

다관절 로봇의 계층적 제어를 위한 HQP의 연산 비용 감소 방법

Computational Cost Reduction Method for HQP-based Hierarchical Controller for Articulated Robot

박민규¹·김동환¹·오용환²·이이수[†]

Mingyu Park¹, Dongwhan Kim¹, Yonghwan Oh², Yisoo Lee[†]

Abstract: This paper presents a method that can reduce the computational cost of the hierarchical quadratic programming (HQP)-based robot controller. Hierarchical controllers can effectively manage articulated robots with many degrees of freedom (DoFs) to perform multiple tasks. The HQP-based controller is one of the generic hierarchical controllers that can provide a control solution guaranteeing strict task priority while handling numerous equality and inequality constraints. However, according to a large amount of computation, it can be a burden to use it for real-time control. Therefore, for practical use of the HQP, we propose a method to reduce the computational cost by decreasing the size of the decision variable. The computation time and control performance of the proposed method are evaluated by real robot experiments with a 15 DoFs dual-arm manipulator.

Keywords: Hierarchical Control, Inverse Dynamics, Manipulation, Optimization

1. Introduction

To perform complex and informal tasks, modern robots such as dual-arm manipulators, humanoid robots, mobile manipulators have many joints and most of them are kinematically redundant. By exploiting a redundant robotic system, multiple tasks can be executed satisfying various constraints among the infinite number of control solutions.

For practical use, the control solution has to satisfy inequality constraints for joint limits and workspace limit avoidance. In addition, the capability of hierarchical task control is important to manage multiple tasks systematically. Moreover, the afore-

mentioned has to be considered in a fast computation to keep a strict real-time system. For the control with high frequency such as 1 kHz, it has to solve the control problem within 1 ms.

Kinematically redundant manipulators generally solve inverse dynamics or inverse kinematics problems to calculate control solutions. In this paper, we focus on the inverse dynamics problem which requires more computation time compared to the inverse kinematics. In the inverse dynamics, joint space torque that can create desired end-effector motion or force at the Cartesian coordinate can be calculated based on the Jacobian matrix and robot dynamics. Nowadays, inverse dynamics solvers can be classified into three approaches: a method based on pseudo inverse, a method based on quadratic programming (QP), and a method based on hierarchical quadratic programming (HQP).

The operational space formulation (OSF) is a typical and classic method for inverse dynamics control [1]. In the OSF, joint acceleration energy minimized torque solution is calculated through the dynamically consistent inverse, i.e., inertia weighted pseudo inverse. In addition, the OSF can provide strict task

Received : Oct. 18. 2021; Revised : Dec. 16. 2021; Accepted : Dec. 28. 2021

※ This project was supported by the Korea Institute of Science, and Technology Institutional Programs under Grant 2E31593

1. BS Student, Kwangwoon University, Seoul, Korea (pmg6484, akdntm142@kw.ac.kr)

2. Principal Researcher, Center for Intelligent and Interactive Robotics, Korea Institute of Science and Technology (KIST), Seoul, Korea (oyh@kist.re.kr)

† Senior Researcher, Corresponding author: Center for Intelligent and Interactive Robotics, Korea Institute of Science and Technology (KIST), Seoul, Korea (yisoo.lee@kist.re.kr)

priority by removing the effect from lower priority tasks to the higher priority tasks based on the null-space projection approach. However, since the OSF cannot apply inequality constraints explicitly, it requires additional algorithms [2,3] for considering the inequality constraints such as joint torque limits, singularity avoidance, and obstacle avoidance.

Consideration of the inequality constraints are becoming more important for practical use of robots. Thus, the QP-based controller is proposed to consider inequalities when solving inverse dynamics for floating base robots [4]. By solving a numerical quadratic program in the QP-based methods, both equality and inequality constraints can be considered. However, it cannot guarantee a strict task hierarchy although it can set soft task priority. Thus, the lower priority task can affect higher priority tasks.

As an extension of the QP-based method, the HQP-based method is proposed for floating base robots [5]. By solving multiple QPs, the HQP can satisfy equality and inequality constraints and can satisfy a strict hierarchy for task control. However, since it has to solve QPs more than the number of task hierarchies, it requires a large computational cost for multiple task execution.

The HQP-based methods regarding the floating base robots such as humanoids are generally focusing on the computational cost since those robots have high degrees of freedom (DoFs) and many constraints. De Lasa et al. [6] improve the computational time by reducing equality and inequality constraints of lower priority QPs. Herzog et al., Mansard, Lee et al. [7-9] decrease the number of decision variables. Escande et al. [10] suggest a method to reduce the number of QP solving.

Recently, the HQP-based approach is exploited for fixed base robots [11]. Since the number of joints of the fixed base robot is increasing, e.g., dual-arm manipulators [12], a large computational cost is also required. However, methods for improving the computational cost of the HQP for fixed base robots are rare. To the best of our knowledge, a study in [13] is the only method regarding the computational cost issue. Since the dynamics of the floating base robot and the fixed base robot are different, so the aforementioned computational cost reduction approaches for floating base robots can't be applied directly to the fixed base robot.

Therefore, the main contribution of this study comes from the proposal of the computationally improved HQP formulation for the fixed base robot. Motivated by the approach in the methods

for the floating base robots, we reduce the size of decision variables and reformulate the HQP problem. The proposed method has an advantage since it can minimize both the joint space and the task space variables and can consider two or more hierarchical tasks. Compared to our study, the most recent study [13] can consider only two-level tasks and cannot optimize task space variables. Thus, our method can be considered as a more general method.

Real robot experiments are executed on the dual-arm manipulator to evaluate the reduction of computational time compared to the OSF and conventional HQP-based method. In addition, the performance of the proposed method is also evaluated by experiments.

2. Background Theory

In this section, we introduce the background theory about HQP-based inverse dynamics controller. Rigid body dynamics of the robot with n joints can be expressed as follows:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\dot{\mathbf{q}}, \mathbf{q}) = \boldsymbol{\tau}, \quad (1)$$

where $\mathbf{M}(\mathbf{q})$ is the inertia matrix, $\mathbf{h}(\dot{\mathbf{q}}, \mathbf{q})$ is the sum of the Coriolis/centrifugal force vector, and gravity force vector. And $\mathbf{q} \in \mathbb{R}^n$ is the joint position vector and $\boldsymbol{\tau}$ is the joint torque vector.

The forward kinematics which describes the relationship between task space and joint space, and its differential relationship are shown as

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad \ddot{\mathbf{x}} = \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}}, \quad (2)$$

where $\mathbf{J}(\mathbf{q})$ is a Jacobian matrix, $\mathbf{x} \in \mathbb{R}^m$ is the operational space coordinate vector, and m is the DoFs of task [14]. After this paragraph, $\mathbf{M}(\mathbf{q})$, $\mathbf{h}(\dot{\mathbf{q}}, \mathbf{q})$ and $\mathbf{J}(\mathbf{q})$ are indicated as \mathbf{M} , \mathbf{h} , and \mathbf{J} , respectively for the sake of brevity. Based on equations (1) and (2), the HQP-based controller solves the inverse dynamics solution.

HQP is the process of solving cascade numerical optimization problems using quadratic programming (QP). In the QP, we can minimize quadratic function while satisfying equality and inequality constraints. From the HQP method in [15], the QP formulation for the first priority task can be derived as:

$$\min_{\ddot{\mathbf{q}}, \boldsymbol{\tau}, \mathbf{w}_1} \|\mathbf{w}_1\|^2, \quad (3)$$

subject to

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{h} = \boldsymbol{\tau}, \\ \ddot{\mathbf{x}}_{d_1} + \mathbf{w}_1 = \mathbf{J}_1\dot{\mathbf{q}} + \mathbf{J}_1\ddot{\mathbf{q}}, \\ \underline{\ddot{\mathbf{q}}} \leq \ddot{\mathbf{q}} \leq \overline{\ddot{\mathbf{q}}}, \\ \underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} \leq \overline{\boldsymbol{\tau}}, \\ \underline{\mathbf{w}_1} \leq \mathbf{w}_1 \leq \overline{\mathbf{w}_1}, \end{cases} \quad (4)$$

where \mathbf{w}_1 is a slack variable vector for relaxing control reference for a feasible solution. Top lines and underlines indicate upper and lower constraints' bound vectors. $\ddot{\mathbf{x}}_{d_1}$ is a desired operational space acceleration vector and the number of subscripts at $\ddot{\mathbf{x}}_{d_1}$, \mathbf{w}_1 , \mathbf{J} denotes the priority of the tasks. After solving the QP optimization, by applying the optimal torque $\boldsymbol{\tau}$ to the robot, the end-effector acceleration can be generated as $\ddot{\mathbf{x}}_{d_1} + \mathbf{w}_1$ while satisfying all the constraints. When there is the second priority task, instead of applying $\boldsymbol{\tau}$ from above, the second QP is solved with obtained optimal slack variable \mathbf{w}_1 .

An optimal slack variable \mathbf{w}_1 is now expressed as \mathbf{w}_1^* in the second QP and it is set as an additional equality constraint. The second QP can be formulated as follows:

$$\min_{\ddot{\mathbf{q}}, \boldsymbol{\tau}, \mathbf{w}_2} \|\mathbf{w}_2\|^2, \quad (5)$$

subject to

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{h} = \boldsymbol{\tau}, \\ \ddot{\mathbf{x}}_{d_1} + \mathbf{w}_1^* = \mathbf{J}_1\dot{\mathbf{q}} + \mathbf{J}_1\ddot{\mathbf{q}}, \\ \ddot{\mathbf{x}}_{d_2} + \mathbf{w}_2 = \mathbf{J}_2\dot{\mathbf{q}} + \mathbf{J}_2\ddot{\mathbf{q}}, \\ \underline{\ddot{\mathbf{q}}} \leq \ddot{\mathbf{q}} \leq \overline{\ddot{\mathbf{q}}}, \\ \underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} \leq \overline{\boldsymbol{\tau}}, \\ \underline{\mathbf{w}_2} \leq \mathbf{w}_2 \leq \overline{\mathbf{w}_2}, \end{cases} \quad (6)$$

where \mathbf{w}_2 is a slack variable for the second priority task. According to the equality constraint regarding \mathbf{w}_1^* , i.e., $\ddot{\mathbf{x}}_{d_1} + \mathbf{w}_1^* = \mathbf{J}_1\dot{\mathbf{q}} + \mathbf{J}_1\ddot{\mathbf{q}}$, a strict task hierarchy can be obtained since the second QP will calculate a control solution that does not affect the first priority task. Note that it is proved in [5] that this approach can guarantee strict task hierarchy as the same as the well-known null-space projection-based method [1]. As the result of the second QP, $\boldsymbol{\tau}$ generates the end-effector acceleration as $\ddot{\mathbf{x}}_{d_1} + \mathbf{w}_1^*$, and $\ddot{\mathbf{x}}_{d_2} + \mathbf{w}_2$ for the hierarchical tasks.

Similarly, we can consider more hierarchical tasks by solving QPs in cascade manner as introduced above. The generalized equation for a k -th prioritized task can be described as follows:

$$\min_{\ddot{\mathbf{q}}, \boldsymbol{\tau}, \mathbf{w}_k} \|\mathbf{w}_k\|^2, \quad (7)$$

subject to

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{h} = \boldsymbol{\tau}, \\ \ddot{\mathbf{x}}_{d_l} + \mathbf{w}_l^* = \mathbf{J}_l\dot{\mathbf{q}} + \mathbf{J}_l\ddot{\mathbf{q}}, \quad \forall l \in 1, \dots, k-1, \\ \ddot{\mathbf{x}}_{d_k} + \mathbf{w}_k = \mathbf{J}_k\dot{\mathbf{q}} + \mathbf{J}_k\ddot{\mathbf{q}}, \\ \underline{\ddot{\mathbf{q}}} \leq \ddot{\mathbf{q}} \leq \overline{\ddot{\mathbf{q}}}, \\ \underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} \leq \overline{\boldsymbol{\tau}}, \\ \underline{\mathbf{w}_k} \leq \mathbf{w}_k \leq \overline{\mathbf{w}_k}, \end{cases} \quad (8)$$

where the notation of subscript of l indicates a higher priority of tasks than k .

Finally, control torque can be determined as the torque $\boldsymbol{\tau}$ from the QP for the last priority task. It is notable that there are still infinite solutions when the hierarchical tasks do not use the full rank of the robot system. Thus, the torque $\boldsymbol{\tau}$ may come out with a different solution each time when solving HQP. Since obtaining a continuous torque solution is important for robot control, it is necessary to regularize the solution by solving additional QP as shown below:

$$\min_{\ddot{\mathbf{q}}, \boldsymbol{\tau}} \|\boldsymbol{\tau}\|^2, \quad (9)$$

subject to

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{h} = \boldsymbol{\tau}, \\ \ddot{\mathbf{x}}_{d_l} + \mathbf{w}_l^* = \mathbf{J}_l\dot{\mathbf{q}} + \mathbf{J}_l\ddot{\mathbf{q}}, \quad \forall l \in 1, \dots, k, \\ \underline{\ddot{\mathbf{q}}} \leq \ddot{\mathbf{q}} \leq \overline{\ddot{\mathbf{q}}}, \\ \underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} \leq \overline{\boldsymbol{\tau}}. \end{cases} \quad (10)$$

In the above QP example, torque is minimized, which can be used to configure various cost functions such as joint kinetic energy or joint acceleration energy. After that, we can finally obtain an optimal solution from the last QP.

3. Reducing decision variables of HQP

In this section, we present a novel QP formulation with reduced decision variables to decrease the computational cost of the HQP-based method. The proposed method can make the computational cost smaller than conventional HQP-based

method by resizing the decision variables size while satisfying a dynamical relationship of rigid body. As a result, the proposed method can create the same control solution as the conventional method in Section 2 with a decreased computation time. By reducing the computational cost, it can be more practically used for real-time control with high control frequency.

From (1), joint acceleration and torque relationship can be obtained as follows:

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}(\boldsymbol{\tau} - \mathbf{h}). \quad (11)$$

Then, by substituting (10) into (2), i.e., $\ddot{\mathbf{x}} = \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}\ddot{\mathbf{q}}$, we can derive an expression of task space acceleration as

$$\ddot{\mathbf{x}} = \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}\mathbf{M}^{-1}(\boldsymbol{\tau} - \mathbf{h}). \quad (12)$$

By utilizing (12), joint acceleration term $\ddot{\mathbf{q}}$ can be removed still considering both (1) and (2). However, the Coriolis/centrifugal, and gravity force acting on joints cannot be fully compensated by the above formula since \mathbf{h} is compensated on the task space by pre-multiplying $\mathbf{J}\mathbf{M}^{-1}$. As mentioned in the previous study [15], compensation of \mathbf{h} at the joint space is much efficient than compensating at the task space. Thus, we consider the robot dynamics that excludes the force vector \mathbf{h} for solving HQP and include it in the final torque equation after all computations are over. Accordingly, the task space acceleration equation can be described as follows:

$$\ddot{\mathbf{x}} = \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}\mathbf{M}^{-1}\boldsymbol{\tau}. \quad (13)$$

The above (13) replaces dynamics and task constraints in the conventional QP formula introduced in the previous section. Thus, the first QP can be reformulated as

$$\min_{\boldsymbol{\tau}, \mathbf{w}_1} \|\mathbf{w}_1\|^2, \quad (14)$$

subject to

$$\begin{cases} \ddot{\mathbf{x}}_{d_1} = \dot{\mathbf{J}}_1\dot{\mathbf{q}} + \mathbf{J}_1\mathbf{M}^{-1}\boldsymbol{\tau} + \mathbf{w}_1, \\ \underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} \leq \overline{\boldsymbol{\tau}}, \\ \underline{\mathbf{w}}_1 \leq \mathbf{w}_1 \leq \overline{\mathbf{w}}_1. \end{cases} \quad (15)$$

Compared to the first QP of the conventional HQP described as (3), the variable $\ddot{\mathbf{q}}$ is removed from the decision variable. This is possible since $\ddot{\mathbf{q}}$ related constraints are removed by applying

(13) as constraint instead of robot dynamics and task-joint relationship equation. Note that the joint acceleration inequality, i.e., $\ddot{\mathbf{q}} \leq \overline{\ddot{\mathbf{q}}} \leq \underline{\ddot{\mathbf{q}}}$, can be addressed by the joint torque inequality, i.e., $\underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} \leq \overline{\boldsymbol{\tau}}$. Therefore, the above can derive the same solution as the conventional method.

Same as the conventional HQP manner introduced in the previous section, lower priority tasks can be solved by the cascade QPs. The QP for a k -th prioritized task can be expressed as follows:

$$\min_{\boldsymbol{\tau}, \mathbf{w}_k} \|\mathbf{w}_k\|^2, \quad (16)$$

subject to

$$\begin{cases} \ddot{\mathbf{x}}_{d_l} = \dot{\mathbf{J}}_l\dot{\mathbf{q}} + \mathbf{J}_l\mathbf{M}^{-1}\boldsymbol{\tau} + \mathbf{w}_l^*, & \forall l \in 1, \dots, k-1, \\ \ddot{\mathbf{x}}_{d_k} = \dot{\mathbf{J}}_k\dot{\mathbf{q}} + \mathbf{J}_k\mathbf{M}^{-1}\boldsymbol{\tau} + \mathbf{w}_k, \\ \underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} \leq \overline{\boldsymbol{\tau}}, \\ \underline{\mathbf{w}}_k \leq \mathbf{w}_k \leq \overline{\mathbf{w}}_k. \end{cases} \quad (17)$$

In addition, when the tasks do not require the full rank of the robot, a QP for torque regularization can be solved as (9). It can be expressed as follows:

$$\min_{\boldsymbol{\tau}} \|\boldsymbol{\tau}\|^2, \quad (18)$$

subject to

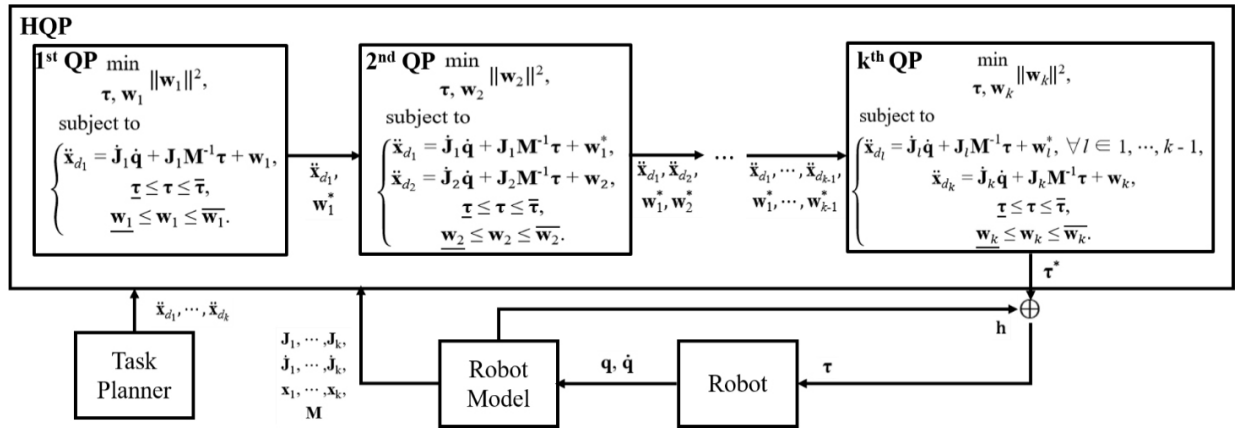
$$\begin{cases} \ddot{\mathbf{x}}_{d_l} = \dot{\mathbf{J}}_l\dot{\mathbf{q}} + \mathbf{J}_l\mathbf{M}^{-1}\boldsymbol{\tau} + \mathbf{w}_l^*, & \forall l \in 1, \dots, k, \\ \underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} \leq \overline{\boldsymbol{\tau}}. \end{cases} \quad (19)$$

The resulting torque $\boldsymbol{\tau}$ from the last QP is now expressed $\boldsymbol{\tau}^*$ as an optimal solution for the controlling hierarchical tasks. Finally, the desired torque $\boldsymbol{\tau}$ can be determined by adding the Coriolis/centrifugal force \mathbf{h} .

$$\boldsymbol{\tau} = \boldsymbol{\tau}^* + \mathbf{h}. \quad (20)$$

Since \mathbf{h} is added to $\boldsymbol{\tau}^*$ in the last QP solution, the compensation for \mathbf{h} is completed in joint space as aforementioned. The entire process of proposed method can be seen at [Fig. 1].

As can be seen in the above QP formulations, since $\ddot{\mathbf{q}}$ is removed, the size of the decision variable reduces from $2n + m_k$ to $n + m_k$ where m_k is the DoFs of the k -th task. For the QP for regularization, the size of the decision variable reduces from $2n$ to n . In general, since the task DoFs is smaller than robot DoFs,



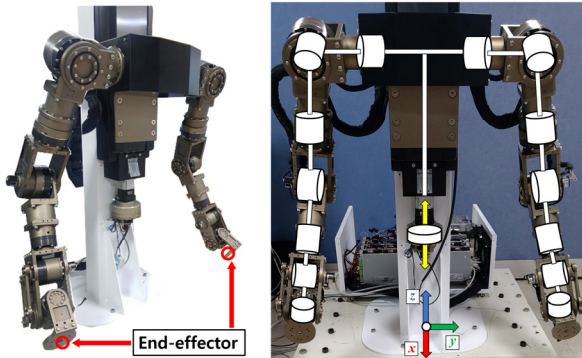
[Fig. 1] Overview of the algorithm of the proposed method

$m_k \leq n$, our proposed method reduces its size by at least 33%. Accordingly, the total computation time of the HQP reduces. Also, by following the cascade manner of the conventional HQP, the proposed method can achieve the advantages of the previous method.

4. Experimental Verification

4.1 Experimental setup

The proposed method is evaluated by the experiments with the 15-DoFs dual-arm manipulator shown in [Fig. 2]. The dual-arm manipulators have 7 DoFs for each arm and a 1 DoF prismatic joint for vertical motion. The HQP is solved by the library qpOASES, and rigid body dynamics parameters are obtained from the library RBDL and URDF model of the robot. Eigen library is utilized for linear algebra computation. The controller is implemented in the computer with Intel 3.1 GHz hexadecore processor and 32 GByte memory with Linux and Xenomai



[Fig. 2] Dual-arm manipulator (left) and schematic kinematic structure of the robot (right)

for 1 kHz real-time control. For joint torque control, Elmo motor drivers are utilized with an EtherCAT communication interface. First, we verify efficacy of the proposed method in real robot control in Section 4.2, and then we conducted comparative experiments in Section 4.3 to show the difference from the conventional method.

4.2 Experiments Results of the Proposed Method

To verify efficacy of the proposed method in the real robot, gravity compensation, trajectory tracking control, compliance against interaction force, null-space motion control experiments are conducted. Snapshots of the experimental results are shown in [Fig. 3].

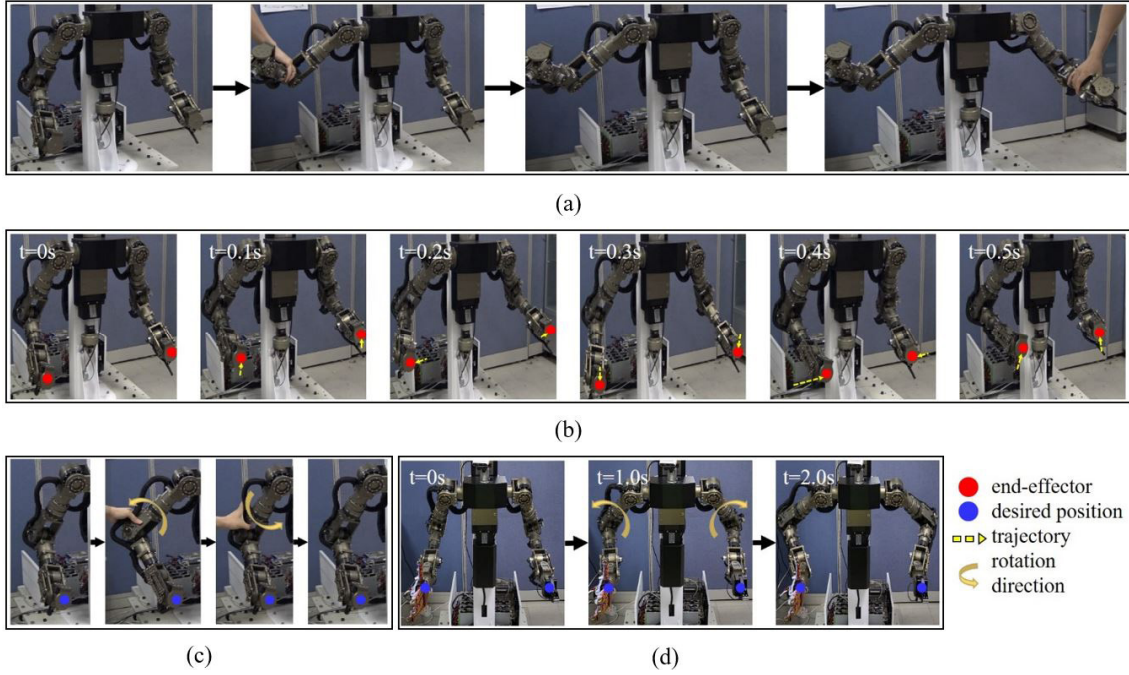
Two prioritized tasks are performed in the experiments. The primary task is controlling the end-effectors at both arms to track a trajectory generated by the cubic spline function. Each end-effector is controlled in 6 DoFs motion, i.e., position and orientation at the Cartesian coordinate.

For the trajectory tracking, a PD control scheme is applied for determining reference acceleration as

$$\ddot{\mathbf{x}}_{d_1} = k_p (\mathbf{x}_{d_1} - \mathbf{x}_1) + k_v (\dot{\mathbf{x}}_{d_1} - \dot{\mathbf{x}}_1), \quad (21)$$

where k_p, k_v are gains, and $\mathbf{x}_{d_1}, \dot{\mathbf{x}}_{d_1}$ are desired position and velocity vector for the first priority task, respectively. The second priority task is a generation of virtual damping at all joints. It is implemented by the following equation:

$$\ddot{\mathbf{x}}_{d_2} = -k_{vj} \dot{\mathbf{q}}, \quad (22)$$

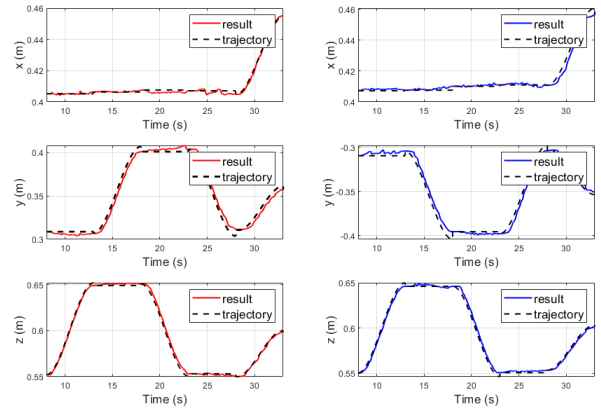


[Fig. 3] Snapshots of the experimental results with the proposed HQP method: (a) The reaction of the robot when compensating the Coriolis/centrifugal and gravity forces of the robot. (b) The resulting motion of the dual-arm manipulators with proposed HQP. (c) The reaction of the robot when a person applies an interaction force while regulating the position of the end-effector. (d) The null-space motion control of the robot using the redundancy while regulating the positions of the end-effectors

where $\ddot{\mathbf{x}}_{d_2} \in \mathbb{R}^n$ is the desired acceleration of the joints, which is the objective for optimization of the second QP, and k_{vj} is a damping gain. For consistency of expression, the notation of the second prioritized task space is expressed as \mathbf{x}_2 rather than \mathbf{q} even it describes the joint space. In the experiments, gains are set as $k_p = 100$, $k_v = 5$, $k_{vj} = 10$.

Before verifying the feed-back controller, a gravity compensation experiment is performed to see the validity of \mathbf{h} term compensation. The result of the gravity compensation experiment can be seen in [Fig. 3(a)]. It is implemented by setting the desired task space accelerations to zero, i.e., $\ddot{\mathbf{x}}_{d_1} = \ddot{\mathbf{x}}_{d_2} = \mathbf{0}$. The robot then adjusts to the external force applied by a person and it maintains the position when the external force does not exist. Through this result, it can be seen that the robot moves as if it is in a state of zero gravity as intended.

The trajectory tracking control motion of the end-effectors with the proposed method can be seen in [Fig. 3(b)]. The trajectory and control result of the robot with the proposed HQP can be seen in [Fig. 4]. The end-effectors track the desired position within an error of about 3mm. Through this result, one can notice that the controller is effectively operating. Note that the tracking performance can be improved by increasing control



[Fig. 4] Position errors of the end-effectors at the left arm (right-side figures) and the right arm (left-side figures)

gains to reduce position error.

Compliant behavior of the robot can be seen in [Fig. 3(c)]. The reaction of the robot can be observed during a person applies an interaction force to the robot while the robot is regulating the position of the end-effector. The robot is controlled to keep the initial position instead of tracking the trajectory with the same task introduced in the previous experiment. Through this result, one can notice that the joints corresponding to the redundancy compliantly behave against the external force while maintaining the position of the end-effector.

In [Fig. 3(d)], while the robot is controlled to regulate the positions of the end-effectors on the first priority task, The PD control scheme is applied to control the shoulder yaw joints angles of both arms on the second priority task as following:

$$\ddot{\mathbf{x}}_{d_2} = k_{pj}(\mathbf{x}_{d_2} - \mathbf{x}_2) + k_{vj}(\dot{\mathbf{x}}_{d_2} - \dot{\mathbf{x}}_2), \quad (23)$$

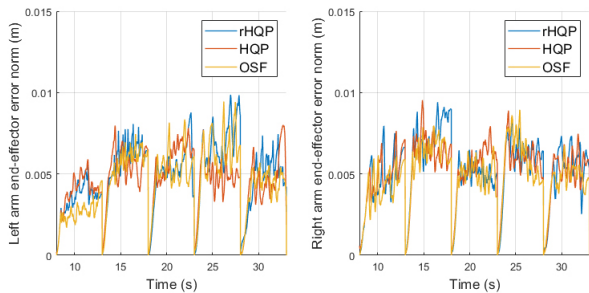
where $\mathbf{x}_2 \in \mathbb{R}^2$ describes the joint angle of the shoulder yaw joints, \mathbf{x}_{d_2} , $\dot{\mathbf{x}}_{d_2}$, $\ddot{\mathbf{x}}_{d_2}$ are the desired position, velocity, and acceleration vector of the second priority task, respectively. The desired angle of the shoulder yaw joint is commanded 0 to 45 degrees and the motion of the robot using redundancy is controlled successfully while regulating the positions of the end-effectors. Note that this is the same effect of the well-known null-space-based hierarchical control method.

4.3 Comparative Experiments

For the comparison, HQP and OSF are also performed to execute the trajectory tracking control with the same hierarchical tasks introduced in the previous subsection. The proposed method is denoted as rHQP in the results for the sake of brevity.

To compare the computation time fairly, we structured the formulation of the three methods to have the same physical meaning. As a result, the control performance of tracking desired positions with the same control gain can be seen in [Fig. 5]. Although the OSF does not take into account inequality constraints, since the performed task does not result in inequality bounding, the three methods generate a similar control solution.

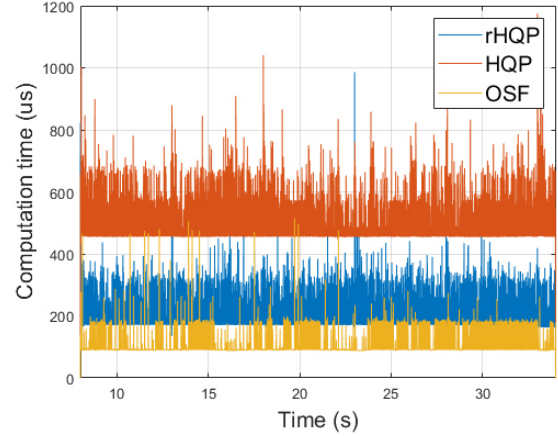
The comparison of computational time for the three methods can be seen in [Table 1] and [Fig. 6]. The proposed rHQP solves about 2.44 times faster than the conventional HQP on average. In addition, when observing the maximum time required for the



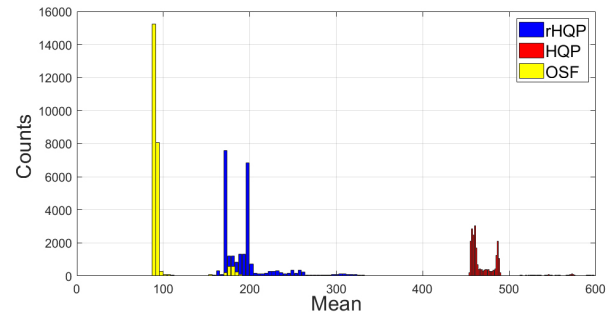
[Fig. 5] Position error norms of the end-effectors at left arm (left-side figure) and the right arm (right-side figure)

[Table 1] Computation time of controllers

Method	Avg. time	Max. time	Min. time
rHQP	198.07 us	986.27 us	135.24 us
HQP	482.73 us	1174.9 us	423.55 us
OSF	98.073 us	514.53 us	86.750 us

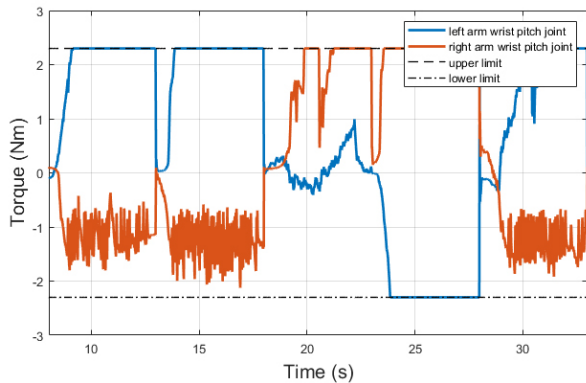


[Fig. 6] Computation time consumed with rHQP (blue), HQP (red), and OSF (yellow)



[Fig. 7] Distribution of computation time of the rHQP (blue), HQP (red), and OSF (yellow)

computation in [Fig. 6], the proposed rHQP is suitable for 1 kHz control frequency since it can always solve within 1 ms. On the other hand, the HQP can disturb the real-time system since the HQP irregularly requires near 1 ms and sometimes over 1 ms. In a real-time system, time consumed for other computations such as communication, and calculation of kinematics and dynamics is required, so even if the calculation time of the controller does not exceed 1 ms, it may be a problem. The distribution of computation time of each method can be seen in [Fig. 7]. The proposed rHQP requires about 986.27us and the conventional HQP requires about 1174.9us for maximum computing time. However, a standard deviation of the rHQP is about 36.8 and the computing time of rHQP is concentrated near the average as can be seen in [Fig. 7]. Accordingly, one can notice that the worst



[Fig. 8] Joint torque constraints of the wrist pitch joints and desired joint torque result

case occurs very rarely.

Although the rHQP can solve faster than the HQP, it is still slower than the OSF. However, the rHQP can satisfy inequality constraints, unlike the OSF. To verify that the solution which is obtained from the rHQP can satisfy the inequality constraints strictly, we performed an additional experiment. In the experiment, the task introduced in this section is performed while the boundary of the wrist pitch joint torque is set to narrow as $-2.3 \text{ Nm} \sim 2.3 \text{ Nm}$. The joint torque result of the wrist pitch joints can be seen in [Fig. 8]. As can be seen in the plotting result, the control solution from the proposed method does not exceed the boundary and satisfies the inequality constraint while performing the task.

5. Conclusion

In this paper, we propose a decision variable reduced HQP-based robot controller. Applying the relationship between task space acceleration with joint space torque leads to a reduction of computational costs. As a result, the reduced HQP can solve an optimization problem much faster than the previous method. Through this, an articulated robot controller that can consider the hierarchical tasks and inequality constraints can be practically used in a system that requires a high control frequency. The performance of the proposed method is verified through real robot experiments.

In our future study, the HQP formulation will be extended to manage workspace and singularity avoidance problems. Since the workspace and singularity are not described explicitly in an equation for the HQP formula, additional study is required for practical use. Accordingly, the practicality of the HQP-based controller can be improved.

References

- [1] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, Feb., 1987, DOI: 10.1109/JRA.1987.1087068.
- [2] K. Jang, S. Kim, S. Park, and J. Park, "A unified framework for overcoming motion constraints of robots using task transition algorithm," *The Journal of Korea Robotics Society*, vol. 13, no. 2, pp. 129-141, June, 2018, DOI: 10.7746/jkros.2018.13.2.129.
- [3] Z. Leon and B. Nemeč, "Kinematic control algorithms for on-line obstacle avoidance for redundant manipulators," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, Lausanne, Switzerland, pp. 1898-1903, 2002, DOI: 10.1109/IRDS.2002.1044033.
- [4] C. Collette, A. Micaelli, C. Andriot, and P. Lemerle, "Robust balance optimization control of humanoid robots with multiple non coplanar grasps and frictional contacts," *2008 IEEE International Conference on Robotics and Automation*, Pasadena, USA, pp. 3187-3193, 2008, DOI: 10.1109/ROBOT.2008.4543696.
- [5] M. de Lasa and A. Hertzmann, "Prioritized optimization for task-space control," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, pp. 5755-5762, 2009, DOI: 10.1109/IROS.2009.5354341.
- [6] M. de Lasa, I. Mordatch, and A. Hertzmann, "Feature-based locomotion controllers," *ACM Transactions on Graphics*, vol. 29, no. 4, pp. 1-10, 2010, DOI: 10.1145/1778765.1781157.
- [7] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 473-491, 2016, DOI: 10.1007/s10514-015-9476-6.
- [8] N. Mansard, "A dedicated solver for fast operational-space inverse dynamics," *2012 IEEE International Conference on Robotics and Automation*, Saint Paul, MN, USA, pp. 4943-4949, 2012, DOI: 10.1109/ICRA.2012.6224851.
- [9] Y. Lee, J. Ahn, J. Lee, and J. Park, "Computationally Efficient HQP-Based Whole-Body Control Exploiting the Operational-Space Formulation," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic, pp. 5174-5179, 2021, DOI: 10.1109/IROS51168.2021.9636867.
- [10] 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, DOI: 10.1177/0278364914521306.
- [11] S. Kim, K. Jang, S. Park, Y. Lee, S. Y. Lee, and J. Park, "Continuous task transition approach for robot controller based on hierarchical quadratic programming," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1603-1610, 2019, DOI: 10.1109/LRA.2019.2896769.
- [12] H. Park, P. Kim, P. Park, J.-R. Jang, Y.-D. Shin, J.-H. Bae, J.-H. Park, and M.-H. Baeg, "Robotic peg-in-hole assembly by hand

arm coordination,” *The Journal of Korea Robotics Society*, vol. 10, no. 1, pp. 42-51, March, 2015, DOI: 10.7746/jkros.2015.10.1.042.

[13] D. J. Braun., Y. Chen., and L. Li., “Operational Space Control Under Actuation Constraints Using Strictly Convex Optimization,” *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 302-309, 2020, DOI: 10.1109/TRO.2019.2943057.

[14] 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, DOI: 10.1177/0278364908091463.

[15] O. Kanoun, F. Lamiroux, and P.-B. Wieber, “Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task,” *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785-792, 2011, DOI: 10.1109/TRO.2011.2142450.



박민규

2017~현재 광운대학교 로봇학부 학부생

관심분야: Robot Control, Mobile Manipulator, Mobile Robot



김동환

2015 ~현재 광운대학교 로봇학부 학부생

관심분야: Robot Control, Robot Manipulator, Mobile Robot



오용환

1991 포항공과대학교 기계공학과(공학사)

1993 포항공과대학교 기계공학과(공학석사)

1999 포항공과대학교 기계공학과(공학박사)

1999~2000 Department of Mechanical Engineering, Waseda University, Japan, 박사후연구원

2001~현재 한국과학기술연구원 책임연구원

관심분야: Mechanical Design of Robotic System, Sensor-based Bipedal Walking, Hand-arm Coordination, Motion/interaction Control of Redundant Manipulators



이이수

2008 서울대학교 조선해양공학과(공학사)

2010 서울대학교 조선해양공학과(공학석사)

2017 서울대학교 융합과학기술대학원 지능형시스템(공학박사)

2018~2020 Istituto Italiano di Tecnologia, Italy, 박사후 연구원

2020~현재 한국과학기술연구원 선임연구원

관심분야: Robot manipulation, Whole-body Control, Humanoid robots