A Constraints-based Method of the Inverse Kinematics for Redundant Manipulators

Y. W. SUNG*, D. K. CHO*, M. J. CHUNG*, AND K. KOH**

*Dept. of Electrical Engineering, Korea Advanced Institute of Science and Technology, 373-1 Kusong-dong, Yusong-gu, Taejon, 305-701, Korea

**Research & Development Laboratory, Goldstar Industrial Systems Co. LTD, 533 Hogye-Dong, Anyang, Kyungki-Do, 430-080, Korea

Abstract - A redundant manipulator can achieve additional tasks by utilizing the degree of redundancy, in addition to a basic motion task. While some additional tasks can be represented as an objective function to be optimized, other additional tasks can be represented as kinematic inequality constraints. In this paper, we reformulate a redundancy resolution problem with multiple criteria into a local constrained optimization problem, and propose a new method for solving it. The proposed method is especially efficient when the number of additional tasks is larger than the degree of redundancy. It also systematically assigns the priorities between the additional tasks. Besides of computational efficiency, the method has a cyclic property.

1. Introduction

Because of its dexterity and versatility, a kinematically redundant manipulator has been a major research topic in robotics over the past two decades[1-9]. A redundant manipulator can be defined as a manipulator that has more degrees of freedom than those necessary to perform a given task.

Let $\mathbf{x} \in R^m$, $\boldsymbol{\theta} \in R^n$ be the task space and the joint space variables respectively, then the relationship between these two variables is defined as (1) by a kinematic function $\mathbf{f}: R^n \rightarrow R^m$.

$$\mathbf{x}(t) = \mathbf{f}(\boldsymbol{\theta}(t)) \tag{1}$$

At velocity level,

$$\dot{\mathbf{x}} = \mathbf{J}(\boldsymbol{\theta})\,\dot{\boldsymbol{\theta}} \tag{2}$$

where $\mathbf{J} = \dfrac{\partial \mathbf{f}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ is the $m \times n$ Jacobian matrix.

The difference $n-m$ is termed the degrees of redundancy (DOR). The advantage of a redundant manipulator is that it can perform various additional tasks in addition to the basic motion task. Two possible approaches for resolving the redundancy are the local optimization method and the global optimization method[2]. Although the former has some drawbacks, it is adequate for real time implementation. One way of locally resolving the redundancy is to transform a redundant system into a nonredundant one by extending the dimension of task space. Baillieul[3] and Chang[5] used a necessary condition for $\boldsymbol{\theta}$ to optimize an objective function $H(\boldsymbol{\theta})$ subject to the positional constraint (1) :

$$\mathbf{z}(\boldsymbol{\theta})\,\mathbf{h}(\boldsymbol{\theta}) = 0 \tag{3}$$

where $\mathbf{z}$ is an $r \times n$ matrix that represents the null space of the Jacobian $\mathbf{J}$ and $\mathbf{h}$ is the n dimensional gradient vector of $H(\boldsymbol{\theta})$. Chang[5] solved (1) and (3) numerically. Baillieul proposed the extended Jacobian method that resolved the redundancy at velocity level. By partially differentiating (3) with respect to time, (3) becomes

$$\frac{\partial \mathbf{zh}}{\partial \boldsymbol{\theta}}\,\dot{\boldsymbol{\theta}} = 0 \tag{4}$$

Baillieul used (2) and (4) to obtain the extended Jacobian

$$\begin{bmatrix} \dot{\mathbf{x}} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{J} \\ \dfrac{\partial \mathbf{zh}}{\partial \boldsymbol{\theta}} \end{bmatrix} \dot{\boldsymbol{\theta}} \triangleq \begin{bmatrix} \mathbf{J} \\ \mathbf{J}_c \end{bmatrix} \dot{\boldsymbol{\theta}} \triangleq \mathbf{J}_e\,\dot{\boldsymbol{\theta}} \tag{5}$$

In (5), $\mathbf{J}_e$ is square and the solution for joint velocities can be obtained by using the inverse of $\mathbf{J}_e$.

Above mentioned two methods are basically local optimization methods with equality constraints of (1). Despite some advantages, these methods are not efficient when we deal with a multiple criteria problem[4] , or when there exist inequality constraints. Requirements for some additional tasks, such as obstacle avoidance, joint limit avoidance, etc., can be expressed as inequality constraints. A few researchers have dealt with redundancy resolution method that allows inequality constraints to be incorporated. Seraji developed the configuration control approach[6] as a unified motion control of a redundant manipulator. In the improved version of that approach[7], he used a damped least squares technique with replacing the active inequality constraints by equality ones. However, Seraji's approach has difficulties in determining the weighting factors, particularly when there are multiple objectives. Carignan[8], in his direct Lagrangian approach, converted the inequality constraints represented at position level into those at velocity level and formed a local constrained optimization problem to minimize the weighted velocity norm and solved it numerically by using duality. His method seems to be well suited only to the optimization of velocity norm and computationally inefficient.

In this paper, a new and computationally efficient method for redundancy resolution is presented, which can directly deal with inequality constraints. The proposed method is well suited to a multiple criteria problem, especially when the number of additional tasks is larger than DOR.

2. Inverse Kinematic Solutions

2.1 Problem Formulation

Equation (1) describes a basic motion task for a redundant manipulator to follow. As mentioned in Section 1, requirements for some additional tasks are represented as an inequality constraint or a set of inequality constraints. Meanwhile, some additional tasks, such as singularity avoidance, can be performed by optimizing a proper objective function. Assume that $H(\boldsymbol{\theta})$ is an objective function to be maximized for one additional task and the other additional tasks are represented by a set of inequality constraints as (6).

$$G_i(\boldsymbol{\theta}) \le 0, \quad i=1, ..., l \tag{6}$$

Also assume that $H(\boldsymbol{\theta})$ and $G_i(\boldsymbol{\theta})$, $(i=1, ..., l)$ have first-order partial derivatives with respect to joint variables. We can reformulate the redundancy resolution problem with multiple criteria into the following local constrained optimization problem:

$$\begin{aligned} &\textit{maximize } H(\boldsymbol{\theta}) \\ &\textit{subject to } \ \mathbf{x} - \mathbf{f}(\boldsymbol{\theta}) = 0 \\ &\qquad\qquad \mathbf{G}(\boldsymbol{\theta}) \le 0 \end{aligned} \tag{7}$$

where $\mathbf{G}(\boldsymbol{\theta}) = [G_1(\boldsymbol{\theta})\ G_2(\boldsymbol{\theta})\ \cdots\ G_l(\boldsymbol{\theta})]^T$.

The Lagrangian function $L(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ is given as follows:

$$L(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = H(\boldsymbol{\theta}) + \boldsymbol{\lambda}^T(\mathbf{x} - \mathbf{f}(\boldsymbol{\theta})) + \boldsymbol{\mu}^T \mathbf{G}(\boldsymbol{\theta}) \tag{8}$$

where $\boldsymbol{\lambda} = [\lambda_1 \cdots \lambda_m]^T$ and $\boldsymbol{\mu} = [\mu_1 \cdots \mu_l]^T$.

The necessary condition for optimality is obtained as (9) ~ (12) by the Kuhn-Tucker theorem[10].

$$\frac{\partial L}{\partial \boldsymbol{\theta}} = \mathbf{h}^T - \boldsymbol{\lambda}^T \mathbf{J} + \boldsymbol{\mu}^T \mathbf{g}^T = 0 \tag{9}$$

$$\frac{\partial L}{\partial \boldsymbol{\lambda}} = \mathbf{x} - \mathbf{f}(\boldsymbol{\theta}) = 0 \tag{10}$$

$$\mu_i G_i(\boldsymbol{\theta}) = 0, \quad i = 1, \cdots, l \tag{11}$$

$$\boldsymbol{\mu} \le 0 \tag{12}$$

where $\mathbf{h} \triangleq (\frac{\partial H}{\partial \boldsymbol{\theta}})^T$ is the n dimensional gradient vector, and $\mathbf{g} \triangleq [\mathbf{g}_1\ \mathbf{g}_2\ \cdots\ \mathbf{g}_l] = [(\frac{\partial G_1}{\partial \boldsymbol{\theta}})^T\ (\frac{\partial G_2}{\partial \boldsymbol{\theta}})^T\ \cdots\ (\frac{\partial G_l}{\partial \boldsymbol{\theta}})^T]$ is the $n \times l$ matrix.

From (9), we get

$$\mathbf{h} + \mathbf{g}\boldsymbol{\mu} = \mathbf{J}^T \boldsymbol{\lambda} \tag{13}$$

Let $\mathbf{z}$ be a full-rank $r \times n$ matrix representing the null space of the Jacobian $\mathbf{J}$. One possible formulation for z is

$$\mathbf{z} = [\mathbf{J}_{n-m} \mathbf{J}_m^{-1} : -\mathbf{I}_{n-m}] \tag{14}$$

where $\mathbf{J}_m = [\mathbf{J}^1 \cdots \mathbf{J}^m]^T$ and $\mathbf{J}_{n-m} = [\mathbf{J}^{m+1} \cdots \mathbf{J}^n]^T$ with $\mathbf{J}^i$ being the ith column vector of $\mathbf{J}$. Premultiplied by $\mathbf{z}$, (13) becomes

$$\mathbf{z}(\mathbf{h} + \mathbf{g}\boldsymbol{\mu}) = 0 \tag{15}$$

The equations (10),(11),(12) and (15) will be denoted as a set of equations for optimal solution(EOS) from now on. Instead of directly solving the nonlinear equations of the EOS at position level, we propose an algorithm to solve it at velocity level for the computational efficiency.

2.2 The Proposed Algorithm

Among 2^l cases, only the following two cases need to be considered in the proposed method. Note that when an inequality constraint is inactive, *i.e.*, $G_i(\boldsymbol{\theta}) < 0$, the corresponding Lagrange multiplier μ_i must be zero because equation (11) must be held.

Case I : All of the inequality constraints are inactive, so all the Lagrange multipliers are zero. In this case, the EOS is reduced to (16).

$$\begin{aligned} \mathbf{x} &= \mathbf{f}(\boldsymbol{\theta}) \\ \mathbf{z}\mathbf{h} &= 0 \end{aligned} \tag{16}$$

The solution to (16) must satisfy all the inequality constraints given by (6).

Case II : One of the inequality constraints, say G_j, is active. In this case, Lagrange multipliers except μ_j are zero, so the EOS is reduced to (17).

$$\begin{aligned} \mathbf{x} &= \mathbf{f}(\boldsymbol{\theta}) \\ \mathbf{z}\mathbf{h} + \mathbf{z}\mathbf{g}_j \mu_j &= 0 \\ G_j(\boldsymbol{\theta}) &= 0 \end{aligned} \tag{17}$$

The equations (17) must be solved for the unknown $\boldsymbol{\theta}$ and μ_j, and its solution must satisfy the condition given by (12).

In the proposed method, any solution of a fixed time step belongs to either Case I or Case II. The proposed method gets a solution of the next time step based on the Case to which the solution of the current time step belongs. By partially differentiating (16) with respect to time, we have (18).

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{J}\dot{\boldsymbol{\theta}} \\ \frac{\partial \mathbf{z}\mathbf{h}}{\partial \boldsymbol{\theta}} \dot{\boldsymbol{\theta}} &= 0 \end{aligned} \tag{18}$$

or in matrix form

$$\begin{bmatrix} \dot{\mathbf{x}} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{J} \\ \frac{\partial \mathbf{z}\mathbf{h}}{\partial \boldsymbol{\theta}} \end{bmatrix} \dot{\boldsymbol{\theta}} \triangleq \begin{bmatrix} \mathbf{J} \\ \mathbf{J}_c \end{bmatrix} \dot{\boldsymbol{\theta}} \triangleq \mathbf{J}_e \dot{\boldsymbol{\theta}} \tag{19}$$

Here $\mathbf{J}_e$ is an n dimensional square matrix and if it is not singular, (19) can be solved as follows:

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_e^{-1} \begin{bmatrix} \dot{\mathbf{x}} \\ 0 \end{bmatrix}, \qquad \mu_i = 0, \quad i = 1, ..., l \tag{20}$$

If a solution of current time step belongs to Case I, a solution candidate of the next time step is obtained by using (20), which means that all of the inequality constraints are disregarded. Therefore, all the Lagrange multipliers are forced to remain zero in (20). The obtained solution satisfies (16) and must satisfy all the inequality constraints.

By partially differentiating (17) with respect to time, we have (21).

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\boldsymbol{\theta}} \tag{21}$$

$$\frac{\partial(\mathbf{z}\,\mathbf{h} + \mathbf{z}\,\mathbf{g}_j\,\mu_j)}{\partial\boldsymbol{\theta}}\dot{\boldsymbol{\theta}} + \mathbf{z}\,\mathbf{g}_j\,\dot{\mu}_j = 0$$

$$\mathbf{g}_j^T\dot{\boldsymbol{\theta}} = 0$$

or in matrix form

$$\begin{bmatrix}\dot{\mathbf{x}}\\0\\0\end{bmatrix} = \begin{bmatrix}\mathbf{J} & 0\\ \dfrac{\partial(\mathbf{zh} + \mathbf{zg}_j\mu_j)}{\partial\boldsymbol{\theta}} & \mathbf{zg}_j\\ \mathbf{g}_j^T & 0\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{\theta}}\\ \dot{\mu}_j\end{bmatrix} \triangleq \begin{bmatrix}\mathbf{J}_1\\ \mathbf{J}_2\\ \mathbf{J}_3\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{\theta}}\\ \dot{\mu}_j\end{bmatrix} \triangleq \mathbf{J}_{ee}\begin{bmatrix}\dot{\boldsymbol{\theta}}\\ \dot{\mu}_j\end{bmatrix} \tag{22}$$

Here $\mathbf{J}_{ee}$ is an $n+1$ dimensional square matrix and if it is not singular, (22) can be solved as follows:

$$\begin{bmatrix}\dot{\boldsymbol{\theta}}\\ \dot{\mu}_j\end{bmatrix} = \mathbf{J}_{ee}^{-1}\begin{bmatrix}\dot{\mathbf{x}}\\0\\0\end{bmatrix}, \quad \dot{\mu}_i = 0 \text{ for } i \neq j \tag{23}$$

If a solution of current time step belongs to Case II, a solution candidate of the next time step is obtained by using (23), which means that it is obtained along the boundary of $G_j = 0$ with the other inequality constraints being disregarded. Therefore, all the Lagrange multipliers except μ_j are forced to remain zero in (23). The obtained solution satisfies (17) and must satisfy the condition given by (12).

The basic idea is as follows. If all of the inequality constraints are inactive at an initial optimal solution and along a whole task space trajectory, $\mathbf{x}(t)$, the solution trajectory $\boldsymbol{\theta}(t)$ for $\mathbf{x}(t)$ can be uniquely determined by recursively applying (20). It is the same one that is obtained by the Extended Jacobian method. In other words, the inactive inequality constraints can be disregarded because those have no effects at all on the constrained optimization problem. The obtained solution trajectory is the one that maximizes the objective function $H(\boldsymbol{\theta})$, or maintains $\mathbf{zh} = 0$, as well as performs the basic motion task and the other additional tasks. However, in general, getting a solution by (20) may be 'blocked' by one or more inequality constraints at some time step. In other words, the solution candidate obtained by (20) may be the one that violates some inequality constraints. We will term these inequality constraints as blocking constraints. If there exist some blocking constraints, the proposed method gets solutions by (23) along the boundary of one of the blocking constraints in such a manner that all of the inequality constraints are satisfied. In this case, the Lagrange multiplier corresponding to that active inequality constraint is used as an indicator for relaxing it or for making it inactive. Before describing the proposed algorithm in detail, we will make some definitions.

A switching point $\boldsymbol{\theta}_s$ is defined as a solution of EOS that satisfies $G_j(\boldsymbol{\theta}_s) = 0$ and $\mu_j = 0$ simultaneously. At a given point $\boldsymbol{\theta}_b$, an index set of blocking constraints, denoted as B, is defined as (24).

$$B \triangleq \{\, j \mid G_j(\boldsymbol{\theta}_b) > 0 \,\} \tag{24}$$

Let $\boldsymbol{\theta}_k$, $\boldsymbol{\mu}(k)$, and $\mu_j(k)$ denote $\boldsymbol{\theta}(t_k)$, $\boldsymbol{\mu}(t_k)$ and $\mu_j(t_k)$ respectively.

The proposed algorithm is described as follows:

Step 1 :

(a) Find an initial optimal solution $\boldsymbol{\theta}_0$ ($\boldsymbol{\mu}(0) = 0$), and set $k = 0$.

(b) Get $\dot{\boldsymbol{\theta}}$, $\dot{\boldsymbol{\mu}}$ by (20), and get $\boldsymbol{\theta}_{k+1}$, $\boldsymbol{\mu}(k+1)$.

(c) Evaluate B at $\boldsymbol{\theta}_{k+1}$.

(d) If $B = \{\ \}$, then select $\boldsymbol{\theta}_{k+1}$, $\boldsymbol{\mu}(k+1)$ as a solution, set $k = k+1$, and go to Step 1(b),
else if $B \neq \{\ \}$, then discard $\boldsymbol{\theta}_{k+1}$, $\boldsymbol{\mu}(k+1)$ and go to Step 2(a).

Step 2 :

(a) Choose a G_j, $j \in B$.

(b) Get $\dot{\boldsymbol{\theta}}$, $\dot{\boldsymbol{\mu}}$ by (23) and get $\boldsymbol{\theta}_{k+1}$, $\boldsymbol{\mu}(k+1)$.

(c) Evaluate B at $\boldsymbol{\theta}_{k+1}$.

(d) If $B \neq \{\}$, then discard $\boldsymbol{\theta}_{k+1}$, $\boldsymbol{\mu}(k+1)$ and go to Step 2(a).
else if $B = \{\}$, then select $\boldsymbol{\theta}_{k+1}$, $\boldsymbol{\mu}(k+1)$ as a solution, set $k = k+1$, and go to Step 3(b).

Step 3 :

(a) For a G_j, $j \in B$, get $\mu_j(k)$ from $\mathbf{zh}+\mathbf{zg}_j\mu_j = 0$ and let $\mu_i(k) = 0$ for $i \in B$, $i \neq j$.

(b) Get $\dot{\boldsymbol{\theta}}$, $\dot{\boldsymbol{\mu}}$ by (23), and get $\boldsymbol{\theta}_{k+1}$, $\boldsymbol{\mu}(k+1)$.

(c) Evaluate B at $\boldsymbol{\theta}_{k+1}$.

(d) If $B \neq \{\ \}$, then discard $\boldsymbol{\theta}_{k+1}$, $\boldsymbol{\mu}(k+1)$ and go to Step 3(a).
else if $B = \{\ \}$, then select $\boldsymbol{\theta}_{k+1}$, $\boldsymbol{\mu}(k+1)$ as a solution, and
if $\mu_j(k+1) < 0$, set $k = k+1$, and go to Step 3(b).
else if $\mu_j(k+1) = 0$, set $k = k+1$, and go to Step 1(b).

In general, we can assume that all of the inequality constraints are inactive at the initial time. In Step 1, the proposed method gets a solution candidate by disregarding all of the inequality constraints. If the solution candidate satisfies all of the inequality constraints, $B = \{\}$, it is selected as a solution and the same procedure is repeated. If some inequality constraints are violated at the solution candidate, $B \neq \{\}$, it must be discarded, and Step 2 follows.

Step 2 is the procedure to determine one inequality constraint along the boundary of which we can get a solution of the next time step. At the beginning of Step 2, the solution of the current time step satisfies $\mathbf{zh}=0$. In this step, the method selects one inequality constraint in the set B and obtains a solution candidate along the boundary of that inequality constraint by (25). Then, inequality constraints are evaluated at the solution candidate so that a new set B is obtained. If the set B is not null, meaning that the solution candidate again violates the inequality constraints G_k, $k \in B$, it is discarded. Then, the same procedure is repeated for the set B. When the obtained solution candidate satisfies all of the inequality constraints, it becomes a solution. Similarly, Step 3 determines one inequality constraint among the blocking constraints. The number of trials to determine one inequality constraint in Step 2 or Step 3 is equal to that of the blocking constraints at worst. Furthermore, the number of the blocking constraints at one point is not so large in real application.

Once one active inequality constraint G_j is determined in Step 2 or Step 3, a solution of the next time step is obtained along the boundary of that inequality constraint as in Step 3. Thus, the proposed method considers one inequality constraint at a time by forcing the others to be inactive, *i.e.*, by forcing all the Lagrange multipliers except the corresponding one to be zero. That is the reason why the Lagrange multiplier corresponding to the chosen active inequality constraint needs to be adjusted in Step 3(a). This is further explained in the next section. The solution obtained in Step 3 lies on the boundary of $G_j=0$ and satisfies (19). Provided that $\mathbf{z}\,\mathbf{g}_j$ is not equal to zero, $\mu_j=0$ means $\mathbf{zh}=0$, and *vice versa*. Therefore, at the point on the boundary of G_j where μ_j becomes zero, $\mathbf{zh}$ becomes zero, too. At this switching point, there is a possibility that we can relax the active inequality constraint, *i.e.*, we can get a solution of the next time step that maximizes $H(\theta)$ without being affected by any inequality constraint (See Step 3(d)).

2.3 Characteristics of the Proposed Algorithm

In the proposed algorithm, the Lagrange multipliers make it possible to systematically assign the priorities between the additional tasks. We will briefly consider the meaning of the Lagrange multipliers. If there is no inequality constraint or all of the inequality constraints are inactive, (16) must be satisfied at an optimal solution of the kth time step, $\theta(k)=\theta(t_k)$. In (16), $\mathbf{z}(\theta(k))\,\mathbf{h}(\theta(k))=0$ means that $\mathbf{z}(\theta(k))$ and $\mathbf{h}(\theta(k))$ are orthogonal, so there is no direction along which we can increase the value of $H(\theta)$. It means there is no self-motion. This is depicted in Fig. 1(a) for the case of one DOR ,where the dimension of $\mathbf{z}$ is one. Meanwhile, (17) must be satisfied at $\theta(k)$ when one inequality constraint is active. In this case, $\mathbf{z}(\theta(k))$ and $\mathbf{h}(\theta(k))+\mathbf{g}_i(\theta(k))\,\mu_i(k)$ are orthogonal to each other. Therefore, $\mathbf{z}\,\mathbf{h}+\mathbf{z}\,\mathbf{g}_i\,\mu_i=0$ holds at the solution $\theta(k)$. In Fig. 1(b), moving $\theta(k)$ toward a point A further increases the value of $H(\theta)$, but it also increases the value of $G_i(\theta)$ to violate the inequality constraint. Therefore, $\theta(k)$ is the optimal solution that maximizes $H(\theta)$ within the feasible region. At $\theta(k)$, $\mu_i(k)$ can be interpreted as a scalar parameter that makes $\mathbf{zh}+\mathbf{zg}_i\mu_i=0$ hold. It also indicates how far the current point is located from the point that satisfies $\mathbf{zh}=0$.

As explained in Section 2.2, the proposed method considers one inequality constraint at a time. Suppose that a set of solutions have been obtained along the boundary of $G_i=0$ until $t=t_k$, at which $\mathbf{zh}+\mathbf{zg}_i\mu_i=0$ holds and $\mu_j(k)$ is equal to zero (See Fig. 1(b)). If $\mathbf{zh}\neq 0$, $\mu_i(k)$ is less than zero and the $(k+1)$th solution must be obtained along the boundary of $G_i=0$ as described in Step 3(d) of section 2.2. Suppose, at $t=t_k$, it is blocked to get $\theta(k+1)$ along the boundary of $G_i=0$ by the inequality constraint G_j (See Fig. 2). Then, we must get the $(k+1)$th solution along the boundary of the blocking constraint G_j, so we need the initial value of $\mu_j(k)$ which makes $\mathbf{zh}+\mathbf{zg}_j\mu_j=0$ hold. That is why $\mu_j(k)$ must be adjusted at the switching point of $\theta(k)$ that does not satisfy $\mathbf{z}(\theta(k))\,\mathbf{h}(\theta(k))=0$. This adjustment is not required at a switching point that satisfies $\mathbf{z}(\theta(k))\,\mathbf{h}(\theta(k))=0$. The condition for obtaining $\mu_j(k)$ is that $\mathbf{z}(\theta(k))\,\mathbf{g}_j(\theta(k))$ is linearly dependent on $\mathbf{z}(\theta(k))\,\mathbf{h}(\theta(k))$. This condition is always met at the point that satisfies $\mathbf{z}\,\mathbf{h}=0$ or when DOR is one. The proposed algorithm fails to give a solution when the following four incidents occur simultaneously: (1) DOR is greater than one. (2) multiple inequality constraints become active at a point $\theta(k)$. (3) $\mathbf{z}\,\mathbf{h}\neq 0$ at $\theta(k)$. (4) There is no $\mathbf{z}\,\mathbf{g}_j$ which is linearly dependent on $\mathbf{z}\,\mathbf{h}$ at $\theta(k)$.

Two functional forms are considered in the proposed method: One is the objective function to be optimized, the other is the constraint function to be satisfied. The inequality constraints form an infeasible region in the joint space. Far away from this infeasible region, all the Lagrange multipliers are zero and the redundancy is fully utilized to optimize the objective function $H(\theta)$, *i.e.*, $H(\theta)$ is assigned the highest priority. On the boundary of the infeasible region, the selected active inequality constraint G_j maintains the highest priority until the corresponding Lagrange multiplier μ_j becomes zero or any other inequality constraint becomes active. Meanwhile, we are free to choose the functional form for an additional task. In other words, the two functional forms, $H(\theta)$ and $G_j(\theta)$, can be interchangeably assigned to an additional task. For example, an obstacle avoidance task can be performed either by maximizing the distance between the obstacle and the manipulator, or by maintaining the distance not smaller than a certain threshold value.

3. Simulation Results

In this section, the performance of the proposed method will be evaluated through two simulations. Consider a 3R planar manipulator shown in Fig.3. The forward kinematic equation is given as (25).

$$x = l_1 c_1 + l_2 c_{12} + l_3 c_{123} \tag{25}$$

$$y = l_1 s_1 + l_2 s_{12} + l_3 s_{123}$$

where l_i represents the length of the ith link and $c_i=\cos(\theta_i)$, $s_{ij}=\sin(\theta_i+\theta_j)$ and so on. The Jacobian matrix is obtained as

$$\mathbf{J} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \end{bmatrix} \quad (26)$$

The basic motion task is to trace a circle trajectory, which is represented as (27).

$$x(t) = -1.0\cos(2\pi t) + 3.0 \quad (27)$$
$$y(t) = -1.0\sin(2\pi t)$$

Let an additional task be to maximize the manipulability measure[9], $H(\boldsymbol{\theta}) = \sqrt{\det(\mathbf{J}\mathbf{J}^T)}$, for avoiding singularities. The initial configuration is $[-1.521, 1.951, 1.353]^T$ which corresponds to $[2.0, 0.0]^T$ in task space. Let the other additional task be an obstacle avoidance. Generally, the minimum distance between workspace obstacles and a manipulator can be expressed as a function of $\boldsymbol{\theta}$. For simplicity, suppose that a point of interest on the manipulator is the third joint. Then, it is required that the third joint avoid a workspace obstacle. Let the obstacle be a circle with radius of $\sqrt{2}$ and center of $[4.3, -3.0]^T$. We need to solve the following optimization problem:

$$\begin{aligned} &\textit{maximize}\ H(\boldsymbol{\theta}) = \sqrt{\det(\mathbf{J}\mathbf{J}^T)} && (28) \\ &\textit{subject to}\ \ \mathbf{x} - \mathbf{f}(\boldsymbol{\theta}) = 0 \\ &-(l_1 c_1 + l_2 c_{12} - 4.3)^2 - (l_1 s_1 + l_2 s_{12} + 3.0)^2 + (\sqrt{2})^2 \le 0 \end{aligned}$$

The results are depicted in Figs.4 and 5. For comparison, the results for the case that does not consider obstacle avoidance are also depicted. In Fig.4, the manipulator configurations and the forbidden region , the interior of the circles , are shown. It is observed that the manipulator successfully performs additional tasks as well as the basic motion task. Getting a solution with disregarding the inequality constraint is blocked by G at $t = 0.36$, which means that the third joint of the manipulator contacts with the boundary of the forbidden region at this time and the solution of the next time step must be obtained along that boundary ,*i.e.*, the higher priority must be given to the obstacle avoidance task. The change of the Lagrange multiplier is depicted in Fig.5(b). The Lagrange multiplier becomes zero at $t = 0.54$, meaning $\mathbf{z}\,\mathbf{h} = 0$ at that configuration. At this point, it is possible to get a solution of the next time step that satisfies $\mathbf{z}\,\mathbf{h} = 0$, without being blocked by the inequality constraint. This means that the active inequality constraint can be relaxed, so a solution can be obtained with disregarding G. In other words, the higher priority is given to singularity avoidance, at this time step. The proposed method maximizes $H(\boldsymbol{\theta})$ with the inequality constraint being satisfied, *i.e.*, the obstacle is avoided. In Fig.5(c), the manipulability measure for both cases are depicted.

In the second simulation, the basic task is same as before and the manipulability is again chosen to be maximized for the task of singularity avoidance. The obtained solution trajectory performing only these two tasks is the same one that is obtained by the Extended Jacobian method, and is shown in Fig.6(a) as a curve. Suppose that the requirements for another additional tasks are expressed by the following inequality constraints:

$$G_1(\boldsymbol{\theta}) = -(\theta_2 - 1.7)^2 - (\theta_3 - 1.3)^2 + 0.2^2 \le 0 \quad (29)$$
$$G_2(\boldsymbol{\theta}) = -(\theta_2 - 1.64)^2 - (\theta_3 - 1.17)^2 + 0.1^2 \le 0$$

The boundary of each inequality constraint is also shown in Fig.6(a). The region enclosed by these two inequality constraints is forbidden for the manipulator. Note that DOR is not 'sufficient' , *i.e.*, DOR is smaller than the number of the additional tasks. Then the redundancy resolution problem can be stated as follows:

$$\begin{aligned} &\textit{maximize}\ H(\boldsymbol{\theta}) = \sqrt{\det(\mathbf{J}\mathbf{J}^T)} && (30) \\ &\textit{subject to}\ \ \mathbf{x} - \mathbf{f}(\boldsymbol{\theta}) = 0 \\ &\qquad -(\theta_2 - 1.7)^2 - (\theta_3 - 1.3)^2 + 0.2^2 \le 0 \\ &\qquad -(\theta_2 - 1.64)^2 - (\theta_3 - 1.17)^2 + 0.1^2 \le 0 \end{aligned}$$

The solution of (30) is shown in Figs.6(b)-(d). In Fig.6(b), the solution trajectory is plotted in θ_2–θ_3 space. The upper right end point corresponds to the task space point of $[2.0, 0.0]^T$, which is the desired position for the end-effector at $t = 0$ and $t = 1.0$. At $t = 0.5$, the end-effector must be located at $[4.0, 0.0]^T$ and the lower left end point corresponds to that position. In Figs.6(b) and (c), we can find the fact that the proposed method has a cyclic property. Figs.6(c) and (d) show the time history of the solution. For comparison, the solution to the problem without considering inequality constraints is also plotted in Fig.6(c). The solutions are recursively obtained by using (20) until it is blocked by the inequality constraint G_1 at $t = 0.09$. From this time, solutions are obtained along the boundary of $G_1 = 0$ by using (23). The change of the corresponding Lagrange multiplier μ_1 is shown in Fig.6(d). At about $t = 0.25$, a blocking inequality constraint G_2 is encountered. At that time instance, the proposed method relaxes the inequality constraint G_1 , therefore making μ_1 be zero , and modifies the value of μ_2 at the corresponding configuration. Then, it gets solutions along the boundary of $G_2 = 0$ until $t = 0.33$, at which μ_2 becomes zero so that the corresponding configuration becomes another switching point. At this point, $\mathbf{z}\,\mathbf{h} = 0$ and the method again relaxes the inequality constraint G_2 and thereafter gets solutions that maintain $\mathbf{z}\,\mathbf{h} = 0$ until any blocking inequality constraint is encountered.

4. Conclusion

In this paper, we showed that a redundancy resolution problem with multiple criteria could be transformed into a local equality and inequality constrained optimization problem, and proposed a method to solve it at velocity level. In the proposed scheme, equality constraints are imposed for the basic motion task. One additional task is performed by optimizing an objective function while the other additional tasks are performed by satisfying a set of inequality constraints. Since the proposed method uses the differential kinematic relationship and the matrices involved are always square, the computation can be efficiently done. The method is efficient especially when the number of additional tasks are larger than DOR. It also gives a way to systematically assign the priorities between the additional tasks by using the Lagrange multipliers. In addition, the method has a cyclic property which is crucial in cyclic tasks. Some limitations are also examined along with these benefits of the proposed method.

References

[1] D. E. Whitney, "Resolved motion rate control of manipulators and human prosthesis," *IEEE Trans. on Man-Machine Systems*, vol. 10, no. 2, pp. 47-53, 1969.

[2] D. N. Nenchev, "Redundancy resolution through local optimization:a review," *J. of Robotic Systems*, vol. 6, no. 6, pp. 769-798, 1989.

[3] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," *in Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 722-728, 1985.

[4] J. Baillieul, "A constraint oriented approach to inverse problems for kinematically redundant manipulators," *in Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1827-1833, 1987.

[5] P. H. Chang, "A closed-form solution for inverse kinematics of robot manipulators with redundancy," *IEEE J. of Robotics and Automation*, vol. 3, no. 5, pp. 393-403, 1987.

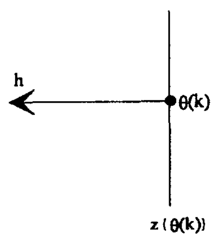
[6] H. Seraji, "Configuration control of redundant manipulators:theory and implementation," *IEEE J. of Robotics and Automation*, vol. 5, no. 4, pp. 472-490, 1989.

[7] H. Seraji and R. Colbaugh, "Improved configuration control for redundant robots," *J. of Robotic Systems*, vol. 7, no. 6, pp. 898-928, 1990.

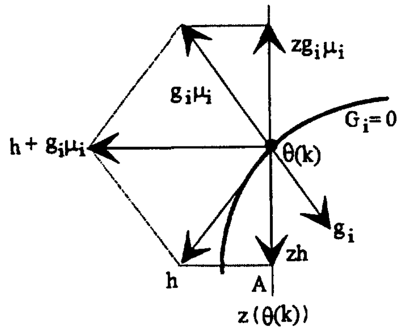
[8] C. R. Carignan, "Trajectory optimization for kinematically redundant arms," *J. of Robotic Systems*, vol. 8, no. 2, pp. 221-248, 1991.

[9] T. Yoshikawa, "Manipulability and redundancy control of robotic mechanisms," *in Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1004-1009, 1985.

[10] D. Luenberger, *Optimization by vector space methods*, Wiley & Sons, 1969.



(a)



(b)

Fig. 1. Relationship between gradient vectors and the null space of J at $\theta(k)$. (a) $zh = 0$. (b) $zh + zg_i\mu_i = 0$.

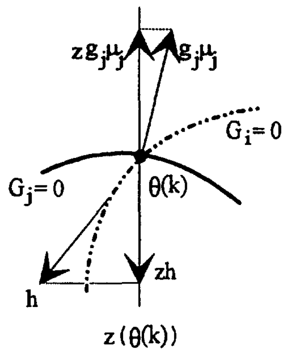


Fig. 2. Adjustment of the Lagrange multiplier μ_j at $\theta(k)$ when getting $\theta(k+1)$ is blocked by the inequality constraint $G_j \le 0$.

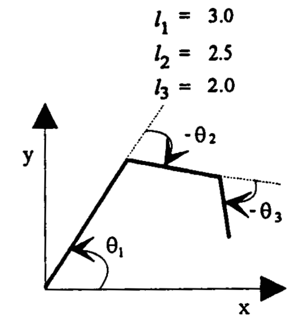
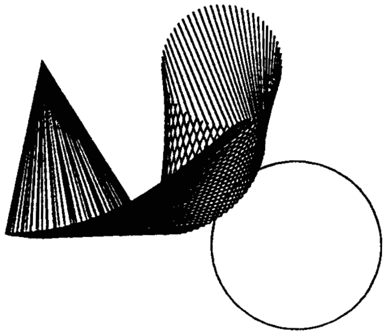
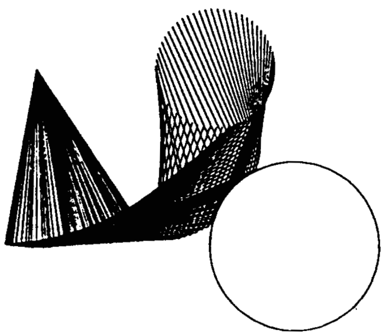


Fig. 3. A planar 3R manipulator.

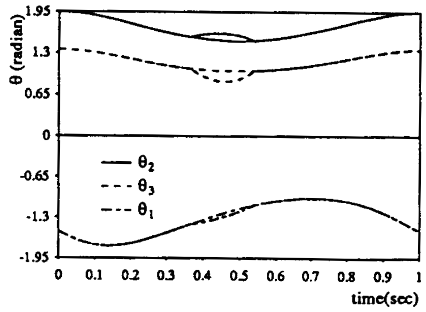


(a)

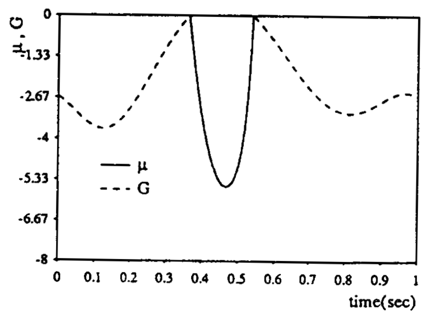


(b)

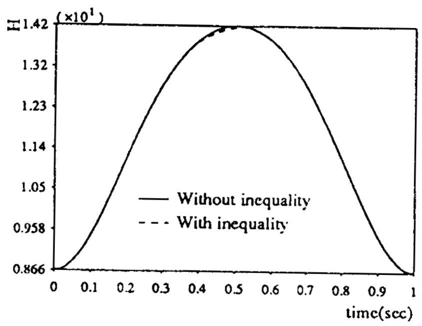
Fig. 4. The configurations of the manipulator when an obstacle is in task space. (a) Without considering obstacle. (b) With considering obstacle.



(a)

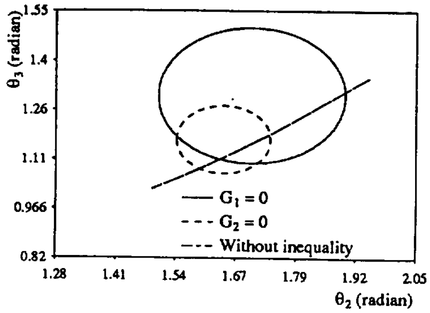


(b)

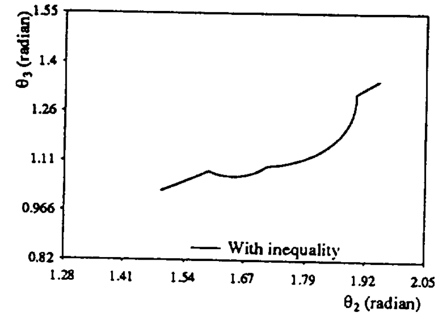


(c)

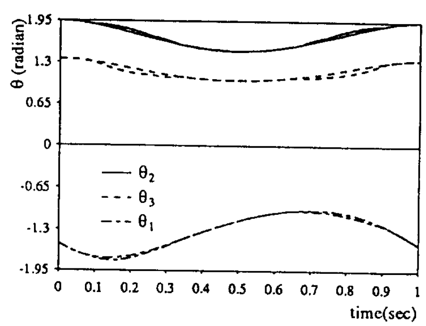
Fig. 5. Results for the first simulation. (a) Joint angles for two cases. (b) The Lagrange multiplier μ and the inequality constraint G. (c) Manipulability measures for two cases.



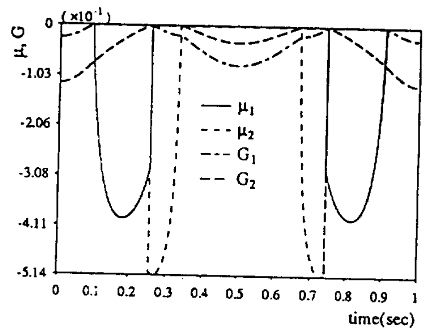
(a)



(b)



(c)



(d)

Fig. 6. Results for the second simulation. (a) Inequality constraints and the solution trajectory obtained without considering inequality constraints. (b) The solution trajectory obtained with considering inequality constraints. (c) The solutions obtained with/without considering inequality constraints. (d) The Lagrange multipliers and the inequality constraints.