

On the Task Priority Manipulation Scheme with High Execution Performance for a Robotic Manipulator

Youngjin Choi* Wan Kyun Chung† Yonghwan Oh‡ Sang Rok Oh§ and Il Hong Suh¶

Department of Mechanical Engineering
Pohang University of Science & Technology(POSTECH), KOREA
Tel: +82-54-279-2844, Fax: +82-54-279-5899

Abstract

When we are to give simultaneously multiple tasks to a robot manipulator, the task priority manipulation scheme plays an important role, especially for the case that they can not be executed simultaneously. Until now, several algorithms for task priority have been used in solving the inverse kinematics for redundant manipulators. In this paper, through the comparative study of existing algorithms, we propose a new method for task priority manipulation which is developed in terms of two important criteria, algorithmic singularity and task error. This manipulation scheme is applied to planar three link manipulator to demonstrate its effectiveness.

1 Introduction

The concept of task-priority algorithm for manipulators with redundancy was introduced by Maciejewski *et al*[5] and Nakamura *et al*[7]. According to the order of priority, the task with higher priority is firstly performed and the task with lower priority should be performed utilizing the redundancy. However, the conventional schemes suffer from kinematic and algorithmic singularities. The kinematic singularity has been the intrinsic characteristics for robot manipulators, on the contrary, the algorithmic singularity is the artificial one which can be compensated by amending the algorithm. To overcome the difficulties encountered near singularities, the damped least squares inverse and singularity robust inverse algorithms were developed in [6, 10]. Although the continuity and good conditioning of the solution is ensured, these are obtained at the expense of the increased primary and secondary task error due to the effect of non-null damping factor.

Chiaverini[2] suggested the task-priority redundancy resolution technique which has no algorithmic singularity. In the task-priority algorithm of Nakamura *et al*, the algorithmic singularity occurs when the matrix product $\mathbf{H} \{ \mathbf{I} - \mathbf{J}^+ \mathbf{J} \}$ loses the rank with full rank

\mathbf{J} and \mathbf{H} , where \mathbf{J} is Jacobian matrix of the task with higher priority and \mathbf{H} is that of task with lower priority. To avoid the algorithmic singularity, Chiaverini pointed out that the secondary task should be performed in terms of orthogonal projection into the null space of the primary task. Chiaverini's scheme is simple and clear because it does not require the pseudoinverse of $\mathbf{H} \{ \mathbf{I} - \mathbf{J}^+ \mathbf{J} \}$, but it causes a large error for the secondary task.

We will investigate advantages/disadvantages of existing algorithms in detail, and make clear the characteristics of two algorithms. Ultimately, we derive the new task priority algorithm from the comparative study and it does not include an algorithmic singularity without affecting the primary task. Also, it reduces the error of secondary task.

Section 2 investigates the characteristics of Nakamura's algorithm and Chiaverini's algorithm and section 3 proposes and explains the new algorithm based on the contents in section 2. Experimental results are given in section 4. Section 5 draws conclusion. For future notations, $\mathcal{R}(\ast)$ and $\mathcal{N}(\ast)$ represent the range space and null space of matrix \ast , respectively.

2 Comparative Study

To begin with, consider Nakamura's algorithm for task priority redundancy resolution proposed in [5, 7]. The task can be specified by *forward differential kinematics* between the m -dimensional task velocity $\dot{\mathbf{p}} \in \mathbb{R}^m$ and the n -dimensional joint velocity $\dot{\mathbf{q}} \in \mathbb{R}^n$ as follows:

$$\dot{\mathbf{p}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad (1)$$

where the Jacobian matrix $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{m \times n}$ is obtained by the kinematic structure of the manipulator. If $n > m$, then the manipulator has the kinematic redundancy in the sense that there still remains $r = n - m$ dimensional set of self motion velocities which do not interfere with the task velocity. In general, the primary task defined by (1) can be implemented as following form

$$\dot{\mathbf{q}} = \mathbf{J}^+ \dot{\mathbf{p}} + \{ \mathbf{I} - \mathbf{J}^+ \mathbf{J} \} \mathbf{z}, \quad (2)$$

*Graduate Student, yjchoi@postech.edu, POSTECH

†Professor, wkchung@postech.edu, POSTECH

‡Researcher, oyh@amadeus.kist.re.kr, KIST

§Senior Researcher, sroh@amadeus.kist.re.kr, KIST

¶Professor, ihsuh@email.hanyang.ac.kr, HanYang University

where the matrix J^+ is the pseudoinverse¹ and z is the arbitrary vector. In Nakamura's algorithm, the vector z should be chosen to fulfill some secondary requirements. The l -dimensional secondary task is also specified by the differential form of

$$\dot{h} = H(q)\dot{q} \in \mathbb{R}^l, \quad (3)$$

where $H(q) \in \mathbb{R}^{l \times n}$ is Jacobian matrix for the secondary task with lower priority. If we are to achieve the minimum in the sense of the least square error of $\|\dot{h} - H\dot{q}\|_2$, then the least square solution z is obtained by

$$z = [H \{I - J^+J\}]^+ \{\dot{h} - HJ^+\dot{p}\}. \quad (4)$$

If $H \{I - J^+J\}$ has full row rank less than r (the dimension of redundancy), the least square solution is not unique and contains an additional homogeneous term. If we allow the duplicate use of notation, then it results in

$$\begin{aligned} \dot{q} &= J^+\dot{p} + \widetilde{H}^+ \{\dot{h} - HJ^+\dot{p}\} \\ &+ \{I - J^+J\} \{I - \widetilde{H}^+\widetilde{H}\} z \end{aligned} \quad (5)$$

where

$$\widetilde{H} = H \{I - J^+J\} \in \mathbb{R}^{l \times n}. \quad (6)$$

Nakamura's algorithm of (5) can be illustrated by Figure 1. The algorithm is constituted by adding the embedded $J^+\dot{p}$ to the embedded $\widetilde{H}^+(\dot{h} - HJ^+\dot{p})$. It is necessary for $J^+\dot{p}$ to be projected onto the range space of H and it is important to subtract $HJ^+\dot{p}$ from \dot{h} for the exact secondary task execution. Also, we can easily know that the J^+ and \widetilde{H}^+ is geometrically orthogonal because

$$\begin{aligned} (J^+)^T \widetilde{H}^+ &= (JJ^T)^+ J(I - J^+J)H^T(\widetilde{H}\widetilde{H}^T)^+ \\ &= 0. \end{aligned}$$

The residual error becomes zero if $\dot{p} \in \mathcal{R}(J)$ and \widetilde{H} retains full row rank, that is $l \leq r$ and $\mathcal{R}(H^+) \cap \mathcal{R}(J^+) = \emptyset$. As shown in Figure 1, if $\mathcal{R}(H^+)$ does not overlap with $\mathcal{R}(J^+)$, then the scheme of (5) is well performed and the algorithmic singularity does not occur. However, since it is always possible for the range spaces of $\mathcal{R}(J^+)$ and $\mathcal{R}(H^+)$ to overlap for an instant, the algorithmic singularity can be always occurred in this algorithm.

Basically, there are three kinds of singularities in solving *inverse kinematics*, the one is the kinematic singularity, another is the algorithmic singularity and the

¹Note the definition of pseudoinverse that $A^+ = A^T(AA^T)^+ = (A^T A)^+ A^T$, especially $A^+ = (A^T A)^{-1} A^T$ if A has full column rank and $A^+ = A^T(AA^T)^{-1}$ if A has full row rank[4].

²Note that $\mathcal{R}(J^T) = \mathcal{R}(J^+)$ and $\mathcal{R}(H^T) = \mathcal{R}(H^+)$ in Figure 1.

other is the representation singularity. The kinematic singularity takes place in the case of either

$$\text{rank}(J) < m \quad \text{or} \quad \text{rank}(H) < l,$$

especially the latter case is also called "the secondary task singularity". The algorithmic singularity occurs in either

$$\begin{aligned} \mathcal{R}(H^+) \cap \mathcal{R}(J^+) &\neq \emptyset && \text{when } l \leq r \\ \text{or} &&& \\ \mathcal{N}(H) \cap \mathcal{N}(J) &\neq \emptyset. && \text{when } l > r, \end{aligned} \quad (7)$$

where $l < r$ means that the secondary task is under-specified and $l > r$ over-specified for the degrees of redundancy. Also, we should find the transformation between the angular velocity and the rotational velocity, an orientation for which the determinant of this transformation matrix becomes zero is termed the representation singularity[9]. The kinematic singularity is the fundamental problem in solving the inverse kinematics, but the algorithmic singularity is an artificial part different from the kinematic singularity. Algorithmic singularity can be eliminated or changed according to the characteristics of algorithm. Irrespective of the class of singularities, it should be noted that the nominal algorithm of (5) does not produce an acceptable solution near singularities.

To eliminate the algorithmic singularity problem, Chiaverini[2] modified the task priority scheme of (5) as following form

$$\begin{aligned} \dot{q} &= J^+\dot{p} + \{I - J^+J\} H^+\dot{h} \\ &+ \{I - J^+J\} \{I - H^+H\} z. \end{aligned} \quad (8)$$

Chiaverini's scheme of (8) can be illustrated by Figure 2. It consists of adding the embedded $J^+\dot{p}$ to the orthogonal projection of the $H^+\dot{h}$ onto the null space of J . If we overlaps Figure 1 with Figure 2, then Figure 3 is obtained. Now, it is possible to compare the results of both algorithms. Although the scheme of (8) does not include algorithmic singularities, it brings the large error for the secondary task as shown in Figure 3. The amount of residual error in Figure 3 can be calculated as following form:

$$\dot{h} - H\dot{q} = (I - HH^+)\dot{h} - HJ^+(\dot{p} - JH^+\dot{h}), \quad (9)$$

and the residual error (9) has a strong relation with the algebraic condition between J^+ and H^+ . For example, the secondary task has always the error except that $HJ^+ = 0$, i.e. $\mathcal{R}(H^+) \perp \mathcal{R}(J^+)$. Also, the smaller the angle between $\mathcal{R}(H^+)$ and $\mathcal{R}(J^+)$, the larger the secondary task error.

Remark 1 (Nakamura et al. (5) vs. Chiaverini (8))
Although Nakamura's algorithm of (5) has algorithmic singularities, the primary and secondary tasks have no error in the normal case where singularities do not occur. On the contrary, Chiaverini's algorithm of (8) has no algorithmic singularities, however, it has always an error for the secondary task except $HJ^+ = 0$.

3 Task Priority Manipulation Scheme using Weighted Pseudoinverse

Since it is a good advantage that Chiaverini's algorithm has no algorithmic singularity, we will use the structure of Chiaverini's algorithm in developing the new algorithm. Now we are to reduce the residual error of (9) by taking a suitable weighted pseudoinverse instead of pseudoinverse in Chiaverini's algorithm. Then the method of (8) can be modified to

$$\begin{aligned} \dot{q} = & J_W^+ \dot{p} + \{I - J_W^+ J\} H^+ \dot{h} \\ & + \{I - J_W^+ J\} \{I - H^+ H\} z \end{aligned} \quad (10)$$

and the residual error of new algorithm (10) is given by

$$\dot{h} - H\dot{q} = (I - HH^+) \dot{h} - HJ_W^+ (\dot{p} - JH^+ \dot{h}), \quad (11)$$

where $J_W^+ = W^{-1} J^T (JW^{-1} J^T)^{-1}$. The algebraic condition to make the residual error of (11) zero can be obtained as follows:

$$HJ_W^+ = 0 \iff HW^{-1} J^T = 0. \quad (12)$$

As a matter of fact, the condition (12) means the inverse weighted orthogonality using the positive definite weight. If we can find the positive definite weight matrix satisfying above condition (12), then we can expect to remedy the limitation of Chiaverini's algorithm which causes a large secondary task error. The following lemma can help the choice of the weight matrix satisfying (12), but it does not always bring the positive definite weight.

Lemma 1 (Magnus & Neudecker : 1988 [4])

Let $W = J^T J + H^T H \geq 0$, then the following statements are equivalent:

1. $\mathcal{R}(H^+) \cap \mathcal{R}(J^+) = \emptyset$
2. $JW^+ J^T = JJ^+$
3. $HW^+ H^T = HH^+$
4. $JW^+ H^T = 0$.

Lemma 1 says that $HW^+ H^T$ and $JW^+ J^T$ are idempotent if $\mathcal{R}(H^+) \cap \mathcal{R}(J^+) = \emptyset$, and the characteristics of the idempotent brings the inverse weighted orthogonality as follows:

$$\begin{aligned} HW^+ H^T &= HW^+ H^T HW^+ H^T \\ &= HW^+ (J^T J + H^T H) W^+ H^T \\ &\quad - HW^+ J^T J W^+ H^T \\ &= HW^+ H^T - HW^+ J^T J W^+ H^T, \\ \therefore HW^+ J^T &= H(J^T J + H^T H) + J^T = 0. \end{aligned} \quad (13)$$

Additionally, we can obtain the following Corollary from above Lemma:

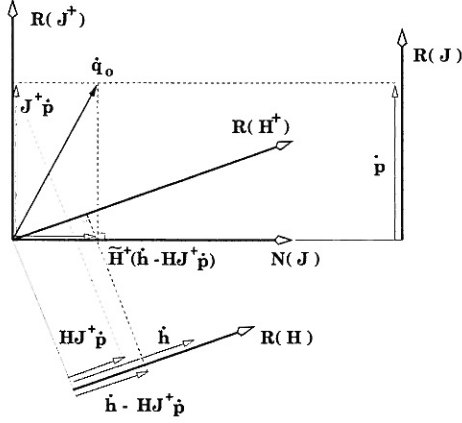


Figure 1: Nakamura's Scheme (5)

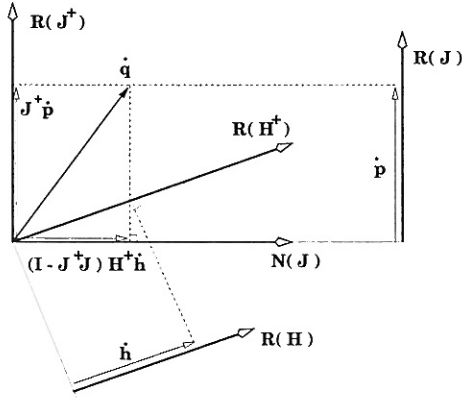


Figure 2: Chiaverini's Scheme (8)

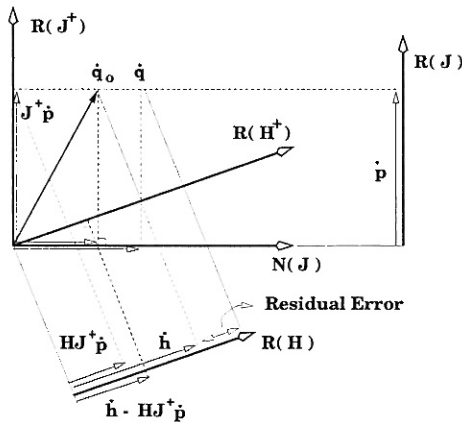


Figure 3: Comparison of Nakamura's Scheme and Chiaverini's Scheme

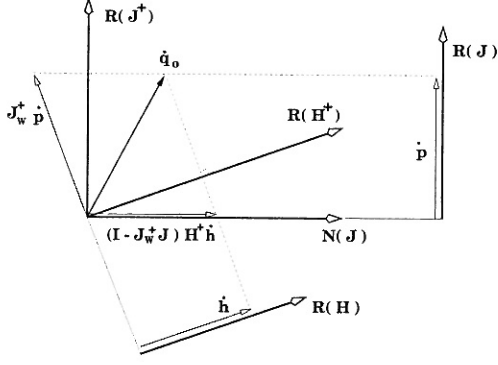


Figure 4: Task Priority Scheme of (10) Using Weighted Pseudoinverse

Corollary 1 (Geometric/Weighted Orthogonality)

If $\mathcal{R}(H^+) \cap \mathcal{R}(J^+) = \emptyset$, then the following statements are always satisfied:

1. $\mathcal{R}(H^+)$ is inverse weighted orthogonal to $\mathcal{R}(J^+)$
2. $\mathcal{R}(H^+)$ is geometric orthogonal to $\mathcal{R}(J_W^+)$.

Proof. 1. Since the condition $\mathcal{R}(H^+) \cap \mathcal{R}(J^+) = \emptyset$ means $HW^+J^T = \mathbf{0}$ by Lemma 1, we can prove the inverse weighted orthogonality as follows:

$$(H^+)^T W^+ J^+ = (HH^T)^+ HW^+ J^T (JJ^T)^+ = \mathbf{0}.$$

2. Using the same procedure, the geometric orthogonality is proved as follows:

$$(H^+)^T J_W^+ = (HH^T)^+ HW^+ J^T (JW^+ J^T)^+ = \mathbf{0}.$$

□

The positive semi-definite weight matrix in Lemma 1 rotates the range space of J^+ as shown in Figure 4. In that Figure, one should observe that the range space of the resulting weighted pseudoinverse of J is geometrically orthogonal to $\mathcal{R}(H^+)$, so that the direction of projection by $(I - J_W^+ J)$ is aligned with the direction of projection by H^+ . However, since the inverse weighted orthogonality of (13) is equivalent to the first item of Lemma 1, the inverse weighted orthogonality of (13) brings another constraint such that $\mathcal{R}(H^+) \cap \mathcal{R}(J^+)$ should be the empty set (\emptyset). However, this is one of two conditions (7) which causes the algorithmic singularity.

To exclude the algorithmic singularity like Chiaverini's algorithm, we need the positive definite weight $W > \mathbf{0}$ satisfying (12), not positive semi-definite weight of Lemma 1. Hence, we define the new weight matrix by adding the positive small constant to the positive semi-definite weight. Although it more or less contaminates the characteristics of the inverse weighted orthogonality with small positive number ϵ

$$H(J^T J + H^T H + \epsilon I)^{-1} J^T \approx \mathbf{0}, \quad (14)$$

it definitely eliminates an algorithmic singularity in new algorithm of (10). And it does not affect the execution performance of primary task, though it somewhat degrades the performance of secondary task. Also, since we use the differential kinematics to solve the inverse kinematics, it is possible to remain the error in the position level. To remedy this problem, the desired velocity should be modified to the reference velocity as follows

$$\dot{p}_{REF} = \dot{p}_d + K_p(p_d - p) \quad (15)$$

$$\dot{h}_{REF} = \dot{h}_d + K_h(h_d - h), \quad (16)$$

where \dot{p}_d, p_d are the desired velocity and position of primary task, \dot{h}_d, h_d are the desired velocity and position of secondary task, p, h are the real positions of manipulator for the primary and secondary task respectively, and K_p, K_h are suitable gain matrices. Now, we modify the algorithm of (10) as following form

$$\begin{aligned} \dot{q} &= J_W^+ \dot{p}_{REF} + \{I - J_W^+ J\} H^+ \dot{h}_{REF} \\ &\quad + \{I - J_W^+ J\} \{I - H^+ H\} z, \end{aligned} \quad (17)$$

where $W = J^T J + H^T H + \epsilon I > \mathbf{0}$ and $J_W^+ = W^{-1} J^T (JW^{-1} J^T)^{-1}$. Making the weight matrix to be strictly positive definite implies that new algorithm of (17) does not include the algorithmic singularity. Also, it does not affect the execution performance of primary task as shown in the following error equation

$$\begin{aligned} &\dot{p}_{REF} - J\dot{q} \\ &= (I - JJ_W^+) \dot{p}_{REF} - J(I - J_W^+ J) H^+ \dot{h}_{REF} = \mathbf{0}. \end{aligned}$$

Also, it is superior to Chiaverini's algorithm in that it brings the smaller error for secondary task as follows.

$$\begin{aligned} &\dot{h}_{REF} - H\dot{q} \\ &= (I - HH^+) \dot{h}_{REF} - HJ_W^+ (\dot{p}_{REF} - JH^+ \dot{h}_{REF}) \\ &= HW^{-1} J^T (JW^{-1} J^T)^{-1} (\dot{p}_{REF} - JH^+ \dot{h}_{REF}) \\ &\approx \mathbf{0}, \end{aligned}$$

because $HW^{-1} J^T \approx \mathbf{0}$. Therefore, the new task priority algorithm (17) does not include any algorithmic singularity and does not cause any task error in primary task and brings the smallest error for the secondary task among existing algorithms.

To generalize the new algorithm (17), now we develop the recursive formulation for n -tasks. Since the weight is dependent on the Jacobian matrices according to tasks, the forward recursive method can not be applied to new algorithm (17). However, the backward recursive method can be applied as follows:

$$\begin{aligned} &\text{From } i = n - 1 \text{ to } i = 1 \\ \dot{q}_i &= J_{iW}^+ \dot{x}_i + (I - J_{iW}^+ J_i) \dot{q}_{i+1} \end{aligned}$$

where the initial $\dot{q}_n = J_n^+ \dot{x}_n + (I - J_n^+ J_n) z$, the weight $W = \sum_{i=1}^n J_i^T J_i + \epsilon I$ and \dot{x}_i is the reference velocity of i -th task. Finally, \dot{q}_1 is the resultant of the recursive formulation of new algorithm.

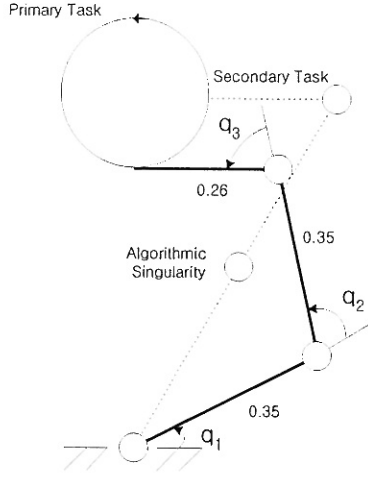


Figure 5: Desired Tasks and Algorithmic Singularity

Remark 2 (Recursive Formulations of Nakamura's and Chiaverini's algorithm : [1])

Nakamura's algorithm is obtained by the forward recursive method

$$\begin{aligned} & \text{From } i = 1 \text{ to } i = n \\ & \dot{q}_i = \dot{q}_{i-1} + (J_i P_{i-1})^+ (\dot{x}_i - J_i \dot{q}_{i-1}), \quad \dot{q}_0 = 0 \\ & P_i = P_{i-1} - (J_i P_{i-1})^+ (J_i P_{i-1}), \quad P_0 = I. \end{aligned} \quad (19)$$

Also, Chiaverini's algorithm is obtained by the backward recursive method

$$\begin{aligned} & \text{From } i = n \text{ to } i = 1 \\ & \dot{q}_i = J_i^+ \dot{x}_i + (I - J_i^+ J_i) \dot{q}_{i+1}, \quad \dot{q}_{n+1} = z. \end{aligned} \quad (20)$$

4 Experimental Results

The experiment was implemented for tasks of position prior to orientation for 3 revolute joint planar manipulators as shown in Figure 5. The primary task tracks the desired circular trajectory in Figure 5 and the secondary task keeps the orientation of a manipulator constantly as follows:

1. Circular Trajectory Tracking : the center point is $(0.0m, 0.65m)$ and a radius is $0.15m$
2. Orientation : $q_1 + q_2 + q_3 = 180^\circ$.

In this case, the Jacobian matrix of primary task is given by

$$J = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \end{bmatrix} \quad (21)$$

The vector spanning the null space of Jacobian matrix can be found by the cross product of the first and second rows of (21) as follows:

$$Z = [l_2 l_3 s_3, -l_3(l_1 s_{23} + l_2 s_3), l_1(l_2 s_2 + l_3 s_{23})]^T \quad (22)$$

where $JZ = 0$. Since the Jacobian matrix of secondary task keeping the orientation of 180° is expressed by $H = [1, 1, 1]$, the algorithmic singularity in Nakamura's algorithm (5) occurs when

$$\begin{aligned} & \mathcal{R}(H^+) \cap \mathcal{R}(J^+) \neq \emptyset \\ & \quad \downarrow \\ & HZ = 0 \leftrightarrow l_1 l_2 s_2 = 0 \leftrightarrow q_2 = 0^\circ \text{ or } 180^\circ. \end{aligned} \quad (23)$$

Here, we utilize two algorithms which do not include an algorithmic singularity:

$$\dot{q} = J^+ \dot{p}_{REF} + \{I - J^+ J\} H^+ \dot{h}_{REF} \quad (24)$$

$$\dot{q} = J_W^+ \dot{p}_{REF} + \{I - J_W^+ J\} H^+ \dot{h}_{REF} \quad (25)$$

where (24) is Chiaverini's algorithm and (25) is our algorithm. To begin with, the gain matrices K_p and K_h in \dot{p}_{REF} , \dot{h}_{REF} are set to $10.0I$ and $10.0I$ respectively. And the ϵ value in weight matrix is set to 0.2. Total execution time is 10 seconds. The link lengths are 0.35m, 0.35m and 0.26m as shown in Figure 5. Above two algorithms do not include the algorithmic singularity and they are excellent in execution performance for the primary task as shown in Figure 6.(a) and (b). Also, Figure 6.(c) shows that our algorithm brings the smaller error than Chiaverini's scheme for the same gains K_p and K_h . On the contrary, Chiaverini's scheme causes the secondary task error successively during the experiment. However, according as the gain values are increased in Chiaverini's algorithm, the secondary task error can be reduced as shown in Figure 6.(c) and 7.(c). Also, the configuration velocity of our algorithm is larger to reduce the secondary task error than that of Chiaverini's algorithm as shown in Figure 6.(d). Finally, we can know that the applied torque profiles are similar for two algorithms as shown in Figure 6.(e).

5 Concluding Remarks

Through the comparative study for the existing task priority manipulation schemes, we suggested new task priority manipulation method. The core of new algorithm was that we utilized the weighted pseudoinverse in place of pseudoinverse. Then the weight matrix satisfying the inverse weighted orthogonality was composed of Jacobian matrices for primary/secondary tasks. However, since it could be positive semi-definite, it was necessary to make the weight strictly positive definite matrix without affecting the performance of the primary task. Also, the new method brings the smaller error for the secondary task comparing to conventional methods. In view of the task error and algorithmic singularity, the validity of new algorithm have been shown through the experiment.

1. Algorithm (24)

2. Algorithm (25)

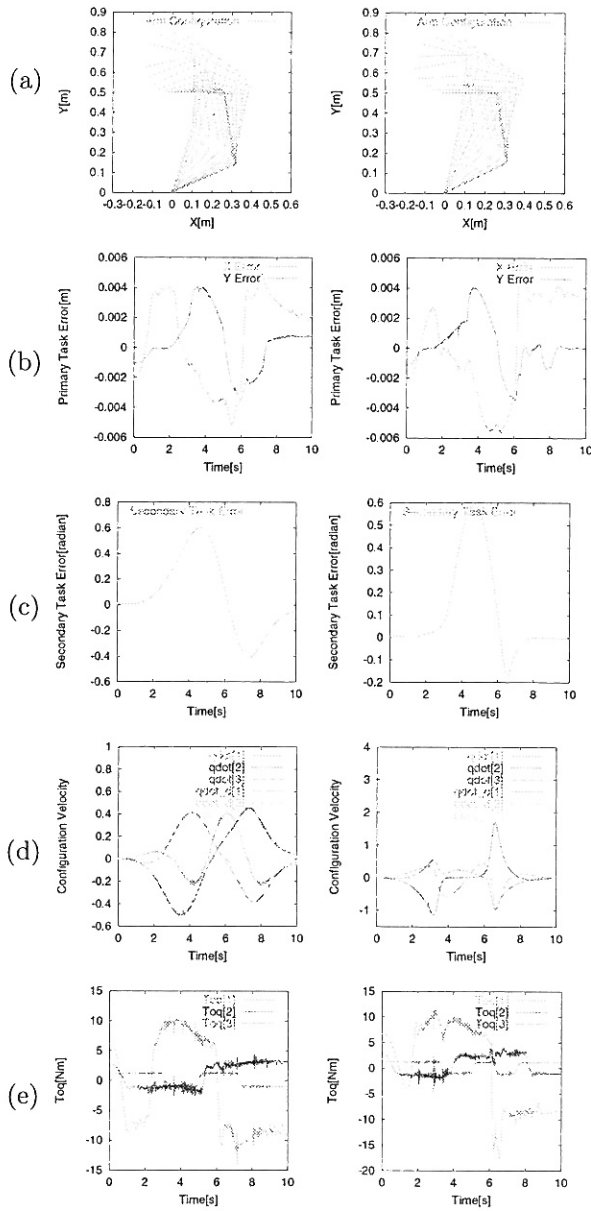


Figure 6: Comparison of Chiaverini's and Our Algorithm (a)Arm Configuration, (b)Primary Task Error, (c)Secondary Task Error, (d)Configuration Velocity \dot{q} and (e)Applied Torque when $K_p = 10.0I$ and $K_h = 10.0I$

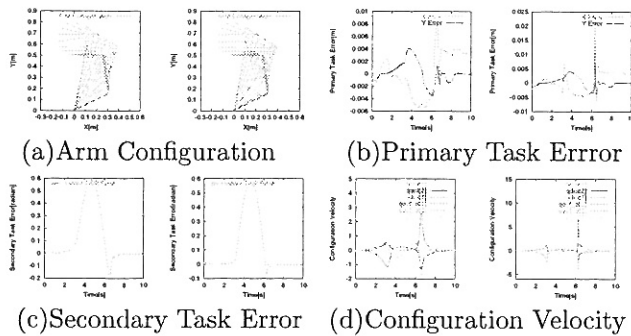


Figure 7: When $K_p = 250.0I$ and $K_h = 250.0I$

References

- [1] P. Baerlocher and R. Boulic, "Task-Priority Formulations for the Kinematically Control of Highly Redundant Articulated Structures", *Proceedings of the IROS*, pp. 323–329, 1998.
- [2] S. Chiaverini, "Singularity-Robust Task-Priority Redundancy Resolution for Real-Time Kinematic Control of Robot Manipulators," *IEEE Trans. on Robotics and Automation*, vol. 13, No. 3, pp. 398–410, 1997.
- [3] J. Lenarcic, "On the Execution of the Secondary Task of Redundant Manipulators," *Proceedings of the 3th ICAM*, pp. 370–375, 1998.
- [4] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, John Wiley & Sons, 1988.
- [5] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments", *Int. J. of Robotics Research*, vol. 4, No. 3, pp. 109–117, 1985.
- [6] Y. Nakamura, *Advanced Robotics, Redundancy and Optimization*, Addison Wesley, 1991.
- [7] Y. Nakamura, H. Hanafusa and T. Yoshikawa, "Task-Priority Based Redundancy Control of Robot Manipulators," *Int. J. of Robotics Research*, vol. 6, No. 2, pp. 3–15, 1987.
- [8] J. Park, Y. Choi, W. K. Chung and Y. Youm, "Accommodation of Multiple Tasks using Weighted Pseudo-Inverse for Kinematically Redundant Manipulators," *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 4041–4047, 2001.
- [9] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*, McGraw-Hill, 1996.
- [10] C. W. Wampler, "Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-16, No. 1, pp. 93–101, 1986.