Prioritized Inverse Kinematics with Multiple Task Definitions

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

Abstract— **We are proposing a general framework that incorporates multiple task definitions in the prioritized inverse kinematics problem. First, a mathematical description of multiple task definitions is constructed that provides an efficient way to show unprioritized or prioritized accumulations of tasks. Then, smooth transitions between all task definitions are studied, so a method, called *task transition control*, is developed that interpolates joint trajectories using barycentric coordinates and linear dynamical systems to overcome difficulties of interpolating task trajectories in the conventional methods. Consequently, smooth, arbitrary, and consecutive task transitions are achieved in a simple, direct, and general manner and also boundedness of joint trajectories is assured regardless of singularity. Lastly, the idea is tested by two kinematic simulations: obstacle avoidance with the KUKA LWR and task scheduling of a humanoid robot.**

I. Introduction

There have been intensive studies to construct mathematical or algorithmic structures that incorporate multiple tasks in the inverse kinematics problem. Basically, this is to determine smooth joint trajectories of a mechanism that realize multiple task trajectories which are not always realizable or consistent. Much of earlier research has tried to assign priority between tasks [1][2][3][4] to overcome incompatibility, also known as algorithmic singularity, between tasks. In spite of their successive achievements, there are some cases that fixed priority structures cannot fulfill task requirements, i.e., obstacle avoidance [5][6], joint limits [7][8], and task scheduling [9][10], in which tasks need to be added, removed, or switched during operations.

The majority of flexible priority structures have been implemented by introducing a parameter for each task that provides smooth transitions in the task activation and deactivation processes. Maciejewski [2] and Zlajpah [5] added the obstacle avoidance task at the lower-priority level when a manipulator approaches to obstacles. A problem was that if the secondary task is not compatible with the primary task, then the whole operation has to be stopped to prevent collisions. Later, Petric [6] showed a variant that adds the obstacle avoidance task at the higher-priority level. A more general structure was proposed by Mansard [7] that allows activations of arbitrary number of tasks at any priority level using the continuous inverse of the Jacobian multiplied by the smooth activation matrix. Lee [8] also suggested an alternative generic structure that modifies desired task trajectories while keeping classical priority structures.

However, some difficulties are found in using the aforementioned methods. First, to find the inverse mapping in the general case is complicated because smoothness of joint trajectories has to be assured by activation parameters defined in the task space. Second, the parameters are coupled with task variables in an elaborate way, such that it is hard to analyze the behavior of the inverse mapping during transitions and it becomes aggravated near singularity. Third, it seems difficult to switch priority between tasks by interpolating task trajectories; it has been implemented by interpolating joint trajectories [8][9].

In this paper, we propose a simple, direct, and general method that allows smooth, arbitrary, and consecutive task transitions without difficulties in interpolating task trajectories. To begin with, the background of inverse kinematics with multiple tasks is provided in Section II. Next, the concept of flexible priority structures is expanded to multiple task definitions by constructing an appropriate mathematical description in Section III-A. Then, the idea of interpolating joint trajectories by using barycentric coordinates [11][12] is introduced in Section III-B, and a comparative study shows that joint space representations of the conventional methods are special forms of our proposal in Section III-C. Later, the idea is completed by connecting barycentric coordinates to linear dynamical systems for smooth, arbitrary, and consecutive transitions between multiple task definitions in Section III-D. Finally, the idea is tested with two kinematic simulations in Section IV, and concluding remarks are in Section V.

II. Inverse Kinematics with Multiple Tasks

A task T is defined in the velocity level [13] by a task variable $\dot{\mathbf{x}} \triangleq \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \in \mathbb{R}^m$ and its desired trajectory $\dot{\mathbf{x}}_d(\mathbf{q},t) \in \mathbb{R}^m$. $\mathbf{q} \in \Omega_q \subset \mathbb{R}^n$ is the joint variable, $\dot{\bullet}$ is the time derivative of $\bullet$, and $\mathbf{J} \triangleq \partial \mathbf{f}/\partial \mathbf{q} \in \mathbb{R}^{m\times n}$ is the Jacobian of the forward kinematic function. The inverse kinematics is to find $\dot{\mathbf{q}}(\mathbf{q},t)$ that minimizes a task velocity error $\dot{\mathbf{e}} \triangleq \dot{\mathbf{x}}_d - \dot{\mathbf{x}}$. Usually, $t \in [t_0,t_f]$ is assumed with its initial and final operating times. In general, multiple tasks $T_1,\cdots,T_k$ are defined with a task variable $\dot{\mathbf{x}}_i \triangleq \mathbf{J}_i(\mathbf{q})\dot{\mathbf{q}} \in \mathbb{R}^{m_i}$ and its desired trajectory $\dot{\mathbf{x}}_{d,i}(\mathbf{q},t) \in \mathbb{R}^{m_i}$ for each task T_i where $\mathbf{J}_i \triangleq \partial \mathbf{f}_i/\partial \mathbf{q} \in \mathbb{R}^{m_i \times n}$ and $i \in \mathbb{N}_{\leq k} \triangleq \{1,\cdots,k\}$. We may consider relations between two tasks T_a and T_b that have inverse kinematic solutions $\dot{\mathbf{q}}_a$ and $\dot{\mathbf{q}}_b$. 1) (equality) $T_a = T_b$ represents $\forall(\mathbf{q},t) \in \Omega_q \times [t_0,t_f]$, $\dot{\mathbf{x}}_a = \dot{\mathbf{x}}_b$ and $\dot{\mathbf{x}}_{d,a} = \dot{\mathbf{x}}_{d,b}$. 2) (equivalence) $T_a \leftrightarrow T_b$ represents $\forall(\mathbf{q},t) \in \Omega_q \times [t_0,t_f]$, $\dot{\mathbf{q}}_a = \dot{\mathbf{q}}_b$. 3) (priority) $T_a \sim T_b$ / $T_a \prec T_b$ / $T_a \succ T_b$ represents the priority of T_a is same to / higher than / lower than the priority of T_b.

[1]S. An and [2]D. Lee are with the Chair of Automatic Control Engineering, Technical University of Munich, D-80333 Munich, Germany. sangik.an@tum.de and dhlee@tum.de

Depending on the priority relations, $T_1, \cdots, T_k$ can be accumulated in the unprioritized or prioritized manner. An unprioritized task $(T_1, \cdots, T_k)$ is an unordered set of tasks such that $\forall i, j \in \mathbb{N}_{\leq k}$, $T_i \sim T_j$. The unprioritized inverse kinematics (UIK) for $(T_1, \cdots, T_k)$ is to find $\dot{\mathbf{q}}$ that minimizes the total task error $\|\dot{\mathbf{e}}\| \triangleq (\sum_{i=1}^{k} \|\dot{\mathbf{e}}_i\|^2)^{1/2}$ where $\dot{\mathbf{e}}_i \triangleq \dot{\mathbf{x}}_{d,i} - \dot{\mathbf{x}}_i$ and $\| \bullet \|$ is the Euclidean norm of $\bullet$. A prioritized task $[T_1, \cdots, T_k]$ is a totally ordered set of tasks such that $\forall i, j \in \mathbb{N}_{\leq k}$, $T_i \prec T_j$ if $i < j$. The prioritized inverse kinematics (PIK) for $[T_1, \cdots, T_k]$ is to find $\dot{\mathbf{q}}$ that minimizes task errors, $\|\dot{\mathbf{e}}_i\|$, $1 \leq i \leq k$, on the condition that the minimum task errors of the higher priority tasks, $\|\dot{\mathbf{e}}_j\|$, $1 \leq j < i$, are not changed.

The minimum norm solution of the UIK is given by

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}(\mathbf{q})\dot{\mathbf{x}}_d \tag{1}$$

where $\dot{\mathbf{x}}_d \triangleq \begin{bmatrix} \dot{\mathbf{x}}_{d,1}^T & \cdots & \dot{\mathbf{x}}_{d,k}^T \end{bmatrix}^T \in \mathbb{R}^m$, $m \triangleq \sum_{i=1}^{k} m_i$, and $\mathbf{J}^{\dagger} \triangleq \mathbf{V}\Sigma^{\dagger}\mathbf{U}^T \in \mathbb{R}^{n \times m}$ is the Moore-Penrose pseudoinverse [14] defined from the singular value decomposition (SVD) $\mathbf{J} \triangleq \begin{bmatrix} \mathbf{J}_1^T & \cdots & \mathbf{J}_k^T \end{bmatrix}^T = \mathbf{U}\Sigma\mathbf{V}^T \in \mathbb{R}^{m \times n}$. The solution of the PIK has been proposed in many ways [1][2][3][4] and recently methods using the QR decomposition (QRD) and the Cholesky decomposition (CLD) have been proposed that eliminate the inteference between orthogonalization and inversion processes [15] as shown in the following two propositions.

Proposition 1 (An's First Method)**.** *The recursive form* (2) *or the closed form* (3) *can be a solution of the PIK*

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_k,\ \dot{\mathbf{q}}_i = \dot{\mathbf{q}}_{i-1} + \hat{\mathbf{J}}_i^T \mathbf{C}_{ii}^{\dagger}(\dot{\mathbf{x}}_{d,i} - \mathbf{C}_{i,i-1}^{A}\hat{\mathbf{J}}_{i-1}^{A}\dot{\mathbf{q}}_{i-1}),\ \dot{\mathbf{q}}_0 = \mathbf{0} \tag{2}$$

$$\dot{\mathbf{q}} = \hat{\mathbf{J}}^T(\mathbf{I}_m + \mathbf{C}_D^{\ddagger}\mathbf{C}_L)^{-1}\mathbf{C}_D^{\ddagger}\dot{\mathbf{x}}_d \tag{3}$$

where $\mathbf{J}(\mathbf{q}) \triangleq \underbrace{\mathbf{C}(\mathbf{q})}_{m \times m}\underbrace{\hat{\mathbf{J}}(\mathbf{q})}_{m \times n} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{k1} & \mathbf{C}_{k2} & \cdots & \mathbf{C}_{kk} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{J}}_1 \\ \hat{\mathbf{J}}_2 \\ \vdots \\ \hat{\mathbf{J}}_k \end{bmatrix}$ *is the reduced QRD* $\mathbf{J}^T = \mathbf{Q}\mathbf{R} = \hat{\mathbf{J}}^T\mathbf{C}^T$, $\mathbf{C}_{ij}^{A} \triangleq \begin{bmatrix} \mathbf{C}_{i1} & \cdots & \mathbf{C}_{ij} \end{bmatrix}$, $\hat{\mathbf{J}}_i^{A} \triangleq \begin{bmatrix} \hat{\mathbf{J}}_1^T & \cdots & \hat{\mathbf{J}}_i^T \end{bmatrix}^T$, $\mathbf{C}_D \triangleq \mathrm{diag}(\mathbf{C}_{ii})$, $\mathbf{C}_D^{\ddagger} \triangleq \mathrm{diag}(\mathbf{C}_{ii}^{\dagger})$, $\mathbf{C}_L \triangleq \mathbf{C} - \mathbf{C}_D$, *and* $\mathbf{I}_m \triangleq \mathrm{diag}(1) \in \mathbb{R}^{m \times m}$.

Proposition 2 (An's Second Method)**.** *The recursive form* (4) *or the closed form* (5) *can be a solution of the PIK, as well as reconditions the QRD such that* $|c_{jj}| \leq 1$, $1 \leq j \leq m$

$$\dot{\mathbf{q}} = \tilde{\mathbf{R}}^{-1}\dot{\mathbf{q}}_k,\ \dot{\mathbf{q}}_i = \dot{\mathbf{q}}_{i-1} + \hat{\mathbf{J}}_i^T \mathbf{C}_{ii}^{\dagger}(\dot{\mathbf{x}}_i - \mathbf{C}_{i,i-1}^{A}\hat{\mathbf{J}}_{i-1}^{A}\dot{\mathbf{q}}_{i-1}),\ \dot{\mathbf{q}}_0 = \mathbf{0} \tag{4}$$

$$\dot{\mathbf{q}} = \tilde{\mathbf{R}}^{-1}\hat{\mathbf{J}}^T(\mathbf{I}_m + \mathbf{C}_D^{\ddagger}\mathbf{C}_L)^{-1}\mathbf{C}_D^{\ddagger}\dot{\mathbf{x}} \tag{5}$$

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

An's solutions have shown more accurate task executions compared to the conventional solutions with the similar boundedness of joint velocities. Since we will interpolate multiple inverse solutions, the quality of each solution will directly affect to the final performance. Also, the closed form solutions will ease the proof of Theorem 1 in Section III.

III. Prioritized Inverse Kinematics with Multiple Task Definitions

A. Mathematical Description of Multiple Task Definitions

We define a *basic task* $T \triangleq (T_1, \cdots, T_k)$ as an unprioritized task that is given before the initial operating time with basic subtasks T_j and assume that the definition of T is consistent during operations. Then, we may define an *(induced) task* $T^i \triangleq [T_1^i, \cdots, T_{k^i}^i]$ as a prioritized task induced from the basic task by accumulating a part of basic subtasks on the condition that $m^i \leq \min\{m, n\}$ and $T_a^i \neq T_b^i$ if $a \neq b$. Also, an *(induced) task set* $\mathscr{T} \triangleq \{T^1, \cdots, T^l\}$ can be defined as a set that contains all tasks considered where $l \in \mathbb{N}$ is the number of tasks. For each T^i, there exists a prioritized inverse solution $\dot{\mathbf{q}}^i$ given by Proposition 1 or 2. We define a prioritized inverse solution set $\mathscr{Q} \triangleq \{\dot{\mathbf{q}}^1, \cdots, \dot{\mathbf{q}}^l\}$ as a set that contains inverse solutions of all tasks. For the mathematical completion, we introduce a *null task* $\varnothing$ that has an inverse solution $\dot{\mathbf{q}}_{\varnothing} = \mathbf{0}$. Usually, the mapping $\mathscr{T} \mapsto \mathscr{Q}$ is surjective because there could be tasks that are equivalent to each other.

Theorem 1 (Equivalent Induced Tasks)**.** *Equivalent tasks appear in the following cases:*

1) $\forall A \triangleq \{a_1, \cdots, a_\alpha\} \subset \mathbb{N}_{\leq k}$, *let* $T^a \triangleq (T_{a_1}, \cdots, T_{a_\alpha})$ *and* T^b *be an unprioritized task generated by changing the order of basic subtasks in* T^a*; then,* $T^a \leftrightarrow T^b$.
2) $\forall A \triangleq \{a_1, \cdots, a_\alpha\} \subset \mathbb{N}_{\leq k}$, *let* $T^a \triangleq [T_{a_1}, \cdots, T_{a_\alpha}]$ *and* T^b *be a prioritized task generated by changing the order of basic subtasks in* T^a*; then,* $T^a \leftrightarrow T^b$ *if* $\forall t \in [t_0, t_f]$, $\mathrm{rank}(\begin{bmatrix} \mathbf{J}_{a_1}^T & \cdots & \mathbf{J}_{a_\alpha}^T \end{bmatrix}^T) = m_{a_1} + \cdots + m_{a_\alpha}$.
3) $\forall A \triangleq \{a_1, \cdots, a_\alpha\} \subset \mathbb{N}_{\leq k}$ *and* $\forall B \triangleq \{b_1, \cdots, b_\beta\} \subset \mathbb{N}_{\leq k}$ *with* $A \cap B = \varnothing$, *let* $T^a \triangleq (T_{a_1}, \cdots, T_{a_\alpha})$, $T^b \triangleq [T_{b_1}, \cdots, T^a, \cdots, T_{b_\beta}]$, T^c *be an unprioritized task generated by changing the order of basic subtasks in* T^a, *and* T^d *be a prioritized task generated by switching* T^a *to* T^c *in* T^b*; then,* $T^b \leftrightarrow T^d$.

Proof. See Appendix I. □

It is preferable to remove equivalence relation between tasks explained in Theorem 1 to reduce l. However, the difficulty arises on Case 2) that requires to know whether a task will meet singularities or not before the initial time. Originally, priority has been introduced due to singularity. If there is no singularity, the unprioritized and prioritized inverse solutions are identical as shown in the proof of Case 2); therefore, priority is not neccessary. We assume that if a prioritized task is defined, then there is singularity or possibility to meet unknown singularity during operations. To better understand Theorem 1, an example of all possible tasks is shown in Fig. 1. Task accumulations are done as discussed above and we assume $T_i = (T_i) = [T_i] = [(T_i)]$.

B. Barycentric Coordinates of Inverse Solutions

In the case that a task definition needs to be changed during operations, a smooth transition between tasks is necessary to prevent discontinuous jumps in the joint trajectory. In order to explain the concept of our proposal,

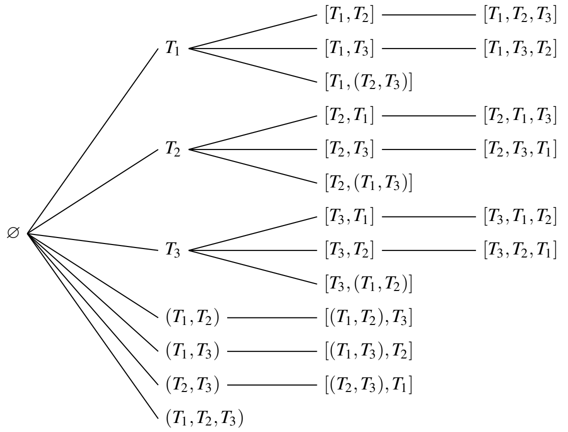
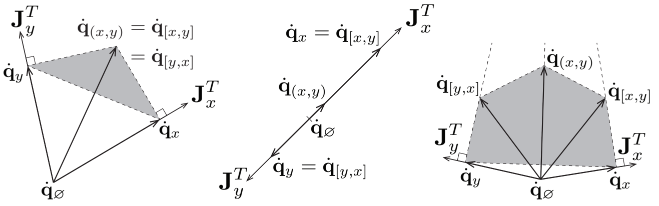


Fig. 1. All possible tasks when $k=3$ and $m^i \leq m \leq n$.



$\mathbf{J}_y^T$
$\dot{\mathbf{q}}_{(x,y)} = \dot{\mathbf{q}}_{[x,y]}$
$= \dot{\mathbf{q}}_{[y,x]}$
$\dot{\mathbf{q}}_y$
$\mathbf{J}_x^T$
$\dot{\mathbf{q}}_x$
$\dot{\mathbf{q}}_\varnothing$
$\dot{\mathbf{q}}_x = \dot{\mathbf{q}}_{[x,y]}$
$\mathbf{J}_x^T$
$\dot{\mathbf{q}}_{(x,y)}$
$\dot{\mathbf{q}}_\varnothing$
$\mathbf{J}_y^T$
$\dot{\mathbf{q}}_y = \dot{\mathbf{q}}_{[y,x]}$
$\dot{\mathbf{q}}_{(x,y)}$
$\dot{\mathbf{q}}_{[y,x]}$
$\dot{\mathbf{q}}_{[x,y]}$
$\mathbf{J}_y^T$
$\mathbf{J}_x^T$
$\dot{\mathbf{q}}_y$
$\dot{\mathbf{q}}_\varnothing$
$\dot{\mathbf{q}}_x$

(a) without singularity (b) with singularity (c) near singularity

Fig. 2. Multiple inverse solutions of a two-link manipulator.

we show a simple case with a two-link manipulator that derives our basic idea. Let's suppose that the basic task is given by $T = (T_x, T_y)$ which are the x- and y-directional motions of the end-effector and the task set is defined as $\mathscr{T} = \{T_x, T_y, (T_x, T_y), [T_x, T_y], [T_y, T_x]\}$ along with the inverse solution set $\mathscr{Q} = \{\dot{\mathbf{q}}_x, \dot{\mathbf{q}}_y, \dot{\mathbf{q}}_{(x,y)}, \dot{\mathbf{q}}_{[x,y]}, \dot{\mathbf{q}}_{[y,x]}\}$. Then, we may consider smooth task transitions in three cases: 1) without, 2) with, and 3) near the algorithmic singularity between T_x and T_y as shown in Fig. 2. If there is no singularity, Case 2) in Theorem 1 gives $\dot{\mathbf{q}}_{(x,y)} = \dot{\mathbf{q}}_{[x,y]} = \dot{\mathbf{q}}_{[y,x]}$. If there is sigularity, two inverse solutions $\dot{\mathbf{q}}_x$ and $\dot{\mathbf{q}}_y$ align on the same line and the priority relation gives $\dot{\mathbf{q}}_x = \dot{\mathbf{q}}_{[x,y]}$ and $\dot{\mathbf{q}}_y = \dot{\mathbf{q}}_{[y,x]}$. $\dot{\mathbf{q}}_{(x,y)}$ is determined somewhere between two inverse solutions. The inverse solutions near singularity have to be calculated considering boundedness of solutions. The damped least-squares pseudoinverse (DLPS) [15][16][17] provides smooth shaping of the inverse solutions near singularity and it may separate all solutions similarly to (c) of Fig 2.

Now, let's think of a method that gives smooth task transitions between all tasks regardless of singularity. Additionally, we consider consecutive transitions, such that new transition is required during transitions. If we try to do it by interpolating task trajectories, it is not a trivial problem even in this simple case and the difficulty increases exponentially as the number of tasks grows. Instead, we can interpolate joint trajectories directly; then, for any kind of transitions, the inverse solution should be found inside of polygons formed by connecting inverse solutions of tasks, i.e., triangle for (a), line for (b), and pentagon for (c) of Fig 2, in order to prevent unnecessary transition motions. We introduce the barycentric coordinate [11][12] which is a popular way in the computer graphics that parameterize points inside of a convex polytope given by a set of vertices, but there are differences in using it. First, we are designing smooth transitions of the barycentric coordinates instead of calculating the coordinates of points from the verticies. Also, the inverse solutions of tasks will not always form a convex polytope. It could be a concave polytope or just a set of points in the joint space.

Definition 1 (Barycentric Coordinates)**.** The barycentric coordinates of $\dot{\mathbf{q}}$ with respect to $\mathscr{Q}$ is defined as any set of real coefficients $w^1, \cdots, w^l$ depending on $(\mathbf{q}, t)$, such that, $\forall(\mathbf{q}, t) \in \Omega_q \times [t_0, t_f]$, all the following properties hold:

- Nonnegativity: $w^i \geq 0$,
- Linearity: $\dot{\mathbf{q}} = \sum_{i=1}^{l} w^i \dot{\mathbf{q}}^i$ with $\sum_{i=1}^{l} w^i = 1$, and
- Smoothness: $w^i \in C^s(\mathbf{q}, t)$

where $s \in \mathbb{N}^{\geq 0}$ depends on the degree of smoothness needed.

Theorem 2 (Boundedness of Barycentric Coordinates)**.** *The inverse solution $\dot{\mathbf{q}}$ given by the barycentric coordinates with respect to $\mathscr{Q}$ is a point inside of the minimum enclosing convex polytope of $\mathscr{Q}$ and $\|\dot{\mathbf{q}}\| \leq \max\{\|\dot{\mathbf{q}}^1\|, \cdots, \|\dot{\mathbf{q}}^l\|\}$.*

Proof. See Appendix II. □

Theorem 2 reveals us an important feature of the proposed method that boundedness of barycentric coordinates can be achieved by bounding each inverse solutions of tasks usually treated by the DLPS, regardless of the existence of singularity and the consecutive task transitions. In other words, we may decouple two complex problems which are to stabilize the inverse solutions and to generate smooth task transitions that is difficult to get by interpolating task trajectories. However, there is still an issue related to convergence of $\dot{\mathbf{q}}$ because w^i depends on $(\mathbf{q}, t)$; w^i can chatter if two tasks are inconsistent in the switching manifold. We may consider the smooth initialization and finalization, starting from and ending to $\dot{\mathbf{q}} = \mathbf{0}$; then, we can include the null task $\varnothing$ to $\mathscr{T}$ that induces $\mathscr{Q}$ to include additional point $\dot{\mathbf{q}}_\varnothing = \mathbf{0}$.

C. Comparative Study

It is interesting to see how the conventional methods can be represented in the joint space. Three examples are listed in a simple case that two tasks T_1 and T_2 are given. Maciejewski [2] and Zlajpah [5]'s method is given by

$$\begin{aligned}
\dot{\mathbf{q}} &= \mathbf{J}_1^\dagger \dot{\mathbf{x}}_{d,1} + h(\mathbf{J}_2\mathbf{N}_1)^\dagger(\dot{\mathbf{x}}_{d,2} - \mathbf{J}_2\mathbf{J}_1^\dagger \dot{\mathbf{x}}_{d,1}) \\
&= (1-h)\mathbf{J}_1^\dagger \dot{\mathbf{x}}_{d,1} + h\left[\mathbf{J}_1^\dagger \dot{\mathbf{x}}_{d,1} + (\mathbf{J}_2\mathbf{N}_1)^\dagger(\dot{\mathbf{x}}_{d,2} - \mathbf{J}_2\mathbf{J}_1^\dagger \dot{\mathbf{x}}_{d,1})\right] \\
&= (1-h)\dot{\mathbf{q}}_1 + h\dot{\mathbf{q}}_{[1,2]}
\end{aligned}$$

where $h \in [0,1]$ is the activation parameter that insert T_2 on the lower-priority level. On the other hand, Petric [6]'s method given by

$$\begin{aligned}
\dot{\mathbf{q}} &= h\mathbf{J}_1^\dagger \dot{\mathbf{x}}_{d,1} + (\mathbf{I} - h\mathbf{J}_1^\dagger \mathbf{J}_1)\mathbf{J}_2^\dagger \dot{\mathbf{x}}_{d,2} \\
&= (1-h)\mathbf{J}_2^\dagger \dot{\mathbf{x}}_{d,2} + h\left[\mathbf{J}_1^\dagger \dot{\mathbf{x}}_{d,1} + (\mathbf{I} - \mathbf{J}_1^\dagger \mathbf{J}_1)\mathbf{J}_2^\dagger \dot{\mathbf{x}}_{d,2}\right] \\
&= (1-h)\dot{\mathbf{q}}_2 + h\dot{\mathbf{q}}_{[1,2]}
\end{aligned}$$

inserts T_1 on the higher-priority level. When calculating $\dot{\mathbf{q}}_{[1,2]}$, the former used Nakamura [1] and Maciejewski [2]'s method and the latter used Chiaverini [3]'s method. Lee [8]'s method is given by

$$\dot{\mathbf{q}} = \mathbf{J}_1^\dagger \dot{\mathbf{x}}_1^* + (\mathbf{J}_2\mathbf{N}_1)^\dagger (\dot{\mathbf{x}}_2^* - \mathbf{J}_2\mathbf{J}_1^\dagger \dot{\mathbf{x}}_1^*) \tag{6}$$

$$\dot{\mathbf{x}}_1^* = h_1\dot{\mathbf{x}}_{d,1} + (1-h_1)\mathbf{J}_1\mathbf{J}_2^\dagger h_2 \dot{\mathbf{x}}_{d,2} \tag{7}$$

$$\dot{\mathbf{x}}_2^* = h_2\dot{\mathbf{x}}_{d,2} + (1-h_2)\mathbf{J}_2\mathbf{J}_1^\dagger h_1 \dot{\mathbf{x}}_{d,1} \tag{8}$$

and can be modified into

$$\dot{\mathbf{q}} = h_1(1-h_2)\dot{\mathbf{q}}_1 + h_2(1-h_1)\dot{\mathbf{q}}_2 + h_1h_2\dot{\mathbf{q}}_{[1,2]} + w_\varnothing \dot{\mathbf{q}}_\varnothing \tag{9}$$

with $w_\varnothing \triangleq 1 - h_1 - h_2 + h_1h_2$ and $\{h_1, h_2\} \subset [0,1]$. The derivation is shown in Appendix III.

We can see that the inverse solutions of the listed methods can be represented as the barycentric coordinates in the joint space. Indeed, the first two methods show no differences of interpolating task trajectories or joint trajectories except for the computation methods, but those are difficult to generalize. The third method provides more general structure, but we cannot say that (6) $\sim$ (8) and (9) will give the same results if there is singularity because the derivation of (9) requires $[\mathbf{J}_1^T \ \mathbf{J}_2^T]^T$ has its full rank. Additionally, the barycentric coordinates given by (9) has a constraint in determining their values: there are four inverse solutions ($\dot{\mathbf{q}}_1, \dot{\mathbf{q}}_2, \dot{\mathbf{q}}_{[1,2]}, \dot{\mathbf{q}}_\varnothing$) but only three parameters ($h_1, h_2, w_\varnothing$), so transitions between two tasks could be influenced by others; for example, transitions between $\dot{\mathbf{q}}_1$ and $\dot{\mathbf{q}}_2$ are always influenced by $\dot{\mathbf{q}}_{[1,2]}$.

It could be also helpful to see methods for the transition between different priority hierachies: $T_{[1,2]}$ and $T_{[2,1]}$. Keith [9] directly interpolated two joint trajectories

$$\dot{\mathbf{q}} = h\dot{\mathbf{q}}_{[2,1]} + (1-h)\dot{\mathbf{q}}_{[1,2]} \tag{10}$$

, while Lee [8] kept a similar structure of (6) $\sim$ (8)

$$\dot{\mathbf{q}} = \mathbf{J}_1^\dagger \dot{\mathbf{x}}_1^\# + (\mathbf{J}_2\mathbf{N}_1)^\dagger (\dot{\mathbf{x}}_2^\# - \mathbf{J}_2\mathbf{J}_1^\dagger \dot{\mathbf{x}}_1^\#)$$
$$\dot{\mathbf{x}}_1^\# = h\mathbf{J}_1\dot{\mathbf{q}}_{[2,1]} + (1-h)\mathbf{J}_1\dot{\mathbf{q}}_{[1,2]}$$
$$\dot{\mathbf{x}}_2^\# = h\mathbf{J}_2\dot{\mathbf{q}}_{[2,1]} + (1-h)\mathbf{J}_2\dot{\mathbf{q}}_{[1,2]}$$

and showed that it can be modified into (10). However, the full rank of $[\mathbf{J}_1^T \ \mathbf{J}_2^T]^T$ is still required: if there is singularity, $\mathbf{N}_1\left(\mathbf{I} - (\mathbf{J}_2\mathbf{N}_1)^\dagger(\mathbf{J}_2\mathbf{N}_1)\right) \neq \mathbf{0}$ breaks the proof in [8]. Both methods allow the transition between only two task definitions and, as far as we know, a method that provides transitions between all possible task definitions has not yet been proposed. When a basic task is defined as $T = (T_1, T_2)$ with $m \leq n$, all possible task definitions become $\mathscr{T} = \{\varnothing, T_1, T_2, (T_1, T_2), [T_1, T_2], [T_2, T_1]\}$ and the general form of $\dot{\mathbf{q}}$ can be given with the barycentric coordinates as

$$\dot{\mathbf{q}} = w_\varnothing\dot{\mathbf{q}}_\varnothing + w_1\dot{\mathbf{q}}_1 + w_2\dot{\mathbf{q}}_2 + w_{(1,2)}\dot{\mathbf{q}}_{(1,2)} + w_{[1,2]}\dot{\mathbf{q}}_{[1,2]} + w_{[2,1]}\dot{\mathbf{q}}_{[2,1]}.$$

From this comparison, we can observe that to interpolate joint trajectories is much more simple, direct, and general than to interpolate task trajectories and removes difficulties discussed in Section I. However, there is still an unsolved problem: how to change barycentric coordinates to achieve smooth, arbitrary, and consecutive transitions.

D. Smooth Task Transitions

The nonnegativity and linearity properties of barycentric coordinates give us all possible task transitions along with boundedness of inverse solutions. Meanwhile, the smoothness property is for the smooth transitions and there could be infinite number of ways to smoothly traverse between inverse solutions of tasks. Therefore, this is a design problem of $w^i(\mathbf{q}, t)$ that produces smooth task transitions.

Theorem 3 (Task Transition Control (TTC))**.** *The TTC given by* (11) $\sim$ (15) *provides smooth, arbitrary, and consecutive task transitions within $\mathscr{T} = \{T^1, \cdots, T^l\}$, as well as bounds the inverse solution such that $\|\dot{\mathbf{q}}\| \leq \max\{\|\dot{\mathbf{q}}^1\|, \cdots, \|\dot{\mathbf{q}}^l\|\}$*

$$\dot{\mathbf{q}} = \mathbf{Q}\mathbf{w} = \sum_{i=1}^{l} w^i \dot{\mathbf{q}}^i \tag{11}$$

$$\mathbf{w}^{(s+1)} = -\sum_{j=1}^{s} k_j \mathbf{w}^{(j)} + k_0(\mathbf{w}_d - \mathbf{w}) \tag{12}$$

$$\mathbf{w}_d(\mathbf{q}, t) \in \{\hat{\mathbf{e}}^1, \cdots, \hat{\mathbf{e}}^l\} \subset \mathbb{R}^l \tag{13}$$

$$\mathbf{w}(t_0) \in \{\mathbf{a} \in \mathbb{R}^n : \mathbf{1}^T\mathbf{a} = 1, \ \mathbf{a} \geq \mathbf{0}\} \tag{14}$$

$$\mathbf{w}^{(j)}(t_0) = \mathbf{0}, \ \forall j \in \mathbb{N}_{\leq k} \tag{15}$$

where $\mathbf{Q} \triangleq [\dot{\mathbf{q}}^1 \ \cdots \ \dot{\mathbf{q}}^l] \in \mathbb{R}^{n \times l}$, $\mathbf{w} \triangleq [w^1 \ \cdots \ w^l]^T \in \mathbb{R}^l$, $\mathbf{w}^{(j)} \triangleq d^j\mathbf{w}/dt^j$, $s \in \mathbb{N}^{\geq 0}$, $\mathbf{1} \triangleq [1 \ \cdots \ 1]^T \in \mathbb{R}^l$, $\{\hat{\mathbf{e}}^1, \cdots, \hat{\mathbf{e}}^l\}$ is a set of the standard basis in $\mathbb{R}^l$, and $\{k_0, \cdots, k_k\} \subset \mathbb{R}$ are stabilizing control gains that do not generate overshoot of the $(s+1)$-th order linear dynamical system.

Proof. See Appendix IV □

Task transitions are triggered by the discrete value of $\mathbf{w}_d$ and the smooth transitions are achieved by the $(s+1)$-th order dynamics of (12). The convergence behavior of $\mathbf{w}$ can be tuned by the control gains $\{k_0, \cdots, k_k\}$; for example, if $k = 2$, the typical choice could be $k_1 = 2\zeta\omega_n$, $k_0 = \omega_n^2$, and $\omega_n = 2\pi f_n$ with $\zeta \geq 1$ to prevent overshoot of the response. A simple way to design task transition rules is to divide the domain $\Omega_q \times [t_0, t_f]$ into multiple partitions $P_1, \cdots, P_l$ where $\Omega_q \times [t_0, t_f] = \bigcup_{i=1}^{l} P_i$ and $P_i \bigcap P_j = \varnothing$ if $i \neq j$; then, the desired barycentric coordinates can be determined by $\mathbf{w}_d = \hat{\mathbf{e}}_i$ if $(\mathbf{q}, t) \in P_i$. There could be anxiety about the numerical stability that $\mathbf{w}$ may leave the simplex Δ^{l-1} by the numerical inaccuracy of the machine, but the convergence behavior of the $(s+1)$-th order linear system always pulls a state to the vertices of Δ^{l-1}, such that the variation of $\mathbf{w}$ is always toward Δ^{l-1} even if it is outside of Δ^{l-1}. To efficiently implement the TTC, we do not need to calculate inverse solutions of inactive tasks ($w^i \approx 0$). Also, the computation of inverse solutions of all active tasks can be easily parallelized if needed.

IV. Simulation Results and Discussions

Two simulations are conducted. The first simulation shows an application of the obstacle avoidance control with the KUKA LWR and the second simulation tests the task scheduling of a humanoid robot. In both cases, the joint limit is a hard constraint that has to be considered, so we first explain how to handle this in the proposed method.

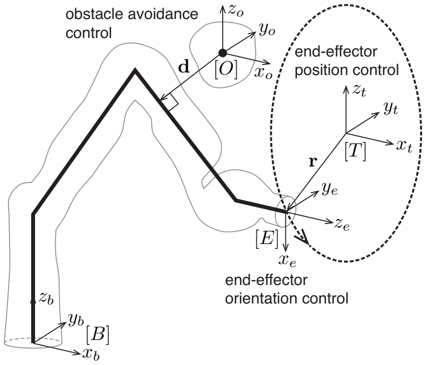


Fig. 3. Scenario of the simulation with the KUKA LWR.

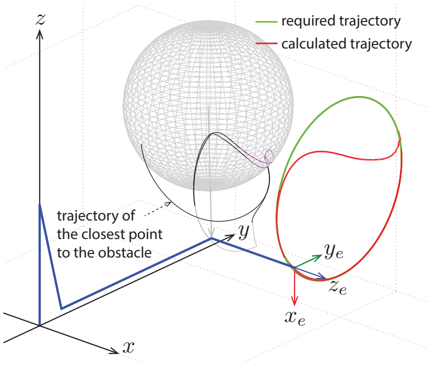


Fig. 4. Trajectories of the end-effector and the closest point.

TABLE I
TASK TRANSITION RULE FOR THE KUKA LWR

	$t < t_0$	$t_0 \le t < t_1$	$t_1 \le t < t_f$	$t_f \le t$
$d \ge d_b$	$\varnothing$	$[T_p, T_o]$	$[T_o, T_p]$	$\varnothing$
$d_c \le d < d_b$	$\varnothing$	$[T_p, T_o, T_c]$	$[T_o, T_p, T_c]$	$\varnothing$
$d < d_c$	$\varnothing$	$[T_c, T_p, T_o]$	$[T_c, T_o, T_p]$	$\varnothing$

Let's say the joint limit is given as $\mathbf{q} \in [\underline{\mathbf{q}},\ \overline{\mathbf{q}}] \subset \mathbb{R}^n$ and we are looking for the PIK solution of the i-th induced task T^i that prevents the robot from approaching to the joint limit. The basic idea comes from [18] that penalizes joints heavily when the joints approach to the limit closely by using the weighting matrix $\mathbf{H} \triangleq \operatorname{diag}(h_a) \in \mathbb{R}^{n\times n}$ where $a \in \mathbb{N}_{\le n}$, $\delta h_a(t) \triangleq h_a(t) - h_a(t-\delta t)$, and

$$h_a(q_a) \triangleq \begin{cases} \dfrac{(\overline{q}_a - \underline{q}_a)^2}{4(\overline{q}_a - q_a)(q_a - \underline{q}_a)} & \delta h_a \ge 0 \\ 1 & \delta h_a < 0 \end{cases}.$$

The penalized forward kinematic equation for T^i can be written as $\dot{\mathbf{x}}^i = \mathbf{J}^i \dot{\mathbf{q}} = (\mathbf{J}^i \mathbf{H}^{-1})(\mathbf{H}\dot{\mathbf{q}}) \triangleq \mathbf{J}_H^i \dot{\mathbf{q}}_H$ and $\dot{\mathbf{q}}_H^i$ can be calculated from Proposition 1 or 2 with $\mathbf{J}_H^i$. Since we have found the inverse solution on the penalized joint space, we need to come back to the original joint space by $\dot{\mathbf{q}}^i = \mathbf{H}^{-1}\dot{\mathbf{q}}_H^i$. For each time step, all induced tasks share the same $\mathbf{H}$, therefore (11) can be written as $\dot{\mathbf{q}} = \sum_{i=1}^{l} w^i \dot{\mathbf{q}}^i = \mathbf{H}^{-1}\left(\sum_{i=1}^{l} w^i \dot{\mathbf{q}}_H^i\right)$ and the joint limit avoidance is assured by the TTC. On the discrete time domain, discontinuity of h_a can generate a chatter in $\dot{\mathbf{q}}$, so it is better to filter out h_a.

A. Obstacle Avoidance with the KUKA LWR

The end-effector posture control is the most crucial task of manipulators and either the position or the orientation can have priority. For example, if a robot is required to bring a glass, priority can change whether the glass is empty or filled with water. Also, the obstacle avoidance control is one of the fundamental tasks [2][5][6] that the task transition is required. Therefore, we have constructed a simulation scenario for the 7-DOF manipulator, KUKA LWR IV+, as shown in Fig. 3. We define the basic task $T = (T_p, T_o, T_c)$ with position control (T_p), orientation control (T_o), and obstacle avoidance control (T_c). The task set is defined as $\mathscr{T} = \{\varnothing, [T_p, T_o], [T_p, T_o, T_c], [T_c, T_p, T_o], [T_o, T_p], [T_o, T_p, T_c], [T_c, T_o, T_p]\}$ with the task transition rule listed in Table I. d is the minimum distance between the robot and the obstacle, d_b is the boundary distance for T_c to be triggered, and d_c is the critical distance for T_c to be handled primarily. Priority between T_p and T_o is switched at time t_1 and warm starting and ending ($\dot{\mathbf{q}}(t_0) = \dot{\mathbf{q}}(\infty) = \mathbf{0}$) are considered. The end-effector position is planned to follow a circle with a constant rotatating velocity and the orientation is supposed to stay in its initial value. We use the first order linear system for the smooth transition of the barycentric coordinates and the parameters are set as follows: $k_0 = 5$, $d_b = 0.3$m, $d_c = 0.2$m, $t_1 = 6$s, and $t_f = 12$s. The obstacle avoidance control is implemented based on [5] with the nominal velocity $v_0 = 0.1$m/s and the rotation vector is used for the orientation representation. Also, the closed-loop inverse kinematics is used with the feedback gain $k_{CLIK} = 5$.

Fig. 4 shows paths that the end-effector and the closest-point to the obstacle traversed. The original path was modified when the manipulator was near the obstacle while the accuracy was preserved in other places. This result can be clearly observed in Fig. 5 where parts A and B are the periods when the modification occurred in the only lowest-priority task. During these periods, the minimum distance between the manipulator and the obstacle was maintained near the critical distance and the frequent switching of priority appeared. As a result, chattering was generated. Nevertheless, smoothness of the transition has been achieved because the transition has occurred as planned by the first-order linear system. Indeed, the frequent switching of priority stems from two inconsistent task definitions: the desired trajectory of the end-effector violates the obstacle avoidance control. As we mentioned in Section III-B, there is an issue related to convergence of barycentric coordinates and we remain this problem as a future work. Finally, we can check that the joint trajectories are bounded inside of the joint limit.

B. Task Scheduling of a Humanoid Robot

Humanoid robot is a general-purpose system that has capabilities to perform multiple tasks on the same time. Usually, a scenario for the robot consists of sequence of multiple tasks as shown in Fig. 6. When we design scenarios for a complex

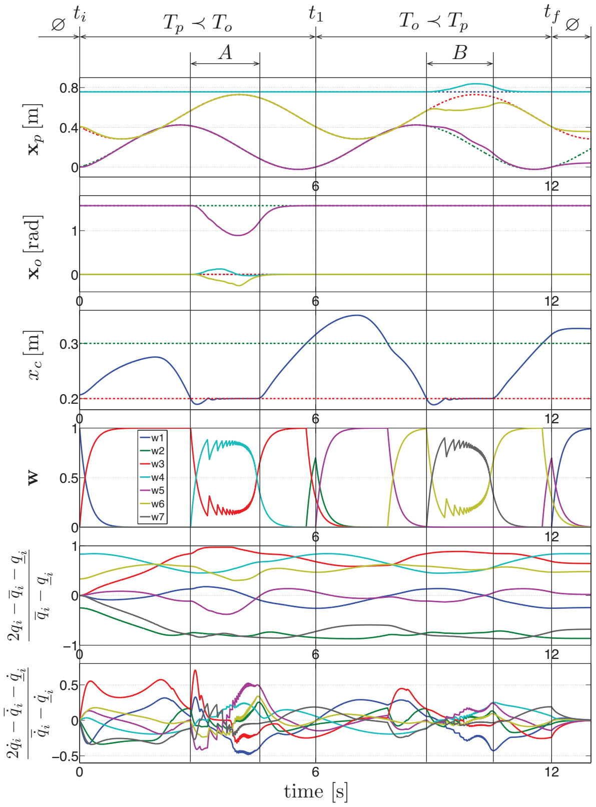


Fig. 5. Simulation results of the TTC with the KUKA LWR. The joint angles and the velocities are rescaled on the region $[-1, 1]$ using the limit values listed on the robot datasheet.

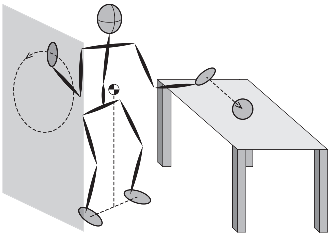


Fig. 6. Scenario: the robot stands while maintaining the projection of the COM on the center of the supporting polygon, drawing a circle on the wall with the right hand, and taking an object on the table with the left hand.

robotic system, it is not easy to check compatibility of all tasks beforehand and also to generate smooth transitions between tasks during operations. We may tackle this problem with the proposed method. For the simulation, we have constructed a 32-DOF humanoid kinematic model with 3-DOF in the waist, 3-DOF in the neck, 7-DOF in each arm, and 6-DOF in each leg. The joint limits are determined manually and it is assumed that 1kg masses are attached on the center of all joints. Based on the scenario in Fig. 6, the basic tasks are determined as COM x and y (T_{BC}); feet relative posture (T_{FR}); torso height and orientation (T_{TO}); right hand (T_{RH}); and left hand (T_{LH}). Both T_{FR} and T_{TO} are assumed to stay on the initial values. The induced tasks are defined as $T^1 = \varnothing$, $T^2 = [(T_{BC}, T_{FR}), T_{TO}]$, $T^3 = [(T_{BC}, T_{FR}), T_{RH}, T_{LH}, T_{TO}]$, and $T^4 = [(T_{BC}, T_{FR}), T_{LH}, T_{RH}, T_{TO}]$. The task schedule is given in Table II with $t_1 = 1$s, $t_2 = 5$s, and $t_f = 11$s.

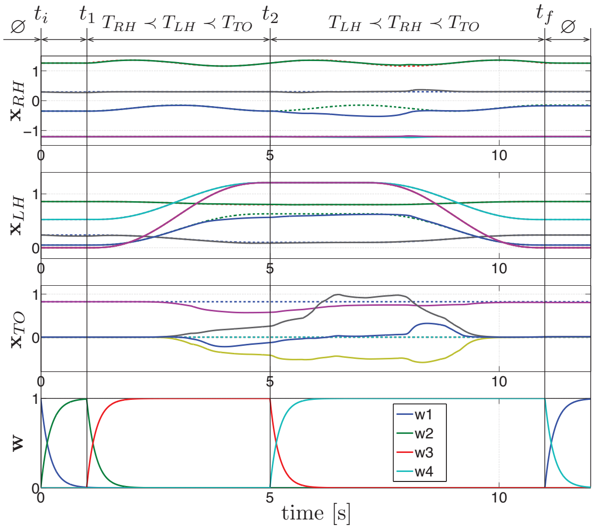


Fig. 7. Simulation results of the TTC with the humanoid.

TABLE II

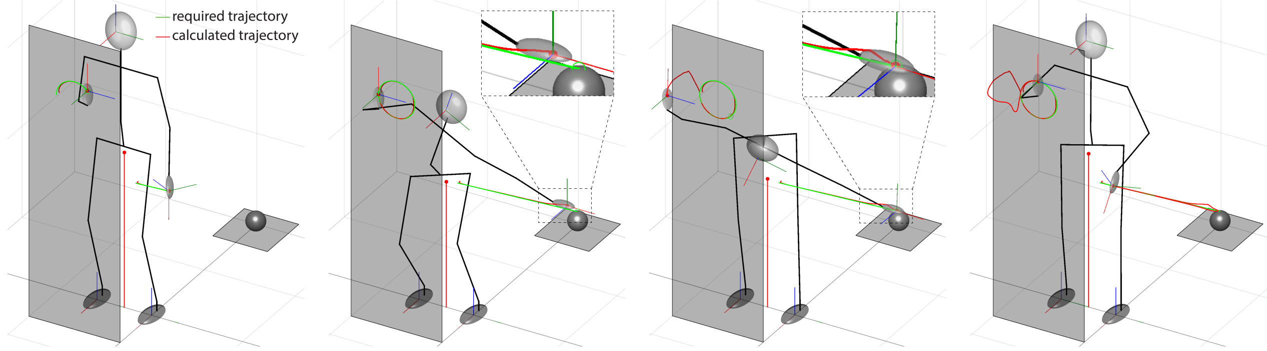
TASK TRANSITION RULE FOR THE HUMANOID ROBOT

$t < t_0$	$t_0 \le t < t_1$	$t_1 \le t < t_2$	$t_2 \le t < t_f$	$t_f \le t$
T^1	T^2	T^3	T^4	T^1

Fig. 7 and Fig. 8 show the simulation results. Due to the limited space, task errors of (T_{BC}, T_{FR}) are not shown. Note that both are performed accurately for the whole operation time. In the beginning, the robot has sufficient DOFs, so all tasks are executed correctly, but, when the left arm is fully stretched toward the object, it cannot move further because of the higher-priority tasks. At $t = t_2$, priority between T_{RH} and T_{LH} is switched; then, the left hand can reach to the object, while the right hand leaves from the desired trajectory to maintain the COM on the center of the supporting polygon. Later on, the robot recovers the lost DOFs when the left hand moves to the initial postures and the right hand can come back to the desired trajectory. The simulation results show that the proposed method allows smooth task transitions and also provides a convenient way to design sequence of tasks.

V. Conclusions

The value of this study is to open new possibilities to handle multiple task definitions more effectively. To find smooth inverse mappings on the existence of discrete task definitions, most of previous studies have focused on the task space in which discontinuity occurs, but we have turned our attention to the joint space in which continuity should be assured. So joint trajectories were interpolated to separate two complicated problems: to find inverse mappings and to make inverse mappings smooth. As a result, the task transition control has been developed that allows smooth, arbitrary, and consecutive transitions between all possible task definitions in a simple, direct, and general manner and verified by two kinematic simulations with the 7-DOF manipulator and the



(a) $t = 2.5$s (b) $t = 5.0$s (c) $t = 7.5$s (d) $t = 10.0$s

Fig. 8. Snapshots of the humanoid robot simulation.

32-DOF humanoid robot. Also, it was successfully applied to a real example of the incremental kinesthetic teaching [19] in which the method enables human to physically interact with the KUKA LWR during operations to iteratively refine learned movements. One of our future plans is to find a way that guarantees convergence of barycentric coordinates when there are conflicting task definitions.

Acknowledgment

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

Appendix I
Proof of Theorem 1

Case 1): For any change in the order of basic subtasks in T^a, there exists a permutation matrix $\mathbf{P}$, such that the UIK of T^b is given by $\dot{\mathbf{q}}^b = (\mathbf{P}\mathbf{J}^a)^\dagger \mathbf{P}\dot{\mathbf{x}}_d^a$ using (1). Let $\mathbf{J}^a \triangleq \mathbf{U}^a\Sigma^a\mathbf{V}^{aT}$ be the SVD; then, we get $(\mathbf{P}\mathbf{J}^a)^\dagger = \mathbf{V}^a\Sigma^{a\dagger}\mathbf{U}^{aT}\mathbf{P}^T$ from $\mathbf{P}^{-1} = \mathbf{P}^T$ and therefore $\dot{\mathbf{q}}^b = (\mathbf{V}^a\Sigma^{a\dagger}\mathbf{U}^{aT})(\mathbf{P}^T\mathbf{P})\dot{\mathbf{x}}_d^a = \mathbf{J}^{a\dagger}\dot{\mathbf{x}}_d^a = \dot{\mathbf{q}}^a$.

Case 2): Let $\mathbf{W}^a \triangleq \mathbf{J}^{aT}\mathbf{J}^a + \delta^2\mathbf{I}_n = \tilde{\mathbf{R}}^{aT}\tilde{\mathbf{R}}^a$ be the CLD, $\mathbf{J}_R^a \triangleq \mathbf{J}^a\tilde{\mathbf{R}}^{a-1} = \mathbf{C}^a\hat{\mathbf{J}}^a$ be the QRD of $\mathbf{J}_R^{aT}$, and suppose $\forall t \in [t_0, t_f]$, $\operatorname{rank}(\mathbf{J}^a) = m^a \triangleq m_{a1} + \cdots + m_{a\alpha}$; then, $\mathbf{C}_D^{a\ddagger} = \mathbf{C}_D^{a-1}$, $\mathbf{J}_R^{a\dagger} = \mathbf{J}_R^{aT}(\mathbf{J}_R^a\mathbf{J}_R^{aT})^{-1} = \hat{\mathbf{J}}^{aT}\mathbf{C}^{a-1}$, and the PIK given by (5) can be rewritten as $\dot{\mathbf{q}}_R^a \triangleq \tilde{\mathbf{R}}^a\dot{\mathbf{q}}^a = \hat{\mathbf{J}}^{aT}\mathbf{C}^{a-1}\dot{\mathbf{x}}_d^a = \mathbf{J}_R^{a\dagger}\dot{\mathbf{x}}_d^a$ which is the solution of the UIK in the transformed coordinate by $\tilde{\mathbf{R}}^a$. Consider a permutation matrix $\mathbf{P}$ similarly to the case 1); then, $\mathbf{W}^b = (\mathbf{P}\mathbf{J}^a)^T(\mathbf{P}\mathbf{J}^a) + \delta^2\mathbf{I}_n = \mathbf{W}^a$ or $\tilde{\mathbf{R}}^b = \tilde{\mathbf{R}}^a$ which means that the coordinate transformation is invariant to the permutation of basic subtasks. Therefore, $\dot{\mathbf{q}}^b = \tilde{\mathbf{R}}^{b-1}\dot{\mathbf{q}}_R^b = \tilde{\mathbf{R}}^{b-1}(\mathbf{P}\mathbf{J}_R^a)^\dagger\mathbf{P}\dot{\mathbf{x}}_d^a = \tilde{\mathbf{R}}^{b-1}\mathbf{J}_R^{a\dagger}\dot{\mathbf{x}}_d^a = \tilde{\mathbf{R}}^{a-1}\dot{\mathbf{q}}_R^a = \dot{\mathbf{q}}^a$. If the PIK is given by (3), then we can let $\mathbf{W} = \mathbf{I}_n$.

Case 3): Suppose T^a is the i-th subtask of T^b and let $\mathbf{P}$ be a permutation matrix for T^c, $\dot{\mathbf{x}}^b = \mathbf{J}_R^b\dot{\mathbf{q}}_R$ be the coordinate transformation by $\tilde{\mathbf{R}}^b$, and $\mathbf{J}_R^b = \mathbf{C}^b\hat{\mathbf{J}}^b$ be the QRD of $\mathbf{J}_R^{bT}$; then, T^d is given as

$$\begin{bmatrix} \dot{\mathbf{x}}_{d,1}^d \\ \vdots \\ \dot{\mathbf{x}}_{d,i}^d \\ \vdots \\ \dot{\mathbf{x}}_{d,k^d} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{x}}_{d,b_1} \\ \vdots \\ \mathbf{P}\dot{\mathbf{x}}_d^a \\ \vdots \\ \dot{\mathbf{x}}_{d,b_\beta} \end{bmatrix} \text{ and } \begin{bmatrix} \dot{\mathbf{x}}_1^d \\ \vdots \\ \dot{\mathbf{x}}_i^d \\ \vdots \\ \dot{\mathbf{x}}_{k^d}^d \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{R,b_1}\dot{\mathbf{q}}_R \\ \vdots \\ \mathbf{P}\mathbf{J}_R^a\dot{\mathbf{q}}_R \\ \vdots \\ \mathbf{J}_{R,b_\beta}\dot{\mathbf{q}}_R \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{11}^{bA}\hat{\mathbf{J}}_1^{bA}\dot{\mathbf{q}}_R \\ \vdots \\ \mathbf{P}\mathbf{C}_{ii}^{bA}\hat{\mathbf{J}}_i^{bA}\dot{\mathbf{q}}_R \\ \vdots \\ \mathbf{C}_{k^bk^b}^{bA}\hat{\mathbf{J}}_{k^b}^{bA}\dot{\mathbf{q}}_R \end{bmatrix}$$

where $\mathbf{C}_{ij}^{bA}$ and $\hat{\mathbf{J}}_i^{bA}$ are defined in the same manner of Proposition 1. Consider the recursive formulation of the PIK given by (4) and remind $\tilde{\mathbf{R}}^d = \tilde{\mathbf{R}}^b$; then, $\dot{\mathbf{q}}_j^d = \dot{\mathbf{q}}_j^b$, $\forall j \in \mathbb{N}_{\leq i-1}$ because the recursions up to $i-1$ are not affected by the order change of T^a. Also, the i-th recursion becomes $\dot{\mathbf{q}}_i^d = \dot{\mathbf{q}}_{i-1}^d + \tilde{\mathbf{R}}^{b-1}\hat{\mathbf{J}}_i^T(\mathbf{P}\mathbf{C}_{ii}^b)^\dagger(\mathbf{P}\dot{\mathbf{x}}_d^a - \mathbf{P}\mathbf{C}_{i,i-1}^{bA}\hat{\mathbf{J}}_{i-1}^{bA}\dot{\mathbf{q}}_{i-1}^d) = \dot{\mathbf{q}}_i^b$ because $\dot{\mathbf{q}}_{i-1}^d = \dot{\mathbf{q}}_{i-1}^b$ and $(\mathbf{P}\mathbf{C}_{ii}^b)^\dagger\mathbf{P} = \mathbf{C}_{ii}^{b\dagger}\mathbf{P}^T\mathbf{P} = \mathbf{C}_{ii}^{b\dagger}$; therefore, the recursions up to the k^d-th step are identical and we get $\dot{\mathbf{q}}^d \triangleq \dot{\mathbf{q}}_{k^d}^d = \dot{\mathbf{q}}_{k^b}^b \triangleq \dot{\mathbf{q}}^b$. We can let $\tilde{\mathbf{R}}^b = \mathbf{I}_n$ for (2).

Appendix II
Proof of Theorem 2

The proof consists of three parts. First, we prove that the barycentric coordinates of a convex polytope is an internal point. Then, we show that the barycentric coordinates of a set of points is an internal point of the minimum enclosing convex polytope. Lastly, we find the boundedness condition.

Part 1: Suppose that $\mathscr{Q} \triangleq \{\dot{\mathbf{q}}^1, \cdots, \dot{\mathbf{q}}^l\} \subset \mathbb{R}^n - \varnothing$ is a set of vertices of a convex polytope $\mathscr{P} \subset \mathbb{R}^n$; then, by the definition of convexity, $\forall \dot{\mathbf{q}}_a, \dot{\mathbf{q}}_b \in \mathscr{P}$ and $\forall w_a, w_b \in \mathbb{R}^{\geq 0}$ with $w_a + w_b = 1$, we get $w_a\dot{\mathbf{q}}_a + w_b\dot{\mathbf{q}}_b \in \mathscr{P}$ which means that a line segment connecting two internal points is also included by the convex polytope. $\forall w^1, \cdots, w^l \in \mathbb{R}^{\geq 0}$ with $\sum_{i=1}^l w^i = 1$, we can assume that $\dot{\mathbf{q}}^1$ and $\dot{\mathbf{q}}^2$ are two adjacent vertices without loss of generality; then, $\dot{\mathbf{q}}$ with the barycentric coordinates can be rewritten by $\dot{\mathbf{q}} = \sum_{i=3}^l w^i\dot{\mathbf{q}}^i$ if $w^1 + w^2 = 0$ and $\dot{\mathbf{q}} = \bar{w}^1\dot{\bar{\mathbf{q}}}^1 + \sum_{i=3}^l w^i\dot{\mathbf{q}}^i$ if $w^1 + w^2 > 0$ where $\bar{w}^1 \triangleq w^1 + w^2$ and $\dot{\bar{\mathbf{q}}}^1 \triangleq w^1/\bar{w}^1\dot{\mathbf{q}}^1 + w^2/\bar{w}^1\dot{\mathbf{q}}^2$. In both cases, new coordinates form the barycentric cooridinates of $\dot{\mathbf{q}}$ with the smaller number of points because $\sum_{i=3}^l w^i = 1$ if $w^1 + w^2 = 0$ and $\bar{w}^1 + \sum_{i=3}^l w^i = 1$ if $w^1 + w^2 > 0$. Also, a polytope generated by the smaller number of vertices is convex enclosed by $\mathscr{P}$ because $\dot{\bar{\mathbf{p}}}$ is a point on the edge connecting two adjacent vertices $\dot{\mathbf{q}}^1$ and $\dot{\mathbf{q}}^2$ ($w^1/\bar{w}^1 \geq 0$, $w^2/\bar{w}^2 \geq 0$, $w^1/\bar{w}^1 + w^2/\bar{w}^2 = 1$); therefore, we can repeat this process until only one point remains inside of $\mathscr{P}$.

Part 2: $\forall \mathscr{Q} \subset \mathbb{R}^n - \varnothing$, $\exists A \triangleq \{a_1, \cdots, a_\alpha\} \subset \mathbb{N}_{\leq l}$, such that $\mathscr{Q}_A \triangleq \{\dot{\mathbf{q}}^{a_1}, \cdots, \dot{\mathbf{q}}^{a_\alpha}\} \subset \mathscr{Q}$ is a set of vertecies of the minimum enclosing convex polytope $\mathscr{P} \subset \mathbb{R}^n$ of $\mathscr{Q}$. Let $B \triangleq \{b_1, \cdots, b_\beta\} = \mathbb{N}_{\leq l} - A$; then, $\mathscr{Q}_B \triangleq \{\dot{\mathbf{q}}^{b_1}, \cdots, \dot{\mathbf{q}}^{b_\beta}\} \subset \mathscr{Q}$ becomes a set of points enclosed by $\mathscr{P}$ and $\dot{\mathbf{q}}$ with the barycentric coordinates can be rewritten by $\dot{\mathbf{q}} = \sum_{i=1}^\beta w^{b_i}\dot{\mathbf{q}}^{b_i}$

if $\sum_{i=1}^{\alpha} w^{a_i} = 0$ and $\dot{\mathbf{q}} = \sum_{i=1}^{\beta+1} w^{b_i}\dot{\mathbf{q}}^{b_i}$ if $\sum_{i=1}^{\alpha} w^{a_i} > 0$ where $w^{b_{\beta+1}} \triangleq \sum_{i=1}^{\alpha} w^{a_i}$ and $\dot{\mathbf{q}}^{b_{\beta+1}} \triangleq \sum_{i=1}^{\alpha} w^{a_i}/w^{b_{\beta+1}}\dot{\mathbf{q}}^{a_i}$. In both cases, new coordinates form the barycentric coordinates of $\dot{\mathbf{q}}$ with the smaller number of points enclosed by $\mathscr{P}$ by definition and we can repeat this process until only one point remains inside of $\mathscr{P}$.

Part 3: $\forall \mathscr{Q} \subset \mathbb{R}^n - \varnothing$, there exist $r \in \mathbb{R}^{\geq 0}$ and the minimum enclosing convex polytope $\mathscr{P} \subset \mathbb{R}^n$, such that $\mathscr{P} \subset \{\mathbf{a} \in \mathbb{R}^n : \|\mathbf{a}\| \leq r\}$ and the minimum r is given by $\max\{\|\dot{\mathbf{q}}^1\|, \cdots, \|\dot{\mathbf{q}}^l\|\}$. Therefore, based on the proof of the second part, $\forall w^1, \cdots, w^l \in \mathbb{R}^{\geq 0}$ with $\sum_{i=1}^{l} w^i = 1$, we get $\dot{\mathbf{q}} = \sum_{i=1}^{l} w^i\dot{\mathbf{q}}^i \in \mathscr{P} \subset \{\mathbf{a} \in \mathbb{R}^n : \|\mathbf{a}\| \leq \max\{\|\dot{\mathbf{q}}^1\|, \cdots, \|\dot{\mathbf{q}}^l\|\}$.

Appendix III
Joint Space Representation of Lee [8]'s method

Substitute (7) and (8) into (6) and rearrange it as:

$$\begin{aligned}
\dot{\mathbf{q}} &= h_1\mathbf{J}_1^\dagger\dot{\mathbf{x}}_{d,1} + h_2\mathbf{A}\dot{\mathbf{x}}_{d,2} + h_1h_2(\mathbf{B}\dot{\mathbf{x}}_{d,1} + \mathbf{C}\dot{\mathbf{x}}_{d,2})\\
\mathbf{A} &= \mathbf{J}_2^\dagger - \mathbf{N}_1\mathbf{J}_2^T\left[(\mathbf{J}_2\mathbf{J}_2^T)^\dagger - (\mathbf{J}_2\mathbf{N}_1\mathbf{J}_2^T)^\dagger(\mathbf{I} - \mathbf{J}_2\mathbf{J}_1^\dagger\mathbf{J}_1\mathbf{J}_2^\dagger)\right]\\
\mathbf{B} &= -(\mathbf{J}_2\mathbf{N}_1)^\dagger\mathbf{J}_2\mathbf{J}_1^\dagger\\
\mathbf{C} &= -\mathbf{J}_1^\dagger\mathbf{J}_1\mathbf{J}_2^\dagger + (\mathbf{J}_2\mathbf{N}_1)^\dagger\mathbf{J}_2\mathbf{J}_1^\dagger\mathbf{J}_1\mathbf{J}_2^\dagger.
\end{aligned}$$

If we assume that $[\mathbf{J}_1^T \ \mathbf{J}_2^T]^T$ has its full rank, $\mathbf{A}$ and $\mathbf{C}$ can be simplified to:

$$\begin{aligned}
\mathbf{A} &= \mathbf{J}_2^\dagger - \mathbf{N}_1\mathbf{J}_2^T\left[(\mathbf{J}_2\mathbf{J}_2^T)^\dagger - (\mathbf{J}_2\mathbf{N}_1\mathbf{J}_2^T)^\dagger(\mathbf{J}_2\mathbf{N}_1\mathbf{J}_2^\dagger)\right] = \mathbf{J}_2^\dagger\\
\mathbf{C} &= \mathbf{N}_1\mathbf{J}_2^\dagger - \mathbf{J}_2^\dagger + (\mathbf{J}_2\mathbf{N}_1)^\dagger(\mathbf{I} - \mathbf{J}_2\mathbf{N}_1\mathbf{J}_2^\dagger) = (\mathbf{J}_2\mathbf{N}_1)^\dagger - \mathbf{J}_2^\dagger
\end{aligned}$$

and $\dot{\mathbf{q}}$ can be rewritten as

$$\begin{aligned}
\dot{\mathbf{q}} &= h_1(1-h_2)\mathbf{J}_1^\dagger\dot{\mathbf{x}}_{d,1} + h_2(1-h_1)\mathbf{J}_2^\dagger\dot{\mathbf{x}}_{d,2}\\
&\quad + h_1h_2\left[\mathbf{J}_1^\dagger\dot{\mathbf{x}}_{d,1} + (\mathbf{J}_2\mathbf{N}_1)^\dagger(\dot{\mathbf{x}}_{d,2} - \mathbf{J}_2\mathbf{J}_1^\dagger\dot{\mathbf{x}}_{d,1})\right]\\
&= h_1(1-h_1)\dot{\mathbf{q}}_1 + h_2(1-h_2)\dot{\mathbf{q}}_2 + h_1h_2\dot{\mathbf{q}}_{[1,2]} + w_\varnothing\dot{\mathbf{q}}_\varnothing
\end{aligned}$$

where $w_\varnothing\dot{\mathbf{q}}_\varnothing = \mathbf{0}$, $\forall w_\varnothing \in \mathbb{R}$.

Appendix IV
Proof of Theorem 3

Let's define $\Delta^{l-1} \triangleq \{\mathbf{a} \in \mathbb{R}^l : \mathbf{1}^T\mathbf{a} = 1, \ \mathbf{a} \geq \mathbf{0}\}$ as the standard simplex in $\mathbb{R}^l$; then, the vertices of Δ^{l-1} are given by $\{\hat{\mathbf{e}}^1, \cdots, \hat{\mathbf{e}}^l\}$ and, $\forall(\mathbf{w}, i) \in \Delta^{l-1} \times \mathbb{N}_{\leq l}$, $\hat{\mathbf{e}}^i - \mathbf{w}$ is a tangent vector of Δ^{l-1}. From the initial condition $\mathbf{w}(t_0) \in \Delta^{l-1}$, $\mathbf{w}^{(j)}(t_0) = \mathbf{0}$ and the input condition $\mathbf{w}_d \in \{\hat{\mathbf{e}}^1, \cdots, \hat{\mathbf{e}}^l\}$, we get $\mathbf{w}^{(s+1)}(t_0) = k_p(\mathbf{w}_d(t_0) - \mathbf{w}(t_0))$ which is also a tangent vector of Δ^{l-1}. Therefore, the variation of $\mathbf{w}^{(j)}$ becomes parallel to Δ^{l-1} right after the initial time and $\mathbf{w}$ remains on the hyperplain $\mathscr{H} \triangleq \{\mathbf{a} \in \mathbb{R}^l : \mathbf{1}^T\mathbf{a} = 1\}$ because (12) does not generate normal directional variations once $\mathbf{w}^{(j)}$ and $\mathbf{w}_d - \mathbf{w}$ are parallel to Δ^{l-1}.

Consider the *i*-th linear system of (12), $d^{s+1}w^i/dt^{s+1} + k_sd^sw^i/dt^s + \cdots + k_0w^i = k_0w_d^i$, that corresponds to the *i*-th varycentric coordinate where $w_d^i \in \{0, 1\}$. Since we assumed that $\{k_0, \cdots, k_k\} \subset \mathbb{R}$ are stabilizing control gains that do not generate overshoot, any discontinuous variations of w_d^i will not make w^i leave the boundary $[0, 1]$; therefore, $\mathbf{w} \in \Delta^{l-1} \subset \mathscr{H}$ for all arbitrary and consecutive changes of $\mathbf{w}_d$ and, by Theorem 2, the inverse solution is bounded such that $\|\dot{\mathbf{q}}\| \leq \max\{\|\dot{\mathbf{q}}^1\|, \cdots, \|\dot{\mathbf{q}}^l\|\}$.

In (12), all discontinuities occur in the $(s+1)$-th derivative of $\mathbf{w}$, so continuity is assured for all lower derivatives. It guarantees smooth task transitions.

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] 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.

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

[5] 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.

[6] T. Petrič and L. Žlajpah, "Smooth continuous transition between tasks on a kinematic control level: Obstacle avoidance as a control problem," *Robotics and Autonomous Systems*, vol. 61, pp. 948–959, Sept. 2013.

[7] N. Mansard, O. Khatib, and A. Kheddar, "A Unified Approach to Integrate Unilateral Constraints in the Stack of Tasks," *IEEE Transactions on Robotics*, vol. 25, pp. 670–685, June 2009.

[8] J. Lee, N. Mansard, and J. Park, "Intermediate Desired Value Approach for Task Transition of Robots in Kinematic Control," *IEEE Transactions on Robotics*, vol. 28, pp. 1260–1277, Dec. 2012.

[9] F. Keith, P.-b. Wieber, N. Mansard, and A. Kheddar, "Analysis of the Discontinuities in Prioritized Tasks-Space Control Under Discreet Task Scheduling Operations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3887–3892, 2011.

[10] K. Hu, C. Ott, and D. Lee, "Online Human Walking Imitation in Task and Joint Space based on Quadratic Programming," in *IEEE International Conference on Robotics and Automation*, 3458–3464, 2014.

[11] J. Warren, "Barycentric coordinates for convex polytopes," *Advances in Computational Mathematics*, vol. 6, pp. 97–108, 1996.

[12] M. Meyer, A. Barr, H. Lee, and M. Desbrun, "Generalized Barycentric Coordinates on Irregular Polygons," *Journal of Graphics Tools*, vol. 7, pp. 13–22, Jan. 2002.

[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] S.-i. An and D. Lee, "Prioritized Inverse Kinematics using QR and Cholesky Decompositions," in *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 5062–5069, 2014.

[16] 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.

[17] 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.

[18] T. F. Chan and R. V. Dubey, "A Weighted Least-Norm Solution Based Scheme for Avoiding Joint Limits for Redundant Joint Manipulators," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 2, pp. 286–292, 1995.

[19] M. Saveriano, S.-i. An, and D. Lee, "Incremental Kinesthetic Teaching of End-Effector and Null-Space Motion Primitives," in *IEEE International Conference on Robotics and Automation*, 2015.