



Weighted hierarchical quadratic programming: assigning individual joint weights for each task priority

Keunwoo Jang¹ · Sanghyun Kim¹ · Suhan Park¹ · Junhyung Kim¹ · Jaeheung Park^{1,2}

Received: 15 November 2021 / Accepted: 1 June 2022 / Published online: 5 July 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract

This paper proposes a novel method that computes the optimal solution of the weighted hierarchical optimization problem for both equality and inequality tasks. The method is developed to resolve the redundancy of robots with a large number of Degrees of Freedom (DoFs), such as a mobile manipulator or a humanoid, so that they can execute multiple tasks with differently weighted joint motion for each priority level. The proposed method incorporates the weighting matrix into the first-order optimality condition of the optimization problem and leverages an active-set method to handle equality and inequality constraints. In addition, it is computationally efficient because the solution is calculated in a weighted joint space with symmetric null-space projection matrices for propagating recursively to a low priority task. Consequently, robots that utilize the proposed method effectively show whole-body motions handling prioritized tasks with differently weighted joint spaces. The effectiveness of the proposed method was validated through experiments with a nonholonomic mobile manipulator as well as a humanoid.

Keywords Redundant robots · Whole-body control · Optimization · Mobile manipulators

1 Introduction

Robots with high Degrees of Freedom (DoFs) such as mobile manipulators and humanoids are designed for human-centered environments. To control these robots, whole-body control frameworks [13,15] have been used to perform various prioritized tasks to consider motion distribution with kinematic redundancy. However, in practice, depending on the scenarios (e.g., locomotion, manipulation, and loco-manipulation), it is difficult to generate natural whole-body behavior only with whole-body controllers. This is because typical whole-body controllers produce joint motion minimizing the same metric for all prioritized tasks. For example, when the mobile manipulator tracks the desired trajectory by using a whole-body task, it often comes into singular configuration or reaches the boundary of the workspace due to dynamic and kinematic difference of mobile base and manipulator [1,17]. Also, during locomotion phase of the

humanoid, the movement of the upper body by whole-body control may adversely affect walking performance [26].

In this paper, we propose a novel Weighted Hierarchical Quadratic Programming (WHQP) framework to characterize joint movement for each task. By combining two concepts of HQP [6] and weighted least-squares norm [2], it can handle various inequality and equality tasks with differently weighted joint motion for each priority effectively. Consequently, the proposed controller can generate natural whole-body behavior without additional subtasks which restrict undesirable movements, as shown in Fig. 1.

1.1 Related works

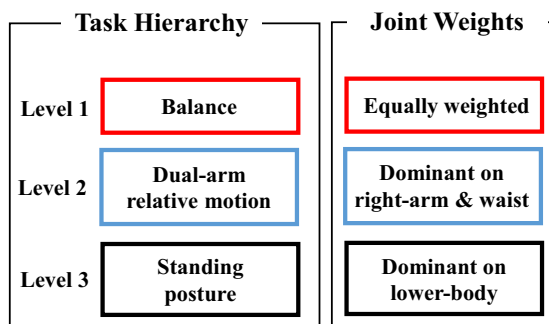
HQP has been actively studied in that it can handle both equality and inequality tasks while ensuring strict priorities of tasks. Kim et al. [12] proposed an HQP-based task transition method that can insert, remove, swap the priorities of the tasks while ensuring continuous control inputs. Tassi et al. [24] extended HQP in order to produce an impedance-like motion under external disturbance by augmenting the variable for Cartesian velocity. To enhance the computational efficiency, Lee et al. [14] utilizes operational space formu-

✉ Sanghyun Kim
kim87@kimm.re.kr

¹ Seoul National University, Suwon, Republic of Korea

² Advanced Institute of Convergence Technology, Suwon, Republic of Korea

Fig. 1 An example application of the proposed method with individual weighting matrix for each task priority: a box-taping scenario of humanoid



lation [19] so that the size of the decision variable in QP is reduced when controlling the whole-body of the humanoid.

On the other hand, the weighted least-squares norm which is based on the weighted pseudo-inverse can treat the redundancy of the robots without additional constraints. Dariush et al. [4] penalized the motion of joints by designing a weighting matrix that determines contribution to the main task according to the extent of the proximity to collision in order to avoid self-collision or obstacle. Similarly, Farelo et al. [8] generated optimized joint motion for wheelchair-mounted arm by using weighted pseudo-inverse that considers not only joint limit of arm but also motion limit of wheelchair. Tsuchihara et al. [25] utilized weighted pseudo-inverse that restricts chest motion of humanoid to improve stability in manipulation. Park et al. [21] and Choi et al. [3] combined the task-priority method and the weighted pseudo-inverse method in order to generate the joint motion minimizing the residual error caused by singularity-robust framework. However, since these methods treat the same joint weights for all tasks, they cannot assign individual joint weights for each task priority.

To tackle this issue, Wu et al. [27] recently proposed a two-level prioritized whole-body Cartesian impedance controller with individual weighting matrices. In contrast to aforementioned methods [3,4,8,21,25] with the same weighted joint distribution for all tasks, the proposed controller used individual weighting matrices to generate different motion patterns for each task. However, it cannot calculate the solution when extending to multiple tasks since the main task and its null-space are only considered. Also, [27] cannot deal with inequality constraints.

1.2 Overview of the Paper

The proposed WHQP can efficiently compute an analytic solution in a weighted least-squares norm manner for prioritized equality and inequality tasks. By reformulating HQP with Complete Orthogonal Decomposition (COD) [6], the WHQP assigns the weighting matrix to each task hierarchy

and derives the first-order optimality conditions. Based on these conditions, the active-set method [6,16] is exploited to handle inequality tasks.

Therefore, the main contributions of this paper are as follows. First, our algorithm calculates the optimal solution given the joint weights for each priority level. Thus, the proposed algorithm helps to generate the preferred whole-body motion patterns according to various scenarios such as locomotion, manipulation, and loco-manipulation. Second, unlike previous works [3,4,8,21,25,27] with the weighted optimization problem, the proposed method can handle inequality constraints as well as equality constraints. Finally, by virtue of symmetric null-space projection matrices, the proposed solver is computationally efficient. We show that our formulation can control various redundant robots including a nonholonomic mobile manipulator and humanoid in real-time.

The remainder of the paper is organized as follows. In Sect. 2, mathematical problem statements are explained. Section 3 describes how to formulate the WHQP subject to equality tasks and calculate a solution in a weighted least-squares norm sense. Based on the formulation of the WHQP, inequality tasks are handled by using active-set method in Sect. 4. Next, Sects. 5 and 6 describe the experimental validations and discussion of the proposed WHQP, respectively. Finally, the paper is concluded in Sect. 7.

2 Problem statement

This section provides the modification of the weighted pseudo-inverse to handle individual weights of joints for each task hierarchy and its limitations.

2.1 Pseudo-inverse with weighted least-squares norm for each task

Let us assume that there are p prioritized tasks and p weighting matrices for each task of n -DoFs redundant robot:

$$\dot{x}_k = J_k \dot{q}, \quad (k = 1, \dots, p) \tag{1}$$

where $\dot{x}_k \in \mathbb{R}^{m_k}$, $J_k \in \mathbb{R}^{m_k \times n}$, and $q \in \mathbb{R}^n$ are the task space velocity, the Jacobian matrix, and the joint position of the robot, respectively. The priorities are indicated by the number of subscripts: the smaller the subscript number, the higher the task priority. Also, the weighting matrix $W_k \in \mathbb{R}^{n \times n}$ is assumed to be symmetric and positive definite.

To execute the first task \dot{x}_1 while assigning dominant joints or optimizing a performance criterion through the weighting matrix W_1 , an optimization problem is formulated as

$$\begin{aligned} \min_{\dot{q}_1} & \quad \frac{1}{2} \dot{q}_1^T W_1 \dot{q}_1, \\ \text{s. t.} & \quad J_1 \dot{q}_1 = \dot{x}_1. \end{aligned} \tag{2}$$

The optimal solution $\dot{q}_1^* \in \mathbb{R}^n$ is calculated analytically as

$$\dot{q}_1^* = J_1^{W_1+} \dot{x}_1 = W_1^{-1} J_1^T (J_1 W_1^{-1} J_1^T)^{-1} \dot{x}_1, \tag{3}$$

where $J_1^{W_1+} \in \mathbb{R}^{n \times m_1}$ is the weighted pseudo-inverse of J_1 .

By utilizing the weighted pseudo-inverse solution in (3), a solution that executes multiple tasks with differently weighted joint solution for each task priority can be computed. Considering the secondary task \dot{x}_2 with the corresponding weighting matrix W_2 , the total solution \dot{q}_2^* for two prioritized tasks would be

$$\dot{q}_2^* = \dot{q}_1^* + \hat{\dot{q}}_2^*, \tag{4}$$

where $\hat{\dot{q}}_2^*$ denotes the contribution to the secondary task without modifying the first task. This can be obtained by solving the following optimization problem as

$$\begin{aligned} \min_{\dot{q}_2} & \quad \frac{1}{2} \dot{q}_2^T W_2 \dot{q}_2, \\ \text{s. t.} & \quad J_1 \dot{q}_2 = \dot{x}_1 - J_1 \dot{q}_1^* = 0, \\ & \quad J_2 \dot{q}_2 = \dot{x}_2 - J_2 \dot{q}_1^*, \end{aligned} \tag{5}$$

The solution for (5) is calculated as

$$\dot{q}_2^* = (J_2 N_1^{W_2})^{W_2+} (\dot{x}_2 - J_2 \dot{q}_1^*). \tag{6}$$

where $N_1^{W_2} = I - W_2^{-1} J_1^T (J_1 W_2^{-1} J_1^T)^{-1} J_1 \in \mathbb{R}^{n \times n}$ is the projector onto the null-space of J_1 weighted by W_2 . Note that the projector $N_1^{W_2}$ is idempotent, but not symmetric.

Thus, it is straightforward to obtain a general solution for p prioritized tasks:

$$\dot{q}_p^* = \sum_{k=1}^p (J_k N_{k-1}^{W_k})^{W_k+} (\dot{x}_k - J_k \dot{q}_{k-1}^*), \tag{7}$$

where $N_0^{W_1} = I$, $\dot{q}_0^* = 0$, and $N_{k-1}^{W_k}$ is the projector onto the null-space of the augmented Jacobian $J_{k-1} = [J_1^T, J_2^T, \dots, J_{k-1}^T]^T \in \mathbb{R}^{\sum_{i=1}^{k-1} m_i \times n}$ weighted by W_k . At this point, all tasks are assumed to be full rank. The projector $N_{k-1}^{W_k}$ can be computed in two ways as

$$N_{k-1}^{W_k} = N_{[1]}^{W_k} N_{[2]}^{W_k} \cdots N_{[k-1]}^{W_k} = \prod_{j=1}^{k-1} N_{[j]}^{W_k}, \tag{8a}$$

$$N_{k-1}^{W_k} = I - \underline{J}_{k-1}^{W_k+} \underline{J}_{k-1}, \tag{8b}$$

where $N_{[j]}^{W_k} = I - (J_j \prod_{i=0}^{j-1} N_{[i]}^{W_k})^{W_k+} (J_j \prod_{i=0}^{j-1} N_{[i]}^{W_k})$, $N_{[0]}^{W_k} = I$, and $N_{[1]}^{W_k} = I - W_k^{-1} J_1^T (J_1 W_k^{-1} J_1^T)^{-1} J_1$.

2.2 Problem statement

When computing the null-space projection matrix in (8), the matrix is asymmetric and its size remains constant. Owing to these properties, the computational cost increases exponentially when propagating to multiple tasks. To improve computational efficiency, [5] proposed a scheme to accelerate the computation by decomposing a symmetric null-space projection matrix. Therefore, our goal is to efficiently compute weighted least-squares norm solution by transforming the asymmetric null-space projection matrix in (8) into a symmetric matrix. Furthermore, we extend the pseudo-inverse with weighted least-squares norm for each task priority of Sect. 2.1 to deal with inequality as well as equality tasks.

3 WHQP with equality tasks

This section formulates the WHQP that minimizes the violation of the equality task in a weighted least-squares norm manner. When formulating the WHQP, Jacobian matrix and joint velocity are transformed into a weighted Jacobian matrix and joint velocity. This makes the null-space projection matrix symmetric, which enables the proposed solution to be computed more efficiently than the traditional method using weighted pseudo-inverse.

For the first task \dot{x}_1 , the weighted Jacobian matrix J_{1, W_1} and weighted joint velocity \dot{q}_{W_1} are defined as

$$\begin{aligned} J_{1, W_1} &= J_1 W_1^{-\frac{1}{2}}, \\ \dot{q}_{W_1} &= W_1^{\frac{1}{2}} \dot{q}. \end{aligned} \tag{9}$$

Then, the WHQP is formulated as

$$\begin{aligned} \min_{\dot{q}_{W_1}, s_1} & \quad \frac{1}{2} \|s_1\|_2^2, \\ \text{s. t.} & \quad J_{1, W_1} \dot{q}_{W_1} = \dot{x}_1 + s_1, \end{aligned} \tag{10}$$

where $s_1 \in \mathbb{R}^{m_1}$ is the slack variable that alleviates the task \dot{x}_1 . Note that the variable to optimize is not \dot{q} , but \dot{q}_{W_1} .

To solve this problem, the Lagrangian \mathcal{L}_1 is computed as

$$\mathcal{L}_1 = \frac{1}{2} s_1^T s_1 + \lambda_1^T (J_{1,W_1} \dot{q}_{W_1} - \dot{x}_1 - s_1), \tag{11}$$

where $\lambda_1 \in \mathbb{R}^{m_1}$ is the Lagrange multiplier. From the first-order optimality conditions, the Lagrangian differentiated by λ_1, s_1 , and \dot{q}_{W_1} should be zero:

$$\begin{aligned} \frac{\partial \mathcal{L}_1}{\partial \lambda_1} &= J_{1,W_1} \dot{q}_{W_1} - \dot{x}_1 - s_1 = 0, \\ \frac{\partial \mathcal{L}_1}{\partial s_1} &= s_1 - \lambda_1 = 0, \\ \frac{\partial \mathcal{L}_1}{\partial \dot{q}_{W_1}} &= J_{1,W_1}^T \lambda_1 = 0. \end{aligned} \tag{12}$$

To obtain the weighted solution $\dot{q}_{W_1}^*$, the pseudo-inverse of the weighted Jacobian matrix J_{1,W_1} is computed using COD [6,9] which is cheaper than computing the singular value decomposition as

$$\begin{aligned} J_{1,W_1} &= [V_{1,W_1} \ U_{1,W_1}] \begin{bmatrix} 0 & 0 \\ L_{1,W_1} & 0 \end{bmatrix} [Y_{1,W_1} \ Z_{1,W_1}]^T \\ &= U_{1,W_1} L_{1,W_1} Y_{1,W_1}^T, \end{aligned} \tag{13}$$

where $U_{1,W_1} \in \mathbb{R}^{m_1 \times r_1}$ and $V_{1,W_1} \in \mathbb{R}^{m_1 \times (m_1 - r_1)}$ are the orthonormal bases for the column space of J_{1,W_1} . r_1 is the rank of J_{1,W_1} . $Y_{1,W_1} \in \mathbb{R}^{n \times r_1}$ and $Z_{1,W_1} \in \mathbb{R}^{n \times (n - r_1)}$ are the orthonormal bases for the row space of J_{1,W_1} . $L_{1,W_1} \in \mathbb{R}^{r_1 \times r_1}$ is a lower triangular matrix. With this decomposition, the weighted solution and slack variable satisfying (12) are obtained as

$$\dot{q}_{W_1}^* = J_{1,W_1}^+ \dot{x}_1 = Y_{1,W_1} L_{1,W_1}^{-1} U_{1,W_1}^T \dot{x}_1, \tag{14}$$

$$s_1^* = U_{1,W_1} U_{1,W_1}^T \dot{x}_1 - \dot{x}_1 = -V_{1,W_1} V_{1,W_1}^T \dot{x}_1, \tag{15}$$

where J_{1,W_1}^+ denotes the pseudo-inverse of J_{1,W_1} . Based on (13) and (15), the last optimality condition of (12) is satisfied. In the end, the complete solution \dot{q}_1^* for the first task is obtained by transforming the weighted solution to the original joint space as

$$\dot{q}_1^* = W_1^{-\frac{1}{2}} \dot{q}_{W_1}^*. \tag{16}$$

Similarly, the WHQP for the secondary task \dot{x}_2 is formulated as

$$\begin{aligned} \min_{\dot{q}_{W_2}, s_2} & \frac{1}{2} \|s_2\|_2^2. \\ \text{s. t.} & J_{1,W_2} \dot{q}_{W_2} = 0, \\ & J_{2,W_2} \dot{q}_{W_2} = \dot{x}_2 - J_2 \dot{q}_1^* + s_2. \end{aligned} \tag{17}$$

Similar to (6), the weighted solution $\dot{q}_{W_2}^*$ lies in the null-space of J_{1,W_2} and at the same time executes the residual task after subtracting the effect of \dot{q}_1^* on the secondary task space. It is obtained as

$$\dot{q}_{W_2}^* = (J_{2,W_2} N_{1,W_2})^+ (\dot{x}_2 - J_2 \dot{q}_1^*), \tag{18}$$

where $N_{1,W_2} = I - J_{1,W_2}^+ J_{1,W_2}$ is the null-space projection matrix of J_{1,W_2} .

Unlike (6), N_{1,W_2} is both idempotent and symmetric. Thus, N_{1,W_2} can be represented as

$$N_{1,W_2} = Z_{1,W_2} Z_{1,W_2}^T, \tag{19}$$

where $Z_{1,W_2} \in \mathbb{R}^{n \times (n - r_1)}$ is the null-space bases of J_{1,W_2} and obtained by decomposing J_{1,W_2} . Then, the solution in (18) can be represented in a more efficient form [5] as

$$\begin{aligned} \dot{q}_{W_2}^* &= Z_{1,W_2} (J_{2,W_2} Z_{1,W_2})^+ (\dot{x}_2 - J_2 \dot{q}_1^*) \\ &= Y_{2,W_2} L_{2,W_2}^{-1} U_{2,W_2}^T (\dot{x}_2 - J_2 \dot{q}_1^*), \end{aligned} \tag{20}$$

where $Y_{2,W_2} = Z_{1,W_2} \tilde{Y}_{2,W_2} \in \mathbb{R}^{n \times r_2}$, $\tilde{Y}_{2,W_2} \in \mathbb{R}^{(n - r_1) \times r_2}$, $L_{2,W_2} \in \mathbb{R}^{r_2 \times r_2}$, and $U_{2,W_2} \in \mathbb{R}^{m_2 \times r_2}$ are obtained by decomposing $J_{2,W_2} Z_{1,W_2} \in \mathbb{R}^{m_2 \times (n - r_1)}$ as

$$J_{2,W_2} Z_{1,W_2} = [V_{2,W_2} \ U_{2,W_2}] \begin{bmatrix} 0 & 0 \\ L_{2,W_2} & 0 \end{bmatrix} [\tilde{Y}_{2,W_2} \ \tilde{Z}_{2,W_2}]^T. \tag{21}$$

Using (20), the optimal slack variable s_2^* is computed as

$$\begin{aligned} s_2^* &= J_{2,W_2} \dot{q}_{W_2}^* - \dot{x}_2 + J_2 \dot{q}_1^*, \\ &= U_{2,W_2} U_{2,W_2}^T (\dot{x}_2 - J_2 \dot{q}_1^*) - \dot{x}_2 + J_2 \dot{q}_1^* \\ &= V_{2,W_2} V_{2,W_2}^T (J_2 \dot{q}_1^* - \dot{x}_2). \end{aligned} \tag{22}$$

Therefore, the total solution for two prioritized tasks \dot{x}_1, \dot{x}_2 is

$$\dot{q}_2^* = W_1^{-\frac{1}{2}} \dot{q}_{W_1}^* + W_2^{-\frac{1}{2}} \dot{q}_{W_2}^*. \tag{23}$$

More generally, WHQP for the k th task \dot{x}_k is formulated as

$$\begin{aligned} \min_{\dot{q}_{W_k}, s_k} & \frac{1}{2} \|s_k\|_2^2. \\ \text{s. t.} & J_{k,W_k} \dot{q}_{W_k} = \dot{x}_k - J_k \dot{q}_{k-1}^* + s_k \\ & \underline{J}_{k-1,W_k} \dot{q}_{W_k} = 0 \end{aligned} \tag{24}$$

where $\underline{J}_{k-1,W_k} = [J_{1,W_k}^T, J_{2,W_k}^T, \dots, J_{k-1,W_k}^T]^T$ is the augmented Jacobian matrix weighted by $W_k^{-\frac{1}{2}}$.

To solve (24), the Lagrangian \mathcal{L}_k is constructed as

$$\mathcal{L}_k = \frac{1}{2} s_k^T s_k + \lambda_k^T (J_{k,W_k} \dot{q}_{W_k} - \dot{x}_k + J_k \dot{q}_{k-1}^* - s_k) + \underline{\lambda}_{k-1}^T (\underline{J}_{k-1,W_k} \dot{q}_{W_k}). \tag{25}$$

Then, the optimality conditions for the k th task are directly obtained as

$$\begin{aligned} \frac{\partial \mathcal{L}_k}{\partial \lambda_k} &= J_{k,W_k} \dot{q}_{W_k} - \dot{x}_k + J_k \dot{q}_{k-1}^* - s_k = 0, \\ \frac{\partial \mathcal{L}_k}{\partial \underline{\lambda}_{k-1}} &= \underline{J}_{k-1,W_k} \dot{q}_{W_k} = 0, \\ \frac{\partial \mathcal{L}_k}{\partial s_k} &= s_k - \lambda_k = 0, \\ \frac{\partial \mathcal{L}_k}{\partial \dot{q}_{W_k}} &= J_{k,W_k}^T \lambda_k + \underline{J}_{k-1,W_k}^T \underline{\lambda}_{k-1} = 0. \end{aligned} \tag{26}$$

To calculate the solution, the Jacobian matrices J_{k,W_k} , \underline{J}_{k-1,W_k} are decomposed as in [6]:

$$\begin{aligned} J_{k,W_k} &= [V_{k,W_k} \ U_{k,W_k}] \begin{bmatrix} N_{k,W_k} & 0 \\ M_{k,W_k} & L_{k,W_k} \end{bmatrix} [\underline{Y}_{k-1,W_k} \ Y_{k,W_k}]^T \\ &= E_{k,W_k} H_{k,W_k} \underline{Y}_{k,W_k}^T, \end{aligned} \tag{27}$$

where

$$\begin{aligned} N_{k,W_k} &= V_{k,W_k}^T J_{k,W_k} \underline{Y}_{k-1,W_k} \in \mathbb{R}^{(m_k-r_k) \times \sum_{i=1}^{k-1} r_i}, \\ M_{k,W_k} &= U_{k,W_k}^T J_{k,W_k} \underline{Y}_{k-1,W_k} \in \mathbb{R}^{r_k \times \sum_{i=1}^{k-1} r_i}, \\ \underline{Y}_{k-1,W_k} &= [Y_{1,W_k}, \dots, Y_{k-1,W_k}] \in \mathbb{R}^{n \times \sum_{i=1}^{k-1} r_i}, \\ Y_{k,W_k} &= Z_{k-1,W_k} \tilde{Y}_{k,W_k} \in \mathbb{R}^{n \times r_k}. \end{aligned} \tag{28}$$

V_{k,W_k} , U_{k,W_k} , L_{k,W_k} , and \tilde{Y}_{k,W_k} are obtained from decomposing $J_{k,W_k} Z_{k-1,W_k}$. The basis \underline{Y}_{k-1,W_k} is obtained by recursively decomposing $k - 1$ times from J_{1,W_k} to J_{k-1,W_k} . Using the representation of (27), the stacked matrix \underline{J}_{k-1,W_k} is described as

$$\begin{aligned} \underline{J}_{k-1,W_k} &= \begin{bmatrix} E_{1,W_k} \cdots & 0 \\ \vdots & \ddots \\ 0 & \cdots & E_{k-1,W_k} \end{bmatrix} \begin{bmatrix} H_{1,W_k} & 0 \\ \vdots & \vdots \\ N_{k-1,W_k} & 0 \\ M_{k-1,W_k} & L_{k-1,W_k} \end{bmatrix} \begin{bmatrix} Y_{1,W_k}^T \\ \vdots \\ Y_{k-1,W_k}^T \end{bmatrix} \\ &= \underline{E}_{k-1,W_k} \underline{H}_{k-1,W_k} \underline{Y}_{k-1,W_k}^T. \end{aligned} \tag{29}$$

Then, the weighted solution $\dot{q}_{W_k}^*$ satisfying (26) is computed as

$$\dot{q}_{W_k}^* = Y_{k,W_k} L_{k,W_k}^{-1} U_{k,W_k}^T (\dot{x}_k - J_k \dot{q}_{k-1}^*). \tag{30}$$

Next, the optimal slack variable s_k^* is obtained as

$$s_k^* = V_{k,W_k} V_{k,W_k}^T (J_k \dot{q}_{k-1}^* - \dot{x}_k). \tag{31}$$

The Lagrange multipliers, λ_k^* and $\underline{\lambda}_{k-1}^*$, satisfying (26) are directly computed as

$$\lambda_k^* = s_k^*, \tag{32}$$

$$\underline{\lambda}_{k-1}^* = -\underline{J}_{k-1,W_k}^{\ddagger T} J_{k,W_k}^T s_k^*, \tag{33}$$

where $\underline{J}_{k-1,W_k}^{\ddagger}$ is the pseudo-inverse matrix of \underline{J}_{k-1,W_k} which fulfills Moore-Penrose conditions except that $\underline{J}_{k-1,W_k} \underline{J}_{k-1,W_k}^{\ddagger}$ is symmetric. Each component of the Lagrange multipliers $\underline{\lambda}_{k-1}^*$ can be obtained recursively in a descending order as follows:

$$\underline{\lambda}_{k-1}^* = \begin{bmatrix} \underline{\lambda}_{k-2}^* \\ \underline{\lambda}_{k-1}^* \end{bmatrix} = \begin{bmatrix} -\underline{J}_{k-2,W_k}^{\ddagger T} (J_{k,W_k}^T s_k^* + J_{k-1,W_k}^T \underline{\lambda}_{k-1}^*) \\ -U_{k-1,W_k} L_{k-1,W_k}^{-1} Y_{k-1,W_k}^T J_{k,W_k}^T s_k^* \end{bmatrix} \tag{34}$$

The obtained Lagrange multipliers are used as an indicator to determine which task to be deactivated depending on whether they are negative or not in the following section.

Finally, the total solution for k prioritized tasks is

$$\begin{aligned} \dot{q}_k^* &= \dot{q}_{k-1}^* + W_k^{-\frac{1}{2}} \dot{q}_{W_k}^* \\ &= \dot{q}_{k-1}^* + W_k^{-\frac{1}{2}} Y_{k,W_k} L_{k,W_k}^{-1} U_{k,W_k}^T (\dot{x}_k - J_k \dot{q}_{k-1}^*). \end{aligned} \tag{35}$$

Additionally, the solution can be computed in a recursive form as follows:

$$\dot{q}_k^* = \underline{J}_{k,W_k}^{\ddagger} \dot{x}_k, \tag{36}$$

where

$$\begin{aligned} \underline{J}_{k,W_k}^{\ddagger} &= \left[(I - W_k^{-\frac{1}{2}} Y_{k,W_k} L_{k,W_k}^{-1} U_{k,W_k}^T J_k) J_{k-1,W_k-1}^{\ddagger} \right. \\ &\quad \left. W_k^{-\frac{1}{2}} Y_{k,W_k} L_{k,W_k}^{-1} U_{k,W_k}^T \right] \\ \dot{x}_k &= [\dot{x}_{k-1}^T, \dot{x}_k^T]^T. \end{aligned} \tag{37}$$

Since the solution minimizing weighted least-squares norm is the generalized solution to cover Euclidean least-squares norm solution, the formulas (36) and (37) can be utilized to compute the solution of the HQP if the weighting matrix is set to the identity matrix.

4 WHQP with inequality tasks

To obtain the optimal solution for p inequality tasks, the WHQP for the k th inequality task is formulated by rewriting

Algorithm 1 WHQP

```

Input:  $\mathcal{W}_0$  : an initial working set
Output:  $\dot{q}^*$  : an optimal solution
1:  $\mathcal{W} = \mathcal{W}_0, iter = 1, \dot{q} = 0$ 
2: while  $iter \leq p - 1$  do
    // Compute the optimal solution in (36)
3:  $\dot{q}^* = WHQP\_equality(\underline{J}_p, \underline{W}_p, \dot{x}_p, \mathcal{W})$ 
    // Compute the step length in (40)
4:  $\alpha, k, r = ComputeStepLength(\underline{J}_p, \dot{x}_p, \dot{q}^*, \dot{q})$ 
5:  $\dot{q} := \dot{q} + \alpha(\dot{q}^* - \dot{q})$ 
    // Update the working set
6: if  $\alpha < 1$  then
7:    $\mathcal{W} \cup (J_k[r], \dot{x}_k[r])$ 
8: continue
9: else
    // Check Lagrange multipliers
10: for  $i = iter$  to  $p$  do
11:    $\lambda_{i-1}^*, s_i^* = ComputeLambda(i)$ 
12:    $\mu, k, r = \min(\lambda_{i-1}^*, s_i^*)$ 
13:   if  $\mu < 0$  then
14:      $\mathcal{W} \setminus (J_k[r], \dot{x}_k[r])$ 
15:     break
16:   end if
17:    $iter = i$ 
18: end for
19: end if
20: end while
    return  $\dot{q}^* := \dot{q}$ 

```

(24) as

$$\begin{aligned}
 \min_{\dot{q}_{W_k}, s_k} \quad & \frac{1}{2} \|s_k\|_2^2, \\
 \text{s. t.} \quad & J_{k, W_k} \dot{q}_{W_k} \leq \dot{x}_k - J_k \dot{q}_{k-1}^* + s_k, \\
 & \underline{J}_{k-1, W_k} \dot{q}_{W_k} \leq \dot{x}_{k-1} - \underline{J}_{k-1} \dot{q}_{k-1}^*.
 \end{aligned} \tag{38}$$

The optimality conditions for (38) are additionally considered in (26) as follows:

$$\begin{aligned}
 \lambda_k^T (J_{k, W_k} \dot{q}_{W_k} - \dot{x}_k + J_k \dot{q}_{k-1}^* - s_k) &= 0, \\
 \underline{\lambda}_{k-1}^T (\underline{J}_{k-1, W_k} \dot{q}_{W_k} - \dot{x}_{k-1} + \underline{J}_{k-1} \dot{q}_{k-1}^*) &= 0, \\
 \lambda_k &\geq 0, \\
 \underline{\lambda}_{k-1} &\geq 0.
 \end{aligned} \tag{39}$$

The conditions of (39) denote complementary conditions that if an inequality task becomes active, that is, it becomes an equality task, its corresponding Lagrange multiplier must be greater than or equal to zero.

Based on the above conditions, the active-set method [16] is adopted. The active-set method iterates a loop until the optimal solution and optimal working set are determined. The working set denotes a set of equality tasks. The optimal solution indicates that it does not activate or violate any other task except the working set. The optimal working set indicates that the corresponding Lagrangian multipliers are non-negative.

The detailed procedure is described in Algorithm 1. At first, the algorithm estimates an initial working set \mathcal{W}_0 for warm-start. Typically, \mathcal{W}_0 contains only equality tasks if there is no initial guess for the inequality tasks. Given the working set \mathcal{W} , *WHQP_equality* computes the optimal solution by using (36) (see Line 3).

Then, *ComputeStepLength* finds a step length α in order for the current solution to step toward the optimal solution without violating the tasks as (see Line 4)

$$\alpha = \min(1, \min_{k,r}(\alpha_{k,r})), \tag{40}$$

where

$$\alpha_{k,r} = \begin{cases} \frac{\dot{x}_k[r] - J_k[r] \dot{q}^{(j)}}{J_k[r](\dot{q}^{*(j)} - \dot{q}^{(j)})} & \text{if } J_k[r] \dot{q}^{(j)} \leq \dot{x}_k[r] \\ 1, & \text{otherwise.} \end{cases} \tag{41}$$

In (40) and (41), k and r denote the indices of the task priority and row of the Jacobian matrix and task vector, respectively. In addition, $\dot{q}^{*(j)}$ and $\dot{q}^{(j)}$ represent the optimal solution and current solution at iteration j , respectively. Calculation of (41) is performed for each row of \underline{J}_p that is not in the working set \mathcal{W} .

Starting at the current solution $\dot{q}^{(j)}$, the solution that is translated toward the optimal solution $\dot{q}^{*(j)}$ with a step length α is calculated as follows (see Line 5):

$$\dot{q}^{(j+1)} = \dot{q}^{(j)} + \alpha(\dot{q}^{*(j)} - \dot{q}^{(j)}) \tag{42}$$

If the step length is less than 1, the corresponding row of the Jacobian matrix, $J_k[r]$ and component of the task vector, $\dot{x}_k[r]$, are added to the working set, and a new iteration begins with the new working set (see Line 6–8).

Once the obtained step length is equal to 1, which indicates that the optimal solution does not activate the remaining inequality tasks, *ComputeLambda* computes the Lagrangian multipliers using (32) and (33) (see Line 11). Next, to ensure that the optimality conditions of (39) are satisfied, it is checked whether or not the minimum Lagrangian multiplier is negative (see Line 12–13). If there exist negative Lagrangian multipliers, the corresponding row of the Jacobian matrix and component of the task vector are removed from the working set, and the next iteration begins (see Line 14–15).

5 Experimental results

The WHQP was validated through various experiments using the differentially-driven mobile base with the 7-DoFs manipulator [13] and the humanoid, *DYROS-JET* [23], with 28-DoFs. The source code is available at <http://github.com/>

gory15/weightedhqp. Additionally, video clips of the experimental results are available in the supplementary material.

5.1 Simulation experiment with nonholonomic mobile manipulator

To maximize the dexterity and mobility of the mobile manipulator, the controller should provide different motion patterns depending on the locomotion, manipulation, and loco-manipulation phases. In particular, because the mobile manipulator comprises two independent systems (mobile base and manipulator), assigning dominant subsystem for each task priority is effective.

5.1.1 Scenario description

To validate this, we designed the following scenario with an 11-DoFs nonholonomic mobile manipulator on simulator, *CoppeliaSim*. The scenario has two phases and each phase is performed for 5 sec. In the first phase, the mobile manipulator reaches the target point above a laptop by tracking the desired trajectory of the end-effector. Then, the robot begins to keep the end-effector focused on the moving target point above the laptop while avoiding an obstacle in the second phase.

5.1.2 Task and weighting matrix description

During the first phase, the robot has a single task of tracking the end-effector position trajectory as

$$J_1 \dot{q} = \dot{x}_1 \tag{43}$$

where $J_1 \in \mathbb{R}^{3 \times 11}$ is the whole-body translational Jacobian of the end-effector [13] and $\dot{x}_1 \in \mathbb{R}^3$ is the desired linear velocity of the end-effector.

In the second phase, the robot executes three prioritized tasks. As the highest priority task, an 11-DoFs joint limit avoidance task [22] is assigned as

$$\begin{aligned} \dot{\underline{x}}_1 &\leq J_1 \dot{q} \leq \dot{\bar{x}}_1, \\ J_1 &= I_{11}, \\ \dot{\underline{x}}_1 &= \epsilon \frac{q - \underline{q}}{\Delta t} \text{ and } \dot{\bar{x}}_1 = \epsilon \frac{\bar{q} - q}{\Delta t}. \end{aligned} \tag{44}$$

$I_{11} \in \mathbb{R}^{11 \times 11}$ denotes an identity matrix and $\dot{\underline{x}}_1$ and $\dot{\bar{x}}_1 \in \mathbb{R}^{11}$ are the lower and upper bound velocity, respectively. \underline{q} and $\bar{q} \in \mathbb{R}^{11}$ are the lower and upper limit of joint position. $q \in \mathbb{R}^{11}$ is the current joint position, ϵ is the tuning parameter and Δt is the control loop period. Then, the obstacle avoidance is formed as an 1-DoF inequality task [7] with the second

priority as

$$\begin{aligned} J_2 \dot{q} &\leq \dot{x}_2, \\ J_2 &= n^T J_{obs}, \\ \dot{x}_2 &= \epsilon \frac{d - d_{thre}}{\Delta t}, \end{aligned} \tag{45}$$

where $n \in \mathbb{R}^3$ is the direction vector from the closest point on the robot to the obstacle and $J_{obs} \in \mathbb{R}^{3 \times 11}$ is the Jacobian matrix of the closest point. d is the distance between the robot and obstacle and d_{thre} denotes the threshold distance. Finally, the lowest priority task is designed as the 3-DoFs gaze task [11] which aligns the target point with the line of sight from the end-effector. The gaze task is defined as

$$\begin{aligned} J_3 \dot{q} &= \dot{x}_3, \\ J_3 &= p_{ee,t} \times (p_{ee,l} \times J_w) - p_{ee,l} \times J_v, \\ \dot{x}_3 &= -\lambda(p_{ee,l} \times p_{ee,t}), \end{aligned} \tag{46}$$

where $p_{ee,l} = p_l - p_{ee}$ and $p_{ee,t} = p_t - p_{ee}$. $p_l \in \mathbb{R}^3$ is the position of arbitrary point on the line of sight, $p_{ee} \in \mathbb{R}^3$ is the position of the end-effector, $p_t \in \mathbb{R}^3$ is the position of the target point, $J_w \in \mathbb{R}^{3 \times 11}$ is the rotational Jacobian matrix, $J_v \in \mathbb{R}^{3 \times 11}$ is the translational Jacobian matrix, and λ is gain constant.

For analyzing the effectiveness of the WHQP, the combinations of the weighting matrices of the three comparative cases are presented in Fig. 2. For Phase 1, to check the contribution of the mobile base to the tracking task depending on the weighting matrix, the weighting matrix is set from the identity matrix (Case 1) to the mobile-dominant matrix (Case 3). In the case of Phase 2, to demonstrate the effectiveness of the individual weighting matrix, the weighting matrix of Case 1 is designed as the manipulator-dominant matrix for all tasks (one weighting matrix for all prioritized tasks). Case 2 assigns the identity matrix for all tasks (equivalent to pseudo-inverse method [6]). Case 3 uses individual weighting matrices for each task priority.

5.1.3 Results

Figure 2 illustrates the snapshots of the simulation results and the ratio of the joint velocity norm between the mobile base, $\dot{q}_b \in \mathbb{R}^4$, and manipulator, $\dot{q}_m \in \mathbb{R}^7$, over time. The ratio is calculated as $\frac{\|\dot{q}_b\|_2^2}{\|\dot{q}\|_2^2}$ and $\frac{\|\dot{q}_m\|_2^2}{\|\dot{q}\|_2^2}$, where $\dot{q} = [\dot{q}_b^T, \dot{q}_m^T]^T$.

In Phase 1 of Case 1, the mobile manipulator began to track the trajectory only by using the manipulator and used both the mobile robot and manipulator together after the arm was stretched around 1.5 sec, which comes into the singular configuration. In practice, this phenomenon occurs frequently due to the difference in dynamics and performance between the mobile base and the manipulator [10,27]. However, since

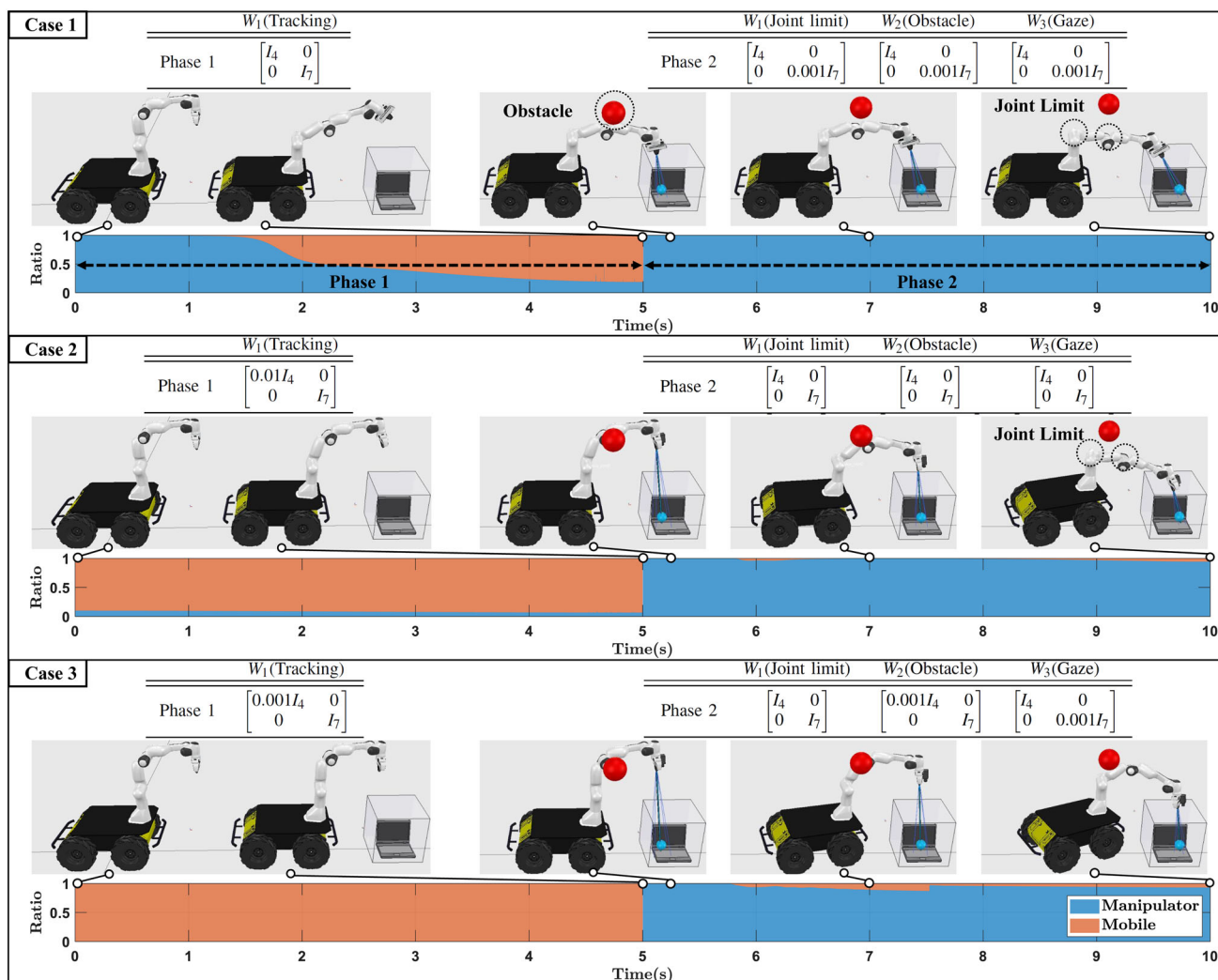


Fig. 2 Experimental results with a nonholonomic mobile manipulator in simulation. A scenario composed of two phases is designed: The first phase is for the robot to approach a laptop. Then, the robot begins to inspect the laptop while avoiding the obstacle during the second phase. Snapshots and motion ratio of the mobile and manipulator are depicted for different combinations of the weighting matrices. For Phase 1, the

resultant postures are different as the weighting matrix changes from the the identity matrix (Case 1) to the mobile-dominant matrix (Case 3). For Phase 2, compared to the same weighting matrix (Case 1) and identity matrix (Case 2), the robot succeeded to execute the task without reaching the posture of singularity or joint limit by assigning individual weighting matrix (Case 3)

the robot tracked the trajectory with the mobile base moving relatively more than the manipulator in Case 2 and only the mobile base moving in Case 3, the robot did not reach the singular posture.

After the end of Phase 1, the robot began to align the line from the end-effector with the blue ball over the laptop with the desired trajectory of the target point while avoiding the obstacle around 6 sec. In Case 1, the resultant posture reached the joint limit and singularity because all the tasks were executed by the manipulator. In Case 2, the configuration of the manipulator reached the joint limit and singularity even though the mobile robot helped to avoid the obstacle. This implies that the weighting matrices given in Case 2 did not

properly distribute the movement between the mobile base and manipulator. In contrast, the gaze task was executed by the manipulator and the obstacle was avoided by the mobile robot in Case 3. Since the whole-body motion was properly distributed by assigning individual weighting matrices for each task priority, the robot could execute the tasks successfully without additional consideration of the necessary constraints.

Therefore, setting individually proper weighting matrices for given tasks is very effective for performing various and complex scenarios for the redundant robot.

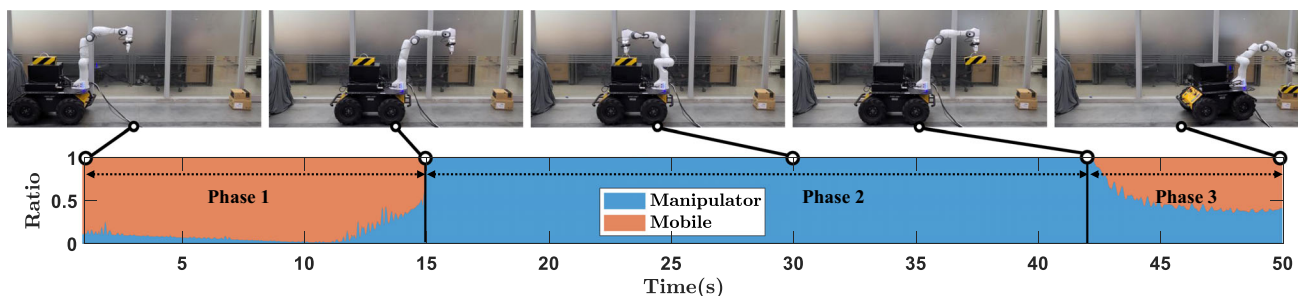


Fig. 3 Experimental results with a nonholonomic mobile manipulator in delivery scenario. The mobile manipulator tracks the end-effector trajectory by using the whole-body, but generates various motion patterns by changing the weighting matrix over time. During Phase 1, the robot

dominantly used the mobile robot to reach the target position. Then, the robot only exploited the manipulator to approach and pick up the box. For the last phase, the robot stacked the box on the other boxes by using the whole-body coordinately

5.2 Real experiment with nonholonomic mobile manipulator

5.2.1 Scenario description

In this section, a typical delivery scenario consisting of locomotion, manipulation, and loco-manipulation phases was designed with a real robot. The scenario has three phases. The first phase is for the robot to execute locomotion task for 15 s. The second phase is to perform manipulation task of picking up a box from 15 to 42 s. In the last phase of loco-manipulation, the robot delivers the box in front of the door from 42 to 50 s.

5.2.2 Task and weighting matrix description

Each phase includes the same task hierarchies: a joint limit avoidance task (the first priority) and tracking task for the trajectory of the end-effector (the second priority). The joint limit avoidance task ($J_1 \in \mathbb{R}^{11 \times 11}$ and $[\dot{x}_1, \bar{x}_1] \in \mathbb{R}^{11}$) is same as (44). The second priority task is to track the desired trajectory of the end-effector as

$$J_2 \dot{q} = \dot{x}_2, \tag{47}$$

$$J_2 = \begin{bmatrix} J_v \\ J_w \end{bmatrix} \text{ and } \dot{x}_3 = \begin{bmatrix} \dot{p}_{ee,d} \\ \delta\Phi \end{bmatrix},$$

where $J_2 \in \mathbb{R}^{6 \times 11}$ is the whole-body Jacobian of the end-effector, $\dot{p}_{ee,d} \in \mathbb{R}^3$ is the desired linear velocity of the end-effector, and $\delta\Phi \in \mathbb{R}^3$ is the orientation error.

To generate various types of motion patterns, different weights are assigned to the tracking task (J_2, \dot{x}_2) for each phase:

$$W_2 = \begin{bmatrix} 0.001I_4 & 0 \\ 0 & I_7 \end{bmatrix} \rightarrow \begin{bmatrix} I_4 & 0 \\ 0 & 0.001I_7 \end{bmatrix} \rightarrow \begin{bmatrix} I_4 & 0 \\ 0 & I_7 \end{bmatrix} \tag{48}$$

In the first phase, the weighting matrix is set such that the trajectory is dominantly tracked by the mobile robot. Then,

the robot mainly exploits the manipulator to track the trajectory in the second phase. In the last phase, the weights are equally distributed to the manipulator and mobile base.

5.2.3 Results

As depicted in Fig. 3, from 0 to 15 s, the robot tracked the desired locomotion trajectory through mobile dominant behavior¹. Then, our WHQP-based controller generated the motions of the manipulator to accurately pick up a box during the manipulation phase. Finally, the robot succeeded in placing a box on the stacked boxes using whole-body motion from 42 to 50 s.

5.3 Real experiment with humanoid

5.3.1 Scenario description

Similar to Sects. 5.1 and 5.2, we validated the performance of the WHQP by implementing the inverse kinematics controller for a humanoid. A box-taping scenario was designed by using the 16-DoFs upper body of the humanoid for 15 s.

5.3.2 Task and weighting matrix description

The humanoid executes two prioritized tasks. The first task is to maintain the initial position of the left hand as

$$J_1 \dot{q} = \dot{x}_1, \tag{49}$$

$$J_1 = [J_{left}, 0_{6 \times 7}, J_{waist}] \text{ and } \dot{x}_1 = \begin{bmatrix} \dot{p}_{left,d} \\ \delta\Phi_{left} \end{bmatrix},$$

¹ Although we set a weighting matrix for the movement of the mobile robot alone during the locomotion phase, the manipulator moved slightly. This is due to the low tracking performance of a mobile robot caused by the difference in control frequency between a mobile robot (10 Hz) and manipulator (1000 Hz).

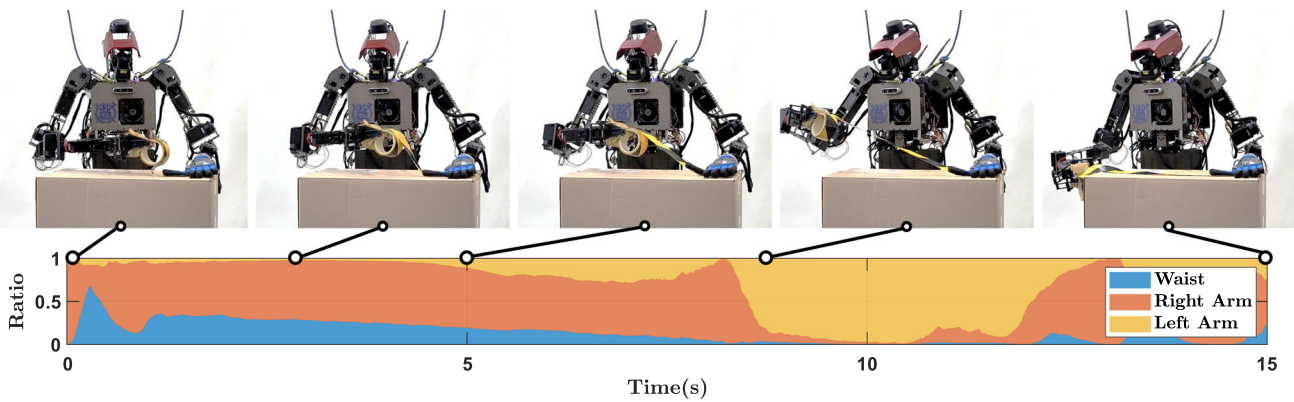


Fig. 4 Experimental results with a humanoid in box-taping scenario. The upper-body of the humanoid executes two prioritized tasks: Initial position of the left hand is maintained as the top priority. The relative motion between the right and left hand is generated as the lower priority.

Since the two tasks share the joints of the waist and left arm, the weighting matrices are designed in order to execute the tasks effectively: The first task only uses the left arm without waist and the second task only exploits the waist and right arm without the left arm

where $J_1 \in \mathbb{R}^{6 \times 16}$ is the Jacobian matrix for the left hand, $J_{left} \in \mathbb{R}^{6 \times 7}$ is the Jacobian matrix from the left shoulder to the left hand, and $J_{waist} \in \mathbb{R}^{6 \times 2}$ is the Jacobian matrix for the waist roll and yaw joint. $\dot{p}_{left,d} \in \mathbb{R}^3$ and $\delta\Phi_{left} \in \mathbb{R}^3$ are the desired linear velocity and orientation error for the left hand, respectively. Then, the relative motion task [18] between the right and left hand is assigned as the second priority:

$$J_2 \dot{q} = \dot{x}_2, \quad J_2 = [-J_{left}, J_{right}, J_{waist}] \text{ and } \dot{x}_2 = \begin{bmatrix} \dot{p}_{rel,d} \\ \delta\Phi_{rel} \end{bmatrix}, \quad (50)$$

where $J_2 \in \mathbb{R}^{6 \times 16}$ is the Jacobian matrix for the relative motion, $J_{right} \in \mathbb{R}^{6 \times 7}$ is the Jacobian matrix from the right shoulder to the right hand. $\dot{p}_{rel,d} \in \mathbb{R}^3$ and $\delta\Phi_{rel} \in \mathbb{R}^3$ are the desired relative linear velocity and relative orientation error, respectively.

Since the joints for executing two prioritized tasks share waist and left arm joints, individual weighting matrices are assigned for each task, as follows:

$$W_1 = \begin{bmatrix} 0.01I_7 & & \\ & I_7 & \\ & & I_2 \end{bmatrix} \text{ and } W_2 = \begin{bmatrix} I_7 & & \\ & 0.01I_7 & \\ & & 0.01I_2 \end{bmatrix} \quad (51)$$

The weighting matrix is set such that the left arm is predominantly used for the first priority task (W_1) and the waist and right arm are predominantly used for the relative motion task of the second priority (W_2).

5.3.3 Results

Figure 4 depicts the snapshots of box-taping scenarios and a motion ratio for the left-arm, right-arm, and waist. Thanks

to the WHQP formulation with weighting matrices, the humanoid robot achieved dexterous manipulation through whole-body behavior. Consequently, the WHQP for a non-holonomic mobile manipulator and a humanoid presented in Sects. 5.1 and 5.2, respectively, can deal with complex real-world scenarios without any additional constraints or planners.

6 Discussions and implementation details

6.1 Computation cost

To validate the computational efficiency of the WHQP, the computation time for the equality tasks was compared with those of the two methods in Sect. 2.1: weighted pseudo-inverse (7) with (8a) and (7) with (8b). The total number of DoFs was set to 30, and the number of task hierarchies p ranged from 1 to 10. Also, the task dimension m_k was uniformly distributed depending on the number of task hierarchies, $\sum_{k=1}^p m_k \approx 30$. We randomly generated Jacobian matrices with full rank and task vectors and performed 500 calculations for each number of task priorities. Finally, an Intel Core i7 with a 16 GB RAM computer was used.

As shown in Fig. 5, the WHQP exhibits a low computation time even when the number of task hierarchies increases. This is because the null-space projection matrix is decomposed into the product of the orthogonal bases. Thus, the size of the inverse matrix is reduced when recursively propagating to compute the solution of a low priority task. In contrast, the other two approaches have a higher computation time than that of the WHQP because the null-space projection matrix of the weighted pseudo-inverse is inherently asym-

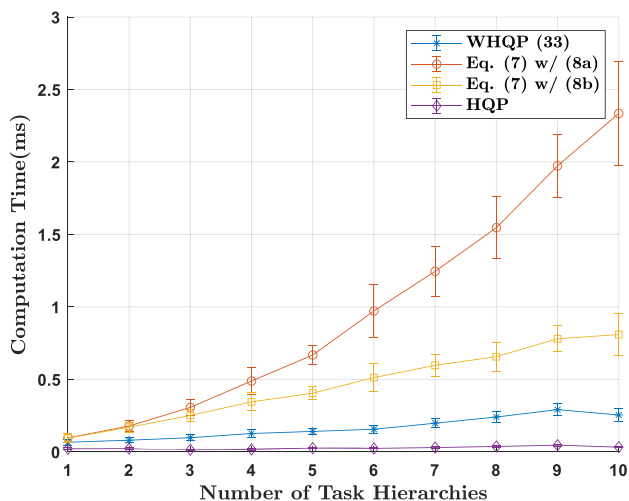


Fig. 5 Computation time for the equality tasks with respect to the number of task hierarchies. Total DoFs were assumed as 30 and task dimension for each task hierarchy was uniformly distributed. The proposed WHQP maintains lower computation time than other weighting pseudo-inverse methods (7) with (8a) and (8b). Additionally, computation time by the HQP is plotted for fair comparison. By assigning weighting matrix, the WHQP has higher computation cost than the HQP

metric. Therefore, the size of the inverse matrix cannot be reduced.

On the other hand, the proposed WHQP has inherently higher computation cost than the HQP as shown in Fig. 5. When computing the total solution for k -th task in (35), the operation of the COD is performed $k(k + 1)/2$ times, whereas the HQP performs k operations of the COD. This is because the null-space matrix Z_{k-1, W_k} is obtained by recursively decomposing from J_{1, W_k} to J_{k-1, W_k} .

6.2 Composite weighting matrix in same hierarchy

Since our formulation handles one weighting matrix at each level, a composite weighting matrix is required if there are two or more tasks at the same task level. In practice, we present a simple construction method using a linear combination [4].

$$W = a_1 W_1 + a_2 W_2 + \dots + a_n W_n, \tag{52}$$

where $\sum_{i=1}^n a_i = 1$ and $0 \leq a_i \leq 1$. The magnitude of a_i indicates which weighting matrix is more dominant at the same level.

6.3 Nullity of WHQP

If the remaining nullity of the WHQP is 0,² the result of the WHQP becomes the same as that of the original HQP solver.

² This implies $\sum_{i=1}^p r_i = n$ when the number of DoFs of the robot is n .

It is evident that a weighted Euclidean norm is equivalent to a normal Euclidean norm when the matrix is not redundant [20]. Therefore, to enhance the effectiveness of the weighting matrices of tasks in the WHQP, it is recommended to configure a set of the tasks where the nullity of the controller exists.

7 Conclusions

Whole-body controllers with hierarchical optimization have great advantages in controlling redundant robots, including mobile manipulators and humanoids. However, the motion generated by these controllers is often undesirable, without additional constraints. In this study, a novel weighted hierarchical quadratic programming method to assign joint weights for each task priority is proposed. The proposed method can be summarized as follows. First, because our algorithm treats a weighting matrix for each task, it can generate various whole-body behaviors depending on the scenarios, in comparison with previous studies. Next, our algorithm using the active-set method can efficiently handle inequality tasks as well as equality tasks. Finally, we demonstrated the effectiveness of the proposed controller through several experiments with a real mobile manipulator and humanoid. With these excellent results, we expect that our method can be applied to other highly redundant systems, such as aerial manipulators. Our future work will involve the extension of the proposed framework for automatically generating suitable weighting matrices for each task.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s11370-022-00431-9>.

Acknowledgements This work has supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A2C3005914) and the Industrial Strategic Technology Development Program (No. 20018745) funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea). Furthermore, this study is a part of the research project “Development of core machinery technologies for autonomous operation and manufacturing”, which has been supported by a grant from National Research Council of Science & Technology under the R&D Program of Ministry of Science, ICT and Future Planning.

References

1. Antonelli G, Chiaverini S (2003) Fuzzy redundancy resolution and motion coordination for underwater vehicle-manipulator systems. *IEEE Trans Fuzzy Syst* 11(1):109–120. <https://doi.org/10.1109/TFUZZ.2002.806321>
2. Chan TF, Dubey RV (1995) A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. *IEEE Trans Robot Autom* 11(2):286–292

3. Choi Y, Oh Y, Oh SR, Park J, Chung WK (2004) Multiple tasks manipulation for a robotic manipulator. *Adv Robot* 18(6):637–653
4. Dariush B, Zhu Y, Arumbakkam A, Fujimura K (2010) Constrained closed loop inverse kinematics. In: IEEE international conference on robotics and automation, IEEE, pp 2499–2506
5. Escande A, Mansard N, Wieber P (2010) Fast resolution of hierarchized inverse kinematics with inequality constraints. In: IEEE international conference on robotics and automation, IEEE, pp 3733–3738
6. Escande A, Mansard N, Wieber PB (2014) Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *Int J Robot Res* 33(7):1006–1028
7. Fang C, Rocchi A, Hoffman EM, Tsagarakis NG, Caldwell DG (2015) Efficient self-collision avoidance based on focus of interest for humanoid robots. In: IEEE-RAS international conference on humanoid robots, IEEE, pp 1060–1066
8. Farello F, Alqasemi R, Dubey R (2010) Optimized dual-trajectory tracking control of a 9-dof wrra system for all tasks. In: 2010 IEEE international conference on robotics and automation, pp 1786–1791, <https://doi.org/10.1109/ROBOT.2010.5509554>
9. Golub G, Van Loan C (1996) Matrix computations. John Hopkins University Press, Baltimore, MD
10. Jia Y, Xi N, Cheng Y, Liang S (2014) Coordinated motion control of a nonholonomic mobile manipulator for accurate motion tracking. In: 2014 IEEE/RSJ international conference on intelligent robots and systems, pp 1635–1640, <https://doi.org/10.1109/IROS.2014.6942774>
11. Kanoun O, Lamiroux F, Wieber PB (2011) Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task. *IEEE Trans Rob* 27(4):785–792. <https://doi.org/10.1109/TRO.2011.2142450>
12. Kim S, Jang K, Park S, Lee Y, Lee SY, Park J (2019) Continuous task transition approach for robot controller based on hierarchical quadratic programming. *IEEE Robot Autom Lett* 4(2):1603–1610. <https://doi.org/10.1109/LRA.2019.2896769>
13. Kim S, Jang K, Park S, Lee Y, Lee SY, Park J (2019b) Whole-body control of non-holonomic mobile manipulator based on hierarchical quadratic programming and continuous task transition. In: IEEE international conference on advanced robotics and mechatronics, IEEE, pp 414–419
14. Lee Y, Ahn J, Lee J, Park J (2021a) Computationally efficient hqp-based whole-body control exploiting the operational-space formulation. In: 2021 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 5197–5202, <https://doi.org/10.1109/IROS51168.2021.9636867>
15. Lee Y, Kim S, Park J, Tsagarakis N, Lee J (2021) A whole-body control framework based on the operational space formulation under inequality constraints via task-oriented optimization. *IEEE Access* 9:39813–39826
16. Nocedal J, Stephen JW (2006) *Numer Optim*. Springer, Berlin
17. Omrčen D, Žlajpah L, Nemeč B (2004) Autonomous motion of a mobile manipulator using a combined torque and velocity control. *Robotica* 22(6):623–632. <https://doi.org/10.1017/S0263574704000293>
18. Park HA, Lee CG (2016) Dual-arm coordinated-motion task specification and performance evaluation. In: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 929–936, <https://doi.org/10.1109/IROS.2016.7759161>
19. Park J, Khatib O (2006) Contact consistent control framework for humanoid robots. In: Proceedings 2006 IEEE international conference on robotics and automation, 2006. ICRA 2006., pp 1963–1969, <https://doi.org/10.1109/ROBOT.2006.1641993>
20. Park J, Chung W, Youm Y (1996) Weighted decomposition of kinematics and dynamics of kinematically redundant manipulators. In: Proceedings of IEEE international conference on robotics and automation, vol 1, pp 480–486
21. Park J, Choi Y, Chung WK, Youm Y (2001) Multiple tasks kinematics using weighted pseudo-inverse for kinematically redundant manipulators. In: IEEE international conference on robotics and automation, vol 4, pp 4041–4047 <https://doi.org/10.1109/ROBOT.2001.933249>
22. Rocchi A, Hoffman EM, Caldwell DG, Tsagarakis NG (2015) Openso: A whole-body control library for the compliant humanoid robot coman. In: IEEE international conference on robotics and automation, pp 6248–6253, <https://doi.org/10.1109/ICRA.2015.7140076>
23. Sim J, Kim S, Park S, Kim S, Kim M, Park J (2021) Design of jet humanoid robot with compliant modular actuators for industrial and service applications. *Appl Sci* 11(13)
24. Tassi F, De Momi E, Ajoudani A (2021) Augmented hierarchical quadratic programming for adaptive compliance robot control. In: 2021 IEEE international conference on robotics and automation (ICRA), pp 3568–3574, <https://doi.org/10.1109/ICRA48506.2021.9561506>
25. Tsuchihiro S, Yamaguchi A, Takamatsu J, Ogasawara T (2015) Using a weighted pseudo-inverse matrix to generate upper body motion for a humanoid robot doing household tasks. In: IEEE international conference on robotics and biomimetics, pp 333–338, <https://doi.org/10.1109/ROBIO.2015.7418789>
26. Wang J, Kim S, Vijayakumar S, Tonneau S (2021) Multi-fidelity receding horizon planning for multi-contact locomotion. In: 20th IEEE-RAS international conference on humanoid robots
27. Wu Y, Lamon E, Zhao F, Kim W, Ajoudani A (2021) Unified approach for hybrid motion control of moca based on weighted whole-body cartesian impedance formulation. *IEEE Robot Autom Lett* 6(2):3505–3512. <https://doi.org/10.1109/LRA.2021.3062316>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.