Proceedings of DETC98
1998 ASME Design Engineering Technical Conference
September 13-16, 1998, Atlanta, Georgia

DETC98/MECH-5864

PERFORMANCE BASED REDUNDANCY RESOLUTION WITH MULTIPLE CRITERIA

Chetan Kapoor
Chief Scientist, Associate Director

Robotics Research Group
Mechanical Engineering Department
The University of Texas at Austin
chetan@mail.utexas.edu

Murat Cetin
Research Assistant

Robotics Research Group
Mechanical Engineering Department
The University of Texas at Austin
murat@mail.utexas.edu

Delbert Tesar
Professor, Director

Robotics Research Group
Mechanical Engineering Department
The University of Texas at Austin
tesar@mail.utexas.edu

ABSTRACT

This paper presents a scaleable approach to redundancy resolution problems involving multiple criteria. The proposed scheme relies on the disintegration of the inverse kinematics solution from multicriteria decision making. The scheme is composed of systematic generation of a set of inverse kinematics solutions and evaluating these solutions using multiple criteria. The proposed approach is especially promising for problems that can accommodate extensive number of criteria. Weights assigned to different criteria are monitored and adjusted to achieve desired performance goals. The scheme is implemented using an object-oriented operational software framework, and its functionality is tested under interactive real-time control of a 10 degrees of freedom manipulator.

1. INTRODUCTION

Kinematic redundancy has become increasingly popular in robotics through the attempts to enhance the overall performance of robots in a wide variety of tasks. The extra degrees of freedom (DOF) offered by redundancy can be used to optimize different performance measures while solving the inverse kinematics problem. These performance criteria can be defined in terms of the kinematic or dynamic parameters, and can be related to the different aspects of performance.

Numerous studies have revealed the importance of using multiple criteria for performance enhancement. In general, redundant robots are used for sophisticated tasks in uncertain and dynamic environments. They demand a built-in intelligence that draws its strength from the incorporation of diverse, task-independent performance criteria. However, introduction of multiple criteria changes the redundancy problem from a well-defined inverse kinematics problem to a multicriteria decision-making problem. With this perspective, the overall redundancy resolution problem can be supplied a set of alternatives, essentially the candidate inverse kinematics solutions. These alternatives can then be evaluated using multiple criteria. Eventually, a solution can be picked by comparing the outcomes through a decision making process in which the operator may be involved.

In this paper, we discuss a methodology for the generation of the candidate inverse kinematics alternatives for redundancy resolution. We will also present a process through which one of these alternatives is selected using multiple criteria.

2. BACKGROUND

Early treatments of redundancy tried to address the isolated problems most commonly faced by non-redundant robots. Some of these treatments include evading singularities (Yoshikawa, 1985), obstacle avoidance (Maciejewski and Klein, 1985), joint limit avoidance (Liegeois, 1977), minimization of joint torques (Hollerbach and Suh, 1987), and increasing flexibility through velocity ratio (Dubey and Luh, 1987).

An extensive survey of these (local) redundancy resolution techniques can be found in (Nenchev, 1989). Among the relevant methods for our discussion, the formulation due to Liegeois (1977) accommodates the gradient vector of a performance criterion in the homogeneous part of the solution to the inverse kinematics problem. The particular solution, defined via the "generalized inverse" (Ben-Israel and Greville, 1974), is used (at the velocity or acceleration level) to satisfy the end-effector constraints. Other feasible methods such as the "decomposition" and the "extended task space" techniques square the Jacobian matrix by modifying its structure (Chevallerau and Khalil (1988), Baillieul (1985)). The solution to the inverse kinematic problem and also the redundancy resolution problem is found by inverting this modified matrix.

When there has been more than one criterion for redundancy resolution, the popular practice has been to form a linear combination of the gradient vectors as the homogeneous part of Liegeois' formulation. However, such a practice should assign the appropriate weights to performance criteria in the linear combination throughout the operation. Variable weighting ought to be considered instead of constant weighting which may result in over compromised solutions in part for some of the criteria (Seraji and Colbaugh, 1990).

While some of the recent approaches have directly faced the problematic variable weighting issue, others have chosen to redefine the multicriteria management problem.

In an attempt to avoid weighting problem, Pamanes and Zeghloul (1991) performed the optimization of multiple criteria by assigning single performance measures to pre-specified points along the trajectory. Backes and Long (1993) partitioned the task into subtasks. The overall behavior and correspondingly the performance of the robot were viewed as aggregates of the robot's behavior and performance in these subtasks.

Facing the variable weighting problem, McGhee et al. (1994) carried out a probability based weighting scheme in which the relative weights were assigned using probability functions. These functions were determined by extensive experimentation along the trajectory before the actual task was run.

At the acceleration level, Hanson and Tolson (1995) made use of the fuzzy supervisor in fuzzy logic controllers. The variable weights were selected according to the state of the robot under physical limitations. Yet, the fuzzy logic approach created the problem of extra parameter assignment to construct the "fuzzy membership functions" while trying to alleviate the weighting problem.

The studies that have been discussed so far actually result in the selection of an inverse kinematics solution without any guarantees for optimality. Unfortunately, they also do not provide any insight about the other candidate inverse kinematics solutions.

Not resolving the redundancy, Cleary and Tesar (1990) proposed monitoring different criteria along the self-motion trajectory of the robot. The simulation was aimed to reveal the nature of the candidate inverse kinematics solutions in terms of multiple criteria.

A systematic generation of inverse kinematic solutions for redundancy resolution was later discussed by Hooper and Tesar (1994). The candidate solutions were generated by creating perturbations at each of the joint variables, and then satisfying the end-effector constraints.

3. PROPOSED SCHEME

This section discusses an approach that borrows ideas from the studies mentioned in the last two paragraphs. The approach consists of two schemes: the generation of the candidate inverse kinematics solutions and evaluation of these alternatives via performance criteria.

3.1 Generate Options

We employ a two-level scheme for the generation of inverse kinematics solutions. First level begins with the selection of $m < n$ joints of the redundant manipulator. These m joints are selected corresponding to a full rank submatrix of the Jacobian matrix, $J_{m\times n}$. The existence of faulty joints and the resulting condition of the submatrix usually influence the selection of the full rank submatrix. Thus, without any faulty joints, the full rank submatrix with the largest determinant becomes the desirable candidate since it is better conditioned than any other submatrix. Yet, the search for the submatrix with the largest determinant is a computationally expensive one and not employed in this study. The full rank submatrix, $J_{m\times m}$, is then used to find a configuration that satisfies the end-effector constraints.

Consider the following scheme to generate a single inverse kinematics solution of this nature:

Let $\overline{\theta}_{n\times 1}$ denote the current set of joint variables so that

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

gives the current end-effector position. The difference between the actual and desired end-effector position is given by

$$\delta x = x - \overline{x} \tag{2}$$

The required changes in the joint variables of the selected joints can be found using

$$\delta\theta_{m\times 1} = J^{-1}_{m\times m}\,\delta x \tag{3}$$

The new set of joint variables thus become

$$\theta_{n\times 1} = \overline{\theta} + \left[\frac{\delta\theta_m}{0_{n-m}}\right] \tag{4}$$

which is iterated through the forward kinematics, using Eq. (1)-(4), until

$$\left|x - \overline{x}\right| < \varepsilon \tag{5}$$

where ε is the error tolerated for the end-effector constraints.

The second level of the scheme makes use of the so-called "direct-search" technique (Hooper and Tesar, 1994). Naive "direct-search" creates perturbations in all of the joint variables of the redundant manipulator around a set of base values. For each joint variable θ_i, a set of at most three values is produced: $\theta_i - \delta\theta_i$, θ_i, $\theta_i + \delta\theta_i$. For n joints, the generation amounts to 3^n combinations. This number can be decreased by decreasing the number of options generated for each or some of the joints. Another way to relieve the effects of exploration may be to use a closed-form solution for m joints, which has also been employed in (Hooper and Tesar, 1994).

In our scheme, "direct-search" is used to perturb the joints that are not included in the full rank submatrix of the Jacobian. These joints constitute a subspace of the configuration space, which does not lie within the range space of the submatrix. The perturbations in these n–m joints result in a change in the end-effector position x, due to Eq. (1). The corresponding change in the manipulator configuration can be calculated by Eq. (2) and the required joint variables of the full rank submatrix to compensate this change can be computed by Eq. (3). For different sets of perturbations at $n-m$ joints, we end up with a finite subset of options that satisfy the same end-effector constraints. It should be noted that the necessary changes in the m joints could be related to the perturbations in the n–m joints explicitly. This can be calculated as follows.

$$\begin{aligned} -\delta x &= J\delta\theta = J\left[\frac{0}{\delta\theta_{n-m}}\right] \\ -\delta x &= J_{m\times(n-m)}\delta\theta_{n-m} \\ \delta\theta_m &= -J^{-1}_{m\times m}J_{m\times(n-m)}\delta\theta_{n-m} \end{aligned} \tag{6}$$

where $\delta\theta_{n-m}$ are the perturbations in the n-m joints and $\delta\theta_m$ are the compensating changes in the remaining m joints. However, iterations are still necessary even with a formulation like Eq. (6) to check if the end-effector constraints are satisfied via Eq. (5). If these perturbations are relatively small, the iterations mentioned above take place rapidly and a geometric convergence is assured (Aubin and Ekeland, 1984).

3.2 Selection

The "generate options" scheme creates a finite number of inverse kinematics solutions around the current configuration of the manipulator on the self-motion manifold. "Self-motion" is the motion that the manipulator can conduct without moving its end-effector as shown in Fig. 1.

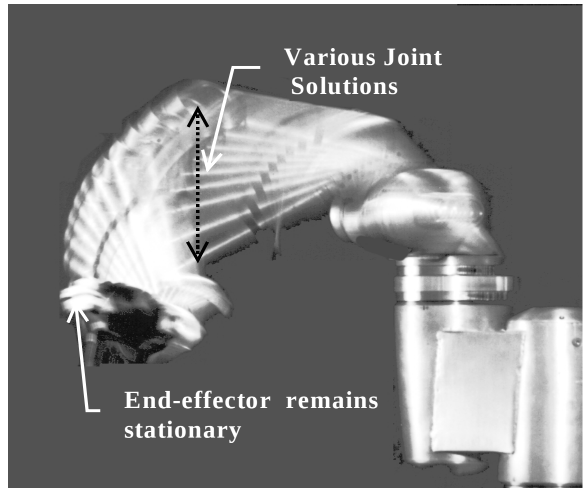

Figure 1. A Redundant 7 DOF Arm in Self-Motion.

The next step in our scheme will be to select the best solution from the finite set of options generated in the previous step.

Let $\psi_i(\theta): \Re^n \mapsto \Re$, $i = 1,\ldots,\ell$ denote the performance criteria defined as a function of the configuration $\theta \in \Re^n$. Let $\theta^1, \theta^2, \ldots, \theta^k$ be the set of configurations, or inverse kinematics options, supplied by the "generate options" scheme. For a set of weights $w_1, w_2, \ldots, w_\ell$, we calculate the following set of values

$$R_j = \frac{1}{\ell}\sum_{i=1}^{\ell} w_i \overline{\psi}_i(\theta^j) \qquad j = 1,\ldots,k \tag{7}$$

where

$$\overline{\psi}_i(\theta^j) = \frac{\psi_i(\theta^j) - \min_j\left\{\psi_i(\theta^j)\right\}}{\max_j\left\{\psi_i(\theta^j)\right\} - \min_j\left\{\psi_i(\theta^j)\right\}}$$

and $0 \le w_i \le 1$ for $i = 1,\ldots,\ell$. Notice that $\overline{\psi}_i(\cdot)$ is the normalized form of the function $\psi_i(\cdot)$ and $0 \le R_j \le 1$.

Eventually, we pick the configuration θ^j that yields the minimum value for R_j, thus

$$\theta^* = \arg\min_j R_j \tag{8}$$

becomes the resulting configuration in the redundancy resolution. Without the need for calculating the gradients of the objective functions and satisfying the end-effector constraints, the above process is a sub-optimal but efficient way of resolving the redundancy in robots with higher degrees of redundancy and computationally expensive performance criteria.

Together with the generation of options and the selection, the overall redundancy resolution scheme can be described as shown in Fig. 2.

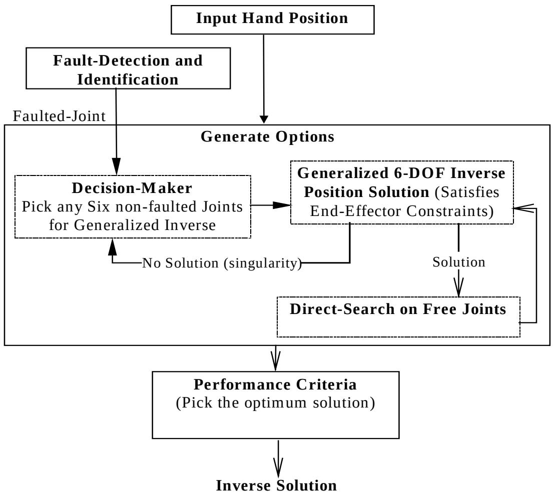

Figure 2. Redundancy Resolution Scheme.

4. EXAMPLE

Real-time experiments have been conducted with this scheme on the Robotics Research Dual Arm Robot, KB1207. KB1207 has two 7 DOF arms that are mounted on a 3 DOF torso (a total of 17 DOF).

For the sample experiment, the end-effector of the 10 DOF left substructure of KB1207 was required to follow the trajectory depicted in Fig. 3 without changing its orientation. The trajectory consists of traversing a square (side 0.5 m.) in the YZ plane followed by a circle inscribing the square boundary as shown in Fig. 3.

For the "generate options" scheme, a total of 81 candidate inverse kinematics solutions was generated at each instance of the trajectory. The generation was realized by perturbing the first, second, third, and fifth joints by increments of 0.002 radians.

For the "selection" scheme, the following three performance criteria were used.

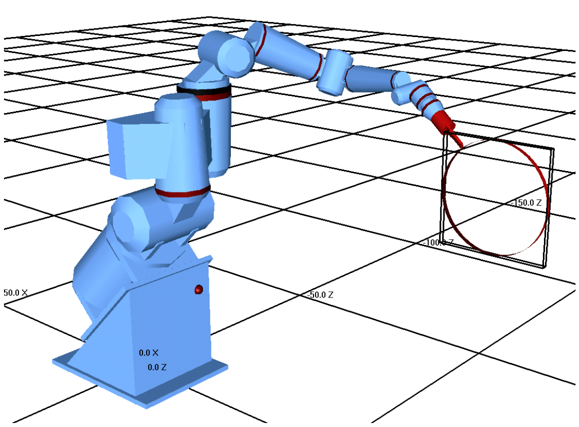

Figure 3. 10 DOF Manipulator with Sample Trajectory.

Joint Range Availability (JRA):

$$\psi_1 = \sum_{i=1}^{10} \frac{(\theta_i - \theta_i')^2}{\theta_{i,\max}}$$

where $\theta_i, \theta_i', \theta_{i,\max}$ are the joint displacement, mid-range displacement and the displacement at the joint limit respectively. JRA tends to keep the deviation of the joint angles from their mid-ranges as small as possible. Thus, a smaller JRA means a better solution.

Joint Velocity Measure (JVM):

$$\psi_2 = \sum_{i=1}^{10} (\theta_i - \theta_{i,curr})^2$$

where $\theta_i, \theta_{i,curr}$ are the desired and current positions of the joint *i*, respectively. JVM gives an indication about the magnitude of the joint velocities by computing the displacement to be traveled. Thus, the solution with a smaller JVM is a better solution.

Dexterity Measure (DM):

$$\psi_3 = \frac{\sigma_{\max}}{\sigma_{\min}}$$

where $\sigma_{\max}, \sigma_{\min}$ are the maximum and minimum singular values of the Jacobian matrix. DM varies between 1 and $+\infty$. DM=1 indicates the optimum configuration for which the Jacobian matrix is perfectly conditioned.

Figure 4 depicts the variation of each criterion value for a sub-section of the complete trajectory. Two lines, the full line and the dashed line, show the two different results that were achieved by using different weights for the three criteria.

The full line shows the variations in the criteria when all the weights in Eq. (7) are fixed and equal to one. While executing this trajectory, it was seen that the criterion value for the dexterity criterion exceeded 300. In the next run, we tried to keep the value of the dexterity criterion below 300 by actively varying the weights of the criteria. We were able to monitor the criterion values and change the weights through a graphical user interface. For this second run, we started with $w_1 = w_2 = w_3 = 1$ and then changed the weights to $w_1 = 0.8$, $w_2 = 0.2$, $w_3 = 1.0$ when the DM hit the 250 threshold. This change of weights is marked on the plots by the vertical dotted line. The dashed line depicts the variations in the criteria after the modification of the weights. As can be seen, the proposed scheme is responsive to the variable weighting strategy. Thus, it produces the anticipated behavior by steering away from the target constraint value. As the weights of the JRA and JVM are reduced, we observe an increase in the corresponding criteria values. Yet, the DM changes its trend staying below the target value.

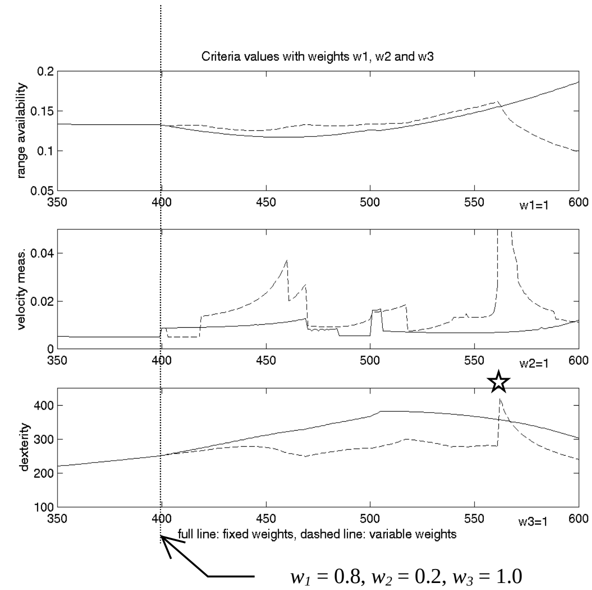

Figure 4. Criteria Values with the Proposed Scheme.

Figure 4 also depicts abrupt increases in both the DM and JVM values, which could not be avoided with just monitoring the criteria values and adjusting the weights promptly. This limitation of the decision making scheme is due to what may be called the "inertia" effect. Thus, we do not necessarily observe the same abrupt changes in criteria values at the designated location if we set out with the modified set of weights in the previous example. The "inertia" effect is in fact substantiated by both the sub-optimal nature of the "generate options" scheme and lack of an intelligent weighting strategy to guide the search in the criteria space. Such anomalies can be prevented through better exploration with sophisticated optimization techniques that incorporate intelligent weighting strategies on the self-motion topology. This is still an active area of research within the Robotics Research Group, The University of Texas at Austin.

In our test environment an operational frequency of approximately 60 Hertz was achieved by this scheme on a Silicon Graphics R10000 machine. Speeds upwards of 300 hertz have also been achieved by this scheme for a 7 DOF manipulator with three criteria on a SUN-Sparc 20 workstation.

The software for this application was developed using OSCAR (Operational Software Components for Advanced Robotics), and object-oriented framework for developing control software for intelligent machines (Kapoor and Tesar, 1996). This framework presents the applications developer with a generalized and extensible environment that supports mathematical abstractions, generalized kinematics (both forward and inverse), generalized dynamics, deflection modeling, performance criteria, criteria fusion, and fault-tolerance. The OSCAR framework is equally applicable to real-time control (as demonstrated with this example) and simulation. Additionally, OSCAR supports a distributed computational environment that includes VME based computers, UNIX workstations, and Windows NT. C++ is the programming language supporting OSCAR. OSCAR provides application testing and development support through a variety of manual controllers, manipulators, and simulation computers (Kapoor and Tesar, 1996).

5. CONCLUSIONS

An efficient approach has been proposed for the resolution of kinematic redundancy in robots. The approach consists of two schemes. First, the candidate inverse kinematics solutions that satisfy the end-effector constraints are generated. For this purpose, a full rank submatrix of the Jacobian matrix is selected, and then a set of configurations are created by perturbing the corresponding "locked" joints of the manipulator. These locked joints constitute the subspace of the configuration space, which is not spanned by the columns of the submatrix. Second, after the candidate solutions are generated, the optimum solution is picked through a selection scheme, which incorporates variable weighting of multiple performance criteria. The proposed approach has been shown to be both responsive to the variable weighting strategy and efficient for the real-time control of a 10 DOF robot.

The ongoing research focuses on improving the current selection scheme, the weighting or the parametric method, which is indeed one of the widely recognized vector optimization methods (Gembicki, 1974). The parametric method relies on the convexity of the feasible solution set to explore all of the compromise solutions between the criteria. Recall that the feasible solution set is the set of candidate inverse kinematics solutions that may satisfy additional inequality constraints on the criteria values and kinematic parameters. The feasible set can be explored for the compromise or non-dominated solutions more effectively using other vector optimization techniques that convey the "goal attainment" information from the user. Such techniques not only discharge the convexity assumption but also retain the user intervention at a higher level than having to modulate the weights continuously. Another aspect of the ongoing research is to incorporate local functional information such as sensitivity of the criteria to the perturbations (derivative information) to the exploration of the feasible set via sophisticated optimization techniques. Such information is sought to avoid the abrupt variations in the criteria, which are simply unavoidable using only the criteria values.

ACKNOWLEDGMENTS

Authors acknowledge the work done by Kevin Rackers in developing hardware interfacing software for the Robotics Research Dual Arm robot.

REFERENCES

Aubin, J. P. and Ekeland, I., 1984, *Applied Nonlinear Analysis*, John Wiley and Sons, New York.

Backes, P. G., and Long, M. K., 1993, "Merging Concurrent Behaviors on a Redundant Manipulator," *IEEE International Conference on Robotics and Automation*, pp. 638-645.

Baillieul, J., 1985, "Kinematic Programming Alternatives for Redundant Manipulators," *IEEE International Conference on Robotics and Automation*, pp. 722-728.

Ben-Israel, A., and Greville, T.N.E., 1974, *Generalized Inverses: Theory and Applications*, John Wiley and Sons, New York.

Chevallerau, C. and Khalil, W., 1988, "A New Method for the Solution of the Inverse Kinematics of Redundant Robots," *IEEE International Conference on Robotics and Automation*, Vol. 2, pp.37-42.

Cleary, K., and Tesar, D., 1990, "Incorporating Multiple Criteria in the Operation of Redundant Manipulators," *IEEE International Conference on Robotics and Automation*, pp. 618-624.

Dubey, R., and Luh, J.Y.S., 1987, "Redundant Robot Control for Higher Flexibility," *IEEE International Conference on Robotics and Automation*, pp. 1066-1072.

Gembicki, F. W., 1974, "Performance and Sensitivity Optimization: A Vector Index Approach," Ph.D. Dissertation, Case Western Reserve University.

Hanson, M. L., and Tolson, R. H., 1995, "Using a Fuzzy Supervisor to Optimize Multiple Criteria in Redundant Robots," *Proc.* 10th *IEEE International Symposium on Intelligent Control*, pp. 145-150.

Hollerbach, J. M., and Suh, K. C., 1987, "Redundancy Resolution of Manipulators through Torque Optimization," *IEEE Journal Robotics and Automat.*, Vol. RA-3, No. 4, pp. 308-316.

Hooper, R. N., and Tesar, D., 1994, "Multicriteria Inverse Kinematics for General Serial Robots," Ph.D. Dissertation, The University of Texas at Austin.

Kapoor C., and Tesar, D., 1996, "A Reusable Operational Software Architecture for Advanced Robotics," Ph.D. Dissertation, The University of Texas at Austin.

Liegeois, A., 1977, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-7, No. 12, pp. 868-871.

Maciejewski, A. A., and Klein, C. A., 1985, "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments," *International Journal of Robotics Research*, Vol. 4, No. 3, pp. 109-117.

McGhee, S., Chan, T. F., Dubey, R. V., and Kress, R. L., 1994, "Probability-Based Weighting of Performance Criteria for a Redundant Manipulator," *IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 1887-1894.

Nenchev, D.N., 1989, "Redundancy Resolution through Local Optimization: A Review," *Journal of Robotic Systems*, Vol. 6, No. 6, pp. 769-798.

Pamanes, J. A. G., and Zeghloul, S., 1991, "Optimal Placement of Robotic Manipulators Using Multiple Kinematic Criteria," *IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 933-938.

Seraji, H., and Colbaugh, R., 1990, "Improved Configuration Control for Redundant Robots," *Journal of Robotic Systems*, Vol. 7, No.6, pp. 897-928.

Yoshikawa, T., 1985, "Manipulability and Redundancy Control of Robotic Manipulators," *IEEE International Conference on Robotics and Automation*, pp. 1004-1009.