The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

A Task-priority Based Framework for Multiple Tasks in Highly Redundant Robots

Jae Won Jeong, *Student Member, IEEE* and Pyung Hun Chang, *Member, IEEE*

Abstract— **A task-priority based framework for multiple tasks of highly redundant robots was derived using the Lagrangian multiplier method. The framework was proved to prioritize a generic number of tasks without algorithmic problems – so called an algorithmic singularity and an algorithmic error. The computational efficiency of the framework excels other conventional task-priority strategies. The efficiency and efficacy of the framework was demonstrated theoretically and experimentally through comparative study.**

I. Introduction

It is challenging to control highly redundant robots such as humanoid robots. The difficulty mainly arises from the fact that a highly redundant robot is generally required to execute multiple tasks simultaneously while controlling a large number of degrees of freedom (DOFs). Task descriptions may involve the combination of balance and gaze for a humanoid robot to reach its hand for an object in space. Large number of DOFs of highly redundant robots involves high computational complexity which still remains as a barrier for applying advanced algorithms in spite of recent development on hardware.

Task-priority strategy [1]-[4] can give a rational solution for compromising conflicts between tasks. Suppose that a target is located too far for a humanoid robot to reach its hand keeping balance. As a result, the humanoid cannot accomplish both tasks and there can be large error on both sides. These conflicts can be avoided by performing tasks according to the order of priority. The task with the higher priority is first performed and the task with the lower priority is performed next, utilizing kinematic redundancy.

Conventional task-priority strategies suffer from two algorithmic problems – an algorithmic singularity [5] and an algorithmic error [6], which is also known as the residual error [4]. An algorithmic singularity occurs when the task with lower priority conflicts with the task with higher priority [7]. The task-priority strategy gives ill-conditioned solutions close to an algorithmic singularity region. The singularity-robust (SR) inverse [8], which is also known as damped least-squares inverse [9], has been developed to overcome the difficulties encountered near singularities. Although good conditioning of the solution is ensured, these are obtained at the expense of increased task error. A few methods, which have no algorithmic singularity, have been proposed [7], [10]. These methods optimize the solution for each task first and project it onto the null space of prior tasks next. The main drawback of these methods is that large error, named by algorithmic error, exists in performing tasks with the lower priority even if tasks are compatible. A humanoid robot may not be able to locate hands or legs accurately due to the algorithmic error. A weighted pseudo-inverse method has been proposed for compensating the algorithmic error [4].

A recursive formula is needed to prioritize multiple tasks in conventional task-priority strategies. This formula is generally concomitant with large amount of computational effort as the number of tasks increases. Furthermore, the computational complexity can be more burdensome for highly redundant robots because it is required to compute large dimensional matrices. The heavy charge of computation requires a large expense for powerful hardware of real-time control. A few algorithms have been developed to improve computational efficiency. The incremental method has been proposed to evaluate the null space projector efficiently [11].

A new task-priority framework has been proposed for a generic number of tasks in this paper. The proposed framework is derived through minimization of task-error subject to the prior tasks execution by using the Lagrangian multiplier method. The proposed framework simultaneously has three merits as follows: 1) The proposed framework is robust for algorithmic singularities, 2) while it does not have an algorithmic error. 3) The computational efficiency of the proposed framework excels other conventional task-priority strategies.

The paper is structured as follows: In section II, the proposed method will be derived and analyzed. In section III, the comparison will be covered. In section IV, the proposed method will be evaluated by simulations. Finally, the conclusion will be described in section V.

II. Optimal Framework

A. Problem Statement

It is assumed that there are k tasks to be performed and each task requires m_i DOFs. The kinematic equation for each task is given as

$$\mathbf{x}_i = \mathbf{f}_i(\boldsymbol{\theta}) \quad \text{for } 1 \le i \le k, \tag{1}$$

where $\mathbf{x}_i \in \Re^{m_i}$ denotes the i th task vector with respect to the base frame, $\boldsymbol{\theta} \in \Re^n$ joint vector, and $\mathbf{f}_i \in \Re^{m_i}$ a vector

Manuscript received March 1, 2009.

Jae Won Jeong is with Korea Advanced Institute of Science and Technology, 373-1, Guseong-dong, Yuseong-gu, Deajeon, Republic of Korea (e-mail: mechjjw@mecha.kaist.ac.kr).

Pyung Hun Chang is with Korea Advanced Institute of Science and Technology, 373-1, Guseong-dong, Yuseong-gu, Deajeon, Republic of Korea (e-mail: phchang@mecha.kaist.ac.kr).

consisting of m_i scalar equations. It is assumed that the i th task has lower priority with respect to the previous $i-1$ th task.

From the kinematic equations, the second-order differential kinematic equations are determined as

$$\ddot{\mathbf{x}}_i = \mathbf{J}_i(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \dot{\mathbf{J}}_i(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \quad \text{for } 1 \le i \le k, \tag{2}$$

where $(\dot{\ })$ denotes the first order time derivative, $(\ddot{\ })$ the second order time derivative and $\mathbf{J}_i = (\partial \mathbf{f}_i)/(\partial \boldsymbol{\theta}) \in \Re^{m_i \times n}$ Jacobian of i th task. Equation (2) can be rewritten as

$$\mathbf{J}_i(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} = \mathbf{h}_i \quad \text{for } 1 \le i \le k, \tag{3}$$

where $\mathbf{h}_i \in \Re^{m_i}$ is defined as

$$\mathbf{h}_i \triangleq \ddot{\mathbf{x}}_i - \dot{\mathbf{J}}_i(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \quad \text{for } 1 \le i \le k. \tag{4}$$

The problem is to solve joint acceleration, $\ddot{\boldsymbol{\theta}} \in \Re^n$, for given tasks, $\mathbf{h}_i \in \Re^{m_i}$, according to the order of priority.

B. Optimal Solution

Let $H_i(\ddot{\boldsymbol{\theta}})$ be the criteria function for minimization of the i th task error as

$$H_i(\ddot{\boldsymbol{\theta}}) = \frac{1}{2}(\mathbf{J}_i\ddot{\boldsymbol{\theta}} - \mathbf{h}_i)^T(\mathbf{J}_i\ddot{\boldsymbol{\theta}} - \mathbf{h}_i). \tag{5}$$

Also, let $\mathbf{F}_i(\ddot{\boldsymbol{\theta}}) \in \Re^{m_i}$ be the constraint function as

$$\begin{aligned}\mathbf{F}_i(\ddot{\boldsymbol{\theta}}) &= \mathbf{J}^{\mathbf{A}}_{i-1}\ddot{\boldsymbol{\theta}} - \mathbf{h}^{\mathbf{A}}_{i-1} \\ &= \mathbf{0}\end{aligned}, \tag{6}$$

where the augmented Jacobian, $\mathbf{J}^{\mathbf{A}}_{i-1}$, and the augmented task vector, $\mathbf{h}^{\mathbf{A}}_{i-1}$, are defined as

$$\mathbf{J}^{\mathbf{A}}_{i-1} \triangleq \begin{bmatrix}\mathbf{J}_1^T & \mathbf{J}_2^T & \cdots & \mathbf{J}_{i-1}^T\end{bmatrix}^T, \text{ and} \tag{7}$$

$$\mathbf{h}^{\mathbf{A}}_{i-1} \triangleq \begin{bmatrix}\mathbf{h}_1^T & \mathbf{h}_2^T & \cdots & \mathbf{h}_{i-1}^T\end{bmatrix}^T. \tag{8}$$

Let us define the Lagrangian function as the following:

$$L_i(\ddot{\boldsymbol{\theta}}) = H_i(\ddot{\boldsymbol{\theta}}) + \boldsymbol{\lambda}_i^T\mathbf{F}_i(\ddot{\boldsymbol{\theta}}), \tag{9}$$

where $\boldsymbol{\lambda}_i$ denotes an m_i-dimensional Lagrangian multiplier vector for the i th task. At the stationary points of L_i,

$$\begin{aligned}\frac{\partial}{\partial\ddot{\boldsymbol{\theta}}}L_i(\ddot{\boldsymbol{\theta}}) &= \frac{\partial}{\partial\ddot{\boldsymbol{\theta}}}H_i(\ddot{\boldsymbol{\theta}}) + \frac{\partial}{\partial\ddot{\boldsymbol{\theta}}}\boldsymbol{\lambda}_i^T\mathbf{F}_i(\ddot{\boldsymbol{\theta}}) \\ &= \mathbf{0}\end{aligned}. \tag{10}$$

From (5), (6), and (10), we can get the following equation

$$\mathbf{J}_i^T\mathbf{J}_i\ddot{\boldsymbol{\theta}} = \mathbf{J}_i^T\mathbf{h}_i - \mathbf{J}^{\mathbf{A}\,T}_{i-1}\boldsymbol{\lambda}_i \tag{11}$$

It is assumed that there are r_i feasible directions (i.e. dimensions of joint acceleration space which can not affect prior tasks) for the i th task as

$$r_i = \rho(\mathbf{J}^{\mathbf{A}}_i) - \rho(\mathbf{J}^{\mathbf{A}}_{i-1}), \tag{12}$$

where $\rho(\bullet)$ denotes the rank of $\bullet$.

Now, define $\mathbf{Z}_i \in \Re^{r_i \times n}$ as a matrix consisting of the r_i basis vectors spanning the null space of prior tasks. Then, $\mathbf{Z}_i$ satisfies the following relationship

$$\mathbf{Z}_i\mathbf{J}^{\mathbf{A}\,T}_{i-1} = \mathbf{0}. \tag{13}$$

Note that $\mathbf{Z}_i$ can be evaluated by using either singular value decomposition (SVD) [13] or the method in [14].

Multiplying $\mathbf{Z}_i$ on both sides of (11) leads to

$$\mathbf{Z}_i\mathbf{J}_i^T\mathbf{J}_i\ddot{\boldsymbol{\theta}} = \mathbf{Z}_i\mathbf{J}_i^T\mathbf{h}_i. \tag{14}$$

By using (14), let us define the extended Jacobian as

$$\mathbf{J}_{\mathbf{E}} \triangleq \begin{bmatrix}\mathbf{J}_1 \\ \mathbf{Z}_2\mathbf{J}_2^T\mathbf{J}_2 \\ \vdots \\ \mathbf{Z}_k\mathbf{J}_k^T\mathbf{J}_k\end{bmatrix}, \tag{15}$$

and the extended task vector as

$$\mathbf{h}_{\mathbf{E}} \triangleq \begin{bmatrix}\mathbf{h}_1 \\ \mathbf{Z}_2\mathbf{J}_2^T\mathbf{h}_2 \\ \vdots \\ \mathbf{Z}_k\mathbf{J}_k^T\mathbf{h}_k\end{bmatrix}. \tag{16}$$

Now, the task-priority based kinematics equation is determined as

$$\mathbf{J}_{\mathbf{E}}\ddot{\boldsymbol{\theta}} = \mathbf{h}_{\mathbf{E}}. \tag{17}$$

The minimum norm solution for (17) is obtained as

$$\ddot{\boldsymbol{\theta}} = \mathbf{J}_{\mathbf{E}}^+\mathbf{h}_{\mathbf{E}}, \tag{18}$$

where $(\bullet)^+$ denotes the Moore-Penrose pseudo-inverse of $\bullet$.

C. Important Properties

Let $\mathbf{J}_{i\_feasible} \in \Re^{r_i \times n}$ be a matrix consisting feasible rows of $\mathbf{J}_i$ as

$$\rho\left(\begin{bmatrix}\mathbf{J}^{\mathbf{A}}_{i-1} \\ \mathbf{J}_{i\_feasible}\end{bmatrix}\right) = \rho\left(\begin{bmatrix}\mathbf{J}^{\mathbf{A}}_{i-1} \\ \mathbf{J}_i\end{bmatrix}\right), \tag{19}$$

and let $\mathbf{J}_{i\_conflict} \in \Re^{(m_i - r_i) \times n}$ be a matrix consisting unfeasible rows of $\mathbf{J}_i$ as

$$\rho\left(\begin{bmatrix}\mathbf{J}^{\mathbf{A}}_{i-1} \\ \mathbf{J}_{i\_conflict}\end{bmatrix}\right) = \rho(\mathbf{J}^{\mathbf{A}}_{i-1}). \tag{20}$$

Without loss of generality, (3) can be rewritten as

$$\begin{bmatrix}\mathbf{J}_{i\_conflict} \\ \mathbf{J}_{i\_feasible}\end{bmatrix}\ddot{\boldsymbol{\theta}} = \begin{bmatrix}\mathbf{h}_{i\_conflict} \\ \mathbf{h}_{i\_feasible}\end{bmatrix}, \tag{21}$$

where $\mathbf{h}_{i\_conflict} \in \Re^{(m_i - r_i) \times n}$ and $\mathbf{h}_{i\_feasible} \in \Re^{r_i \times n}$ denote the corresponding task vectors of $\mathbf{h}_i$. If the i th task is compatible with prior tasks, then $\mathbf{J}_{i\_feasible} = \mathbf{J}_i$, and $\mathbf{h}_{i\_feasible} = \mathbf{h}_i$.

Theorem 1. *(Algorithmic Singularity)*

If the Jacobians, $\mathbf{J}_i$, *have full rank (i.e. there is no kinematic singularities), then there is no algorithmic singularities in* (18).

Proof From (21), the left-hand side of (14) is derived as

$$\mathbf{Z}_i\mathbf{J}_i^T\mathbf{J}_i\ddot{\boldsymbol{\theta}} = \mathbf{Z}_i\begin{bmatrix}\mathbf{J}_{i\_conflict}\\ \mathbf{J}_{i\_feasible}\end{bmatrix}^T\begin{bmatrix}\mathbf{J}_{i\_conflict}\\ \mathbf{J}_{i\_feasible}\end{bmatrix}\ddot{\boldsymbol{\theta}}$$
$$= \begin{bmatrix}\mathbf{0}_{r_i\times(m_i-r_i)} & (\mathbf{Z}_i\mathbf{J}_{i\_feasible}^T)_{r_i\times r_i}\end{bmatrix}\begin{bmatrix}\mathbf{J}_{i\_conflict}\\ \mathbf{J}_{i\_feasible}\end{bmatrix}\ddot{\boldsymbol{\theta}}, \quad (22)$$
$$= (\mathbf{Z}_i\mathbf{J}_{i\_feasible}^T)\mathbf{J}_{i\_feasible}\ddot{\boldsymbol{\theta}}$$

and the right-hand side of (14) is derived as

$$\mathbf{Z}_i\mathbf{J}_i^T\mathbf{h}_i = \mathbf{Z}_i\begin{bmatrix}\mathbf{J}_{i\_conflict}\\ \mathbf{J}_{i\_feasible}\end{bmatrix}^T\begin{bmatrix}\mathbf{h}_{i\_conflict}\\ \mathbf{h}_{i\_feasible}\end{bmatrix}$$
$$= \begin{bmatrix}\mathbf{0}_{r_i\times(m_i-r_i)} & (\mathbf{Z}_i\mathbf{J}_{i\_feasible}^T)_{r_i\times r_i}\end{bmatrix}\begin{bmatrix}\mathbf{h}_{i\_conflict}\\ \mathbf{h}_{i\_feasible}\end{bmatrix}. \quad (23)$$
$$= (\mathbf{Z}_i\mathbf{J}_{i\_feasible}^T)\mathbf{h}_{i\_feasible}$$

From (22) and (23), (17) is equivalent to

$$\begin{bmatrix}\mathbf{J}_{1m_1\times n}\\ \mathbf{Z}_2\mathbf{J}_{2\_feasible}^T\mathbf{J}_{2\_feasible}\\ \vdots\\ \mathbf{Z}_k\mathbf{J}_{k\_feasible}^T\mathbf{J}_{k\_feasible}\end{bmatrix}\ddot{\boldsymbol{\theta}} = \begin{bmatrix}\mathbf{J}_1\\ \mathbf{Z}_2\mathbf{J}_{2\_feasible}^T\mathbf{h}_{2\_feasible}\\ \vdots\\ \mathbf{Z}_k\mathbf{J}_{k\_feasible}^T\mathbf{h}_{k\_feasible}\end{bmatrix}. \quad (24)$$

Since $(\mathbf{Z}_i\mathbf{J}_{i\_feasible}^T)_{r_i\times r_i}$ is invertible, (24) is equivalent to

$$\begin{bmatrix}(\mathbf{J}_1)_{m_1\times n}\\ (\mathbf{J}_{2\_feasible})_{r_2\times n}\\ \vdots\\ (\mathbf{J}_{k\_feasible})_{r_k\times n}\end{bmatrix}\ddot{\boldsymbol{\theta}} = \begin{bmatrix}(\mathbf{h}_1)_{m_1\times n}\\ (\mathbf{h}_{2\_feasible})_{r_2\times n}\\ \vdots\\ (\mathbf{h}_{k\_feasible})_{r_k\times n}\end{bmatrix}. \quad (25)$$

Therefore, the optimal solution, (18), shares the same solution as

$$\ddot{\boldsymbol{\theta}} = \begin{bmatrix}(\mathbf{J}_1)_{m_1\times n}\\ (\mathbf{J}_{2\_feasible})_{r_2\times n}\\ \vdots\\ (\mathbf{J}_{k\_feasible})_{r_k\times n}\end{bmatrix}^+\begin{bmatrix}(\mathbf{h}_1)_{m_1\times n}\\ (\mathbf{h}_{2\_feasible})_{r_2\times n}\\ \vdots\\ (\mathbf{h}_{k\_feasible})_{r_k\times n}\end{bmatrix}. \quad (26)$$

Since the Jacobian, $\begin{bmatrix}\mathbf{J}_1^T & \mathbf{J}_{2\_feasible}^T & \cdots & \mathbf{J}_{k\_feasible}^T\end{bmatrix}^T$, has full rank, there is no algorithmic singularities in(18). □

Remark. It may be necessary to introduce a damping factor, λ, for ensuring good conditioning near algorithmic singularities since the numerical error can be magnified by the condition number of $\mathbf{J}_\mathbf{E}$. Introducing a damping factor, the solution in the proposed framework can be modified as follows:

$$\ddot{\boldsymbol{\theta}} = \begin{bmatrix}\mathbf{J}_1\\ \mathbf{Z}_2(\mathbf{J}_2^T\mathbf{J}_2+\lambda\mathbf{I})\\ \vdots\\ \mathbf{Z}_k(\mathbf{J}_k^T\mathbf{J}_k+\lambda\mathbf{I})\end{bmatrix}^+\begin{bmatrix}\mathbf{h}_1\\ \mathbf{Z}_2\mathbf{J}_2^T\mathbf{h}_2\\ \vdots\\ \mathbf{Z}_k\mathbf{J}_k^T\mathbf{h}_k\end{bmatrix}. \quad (27)$$

Similar to the SR inverse, the damping factors need to be tuned for minimizing degradation of the task accuracy.

Lemma 1. *The optimal solution,* (18)*, can be obtained as follows:*

$$\ddot{\boldsymbol{\theta}} = \sum_{i=1}^{k}\hat{\mathbf{J}}_{i\_feasible}^+(\mathbf{h}_{i\_feasible} - \mathbf{J}_{i\_feasible}\ddot{\boldsymbol{\theta}}_{i\_prev}),$$
$$\text{where}\begin{cases}\hat{\mathbf{J}}_{i\_feasible} \triangleq \mathbf{J}_{i\_feasible}\mathbf{N}_i,\\ \mathbf{N}_i = \mathbf{I} - \sum_{j=1}^{i-1}\hat{\mathbf{J}}_{j\_feasible}^+\hat{\mathbf{J}}_{j\_feasible},\ \mathbf{N}_1 = \mathbf{I},\ \text{and}\\ \ddot{\boldsymbol{\theta}}_{i\_prev} = \sum_{j=1}^{i-1}\ddot{\boldsymbol{\theta}}_j\end{cases}. \quad (28)$$

In (28), $\mathbf{N}_i$ denotes the null space projector for i th task

Proof Michael Mistry, *et al.*proved that (26) is equivalent to (28) in [15]. □

The solution must satisfy $\mathbf{J}_{i\_feasible}\ddot{\boldsymbol{\theta}} = \mathbf{h}_{i\_feasible}$, which means that the i th subtask is executed as much as possible within the feasible subspace.

Definition 1. *Let us define an algorithmic error as*

$$\mathbf{e}_{i\_algorithm} = \mathbf{h}_{i\_feasible} - \mathbf{J}_{i\_feasible}\ddot{\boldsymbol{\theta}}. \quad (29)$$

Theorem 2. *(Algorithmic error)*
If (28) holds, then there is no algorithmic error in (18)*.*

Proof By substituting (28) into (29), an algorithmic error is derived as

$$\mathbf{e}_{i\_algorithm} = \mathbf{h}_{i\_feasible} - \mathbf{J}_{i\_feasible}\ddot{\boldsymbol{\theta}}$$
$$= (\mathbf{I} - \mathbf{J}_{i\_feasible}\hat{\mathbf{J}}_{i\_feasible}^+)\mathbf{h}_{i\_feasible} \quad . (30)$$
$$-(\mathbf{J}_{i\_feasible} - \mathbf{J}_{i\_feasible}\hat{\mathbf{J}}_{i\_feasible}^+\mathbf{J}_{i\_feasible})\ddot{\boldsymbol{\theta}}_{i\_prev}$$

Since $\mathbf{N}_i$ is idempotent, it is easy to show

$$\mathbf{J}_{i\_feasible}\hat{\mathbf{J}}_{i\_feasible}^+ = \mathbf{I}. \quad (31)$$

Substituting (31) into (30) leads to $\mathbf{e}_{i\_algorithm} = \mathbf{0}$. Therefore, the proposed optimal framework does not have an algorithmic error. □

III. Simulation

The proposed optimal framework was validated and compared with other task-priority strategies, which can be applied for a generic number of tasks, are the resolved acceleration method [1], [3] and two resolved torque method — the operational space formulation [12] and the unifying framework [10].

A. Settings

The proposed method was implemented and verified in SimStudio environment, which is accurate dynamics simulation software. A humanoid robot which has 28 DOFs was used for a simulated experiment.

The humanoid robot stabbed a dummy with a fencing sword in this simulation. The required tasks are described as follows:

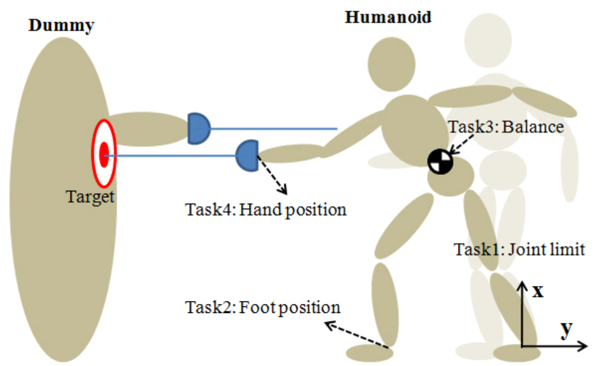

Fig. 1. Task description: A humanoid robot is required to execute four tasks simultaneously.

1) Joint limit

Joint limit task is defined as keeping each joint angle within its range of motion.

2) Self-balance

The self-balance task is defined by the global center of gravity coordinates, $\mathbf{x}_{cog}$, and the associated Jacobian, $\mathbf{J}_{cog}$, which can be expressed as,

$$\mathbf{x}_{cog} = \frac{1}{M}\sum_{i=1}^{n} m(i)\mathbf{x}_{com}(i), \tag{32}$$

$$\mathbf{J}_{cog} = \frac{1}{M}\sum_{i=1}^{n} m(i)\mathbf{J}_{com}(i), \tag{33}$$

where $m(i)$ denotes the mass of the i th link, $\mathbf{x}_{com}(i)$ the center of mass of the i th link, $\mathbf{J}_{com}(i)$ corresponding Jacobian, and M the total mass of the humanoid robot. The global center of gravity is controlled at the center of feet.

3) Left foot position

The humanoid robot stepped one pace forward.

4) Left hand position

The left hand position task is defined as stabbing the center of the target on a dummy with fencing sword.

To validate the proposed method, two simulations have been performed. The objectives of each simulation are as follows:

- One was performed to compare the subtask accuracy when given tasks were compatible.
- The other was performed to compare the algorithmic singularity robustness when given tasks were not compatible.

In the latter case, the target is located too far to reach the fencing sword while keeping joint-limit and self-balance.

B. *Results and Discussion*

1) Subtask accuracy

A sequence of snapshots is shown in Fig. 5. The position error of the sword's tip is described in Fig. 2. The simulation results show that the subtask was executed accurately using the proposed framework. As a result, the humanoid robot accurately stabbed the center of the target (see Fig. 3.).

The subtask was executed accurately using the resolved acceleration method and the operational space formulation. However, the unifying framework failed to execute the subtask accurately due to the algorithmic error. Based on the

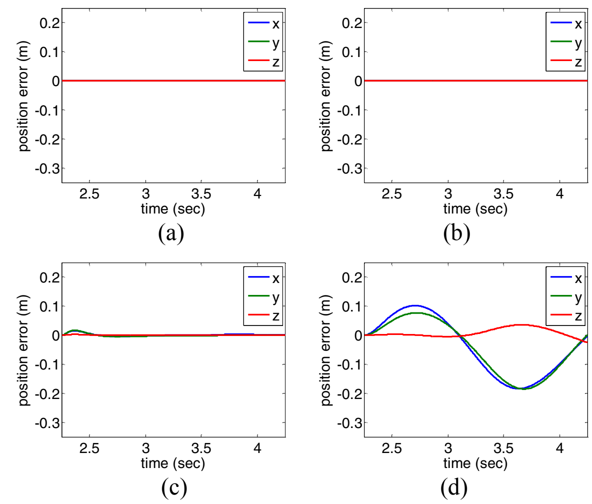

Fig. 2. Position error of the sword's tip when tasks are compatible: (a) the proposed framework, (b) the resolved acceleration method, (c) the operational space formulation, (d) the unifying framework

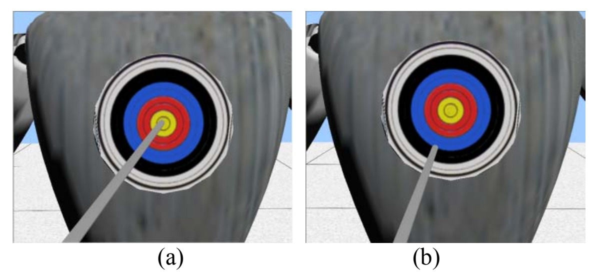

Fig. 3. Comparison of an algorithmic error: (a) the proposed optimal framework, (b) the unifying framework

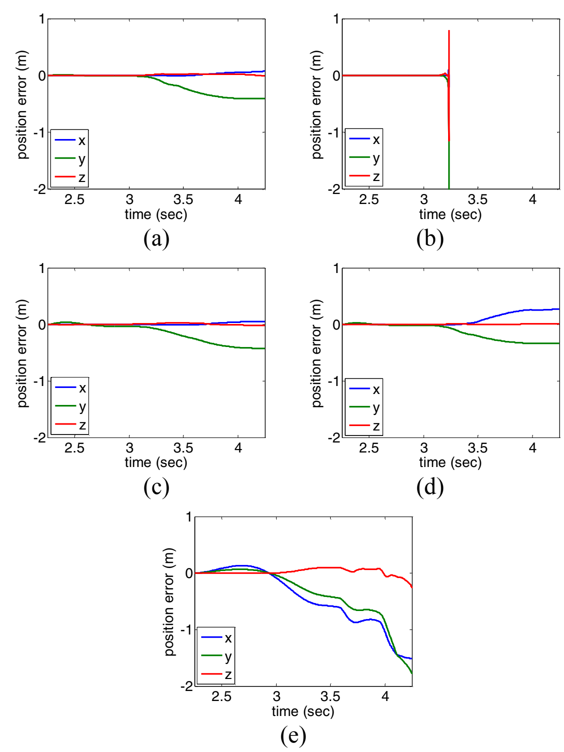

Fig. 4. Position error of the sword's tip when tasks are not compatible: (a) the proposed framework, (b) the resolved acceleration method , (c) the resolved acceleration method (with SR inverse), (d) the operational space formulation, (e) the unifying framework

reformulation (see Appendix), the algorithmic error of the unifying framework can be derived as follows:

$$\begin{aligned}\mathbf{e}_{i\_algorithm} &= \mathbf{h}_{i\_feasible} - \mathbf{J}_{i\_feasible}\ddot{\boldsymbol{\theta}}_{i\_unifying\_framework} \\ &= \mathbf{h}_{i\_feasible} \\ &\quad - \mathbf{J}_{i\_feasible}(\ddot{\boldsymbol{\theta}}_{i\_prev} + \mathbf{N}_i(\mathbf{J}_i^{\mathbf{M}^2+}\mathbf{h}_i)) \\ &\neq \mathbf{0}\end{aligned} \quad . \quad (34)$$

Therefore, large tracking error occurred and the humanoid robot could not stab the center of the target (see Fig. 3.).

2) Algorithmic singularity robustness

The target is located too far to reach the fencing sword while keeping joint-limit and self-balance. A sequence of snapshots is shown in Fig. 5. The position error of the sword's tip is described in Fig. 4. The simulation results show that the given tasks were performed according to priority without algorithmic singularities using the proposed framework. The humanoid robot could not reach its hand more because tasks with the higher priority, the joint-limit task and the self-balance task, must be performed first.

Algorithmic singularities are defined as the configurations at which $\hat{\mathbf{J}}_i = \mathbf{J}_i\mathbf{N}_i$, the Jacobian projected on the null space of prior tasks, loses its rank. An algorithmic singularity occurs whenever tasks conflict as

$$\rho(\mathbf{J}_i^{\mathbf{A}}) < \rho(\mathbf{J}_{i-1}^{\mathbf{A}}) + \rho(\mathbf{J}_i)\,. \quad (35)$$

The resolved acceleration method without using the SR inverse showed unstable responses due to algorithmic singularities. With the SR inverse, the resolved acceleration method showed good tracking performance. The unifying framework does not have algorithmic singularities because $\hat{\mathbf{J}}_i$ is not used. However, the unifying framework had large tracking error because the tip of the sword converged to non-optimal position after conflict occurred.

3) Computational efficiency

Using the proposed optimal framework, we were able to perform the computation of the control torque in less than 1.4 msec using a PC with a 2.4 GHz Pentium CPU. Compared with other methods (the resolved acceleration method: 2.7 msec, the unifying framework: 3.2 msec, and the operational space formulation: 4.8 msec), the proposed optimal framework showed the best computational efficiency.

There are two reasons why the proposed optimal framework is more efficient than other methods. First, the null space projector in the proposed optimal framework can

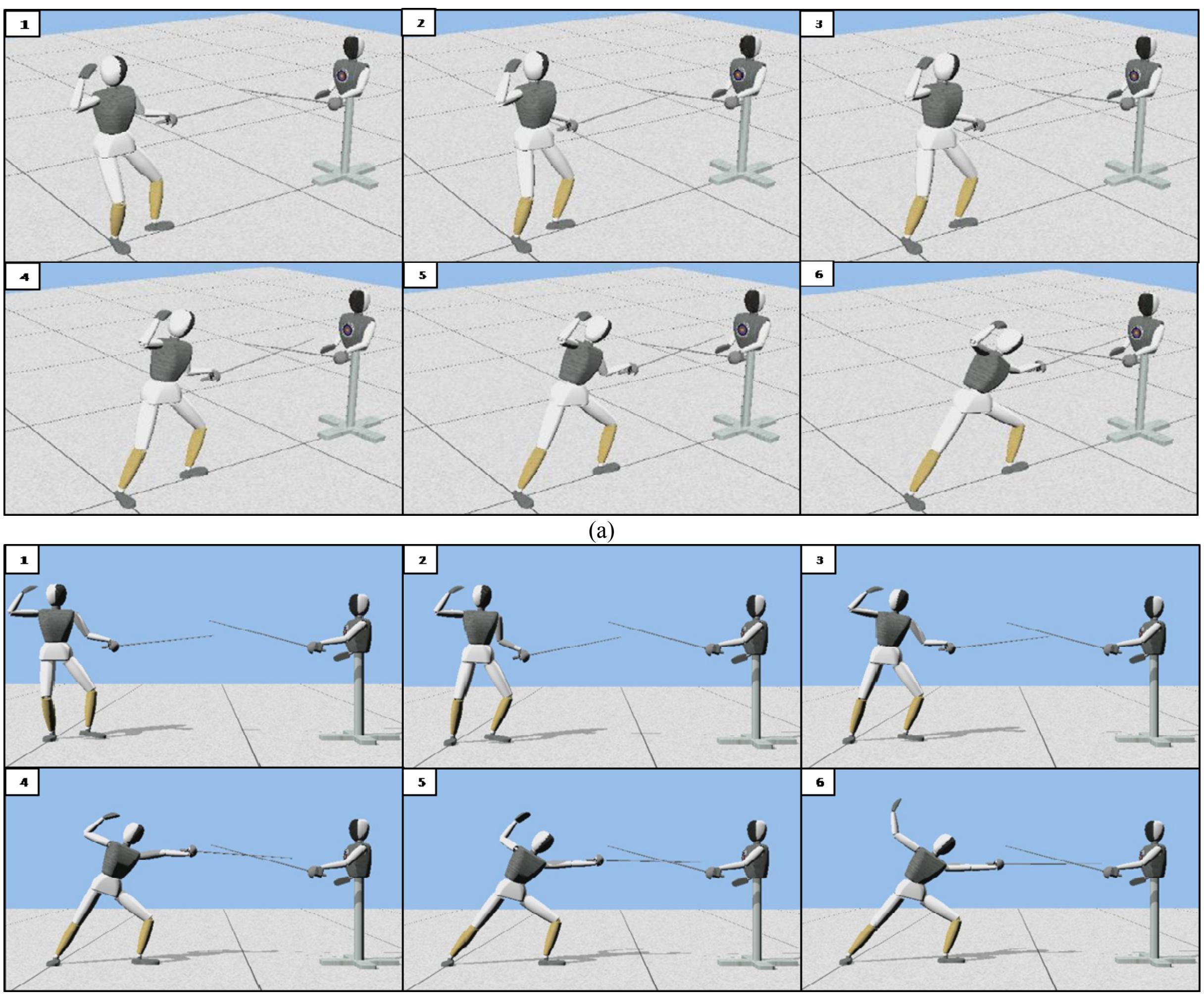

Fig. 5. Snapshots of the proposed optimal framework: (a) when tasks are compatible, (b) when tasks are not compatible.

be evaluated more efficiently than other methods. The null space matrices, $\mathbf{Z}_i$, can be updated efficiently by using the method in [23] with $O(n^2)$ the computational complexity. The computational costs for the null space projectors, $\mathbf{N}_i$, in other methods reaches $O(kn^2)$. Second, the control torque can be computed more efficiently than the resolved torque methods using Newton-Euler rigid body dynamics formulations with $O(n)$ computational complexity. The resolved torque methods requires explicit computation of $\mathbf{M}$ and $\mathbf{M}^{-1}$ with computational costs at least reaching $O(n^2)$ and $O(n^3)$, respectively.

IV. Conclusion

A new task-priority based framework for a generic number of tasks is described in the paper. The proposed framework is derived through minimization of task-error subject to the prior tasks execution by using the Lagrangian multiplier method. Based on the proposed framework, an inverse kinematics solution in acceleration level can be obtained without an algorithmic singularity and an algorithmic error. The computational efficiency of the proposed framework is better than other methods. This is useful for real-time control of highly redundant robots with many task-priority levels. The proposed framework has been verified through the simulation applied to a humanoid robot which has 28 DOFs.

Appendix

A. Reformulation of the unifying framework

The joint space dynamics equation is given as

$$\boldsymbol{\tau} = \mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) + \mathbf{G}(\boldsymbol{\theta}), \quad (36)$$

where $\boldsymbol{\tau} \in \Re^n$ denotes joint torque, $\mathbf{M}(\boldsymbol{\theta}) \in \Re^{n\times n}$ inertia tensor, $\mathbf{C}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) \in \Re^n$ coriolis and centrifugal torque, and $\mathbf{G}(\boldsymbol{\theta}) \in \Re^n$ gravity torque.

The unifying framework was proposed as follows:

$$\boldsymbol{\tau} = \sum_{i=1}^{k} \boldsymbol{\tau}_i + \mathbf{C} + \mathbf{G},$$

$$\text{where } \begin{cases} \boldsymbol{\tau}_i = (\mathbf{J}_i\mathbf{M}^{-1})^{+}[\mathbf{h}_i - \mathbf{J}_i\mathbf{M}^{-1}(\boldsymbol{\tau}_{i\_post} - (\mathbf{C}+\mathbf{G}))] \\ \boldsymbol{\tau}_{i\_post} = \sum_{j=i+1}^{k} \boldsymbol{\tau}_i \end{cases}. \quad (37)$$

Let us define $\ddot{\boldsymbol{\theta}}_{i\_post}$ as

$$\ddot{\boldsymbol{\theta}}_{i\_post} = \mathbf{M}^{-1}[\boldsymbol{\tau}_{i\_post} - (\mathbf{C}+\mathbf{G})]. \quad (38)$$

Substituting (38) into (37), $\boldsymbol{\tau}_i$ is obtained as follows:

$$\begin{aligned} \boldsymbol{\tau}_i &= (\mathbf{J}_i\mathbf{M}^{-1})^{+}(\mathbf{h}_i - \mathbf{J}_i\ddot{\boldsymbol{\theta}}_{i\_post}) \\ &= \mathbf{M}\mathbf{J}_i^{\mathbf{M}^2+}(\mathbf{h}_i - \mathbf{J}_i\ddot{\boldsymbol{\theta}}_{i\_post}) \end{aligned}. \quad (39)$$

The unifying framework can be reformulated as

$$\boldsymbol{\tau} = \mathbf{M}\sum_{i=1}^{k} \ddot{\boldsymbol{\theta}}_{i\_unifying\_framework} + \mathbf{C} + \mathbf{G}, \quad (40)$$

$$\text{where } \begin{cases} \ddot{\boldsymbol{\theta}}_{i\_unifying\_framework} = \mathbf{N}_i(\mathbf{J}_i^{\mathbf{M}^2+}\mathbf{h}_i), and \\ \mathbf{N}_i = \mathbf{N}_{i-1}(\mathbf{I} - \mathbf{J}_{i-1}^{\mathbf{M}^2+}\mathbf{J}_{i-1}) \end{cases}.$$

References

[1] Y. Nakamura, H. Hanafusa and T. Yoshikawa, “Task-priority based redundancy control of robot manipulators,” Int. J. Robotics Res., vol. 6, no. 2, pp. 3–15, 1987.

[2] A. A. Maciejewski and C. A. Klein, “Obstable avoidance for kinematically redundant manipulators in dynamically varying environments,” Int. J. Robotics Res., vol. 4, no. 3, pp. 109–117, 1985.

[3] B. Siciliano and J.-J. E. Slotine, “A general framework for managing multiple tasks in highly redundnat robotic systems,” Proc. Int. Conf. on Advanced Robotics, Pisa, p.323, 1991.

[4] Choi Young Jin and Chung Wan Kyun, "Multiple tasks manipulation for a robotic manipulator," Advanced Robotics, 2004

[5] C. Wampler II, “Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods,” IEEE Transactions on Systems, Man and Cybernetics, vol. 16, no. 1, pp. 93–101, 1986.

[6] Dragomir N. Nenchev, and Yuichi Tsumaki Masaru Uchiyama, “Singularity-Consistent Parameterization of Robot Motion and Control,” International Journal of Robotics Research, vol. 19, no. 2, pp. 159-182, 2000.

[7] Chiaverini. S.,"Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," IEEE Transactions on Robotics and Automation, 1997.

[8] Y. Nakamura, "Inverse Kinematics Solutions with Singularity Robustness for Robot Manipulator Control", J. Dynamic Systems, Measurement, and Control, 1986.

[9] J. Baillieul, “Kinematic programming alternatives for redundant manipulators,” Proc. *IEEE Int. Conf. on Robotics and Automation*, St. Louis, MO, 1985, p. 722.

[10] Jan Peters, Michael Mistry, Firdaus Udwadia, Jun Nakanishi, and Stefan Schaal, " A unifying framework for robot control with redundant DOFs," Autonomous Robots, 2008.

[11] Paolo Baerlocher and Ronan Boulic, “An inverse kinematics architecture enforcing an arbitrary number of strict priority levels,” Visual Computer, vol. 20, no. 6, pp.402–417, 2004.

[12] O. Khatib, Luis Sentis, and Jaeheung Park, "A Unified Framework for Whole Body Humanoid Robot Control With Multiple Constraints and Contacts," Springer Tracts in Advanced Robotics, 2008.

[13] Gene H. Golub and Charles F. Van Loan, Matrix Computations, 3rd ed.: Johns Hopkins, 1996.

[14] P. H. Chang, “A closed-form solution for inverse kinematics of robot manipulators with redundancy,” IEEE Trans. on Robotics and Automation, vol. 3, no. 5, pp. 393-403, Oct. 1987.

[15] Michael Mistry, Jun Nakanishi, and Stefan Schaal, “Task Space Control with Prioritization for Balance and Locomotion,” International Conference on Intelligent Robots and Systems, 2007.

[16] P. Baerlocher and R. Boulic, “Task-priority formulations for the kinematic control of highly redundant articulated structures,” *Proc. IEEE Int. Conf. on Robotics and Automation*, 1998, p. 323.

[17] G. Antonelli, “Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems,” in Proceedings 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, May 2008, pp. 1993–1998.

[18] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization.* Reading, MA: Addison-Wesley, 1991.