Continuous Task Transition Approach for Robot Controller based on Hierarchical Quadratic Programming

Sanghyun Kim, Keunwoo Jang, Suhan Park, Yisoo Lee, Sang Yup Lee, Jaeheung Park

Continuous Task Transition Approach for Robot Controller based on Hierarchical Quadratic Programming

Sanghyun Kim[1,2], Keunwoo Jang[1], Suhan Park[1], Yisoo Lee[1,3], Sang Yup Lee[1], and Jaeheung Park[1,4]

***Abstract*—The robots with high Degrees of Freedom (DoF) such as humanoids and mobile manipulators are expected to perform multiple tasks simultaneously. Hierarchical Quadratic Programming (HQP) can effectively compute a solution for strictly prioritized tasks. However, the continuity of control input is not guaranteed when the priorities of the tasks are modified during operation. This paper proposes a continuous task transition method for HQP based controller to insert, remove, and swap arbitrary tasks without discontinuity. Smooth task transition is assured because our approach uses activation parameters of the new and existing tasks without modifying control structure. The proposed approach is applied to various task transition scenarios including joint limit, singularity, and obstacle avoidance to guarantee the stable execution of the robot. The proposed control scheme was implemented on a 7-DoF robotic arm, and its performance was demonstrated by the continuity of control input during various task transition scenarios.**

***Index Terms*—Motion Control, Redundant Robots, Optimization and Optimal Control, Manipulation Planning**

I. INTRODUCTION

THE robots with high Degrees of Freedom (DoF) such as humanoids and wheeled mobile manipulators can be used in various fields including a daily-life assistance and a disaster area. Therefore, many studies for controlling these robots have concentrated on performing various tasks simultaneously. Especially, since the pioneering work of Siciliano and Slotine proposing the recursive formulation of n tasks for the inverse kinematics [1], the hierarchical controllers have been actively studied to handle multiple tasks with strict priorities, Stack of Tasks (SoT) [2]–[5].

These control schemes with a predefined SoT can calculate control input without conflict between prioritized tasks. However, the robot needs to deal with dynamically changing SoT in order to perform complex tasks effectively. The sudden task transition causes the discontinuity of the control input, which can adversely affect stability and durability of the robot. Especially, when the avoidance tasks such as the joint limit avoidance task and the obstacle avoidance tasks are added to or removed from the existing SoT, the chattering of the robot may be induced by the discontinuity [6].

Hence, in this paper, we propose a novel continuous task transition strategy for a high DoF robot in a hierarchical controller to handle complex tasks of the robot more effectively.

A. Related Works

The task transition methods for continuous control input have been developed in the inverse kinematics and dynamics controllers. For the inverse kinematics controller, the linear interpolation method between the solutions of an existing SoT and a new SoT was proposed in [7]. Although this approach is easy to implement, it is necessary to obtain solutions for different SoTs at the same time during the transition period. Also, the intermediate desired value approach in the task space was proposed in [8]. The main idea of this algorithm is that the desired value of the task for inserting or removing is modified without changing the control structure. Although this method allows the continuous transition in multiple priority tasks, the computational cost increases drastically depending on the number of tasks. For example, $n!$ operations of pseudo-inverse are needed for n prioritized tasks. Also, it is difficult to handle inequality constraints. Jarquín *et al.* proposed a continuous task transition strategy with a controller based on Hierarchical Quadratic Programming (HQP) [9]. The proposed method can swap the priorities of two consecutive tasks by merging both priority levels in transition and modifying the weight of each slack. Although this algorithm does not increase the computational cost, it cannot handle inequality constraints and cannot ensure the priorities of tasks during the transition phase.

On the other hand, for the inverse dynamics controller, the intermediate desired value approach of [8] was further developed to be applied in the operational space control framework [6]. Although this method enables an effective and stable transition in the operational space, it has the same disadvantages as the intermediate desired value approach in the inverse kinematics [8]. Recently, Liu *et al.* proposed a hierarchical controller based on Linear Quadratic Programming (LQP) and a generalized projector for null space smoothing [10].

Manuscript received: September, 10, 2018; Revised December, 14, 2018; Accepted January, 19, 2019.

This paper was recommended for publication by Editor Paolo Rocco upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by Industrial Strategic Technology Development Program (No. 10077538) funded by the Ministry of Trade, Industry & Energy (MI, Korea).

[1]Sanghyun Kim, Keunwoo Jang, Suhan Park, Yisoo Lee, Sang Yup Lee, and Jaeheung Park are with Graduate School of Convergence Science and Technology, Seoul National University, Suwon, Republic of Korea. (ggory15, jkw0701, psh117, howcan11, sangyup8378, park73)@snu.ac.kr

[2]Sanghyun Kim is also with CNRS, LAAS, Toulouse, France.

[3]Yisoo Lee is now with IIT, Gevoa, Italy.

[4]Jaeheung Park is also with Advanced Institutes of Convergence Technology (AICT), Suwon, Korea. He is a corresponding author of this paper.

Digital Object Identifier: see top of this page

By increasing or decreasing the activation parameter of the generalized projector, this scheme could calculate continuous trajectories during task transitions. However, there may be no feasible solutions depending on the state of the robot.

B. Overview of Our Approach

In this paper, we propose a continuous task transition strategy of the robot controller based on the HQP for inserting, removing, and swapping arbitrary equality and inequality tasks. HQP which is one of the constrained Quadratic Programming (QP) was proposed to treat not only equality constraints but also inequality constraints of prioritized tasks. The main characteristic of the HQP is that the lower priority task cannot affect the higher priority tasks by solving the cascade of quadratic programming with slack variables [5], [11]. Although there are analytical approaches such as the task-priority inverse kinematics [12], [13] and the saturation in the null space method [14] for treating inequality constraints, these concepts can treat inequality constraints only at the joint level and are hard to handle two or more tasks in the same priority level [10], [15]. On the contrary, since the HQP can deal with bilateral inequality constraints and is easy to implement, it has been adopted to high DoF robots including humanoids [16], underwater robots [17], and dual-arm manipulators [18].

However, the HQP framework also derives the discontinuous inputs when the SoT is changed. To prevent this problem, our framework proposes the activation parameter which interpolates the feasible solution areas between the existing SoT and the new SoT. Specifically, the main role of the activation parameter in this paper is to modify the bounded interval and add the offset of the feasible solution area for the new SoT in order to change smoothly from the existing SoT to the new SoT. Thus, our algorithm ensures continuous task transition during the change of the SoT in real-time.

The main advantages of the proposed control framework in this paper are as follows. First, our approach with the activation parameter can handle not only equality constraints but also inequality constraints during the transition. Next, the proposed scheme can be applied to not only the inverse kinematics problem but also the inverse dynamics problem without modifying control structure. Finally, our method deals with continuous transition between non-consecutive tasks as well as consecutive tasks.

Based on the proposed strategy, we show that our algorithm can be applied to various task transition scenarios including joint limit, singularity, and obstacle avoidance to guarantee the stable execution of the robot. Through various experiments with the 7-DoF manipulator, we could ensure continuous reference torque during the task transition.

The remainder of this paper is as follows. Section II reviews the HQP controller for inverse kinematics and dynamics. Next, we present the strategy for ensuring continuous task transition in Sec. III. Section IV presents various applications including joint limit, singularity, and obstacle avoidance algorithms using the proposed control strategy and Section V describes the experimental validations of the proposed method. Finally, the paper is concluded in Sec. VI.

TABLE I. Notation and symbols

Symbol	Description
$\mathscr{T}_i \in \mathbb{R}^{m_i}$	m_i-dimensional equality or inequality task
$\underline{\dot{x}}_{d_i}$	lower bound of i-th inequality task
$\bar{\dot{x}}_{d_i}$	upper bound of i-th inequality task
$[\underline{\dot{x}}_{d_i}, \bar{\dot{x}}_{d_i}]$	inequality bound set of i-th task
w_i	i-th slack variable
w_i^*	optimal value of i-th slack variable
$\mathscr{T}_i \prec \mathscr{T}_{i+1}$	$\mathscr{T}_i$ has higher priority than $\mathscr{T}_{i+1}$

II. HIERARCHICAL QUADRATIC PROGRAMMING: A REVIEW

In this section, we briefly review a basic formulation of HQP for the inverse kinematics and inverse dynamics control. To enhance readability, Table I denotes the symbols and their corresponding meanings in this paper.

Firstly, let's consider the inverse kinematics problem of an n-DoF robot. If considering a m_1-dimensional single task, the Jacobian based inverse kinematics solution is,

$$\dot{q}^* = J_1^+ \dot{x}_{d_1}, \tag{1}$$

where $\dot{q}^* \in \mathbb{R}^n$, $J_1^+ \in \mathbb{R}^{n\times m_1}$, and $\dot{x}_{d_1} \in \mathbb{R}^{m_1}$ are the desired joint velocity, the pseudo inverse of task Jacobian matrix (J_1), and the desired velocity in the task space defined by J_1, respectively. This is in fact a solution of the following optimization.

$$\min_{\dot{q}} \quad \|J_1\dot{q} - \dot{x}_{d_1}\|_2. \tag{2}$$

Next, when considering two tasks with priorities ($\mathscr{T}_1 \prec \mathscr{T}_2$), (1) can be extended as,

$$\dot{q}^* = J_1^+\dot{x}_{d_1} + (J_2N_1)^+(\dot{x}_{d_2} - J_2J_1^+\dot{x}_{d_1}), \tag{3}$$

where $N_1 = I - J_1^+J_1 \in \mathbb{R}^{n\times n}$ is the null space projection matrix of J_1. $J_2 \in \mathbb{R}^{m_2\times n}$ and $x_{d_2} \in \mathbb{R}^{m_2}$ are the task Jacobian matrix and the desired velocity of $\mathscr{T}_2 \in \mathbb{R}^{m_2}$ [8]. The solution of (3) always satisfies the hierarchy of $\mathscr{T}_1 \prec \mathscr{T}_2$, since the solution for $\mathscr{T}_2$ is calculated in the null space of $\mathscr{T}_1$. This process can be obtained by a QP formulation as,

$$\begin{aligned} \min_{\dot{q},w_2} \quad & \|w_2\|_2, \\ \text{s. t.} \quad & J_2\dot{q} + w_2 = \dot{x}_{d_2} \\ & J_1\dot{q} + w_1^* = \dot{x}_{d_1} \end{aligned} \tag{4}$$

where $w_2 \in \mathbb{R}^{m_2}$ is a slack variable for $\mathscr{T}_2$ which is used to relax the infeasible constraints in the $\mathscr{T}_2$ and it is used to solve the objective function, even if there is no feasible solution to $\mathscr{T}_2$. And $w_1^* \in \mathbb{R}^{m_1}$ is the optimal slack variable of QP formulation for $\mathscr{T}_1$. Note that w_1^* is zero when a feasible solution of $\mathscr{T}_1$ can be obtained. Thus, the solution of (4) is the same as that of (3) if there is a feasible solution to satisfy both tasks. Also, when the nullity of J_1 is empty, the lower priority task, $\mathscr{T}_2$, is totally ignored by w_2. The solution at this time is equivalent to that of (2).

More generally, when considering k tasks of inequality constraints with priorities, a solution can be obtained by the following HQP formulation.

$$\begin{aligned} \min_{\dot{q},w_k} \quad & \|w_k\|_2. \\ \text{s. t.} \quad & \underline{\dot{x}}_{d_k} \le J_k\dot{q} + w_k \le \bar{\dot{x}}_{d_k} \\ & \underline{\dot{x}}_{d_i} \le J_i\dot{q} + w_i^* \le \bar{\dot{x}}_{d_i}, \forall i \in 1,\ldots,k-1 \end{aligned} \tag{5}$$

With the same approach, the HQP formulation for inverse dynamics with a single task can be derived as,

$$\begin{aligned} \min_{\ddot{q},\tau,w} \quad & \|w\|_2, \\ \text{s. t.} \quad & M\ddot{q}+C\dot{q}+g=\tau \\ & J\ddot{q}+\dot{J}\dot{q}+w=\ddot{x}_d \end{aligned} \tag{6}$$

where $\tau \in \mathbb{R}^n$, $M \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{n\times n}$, and $g \in \mathbb{R}^n$ are the joint torque vector, the inertia matrix, Coriolis and centrifugal matrix, and gravity vector of the robot, respectively. If there is a feasible solution, (6) is equivalent to the following equation [16].

$$\ddot{x}_d - \dot{J}\dot{q} + JM^{-1}(C\dot{q}+g) = JM^{-1}\tau. \tag{7}$$

Consequently, the general formulation of HQP for inverse dynamics problem is represented as,

$$\begin{aligned} \min_{\ddot{q},\tau,w_k} \quad & \|w_k\|_2. \\ \text{s. t.} \quad & M\ddot{q}+C\dot{q}+g=\tau \\ & \underline{\ddot{x}}_{d_k} - \dot{J}_k\dot{q} \le J_k\ddot{q}+w_k \le \bar{\ddot{x}}_{d_k} - \dot{J}_k\dot{q} \\ & \underline{\ddot{x}}_{d_i} - \dot{J}_i\dot{q} \le J_i\ddot{q}+w_i^* \le \bar{\ddot{x}}_{d_i} - \dot{J}_i\dot{q}, \forall i \in 1,\ldots,k-1 \end{aligned} \tag{8}$$

III. TASK TRANSITION STRATEGY

In this section, we introduce the continuous task transition strategy for the HQP based controller. In this paper, we derive the proposed task transition method with (8), but it can also be applied to (5). The following subsections describe the details of how to insert tasks, remove tasks, and swap tasks, respectively.

A. Insertion or Removal Case

Let's consider that there is a single task, $\mathscr{T}_2$, in the HQP for the inverse dynamics and it can be expressed as,

$$\begin{aligned} \min_{\ddot{q},\tau,w_2} \quad & \|w_2\|_2. \\ \text{s. t.} \quad & M\ddot{q}+C\dot{q}+g=\tau \\ & \underline{\ddot{x}}_{d_2} - \dot{J}_2\dot{q} \le J_2\ddot{q}+w_2 \le \bar{\ddot{x}}_{d_2} - \dot{J}_2\dot{q} \end{aligned} \tag{9}$$

When a new higher priority task, $\mathscr{T}_1$, is inserted, the HQP suddenly changes from (9) to the following equation,

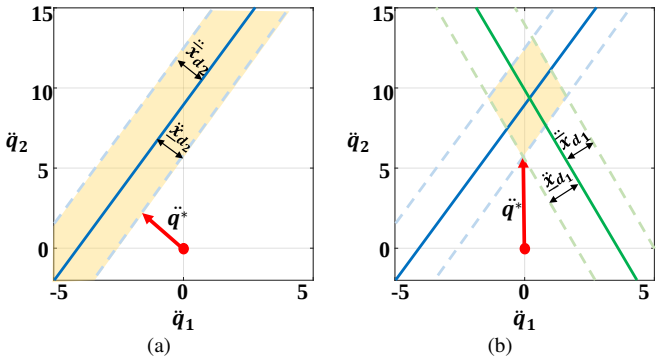
$$\begin{aligned} \min_{\ddot{q},\tau,w_2} \quad & \|w_2\|_2. \\ \text{s. t.} \quad & M\ddot{q}+C\dot{q}+g=\tau \\ & \underline{\ddot{x}}_{d_2} - \dot{J}_2\dot{q} \le J_2\ddot{q}+w_2 \le \bar{\ddot{x}}_{d_2} - \dot{J}_2\dot{q} \\ & \underline{\ddot{x}}_{d_1} - \dot{J}_1\dot{q} \le J_1\ddot{q}+w_1^* \le \bar{\ddot{x}}_{d_1} - \dot{J}_1\dot{q} \end{aligned} \tag{10}$$

Thus, it is obvious that the discontinuity of $\ddot{q}^*$ occurs, as shown in Fig. 1(a) and 1(b).

In this study, the activation parameter, β, of which the value is between 0 and 1 with respect to an activation level, is defined to solve the discontinuity problem. The objective of the activation parameter is to interpolate the solution between the existing SoT and the new SoT by modifying the effect of the existing tasks and the inequality bound of the new task.

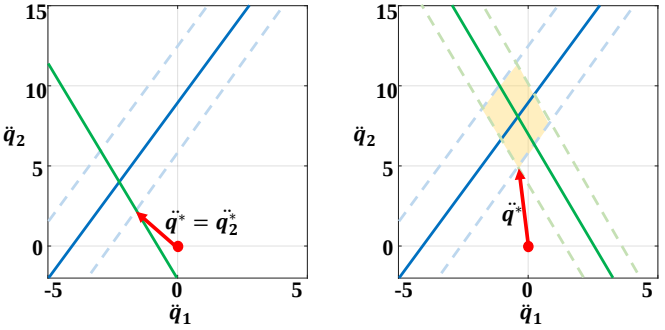
Using the variable β, the proposed continuous transition with inserting higher task is expressed as,

$$\begin{aligned} \min_{\ddot{q},\tau,w_2} \quad & \|w_2\|_2, \\ \text{s. t.} \quad & M\ddot{q}+C\dot{q}+g=\tau \\ & \underline{\ddot{x}}_{d_2} - \dot{J}_2\dot{q} \le J_2\ddot{q}+w_2 \le \bar{\ddot{x}}_{d_2} - \dot{J}_2\dot{q} \\ & \beta(\underline{\ddot{x}}_{d_1} - \dot{J}_1\dot{q}) \le J_1\ddot{q} - (1-\beta)J_1\ddot{q}_2^* + w_1^* \le \beta(\bar{\ddot{x}}_{d_1} - \dot{J}_1\dot{q}) \end{aligned} \tag{11}$$



(a) (b)

Fig. 1. 2D illustration of solution of HQP. Green and blue lines denote solution for $\mathscr{T}_1$ and $\mathscr{T}_2$. A yellow area shows the feasible solution area and a red arrow indicates the minimum l_2 norm solution: (a) illustration of (9), (b) illustration of (10).



(a) (b)

Fig. 2. 2D illustration of solution of HQP with inserting task: (a) illustration of (11) with $\beta = 0$, (b) (11) with $\beta = 0.7$.

where $\ddot{q}_2^*$ is the solution of (9). In (11), the term of $(1-\beta)J_1\ddot{q}_2^*$ is the offset value to move the feasible solution area of (10) closer to that of (9). Also, the terms of $\beta(\underline{\ddot{x}}_{d_1} - \dot{J}_1\dot{q})$ and $\beta(\bar{\ddot{x}}_{d_1} - \dot{J}_1\dot{q})$ modify the desired values of $\mathscr{T}_1$.

When $\beta = 0$, (11) can be represented as,

$$\begin{aligned} \min_{\ddot{q},\tau,w_2} \quad & \|w_2\|_2, \\ \text{s. t.} \quad & M\ddot{q}+C\dot{q}+g=\tau \\ & \underline{\ddot{x}}_{d_2} - \dot{J}_2\dot{q} \le J_2\ddot{q}+w_2 \le \bar{\ddot{x}}_{d_2} - \dot{J}_2\dot{q} \\ & J_1\ddot{q} - J_1\ddot{q}_2^* = 0 \end{aligned} \tag{12}$$

and the feasible solutions of both (9) and (12) are always the same as $\ddot{q}_2^*$, as shown in Fig. 2(a). Also, note that the solution with $\beta = 1$ in (11) is equal to that of (10). When β has a value between 0 and 1, the feasible solution can be derived by internal division between the feasible solution area of (9) and (10), as shown in Fig. 2(b). Consequently, by increasing the value of β from 0 to 1, the continuity of the task transition for inserting the higher priority task can be ensured. It is important to note that our strategy can treat two hierarchical inequality tasks strictly during the transition.

In case of removing task in the existing SoT, the formulation of (11) can also be used. The only difference between insertion and removal is that the activation parameter, β, should decrease from 1 to 0, during the removal of a task.

By using the proposed approach for inserting a task and removing another task, to replace an existing $\mathscr{T}_1$ with a new

$\mathscr{T}_2$, the proposed transition algorithm is composed as,

$$\begin{aligned}\min_{\ddot{q},\tau,w_2} \quad & \|w_2\|_2, \\ \text{s. t.} \quad & M\ddot{q}+C\dot{q}+g=\tau \\ & \beta_2(\underline{\ddot{x}}_{d_2}-\dot{J}_2\dot{q}) \le J_2\ddot{q}-(1-\beta_2)J_2\ddot{q}^*_{1i}+w_2 \le \beta_2(\bar{\ddot{x}}_{d_2}-\dot{J}_2\dot{q}) \\ & \beta_1(\underline{\ddot{x}}_{d_1}-\dot{J}_1\dot{q}) \le J_1\ddot{q}-(1-\beta_1)J_1\ddot{q}^*_{2i}+w_1^* \le \beta_1(\bar{\ddot{x}}_{d_1}-\dot{J}_1\dot{q})\end{aligned} \tag{13}$$

where β_1 and β_2 are the activation parameters for each task and these are composed using a monotone function (e.g. cubic spline and hyperbolic tangent sigmoid), as in Fig. 3. Also, $\ddot{q}^*_{1i}$ is the solution of the existing task, $\mathscr{T}_1$, with β_1 and $\ddot{q}^*_{2i}$ is that of the new task, $\mathscr{T}_2$, with β_2 and these are obtained by the following equations.

$$\begin{aligned}\min_{\ddot{q},\tau,w_1} \quad & \|w_1\|_2, \\ \text{s. t.} \quad & M\ddot{q}+C\dot{q}+g=\tau \\ & \beta_1(\underline{\ddot{x}}_{d_1}-\dot{J}_1\dot{q}) \le J_1\ddot{q}+w_1 \le \beta_1(\bar{\ddot{x}}_{d_1}-\dot{J}_1\dot{q})\end{aligned} \tag{14}$$

$$\begin{aligned}\min_{\ddot{q},\tau,w_2} \quad & \|w_2\|_2, \\ \text{s. t.} \quad & M\ddot{q}+C\dot{q}+g=\tau \\ & \beta_2(\underline{\ddot{x}}_{d_2}-\dot{J}_2\dot{q}) \le J_2\ddot{q}+w_2 \le \beta_2(\bar{\ddot{x}}_{d_2}-\dot{J}_2\dot{q})\end{aligned} \tag{15}$$

As shown in Fig. 4, the proposed algorithm with (13) can generate the continuous control input during the task transition due to the activation parameters. The number of solving QP for (13) is four.

More generally, if replacing the k-th task ($\mathscr{T}_k$) with a new task ($\mathscr{T}_{\hat{k}}$) in the SoT with n tasks, the continuous transition is represented as (16) and the number of QP operations is $2n-k+1$.

$$\begin{aligned}\min_{\ddot{q},\tau,w_n} \quad & \|w_n\|_2, \\ \text{s. t.} \quad & M\ddot{q}+C\dot{q}+g=\tau \\ & \underline{\ddot{x}}_{d_i}-\dot{J}_i\dot{q} \le J_i\ddot{q}+w_i \le \bar{\ddot{x}}_{d_i}-\dot{J}_i\dot{q}, \forall i \in k+1,\ldots,n \\ & \beta_2(\underline{\ddot{x}}_{d_{\hat{k}}}-\dot{J}_{\hat{k}}\dot{q}) \le J_{\hat{k}}\ddot{q}-(1-\beta_2)J_{\hat{k}}\ddot{q}^*_{\hat{k}i}+w^*_{\hat{k}} \le \beta_2(\bar{\ddot{x}}_{d_{\hat{k}}}-\dot{J}_{\hat{k}}\dot{q}) \\ & \beta_1(\underline{\ddot{x}}_{d_k}-\dot{J}_k\dot{q}) \le J_k\ddot{q}-(1-\beta_1)J_k\ddot{q}^*_{ki}+w^*_k \le \beta_1(\bar{\ddot{x}}_{d_k}-\dot{J}_k\dot{q}) \\ & \underline{\ddot{x}}_{d_j}-\dot{J}_j\dot{q} \le J_j\ddot{q}+w^*_j \le \bar{\ddot{x}}_{d_j}-\dot{J}_j\dot{q}, \forall j \in 1,\ldots,k-1\end{aligned} \tag{16}$$

where $\ddot{q}^*_{ki}$ and $\ddot{q}^*_{\hat{k}i}$ are the solutions for the existing SoT with β_1 and the solution for the new SoT with β_2.

B. Swapping Priorities

In this section, we explain the more general case for task transition - swapping tasks between the existing tasks ($\mathscr{T}_1 \prec \mathscr{T}_2$). Let's define the two tasks as (10) and then the continuous task transition for swapping is expressed similarly to (13), as follows:

$$\begin{aligned}\min_{\ddot{q},\tau,w_2} \quad & \|w_2\|_2, \\ \text{s. t.} \quad & M\ddot{q}+C\dot{q}+g=\tau \\ & \beta_2(\underline{\ddot{x}}_{d_2}-\dot{J}_2\dot{q}) \le J_2\ddot{q}-(1-\beta_2)J_2\ddot{q}^*_{12i}+w_2 \le \beta_2(\bar{\ddot{x}}_{d_2}-\dot{J}_2\dot{q}) \\ & \beta_1(\underline{\ddot{x}}_{d_1}-\dot{J}_1\dot{q}) \le J_1\ddot{q}-(1-\beta_1)J_1\ddot{q}^*_{21i}+w_1^* \le \beta_1(\bar{\ddot{x}}_{d_1}-\dot{J}_1\dot{q})\end{aligned} \tag{17}$$

where β_1 and β_2 are the same activation parameters as those in the case of (13). Also, $\ddot{q}^*_{12i}$ and $\ddot{q}^*_{21i}$ are the solutions of $\mathscr{T}_1 \prec \mathscr{T}_2$ with β_1 and $\mathscr{T}_2 \prec \mathscr{T}_1$ with β_2, respectively.

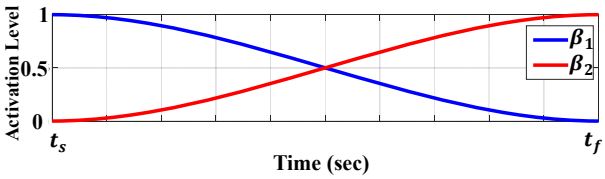


Fig. 3. Activation level w.r.t. replace tasks. A cubic spline is applied to each profile. t_s and t_f mean the start time and end time of transition period.

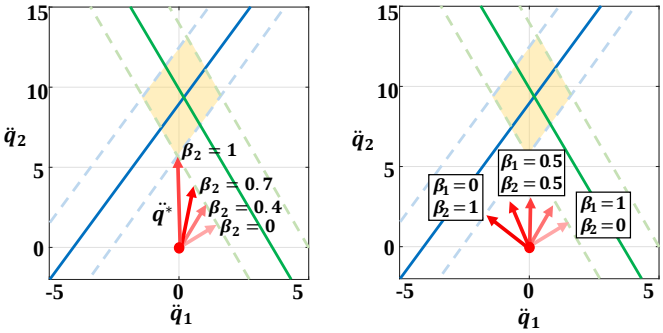


Fig. 4. 2D illustration of solution of HQP with β_1 and β_2: (a) illustration of (13) with $\beta_1 = 1$ and $0 \le \beta_2 \le 1$, (b) illustration of (13) with the activation parameters in Fig. 3.

Note that the solution of (17) with $\beta_1 = 0$ and $\beta_2 = 1$ is equal to the solution of the following equation which can solve the SoT with $\mathscr{T}_2 \prec \mathscr{T}_1$.

$$\begin{aligned}\min_{\ddot{q},\tau,w_1} \quad & \|w_1\|_2. \\ \text{s. t.} \quad & M\ddot{q}+C\dot{q}+g=\tau \\ & \underline{\ddot{x}}_{d_1}-\dot{J}_1\dot{q} \le J_1\ddot{q}+w_1 \le \bar{\ddot{x}}_{d_1}-\dot{J}_1\dot{q} \\ & \underline{\ddot{x}}_{d_2}-\dot{J}_2\dot{q} \le J_2\ddot{q}+w_2^* \le \bar{\ddot{x}}_{d_2}-\dot{J}_2\dot{q}\end{aligned} \tag{18}$$

For solving the HQP in (17), the number of QP operations is six. Likewise, if swapping the k-th and $k+1$-th in the SoT with n tasks, the number of operations is $3n-2k+2$.

IV. APPLICATIONS

This section describes various examples of our task transition method including joint limit, singularity, and obstacle avoidance.

A. Joint Limit Avoidance

In this section, the joint limit avoidance algorithm is proposed with the continuous inserting and removing tasks, as mentioned in Sec. III-A. Expanding the concept of the joint limit avoidance algorithm in the operational space controller [6], the proposed joint limit avoidance algorithm consists of a bilateral inequality constraint instead of a equality constraint.

Consider that n-DoF robot is controlled by using the HQP. When the i-th joint value, q_i, comes close to its joint limits while operating the predefined tasks, a joint limit avoidance task, $\mathscr{T}_{jl,i} \in \mathbb{R}^1$, with $J_{jl,i} \in \mathbb{R}^{1\times n}$, $\beta_{jl,i}$, and $[\underline{\dot{x}}_{jl,i}, \bar{\dot{x}}_{jl,i}] \in \mathbb{R}^1$ is inserted as the highest priority task in the HQP controller using (11). $J_{jl,i} \in \mathbb{R}^{1\times n}$ is a matrix with all 0 except only the i-th element having 1 and $\beta_{jl,i}$ is an activation parameter with respect to the range of the i-th joint. As shown in Fig. 5(a), $\beta_{jl,i}$ will increase when the corresponding joint value approaches the joint limit.

To avoid a joint limit, $[\underline{\ddot{x}}_{jl,i}, \bar{\ddot{x}}_{jl,i}]$ is expressed as,

$$\underline{\ddot{x}}_{jl,i} = \begin{cases} k_p((\underline{q}_i + \alpha_{jl,i}) - q_i) - k_v \dot{q}_i, & \text{if } \ q_i < \underline{q}_i + \alpha_{jl,i} \\ \underline{\ddot{x}}_{dl,i}, & \text{otherwise,} \end{cases}$$
$$\bar{\ddot{x}}_{jl,i} = \begin{cases} k_p((\bar{q}_i - \alpha_{jl,i}) - q_i) - k_v \dot{q}_i, & \text{if } \ q_i > \bar{q}_i - \alpha_{jl,i} \\ \bar{\ddot{x}}_{dl,i}, & \text{otherwise,} \end{cases} \tag{19}$$

where $\dot{q}_i$, k_p, k_v, and $\alpha_{jl,i}$ denote the joint velocity of i-th joint, proportional and derivative gains, and activation buffer, respectively. The terms $[\underline{q}_i, \bar{q}_i]$ and $[\underline{\ddot{x}}_{dl,i}, \bar{\ddot{x}}_{dl,i}]$ are the lower and upper joint limit, and the lower and upper default joint acceleration limit of the i-th joint. Therefore, this task can control the joint into the feasible range of the joint when it comes close to joint limits.

B. Singularity Avoidance

To avoid high joint acceleration by kinematic and algorithmic singularities, the damping method in the HQP based controller has been used in [9]. However, this approach can adversely affect a solution for original tasks [8]. To overcome this drawback, Han and Park proposed the singularity avoidance algorithm using Singular Value Decomposition (SVD) [6].

In this paper, we propose a new singularity avoidance algorithm using QR factorization instead of SVD because the complexity of QR factorization ($O(mn^2 - n^3/3)$) is smaller than that of SVD ($O(mn^2 + n^3)$) for a thin matrix $A \in \mathbb{R}^{m \times n}$ [19].

Consider a task, $\mathscr{T}_t \in \mathbb{R}^m$, with $J_t \in \mathbb{R}^{m \times n}$ and $[\underline{\ddot{x}}_t, \bar{\ddot{x}}_t] \in \mathbb{R}^m$. By using Householder QR factorization, when the task Jacobian J_t is a rank deficient matrix with rank r, it is decomposed as,

$$J_t = \underbrace{\begin{bmatrix} Q_{t,ns} & Q_{t,s} \end{bmatrix}}_{\mathbf{Q}} \underbrace{\begin{bmatrix} R_{t,ns} & 0 \\ \hline 0 & 0 \end{bmatrix}}_{\mathbf{R}} Z^T, \tag{20}$$

where $Q \in \mathbb{R}^{m \times m}$ is an orthonormal matrix and $Z \in \mathbb{R}^{n \times n}$ is a unimodular matrix (i.e. $det(Z) = \pm 1$). Also, $R \in \mathbb{R}^{m \times n}$ is a rank deficient matrix which has an upper triangular matrix $R_{t,ns} \in \mathbb{R}^{r \times r}$. Thus, the orthonormal bases of $Q_{t,ns} \in \mathbb{R}^{m \times r}$ with respect to $R_{t,ns}$ indicate non-singular directions of $\mathscr{T}_t$. Also, $Q_{t,s} \in \mathbb{R}^{m \times (m-r)}$ is the vector space of bases for singular directions.

By using this concept, the task $\mathscr{T}_t$ can be decomposed into two sub-tasks ($\mathscr{T}_{t,ns}$, $\mathscr{T}_{t,s}$), as below:

$$J_{t,ns} = Q_{t,ns}^T J_t \quad \text{and} \quad J_{t,s} = Q_{t,s}^T J_t, \tag{21}$$

$$[\underline{\ddot{x}}_{t,ns}, \bar{\ddot{x}}_{t,ns}] = Q_{t,ns}^T [\underline{\ddot{x}}_t, \bar{\ddot{x}}_t] \quad \text{and} \quad [\underline{\ddot{x}}_{t,s}, \bar{\ddot{x}}_{t,s}] = Q_{t,s}^T [\underline{\ddot{x}}_t, \bar{\ddot{x}}_t]. \tag{22}$$

$\mathscr{T}_t$ can be strictly decomposed into $\mathscr{T}_{t,ns}$ for non-singular direction's movement and $\mathscr{T}_{t,s}$ for singular direction's movement.

Therefore, singularities of $\mathscr{T}_t$ can be avoided by removing $\mathscr{T}_{t,s}$ when a robot comes close to a singularity region. In this paper, a manipulability index, $\sqrt{det(J_t J_t^T)}$, is used as a criterion for the singularity. As the value of the manipulability index decreases, $\mathscr{T}_{t,s}$ is gradually removed by decreasing the activation parameter, as shown in Fig. 5(b). Hence, even within the singularity area, the robot can perform given tasks because of the deactivated $\mathscr{T}_{t,s}$.

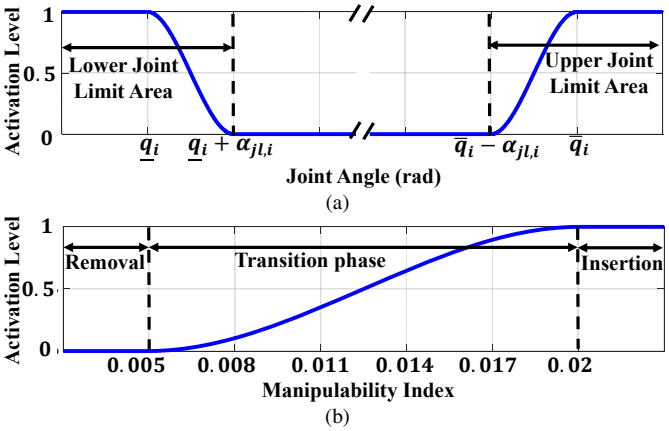


Fig. 5. Activation parameters: (a) joint limit, (b) singularity.

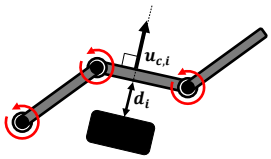


Fig. 6. Illustration of obstacle avoidance.

C. Obstacle Avoidance

The task for obstacle avoidance is inserted when the shortest distance between each link and each obstacle becomes smaller than a certain threshold[1].

Let's consider that the shortest distance between the i-th link and each obstacle is within a threshold, as shown in Fig. 6. Then, the task for avoiding an obstacle with Jacobian, $J_{c,i} \in \mathbb{R}^{1 \times n}$ and $\ddot{x}_{c,i} \in \mathbb{R}^1$ are represented as follows.

$$J_{c,i} = u_{c,i}^T J_i, \tag{23}$$

$$\ddot{x}_{c,i} = k_p((d_{ref} + \alpha_c) - d_i) - k_v \dot{d}_i \tag{24}$$

where $u_{c,i} \in \mathbb{R}^3$ and $J_i \in \mathbb{R}^{3 \times n}$ are the direction vector between the i-th link and the obstacle and Jacobian for translation on the i-th link, respectively. Also, d_i, d_{ref}, and α_c are the shortest distance between the i-th link and the obstacle, the threshold, and the buffer length, respectively.

We design the activation parameter for this task similar to that for joint limit avoidance. When the distance between each link and each obstacle becomes smaller than $d_{ref} + \alpha_c$, the activation value gradually increases to 1.

V. EXPERIMENTS

The proposed control framework was verified through experiments with a 7-DoF robotic manipulator. The subsections below describe the details of our system configuration and experimental results with the robot.

A. System Overview

The kinematic structure of our 7-DoF manipulator is shown in [20]. The actuators of the robot are torque controlled electric motors and these motors are controlled using *EtherCAT* in the

[1]To decrease computational cost, we used the collision model based on hyper-ellipsoids.

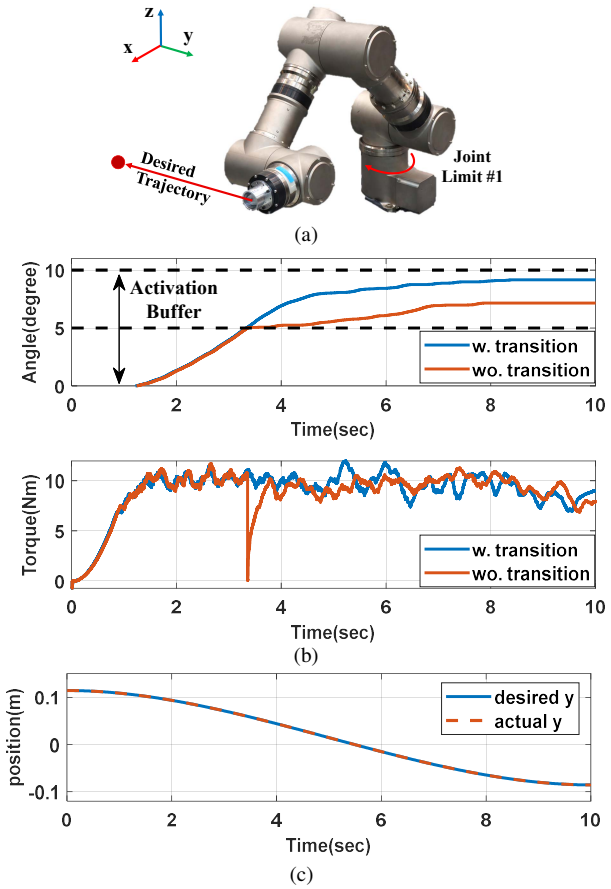


Fig. 7. Experiment for joint limit avoidance: (a) initial state and desired position, (b) joint angle and command torque for the first joint, (c) desired and actual position in the y-direction of the proposed algorithm.

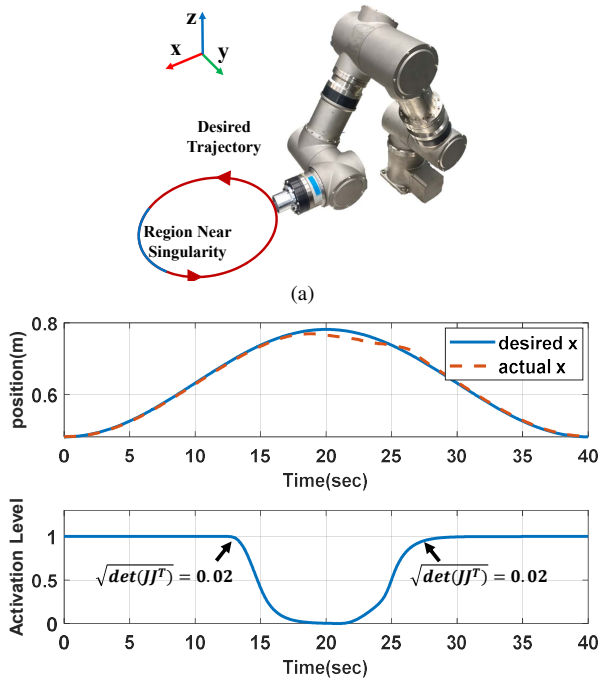


Fig. 8. Experiment for singularity avoidance: (a) initial state and desired position, (b) desired and actual position in the x-direction and activation parameter.

Xenomai real-time Linux kernel. The control frequency of the manipulator is 2kHz. The specification of the computer for the controller is *i*7 4.2GHz with 16GB RAM.

B. Experimental Results

Several experiments with the robotic manipulator were conducted to verify each performance of the joint limit, singularity, and obstacle avoidance algorithms in Sec. IV. First, the experiment for validating the joint limit avoidance was conducted with a high priority task, $\mathscr{T}_1 \in \mathbb{R}^6$, to move the end-effector -20cm in the y-direction and a low priority task, $\mathscr{T}_2 \in \mathbb{R}^7$, for maintaining the initial joint posture, as shown in Fig. 7(a). In this experiment, we set the joint range of the first joint as $[-10°, 10°]$ with $\alpha_{jl} = 5°$. Thus, when the first joint is out of the joint range, the SoT changes from $\mathscr{T}_1 \prec \mathscr{T}_2$ to $\mathscr{T}_{jl,1} \prec \mathscr{T}_1 \prec \mathscr{T}_2$ by using the activation parameter, as mentioned in Sec. IV-A. Fig. 7(b) shows the joint angle and command torque of the first joint when inserting the joint limit avoidance task with and without the proposed transition algorithm. The joint limit could be avoided by the joint limit avoidance task in both cases. The maximum value of the joint with the transition algorithm (9.1°) is slightly higher than the value without the transition (7.2°), because of the continuous transition by the activation parameter. On the other hand, when the task for avoiding the joint limit was inserted without the transition, the vibration sound was heard because of the sudden discontinuity of the torque. In contrast, by inserting the joint limit avoidance task with the proposed task transition, the continuous torque trajectory could be generated. Fig. 7(c) shows the tracking result of $\mathscr{T}_1$ during the proposed task transition. Although the SoT changed from $\mathscr{T}_1 \prec \mathscr{T}_2$ to $\mathscr{T}_{jl,1} \prec \mathscr{T}_1 \prec \mathscr{T}_2$, the end-effector could track the desired trajectory due to the redundancy of the robot.

Second, the performance of the singularity avoidance was validated with a high priority task, $\mathscr{T}_1 \in \mathbb{R}^6$, for drawing an ellipsoid in the xy-plane and a low priority task, $\mathscr{T}_2 \in \mathbb{R}^7$, for maintaining the initial joint posture, as shown in Fig. 8(a). Following the ellipsoid trajectory forces the robot to move into the region near the singularity, and the robot may become unstable in the region. Thus, when the robot approached the singularity region, the proposed algorithm reconstructed this SoT as $(\mathscr{T}_{1,ns}, \mathscr{T}_{1,s}) \prec \mathscr{T}_2$, as mentioned in Sec. IV-B. Because the proposed algorithm can deactivate the task in the singular direction, $\mathscr{T}_{1,s}$, by using QR decomposition, the robot could deal with singularity, as shown in Fig. 8(b). By decreasing the activation parameter with respect to the manipulability index of the robot, the position of the end-effector does not track the desired trajectory for $\mathscr{T}_1$ in the singular direction.

Finally, the performance for obstacle avoidance was demonstrated through the experiment with a high priority task, $\mathscr{T}_1 \in \mathbb{R}^6$, for moving the end-effector 20cm in the y-direction and a low priority task, $\mathscr{T}_2 \in \mathbb{R}^7$, for maintaining the initial posture of the robot. Because there are obstacles including a ball and a wooden rod near the robot, the task for obstacle avoidance is needed. Thus, the proposed framework added a task for avoiding the obstacles when the distance between a certain link of the robot and the obstacle is less than the threshold, as mentioned in Sec. IV-C. In this experiment, d_{ref}

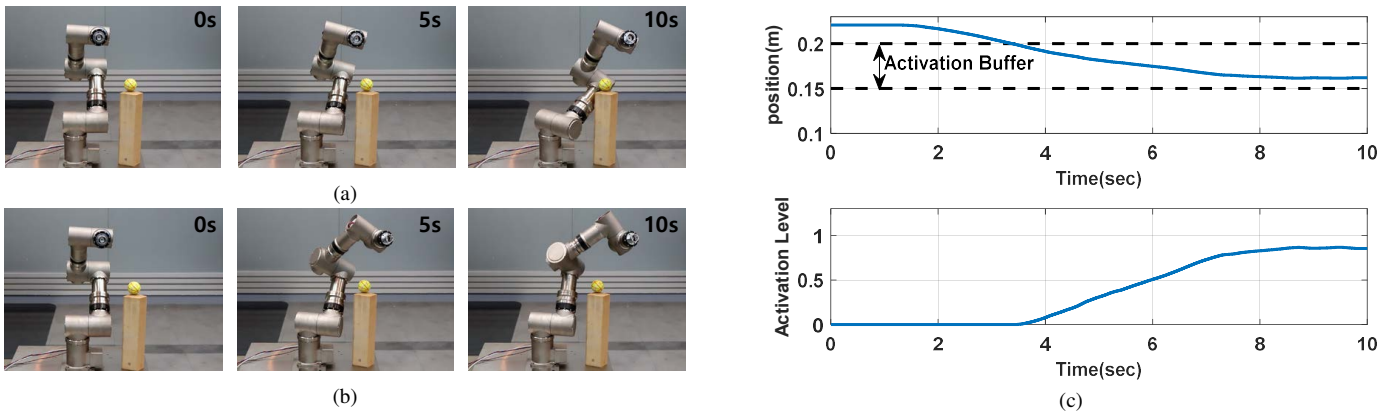


Fig. 9. Experiment for obstacle avoidance: (a) snapshots without the obstacle avoidance task, (b) snapshots with the obstacle avoidance task, (c) minimum distance between the obstacle and third link and activation parameter.

and α_c were set to 15cm and 5cm, respectively. Fig. 9(a) and 9(b) show the snapshots of the experiments with and without the obstacle avoidance task. When the robot executed the existing SoT ($\mathscr{T}_1 \prec \mathscr{T}_2$) only, a collision between the obstacle and the third link occurred, as shown in Fig. 9(a). In contrast, when the robot executed this SoT with the proposed strategy ($[\mathscr{T}_1 \prec \mathscr{T}_2 \rightarrow \mathscr{T}_c \prec \mathscr{T}_1 \prec \mathscr{T}_2]$), the distance between the obstacle and third link was always more than 16cm, as shown in Fig. 9(b) and Fig. 9(c). Therefore, the collision between the robot and the obstacle could be avoided.

We also designed a complex scenario in order to validate the performance of the swapping the prioritized multi-tasks with the continuous task transition strategy. In this scenario, the controller has three tasks: $\mathscr{T}_1 \in \mathbb{R}^7$ for maintaining the initial joint posture, $\mathscr{T}_2 \in \mathbb{R}^6$ to move the end-effector 10cm in the x-direction from the initial posture, and $\mathscr{T}_3 \in \mathbb{R}^2$ to fold the elbow of the robot by controlling the first and fourth joints. With these tasks, the order of task priorities was swapped in real-time during the operation of the robot, as shown in Fig. 10(a).

The results with this scenario are shown in Fig. 10(b) and 10(c). The sequence of rearranging tasks was $[\mathscr{T}_1 \prec \mathscr{T}_3 \prec \mathscr{T}_2 \rightarrow \mathscr{T}_2 \prec \mathscr{T}_3 \prec \mathscr{T}_1]$ for 0 to 5sec, $[\mathscr{T}_2 \prec \mathscr{T}_3 \prec \mathscr{T}_1 \rightarrow \mathscr{T}_3 \prec \mathscr{T}_2 \prec \mathscr{T}_1]$ for 6 to 16sec, and $[\mathscr{T}_3 \prec \mathscr{T}_2 \prec \mathscr{T}_1 \rightarrow \mathscr{T}_1 \prec \mathscr{T}_3 \prec \mathscr{T}_2]$ for 18 to 28sec. By using the proposed swapping strategy in Sec. III-B, the smooth joint movements are guaranteed without jerking. Especially, Fig. 10(b) shows the continuous command torque from the HQP controller with our transition method. By using the activation parameter for swapping, the discontinuity caused by changing the SoT disappeared. Snapshots of the experiments are shown in Fig. 10(c). In contrast, in the experiment for swapping these tasks without the task transition, the robot automatically powered off due to large command torques during the swapping tasks.

The mean computation time during this scenario was 0.00037 ± 0.00013sec whereas that of the original HQP with 3 tasks was 0.00023 ± 0.00014sec. Our algorithm increases the computation since the number of QP operations is greater than that of original formulation to handle priorities strictly during the transition phase.

The C++ source code with QP solver, *qpOASES* [21], is available at [22] for Windows and Ubuntu 16.04 LTS. The video clips of not only the experiments described in this paper but also the experiments with other robots including a nonholonomic mobile manipulator are available in [20].

VI. CONCLUSIONS

The hierarchical controllers with SoTs have great advantages in executing multiple prioritized tasks simultaneously. However, the change of the existing SoT in the controller causes the discontinuity of the control input. In this paper, a novel task transition strategy for the HQP based controller is proposed. The continuous task transition strategy is discussed to avoid the discontinuity of the control input variables when certain tasks are inserted, removed, and rearranged. Our approach can handle both equality and inequality tasks by modifying the offset value of the existing tasks and the bound set of the new task with the activation parameter. Thus, without modifying control structure, our method can deal with continuous task transition between not only consecutive tasks but also non-consecutive tasks. Based on the proposed framework, various applications including the joint limit, singularity and obstacle avoidance are proposed in this paper. We demonstrated the effectiveness of the proposed method by several experiments with a real robot. Our future work will involve the extension of the proposed framework for contact force transition and apply it for mobile manipulators and humanoid robots in order to enhance stability during whole-body contact situation.

REFERENCES

[1] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *proceeding of 5th International Conference on Advanced Robotics*, vol. 2, 1991, pp. 1211–1216.

[2] 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, no. 3, pp. 670–685, 2009.

[3] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Operational space control: A theoretical and empirical comparison," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 737–757, 2008.

[4] A. Dietrich, C. Ott, and J. Park, "The hierarchical operational space formulation: Stability analysis for the regulation case," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1120–1127, 2018.

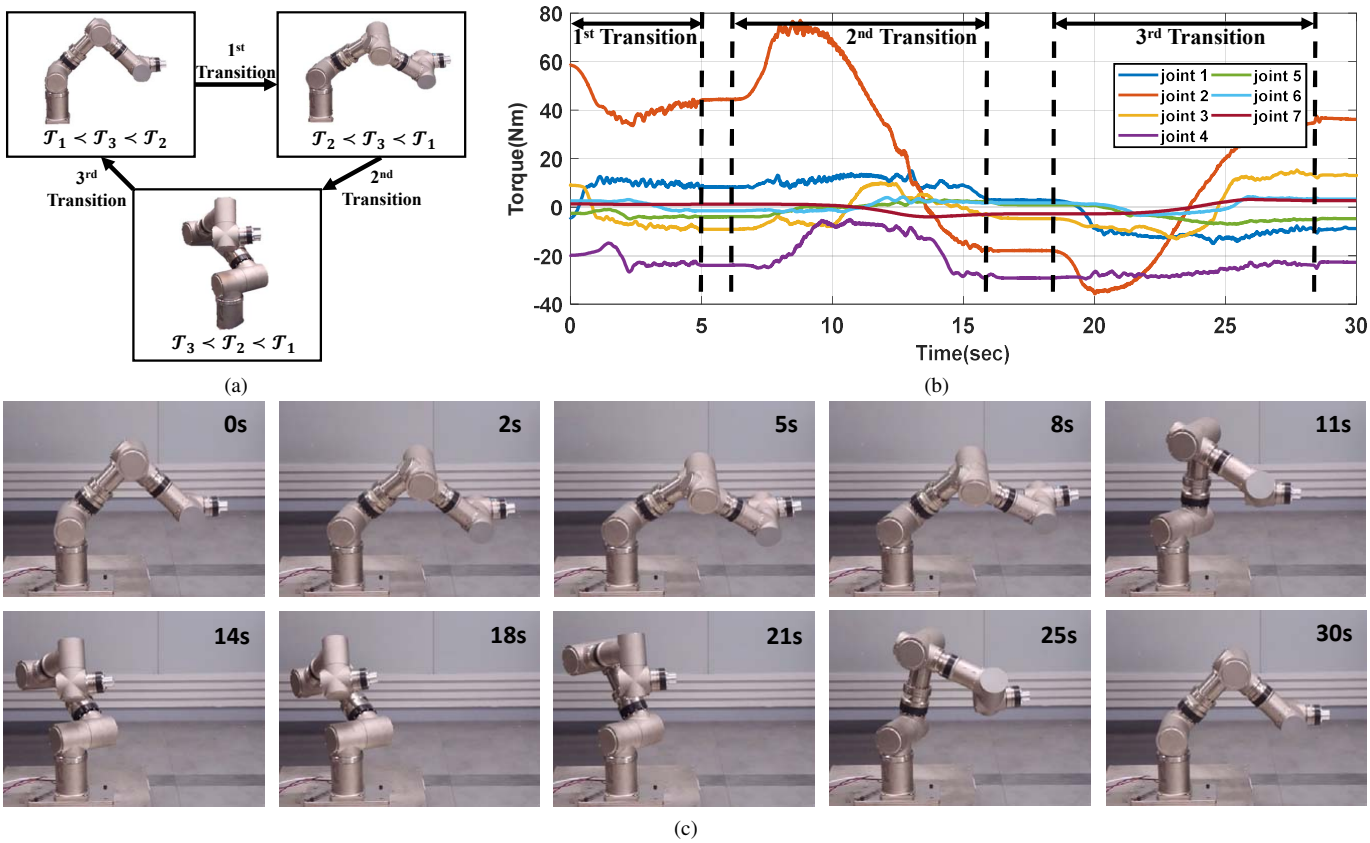


Fig. 10. Experiment for swapping multi-prioritized tasks: (a) sequence of swapping, (b) command torque profiles by proposed algorithm, (c) snapshots during experiment.

[5] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: generalizing the task priority framework to inequality tasks," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.

[6] H. Han and J. Park, "Robot control near singularity and joint limit using a continuous task transition algorithm," *International Journal of Advanced Robotic Systems*, vol. 10, no. 10, pp. 346–356, 2013.

[7] F. Keith, N. Mansard, S. Miossec, and A. Kheddar, "Optimization of tasks warping and scheduling for smooth sequencing of robotic actions," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 1609–1614.

[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, no. 6, pp. 1260–1277, 2012.

[9] G. Jarquín, A. Escande, G. Arechavaleta, T. Moulard, E. Yoshida, and V. Parra-Vega, "Real-time smooth task transitions for hierarchical inverse kinematics," in *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*. IEEE, 2013, pp. 528–533.

[10] M. Liu, Y. Tan, and V. Padois, "Generalized hierarchical control," *Autonomous Robots*, vol. 40, no. 1, pp. 17–31, 2016.

[11] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[12] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf, "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results," *Frontiers in Robotics and AI*, vol. 3, pp. 1–18, 2016.

[13] P. Di Lillo, F. Arrichiello, G. Antonelli, and S. Chiaverini, "Safety-related tasks within the set-based task-priority inverse kinematics framework," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6130–6135.

[14] F. Flacco, A. De Luca, and O. Khatib, "Control of redundant robots under hard joint constraints: Saturation in the null space," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 637–654, 2015.

[15] J. Wan, H. Wu, R. Ma, and L. Zhang, "A study on avoiding joint limits for inverse kinematics of redundant manipulators using improved clamping weighted least-norm method," *Journal of Mechanical Science and Technology*, vol. 32, no. 3, pp. 1367–1378, 2018.

[16] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soueres, and J.-Y. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 346–362, 2013.

[17] O. Kermorgant, Y. Pétillot, and M. Dunnigan, "A global control scheme for free-floating vehicle-manipulators," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 5015–5020.

[18] E. M. Hoffman, A. Laurenzi, L. Muratore, N. G. Tsagarakis, and D. G. Caldwell, "Multi-priority cartesian impedance control based on quadratic programming optimization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 309–315.

[19] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.

[20] S. Kim, "Task transition algorithms," https://ggory15.github.io/tasktransition-project, 2018, [Online; accessed 1-December-2018].

[21] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpoases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[22] S. Kim, "Continuous task transition approach for robot controller based on hierarchical quadratic programming," https://ggory15.github.io/HQP-transition, 2018, [Online; accessed 1-December-2018].