

**“ON THE COMPUTATIONAL EFFICIENCY OF FOUR NUMERICAL
TECHNIQUES FOR SOLVING THE PRACTICAL INVERSE KINEMATICS
PROBLEM OF REDUNDANT MANIPULATORS”**

By

Tarek M. Abdel - Rahman
Mechanical Engineering Department,
Faculty of Engineering, Qatar University,
Doha, Qatar - Arabian Gulf.

ABSTRACT

In this paper we compare the computational efficiency of four methods for solving the practical inverse kinematics problem of redundant manipulators. Two methods use recursive approaches for sequential determination of feasible joint rates while the other two methods derive complete joint rate solutions at first and then check the joint kinematical constraints. The methods were implemented on a 386 microcomputer. Issues of improving efficiency by changing velocity reference frames and workspace decomposition were examined.

1. INTRODUCTION

Non-redundant manipulators have limited dexterous workspaces because of joint kinematical constraints and manipulator kinematical singularities. Dexterity can be improved by employing redundant manipulators. The redundant degrees of freedom can be used for singularity, obstacle or joint constraint avoidance or for optimization of a selected criterion for resolving the kinematical redundancy in the inverse velocity problem. Algorithms for redundancy resolution differ in their computational complexity. As the complexity increases, the computational cycle time increases and errors due to velocity linearization also increase. The effect of these increases on the performance is evaluated in this paper.

Most of the algorithms proposed in the literature either ignore the joint kinematical constraints or consider them indirectly via weighting matrices in quadratic joint rate functions. The later approach has been evaluated and compared with the direct consideration of the kinematical constraints in the solution procedure. Four solution methods were applied to solve the practical inverse kinematics problem of the 10 DOF manipulator shown in Fig. 1.

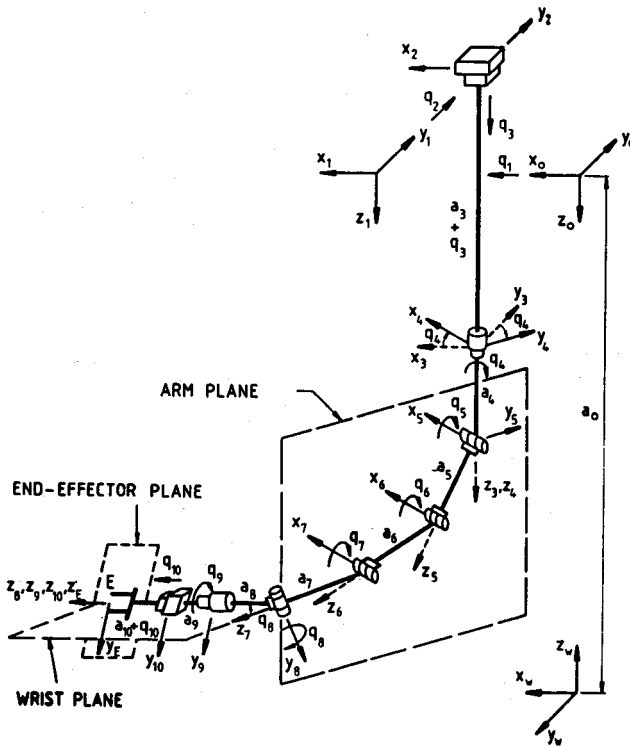


Fig. 1 : A Ten-Degrees-of-Freedom Manipulator.

The methods were the LU factorization [1], the Singular Value Decomposition (SVD) method [2], the Most-Effective-Direction (MED) method [3] and the Recursive-Orthogonal-Motion-Resolution (ROMR) method [4]. The first two methods considered the kinematical constraints via the weighting matrices approach while the last two methods considered the joint rate constraints recursively in the solution procedures.

The first three methods: the LU, SVD and MED were implemented on a Wang VS100 computer in [3]. In this paper, the programs of [3] were run on a 16 MHz-386 Compaque microcomputer and the results were compared with recent ones determined by applying the ROMR method and the workspace decomposition approach [4].

2. THE PRACTICAL INVERSE KINEMATICS PROBLEM

The inverse kinematics problem can be formulated as follows : Let \underline{x} denote a vector of dimension m containing the desired positions and orientation of the end-effector in workspace coordinates referred to an inertial base frame, and let \underline{q} denote a vector of dimension n containing the joint positions. The relation between the joint rates $\dot{\underline{q}}$ and the end-effector velocity $\dot{\underline{x}}$ is given by:

$$\dot{\underline{x}} = J(\underline{q}) \dot{\underline{q}} \quad (1)$$

where $J(\underline{q})$ is the Jacobian matrix of dimension $m \times n$. If $n > m$, the manipulator is kinematically redundant. In practice, each joint variable q_i has position, rate and acceleration constraints that can be described by:

$$\begin{aligned} q_{i,\min} &< q_i < q_{i,\max} \\ \dot{q}_{i,\min} &< \dot{q}_i < \dot{q}_{i,\max} \\ \ddot{q}_{i,\min} &< \ddot{q}_i < \ddot{q}_{i,\max} \end{aligned} \quad i=1, 2, \dots, n \quad (2)$$

The position boundaries in (2) are constants for most designs while the rate and acceleration bounds are function of links and payload inertia, the manipulator configuration and maximum motor torques. If \underline{T} denote an n -element torque vector of the joint actuator, the torque constraints can be defined by:

$$\underline{T}_{\min} < \underline{T} < \underline{T}_{\max} \quad (3)$$

The practical inverse kinematics problem aims at finding a $\dot{\underline{q}}$ solution of (1) for a given $\dot{\underline{x}}$ such that the constraints (2) and (3) are satisfied.

By sampling the motion at various stages, we can map the position and dynamics constraints onto the joint rate space [4], hence, (2) and (3) yield:

$$\underline{\dot{q}}_{inf}^k < \underline{\dot{q}}^k < \underline{\dot{q}}_{sup}^k \quad (4)$$

at the time stage k . In this paper, we address the solution of the practical inverse kinematics problem with joint rate constraints in the form (4) and assume that they represent (2) and (3).

3. THE SOLUTION METHODS AND PERFORMANCE INDEX

Redundancy resolution methods vary in their approaches and hence provide diversified solutions for the local inverse kinematics problem. For comparison, we apply a constant velocity command $\underline{\dot{x}}_c$ for a period of time and compute the performance index from:

$$\text{Performance Index} = \frac{\text{Average End-Effector Speed}}{\text{End-Effector Speed Command}}$$

Four redundancy resolution methods were considered. The first two, the LU factorization and the SVD methods, resolve all the joint rates simultaneously. Each of the two methods determines, at first, a generalized inverse of the Jacobian matrix and then computes a solution for the given velocity command. When one of the computed joint rates exceeds its kinematical constraints, all the joint rates are scaled down proportionally to obtain the largest feasible solution. The LU solution minimizes a quadratic function of weighted joint rates, while the SVD solution minimizes the norm of the joint rates.

The operation counts of the LU and SVD methods are:

$$\begin{aligned} \text{LU} & : m^2 \frac{(3n-m)}{6} + (k^2 + k + 1)(n+m) + \frac{1}{3} k^2 \\ \text{SVD} & : 2mn^2 + f(n) \end{aligned}$$

where an operation is a multiply or division plus an add, k is the rank of the Jacobian matrix and $f(n)$ is a term that accounts for the iterative phase of the SVD method. If normal convergence occurs: $f(n) \cong 4n^3$. The last two methods,

the MED and the ROMR, determine feasible joint rates recursively. The methods compute the contribution of each joint rate to the end-effector velocity and then determine the residual motion to be provided by the remaining joints. The ROMR method follows the same recursive resolution approach of the MED method but the resolution occurs in orthogonalized directions. The computation of the orthogonalized resolution directions requires a relatively small additional computational time but is essential for fast convergence to a solution. In many cases, the orthogonalized resolution direction could be derived analytically and hence reduces the computational time. In such cases the number of operations becomes $mn^2 + 2n^2 + mn$ for the two recursive methods.

4. A CASE STUDY

The manipulator shown in Fig. 1 was used in evaluating the computational efficiency of the four methods. The Jacobian matrix of the manipulator was a 6×10 matrix [3]. To facilitate the computation of the recursive methods, the Jacobian matrix was decomposed in the joint space. The end-effector velocity equation (1) became:

$$\dot{\underline{x}} = J_{j1} [\dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3]^T + J_{j2} [\dot{q}_4 \quad \dots \quad \dot{q}_{10}]^T \quad (5)$$

where the subscripts $j1$ and $j2$ refer to the first and second decomposed joint spaces. Velocity resolution was done, at first, in the joint space, $j2$, of the distal joints with the first three joint rates set equal to zero. If some linear components remained unresolved the errors were compensated by the first three joints.

Experience in using the MED method [4] showed difficulty in convergence if the Jacobian matrix columns were not nearly orthogonal. To improve the performance of the MED method the reference frame of the end-effector velocity was changed to the fourth frame instead of the initial frame. The performance was then evaluated by applying six velocity commands v_1, v_2, \dots, v_6 of unity magnitudes in the various six dimension directions, e.g. $v_1 = (1 \ 0 \ 0 \ 0 \ 0 \ 0)^T$. Each command was applied for 10 secs.

Computer programs for resolving the kinematics problem were written in Fortran and implemented on a 16MHz-386 Compaque microcomputer. All methods yielded satisfactory results if no computational time constraints were introduced. Many runs were done to determine appropriate cycle times for the four methods. The smallest appropriate time cycles were, 1, .5, and .25 sec. for the SVD, LU and recursive methods respectively. When the cycle times were considered we obtained the results shown in Fig. 2.

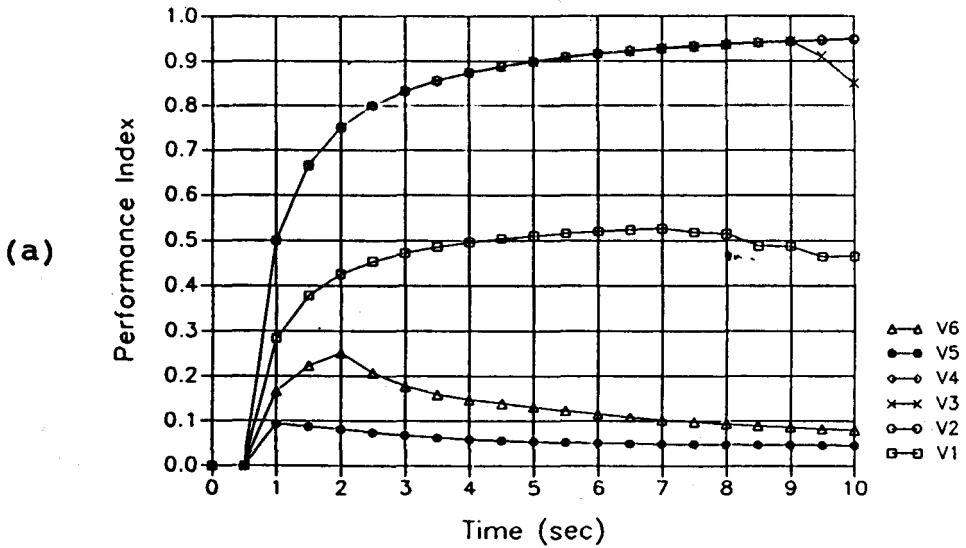


Fig. 2 (a)

Fig. 2-a shows the performance index of the LU method, while Figs. 2-b and 2-c show the performance index of the SVD and the two recursive methods. The Figures illustrate that the performance degrades as the cycle time increases and that small index values occur for the v_5 and v_6 angular velocity commands. This was attributed to the kinematical constraints associated with the few number of joints that contribute to those commands. The results reported of the MED method has shown improvement than those of [2]. The improvement was due to the selection of an intermediate reference frame that enhanced the structure of the associated Jacobian matrix. The recursive approaches yielded the best performances. This is attributed to smaller cycle times and consideration of the joint kinematical constraints directly in the solution procedure.

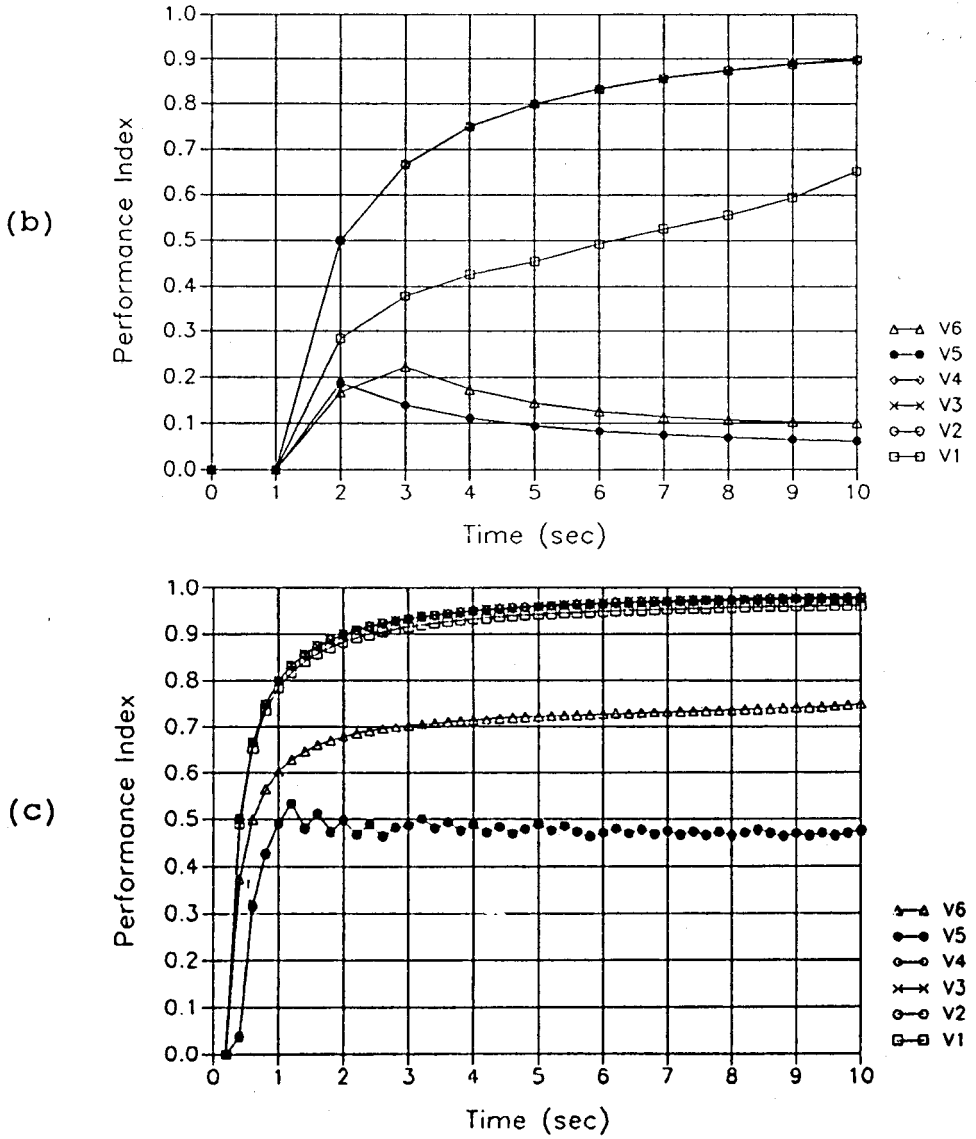


Fig. 2: Performance Index Variations:
 (a) LU Method, (b) SVD Method
 (c) MED and ROMR Methods

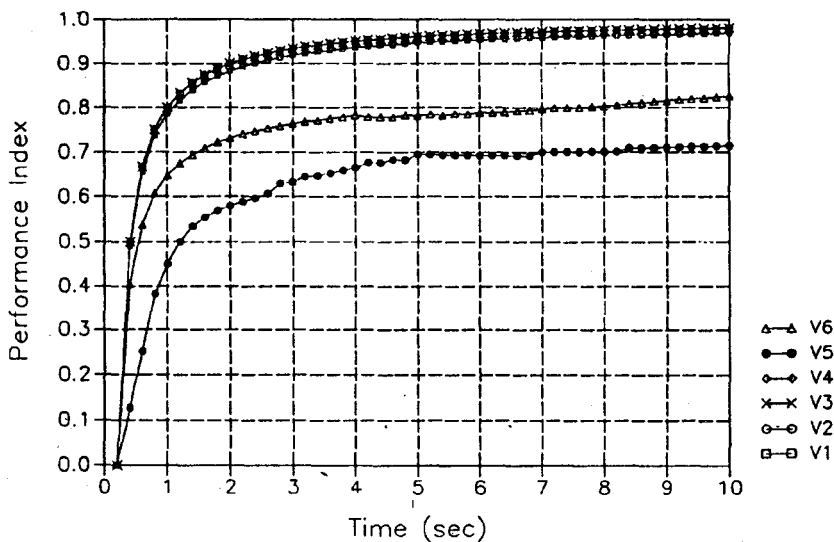
To further improve the performance of the v_5 and v_6 command cases, the workspace was decomposed into two sub-workspaces. The first sub-workspace comprised the joints of major contribution to motion in the directions of the v_5 and v_6 commands and the second sub-workspace comprised the remaining joints. The velocity was resolved at first in the first sub-workspace then in the second sub-workspace. The results are shown in Fig. 3. In this figure, the increase of the v_5 and v_6 performance indices illustrates the advantages of the workspace decomposition.

5. CONCLUSION

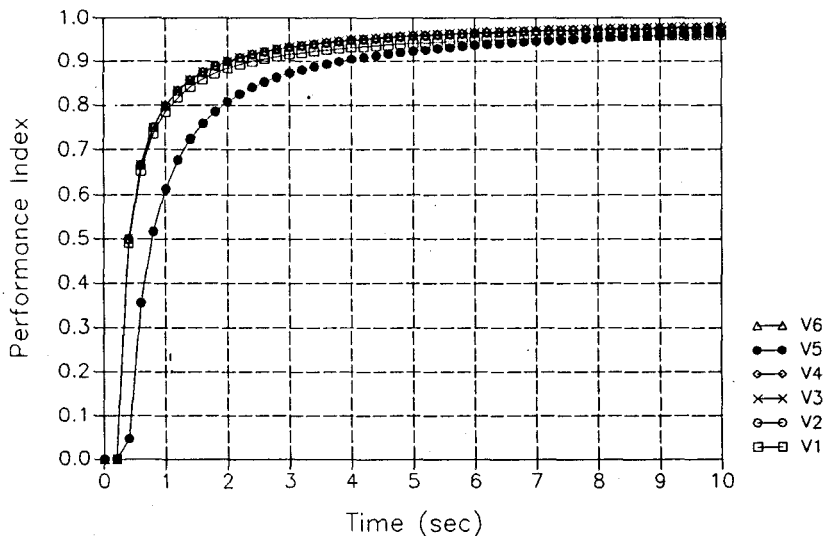
Two approaches for solving the practical inverse problem of redundant manipulators were evaluated. The first approach, simultaneously considers all the joints while the second approach uses the joint kinematics constraints for sequential resolution of the velocity. Four solution methods were implemented on a 16MHz - 386 microcomputer. The methods evaluated were the LU, SVD, MED and ROMR methods. The cycle times of the recursive methods were less than half of the cycle time of the non recursive methods. Furthermore, the velocity errors in the recursive methods were much smaller than those of the non recursive ones.

Results showed that the MED method performance can be improved by representing the end-effector velocity in an intermediate joint reference frame. The ROMR method yielded the best performance amongst the four methods. The ROMR has also shown better capability in determining robust solution and identifying singular and near singular configurations.

The recursive methods performance can be improved by decomposing the workspace into two sub-workspaces and solving two smaller-dimension inverse kinematics problems. In manipulator design, the decomposition implies selecting structural configurations that allow decomposition of the workspace into two orthogonal complement sub-workspaces. Furthermore, if kinematical redundancy in each of the sub-workspaces is achieved better performance would be expected.



(a) MED Method



(b) ROMR Method

Fig. 3: Performance Index Variation for the Recursive Decomposed Workspace Methods

REFERENCES

1. **Abdelmalek, N.N.** "On the Solution of Linear Least Squares Problems and Pseudo-Inverses", *Computing*, Vol. 13, pp. 215-228, 1974.
2. **Klema, V.C., and Laub, A.J.** "The Singular Value Decomposition: Its Computation and Some Applications", *IEEE Journal of Automatic Control*, AC-25, No. 2, pp. 164-176, 1980.
3. **Abdel-Rahman, T.M. and Elghamrawy, T.M.** "Most-Effective Direction Joint Control Algorithm of a Ten-Degrees of Freedom Manipulator", *Proceedings of the IASTED International Symposium on Robotics and Automation*, Lugano, Switzerland, June 24-26, pp. 175-179, 1985.
4. **Abdel-Rahman, T.M., and Takriti, M.S.** "On-line Computational Schemes for the Inverse Problem in Robotics", *Proceedings of the IASTED International Symposium on Robotics and Artificial Intelligence*, Toulouse, France, June 18-20, pp. 249-267, 1986.
5. **Abdel-Rahman, T.M.** "A Generalized Practical Method for Analytical Solution of the Constrained Inverse-Kinematics Problems of Redundant Manipulators", *The International Journal of Robotics Research*, 1991 (In Press).