A\dot{\mathbf{q}}_{i-1}), \qquad \dot{\mathbf{q}}_0 = \mathbf{0} \tag{15}$$

Proof. From (13), the i-th task can be written by $\dot{\mathbf{x}}_i = (\mathbf{C}_{i,i-1}^A\hat{\mathbf{J}}_{i-1}^A + \mathbf{C}_{ii}\hat{\mathbf{J}}_i)\dot{\mathbf{q}}$. Let's define $\dot{\mathbf{q}} \triangleq \dot{\mathbf{q}}^- + \dot{\mathbf{q}}^\circ + \dot{\mathbf{q}}^+$ that has properties for preserving the task priority

$$\hat{\mathbf{J}}_{i-1}^A\dot{\mathbf{q}}^\circ = \mathbf{0}, \qquad\qquad \hat{\mathbf{J}}_i^A\dot{\mathbf{q}}^+ = \mathbf{0} \tag{16}$$

, then the i-th task error can be written by

$$\dot{\mathbf{e}}_i = \dot{\mathbf{x}}_i - \mathbf{C}_{ii}^A\hat{\mathbf{J}}_i^A\dot{\mathbf{q}}^- - \mathbf{C}_{ii}\hat{\mathbf{J}}_i\dot{\mathbf{q}}^\circ$$

and the minimum norm solution of the i-th task can be obtained by

$$\dot{\mathbf{q}}^\circ = \hat{\mathbf{J}}_i^T\mathbf{C}_{ii}^\dagger(\dot{\mathbf{x}}_i - \mathbf{C}_{ii}^A\hat{\mathbf{J}}_i^A\dot{\mathbf{q}}^-) \tag{17}$$

using $(\mathbf{C}_{ii}\hat{\mathbf{J}}_i)^\dagger = \hat{\mathbf{J}}_i^T\mathbf{C}_{ii}^\dagger$. Since (17) satisfies (16), if we define $\dot{\mathbf{q}}_{i-1} \triangleq \dot{\mathbf{q}}^-$ and $\dot{\mathbf{q}}_i \triangleq \dot{\mathbf{q}}^- + \dot{\mathbf{q}}^\circ$, we get $\dot{\mathbf{q}}_{i-1} \in \mathrm{span}\{\hat{\mathbf{J}}_1^T, \cdots, \hat{\mathbf{J}}_{i-1}^T\}$ and the recursive form (15) can be a solution of the PIK. □

Theorem 1 (Proposed Method I (Closed Form))**.** *The closed form* (18) *can be a solution of the PIK*

$$\dot{\mathbf{q}} = \hat{\mathbf{J}}^T(\mathbf{I}_m + \mathbf{C}_D^\ddagger\mathbf{C}_L)^{-1}\mathbf{C}_D^\ddagger\dot{\mathbf{x}} \tag{18}$$

where $\mathbf{C}_D^\ddagger \triangleq \mathrm{diag}(\mathbf{C}_{ii}^\dagger) \in \mathbb{R}^{m\times m}$.

Proof. The equation (12) can be modified as:

$$\mathbf{C}_D\hat{\mathbf{J}}\dot{\mathbf{q}} = \dot{\mathbf{x}} - \mathbf{C}_L\hat{\mathbf{J}}\dot{\mathbf{q}}$$

by using (14), then the minimum norm solution becomes

$$\hat{\mathbf{J}}\dot{\mathbf{q}} = \mathbf{C}_D^\ddagger\dot{\mathbf{x}} - \mathbf{C}_D^\ddagger\mathbf{C}_L\hat{\mathbf{J}}\dot{\mathbf{q}}. \tag{19}$$

If we inspect the block diagonal matrix $\mathbf{C}_D$ and the strictly lower triangular matrix $\mathbf{C}_L$, (19) represents the forward substitution, therefore the process can be repeated preserving the task priority.

$$\begin{aligned}
\hat{\mathbf{J}}\dot{\mathbf{q}} &= \mathbf{C}_D^\ddagger\dot{\mathbf{x}} - \mathbf{C}_D^\ddagger\mathbf{C}_L\hat{\mathbf{J}}\dot{\mathbf{q}} \\
&= (\mathbf{I}_m - \mathbf{C}_D^\ddagger\mathbf{C}_L)\mathbf{C}_D^\ddagger\dot{\mathbf{x}} + (\mathbf{C}_D^\ddagger\mathbf{C}_L)^2\hat{\mathbf{J}}\dot{\mathbf{q}} \\
&\vdots \\
&= \left[\mathbf{I}_m - \mathbf{C}_D^\ddagger\mathbf{C}_L + \cdots + (-\mathbf{C}^\ddagger\mathbf{C}_L)^{l-1}\right]\mathbf{C}_D^\ddagger\dot{\mathbf{x}} \\
&\quad + (-\mathbf{C}_D^\ddagger\mathbf{C}_L)^l\hat{\mathbf{J}}\dot{\mathbf{q}}
\end{aligned}$$

Let's define $\mathbf{D} \triangleq \mathbf{C}_D^\ddagger\mathbf{C}_L$, then

$$\mathbf{D} = \begin{bmatrix}\mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_{22}^\dagger\mathbf{C}_{21} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{C}_{kk}^\dagger\mathbf{C}_{k1} & \mathbf{C}_{kk}^\dagger\mathbf{C}_{k2} & \cdots & \mathbf{C}_{kk}^\dagger\mathbf{C}_{k,k-1} & \mathbf{0}\end{bmatrix}$$

is the strictly lower triangular matrix, therefore it's a nilpotent matrix that has properties $\mathbf{D}^l = \mathbf{0}$ for some positive integer $l \leq k$ and $\mathbf{I}_m + \mathbf{D}$ is invertible, such that

$$(\mathbf{I}_m + \mathbf{D})^{-1} = \mathbf{I}_m - \mathbf{D} + \mathbf{D}^2 - \mathbf{D}^3 + \cdots.$$

Therefore, the process finishes after l repetitions and the result can be compactly expressed in the closed form (18). □

The proposed methods (15) and (18) don't require to calculate the null-space projection matrix nor the orthogonalization process during the formulation of the inverse solutions. Therefore, even if we replace the pseudoinverse to the DLPI, orthogonality between tasks is not affected.

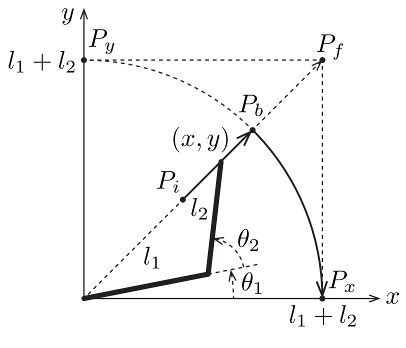


Fig. 1: Two-link manipulator with the desired trajectory of the end-effector moving from P_i to P_f.

B. Damped Least-Squares Pseudoinverse with Varing λ

A degenerate case of considering the task priority occurs when a higher-priority task has both the task and algorithmic singularities, such that higher- and lower-priority tasks need to be damped together at the same time. The difficulty arises from the fact that the small variation in the higher-priority task near the singular configuration can cause large fluctuations in the orthogonalization process, such that the inverse solution tends to be unstable and the large damping gain of the DLPI is demanded to stabilize the solution.

A simple example can be a two-link manipulator that is required to reach to the outside of the workspace as shown in Figure 1 while preserving the task priority that the x-directional motion of the end-effector is higher than y. The expected traveling path is composed of two parts, $\mathscr{P}_1 : P_i \rightarrow P_b$ and $\mathscr{P}_2 : P_b \rightarrow P_x$. While it follows $\mathscr{P}_1$, the manipulator has its full degree-of-freedom (DOF), so both tasks can be performed exactly. However, once it reaches to the boundary ($\theta_2 = 0$), it loses a DOF in the direction normal to the boundary, therefore the inverse solution preserving the task priority should make the effector travel on the boundary toward P_x. The degenerate case can be found near P_x where $\theta \triangleq \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix}^T \approx \mathbf{0}$ and the Jacobian matrix can be approximated as:

$$\mathbf{J}(\theta) \approx \begin{bmatrix} \tilde{\mathbf{J}}_1 \\ \tilde{\mathbf{J}}_2 \end{bmatrix} = \begin{bmatrix} -(l_1+l_2)\theta_1 - l_2\theta_2 & -l_2\theta_1 - l_2\theta_2 \\ l_1+l_2 & l_2 \end{bmatrix}. \quad (20)$$

In this configuration, the primary task has a task singularity because $\tilde{\mathbf{J}}_1 \approx \mathbf{0}$ and also an algorithmic singularity can occur when $|\theta_2| \ll |\theta_1|$ because $\tilde{\mathbf{J}}_1 \approx -\tilde{\mathbf{J}}_2\theta_1$. Therefore, the small variation of θ can change the direction of $\tilde{\mathbf{J}}_1$ largely, then the orthogonalization between two tasks can also be fluctuated inducing instability in the inverse solution.

The proposed method is more vulnerable to this problem compared to the conventional methods because it preserves orthogonality between tasks more accurately. It can be checked from the solutions of the PIK. Let's define the desired trajectory of two tasks using the CLIK

$$\begin{aligned} \dot{x}_d &= \dot{x} + k_x(x - f_x(\theta)) \\ \dot{y}_d &= \dot{y} + k_y(y - f_y(\theta)) \end{aligned}$$

and perform the QRD of $\mathbf{J}^T$ by using the Gram-Schmidt orthogonalization

$$\begin{aligned} \mathbf{J}(\theta) &= \begin{bmatrix} c_{11} & 0 \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{J}}_1 \\ \hat{\mathbf{J}}_2 \end{bmatrix} \\ &= \begin{bmatrix} \sqrt{\mathbf{J}_1\mathbf{J}_1^T} & 0 \\ \mathbf{J}_1\mathbf{J}_2^T/\sqrt{\mathbf{J}_1\mathbf{J}_1^T} & \sqrt{\mathbf{J}_2\mathbf{N}_1^2\mathbf{J}_2^T} \end{bmatrix} \begin{bmatrix} \mathbf{J}_1/\sqrt{\mathbf{J}_1\mathbf{J}_1^T} \\ \mathbf{J}_2\mathbf{N}_1/\sqrt{\mathbf{J}_2\mathbf{N}_1^2\mathbf{J}_2^T} \end{bmatrix} \end{aligned}$$

where $\mathbf{N}_1 = \mathbf{I}_2 - \mathbf{J}_1^\dagger\mathbf{J}_1 = \mathbf{I}_2 - (\mathbf{J}_1^T\mathbf{J}_1)/(\mathbf{J}_1\mathbf{J}_1^T)$, then the solutions of the PIK can be given from Propositions 1 $\sim$ 3.

- Proposition 1 : $\dot{\theta} = \mathbf{J}_1^\dagger \dot{x}_d + (\mathbf{J}_2\mathbf{N}_1)^\dagger(\dot{y}_d - \mathbf{J}_2\mathbf{J}_1^\dagger\dot{x}_d)$
- Proposition 2 : $\dot{\theta} = \mathbf{J}_1^{W\dagger}\dot{x}_d + \mathbf{N}_1^W\mathbf{J}_2^{W\dagger}\dot{y}_d$
- Proposition 3 : $\dot{\theta} = \hat{\mathbf{J}}_1^T c_{11}^\dagger \dot{x}_d + \hat{\mathbf{J}}_2^T c_{22}^\dagger(\dot{y}_d - c_{21}c_{11}^\dagger\dot{x}_d)$

Now we can see the effect of using the QRD more clearly. When we replace the pseudoinverse to the DLPI, we don't need to consider the same damping effect for the QRD because the division operation in the QRD is used to find orthonormal vectors that can be calculated very accurately without worrying about the drastic increase of the inverse solution. However, this property also brings about the instability of the inverse solution in the degenerate case because $c_{11} = \sqrt{\mathbf{J}_1\mathbf{J}_1^T} \approx 0$ but $c_{22} = \sqrt{\mathbf{J}_2\mathbf{N}_1^2\mathbf{J}_2^T} \in [0, \sqrt{\mathbf{J}_2\mathbf{J}_2^T}]$ can be varied sensitively according to the direction of $\mathbf{J}_1$. While, the imperfect projection of the conventional methods can be helpful for this special case because the use of the DLPI gives $\mathbf{J}_1^* \approx \mathbf{J}_1^{W*} \approx \mathbf{0}$ and $\mathbf{N}_1 \approx \mathbf{N}_1^W \approx \mathbf{I}_2$, such that the inverse solutions become $\dot{\theta} \approx \mathbf{J}_2^*\dot{y}_d$ for the Proposition 1 and $\dot{\theta} \approx \mathbf{J}_2^{W*}\dot{y}_d$ for the Proposition 2 that are not sensitive to the direction of $\mathbf{J}_1$.

Increasing damping gains just to prepare the degenerate case eliminates all advantages of using the QRD. Then, possible remedies are to use the varying damping gains [4][6] or the numerical filtering [15]. However, it may not be beneficial to adopt those methods to the structure of the QRD because they use the manipulability measure or the singular values to calculate the distance to the singularities, but the QRD can provide more concrete information about both the task and algorithmic singularities by the block diagonal matrices $\mathbf{C}_{ii}$. Additionally, the inspection of the degenerate case demands the mechanisms to be strictly restricted from the singular configurations. We are proposing a modified DLPI of $\mathbf{C}_{ii}$ with a varying damping gain that uses the determinent of $\mathbf{C}_{ii}$ as the singularity measure

$$\mathbf{C}_{ii}^{\#} \triangleq \mathbf{C}_{ii}^T \left(\mathbf{C}_{ii}\mathbf{C}_{ii}^T + \frac{\lambda_i^{\#2}}{|\mathbf{C}_{ii}|^2 + \varepsilon^2}\mathbf{I}_{m_i} \right)^{-1} \quad (21)$$

where the tunable gain $\lambda_i^{\#}$ is determined considering the boundary of the neighbourhood of the singular configuration and ε can be decided as small as possible just to avoid dividing by zero. The comparison of three different inverse methods in the one-dimensional case is shown in Fig. 2. We can observe that the proposed method ensures smooth shaping of the inverse solution, as well as strongly suppresses near the singular position ($x = 0$).

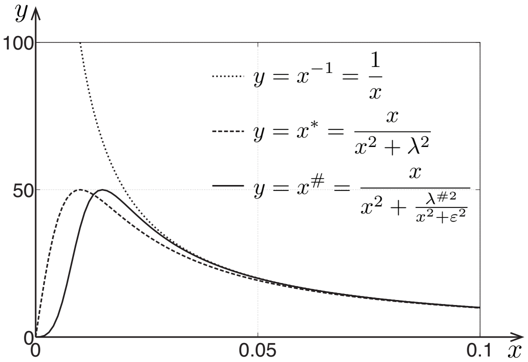


Fig. 2: Comparison of three inverse methods with $\lambda = 0.01$ for $y = x^*$ and $\lambda^{\#} = 0.00013$ for $y = x^{\#}$.

C. Numerical Reconditioning

The problem of numerical imbalance between the diagonal block matrices can occur when the Jacobian is rank deficient or, especially, the number of tasks increases. It can be recognized by representing the i-th Jacobian with its QRD form $\mathbf{J}_i = \mathbf{C}_{i1}\hat{\mathbf{J}}_1 + \mathbf{C}_{i2}\hat{\mathbf{J}}_2 + \cdots + \mathbf{C}_{ii}\hat{\mathbf{J}}_i$ where

$$\mathbf{J}_i\mathbf{J}_i^T = \mathbf{C}_{i1}\mathbf{C}_{i1}^T + \mathbf{C}_{i2}\mathbf{C}_{i2}^T + \cdots + \mathbf{C}_{ii}\mathbf{C}_{ii}^T. \tag{22}$$

Since the right hand side of (22) consists of the positive semidefinite matrices, the norm of $\mathbf{C}_{ii}\mathbf{C}_{ii}^T$ is usually less than $\mathbf{J}_i\mathbf{J}_i^T$, and this tendency becomes more significant when the number of tasks increases. Also, if the i-th task aligns close to the directions of the higher-priority tasks, the QRD reduces or eliminates the diagonal terms of the i-th task. These properties weaken the performance of using the QRD and make difficulties when determining suitable damping gains for each task. This problem can be alleviated if we can remove or minimize the off-diagonal terms, such that $\mathbf{J}_i\mathbf{J}_i^T \approx \mathbf{C}_{ii}\mathbf{C}_{ii}^T$. Similarly to the proposition 2, it can be accomplished by transforming the joint space to the orthogonal space of tasks. We are providing an algorithm that uses the CLD.

Theorem 2 (Proposed Method II (Recursive and Closed Forms))**.** *The recursive form* (23) *and the closed form* (24) *can be solutions of the PIK, as well as reconditions the QRD such that* $|c_{ii}| \leq 1$

$$\dot{\mathbf{q}}_i = \dot{\mathbf{q}}_{i-1} + \hat{\mathbf{J}}_i^T\mathbf{C}_{ii}^{\dagger}(\dot{\mathbf{x}}_i - \mathbf{C}_{i,i-1}^A\hat{\mathbf{J}}_{i-1}^A\dot{\mathbf{q}}_{i-1}),\ \dot{\mathbf{q}}_0 = \mathbf{0},\ \dot{\mathbf{q}} = \tilde{\mathbf{R}}^{-1}\dot{\mathbf{q}}_k \tag{23}$$

$$\dot{\mathbf{q}} = \tilde{\mathbf{R}}^{-1}\hat{\mathbf{J}}^T(\mathbf{I}_m + \mathbf{C}_D^{\ddagger}\mathbf{C}_L)^{-1}\mathbf{C}_D^{\ddagger}\dot{\mathbf{x}} \tag{24}$$

where $\mathbf{W} = \mathbf{J}^T\mathbf{J} + \delta^2\mathbf{I}_n = \tilde{\mathbf{R}}^T\tilde{\mathbf{R}}$ *is the CLD and* $\mathbf{J}_R = \mathbf{J}\tilde{\mathbf{R}}^{-1} = \mathbf{C}\hat{\mathbf{J}}$ *is the QRD.*

Proof. Let $\mathbf{J} = \mathbf{U}\Sigma\mathbf{V}^T$ be the SVD of the Jacobian, $\mathbf{W} = \mathbf{J}^T\mathbf{J} + \delta^2\mathbf{I}_n = \tilde{\mathbf{R}}^T\tilde{\mathbf{R}} > 0$ be the CLD of the symmetric and positive definite matrix, and $\tilde{\mathbf{R}} = \tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}^T$ be the SVD of the upper triangular matrix, then $\mathbf{W}$ can be rewritten as:

$$\mathbf{W} = \mathbf{V}(\Sigma^T\Sigma + \delta^2\mathbf{I}_n)\mathbf{V}^T = \tilde{\mathbf{V}}\tilde{\Sigma}^2\tilde{\mathbf{V}}^T > 0$$

in the form of the Eigenvalue decomposition and we get $\tilde{\Sigma} = (\Sigma^T\Sigma + \delta^2\mathbf{I}_n)^{1/2} \in \mathbb{R}^{n\times n}$ and $\mathbf{V}_r^T\tilde{\mathbf{V}}_r = \mathrm{diag}(\pm 1) \in \mathbb{R}^{r\times r}$ where $\mathbf{V}_r$ and $\tilde{\mathbf{V}}_r \in \mathbb{R}^{n\times r}$ are the block matrices composed of the first r-column vectors of $\mathbf{V}$ and $\tilde{\mathbf{V}}$. The joint space of the forward kinematics (2) can be transformed by using $\tilde{\mathbf{R}}$ as $\mathbf{J}_R\dot{\mathbf{q}}_R = \dot{\mathbf{x}}$ where $\dot{\mathbf{q}}_R = \tilde{\mathbf{R}}\dot{\mathbf{q}}$ and

$$\begin{aligned}
\mathbf{J}_R &= \mathbf{J}\tilde{\mathbf{R}}^{-1}\\
&= \mathbf{U}\begin{bmatrix}\Sigma_r & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{V}_r^T\\ \mathbf{V}_{n-r}^T\end{bmatrix}\\
&\quad\times\begin{bmatrix}\tilde{\mathbf{V}}_r & \tilde{\mathbf{V}}_{n-r}\end{bmatrix}\begin{bmatrix}(\Sigma_r^2 + \delta^2\mathbf{I}_r)^{-1/2} & \mathbf{0}\\ \mathbf{0} & \delta^{-1}\mathbf{I}_{n-r}\end{bmatrix}\tilde{\mathbf{U}}^T\\
&= \mathbf{U}\begin{bmatrix}\Sigma_r\mathbf{V}_r^T\tilde{\mathbf{V}}_r(\Sigma_r^2 + \delta^2\mathbf{I}_r)^{-1/2} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\tilde{\mathbf{U}}^T\\
&= \mathbf{U}\begin{bmatrix}\mathrm{diag}\frac{\pm\sigma_i}{\sqrt{\sigma_i^2+\delta^2}} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\tilde{\mathbf{U}}^T.
\end{aligned}$$

Let $\mathbf{J}_R = \mathbf{C}\hat{\mathbf{J}}$ be the QRD in the transformed coordinate, then we get

$$\mathbf{J}_R\mathbf{J}_R^T = \mathbf{C}\mathbf{C}^T = \mathbf{U}\begin{bmatrix}\mathrm{diag}\frac{\sigma_i^2}{\sigma_i^2+\delta^2} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{U}^T.$$

If $r = m$, we can always find $0 < \delta \ll \sigma_m$, such that $\mathbf{J}_R\mathbf{J}_R^T \approx \mathbf{I}_m$ and $\mathbf{C}^T \approx \mathbf{C}^{-1}$. The only condition for the lower triangular matrix $\mathbf{C}$ to be an orthogonal matrix is that $\mathbf{C} = \mathrm{diag}(\pm 1)$. Therefore, if the Jacobian has its full rank, the coordinate transformation can orthonormalize the tasks, as well as recondition the QRD such that $\mathbf{C} \approx \mathrm{diag}(\pm 1)$. If $r < m$, there is no way to preserve orthogonality between all tasks, but still the QRD can be reconditioned, such that we can get $|c_{ii}| \leq 1$ because the singular values of $\mathbf{J}_R$ is not greater than 1 $(\sigma_i/\sqrt{\sigma_i^2 + \delta^2} \leq 1)$ and the projection or the reflection never increase the norm of vectors while calculating the QRD. The solution of the PIK in the transformed coordinate is given by (15) or (18) and it can come back to the original joint space by using $\dot{\mathbf{q}} = \tilde{\mathbf{R}}^{-1}\dot{\mathbf{q}}_R$. □

Figure 3 shows the effect of the numerical reconditioning. It is clearly shown that the diagonal terms are numerically well conditioned even though there are algorithmic singularities. An important comment is that the QRD can be calculated differently according to the QRD algorithms when the Jacobian is rank deficient. Our recommendation is to use the modified Gram-Schmidt algorithm [16] that seems to give better results for our purposes. Additional discussion can be the determination of δ. The smaller δ makes the singular values of $\mathbf{J}_R = \mathbf{J}\tilde{\mathbf{R}}^{-1}$ closer to one but the inverse solution $\dot{\mathbf{q}} = \tilde{\mathbf{R}}^{-1}\dot{\mathbf{q}}_R$ larger. Therefore, using the CLD still has two conflict rolls, orthogonality between tasks and boundedness of the inverse solutions, similarly to the DLPI and seems to make the problem more complex. However, once the numerical reconditioning is achieved well, we don't need to find the damping gains differently for each task.

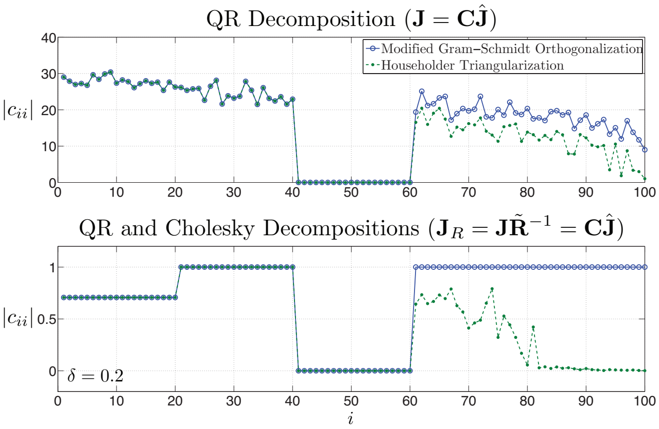


Fig. 3: Comparison of diagonal terms calculated by two different methods with a randomly generated matrix (100×100) that has dependency between rows 1 ∼ 20 and 41 ∼ 60.

$$\mathbf{x}_1 = \begin{bmatrix} \sum_{a=1}^{2k+1} l_a \cos(\sum_{b=1}^{a} \theta_b) \\ \sum_{a=1}^{2k+1} l_a \sin(\sum_{b=1}^{a} \theta_b) \\ \sum_{a=1}^{2k+1} \theta_b \end{bmatrix} \qquad \mathbf{J}_1 = \begin{bmatrix} -\sum_{a=1}^{2k+1} l_a \sin(\sum_{b=1}^{a} \theta_b) & -\sum_{a=2}^{2k+1} l_a \sin(\sum_{b=1}^{a} \theta_b) & \cdots & -l_{2k+1} \sin(\sum_{b=1}^{2k+1} \theta_b) \\ \sum_{a=1}^{2k+1} l_a \cos(\sum_{b=1}^{a} \theta_b) & \sum_{a=2}^{2k+1} l_a \cos(\sum_{b=1}^{a} \theta_b) & \cdots & l_{2k+1} \cos(\sum_{b=1}^{2k+1} \theta_b) \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

$$\mathbf{x}_i = \begin{bmatrix} \sum_{a=1}^{2(k-i+1)} l_a \cos(\sum_{b=1}^{a} \theta_b) \\ \sum_{a=1}^{2(k-i+1)} l_a \sin(\sum_{b=1}^{a} \theta_b) \end{bmatrix} \qquad \mathbf{J}_i = \begin{bmatrix} -\sum_{a=1}^{2(k-i+1)} l_a \sin(\sum_{b=1}^{a} \theta_b) & \cdots & -l_{2(k-i+1)} \sin(\sum_{b=1}^{2(k-i+1)} \theta_b) & 0 & \cdots & 0 \\ \sum_{a=1}^{2(k-i+1)} l_a \cos(\sum_{b=1}^{a} \theta_b) & \cdots & l_{2(k-i+1)} \cos(\sum_{b=1}^{2(k-i+1)} \theta_b) & 0 & \cdots & 0 \end{bmatrix} \tag{25}$$

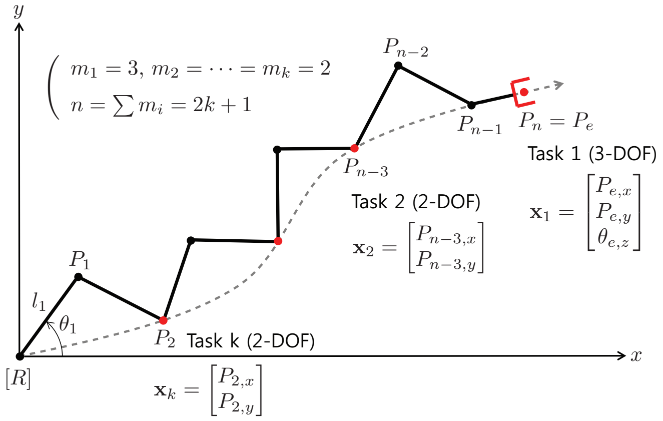

y
$\left(m_1 = 3,\ m_2 = \cdots = m_k = 2 \right.$
$n = \sum m_i = 2k + 1$
P_{n-2}
P_{n-1}
$P_n = P_e$
Task 1 (3-DOF)
$\mathbf{x}_1 = \begin{bmatrix} P_{e,x} \\ P_{e,y} \\ \theta_{e,z} \end{bmatrix}$
P_{n-3}
Task 2 (2-DOF)
$\mathbf{x}_2 = \begin{bmatrix} P_{n-3,x} \\ P_{n-3,y} \end{bmatrix}$
P_1
l_1
θ_1
P_2
Task k (2-DOF)
x
$[R]$
$\mathbf{x}_k = \begin{bmatrix} P_{2,x} \\ P_{2,y} \end{bmatrix}$

Fig. 4: Geometry of the two-dimensional *n*-link manipulator.

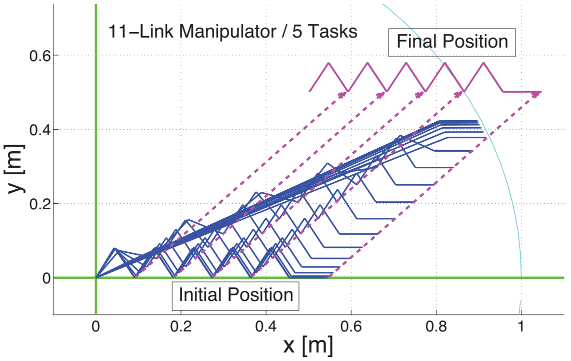


Fig. 5: Desired trajectories of tasks.

IV. Simulation Results and Discussions

Two kinematic simulations are conducted with the *n*-link manipulator in the two-dimensional case and the 7-DOF manipulator in the three-dimensional case. The former focuses on the maximum accuracy of the four different PIK methods, such that the norm of errors will be compared in the variation of the number of tasks without concerning the practical boundedness of the joint velocity. While, the latter will also consider the magnitude of the joint velocities in the determination of the damping gains.

A. n-Link Manipulator

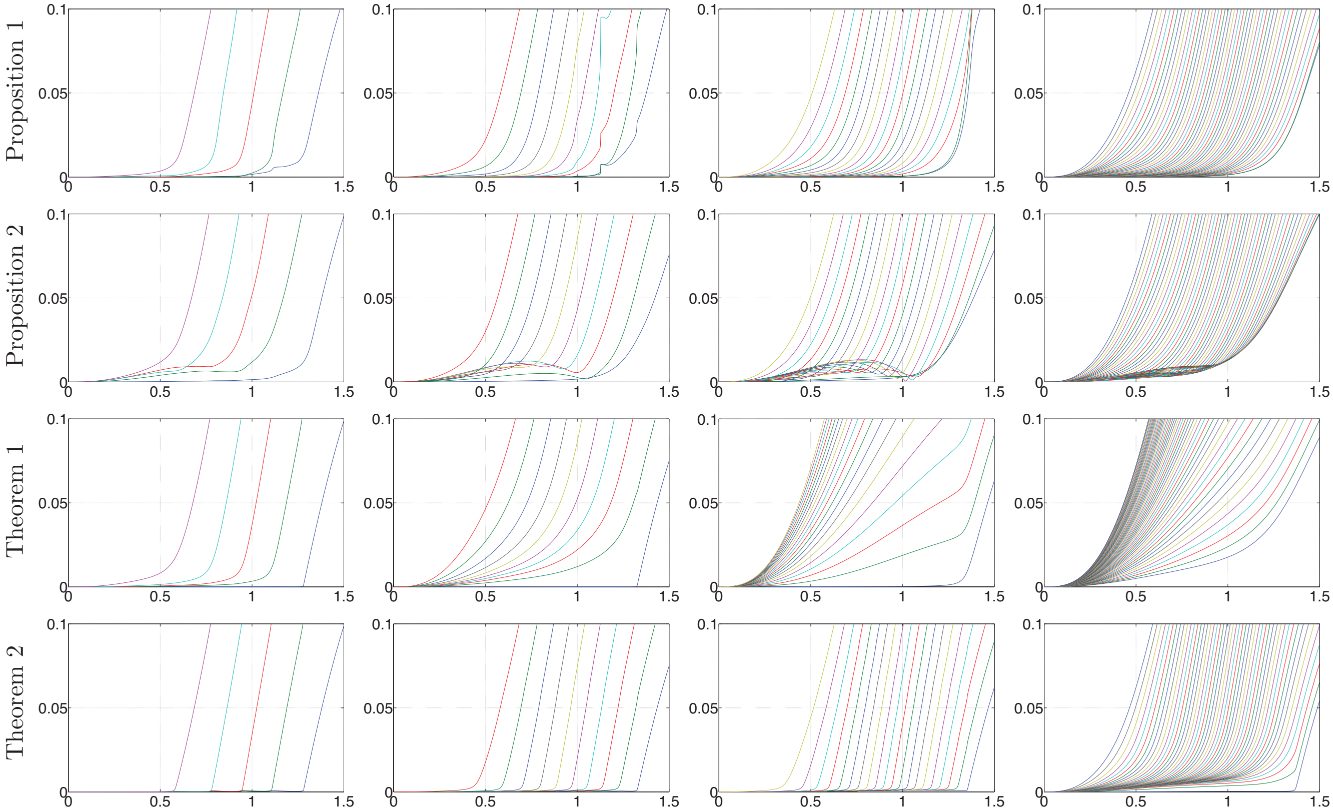
The example shown in Fig. 4 was designed in order to show the performance of the PIKs clearly with the various number of tasks. Generally *k*-tasks are defined. The first is to control the posture of the end-effector $\mathbf{x}_1 = \begin{bmatrix} P_{e,x} & P_{e,y} & \theta_{e,z} \end{bmatrix}^T \in \mathbb{R}^3$ and the *i*-th is to control the intermediate point $\mathbf{x}_i = \begin{bmatrix} P_{n-2i+1,x} & P_{n-2i+1,y} \end{bmatrix}^T \in \mathbb{R}^2$ where $1 \le i \le k$ and $n = 2k+1$. Then, the task variable and its Jacobian can be derived by (25). The priority is highest on the end-effector and becomes lower if the intermediate point is farther from the end-effector.

The desired trajectory of the end-effector position is generated with the destination outside of the workspace by using the fifth-order polynomial. For other tasks, the trajectories are decided to preserve the initial relative positions, such that the desired trajectory of the end-effector is shifted to the initial position of each intermediate point, as shown in Fig. 5. Therefore, the manipulator is required to follow the impossible trajectories and, when the end-effector follows the desired trajectory, the lower-priority tasks suffer from the algorithmic singularities starting from the lowest-priority task and if some links are stretched completely, they also meet the task singularities. The task errors are compared to see how the priority between tasks is preserved accurately. During the simulation, the CLIK is used with the feedback gain $k_i = 10$ for all simulations and the control period is set to 1ms. The tunable damping gains are determined on the third decimal digits right before the chattering occurs or the joint velocity shows impulsive curves.

The simulation results are shown in Fig. 6 where the norms of the task errors ($\|\mathbf{e}_i\| = \|\mathbf{x}_{d,i} - \mathbf{x}_i\|$, $i = 1 \sim k$) are displayed. Four different numbers of tasks are tested with four different inverse solutions. Since we ordered the manipulator to reach outside of the reachable space for all tasks, finally all tasks will make the task errors. If the inverse solution preserves the priority of tasks accurately, the task errors of the higher-priority tasks should be close to zero when the lower-priority tasks cannot be performed perfectly and make the task errors. In this point of view, the fourth method that uses both the QRD and the CLD gives the best results. Even with the 50 tasks and 101 links, the method can accurately perform the tasks preserving the priority compared to other methods.

B. KUKA LWR with Three Tasks

The obstacle avoidance control along with the end-effector position and orientation controls of the 7-DOF manipulator, KUKA LWR, is considered as shown in Fig. 7. The end-effector is supposed to follow a line that reaches to the outside of the workspace while maintaining its initial orientation. A static obstacle is located near the manipulator that may induce the collision if avoding motions are not generated properly. The obstacle avodance is realized based



(a) 5 Tasks (11-DOF) (b) 10 Tasks (21-DOF) (c) 20 Tasks (41-DOF) (d) 50 Tasks (101-DOF)

Fig. 6: Simulation results are displayed with four different numbers of tasks and four prioritized inverse solutions. The horizontal axis is the simulation time [s] and the vertical axis is the norm of the task errors $\|\mathbf{e}_i\| = \|\mathbf{x}_{d,i} - \mathbf{x}_i\|$, $i = 1 \sim k$.

on [17]. The expected senario is that, firstly, the manipulator meets an algorithmic singularity in the configuration that the end-effector position and orientation cannot be controlled simultaneously, then it suffers from the task singularity of the position control where the end-effector reaches to the boundary of the workspace.

The performances of the end-effector position and orientation controls are compared with the norms of task erros, while the preformance of the obstacle avoidance control is measured by the minimum distance between the manipulator and the obstacle, so if the distance is farther, then the performance is better. The damping gains $(\delta, \lambda, \lambda^{\#})$ are tuned by hands in considerations of the boundedness of the joint velocities, such that it's tried to match the maximum magnitudes for four PIK methods if possible. The CLIK gains and the control period are same to the previous simulation.

Fig. 8 shows that the proposed methods execute tasks more accurately even near the algorithmic and kinematic singularities. The interference between tasks became severe for Proposition 1, such that it failed to follow the desired trajectories and converged to a wrong configuration. Additionally, it can be checked that the joint velocities of the proposed methods are smooth and bounded similarly to the Proposition 2 in Fig. 9, but the variations of the proposed methods are more dynamic to execute the tasks more accurately.

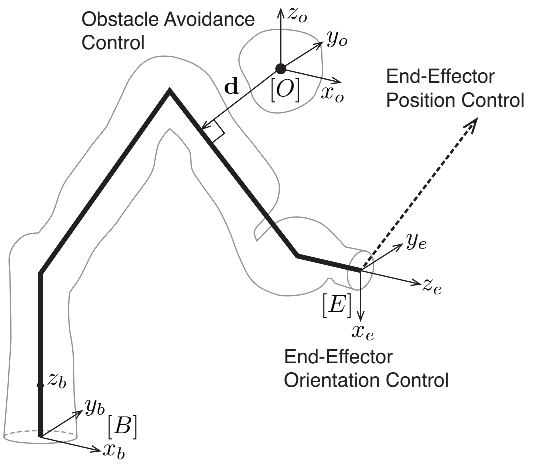

Obstacle Avoidance Control
z_o
y_o
$\mathbf{d}$
$[O]$
x_o
End-Effector Position Control
y_e
z_e
$[E]$
x_e
End-Effector Orientation Control
z_b
y_b
$[B]$
x_b

Fig. 7: Three tasks for the KUKA LWR are defined as: 1) end-effector position control, 2) end-effector orientation control, and 3) obstacle avoidance control in the order of higer to lower priorities.

V. Conclusion

We have proposed two non-iterative solutions in both the recursive and the closed forms for the problem of the prioritized inverse kinematics by using the QR and the Cholesky decompositions with the aim of separating the

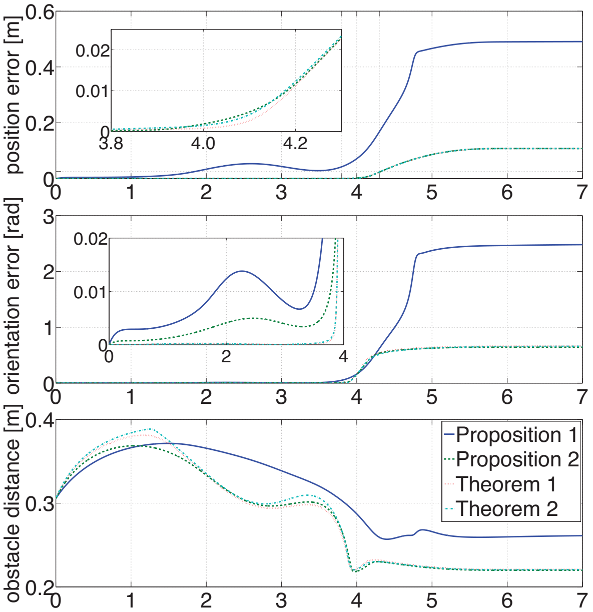


Fig. 8: First two figures show the norms of position and orientation errors and the third figure displays the minimum distance between the manipulator and the obstacle of four PIK methods.

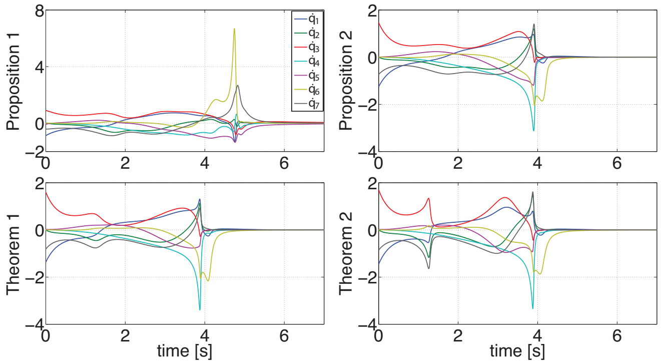


Fig. 9: The joint velocities, $\dot{\mathbf{q}}$ [rad/s], of four PIK methods.

orthogonalization of tasks from the derivation of inverse solutions. Two degenerating properties in dealing with the priority of multiple tasks were explained and, as the remedies, the modified damped least-squares pseudoinverse and the numerical reconditioning method were integrated with the proposed method. Consequently, it was possible for the 101-link manipulator to perform 50 tasks more accurately compared to the conventional methods. Also, the obstacle avoidance control along with the end-effector position and orientation controls for the 7DOFs manipulator, KUKA LWR, has been performed more accurately with the smooth and bounded joint trajectories.

ACKNOWLEDGMENT

This work is supported partially by Technical University Munich - Institute for Advanced Study, funded by the German Excellence Initiative.

REFERENCES

[1] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-Priority Based Redundancy Control of Robot Manipulators," *The International Journal of Robotics Research*, vol. 6, pp. 3–15, June 1987.

[2] A. A. Maciejewski and C. A. Klein, "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments," *The International Journal of Robotics Research*, vol. 4, pp. 109–117, Sept. 1985.

[3] B. Siciliano and J.-J. E. Slotine, "A General Framework for Managing Multiple Tasks in Highly Redundant Robotic Systems," in *International Conference on Advanced Robotics*, pp. 1211–1216, 1991.

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

[5] C. W. Wampler, "Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, pp. 93–101, Jan. 1986.

[6] S. Chiaverini, "Singularity-Robust Task-Priority Redundancy Resolution for Real-Time Kinematic Control of Robot Manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, pp. 398–410, June 1997.

[7] P. Baerlocher and R. Boulic, "Task-Priority Formulations for the Kinematic Control of Highly Redundant Articulated Structures," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, no. October, pp. 323–329, 1998.

[8] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Closed-Loop Inverse Kinematics Schemes for Constrained Redundant Manipulators with Task Space Augmentation and Task Priority Strategy," *The International Journal of Robotics Research*, vol. 10, pp. 410–425, Aug. 1991.

[9] J. Park, Y. Choi, W. K. Chung, and Y. Youm, "Multiple Tasks Kinematics Using Weighted Pseudo-Inverse for Kinematically Redundant Manipulators," *IEEE International Conference on Robotics and Automation*, vol. 4, pp. 4041–4047, 2001.

[10] Y. Choi, Y. Oh, S. R. Oh, J. Park, and W. K. Chung, "Multiple tasks manipulation for a robotic manipulator," *Advanced Robotics*, vol. 18, pp. 637–653, Jan. 2004.

[11] A. Escande, N. Mansard, and P.-B. Wieber, "Fast resolution of hierarchized inverse kinematics with inequality constraints," *2010 IEEE International Conference on Robotics and Automation*, pp. 3733–3738, May 2010.

[12] O. Kanoun, "Real-Time Prioritized Kinematic Control under Inequality Constraints for Redundant Manipulators," in *Robotics: Science and Systems*, pp. 145–152, 2012.

[13] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Transactions on Man-Machine Systems*, vol. 10, pp. 47–53, 1969.

[14] C. A. Klein and C.-H. Huang, "Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 2, pp. 245–250, 1983.

[15] A. A. Maciejewski and C. A. Klein, "Numerical Fintering for the Operation of Robotic Manipulators through Kinematically Singular Configurations," *Journal of Robotic Systems*, vol. 5, no. 6, pp. 527–552, 1988.

[16] L. N. Trefethen and B. David III., *Numerical linear algebra*. Siam, 1997.

[17] L. Žlajpah and B. Nemec, "Kinematic control algorithms for on-line obstacle avoidance for redundant manipulators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, no. October, pp. 1898–1903, 2002.