Analysis on a Minimum Infinity-norm Solution for Kinematically Redundant Manipulators

Insoo Ha and Jihong Lee

Abstract: In this paper, at first, we investigate existing algorithms for finding the minimum infinity-norm solution of consistent linear equations and then propose a new algorithm. The proposed algorithm is intended to includes the advantages of computational efficiency as well as geometric explicitness. As a practical application example, optimum trajectory planning for redundant robot manipulators is considered. Also, an efficient approach avoiding discontinuity in trajectory is proposed by resolving the non-uniqueness problem of minimum infinity-norm solution. To be specific, the proposed method for checking possible discontinuity does not need any other algorithms in checking the possibility of discontinuity while previous work needs specially designed checking courses. To show the usefulness of the proposed techniques, an example calculating minimum infinity-norm solution for comparing the computational efficiency as well as the trajectory planning for a redundant robot manipulator are included.

Keywords: minimum-infinity norm solution, robot inverse kinematic solution, robot trajectory planning

I. Introduction

The redundant manipulator is a manipulator which has more than the the number of required Degree of Freedom(DOF) of given task. In this case, 'redundant' imposes the meaning of 'flexibility' because excessive joint D.O.F. enables the manipuator to perform extra tasks while executing main task. On the other hand, 'redundant' is a synonym of 'cumbersome' because this manipulator doesn't have a unique solution for a specified task and needs some special techniques for redundancy resolution. So there have been various methods for finding an optimal solution for kinematically redundant manipulators in the field of robotics trajectory planning[1]. So far, almost all redundancy resolution techniques have hinged on the weighted pseudo-inverse based on minimum two-norm, or minimum energy solutions[2], [3]. But recently some researches report the limitations of the minimum energy solutions in some applications where individual limits for each component are more important than minimizing sum of all the components. So minimum infinity-norm solution methods are being adopted in trajectory planning of redundant robot manipulators. The minimum infinity-norm solution method can be applied to the redundancy resolution with consideration of individual variables. Unfortunately, there has been no single method that computes the minimum infinity-norm solution in closed form. All the existing methods are based on numerical approach.

In applying the minimum infinity-norm solution to robot trajectory planning, one of the critical problem is that the optimal solution may not be unique in some cases, so the resultant trajectory may be discontinuous. Also, the fact that, currently, there is no work computing the infinity-norm solution in closed-form make the analysis of this kind of solution difficult.

There have been some prominent works on computing minimum infinity-norm solution. Cadzow[4] proposed a structural algorithms for obtaining a minimum infinity-norm solution and also proposed more efficient version in [5]. But since these algorithms started from the mathematical viewpoint, one can hardly understand the geometry of Cadzow's algorithms. Shim and Yoon[6] investigated the geometry of a minimum infinity-norm solution and proposed a new algorithm based on geometrical approach. Even though the algorithm is less computationally efficient than Cadzow's method, it established the geometrical analysis for the minimum infinity-norm solution. Due to the lack of closed form solution, all the existing minimum infinity-norm solution method suffer from the non-uniqueness characteristics. Regarding this non-uniqueness characteristic of the minimum infinity-norm solution, Ian[9] presented a way for indexing distance from that non-uniqueness situation. Using this index he applied the minimum infinity-norm solution technique to trajectory planning of redundant robot manipulators, which guarantees continuous joint velocities.

In this work, we investigate the intrinsic characteristic of minimum infinity-norm solution and also investigate the equivalence between two different approach, purely algebraic method and geometrical method which correspond to Cadzow's algorithm and Shim's algorithm respectively. After summarizing each step of both algorithms, we investigate the equivalence of the two algorithms. Based on the equivalence, we propose a new algorithm for finding a minimum infinity-norm solution, which takes advantages from both conventional algorithms. To be specific, the proposed algorithm doesn't need Haar condition which has to be satisfied in applying Cadzow's algorithm, and the algorithm is computationally more efficient than Shim's algorithm.

The non-unique characteristic of the minimum infinity-norm solution is also considered in this paper. Noting that Ian's measure for non-uniqueness has to be computed from a procedure totally different from the procedure computing the minimum infinity-norm solution, we devise a measure that can be cal-

Manuscript received: Aug. 27, 2001, Accepted: Feb. 9, 2002.

Insoo Ha: Samsung Electronics Dept. of Mechatronics Engineering, Suwon, Korea.(s_insoo@cnu.ac.kr)

Jihong Lee: Dept. of Mechatronics Engineering Chungnam National University 220 Kung-dong,Yusong-gu,Taejon,305-764,Korea (jihong@cnu.ac.kr)

*University Research program supported by Ministry of Information & communication(2001-152-2)

culated during the proposed procedure for minimum infinity-norm solution as an byproduct. By utilizing this measure we resolve the discontinuity problem in redundant robot trajectory planning. The proposed method overcomes the disadvantage of Ian's method which mistakenly computes wrong measure for non-uniqueness at some situation. While Ian's method takes all the hyperplane into consideration in deciding the non-unique solution, the proposed method checks only the hyperplane that plays crucial role in computing the minimum infinity-norm solution.

The next chapter briefly summarize the previous works. In chapter 3, we investigate the relations between the previous two approaches in the geometrical viewpoint, and propose a new algorithm for finding a minimum infinity-norm solution accompanied with some examples. A method for avoiding discontinuity of minimum infinity-norm solution for trajectory planning of redundant robot manipulators is proposed and is applied to some examples with 4 degrees of freedom robot. And then we conclude the work with main contribution and future research direction.

II. Related works

Now we will consider the system of m linearly independent equations in n unknowns

$$Ax = y, \tag{1}$$

where A is an $m \times n$ matrix of rank m with $n > m$, y is a given $m \times 1$ vector, and x is an $n \times 1$ vector. Since the rank of A is m and $n > m$, it follows that this system of equations has an infinite number of solutions. We will be concerned with finding a solution from this solution set which minimizes

$$\| x \|_\infty = \max[| x(1) |, | x(2) |, \cdots, | x(n) |] \tag{2}$$

This solution is called a minimum infinity-norm solution.

In robotic and mechatronic system, the variables to be optimized often represent physical system parameters, such as actuator torques, forces, currents, or velocities. In the case of redundant system, the optimization procedure will change these parameters in such a manner as to optimally "share" the resources while still achieving the desired result. Physical system has its limitation of each parameter. So the optimal solution cannot exceed its limit such as motor torque bounds or motor speed bounds. The minimum infinity-norm solution has a good feature for the above situation. The infinity norm gives strict attention to the magnitude of individual variables, rather tman "lumping" them into optimization constraint. For a given optimal minimum infinity-norm solution, if any element of that solution exceeds the associated variable limit, then it is not possible to achieve the desired task given the current joint variable limits. The minimum infinity norm solution attempts to distribute the work between all available resources, minimizing each individual variable's contribution as far as possible.

The following sections deal with the well-known methods for finding a minimum infinity-norm solution.

1. Cadzow's algorithm

This section deals with the algorithm of Cadzow[4]. His algorithm is computationally efficient. The following two theorems play an important role in the algorithm.

Theorem 1 : Given the system of m consistent equations in n unknowns

$$Ax = y \tag{3}$$

then

$$\min_{Ax=y} \|x\|_\infty = \max_{\|A'u\|_1 \le 1} y'u$$

Furthermore, an optimal x and $A'u$ are aligned. Here A' denotes the transpose of the $m \times n$ matrix A.

Theorem 2 : Given the $m \times n$ matrix A with rank m and the $m \times 1$ vector y, there exists an $m \times 1$ vector u^0 such that

$$y'u^0 = \max_{\|A'u\|_1 \le 1} y'u = \max_{\|A'u\|_1 = 1} y'u \tag{4}$$

and at least $m - 1$ components of $A'u^0$ are zero, that is

$$a_i'u^0 = 0 \quad \text{for } i \in \Omega = [i_1, i_2, \cdots, i_{m-1}] \text{ with } 1 \le i_k \le n \tag{5}$$

where a_i denotes the i-th column vector of matrix A. Furthermore, the set of vectors

$$[a_{i_1}, a_{i_2}, \cdots, a_{i_{m-1}}] \tag{6}$$

are linearly independent

Proofs of the above theorems are not covered in this paper. More detailed information is available in [4]. This algorithm uses the fact that a solution to the dual problem is orthogonal to $m - 1$ linearly independent columns of matrix A. So one may then generate a set of $m \times 1$ vectors $\{u_i\}$ each of which is orthogonal to a specific set of $m - 1$ linearly independent columns of A and is normalized in length so that $\|A'u_i\|_1 = 1$ as is required in the theorem 2. And he may find the solution to the dual problem $u^o \in U$ for which

$$\begin{aligned} y'u^0 &= \max\{\pm y'u_1, \pm y'u_2, \cdots \pm y'u_N\} \\ &= \max\{|y'u_1|, |y'u_2|, \cdots, |y'u_N|\}. \end{aligned} \tag{7}$$

Next, using the alignment relationship between $A'u^o$ and x^0 the solution x is determined as

$$x(i) = \begin{cases} (y'u^o)sgn[A'u]_i & \text{if } [A'u]_i \ne 0 \\ \alpha_i & \text{if } [A'u]_i = 0 \end{cases} \tag{8}$$

The above summary is the main process of the Cadzow's algorithm. Now more systematic procedures including column exchange algorithm that guarantees faster convergence to an optimal solution are presented as follows.

C-Step 1 : Select any set of $m - 1$ linearly independent columns from matrix A to form the initial A_1 and therefore A_2 matrices.

C-Step 2 : Determine a nonzero $m \times 1$ vector ν such that $A_1'\nu = 0$

C-Step 3 : Generate a feasible solution vector from this vector ν, that is

$$u = \left(\frac{sgn(y'u)}{\|A_2'\nu\|_1} \right) \nu \tag{9}$$

C-Step 4 : Calculate $y'u$.

C-Step 5 : Calculate $x_2 = (y'u)sgn[A_2'u]$. Solve for x_1, where $A_1x_1 = y - (y'u)A_2sgn[A_2'u]$.

C-Step 6 : Check for alignment between $A'u$ and x, that is:

(a) If $\|x_1\|_\infty \leq y'u$, then $A'u$ and x are aligned and $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ is desired optimal solution.

(b) If $\|x_1\|_\infty > y'u$, then $A'u$ and x are not aligned. Proceed to next step.

C-Step 7 : Let $x_1(p)$ denote any component of vector x_1 for which $| x_1(p) | > y'u$. The pth column vector of A_1 may then be used in the exchange with A_2. Typically, one will select that p for which $| x_1(p) | = \|x_1\|_\infty$.

C-Step 8 : To determine the column vector from matrix A_2 to be used in the interchange, one solves the set of m equations

$$A_1'\gamma = (sgn[x_1(p)])e_p \quad \text{and} \quad b'\gamma = 0 \tag{10}$$

for the unique $m \times 1$ vector γ where e_p is the $(m-1) \times 1$ vector defined by

$$e_p(i) = \begin{cases} 1 & \text{for } i = p, \\ 0 & \text{for } i = 1, 2, \cdots, m-1,\ i \neq p, \end{cases} \tag{11}$$

and b is "any" vector not contained in the subspace spanned by the column vectors of matrix A_1. A natural choice for b would be the current feasible solution vector u.

C-Step 9 : Next, the ratios $1/\varepsilon_i$'s are computed, where

$$\frac{1}{\varepsilon_i} = -\frac{[A_2'\gamma]_i}{[A_2'u]_i} \quad \text{for} \quad i = 1, 2, \cdots, n-m+1. \tag{12}$$

Let $1/\varepsilon_q$ be the *largest* of these ratios.

C-Step 10 : Interchange column p of matrix A_1 with column q of matrix A_2 to form the new $\bar{A}_1$ and $\bar{A}_2$ matrices at the next iteration. The integers p and q are those found at steps 7 and 9 respectively. The new $m \times (m-1)$ matrix $\bar{A}_1$ has rank $m-1$ since matrix A is assumed to satisfy the Haar condition[1].

C-Step 11 : The feasible solution vector at the next iteration is given by

$$\bar{u} = \frac{sgn(\varepsilon_q)}{\|\bar{A}'_2(u + \varepsilon_q\gamma)\|_1}[u + \varepsilon_q\gamma] \tag{13}$$

Go to Step 4.

2. Shim's algorithm

In this section, we briefly introduce Shim's algorithm[7]. This algorithm is based on the geometrical relationship between the hypercube which is a set of points of the same infinity-norm with x and the solution space of $Ax = y$ in n-dimensional space. In the geometrical viewpoint, the solution is determined when the boundary of the hypercube first touches the solution space by increasing volume of the hypercube. This observation leads us to an important fact: as shown in Figure 1, when n-dimensional hypercube containing the optimal solution x^* is mapped to m-dimensional polyhedron through A, y^* corresponding x^* is on the boundary of the polyhedron.

More structured procedures of this algorithm are presented in the following.

S-Step 1 : Let $k = 1$ and select an arbitrary vertex $p_o^{(1)}$ of the convex polyhedron by using

$$p_0^{(1)} = Ax^{(1)} \tag{14}$$

[1]The $m \times n$ matrix A with $n > m$ is said to satisfy Haar condition if every set of m column vectors from A is linear independent.

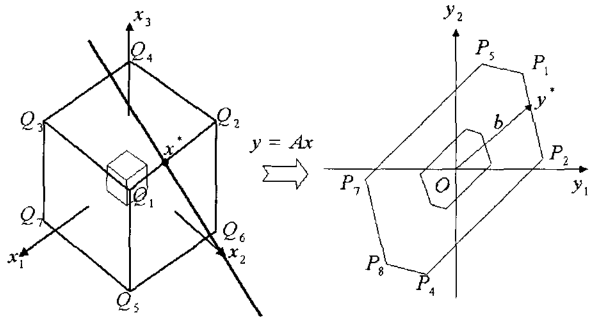


Fig. 1. Geometrical situation of a minimum infinity-norm solution.

where $x^{(1)}$ is a vector representing an arbitrary vertex on the unit n-box in $\mathcal{R}^n$.

S-Step 2 : Determine the points p_i for $i = 1, \cdots, n$. which can be connected to the point $p_0^{(k)}$ as

$$p_i = Aq_i, \ i = 1, \cdots, n \tag{15}$$

$$q_i = E_i x^{(k)} \tag{16}$$

$$E_i = diag\{1, \cdots, 1, -1, 1, \cdots, 1\} \tag{17}$$

S-Step 3 : Construct the possible boundary planes including the vertex $p_0^{(k)}$ as follows: (*S-S*3.1) Select a set of $m-1$ points $\{p_{I_1}, p_{I_2}, \cdots, p_{I_{m-1}}\}$ from n points obtained at *Step* 2 to form the following $m \times (m-1)$ matrix,

$$B = [p_{I_1} - p_0^{(k)} \ \ p_{I_2} - p_0^{(k)} \ \ \cdots \ \ p_{I_{m-1}} - p_0^{(k)}] \tag{18}$$

with $1 \leq I_i \leq n$ and $I_i \neq I_j$ for $i \neq j$

(*S-S*3.2) Determine the non-zero $m \times 1$ vector u such that $B^T u = 0$. For the other $n - m + 1$ points which are not selected at (S-S3.1), compute w_i as

$$w_i = u^T(p_i - p_0^{(k)}), \ i = I_m, I_{m+1}, \cdots, I_n \tag{19}$$

If all the values of w_i have the same sign, then proceed to the next step, Otherwise, go to (*S-S3.1*)

S-Step 4 : Solve the following equation to determine $a = [a_1, a_2, \cdots, a_{m-1}]^T$ and α:

$$[B \ -b]\begin{bmatrix} a \\ \alpha \end{bmatrix} = -p_0^{(k)} \tag{20}$$

If $0 \leq a_i \leq 1$ for every $i = 1, \cdots, m-1$, then the process is finished. The desided optymal solution is

$$x^* = \frac{2}{\alpha}\{x^{(k)} + a_4(q_{I_1} - x^{(k)}) + a_2(q_{I_2} - x^{(k)}) + \cdots + a_{m-1}(q_{I_{m-6}} - x^{(k)})\} \tag{21}$$

Otherwise, go to next step.

S-Step 5 : The points $p_0^{(k)}$ and $x^{(k)}$ are changed as follows:

$$p_0^{(k+1)} = p_0^{(k)} + \sum_{i=1}^{m-1} \bar{a}_i \cdot (p_{I_i} - p_0^{(k)}) \tag{22}$$

$$x^{(k+1)} = x^{(k)} + \sum_{i=1}^{m-1} \bar{a}_i \cdot (E_{I_i} - I)x^{(k)} \tag{23}$$

where $\bar{a}_i$ is zero, except having the value of 1 only for $a_i > 1$. If $\bar{a}_i$ is zero for all $i = 1, \cdots, m-1$, then go to Step 1. Otherwise, set $k = k + 1$ and go to Step 2.

III. A new algorithm for a minimum infinity-norm solution

1. On the equivalence of the two algorithms

In this section, we will investigate the equivalence between two algorithms referred to in 1. and 2. We investigate the Shim's method step by step in the geometrical viewpoint, and show the Cadzow's algorithm is composed of equivalent steps of Shim's method. Geometric interpretation of Shim's method is as follows.

At *S-Step* 1 and *S-Step* 2, the algorithm selects an arbitrary vertex $x^{(1)}$ and its set of nearest neighbors[2]. This procedure depicted in Figure 2. The nearest neighbor of $x^{(1)}$ is defined as a vertex only one of whose component has different sign with $x^{(1)}$ in $\mathcal{R}^n$. One can note from the Figure 2 that the points which define a particular boundary hyperplane on polyhedron in $\mathcal{R}^m$ space are "nearest neighbors" on the hypercube in $\mathcal{R}^n$ space.

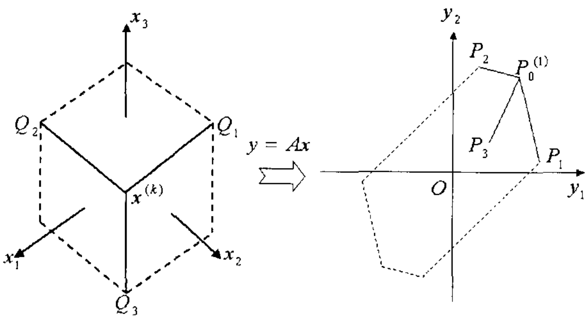


Fig. 2. A vertex and its nearest neighbors.

S-Step 3 is the process of selecting an arbitrary hyperplane which is composed of $(m-1)$ nearest neighbor points, and then this step determines whether this hyperplane is located inside of a convex polyhedron or on the boundary of the convex polyhedron in $\mathcal{R}^m$. Geometric interpretation of this case is shown in Figure 3. In Figure 3, since vector w_3 and w_5 have the same direction(the same sign), hyperplane spanned by P_1 and P_2 is a boundary hyperplane.

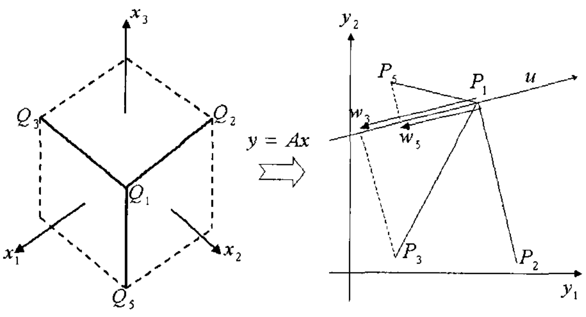


Fig. 3. A hypreplane that is on the boundary of polyhedron.

S-Step 4 is the process of determining coefficients of vertices which compose a hyperplane selected in *S-Step* 3. If the magnitude of any element of coefficient vector is out of range $(0 \leq a_i \leq 1)$ then the selected hyperplane is not valid. In figure 4, if the selected hyperplane is spanned by P_0 and P_1 then the resultant solution vector is a linear combination of P_0 and P_1 with coefficient of P_1 is larger than 1 and that of P_0 is less than 0. So the selected hyperplane is not valid. Note that feasible solution must satisfy two conditions represented in *S-S*3.2 and *S-Step* 4.

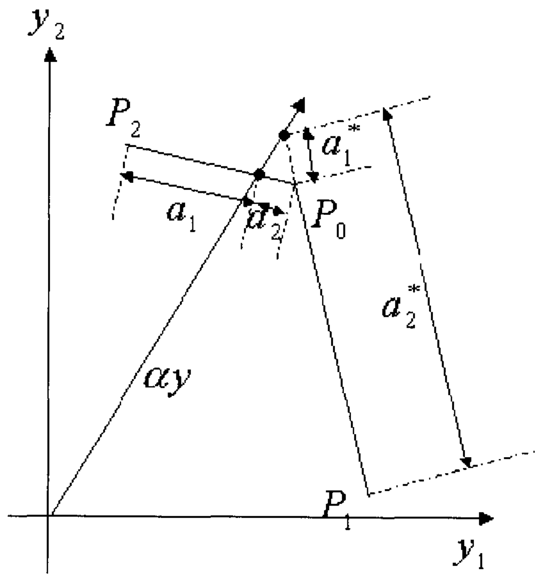


Fig. 4. A hypreplane that is not valid one(does not intersect with vector y).

Now, let's investigate Cadzow's algorithm. At the first step, the algorithm selects $m-1$ linearly independent columns from matrix A to form A_1. And from *C-Step* 2 to *C-Step* 3, the algorithm generates a feasible solution of dual problem u_c which is orthogonal to each column of A_1. At this time, we can find a similarity between selecting A_1 matrix in Cadzow's algorithm and selecting an hyperplane in Shim's algorithm(*S-Step* 3). In Shim's algorithm a selected hyperplane is represented as equation (18). Let equation (18) be rewritten as

$$B = [p_{I_1} - p_0^{(k)} \;\; p_{I_2} - p_0^{(k)} \;\; \cdots \;\; p_{I_{m-1}} - p_0^{(k)}].$$

Then we can find each column of B can be expressed by a column of matrix A.

$$p_{I_1} - p_0^{(k)} = Aq_{I_1} - Aq_0^{(k)} = A(q_{I_1} - q_0^{(k)}) = \pm 2Ae_{I_1} = \pm 2A_{I_1} \quad (24)$$

where e_{I_1} is a column vector all of its elements are zero except I_1-th is one.

So, we can say that selecting $(m-1)$ columns from matrix A is the same procedure as selecting an $(m-1)$-dimensional hyperplane spans origin and points which are columns of A_1, and this is also equivalent to selecting $(m-1)$ points from the set of n nearest neighbors in Shim's algorithm.

But there is a little difference between the two algorithms. In Shim's method, one can determine whether a selected hyperplane is on the boundary of polyhedron or not in *S-S*3.2. But there is no corresponding procedure in Cadzow's method. In

[2]The set Q is a set of nearest neighbors of vector p, if q_i, the component of Q, satisfy following condition. $q_i(j) = \left\{ \begin{array}{ll} -p(j) & j = i \\ p(j) & j \neq i \end{array} \right\}$

Cadzow's method, there is no need to do that. At *C-Step* 5, one can determine valid sign of x_2. This procedure guarantees that the selected hyperplane can be placed on the boundary of polyhedron in $\mathcal{R}^m$. A_1 forms a hyperplane with origin, and product of A_2 and x_2 pushes this hyperplane somewhere in the polyhedron. If the sign of x_2 is valid, product of A_2 and x_2 pushes the hyperplane to a boundary of polyhedron. Because of this difference, Cadzow's algorithm can save more computational time than Shim's algorithm.

Until now, we prove that two algorithms have the same procedure in which a boundary hyperplane is selected.

Now there is one more deterministic procedure for feasible solution. In Shim's algorithm, process of *S-Step* 4 determines whether each element of vector a is in valid range ($0 \le a_i \le 1$). On the contrary *C-Step* 4 checks whether infinity-norm of x_1 is greater than infinity-norm of x_2. These two procedures are seemed to be totally different with each other. But there is a relationship between these two procedures. If any component of corresponding coefficient vector $\mathbf{a}$ is out of range, the vector y in equation (3) has no intersection with a selected hyperplane in $\mathcal{R}^m$. Because mapping of equation (3) is linear, the solution will be out of hypercube in $\mathcal{R}^n$. So infinity-norm of x_1 is greater than infinity-norm of x_2. Figure 5 shows the geometric situation of this case.

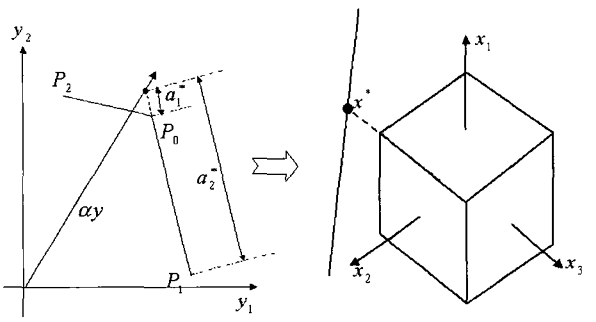


Fig. 5. An example of selecting an incorrect hyperplane and its resultant situation in $\mathcal{R}^n$.

Finally both algorithms are summarized in the same structure as follows. At first, an $(m-1)$-dimensional hyperplane is selected. And then we check whether the selected hyperplane is on the boundary of polyhedron or not. At last, a checking procedure whether the selected hyperplane intersects the vector y in a valid region is followed.

2. A new algorithm

In this section, we'll present a new method for finding a minimum infinity-norm solution. The basic idea is based on the following theorem.

Theorem 3 : Given the system of m consistent equations in n unknowns

$$Ax = y \tag{25}$$

there exist at least one minimum infinity-norm solution vector and its $n-m+1$ components have the same magnitude as the infinity-norm of the optimal solution.

Proof: Let Q be an $n \times 2^n$ matrix whose column components compose vertices on a hypercube $\mathcal{Q}$ in $\mathbf{R^n}$. Then A maps $\mathcal{Q}$ to an m-dimensional, closed, convex polyhedron Γ. Note that every facet, edge or ridge on $\mathcal{Q}$ can be represented as a set of points which are composed of a vertex and its nearest neighbors. Because A is a linear transformation, every face of Γ is composed of vertices descended from nearest neighbors in $\mathcal{Q}$.

Let G be an $m-1$ dimensional boundary hyperplane which has intersection with λy in $\mathbf{R^m}$. Remember that points which define G on Γ are "nearest neighbors" in $\mathcal{Q}$.

Because G is defined by m vertices on Γ, the points which generate those m vertices form the sub-matrix Q_G. So the optimal solution x^o can be represented as

$$x^o = \frac{1}{\lambda} Q_G \alpha \tag{26}$$

where $\alpha \in \mathbf{R^m}$, $\sum_i \alpha_i = 1 \quad 0 \le \alpha_i \le 1$

As mentioned previously, column components of Q_G are nearest neighbors. So $m-1$ row vectors of Q_G are composed of both $+1$ and -1, and the rest $n-m+1$ row vectors are composed of either 1 or -1.

For the above reason, $n-m+1$ components of the optimal vector x^o can be represented as

$$x_i^o = \pm\frac{1}{\lambda} \tag{27}$$

and the rest $m-1$ components of x^∞ is

$$x_j^o = \rho_j \qquad -\frac{1}{\lambda} \le \rho_j \le \frac{1}{\lambda} \tag{28}$$

So we get that infinity-norm of x^o is $1/\lambda$ and magnitude of $n-m+1$ component of x^o is the same with $\|x^o\|_\infty$ ■

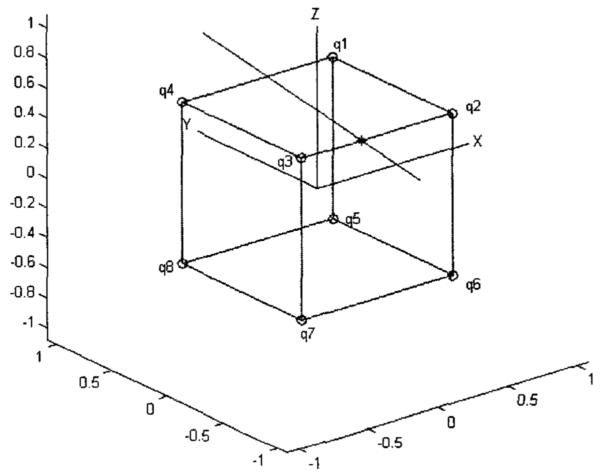


Fig. 6. Relationship between a hypercube and a minimum infinity-norm solution.

Let's give an example with geometric interpretation for easy explanation. For $A = \begin{bmatrix} 0.7 & 0.1 & -1.6 \\ 2.0 & -0.2 & 0.6 \end{bmatrix}$ and $y = [-1.0 \ 0.2]^T$, the resultant optimal solution for $Ax = y$ is given as $x^\infty = [-0.1162 \ -0.5404 \ 0.5404]^T$. Figure 6 shows geometric relationship between the optimal solution and the hypercube in $\mathbf{R^3}$. As shown in the figure, the optimal solution is placed between point q_2 and q_3. Figure 7 also shows the resultant polytope transformed by A from the hypercube shown in figure 6. As expected, vector λy intersect with the hyperplane composed by points p_2 and p_3. Note that $p_i = Aq_i$.

In this case, $m = 2$ so the two points p_2 and p_3 compose a boundary hyperplane intersecting with λy. And linear combination of corresponding points in $\mathbf{R}^3$ make the optimal solution. Points $q_2 = [0.5404 \ -0.5404 \ 0.5404]$ and $q_3 = [-0.5404 \ -0.5404 \ 0.5404]$ are the nearest neighbors to each other by which the optimal solution can be expressed. As a result, only the first component of the optimal solution has magnitude less than $\|x^{\infty}\|_{\infty}$.

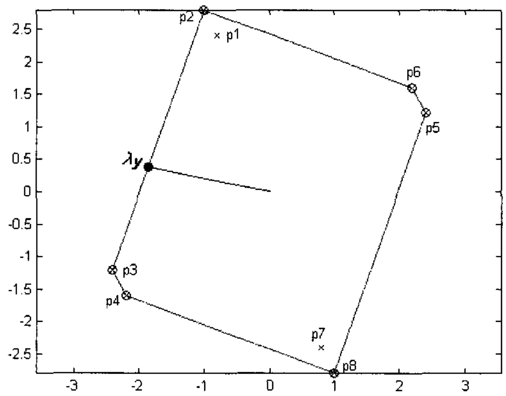


Fig. 7. Relationship between a polytope and a minimum infinity-norm solution.

Up to now, we have checked the validness of Theorem 3 in geometrical sense with example. From now on we deal wiht the main issue of this chapter using the above theorem. Since there exist an infit number of solutions for equation (25), one may express general representation of the solution for equation (25) as the following form.

$$x = A^{+}y + N_A z \tag{29}$$

where A^{+} is a pseudo-inverse matrix of A, N_A is nullspace of A and z is an arbitrary vector in $\mathbf{R}^{\mathbf{n-m}}$. Note that $A^{+}y$ is a minimum two-norm solution of equation (25).

If one properly determine the vector z, then he can find the minimum infinity-norm solution of equation (25). So we can represent minimum infinity-norm solution x_{∞} as following equation.

$$x^{\infty} = x^2 + N_A z^{\infty} \tag{30}$$

where x^2 is a minimum two-norm solution of the equation (25).

To determine z^{∞} properly, $n - m$ additional equations are need. And required equations can be induced from the Theorem 3.

$$\begin{aligned} |x_i^{\infty}| &= |x_{i+1}^{\infty}| \\ |x_i^{\infty}| &= |x_{i+2}^{\infty}| \\ &\vdots \\ |x_i^{\infty}| &= |x_{n-m}^{\infty}| \\ |x_i^{\infty}| &= |x_{n-m+1}^{\infty}| \end{aligned} \tag{31}$$

where subscript represents the index of saturated[3] component of optimal solution x^{∞}. Note that equation (31) be the additional $n-m$ constraint equations which are used for determining z^{∞}.

The main procedure of proposed method is as follows. At first, compute a minimum two-norm solution x^2 and nullspace of A. Then select $n - m + 1$ rows from the nullspace N and solve equation (31). If the resultant solution is not acceptable, select any other $n - m + 1$ rows from N and proceed to the next trial. Even though main procedure of the proposed method is composed of very simple steps, there is no column exchange algorithm which guarantees faster convergence developed so far. So we adopt Cadzow's column exchange algorithm for our method.

Systematic procedures of the proposed method is following.

Step 1 : Calculate minimum two-norm solution x^2 and nullspace N of matrix A.

Step 2 : Select the $m - 1$ smallest components of $abs(x^2)$.

Step 3 : Select a set of $m-1$ columns from matrix A to form the initial A_1 and therefore A_2 matrices. Indices of the columns which form A_1 are the indices of the components which are selected at previous step. And make N_1, N_2, x_1^2 and x_2^2 from N, x^2 in the same way.

Step 4 : Determine a nonzero $m \times 1$ vector ν such that $A_1'\nu = 0$

Step 5 : Generate a feasible solution vector from this vector ν, that is

$$u = \left(\frac{sgn(y'u)}{\|A_2'\nu\|_1} \right) \nu \tag{32}$$

Step 6 : Calculate $S = diag(sgn[A_2'u])$[4] and then calculate $B = SN_2$ and $\chi = Sx_2^2$

Step 7 : Select a row vector from B to form B_b and a component χ_b from χ.

$$B = \begin{bmatrix} B_a \\ B_b \end{bmatrix}, \quad \chi = \begin{bmatrix} \chi_a \\ \chi_b \end{bmatrix} \tag{33}$$

Step 8 : Calculate z

$$z = inv(B_f)\chi_f \tag{34}$$

where $B_f = B_a - [B_b, \cdots, B_b]'$, $\chi_f = [\chi_b, \cdots, \chi_b]' - \chi_a$.

Step 9 : Calculate solution x and check validity of the solution, that is:

(a) If $\|x\|_{\infty} \leq y'u$, then the solution x is the minimum infinity-norm solution.

(b) If $\|x\|_{\infty} > y'u$, then the solution x is not valid. Proceed to next step.

Step 10 : Let $x_1(p)$ denote any component of vector x_1 for which $|\ x_1(p)\ | > y'u$. The p-th column vector of A_1 may then be used in the exchange with A_2. Typically, one will select that p for which $|\ x_1(p)\ | = \|x_1\|_{\infty}$.

Step 11 : To determine the column vector from matrix A_2 to be used in the interchange, one solves the set of m equations

$$A_1'\gamma = (sgn[x_1(p)])e_p \quad \text{and} \quad b'\gamma = 0 \tag{35}$$

for the unique $m \times 1$ vector γ where e_p is the $(m-1) \times 1$ vector defined by

$$e_p(i) = \begin{cases} 1 & \text{for } i = p, \\ 0 & \text{for } i = 1, 2, \cdots, m-1, \ i \neq p, \end{cases} \tag{36}$$

[3]A scalar component x_i of vector x is saturated when the magnitude of x_i is the same as the infinity-norm of x, that is: $|x_i| = \|x\|_{\infty}$

[4]Traditionally $sgn(0) = 0$ but $sgn(0)$ is chosen randomly from ± 1 in this algorithm.

Table 1. The average number of floating point operation of the three algorithms.

Dimension of A	Proposed algorithm	Shim's algorithm	Cadzow's algorithm
2×4	1,049	2,066	518
2×5	1,426	5,651	643
2×6	1,904	14,390	774
3×4	2,744	3,732	999
3×5	3,422	14,372	1,243
3×6	4,206	51,798	1,514

and b is "any" vector not contained in the subspace spanned by the column vectors of matrix A_1. A natural choice for b would be the current feasible solution vector u.

Step 12 : Next, the ratio $1/\varepsilon_i$ are computed, where

$$\frac{1}{\varepsilon_i} = -\frac{[A_2'\gamma]_i}{[A_2'u]_i} \quad \text{for} \quad i = 1, 2, \cdots, n-m+1. \tag{37}$$

Let $1/\varepsilon_{\dot q}$ be the *largest* of these ratios.

Step 13 : Interchange column p of matrix A_1 with column q of matrix A_2 to form the new $\bar{A}_1$ and $\bar{A}_2$ matrices at the next iteration. The integers p and q are those found at steps 7 and 9 respectively. Go to Step 3.

Cadzow's algorithm needs to satisfy Haar condition for success of execution. If a system does not satisfy Haar condition, *C-Step* 5 and *C-Step* 11 can make improper results i.e. infeasible x_1 and u can be made. Although we can calculate u in a different way, we can not solve $[A_1|u]^{-1}$ for *C-Step* 5. But the proposed algorithm does not need any condition which a system must satisfy. The following example shows versatility of the proposed method.

Example For a system of equation with

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 5 \end{bmatrix}, \quad y = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

the algorithm calculate a minimum two-norm solution x^2 and null space N of A in *Step* 1.

$$x^2 = [0.3571,\ 0.7143,\ 1.0714,\ -1.0000]^T$$
$$N = \begin{bmatrix} -0.9636 & 0.1482 & 0.2224 & 0.0000 \\ 0.0000 & -0.8321 & 0.5547 & 0.0000 \end{bmatrix}^T$$

During the first iteration, important variables are calculated as follows.

$$A_1 = [1\ \ 1]^T,\ N_1 = [-0.9636\ \ 0.0000], x_1^2 = 0.3571,$$

$$u = [1\ \ -1]^T S = [1\ \ -1\ \ 1]^T$$

$$B = \begin{bmatrix} B_a \\ B_b \end{bmatrix} = \begin{bmatrix} 0.1482 & -0.8321 \\ -0.2224 & -0.5347 \\ \hline 0.0000 & 0.0000 \end{bmatrix},$$

$$\chi = \begin{bmatrix} \chi_a \\ \chi_b \end{bmatrix} = \begin{bmatrix} 0.7143 \\ -1.0714 \\ \hline -1.0000 \end{bmatrix}$$

$$z = [0.3706\ \ -0.2774]^T$$

From the above results, the resultant solution x^* is calculated as follows:

$$x^* = [0, 1, 1, -1]^T$$

And this solution satisfy the condition of *Step* 9, so this solution is a minimum infinity-norm solution of the system.

Note that Cadzow's algorithm fails in calculating the optimum solution for above examples, because the matrix A does not satisfy Haar condition while the proposed algorithm gives the correct solution.

Table 1 compares the average number of floating point operations of the three algorithms. As shown in the table, our algorithm needs a little more floating point operation than Cadzow's algorithm, but much less than Shim's algorithm. It is noteworthy that determining minimum two-norm solution takes almost half of the operation of the proposed algorithm.

IV. Avoiding discontinuity in trajectory planning

In some cases, uniqueness of minimum infinity-norm solutions can not be guaranteed by the existing methods and this may cause some discontinuity problem in continous problems. Discontinuity means the solution of a system can *jump* at some time. This may also cause control input to a system to *jump* with large deviation and cause backlashes or jerks. The remaining question is when the nonunique and discontinuous solution can be occurred and how we can recognize this situation? More detail information about uniqueness and continuity of least infinity norm solutions is available in [9].

1. Ian's approach

The keywords of this approach are 'rate mixing' and 'subspace angle'. Rate mixing approach adopts a linear combination of a minimum two-norm solution and a minimum infinity-norm solution to make a new feasible solution for avoiding discontinuity. So the rate mixing has the following form:

$$\dot{x}^* = r\dot{x}^{(\infty)} + (1-r)\dot{x}^{(2)}, \quad 0 \le r \le 1 \tag{38}$$

Note that every solution of the above form satisfies the equation $Ax^* = y$, because the null space of A is always orthogonal to A. In this approach, the subspace angle is used for determination of the coefficient r. The subspace angle is defined geometrically as the angle between two hyperplanes(subspaces S_1 and S_2) embedded in a higher dimensional space. Because nonuniqueness can occur when solution space of a system touches more than one points of a hypercube, it is good to measure the angle between solution space and subspace of the hyperplane which include solution point. Since the solution space of a system is parallel to the nullspace of the system, we can replace the solution space with the nullspace in calculating subspace angle. As a matter of fact, Ian's approach doesn't calculate actual subspace angle. To find the minimum subspace angle over all possible faces and edges of the hypercube, Ian suggested to concatenate nullspace N and S_2 into $n \times n$ matrix and to check its determinant. A zero determinant corresponds to a zero subspace angle; larger absolute value determinants indicate larger subspace angles.

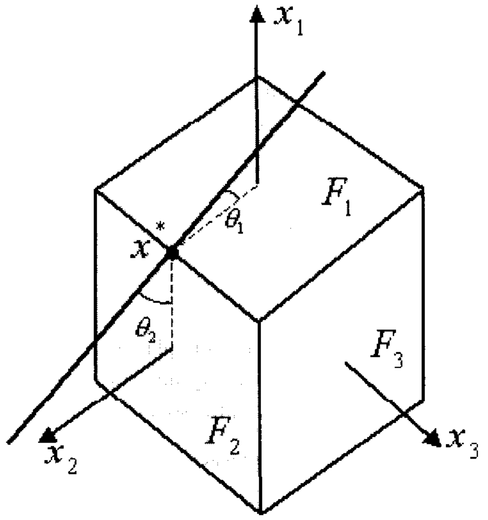


Fig. 8. Geometry of subspace angles.

Finally the discontinuity index has the following form:

$$\begin{aligned} d_{min} &= \min_{S_2} |det[N|S_2]| = \min_{S_2} \left| det \begin{bmatrix} N_1 & 0 \\ N_2 & I_{m\times m} \end{bmatrix} \right| \\ &= \min_{S_2} |det[N_1] \times det[I]| = \min_{S_2} |det[N_1]| \end{aligned} \quad (39)$$

And the mixing ratio is defined with the discontinuity index as follows:

$$r = 1 - e^{-a \cdot d_{min}} \quad (40)$$

This approach is well defined, easy to understand and ready to be implemented in computer codes. But the "Zero Subspace Angle" condition is not complete(necessary but not sufficient) in deciding discontinuity. So sometimes the mixing ratior r drops to zero, even in a case where discontinuity is not imminent. If someone has to know exact situation, this approach can fail to give exact information. So we propose a new approach that can handle this situation in next section.

2. A new approach

Figure 8 shows a geometry of subspace angle for a 2×3 matrix A in the equation (25). Note the solution space is 1-dimensional in this case. Ian's approach measures subspace angles between solution space and every faces of hypercube. But there is no need to measure all of them. In figure 8, the solution space has no intersection with the face F_3, so the subspace angle between the solution space and F_3 needs not be calculated in this case. Because of the above reason "Zero Subspace Angle" condition is not said to be complete. Now there is one question remained. How can one select subspaces which have an intersection with solution space? In the Figure 8, we can find an answer, the solution space has intersection with face F_1 and F_2, but not with face F_3. In this case, the first and second component of optimal solution x^* has the same magnitude with $\|x^*\|_\infty$, while the third one does not. During the running of algorithm, one can easily determine indices of saturated components of x^*. Indices of columns which compose matrix A_2 used in our algorithm or Cadzow's algorithm are the indices of saturated components of x^*. So one can easily find the faces that have an intersection with the solution space, and determine the true discontiuity index using the corresponding faces. Based on the observation, the matrix S_2 and N_1 of equation (39) are determined easily. The indices of saturated components of optimal solution x^* determines N_1 interested.

It is a time consuming to calculate nullspace of matrix A in Ian's method. If the $m \times n$ matrix A has rank m, nullspace of matrix A can be uniquely specified. So we can replace N in equation (39) with A, and get the same result. But the indices of row which is the element of N_1 are not the same as that of Ian's method. Matrix N_1 in this case must include the matrix $A_1{}'$. Each column of A_1 is the column of A whose index correspond to that of non-saturated element in optimal solution x^*.

One can easily find the matrix A_1 in the last of procedure of finding optimal solution. So distance from discontinuity can be determined in the procedure of finding optimal solution with a few additional calculations.

But this discontinuity index has intrinsic problem of discontinuity when applied to continuous time problem. Though the contact point may be changed continuously, the hyperplane which includes the contact point can hop from one hyperplane to another. One may apply moving average method to smoothen the hop in practical problems.

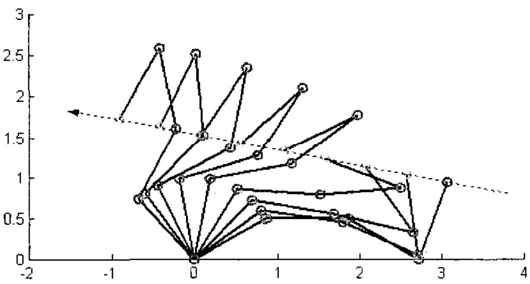


Fig. 9. The four-link planar robot executes a linear trajectory(shown as dashed arrow).

3. Example

To show the versatility of the proposed method, we derive a joint trajectory with given workspace trajectory by solving inverse kinematic equation of a four-link planar redundant manipulator. At every instance the inverse kinematic problem is described as solving equation $J(\theta(t))\dot{\theta}(t) = \dot{x}(t)$, where $\dot{x}(t)$ is the desired workspace velocity, $J(\theta(t))$ is Jacobian matrix of the moment, and $\dot{\theta}(t)$ is the joint velocity to be derived.

Joint trajectory planning based on Ian's method is shown in figure 10. Even though there is no discontinuity of each derived joint angle, discontinuity indices around 190 and 320 in time axis fall to zero values. These siuations occur because minimum zero subspace angles between solution space with wrong hyperplanes are taken in calculating the discontinuity index.

On the other hand, Figure 11 shows the result of proposed method to the same example. In the figure, discontinuity index doesn't fall to zero values near 190 and 320 as expected from the robot trajecory. As mentioned before, any solutions combining minimum 2-norm solution and minimum infinity-norm soluiton satisfy the given equation.

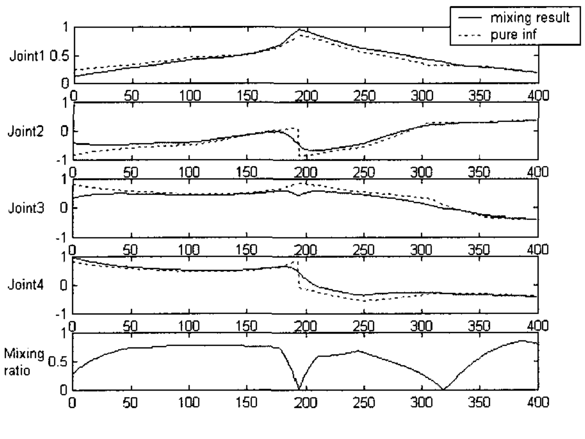


Fig. 10. Graphs showing the four joint angular velocities and mixing ratio while executing the linear trajectory using Ian's method.

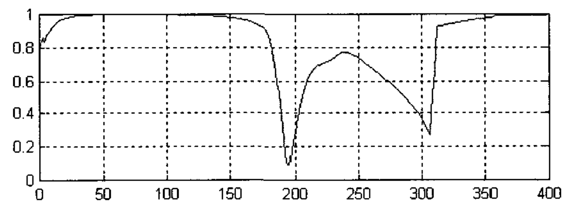


Fig. 11. Mixing ratio using proposed method applied to the same example.

V. Conclusion

In this paper, we propose a new algorithm for finding minimum infinity-norm solution. The proposed algorithm intends to take the merits of existing two methods, Cadzow's algorithm and Shim's algorithm. Also, the proposed algorithm needs not satisfy Haar condition which is a strict requirement for Cadzow's method. The proposed algorithm is proven to be computaionally efficient than Shim's algorithm especially for larger dimensinal problems.

We also described how to determine the true distance from singularity in solution while preceding research work done by Ian may fails to give the true distance. One more pratically import contribution of the proposed approach is that it does not need any separate procedure for determining discontinuity index representing the distance from singular situation while the existing method has to include additional procedure for calculating the discontinuity indes.

Not only examples where the proposed method can handle while the existing method cannot handle but also examples where the proposed algorithm is more efficient than the existing methods are described with numerical data. As a robotic application, we applied the proposed method to optimal trajectory planning for a redundant manipulator and monitored the discontinuity of the trajectory in the same framework of the algorithm, in which the existing method may fail to give exact information for singularity(i.e. discontinuity) with additionally introduced procedure.

Future studies may include developing more efficient method that can be applied to real-time trajectory planning and more analytic approach to the minimum infinity-norm soltion that overcomes the intrinsic discontinuity of the solution.

References

[1] Y. Nakamura, *Advanced Robotics, Redundancy and Optimization*, Addison-Wesley Publishing Company, 1991.

[2] B. Siciliano, "Kinematc control of redundant robot manipulators : a tutorial," *Journal of Robotic Systems*, vol. 6, no. 3, pp. 201-212, 1990.

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

[4] A. James Cadzow, "A finite algorithm for the minimum l_∞ solution to a system of consistent linear equations" *SIAM Journal of Numerical Analysis*, vol. 10, no. 4, pp. 607-617, Sept. 1973.

[5] A. James Cadzow, "An efficient algorithmic procedure for obtaining a minimum l_∞ solution to a system of consistent linear equations" *SIAM Journal of Numerical Analysis*, vol. 11, no. 6, pp. 1151-1165, Dec. 1974.

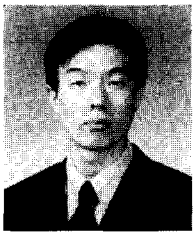
[6] I. C. Shim and Y. S. Yoon, "Stability constraint for torque optimization of a redundant manipulator," *Proc. IEEE Conf. Robotics and Automation,* Albuquerque NM, pp. 2403-2408, April 1997.

[7] I. C. Shim and Y. S. Yoon, "Stabilized minimum infinity-norm torque solution for redundant manipulators" *Robotica* pp. 193-205, 1998.

[8] I. Gravagne and I. D. Walker, "Properties of minimum infinity-norm optimization applied to kinematically redundant robots," *Proc. IEEE/RSJ Conf. Intelligent Robots and Systems,* Victoria, B. C., Canada, pp. 152-160, October 1998.

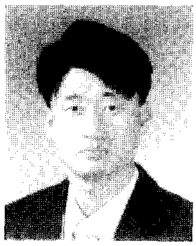
[9] I. Gravagne, "Minimum effort techniques for inverse kinematics of redundant robot manipulators," *Master Thesis,* Dept. of Electical Eng., Clemson University, 1999.

[10] Jihong Lee, "A structured algorithm for the minimum l_∞ solutions and its application to robot velocity workspace analysis" *Robotica*, vol. 19, pp. 343-352, 2001.



In-Soo Ha

He received BS and M.S. degree in Mechatronics Engineering from Chungnam National University, Taejon, Korea, in 1999 and 2001. Now he works for the Samsung Electronics, Suwon, Korean.



Ji-Hong Lee

He received BS degree in electronic engineering from Seoul National University, Seoul, Korea, in 1983 and the M.S and Ph.D. degrees in Electrical Engineering from the korea Advanced Institue of Science and Technology(KAIST), Taejon, Korea, in 1985 and 1991. From 1983 to 1988, he worked for the Hyundai Heavy Industry. From 1991 to 1993, he was an assistant professor in Electronics Control Engineering, Kum-Oh Technical College. He is now a professor in Mechatronics Engineering, Chungnam National University, Taejon, Korea. His current research interests include intelligence control, neural network, fuzzy control, motion capture, and multiple robot path planning.