An Inequality Constraints Based Method for Inverse Kinematics of Redundant Manipulators

Young Whee Sung, Dong Kwon Cho, Myung Jin Chung
Dept. of ElectricalEngineering,
Korea Advanced Institute of Science and Technology,
373-1 Kusong-dong, Yusong-gu, Taejon, 305-701, KOREA

Abstract : In addition to a basic motion task, redundant manipulators can achieve some additional tasks by optimizing proper performance criteria. Some of performance criteria can be transformed to inequality constraints. So the redundancy resolving problem can be reformulated as a local optimization problem with equality constraints for the end effector and inequality constraints for some performance criteria. In this article, we propose a method for solving the inverse kinematics of a manipulator with redundancy using the Kuhn-Tucker theorem to incorporate inequality constraints. With proper choice of inequality constraints, the proposed method gives a way of optimizing multiple criteria in redundant manipulators.

1. Introduction

Because of the dexterity and versatility, redundant manipulator has been a major research topic in robotics over the past two decades[1-10]. A manipulator is defined to be kinematically redundant if it possesses more degrees of freedom than are required to place the end-effector at a desired location. Let $x \in R^m$, $\theta \in R^n$ be the task space and joint space variables respectively, then the relation between these two variables is defined as (1) by a kinematic funtion $f : R^m \rightarrow R^n$.

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

At velocity level

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

where $J = \dfrac{\partial f(\theta)}{\partial \theta}$ is an $m \times n$ matrix

In redundant manipulators, the degrees of redundancy $r(=n-m)$ is greater than zero and makes it possible to accomplish additional tasks such as singularity avoidance, obstacle avoidance, joint-limit avoidance,etc.,with the basic end-effector motion task.

Fundamental approaches for resolving the redundancy are classified into two methods. The one is the local optimization method and the other is the global one. Local optimization methods determine a joint space trajectory by locally optimizing performance criteria and are adequate for real-time implementation. One way of locally resolving the redundancy is to transform a redundant system into a nonredundant one by imposing $n-m$ constraints. Those constraints with (1) fully specify the system which can be solved at position level[6] or at velocity level[4]. While the basic task is expressed as equality constraints of (1), some additional tasks can be represented as inequality constraints[8,9,10].

In this article, we propose a local optimization method to resolve the redundancy which can incorporate inequality constraints. The proposed method can deal with a multi-creteria problem and systematically assign the priority to additional tasks when all of those tasks can not be met simultaneouly. It also is computationally efficient and gives a cyclic solution.

2. The Inverse Kinematics Problem

2.1 Problem Formulation

For brevity, let us assume one degree of redundancy and two additional tasks. So the system is overdetermined. Let $H(\theta)$ be a scalor function to be optimized for one additional task and suppose the requirement for the other additional task can be represented as an inequality form of $G(\theta) \le 0$. Here $H(\theta)$ and $G(\theta)$ have continuous first-order partial derivatives with respect to θ . Then the redundancy resolution problem can be transformed into (3).

$$\begin{aligned} &\textit{maximize } H(\theta) \\ &s.t. \quad x = f(\theta) \\ &\qquad G(\theta) \le 0 \end{aligned} \tag{3}$$

The Lagrangian function $L(\theta, \lambda)$ is given as (4).

$$L(\theta) = H(\theta) + \lambda^T(x - f(\theta)) + \mu G(\theta) \tag{4}$$

where λ is an m dimensional vector and μ is a scalar.

The necessary condition for optimality is obtained as (5) (8) by the Kuhn-Tucker theorem.

$$\frac{\partial L}{\partial \theta} = h^T - \lambda^T J + \mu g^T = 0 \tag{5}$$

$$\frac{\partial L}{\partial \lambda} = x - f(\theta) = 0 \tag{6}$$

$$\mu G(\theta) = 0 \tag{7}$$

$$\mu \le 0 \tag{8}$$

where $h \triangleq (\frac{\partial H}{\partial \theta})^T$ and $g \triangleq (\frac{\partial G}{\partial \theta})^T$ are m dimensional vectors.

By transposing (5), we have (9).

$$h + \mu g = J^T \lambda \tag{9}$$

Premultiplying (9) by z, the $1 \times n$ basis vector of the null space of J, we can get (10)

$$z(h + \mu g) = 0 \tag{10}$$

The equations (6), (7), (8) and (10) will be denoted as Local Constrained Optimization problem (LCO) from now on. To slove LCO, We need to consider the following two cases.

Case I : $\mu = 0$

LCO is reduced to (11).

$$x = f(\theta) \tag{11}$$

$$z h = 0$$

The solution of eq (11) is the same as the task space expansion method except that it must satisfy the inequality constraint.

Case II : $G(\theta) = 0$

The LCO is reduced to (12).

$$x = f(\theta) \tag{12}$$

$$z h + \mu z g = 0$$

$$G(\theta) = 0$$

It must be solved for unknown θ and μ and its solution must satisfy eq (8).

2.2 Proposed Algorithm

We can solve LCO directly according to the above two cases. But we propose new algorithm to solve it at velocity level for computational efficiency. In each case, the differential relationship is given as follows.

Case I :

$$\dot{x} = J \dot{\theta} \tag{13}$$

$$\frac{\partial zh}{\partial \theta} \dot{\theta} = 0$$

$$\dot{\mu} = 0$$

or in matrix form

$$\begin{bmatrix} \dot{x} \\ 0 \end{bmatrix} = \begin{bmatrix} J \\ \frac{\partial zh}{\partial \theta} \end{bmatrix} \dot{\theta} \tag{14}$$

$$\triangleq \begin{bmatrix} J \\ J_c \end{bmatrix} \dot{\theta} \triangleq J_e \dot{\theta}$$

Case II :

$$\dot{x} = J \dot{\theta} \tag{15}$$

$$\frac{\partial (z h + \mu z g)}{\partial \theta} \dot{\theta} + z g \dot{\mu} = 0$$

$$g \dot{\theta} = 0$$

or in matrix form

$$\begin{bmatrix} \dot{x} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} J & 0 \\ \frac{\partial (zh + \mu zg)}{\partial \theta} & zg \\ g & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\mu} \end{bmatrix} \tag{16}$$

$$\triangleq \begin{bmatrix} J_1 \\ J_2 \\ J_3 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\mu} \end{bmatrix} \triangleq J_{ee} \begin{bmatrix} \dot{\theta} \\ \dot{\mu} \end{bmatrix}$$

Now we define the solutions of LCO which satisfy $\mu = 0$ and $G(\theta) = 0$ as switching points. Assume J_e and J_{ee} are not singular along the trajectory. In Case I with $G(\theta) < 0$, the inequality constraint is inactive and has no effect on the solution. So the solution is equal to that of (3) without the inequality constraint. So we can obtain the solution by the extended jacobian method along the task space trajectory until the inequality constraint become active. Provided it is not active through the whole trajectory, the acquired solution is exactly the same as that of the extended Jacobian method. If not, we must get to the switching points.

At switching points, all constraints are satisfied and the inequality constraint is active. So we must decide which joint-space trajectory to follow. In this case, we have two choices: Case I or Case II. Choosing Case I means we are to follow the trajectory which maximizes a criterion $H(\theta)$. But the acquired solution must satisfy the inequality constraint. If not, we need to choose Case II, the meaning of which is that two additional tasks can not be met simultaneouly, so we must get a solution along the boundary of the inequality constraint. This also means that we assign higher priority to the task of the inequality constraint. We may meet another switching point when μ becomes zero and have a chance to release the inequality constraint to find the solution that make both additional tasks be achieved.

The following procedure summarizes the proposed algorithm.

Step 1: Find the solution at the initial time $t = t_0$

If $\mu(t_0) = 0$ and $G(\theta(t_0)) < 0$, goto step 2.

else if $\mu(t_0) = G(\theta(t_0)) = 0$, goto step 3.

else if $\mu(t_0) < 0$ and $G(\theta(t_0)) = 0$, goto step 4.

Step 2: Proceed with (14)

If $G(\theta(t_i)) < 0$, accept the solution and repeat this step.

else if $G(\theta(t_i)) = 0$, goto step 3.

Step 3: Proceed with (14)

If $G(\theta(t_i)) \le 0$, accept the solution and

if $G(\theta(t_i)) = 0$, repeat this step.

else if $G(\theta(t_i)) < 0$, goto step 4.

else if $G(\theta(t_i)) > 0$, drop the solution and goto step 4.

Step 4: Proceed with (15)

If $\mu(t_i) < 0$, repeat this step.

else if $\mu(t_i) = 0$, goto step 3.

With the proposed algorithm we can solve the inverse kinematic problem with computational efficiency even when the number of additional tasks is greater than the degree of redundancy.

3. Several Considerations

3.1 Lagrange multiplier

To satisfy the condition of (7), $G(\theta)$ should be less than or equal to zero when $\mu(t_1)$ equals zero, which means the inequality constraint is inactive. At this point, say $\theta(t_1) \triangleq \theta_1$, $z(\theta_1) h(\theta_1) = 0$. In other word, $z(\theta_1)$ and $h(\theta_1)$ are orthogonal as shown in Fig 1(a). This means that there is no self-motion of the manipulator and θ_1 is the maximum point for $H(\theta)$ that satisfies all constraints. When μ is less than zero, $G(\theta)$ must be zero to be active. At this point, $z(\theta_1)(h(\theta_1) + \mu g(\theta_1)) = 0$. So $z(\theta_1)$ and $h + \mu g(\theta_1)$ are orthogonal and μ plays a key role as a scaling factor that makes (10) be hold. In this case, there can be a direction on $z(\theta_1)$ that further increases the value of $H(\theta)$ but this will also increase $G(\theta)$ to break the inequality constraint. So the optimal solution stops at θ_1 though it it not the maximum point for $H(\theta)$. This situation is depicted in Fig 1(b). However, the basic task represented by the equality constraints is achieved exactly in any case.

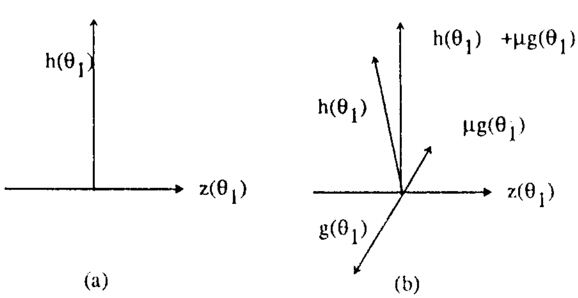

Fig. 1. Realtionship between h, g and z

3.2 Objective Functions

$H(\theta)$ and $G(\theta)$ can be any scalar functions of θ with continuous first partial derivatives with respect to θ. If the task is to minimize $H(\theta)$, we only need to use $\mu \ge 0$ instead of eq (8). So we can use any function as $H(\theta)$ as long as optimizing it represents the desired performance.

Many tasks such as obstacle avoidance, joint limit avoidance, have the form of inequality constraints. Furthermore, optimizing a objective function can also be transformed into an inequality constraint. For example, the main objective of maximizing the manipulabiliby measure[3], $H(\theta) = \sqrt{J J^T}$ is to avoid singularity at which the value of it becomes zero. So the objective can be satisfied by the inequality constraint of the form of (16).

$$G(\theta) = -H(\theta) + d \le 0 \qquad (16)$$

where d is a user-defined threshold value.

Therefore, we have flexibility in selecting $H(\theta)$ and $G(\theta)$ according to the desired additional tasks.

3.3 Algorithmic Singularity and Extensions

When there is no effect of inequality constraint as in the Case I, the algorithmic singularity is the same as that of the extended Jacobian method. In Case II, it occurs when $z(\theta) g(\theta) = 0$ or when g is a linear combination of the rows of J.

The proposed algorithm can be extended without difficulty to multiple degree of redundancy and/or multiple inequality constraints. However, it does not seem to be efficient because of the fact that the number of cases to be tested is 2^l in case of l inequality constraints.

4. Simulation and Discussion

In this section, the capability of the proposed method to achieve multiple additional tasks as well as the basic task will be shown by a simple simulation. Consider a 3R planar manipulator shown in Fig2.

The initial joint value is $[-1.521, 1.951, 1.353]^T$ which corresponds to the task space point of $[2.0, 0.0]^T$. The basic task is for the end-effector to trace a circle with radius is 1 aft the origin at $[3.0, 0.0]^T$ and expressed in (17)

$$x(t) = -1.0\cos(2\pi t) + 3.0 \qquad (17)$$
$$y(t) = -1.0\sin(2\pi t)$$

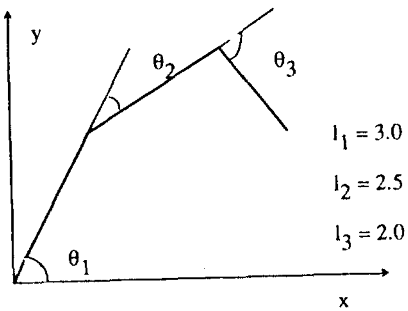

Fig. 2. A 3R planar manipulator

If an additional task is to maximize the manipulability measure, $H(\theta) = \sqrt{J J^T}$, for avoiding singularities, the solution obtained by the extended Jacobian method is shown in Fig 3.

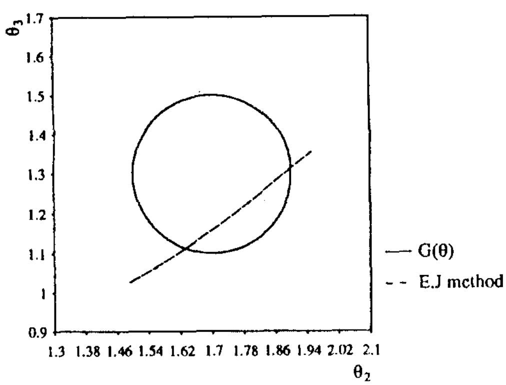

Fig. 3. Inequality constraints and the solution obtained by the extended Jacobian method

Suppose equation (18) defines the other additional task. The feasible region of it is outside the circle in Fig 3.

$$G(\theta) = -(\theta_2 - 1.7)^2 - (\theta_3 - 1.3)^2 + 0.2^2 \le 0 \qquad (18)$$

Then the redundancy resolution problem can be stated as follows.

$$\text{maximize} \ \sqrt{J J^T} \qquad (19)$$

$$s.t \quad x - f(\theta) = 0$$

$$-(\theta_2 - 1.7)^2 - (\theta_3 - 1.3)^2 + 0.2^2 \le 0$$

Note that the degree of the redundancy is not 'sufficient' for the additional tasks.

The solution of (19) is shown in Fig4. In this figure, we can observe that the proposed method makes the manipulator successfully achieve all the tasks whenever possible. When both additional tasks can not be met simultaneouly, the task of inequality constraint has higher priority and the other task is achieved in the optimal manner. For comparison, solutions acquired by the above two methods are shown in Fig 5. The solutions are same when the inequality constraint is inactive. At about t = 0.1 and 0.7, it becomes active and thereafter the proposed method gets the solution along the boundary of it until the method can release the inequality constraint to satisfy all tasks at about t = 0.3 and 0.9. Fig 5 also show that the solution has the cyclic property. Corresponding parameter values are shown in Fig 6. While the inequality constraint is active, the associated multiplier μ satisfies eq (8). Especially at switching points (t = 0.1, 0.3, 0.7and 0.9), μ falls to zero, which means we may be able to release the inequality constraint.

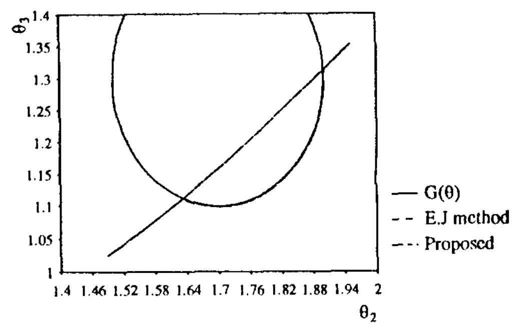

Fig. 4. Solution trajectories obtained by the two methods in θ_2–θ_3 space

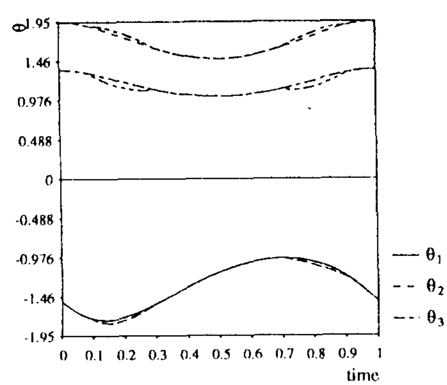

Fig. 5. Solution trajectories obtained by the two methods in θ– time space

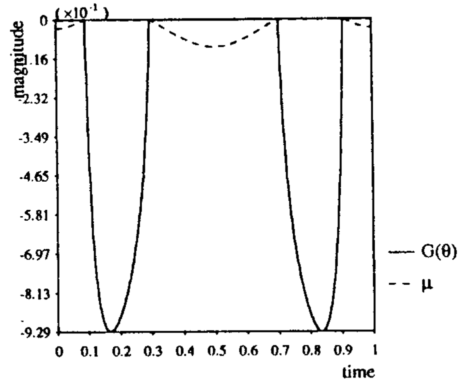

Fig. 6. Time histories of the value of inequality and Lagrangian multiplier μ

5. Conclusions

In this article, we showed that the redundancy resolution problem with multiple criteria can be transformed to a local constrained optimization problem. and we proposed a method to solve the inverse kinematic problem of redundant manipulator.

It is computationally efficient and uses a differential kinematic relation, like the extended Jacobian method. It can directly incorporate inequality constraints and has the cyclic property.

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] Y. Nakamura and H. Hanafusa, "Task priority based redundancy control of robot manipulators," *IEEE J. of Robotics Research,*, vol. 6, no. 2, pp. 3-14, 1987.

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

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

[5] 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.

[6] 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.

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

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

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

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

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