MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0348119 5

ORNL/TM-11435
CESAR-89/53

ornl

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

A New Approach to Solve the Kinematics Resolution of a Redundant Robot

Philippe Belmans
Jean-Christophe Culioli

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available from (615) 576-8401, FTS 626-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161.
NTIS price codes—Printed Copy: A03 Microfiche A01

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM -11435
(CESAR-89/53)

Engineering Physics and Mathematics Division

A New Approach to Solve the Kinematics Resolution of a Redundant Robot

Philippe Belmans and Jean-Christophe Culioli
Center for Engineering Systems Advanced Research

DATE PUBLISHED — March 1990

Prepared for the
Engineering Research Program
Office of Basic Energy Sciences
U.S. Department of Energy

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE–AC05–84OR21400

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0348119 5

CONTENTS

ACKNOWLEDGMENTS v
ABSTRACT vii
1. INTRODUCTION 1
2. THE PSEUDO-INVERSE ALGORITHM 3
3. THE PROPOSED ALGORITHM 5
4. APPLICATION TO A 4-D.O.F. PLANAR ARM 7
5. GENERALIZATION OF THE ALGORITHM TO OTHER CRITERIA . 9
6. CONCLUSION 11
APPENDIX A 13
REFERENCES 17

LIST OF FIGURES

Fig.		Page
1	Four-degree-of-freedom planar arm	7
2	Plotting of the arm and the end effector obtained by simulation . .	8

ACKNOWLEDGMENTS

This research was partially sponsored by an ORNL Postgraduate Research Appointment administered by Oak Ridge Associated Universities and by a NATO and CEA (France) fellowship.

The authors would like to thank Dr. V. Protopopescu and Dr. M. Unseren for many insightful discussions during the preparation of this paper.

ABSTRACT

A classical redundancy resolution scheme for an m-degree of freedom robot involves the numerical or symbolic computation of the Moore-Penrose pseudo-inverse of the Jacobian matrix, which in turn leads to a least norm solution for the joint velocities (see for example [1]). Since the Jacobian matrix may be ill-conditioned, the computation of the inverse may often turn out to be lengthy and/or inaccurate. In this paper, we propose an alternative method to find this least norm solution. Namely we modify the original underdetermined problem by transforming it into a set of determined new problems. We compute in parallel the solutions of these problems and find their least norm simplex combination. We prove that if the dimension of the task-space is n, we need only $(m - n + 1)$ such solutions. We also show that the approach can take into account obstacle avoidance and maximum of manipulability or any other type of analytical criterion. We apply the method to a planar redundant arm.

1. INTRODUCTION

We consider an m degrees of freedom robot. The state it occupies is determined by an articular position vector q of R^m, but usually the specification of a task takes place in another space R^n called operational space or task-space (for most industrial robots $n = 6$). It is easy to express the operational position X analytically as a function of the articular position q:

$$X = F(q); q \in R^m; X \in R^n.$$

However, controlling the kinematics of a robot requires to find the inverse relationship:

$$q = G(X); q \in R^m; X \in R^n.$$

In most of the cases, it is impossible to obtain this relationship analytically and globally. Usually, the problem is solved by linearizing the operator $F(q)$ in the neighborhood of the point q. A small variation ΔX, called task-vector, of the end-effector position is expressed in the operational space (or task space) as:

$$\Delta X = J(q)\Delta q; \Delta q \in R^m; \Delta X \in R^n. \tag{1}$$

where J is the $(n \times m)$ Jacobian matrix and Δq is the corresponding small variation of the articular position. We assume that J has rank n; for a fixed q and for a specified ΔX, the set of solutions is an affine space of dimension $m - n$, which we shall denote by $\mathcal{E}$.

Controlling the kinematics of the robot is equivalent to solve periodically this system, i.e. to determine a suitable control vector Δq as a function of ΔX, the task-vector. In most applications, the operational space has a dimension $n = 6$. In the case of a non-redundant arm, $m = 6$ and provided the Jacobian matrix $J(q)$ is invertible (q is not a singular position), the solution Δq is given by:

$$\Delta q = J^{-1}(q)\Delta X; \Delta X \in R^6; \Delta q \in R^6; q \in R^6. \tag{2}$$

In the case of redundant arms ($m > n = 6$), the Jacobian matrix is not square and cannot be inverted. One method consists of finding the Δq of least norm that verifies Eq. (1). This leads naturally to the introduction of a pseudo-inverse of J, namely $J^T(JJ^T)^{-1}$. To avoid the computation of the pseudo-inverse, algorithms work with square sub-matrices of the Jacobian matrix to compute the solutions, but these algorithms are tailored for 7-d.o.f. manipulators ($m = 7$) and it is not possible to generalize them for robot with more d.o.f. As an alternative, we propose to extract numerically well-behaved square sub-matrices of J, obtain their solutions, and find their least norm simplex combination. One advantage of this approach is its generality to m-d.o.f. arms and its inherent parallel structure.

2. THE PSEUDO-INVERSE ALGORITHM

To solve min $\frac{1}{2}\|\Delta q\|^2$ subject to the constraint (1), we introduce the Lagrangian:

$$\mathcal{L}(\Delta q, \mu) = \frac{1}{2}\|\Delta q\|^2 + < \mu, J\Delta q - \Delta X >$$

and solve the necessary optimality conditions:

$$\Delta q + J^T \mu = 0 \tag{3}$$

$$J\Delta q - \Delta X = 0. \tag{4}$$

Multiplying Eq. (3) by J and using Eq. (4), we get

$$\Delta X + JJ^T \mu = 0.$$

If we assume that $J(q)$ has a maximal rank (q is not a singular position), i.e. n, then we can invert JJ^T:

$$\mu = -(JJ^T)^{-1}\Delta X.$$

Substituting this value of μ in Eq. (3) gives:

$$\Delta q^* = J^T(JJ^T)^{-1}\Delta X \tag{5}$$

where $J^T(JJ^T)^{-1}$ is the Moore-Penrose pseudo-inverse of the Jacobian matrix [2]. Nevertheless, the implementation of this algorithm very often fails because the numerical computation of the pseudo-inverse is often very ill conditioned (small determinants, large differences between various elements, etc.) [3]. Furthermore, the determination of the pseudo-inverse using the SVD (singular value decomposition) technique is too slow for a real-time implementation. Other difficulties in resolving the redundancy using the SVD method to calculate the pseudo-inverse of $J(q)$ are described in [4].

3. THE PROPOSED ALGORITHM

To avoid the computation of the Moore-Penrose pseudo-inverse,[5] applies Gradient Projection Method to a 7-d.o.f. arm and the form of the solution is:

$$\Delta q = \Delta q_p + k\Delta q_h; \Delta q \in R^7; \Delta q_p \in R^7; \Delta q_h \in R^7; k \in R.$$

Here Δq_p is a particular solution of Eq. (1) and Δq_h is the solution of the homogeneous equation:

$$\bar{0} = J(q)\Delta q_h; \Delta q_h \in R^7; \bar{0} \in R^6.$$

This algorithm is computationally efficient and was successfully implemented. However, concerning optimality, no proof was provided that the result is the least norm solution. Moreover, it is not possible to guarantee an upper bound of the computation delay as the algorithm includes *(repeat....until)* statements in the flow control. Besides, no study was carried out to study efficient implementation of the Gradient Projection Method applied to an arbitrary number m of degrees of freedom.

In [6], an interesting combinatorial approach to Inverse Kinematics is proposed. The idea is to extract all the s possible $n \times n$ submatrices of the Jacobian matrix, invert them, and solve in parallel the associated linear systems analogous to Eq. (1). Then, among the s solutions, one picks the one with the least norm. The intuitive interpretation is that it is possible to physically block enough joints to compensate the underspecification of the problem. In that case the time boundedness of the computation is guaranteed. However, the solution obtained is not necessarily the least norm one, since the best solution produced by this scheme always includes the blocking of one of the joints.

We suggest a similar technique, based on the linear properties of the coordinate transformation in the neighborhood of the point q expressed by the Jacobian matrix J and its $s = C_m^n$ sub-matrices $J_1, J_2, \ldots, J_k, \ldots, J_s$ of dimension $n \times n$. If each J_k is assumed invertible, it is possible to obtain n-dimensional vectors Δq_k such that:

$$\Delta q_k = J_k^{-1}\Delta X, (\Delta q_k \in R^n, k \in \{1, \ldots, s\}) \tag{6}$$

Without changing their names, we can rewrite these vectors Δq_k as m-dimensional vectors in the articular space, setting the $(m-n)$ complementary components to 0. Specifically, if the submatrice J_k is formed by blocking columns $i_1, i_2, \ldots, i_{m-n}$ in $J(q)$, then the components of positions $i_1, i_2, \ldots, i_{m-n}$ are set to zero in the corresponding m-dimensional vector Δq_k. Let us consider a maximal set of independent vectors $\Delta q_k, k = 1, \ldots, p$. We now introduce E, the affine space of dimension $dim(E) = p - 1$ spanned by the family $\{\Delta q_1, \ldots, \Delta q_p\}$:

$$E = \left\{ \Delta Q \in R^m \mid \Delta Q(t_1, \ldots, t_p) = t_1\Delta q_1 + \cdots + t_p\Delta q_p; \quad \sum_{k=1}^{p} t_k = 1 \right\} \tag{7}$$

Since the m-dimensional vectors Δq_k are solution of the original problem, E is a subspace of the affine space $\mathcal{E}$ (of dimension $m-n$) defined by the solutions of Eq. (1) and thus $p - 1 \leq m - n$. Now:

- If $p-1 < m-n$, the optimal least norm solution of Eq. (1) may not belong to E, therefore the algorithm does not necessarily give it;
- If $p-1 = m-n$, then E coincides with $\mathcal{E}$ the set of solutions of Eq. (1) and the optimal solution necessarily belongs to E.

We can thus find the least norm vector using the parameterization (7). Specifically, we have to find:

$$\min \frac{1}{2}\|t_1\Delta q_1 + \cdots + t_p\Delta q_p\|^2 \tag{8}$$

subject to the constraint $\sum_{k=1}^{p} t_k = 1$. The corresponding Lagrangian is:

$$\mathcal{L}(t_k,\beta) = \frac{1}{2}\|t_1\Delta q_1 + \cdots + t_p\Delta q_p\|^2 + \beta(\sum_{k=1}^{p} t_k - 1)$$

from which we derive the necessary optimality conditions:

$$\sum_{k=1}^{p} t_k = 1 \tag{9}$$

$$(\forall k \in \{1,\ldots p,\})(t_1 < \Delta q_1, \Delta q_k > + \cdots + t_p < \Delta q_p, \Delta q_k > + \beta = 0). \tag{10}$$

We denote by G the Gramian matrix of the vectors $\{\Delta q_1,\ldots,\Delta q_p\}$ for which $g(i,j) =< \Delta q_i, \Delta q_j >$. We call $t = (t_1,\ldots,t_p)^T$ and e the p-dimensional vector$(1,1,\ldots,1,1)^T$. Then Eqs. (9) and (10) can be expressed in the simpler form

$$e^T t = 1, \quad Gt + \beta e = 0. \tag{11}$$

Since G is invertible ($\{\Delta q_1,\ldots,\Delta q_p\}$ are independent), we get easily

$$t^* = \frac{G^{-1}e}{e^T G^{-1} e}. \tag{12}$$

The least norm ΔQ is thus

$$\Delta Q = \sum_{k=1}^{p} t_k^* \Delta q_k. \tag{13}$$

In conclusion, $(m-n+1)$ independent solutions of Eq. (1) suffice to compute the least norm ΔQ. For instance, in the case of the 7-d.o.f. spatial serial link arm, only two 6-dimensional vectors are required to control the robot. Moreover, the advantage of this algorithm over the Pseudo-Inverse Algorithm is that the square submatrices can be tested for ill-conditioning or singularity. Since, for a m-d.o.f. redundant robot, $C_m^n > m-n$, one has a very good chance to find $m-n+1$ well-conditioned sub-matrices and use those to determine the solution space. In the following, we illustrate the method on a 4-d.o.f. planar arm, using the symbolic manipulation environment Mathematica [7] on Macintosh II.

4. APPLICATION TO A 4-D.O.F. PLANAR ARM

Let us apply the algorithm to the arm illustrated in Fig. 1. We specify the translational position of the robot tip at point X_E, but do not specify orientation of the outermost link and end effector. Since we have 4 degrees of freedom and the task space is 2-dimensional, we need $(4-2+1)=3$ solutions Δq_k. The Cartesian displacement is denoted $\Delta X = (\Delta x, \Delta y)$. Among the 4 2×2 submatrices J_{rs} that can be extracted from J by selecting successive columns (with obvious notation J_{12} corresponds to blocking joints 3 and 4), we select the three having the largest determinant (for a reason of sensivity to noise) and compute the three corresponding Δq_{rs}

$$\Delta q_{rs} = (J_{rs}^{-1})\Delta X.$$

We then compute the associated Gramian matrix G, get t^* from Eq. (12), and Δq from Eq. (13). Figure 2 illustrates the motion of the arm when the end-effector follows a circular trajectory. Fifty elementary displacements ΔX are used to describe the circle. We refer the reader who is familiar with *Mathematica* to the listing of the program given in Appendix A.

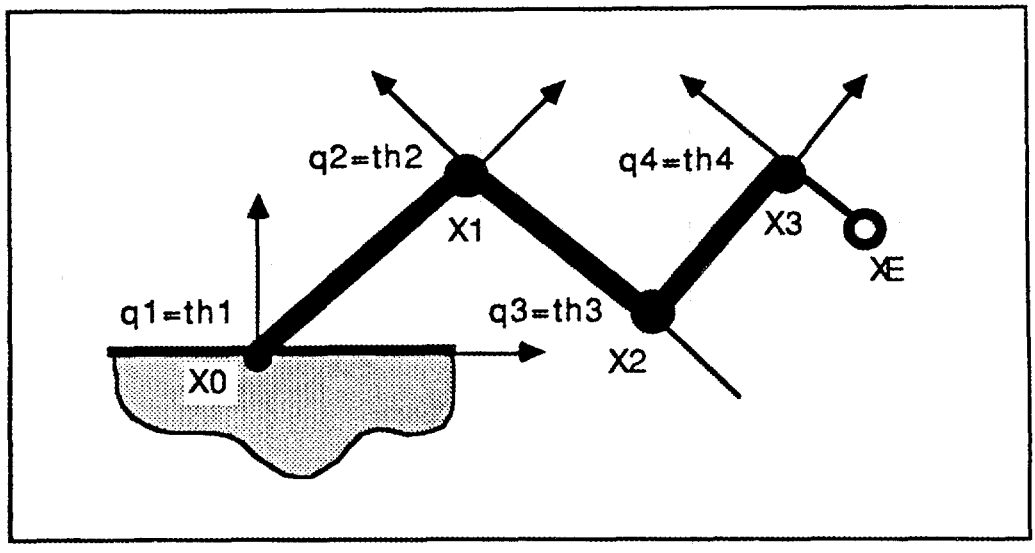

Fig. 1. Four-degree-of-freedom planar arm.

This simulation was performed by a classical sequential program. However, it is possible to improve the time performance of this inversion by parallelizing data processings. Let us recall that the robot has m degrees of freedom and that the task has n degrees of freedom. The first step of the algorithm is to select the $p = (m-n+1)$ matrices that will determine the solution. We extract in parallel m $n \times n$ matrices J_k, compute their determinants, inverses, and associated vectors Δq_k (for a more efficient on-line implementation, the explicit inversion of J_k can

be avoided by using a Gaussian elimination with partial pivoting). We keep the vectors Δq_k corresponding to the largest determinants. Then the computation of the Gramian matrix G is also made in parallel. The remaining computations are the inversion of G (which is symmetric and well-behaved) and the computation of the vectors $t^*(t_1, \ldots, t_p)$ and Δq.

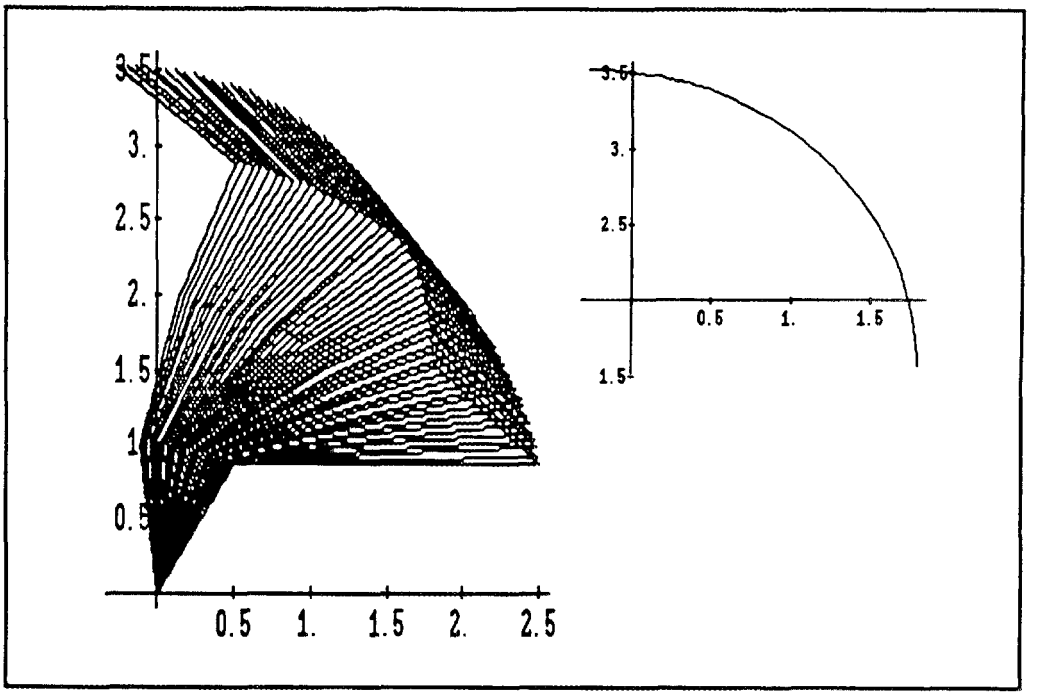

Fig. 2. Plotting of the arm and the end effector obtained by simulation.

5. GENERALIZATION OF THE ALGORITHM TO OTHER CRITERIA

In the preceding section, we have shown that it is possible to resolve the redundancy without explicitly computing the pseudo-inverse of the Jacobian matrix. Our approach was to minimize a criterion over the affine space defined by Eq. (1) that we parameterized using p vector solutions. Suppose that we now also want to minimize a potential function in order to avoid obstacles, (like in Khatib et al. [8], or to maximize a manipulability index, as defined in Yoshikawa [9]. Let us denote by $\mathcal{P}(\Delta q)$ such a criterion, expressed as a function of $\Delta q(t_1, \ldots, t_p)$, the elementary displacement in the articular space. Then, we only need to replace the problem in Eq. (8) by the new problem

$$\min \frac{1}{2}\|t_1\Delta q_1 + t_2\Delta q_2 + \cdots + t_p\Delta q_p\|^2 \quad + \quad \lambda\mathcal{P}(\Delta q(t_1, \ldots, t_p)) \tag{14}$$

where λ is a weight factor, and the real parameters t_k are still subject to the constraint $\sum_{k=1}^{p} t_k = 1$. In this case, one must note that, in general, no closed-form solution like Eq. (13) will be found. This is the subject of our future work.

6. CONCLUSION

In this paper, we have presented a alternative scheme to resolve the kinematics of a redundant robot. This scheme can be efficiently parallelized and contains well-behaved matrices for numerical processing with a computer. The future work that we envision is an extensive comparison of this method with current approaches when the task function [10] of the robot includes several other criteria as well. In particular, the application of decomposition/coordination schemes (see for example Cohen [11] and references therein) should enable us to preserve the parallel structure even with non-additive criteria such as potential and manipulability functions.

APPENDIX A

PLANAR 4 DOF REDUNDANT ARM

■ Homogeneous Transformation Matrix

```
l = 1; le =1;
mat = {{ci,-si,l*ci}, {si,ci,l*si}, {0,0,1}};
m10 = mat /.{ci → Cos[t1], si → Sin[t1]};
m21 = mat /.{ci → Cos[t2], si → Sin[t2]};
m32 = mat /.{ci → Cos[t3], si → Sin[t3]};
me3 = mat /.{ci → Cos[t4], si → Sin[t4]};
m20 = m10.m21;
m30 = m20.m32;
me0 = m30.me3;
```

■ Location of Particular Points of the Arm

```
x1 = {m10[[1,3]], m10[[2,3]]};
x2 = {m20[[1,3]], m20[[2,3]]};
x3 = {m30[[1,3]], m30[[2,3]]};
xe = {me0[[1,3]], me0[[2,3]]};
```

■ Computation of Square Sub-Matrices

```
j12 = {{D[xe[[1]],t1],D[xe[[1]],t2]},
       {D[xe[[2]],t1],D[xe[[2]],t2]}};
j23 = {{D[xe[[1]],t2],D[xe[[1]],t3]},
       {D[xe[[2]],t2],D[xe[[2]],t3]}};
j34 = {{D[xe[[1]],t3],D[xe[[1]],t4]},
       {D[xe[[2]],t3],D[xe[[2]],t4]}};
j14 = {{D[xe[[1]],t1],D[xe[[1]],t4]},
       {D[xe[[2]],t1],D[xe[[2]],t4]}};
```

■ Path Generation

■ Initializations

```
nsteps = 50;
tini = {Pi/3,-Pi/3,0,3/4 Pi};
thetastar = tini;
tlistini := {t1 → N[tini[[1]]], t2 → N[tini[[2]]],
             t3 → N[tini[[3]]], t4 → N[tini[[4]]]};
numx1 = N[x1 /. tlistini];
numx2 = N[x2 /. tlistini];
numx3 = N[x3 /. tlistini];
numxe = N[xe /. tlistini];
positions = {{0,0},numx1,numx2,numx3,numxe,
               numx3,numx2,numx1,{0,0}};
effector = {numxe};
```

■ Generation Loop

```
For[k=1, k≤ nsteps, k=k+1,
deltax = -0.02*Pi * Sin[0.01*Pi*k];
deltay = 0.02*Pi * Cos[0.01*Pi*k];

numj12 = N[j12 /. tlistini];
numj23 = N[j23 /. tlistini];
numj34 = N[j34 /. tlistini];
numj14 = N[j14 /. tlistini];

det12 = Det[numj12]^2;
det23 = Det[numj23]^2;
det34 = Det[numj34]^2;
det14 = Det[numj14]^2;

click = 0;
click = If[det12 == Min[det12,Abs[det23],Abs[det34],Abs[det14]],1,click];
click = If[det23 == Min[det12,Abs[det23],Abs[det34],Abs[det14]],2,click];
click = If[det34 == Min[det12,Abs[det23],Abs[det34],Abs[det14]],3,click];

dt12 = N[Flatten[{Inverse[numj12].{deltax,deltay},0,0}]];
dt23 = N[Flatten[{0,Inverse[numj23].{deltax,deltay},0}]];
dt34 = N[Flatten[{0,0,Inverse[numj34].{deltax,deltay}}]];
 u14 = N[Inverse[numj14].{deltax,deltay}];
dt14 = {u14[[1]],0,0,u14[[2]]};
```

```
dthii = RotateLeft[{dt12,dt23,dt34,dt14},click];
dth1 = dthii[[1]];
dth2 = dthii[[2]];
dth3 = dthii[[3]];

gram = Table[0,{t1,3},{t2,3}];
gram[[1,1]] = dth1.dth1;
gram[[2,2]] = dth2.dth2;
gram[[3,3]] = dth3.dth3;
gram[[2,1]] = gram[[1,2]] = dth1.dth2;
gram[[3,1]] = gram[[3,1]] = dth1.dth3;
gram[[2,3]] = gram[[3,2]] = dth2.dth3;
gmoinsun = Inverse[gram];
tsol = gmoinsun.{1,1,1} / {1,1,1}.gmoinsun.{1,1,1};

dthetastar = tsol[[1]] dth1 + tsol[[2]] dth2 +tsol[[3]] dth3;

thetastar = tini + dthetastar;
tini = thetastar;
numx1 = N[x1 /. tlistini];
numx2 = N[x2 /. tlistini];
numx3 = N[x3 /. tlistini];
numxe = N[xe /. tlistini];
positions = Append[positions,numx1];
positions = Append[positions,numx2];
positions = Append[positions,numx3];
positions = Append[positions,numxe];
positions = Append[positions,numx3];
positions = Append[positions,numx2];
positions = Append[positions,numx1];
positions = Append[positions,0,0];
effector = Append[effector,numxe];
];
```

▪ Plots

```
ListPlot[effector, PlotJoined → True]
ListPlot[positions, PlotJoined → True]
```

REFERENCES

1. A. Liegeois, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," *IEEE Trans. Syst Man, Cybern.*, **SMC-7(12)**, 868-871 (1977).

2. R. Penrose, "A Generalized Inverse for Matrices," Proc. Cambridge Phil. Soc. bf 51, 406-413 (1955).

3. V. Klema and A. Laub, "The Singular Value Decomposition: Its Computation and Some Applications," *Proc.IEEE Trans. on Automatic Control*, **AC-25(2)**, 164-176 (1980).

4. A. Maciejewski and C. Klein, "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments," *International Journal of Robotics Research*, **4(3)**, 109-117 (Fall 1985).

5. R. Dubey, J. Euler, and S. Babcock, "An Efficient Gradient Projection Optimization Scheme for a Seven-Degree-of-Freedom Redundant Robot with Sperical Wrist," *Proc. IEEE Int. Conf. on Robotics and Automation*, Philadelphia, Pennsylvania, pp. 28-36, April 24-29, 1988.

6. S. Hirose and S. Ma, "Redundancy Decomposition Control for Multi-Joint Manipulator," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Scottsdale, Arizona, pp. 119-124, May 14-19, 1989.

7. S. Wolfram, "Mathematica - A System for Doing Mathematics by Computer," Addison-Wesley Publishing Company, August 1988.

8. O. Khatib and J.-F. Le Maitre, "Dynamic Control of Manipulators Operating in a Complex Environment," *Proc. 3rd Int. CISM-IFToMM Symposium*, Udine, Italy, 1978.

9. T. Yoshikawa, "Manipulability and Redundancy Control of Robotic Mechanisms," *Proc. IEEE Int. Conf. on Robotics and Automation*, St. Louis, Missouri, pp. 1004-1009, March 25, 1985.

10. C. Samson, "Une Approche pour la Synthèse et l'Analyse de la Commande des Robots Manipulateurs Rigides," Internal Report I.R.I.S.A., N. 356, Rennes, France (April 1987).

11. G. Cohen, "Auxiliary Problem Principle Extended to Variational Inequalities," *J. Opt. Th. and Appl.* **59**, 325-333 (1988).

INTERNAL DISTRIBUTION

1. B. R. Appleton
2. S. M. Babcock
3. D. L. Barnett
4–8. P. F. R. Belmans
9. F. W. Depiero
10. G. de Saussure
11. J. R. Einstein
12. W. R. Hamel
13. J. N. Herndon
14. J. F. Jansen
15–19. F. C. Maienschein
20–24. R. C. Mann
25. S. A. Meacham
26–30. F. G. Pin
31. V. Protopopescu
32. D. B. Reister
33. P. F. Spelt
34. F. J. Sweeney
35. M. A. Unseren
36. H. A. Vasseur
37. A. Zucker
38. EPMD Reports Office
39–40. Laboratory Records Department
41. Laboratory Records, ORNL–RC
42. Document Reference Section
43. Central Research Library
44. ORNL Patent Section

EXTERNAL DISTRIBUTION

45. Office of Assistant Manager, Energy Research and Development, Department of Energy, Oak Ridge Operations, P.O. Box 2001, Oak Ridge, TN 37831
46. James S. Coleman ER-15, Director of Engineering and Geosciences, Office of Basic Energy Sciences, U.S. Department of Energy, Washington, DC 20545
47–51. Jean-Christophe Culioli, Laboratoire d'Automatique, E.N.S.M.P., 35, rue Saint Honore, 77305 Fontainebleau Cedex, FRANCE
52. John J. Dorning, Department of Nuclear Engineering and Physics, Thorton Hall, McCormick Rd., University of Virginia, Charlottesville, VA 22901
53. Robert M. Haralick, Boeing Clairmont Egtvedt Prof., Department of Electrical Engineering, Director, Intelligent Systems Lab, University of Washington, 402 Electrical Engineeing Bldg., FT-10, Seattle, WA 98195
54. Avi Kak, Department of Electrical Engineering, Purdue University, Lafayette, IN 47907
55. James E. Leiss, 13013 Chestnut Oak Dr., Gaithersburg, MD 20878
56. Oscar P. Manley, Office of Basic Research and Engineering, Mathematical, and Geosciences Division, U.S. DOE, MS ER-15, Washington, DC 20545
57. Mary F. Wheeler, Mathematics Department, University of Houston, 4800 Calhoun, Houston, TX 77204-3476
58. Neville Moray, Department of Mechanical and Industrial Engineering, University of Illinois, 1206 W. Green St., Urbana, IL 61801
59–68. Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831